

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE  
FAKULTA INFORMATIKY A STATISTIKY



**DISERTAČNÍ PRÁCE**



VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE

FAKULTA INFORMATIKY A STATISTIKY



Název disertační práce:

## **Vícenásobná marginalizace a její dopad na efektivnost dodavatelských řetězců**

Autor: **Ing. Jan Zouhar**

Katedra: **Katedra ekonometrie**

Obor: **Ekonometrie a operační výzkum**

Školitel: **Prof. RNDr. Ing. Petr Fiala, MBA, CSc.**



## Abstrakt:

Dvojí (příp. vícenásobná) marginalizace bývá označována za hlavní zdroj neefektivnosti decentralizovaných dodavatelských řetězců (DŘ). Podstata jevu dvojí marginalizace spočívá v tom, že pokud se články DŘ při rozhodování o cenách svého výstupu řídí postuláty mikroekonomické teorie firmy (které jsou odvozeny pro situaci, kdy firma nezávisle vyrábí a prodává přímo koncovému spotřebiteli), dojde k cenové spirále a výsledná cena koncové produkce (tzv. rovnovážná cena) bude svou přílišnou výší poškozovat jak spotřebitelský užitek, tak i zisk řetězce jako celku.

Cílem této disertační práce je zkoumat a kvantifikovat dopad efektu vícenásobné marginalizace v DŘ, které se liší svou strukturou (tj. počtem článků a vazbami mezi nimi) a povahou nákladových a poptávkových funkcí. Hlavním měřítkem dopadu je efektivnost DŘ, definovaná jako poměr mezi ziskem řetězce, jehož články se chovají podle modelu vícenásobné marginalizace, a nejvyšším ziskem, který by řetězec mohl dosáhnout celkovou koordinací cenové politiky. Kromě efektivnosti jsou sledovány i některé další vlastnosti DŘ, jako např. rozložení celkového zisku mezi jednotlivé články nebo nákladové externality v rámci řetězce.

V práci jsou popsány tři varianty modelu DŘ. První z nich je lineární model vícenásobné marginalizace (tj. model s lineárními poptávkovými a nákladovými funkcemi); pro tento zjednodušený model se podařilo analyticky odvodit vzorce popisující chování sledovaných ukazatelů. Druhý model rozšiřuje předchozí analýzu o nelineární vztahy v poptávce a nákladech; analýza je v tomto případě vedena prostřednictvím počítačových experimentů s numerickými algoritmy pro nalezení rovnovážných hodnot v řetězci. Třetí variantou je dynamický model vícenásobné marginalizace, který pomocí multiagentní simulace zkoumá průběh zmiňované cenové spirály.

**Klíčová slova:** *dodavatelské řetězce, vícenásobná marginalizace, multiagentní simulace.*

## Abstract:

Double (or multiple) marginalization is often identified as the main source of a decentralized supply chain's (SC's) inefficiency. In its core lies the fact that if the agents constituting the SC choose their output prices according to the golden rule of profit maximization (that normally applies to a single firm that produces independently and sells directly to the end consumer), the prices in the SC tend to spiral up to an inefficient (equilibrium) level where both the consumer surplus and the SC's total profit are diminished.

The aim of this paper is to analyze and quantify the impact of multiple marginalization on the behaviour of SC's that vary with respect to their structure (i.e. the number of agents and the links between them) and the shape of their cost and demand functions. The main gauge of this impact is the efficiency of a SC, defined as the ratio of the profit of a SC whose agents behave according to the model of multiple marginalization, and the potential profit of the SC (i.e. the maximum profit attainable under the conditions of complete coordination of prices within the chain). Besides efficiency, some other properties of a SC are studied, e.g. the distribution of the SC's profit among the individual agents or cost externalities within the SC.

Three different models of multiple marginalization are studied in the paper. The first one is a linear model of multiple marginalization (i.e. a model with linear demand and cost functions); in this simplified setting we derived explicit formulae for values of the studied indicators. The second model is analogous to the first one only that it allows for non-linear demand and cost functions; in this case, the analysis is carried out using computer experiments with numeric algorithms. The last one is a dynamic model of multiple marginalization which studies the abovementioned price spiral through multi-agent simulation.

**Keywords:** *supply chains, multiple marginalization, multi-agent simulation.*

## Prohlášení:

Prohlašuji, že jsem disertační práci na téma „Vícenásobná marginalizace a její dopad na efektivnost dodavatelských řetězců“ vypracoval samostatně. Veškerou použitou literaturu a další podkladové materiály uvádím v seznamu použité literatury.

V Praze dne 13.10.2009

---

Jan Zouhar



# OBSAH

|                                                                           |           |
|---------------------------------------------------------------------------|-----------|
| ÚVOD .....                                                                | 13        |
| <b>1 OD DVOJÍ K VÍCENÁSOBNÉ MARGINALIZACI .....</b>                       | <b>15</b> |
| 1.1 HORŠÍ NEŽ MONOPOL .....                                               | 15        |
| 1.2 MODELOVÝ RÁMEC PRO ZKOUMÁNÍ VÍCENÁSOBNÉ MARGINALIZACE .....           | 17        |
| <b>2 LINEÁRNÍ MODEL VÍCENÁSOBNÉ MARGINALIZACE .....</b>                   | <b>21</b> |
| 2.1 SÉRIOVÉ DODAVATELSKÉ ŘETĚZCE .....                                    | 21        |
| 2.2 PARALELNÍ DODAVATELSKÉ ŘETĚZCE .....                                  | 28        |
| 2.3 KOMBINOVANÉ DODAVATELSKÉ ŘETĚZCE .....                                | 34        |
| <b>3 DODAVATELSKÉ ŘETĚZCE S NELINEÁRNÍMI FUNKCEMI .....</b>               | <b>43</b> |
| 3.1 NUMERICKÉ ALGORITMY PRO ŘETĚZCE S NELINEÁRNÍMI FUNKCEMI .....         | 44        |
| 3.1.1 Algoritmus pro nalezení rovnovážných cen v sériovém řetězci .....   | 45        |
| 3.1.2 Algoritmus pro nalezení rovnovážných cen v paralelním řetězci ..... | 51        |
| 3.2 POUŽITÉ TVARY FUNKCÍ PRO POPTÁVKU A NÁKLADY .....                     | 57        |
| 3.2.1 Poptávková funkce .....                                             | 57        |
| 3.2.2 Nákladové funkce .....                                              | 59        |
| 3.3 VÝPOČETNÍ EXPERIMENTY .....                                           | 60        |
| 3.3.1 Nelinearita v poptávkové funkci .....                               | 61        |
| 3.3.2 Nelinearita v nákladových funkcích .....                            | 72        |
| 3.3.3 Nelinearita v poptávkové funkci i nákladových funkcích .....        | 80        |
| 3.3.4 Shrnutí výsledků výpočetních experimentů .....                      | 85        |
| 3.4 MODEL S DISKRÉTNÍMI CENAMI .....                                      | 86        |
| 3.4.1 Sériové řetězce .....                                               | 87        |
| 3.4.2 Paralelní řetězce .....                                             | 90        |
| <b>4 DYNAMICKÉ MODEL VÍCENÁSOBNÉ MARGINALIZACE .....</b>                  | <b>95</b> |
| 4.1 MULTIAGENTNÍ SYSTÉMY .....                                            | 96        |
| 4.2 MULTIAGENTNÍ MODEL SÉRIOVÉHO ŘETĚZCE .....                            | 99        |
| 4.2.1 Model racionálního uvažování agentů .....                           | 100       |
| 4.2.2 Dynamika cen v sériovém řetězci .....                               | 107       |

---

|                 |                                                                          |     |
|-----------------|--------------------------------------------------------------------------|-----|
| 4.3             | MULTIAGENTNÍ MODELY PARALELNÍCH ŘETĚZCŮ .....                            | 109 |
| 4.3.1           | Dynamika cen v paralelních řetězcích s minimální mírou informace .....   | 110 |
| 4.3.2           | Dynamika cen v paralelních řetězcích s rozšířenými znalostmi .....       | 111 |
| 4.4             | SHRNUTÍ VÝSLEDKŮ MULTIAGENTNÍCH SIMULACÍ.....                            | 114 |
| ZÁVĚR .....     |                                                                          | 115 |
| LITERATURA..... |                                                                          | 119 |
| PŘÍLOHY.....    |                                                                          | 121 |
| P.1             | POPIS UŽIVATELSKÉHO ROZHRAŇÍ .....                                       | 122 |
| P.1.1.          | Zadávatelství vlastností dodavatelského řetězce a parametrů výpočtu..... | 122 |
| P.1.2.          | Hledání rovnovážných cen pomocí uživatelského rozhraní .....             | 125 |
| P.1.3.          | Multiagentní simulace pomocí uživatelského rozhraní .....                | 131 |
| P.2             | ZDROJOVÉ KÓDY NĚKTERÝCH FUNKCÍ .....                                     | 133 |

## SEZNAM GRAFŮ, OBRÁZKŮ A TABULEK

|                                                                                     |    |
|-------------------------------------------------------------------------------------|----|
| Obrázek 1-1: Příklad struktury sériového, paralelního a kombinovaného řetězce ..... | 17 |
| Obrázek 2-1: Sériový dodavatelský řetězec.....                                      | 21 |
| Obrázek 2-2: Příklad 2 – sériový dodavatelský řetězec .....                         | 26 |
| Tabulka 2-3: Příklad 2 – rovnovážné hodnoty I .....                                 | 27 |
| Tabulka 2-4: Příklad 2 – rovnovážné hodnoty II .....                                | 27 |
| Obrázek 2-5: Příklad 3 – sériový dodavatelský řetězec .....                         | 28 |
| Tabulka 2-6: Příklad 3 – rovnovážné hodnoty .....                                   | 28 |
| Obrázek 2-7: Příklad 4 – paralelní dodavatelský řetězec.....                        | 28 |
| Tabulka 2-8: Příklad 4 – rovnovážné hodnoty .....                                   | 30 |
| Obrázek 2-9: Paralelní dodavatelský řetězec.....                                    | 30 |
| Obrázek 2-10: Příklad kombinovaného dodavatelského řetězce .....                    | 35 |
| Obrázek 2-11: Stejně efektivní struktury kombinovaného řetězce .....                | 39 |
| Graf 3-1: Příklad 6 – poptávková a nákladová funkce .....                           | 48 |
| Graf 3-2: Příklad 6 – zisková funkce článku 2 při různých vstupních cenách.....     | 49 |
| Graf 3-3: Příklad 6 – cenová reakční funkce a funkce maximální výše zisku .....     | 50 |
| Graf 3-4: Různé tvary poptávkové funkce $abda$ a z nich plynoucí tržby.....         | 57 |
| Tabulka 3-5: Typy nastavení parametrů pro konvexní klesající funkci $abda$ .....    | 58 |
| Graf 3-6: Různé tvary poptávkové funkce $abda$ a z nich plynoucí tržby.....         | 58 |
| Graf 3-7: Příklad 7 – poptávkové funkce .....                                       | 62 |
| Graf 3-8: Příklad 7 – efektivnost řetězce a rozložení zisku .....                   | 62 |
| Graf 3-9: Příklad 8 – efektivnost řetězce a rozložení zisku .....                   | 63 |
| Graf 3-10: Příklad 9 – individuální zisky článků .....                              | 64 |
| Graf 3-11: Příklad 9 – efektivnost řetězce a rozložení zisku .....                  | 65 |
| Graf 3-12: Příklad 9 – podíly na zisku při nákladových úsporách článků 1 a 5 .....  | 65 |
| Graf 3-13: Příklad 10 – zisky řetězce při přesunu výrobního procesu .....           | 66 |

|                                                                                                   |     |
|---------------------------------------------------------------------------------------------------|-----|
| Graf 3-14: Příklad 11 – efektivnost řetězce a rozložení zisku .....                               | 67  |
| Graf 3-15: Příklad 12 – efektivnost řetězce a rozložení zisku .....                               | 68  |
| Tabulka 3-16: Porovnání efektivnosti sériových a paralelních řetězců .....                        | 69  |
| Graf 3-17: Příklad 13 – efektivnost řetězců a rozložení zisků .....                               | 70  |
| Graf 3-18: Příklad 13 – efektivnost a rozložení zisku při nákladových úsporách článku 1 a 5 ..... | 70  |
| Graf 3-19: Příklad 14 – individuální zisky článků.....                                            | 71  |
| Graf 3-20: Příklad 15 – nákladové funkce článků .....                                             | 73  |
| Graf 3-21: Příklad 15 – efektivnost a rozložení zisku .....                                       | 73  |
| Graf 3-22: Příklad 15 – koncová cena a poptávané množství.....                                    | 74  |
| Graf 3-23: Příklad 16 – nákladová funkce článku 3.....                                            | 75  |
| Graf 3-24: Příklad 16 – efektivnost a rozložení zisku .....                                       | 75  |
| Graf 3-25: Příklad 17 – efektivnost a rozložení zisku (úspory z rozsahu).....                     | 76  |
| Graf 3-26: Příklad 17 – efektivnost a rozložení zisku (záporné úspory z rozsahu).....             | 77  |
| Graf 3-27: Příklad 18 – efektivnost a rozložení zisku .....                                       | 78  |
| Graf 3-28: Příklad 19 – efektivnost a rozložení zisku .....                                       | 79  |
| Graf 3-29: Příklad 20 – efektivnost a rozložení zisku .....                                       | 80  |
| Graf 3-30: Příklad 21 – poptávkové a nákladové funkce při různých hodnotách parametrů .....       | 81  |
| Graf 3-31: Příklad 21 – efektivnost sériového řetězce .....                                       | 82  |
| Graf 3-32: Příklad 21 – ziskové podíly (sériový řetězec) .....                                    | 82  |
| Graf 3-33: Příklad 21 – efektivnost paralelního řetězce .....                                     | 84  |
| Graf 3-34: Příklad 21 – ziskové podíly (paralelní řetězec) .....                                  | 84  |
| Tabulka 3-35: Příklad 22 – porovnání rovnovážných hodnot při spojitých a diskrétních cenách.....  | 88  |
| Graf 3-36: Příklad 22 – dopad diskrétních cen v různě dlouhých řetězcích.....                     | 89  |
| Tabulka 3-37: Příklad 23 – různé rovnovážné kombinace cen .....                                   | 93  |
| Graf 3-38: Příklad 23 – porovnání rovnovážných bodů .....                                         | 93  |
| Graf 4-1: Apriorní představy agentů o individuálních poptávkových funkcích.....                   | 102 |
| Graf 4-2: Průběh procesu částečného přizpůsobení představ o poptávce.....                         | 104 |
| Graf 4-3: Reakce poptávaného množství na výstupní cenu agenta.....                                | 106 |
| Graf 4-4: Příklad 24 – vývoj cen a zisků v řetězci se třemi články.....                           | 108 |
| Graf 4-5: Příklad 24 – vývoj cen a zisků v řetězci se čtyřmi články .....                         | 108 |
| Graf 4-6: Příklad 24 – vývoj cen a zisků v řetězci s pěti články .....                            | 109 |
| Graf 4-7: Příklad 25 – vývoj cen v paralelním řetězci s minimální informací .....                 | 111 |
| Graf 4-8: Příklad 26 – vývoj cen a zisku v řetězci s rozšířenou informací (3 články) .....        | 113 |
| Graf 4-9: Příklad 26 – vývoj cen a zisku v řetězci s rozšířenou informací (5 článků).....         | 113 |
| Obrázek P-1: Hlavní okno uživatelského rozhraní .....                                             | 123 |
| Obrázek P-2: Pomocné grafy v uživatelském rozhraní .....                                          | 124 |
| Obrázek P-3: Grafický výstup pro výpočet rovnováhy v sériovém řetězci.....                        | 126 |
| Obrázek P-4: Grafický výstup pro výpočet rovnováhy v paralelním řetězci .....                     | 128 |
| Obrázek P-5: Grafický výstup pro výpočet rovnováhy v příkladu 23 (první okno) .....               | 130 |
| Obrázek P-6: Grafický výstup pro výpočet rovnováhy v příkladu 23 (druhé okno) .....               | 130 |
| Obrázek P-7: Grafický výstup pro multiagentní simulaci .....                                      | 131 |



# ÚVOD

**Dvojitá marginalizace** (double marginalization, též překládáno jako dvojitá zisková marže) je pojem, který byl součástí teorie dodavatelských řetězců již takřka od jejich počátků a který je dodnes součástí osnovy všech universitních kurzů, jež se dodavatelskými řetězci nebo obecně produkčními systémy zabývají. Důležitost tohoto pojmu spočívá v tom, že popisuje jednu z hlavních slabin dodavatelského řetězce jakožto formy produkčního systému, a sice jeho náchylnost k prudkému snižování alokační efektivnosti a schopnosti přinášet zisk v případě, že spolu jednotlivé články řetězce vzájemně nespolupracují při tvorbě cenové politiky. Dvojitá marginalizace je tak jedním z hlavních hnacích motorů pro hledání prostředků účinného formování a koordinace dodavatelských řetězců.

Nutno však podotknout, že značná část literatury, která se problémem dvojitá marginalizace zabývá, se zaměřuje právě na možnosti zmírnění jeho dopadu, spíše než na analýzu a kvantifikaci efektu dvojitá marginalizace jako takového. Cílem takových prací bývá nalezení nejvhodnějšího praktického postupu při řízení dodavatelských řetězců. Práce teoretického rázu o vlastnostech dvojitá marginalizace potom většinou sdílejí jeden podstatný společný rys: zabývají se dodavatelskými řetězci tvořenými pouze dvěma navazujícími články – jedná se tedy o analýzu vztahů typu dodavatel-výrobce, velkoobchodník-maloobchodník a podobně<sup>1</sup>. Jedním ze stěžejních záměrů při zrodu této práce bylo zkoumat, jaké důsledky přináší dvojitá zisková marže v řetězcích, které jsou rozsáhlejší co do počtu obsažených nezávislých firem (tj. mají více než dva články) a které případně vykazují různé struktury vazeb mezi články. Cílem této disertační práce je tedy popsat vlastnosti a důsledky jevu, který bude nadále přiléhavěji označován jako **vícenásobná marginalizace**.

V první kapitole bude představen obecný modelový rámec modelu dodavatelského řetězce, který bude sloužit jako východisko pro zkoumání vícenásobné marginalizace v průběhu celé práce. Tento modelový rámec je přímým rozšířením klasického konceptu, kterým se popisuje dvojitá zisková marže. Kromě klasického „sériového“ pojetí dodavatelského řetězce budou představeny řetězce s bohatší architekturou.

Druhá kapitola se věnuje speciálnímu případu popsaného modelu, který se vyznačuje lineárními nákladovými funkcemi a lineární funkcí poptávky po koncové produkci. Takové omezení umožní odvodit některé explicitní vzorce pro kvantitativní popis efektu vícenásobné marginalizace a vlastností uvažovaných dodavatelských řetězců.

---

<sup>1</sup> Viz např. [1], [15].

Třetí kapitola tematicky přímo navazuje na kapitolu druhou a jejím cílem je zkoumat, do jaké míry zůstanou poznatky z druhé kapitoly relevantní, pokud upustíme od předpokladu linearity uvažovaných nákladových a poptávkových funkcí. Jak bude ukázáno, přejdeme-li od funkcí lineárních k funkcím obecnějším, ztrácíme takřka okamžitě možnost odvodit teoreticky jakékoli exaktní výsledky. Pro potřeby zkoumání obecnějšího modelu jsem proto navrhl numerické algoritmy a implementoval je v programovacím jazyce výpočetního prostředí Matlab společnosti Mathworks<sup>2</sup>. Spolu s jednoduchým uživatelským prostředím jsou výsledné programy součástí přiloženého CD. Kapitola obsahuje kromě popisu těchto algoritmů především výsledky a zhodnocení výpočetních experimentů, které byly vykonány za použití zmíněných programů a které poukazují na vlastnosti různých forem řetězců.

Čtvrtá kapitola se zabývá dynamickým modelem vícenásobné marginalizace. Jejím cílem je zkoumat, jakým způsobem dochází k postupnému prosazení efektu vícenásobné marginalizace v dodavatelských řetězcích, jejichž články mají omezené informace o trhu koncové produkce řetězce a o ostatních člancích. Klíčovou otázkou zde je, zda vůbec a případně jak rychle dojde v takovém dodavatelském řetězci k nastolení situace, kterou popisovaly předchozí kapitoly. Za účelem zkoumání dynamiky takovýchto řetězců jsem proto vytvořil reprezentaci řetězce jakožto multiagentního systému (opět v prostředí Matlab) a pomocí počítačové simulace jsem zkoumal vývoj chování jednotlivých článků v různých scénářích dodavatelských řetězců.

Přiložené CD obsahuje elektronickou podobu této disertační práce, a jak bylo řečeno výše, rovněž počítačové implementace všech popsaných algoritmů a uživatelské rozhraní pro snadnější obsluhu některých z nich. Pro spuštění všech funkcí i uživatelského rozhraní je třeba mít nainstalováno výpočetní prostředí Matlab společnosti Mathworks verze 6.1 nebo vyšší. Stručný návod ke spouštění a ovládání funkcí je uveden první příloze k této disertační práci. V druhé příloze předkládám zdrojové kódy klíčových částí softwarové realizace popsaných algoritmů (jedná se pouze o ty části kódů, na které je ve vlastním textu práce odkazováno).

---

<sup>2</sup> Domovská stránka výrobce, na které je k dispozici řada dalších informací, je <http://www.mathworks.com/>.

# 1 OD DVOJÍ K VÍCENÁSOBNÉ MARGINALIZACI

Cílem této kapitoly je představit pojem vícenásobné marginalizace a vybudovat terminologii a modelový rámec pro její zkoumání. Budeme přitom přímo vycházet z tradiční myšlenky dvojí marginalizace a modelu, na kterém se tato demonstruje. Následující oddíl proto koncept dvojí marginalizace ve stručnosti přiblíží (podle [2], [5] a [15]).

## 1.1 Horší než monopol

Dvojí marginalizace popisuje hrozbu nízké efektivity dodavatelského řetězce coby výrobní struktury v případě, že je řetězec tvořen nezávislými firmami, které nespolupracují při tvorbě cen a snaží se sledovat výhradně svoje zájmy. Jádro problému shrnuje vtipně a zároveň trefně autor [2] pomocí krátké ekonomické anekdoty:

„Víš, co je horší, než monopolista na trhu?“

„Ne.“

„Dva monopolisté.“

Jedním ze základních poznatků mikroekonomické teorie firmy je skutečnost, že s monopolní silou na straně prodejce zpravidla roste tržní cena, klesá objem vyráběného a prodaného zboží a spotřebitelský přebytek a rostou náklady mrtvé váhy. Není proto nikterak překvapující, že pokud se na místě jednoho monopolisty objeví dodavatelský řetězec tvořený dvěma články v řadě (např. výrobcem a distributorem), kteří oba disponují monopolní silou, bude tento efekt ještě umocněn. Na rozdíl od případu jednoho monopolisty, který díky monopolní síle realizuje větší zisk, se však může monopolní síla článků v dodavatelském řetězci nakonec obrátit i proti nim samotným. Pokud totiž oba články hledí pouze na svůj individuální zisk a nesnaží se s ostatními články vytvořit cenovou politiku výhodnou z hlediska celého řetězce, spadnou do pastí dvojí marginalizace a volí příliš vysokou cenu, která vyústí nejen ve snížený společenský přebytek, ale i v menší zisk obou článků. Celý proces ukážeme na jednoduchém příkladě.

**PŘÍKLAD 1:** Představme si dvě firmy, výrobce a distributora, kteří spolu tvoří dodavatelský řetězec. Náklady výrobce na jednotku produkce činí 20 Kč, náklady distributora jsou zanedbatelné (nulové). Tržní poptávka po uvažovaném zboží odpovídá předpisu

$$q = 100 - p_D,$$

kde  $q$  je poptávané množství a  $p_D$  je cena, za kterou distributor zboží prodává. Distributor chce maximalizovat svůj zisk, který je součinem poptávaného množství a ziskové marže (která je rozdílem mezi cenou stanovenou výrobcem a cenou, za kterou distributor zboží prodává), neboli

$$\pi_D = (100 - p_D) \cdot (p_D - p_M),$$

kde  $p_M$  je cena požadovaná výrobcem (*manufacturer*). Při libovolné ceně stanovené výrobcem je tedy zisk distributora ryze konkávní funkce jím určené koncové ceny zboží, jejíž maximum najdeme vynulováním první derivace této funkce podle  $p_D$ , odkud obdržíme vztah

$$p_D = \frac{1}{2} \cdot (100 + p_M), \quad (1.1)$$

kterým se bude distributor při volbě koncové ceny řídit. Při dané ceně  $p_M$  bude potom poptávané množství činit (dosazením předchozího vztahu do poptávkové funkce)

$$q = \frac{1}{2} \cdot (100 - p_M).$$

Zisková funkce výrobce lze pak opět vyjádřit jako součin poptávaného množství a ziskové marže (kterou tentokrát představuje převis výstupní ceny nad jednotkovými náklady):

$$\pi_M = \frac{1}{2} \cdot (100 - p_M) \cdot (p_M - 20),$$

její maximum získáme opět z podmínky prvního řádu, nachází se v bodě 60. Bude-li výrobce volit tuto cenu, pak distributor bude podle (1.1) volit cenu 80 Kč, poptávané množství bude 20 kusů, distributor realizuje zisk ve výši  $20 \cdot (80 - 60) = 400$  Kč a zisk výrobce bude celkem 800 Kč. Dohromady tedy dosáhnou oba články zisku 1200 Kč. S danými možnostmi je ovšem řetězec schopen dosahovat zisku vyššího. Celkový zisk obou článků (neboli zisk celého řetězce  $\pi_{SC}$ , kde SC je zkratkou *supply chain*) můžeme vyjádřit jako

$$\pi_{SC} = \pi_D + \pi_M = (100 - p_D) \cdot (p_D - 20),$$

maximum této funkce je v bodě  $p_D = 60$  a nabývá hodnoty 1600, tedy řetězec může dosahovat zisku až o 400 Kč vyššího.

Ukazuje se, že snaha obou článků maximalizovat svůj individuální zisk nevedla nakonec k dosažení maximálního zisku ani jednoho z nich. Tento výsledek se může zdát na první pohled zarážející. Nabízí se otázka, proč články nevolí nižší ceny, když by to oběma přineslo větší zisk? Důvod je ten, že v případě, že články nespoluupracují při tvorbě cen, nejsou nižší ceny stabilní v tom smyslu, že např. distributorovi se jeví jako výhodnější při aktuální ceně vstupu a poptávkové funkci svoji výstupní cenu zvednout a dosáhnout tak většího individuálního zisku, v dané chvíli tedy z pohledu distributora není nižší úroveň ceny

optimální pro dosažení jeho individuálního zisku. Ceny proto mají tendenci postupně vzrůst až na výsledné hodnoty  $p_M = 60$  a  $p_D = 80$ , které lze označit jako **ceny rovnovážné**.

Chování článků popisované dvojí marginalizací tak odpovídá situaci, kdy články nepřizpůsobují svou cenovou politiku skutečnosti, že jsou součástí dodavatelského řetězce, nýbrž rozhodují se, jako kdyby byli sami monopolisty na svém trhu – odhadují individuální funkci poptávky po svém výstupu (tj. vztah volené ceny a poptávaného množství) a při porovnání s nákladovými funkcemi se snaží volit cenu, která optimalizuje individuální zisk. Toto chování, které se může na první pohled zdát jako racionální a které by pro firmu prodávající přímo koncovému spotřebiteli skutečně vedlo k optimalizaci jejího zisku, však způsobí v dodavatelském řetězci cenovou spirálu a cena koncové produkce je vytlačena na nepřiměřeně vysokou cenu. V kapitole 4 bude provedeno několik experimentů, které ukazují, jakým způsobem a tempem konvergují ceny ke svým rovnovážným hodnotám v různých dodavatelských řetězcích, jejichž články optimalizují svůj individuální zisk právě popsáním způsobem za podmínek omezené informace o parametrech dodavatelského řetězce a poptávkové funkce.

Problém dvojí marginalizace a jeho negativní dopad na zisk článků podél celého řetězce byl popudem pro poměrně rozsáhlý výzkum způsobů, kterými lze tento jev potlačit. Nejčastěji se píše o prostředcích tzv. vertikální integrace, ať už prostřednictvím společného vlastnictví článků nebo závazných kontraktů, např. na bázi franchisingu (viz např. [5]). Zkoumání ani podrobný popis těchto prostředků však není náplní této práce.

## 1.2 Modelový rámec pro zkoumání vícenásobné marginalizace

V následujících kapitolách se pokusíme kvantifikovat dopad dvojí marginalizace v různých formách dodavatelských řetězců. Budeme mimo jiné zkoumat, jak se efekt dvojí marginalizace vyvíjí, pokud se zvažovaný řetězec začne „prodlužovat“, tj. pokud je tvořen více než dvěma články. V případě takových řetězců budeme namísto dvojí marginalizace hovořit o marginalizaci vícenásobné.

Při rostoucím počtu článků mohou vznikat různé struktury vazeb článků v řetězci. Budeme rozlišovat základní dva druhy vazeb mezi články: vazby typu one-to-one a vazby typu many-to-one. Podle použitých vazeb pak budeme rozlišovat následující typy řetězců:

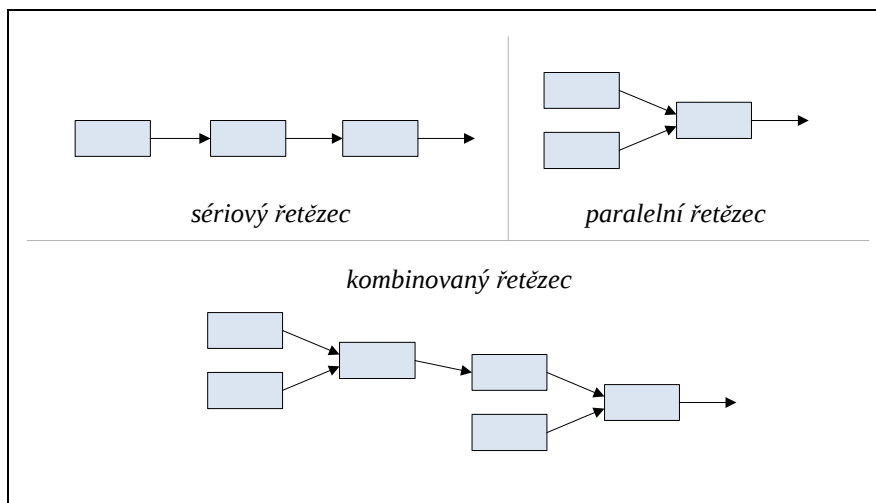
- **sériový** řetězec – obsahuje pouze vazby one-to-one,
- **paralelní** řetězec – je tvořen pouze jednou vazbou typu many-to-one,
- **kombinovaný** řetězec – může kombinovat oba druhy vazeb.

Obrázek 1-1 na následující straně uvádí příklady všech tří typů řetězců.

Při zkoumání řetězců paralelních a kombinovaných je velmi podstatné, jakým způsobem jsou definovány vztahy ve vazbách typu many-to-one. V celé této práci budeme chápat vazbu many-to-one následovně: články na začátku vazby (nazveme je *dodavateli*)

dodávají článku na konci vazby *různé* komponenty vstupující do jeho výrobního procesu; výstup dodavatelů je tedy v komplementárním postavení, dodavatelé si vzájemně nekonkurují. V rámci dodávky svého výstupu jsou všechny články v monopolním postavení. To je velký rozdíl oproti modelům v [4] a [17], které se zabývají podobnými modely s tím, že vazby many-to-one představují dodavatele *stejného* výstupu, dodavatelé tedy tvoří oligopolní tržní strukturu.

**Obrázek 1-1: Příklad struktury sériového, paralelního a kombinovaného řetězce**



Shrňme ještě jednou základní předpoklady modelu vícenásobné marginalizace, které se týkají vlastností jednotlivých článků v řetězci a jejich chování:

- Všechny články disponují v rámci odbytu svého výstupu monopolní silou. Mohou tedy rozhodovat o výši ceny, za kterou prodávají svou produkci.
- Články se snaží optimalizovat svůj individuální zisk, a to způsobem, který je typický pro firmy samostatně vyrábějící a prodávající na trhu koncové produkce: odhadují funkci poptávky po svém výstupu a porovnávají ji s funkcí nákladovou. Nijak vzájemně nekooperují při tvorbě cenové politiky ani nezohledňují své postavení v rámci dodavatelského řetězce.

Sledovanou vlastností dodavatelského řetězce bude především schopnost vytvářet zisk. Abychom mohli kvantifikovat efekt vícenásobné marginalizace, zavedeme ukazatel, který bude vyjadřovat, nakolik se součet zisků v celém řetězci kvůli marginalizačnímu efektu odchýlil od maximálního zisku, kterého by mohl potenciálně řetězec dosahovat. Situaci, kdy se články chovají podle předpokladů modelu vícenásobné marginalizace (tj. sledují popsáním způsobem výhradně svůj individuální zisk), budeme nazývat **decentralizovaným dodavatelským řetězcem**. Pokud se naopak články centralizovaným způsobem rozhodují o cenách tak, aby byl maximalizován zisk řetězce jako celku (což předpokládá nějakým způsobem centralizovanou koordinaci cenové politiky uvnitř řetězce), budeme hovořit o **centralizovaném dodavatelském řetězci** (stejně značení používá např. [15]).

Zisk decentralizovaného řetězce  $\pi_{DSC}$  (*DSC* je zkratkou pro *decentralized supply chain*) pak bude představovat součet individuálních zisků všech článků decentralizovaného řetězce, tj. zisků odpovídajících rovnovážným cenám. Zisk řetězce centralizovaného  $\pi_{CSC}$  je maximální zisk, kterého může řetězec jako celek dosáhnout. Měřítkem efektu vícenásobné marginalizace bude poměr těchto dvou zisků,

$$\varepsilon = \frac{\pi_{DSC}}{\pi_{CSC}}, \quad (1.2)$$

který budeme nazývat **efektivnost decentralizovaného řetězce**.

Klíčovým momentem pro model vícenásobné marginalizace je rozhodování článků o cenách, které požadují za svou produkci. Je zřejmé, že díky výrobnímu procesu se povaha produkce, která putuje podél jednotlivých vazeb řetězcem, mění. Pro účely modelování je ovšem potřeba vztáhnout ceny stanovené všemi články ke stejným jednotkám. Cena určená jednotlivými články bude tedy ve všech následujících modelech spojena s takovým objemem jejich výstupu, který je potřeba na jednu jednotku koncové produkce. Je-li např. jedním z článků dodavatel pneumatik a koncovým výrobkem řetězce je automobil, budeme za cenu výstupu dodavatele pneumatik považovat cenu požadovanou za 5 ks pneumatik. Stejnou konvenci použijeme i pro nákladové funkce jednotlivých článků, jednotkou množství bude opět takový objem produkce každého z článků, který je třeba na výrobu jednotky koncového produktu.

V následující kapitole se omezíme na modely řetězců, kde je funkce poptávky po koncové produkci lineární (podobně jako v příkladu 1) a stejně tak vykazují lineární tvar i nákladové funkce jednotlivých řetězců, tj. články vyrábí s konstantními jednotkovými náklady. Tyto zjednodušující předpoklady nám umožní odvodit explicitní vzorce pro sledované ukazatele.

Pokud je mi známo, paralelní a kombinované řetězce nebyly dosud podrobeny podobné kvantitativní analýze. Všechna předkládaná tvrzení týkající se těchto řetězců i jejich důkazy jsou tudíž původní. Některé ze závěrů, které se týkají řetězců sériových, se v literatuře již objevily (nejznámější je patrně poměr mezi ziskovou marží článků sériového řetězce, který je jako obecný princip pro řetězce se dvěma články prvně objasněn v [1], dále viz např. [4] nebo [17]); odvození uvedených tvrzení o sériových řetězcích je ovšem rovněž původní a koncipované tak, aby vycházelo ze stejného modelového rámce a vůbec způsobu uvažování, který je použit pro analýzu řetězců s paralelní a kombinovanou architekturou.



## 2 LINEÁRNÍ MODEL VÍCENÁSOBNÉ MARGINALIZACE

V celé kapitole budeme předpokládat, že poptávka po koncové produkci dodavatelského řetězce je deterministická a lineární a má tvar

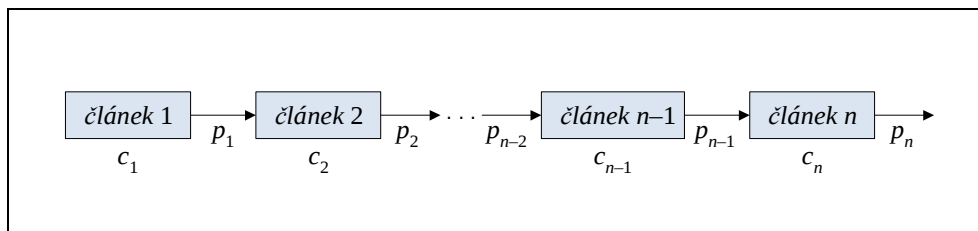
$$q(p) = a - b \cdot p, \quad (2.1)$$

kde  $a$  a  $b$  jsou kladná reálná čísla a  $p$  je cena koncové produkce, kterou určuje poslední, výstupní článek řetězce. Výrobní procesy jednotlivých článků řetězce budou probíhat s konstantními nezápornými jednotkovými náklady, jejich výši budeme označovat indexovaným písmenem  $c$ .

### 2.1 Sériové dodavatelské řetězce

Obecnou strukturu sériového dodavatelského řetězce, který obsahuje celkem  $n$  článků, zachycuje obrázek 2-1. Každý  $i$ -tý článek nakupuje produkci od článku  $i-1$  za jednotkovou cenu  $p_{i-1}$ , s produkcí provede nějaký transformační proces s jednotkovými náklady  $c_i$  a prodává následujícímu článku  $i+1$  za cenu  $p_i$ . Výjimkami jsou články 1 a  $n$ , článek 1 prvotní produkci pouze „vytváří“ s náklady  $c_1$  (čili od nikoho nenakupuje), článek  $n$  produkci prodává koncovému spotřebiteli (za cenu  $p_n$ ).

Obrázek 2-1: Sériový dodavatelský řetězec



Zisk celého řetězce  $\pi_{SC}^S$  (horní index  $S$  značí, že jde o řetězec sériový) lze vyjádřit jako součin poptávaného množství a rozdílu mezi prodejní cenou koncové produkce a součtem jednotkových nákladů podél řetězce:

$$\pi_{SC}^S = \left( p_n - \sum_{i=1}^n c_i \right) \cdot (a - b \cdot p_n) = -b \cdot p_n^2 + \left( a + b \cdot \sum_{i=1}^n c_i \right) \cdot p_n - a \cdot \sum_{i=1}^n c_i. \quad (2.2)$$

Za tímto vztahem stojí následující jednoduchá úvaha: všechny články vyrábí stejný objem produkce roven  $a - b \cdot p_n$  (vzhledem k tomu, jak jsme objem produkce každého článku definovali v oddílu 1.2). Objem peněžních prostředků, které řetězec celkem získá od spotřebitelů, je násobkem tohoto množství a ceny  $p_n$ . Z těchto prostředků je potřeba ještě uhradit všechny náklady produkce vzniklé podél řetězce, které jsou dány součinem vyráběného množství a součtu jednotkových nákladů. Zbylé finanční prostředky se pak „nemohou nikam ztratit“, jsou prostřednictvím cen uvnitř řetězce rozděleny mezi jednotlivé články. Ze vztahu (2.2) odvodíme jednoduše následující tvrzení:

**TVRZENÍ 2.1:** V lineárním modelu dosahuje centralizovaný sériový řetězec zisku

$$\pi_{CSC}^S = \frac{b}{4} \cdot \left( \frac{a}{b} - \sum_{i=1}^n c_i \right)^2. \quad (2.3)$$

**DŮKAZ:** Vzhledem k tomu, že v (2.2) jsou  $a$ ,  $b$  kladné a  $c_i$  nezáporné konstanty, je celkový zisk řetězce ryze konkávní funkcí jedné proměnné  $p_n$ . Centralizovaný řetězec bude proto volit cenu  $p_n$  odpovídající ostrému globálnímu maximu této funkce. Položíme-li derivaci ziskové funkce rovnou nule, dostaneme cenu  $p_n$  maximalizující zisk jako

$$p_n^{opt} = \frac{1}{2} \cdot \left( \frac{a}{b} + \sum_{i=1}^n c_i \right).$$

Této ceně odpovídá podle poptávkové funkce (2.1) poptávané množství výstupu

$$q(p_n^{opt}) = \frac{b}{2} \cdot \left( \frac{a}{b} - \sum_{i=1}^n c_i \right),$$

hodnotu ziskové funkce v bodě optima pak vyjádříme jako

$$\pi_{CSC}^S = \left( p_n^{opt} - \sum_{i=1}^n c_i \right) \cdot q(p_n^{opt}) = \frac{b}{4} \cdot \left( \frac{a}{b} - \sum_{i=1}^n c_i \right)^2. \quad \blacksquare$$

Ceny výstupu článku  $1, 2, \dots, n-1$  může volit centralizovaný řetězec různě, neboť při dané ceně koncové produkce nemají tyto na celkový zisk řetězce vliv, určují ovšem rozdělení tohoto zisku mezi jednotlivé články. Stanoví-li centralizovaný řetězec tyto ceny např. na úrovni součtu ceny vstupu a jednotkových nákladů daného článku, potom  $\pi_{CSC} = \pi_n$ . Rozhodne-li se centralizovaný řetězec zajistit rovnoměrné rozdělení zisku v celém řetězci, stačí dosáhnout stejné ziskové marže pro všechny články. Potom pro každou cenu  $p_i$  platí

$$p_i = \frac{i}{n} \cdot \left( p_n^{opt} - \sum_{j=1}^n c_j \right) + \sum_{j=1}^i c_j, \quad i = 1, 2, \dots, n.$$

Obecně každá posloupnost cen  $p_1, p_2, \dots, p_{n-1}$ , kde  $p_1 > c_1$  a  $p_i > p_{i-1} + c_i$  pro  $i = 2, 3, \dots, n$  zajistí (při kladném celkovém zisku) kladné zisky všem článkům.

Obrátme nyní pozornost k decentralizovanému řetězci. Zisk decentralizovaného řetězce, který odpovídá rovnovážným cenám, popisuje následující tvrzení:

**TVRZENÍ 2.2:** *Decentralizovaný sériový řetězec dosahuje v lineárním modelu zisku*

$$\pi_{DSC}^S = \frac{(2^n - 1) \cdot b}{2^{2n}} \cdot \left( \frac{a}{b} - \sum_{i=1}^n c_i \right)^2. \quad (2.4)$$

**DŮKAZ:** Odvodíme rovnovážné ceny postupem podobným jako v příkladu 1, který popisoval model dvojí marginalizace. Budeme vyjadřovat podmínky optimalizace zisku směrem od posledního ( $n$ -tého) článku, který je v kontaktu přímo se spotřebitelem. Zisk řetězce  $n$ -tého článku je součinem ziskové marže (tj. přebytku ceny výstupu nad součtem ceny vstupu a jednotkových nákladů) a poptávaného množství. Můžeme jej tedy vyjádřit následujícím způsobem:

$$\pi_n = (p_n - c_n - p_{n-1}) \cdot (a - b \cdot p_n).$$

Pro danou (tj. libovolnou konstantní) cenu vstupu a při kladných hodnotách parametrů  $a$ ,  $b$  a  $c_n$  je zisk  $n$ -tého článku ryze konkávní funkcí jím stanovené ceny koncové produkce. Z podmínky prvního řádu tudíž plyne, že chce-li článek  $n$  optimalizovat svůj zisk, bude pro libovolnou cenu vstupu  $p_{n-1}$  volit cenu výstupu

$$p_n^{opt} = \frac{1}{2} \cdot \left( \frac{a}{b} + c_n + p_{n-1} \right),$$

která povede k poptávanému množství

$$q(p_{n-1}) = a - b \cdot p_n^{opt} = \frac{b}{2} \cdot \left( \frac{a}{b} - c_n - p_{n-1} \right). \quad (2.5)$$

Vztah (2.5) představuje poptávané množství jakožto funkci ceny výstupu za podmínky, že článek  $n$  optimalizuje svůj individuální zisk. Položíme nyní  $p_0 = 0$  a dokážeme podobný vztah v obecném tvaru pro všechna  $i = 1, 2, \dots, n-1$ . Přesněji řečeno, ukážeme, že pokud navazující články optimalizují individuální zisk, lze vyjádřit poptávané množství jako funkci ceny výstupu  $i$ -tého článku ve tvaru

$$q(p_i) = \frac{b}{2^{n-i}} \cdot \left( \frac{a}{b} - \sum_{j=i+1}^n c_j - p_i \right), \quad i = 1, 2, \dots, n-1. \quad (2.6)$$

Důkaz (2.6) provedeme úplnou indukcí. Pro  $i = n-1$  jsme již tvrzení dokázali (odvozením vztahu (2.5)). Zbývá nám dokázat, že platí-li tvrzení pro článek  $i$  (kde  $i \in \{2, 3, \dots, n-1\}$ ),

potom platí také pro článek  $i-1$ . Platí-li tvrzení pro článek  $i$ , můžeme jeho ziskovou funkci napsat jako

$$\pi_i = \frac{b}{2^{n-i}} \cdot \left( \frac{a}{b} - \sum_{j=i+1}^n c_j - p_i \right) \cdot (p_i - c_i - p_{i-1}).$$

Tato zisková funkce je pro libovolnou cenu vstupu  $p_{i-1}$  ryze konkávní funkce proměnné  $p_i$ . Z podmínky prvního řádu se snadno přesvědčíme, že chce-li článek  $i$  optimalizovat svůj zisk, bude volit cenu  $p_i$  na úrovni

$$p_i^{opt} = \frac{1}{2} \cdot \left( \frac{a}{b} + c_i - \sum_{j=i+1}^n c_j + p_{i-1} \right), \quad (2.7)$$

Po dosazení této ceny do (2.6) získáme poptávané množství jakožto funkci proměnné  $p_{i-1}$ :

$$q(p_{i-1}) = \frac{b}{2^{n-i}} \cdot \left( \frac{a}{b} - \sum_{j=i+1}^n c_j - \frac{1}{2} \cdot \left( \frac{a}{b} + c_i - \sum_{j=i+1}^n c_j + p_{i-1} \right) \right) = \frac{b}{2^{n-i+1}} \cdot \left( \frac{a}{b} - \sum_{j=i}^n c_j - p_{i-1} \right),$$

čímž je dokázán indukční krok, a tedy i celý vztah (2.6). Díky platnosti (2.6) můžeme prohlásit pro všechna  $i = 1, 2, \dots, n-1$  za platné také tvrzení (2.7), ze kterého obdržíme pro  $i = 1$  a vzhledem k  $p_0 = 0$  vztah pro rovnovážnou cenu prvního článku

$$p_1^{eq} = \frac{1}{2} \cdot \left( \frac{a}{b} + c_1 - \sum_{j=2}^n c_j \right). \quad (2.8)$$

Dosazením (2.8) do (2.6) pro  $i = 1$  zjistíme, že poptávané množství v decentralizovaném řetězci bude

$$q = \frac{b}{2^n} \cdot \left( \frac{a}{b} - \sum_{i=1}^n c_i \right). \quad (2.9)$$

Ze vztahu (2.7) můžeme postupným dosazováním získat explicitní vzorce pro rovnovážné ceny výstupu každého článku. Dostaneme vztah

$$p_i^{eq} = \frac{2^i - 1}{2^i} \cdot \left( \frac{a}{b} - \sum_{j=i+1}^n c_j \right) + \frac{1}{2^i} \cdot \sum_{j=1}^i c_j, \quad i = 1, 2, \dots, n-1 \quad (2.10)$$

a pro poslední článek vyjádříme

$$p_n^{eq} = \frac{2^n - 1}{2^n} \cdot \frac{a}{b} + \frac{1}{2^n} \cdot \sum_{j=1}^n c_j.$$

Celkový zisk decentralizovaného řetězce potom můžeme vyjádřit jako

$$\pi_{DSC}^S = \left( p_n^{eq} - \sum_{i=1}^n c_i \right) \cdot (a - b \cdot p_n^{eq}) = \frac{(2^n - 1) \cdot b}{2^{2n}} \cdot \left( \frac{a}{b} - \sum_{i=1}^n c_i \right)^2$$

a důkaz je hotov. ■

Porovnáním zisku centralizovaného řetězce (2.3) a zisku jeho decentralizovaného protějšku (2.4) vidíme, že zisk decentralizovaného řetězce je pro  $n \geq 2$  menší, než zisk řetězce centralizovaného. Efektivnost decentralizovaného řetězce je dána poměrem těchto zisků, tedy

$$\varepsilon^S = \frac{\pi_{DSC}^S}{\pi_{CSC}^S} = \frac{2^n - 1}{2^{2n-2}}. \quad (2.11)$$

Pro  $n = 2$  dostaneme samozřejmě efektivnost v lineárním modelu dvojí marginalizace, která činí 0,75. Můžeme ověřit, že tato hodnota opravdu odpovídá poměru zisků, které jsme spočetli v příkladu 1. Z (2.11) je dále patrné, že s rostoucím počtem článků dodavatelského řetězce efektivnost rapidně klesá, což není nikterak neočekávaný závěr – s rostoucím počtem článků dochází ke skládání nepříznivého monopolního efektu. Co je možná zajímavější, je skutečnost, že efektivnost řetězce v lineárním modelu vůbec nezávisí na konkrétním tvaru poptávkové funkce nebo jednotkových nákladů, nýbrž pouze na délce řetězce. Jedná se tedy o vlastnost, která je vlastností struktury dodavatelského řetězce, nikoli konkrétních použitých technologií a podobně.

Mohlo by nás dále zajímat, jak je celkový zisk decentralizovaného řetězce rozdělen mezi jednotlivé články. Odpověď na tuto otázku dává tvrzení 2.3:

**TVRZENÍ 2.3:** Zisk  $i$ -tého článku v decentralizovaném sériovém řetězci v lineárním modelu je roven

$$\pi_i = \frac{b}{2^{n+i}} \cdot \left( \frac{a}{b} - \sum_{i=1}^n c_i \right)^2, \quad i = 1, 2, \dots, n. \quad (2.12)$$

**DŮKAZ:** Zisk článků je součinem poptávaného množství a ziskové marže  $\mu$  jednotlivých článků. Ziskovou marži prvního článku získáme pomocí rovnovážné ceny (2.8) jako

$$\mu_1 = p_1 - c_1 = \frac{1}{2} \cdot \left( \frac{a}{b} + c_1 - \sum_{j=2}^n c_j \right) - c_1 = \frac{1}{2} \cdot \left( \frac{a}{b} - \sum_{j=1}^n c_j \right).$$

Pro ostatní články použijeme (2.10) a získáme

$$\mu_i = p_i - c_i - p_{i-1} = \frac{2^i - 1}{2^i} \cdot \left( \frac{a}{b} - \sum_{j=i+1}^n c_j \right) + \frac{1}{2^i} \sum_{j=1}^i c_j - c_i - \frac{2^{i-1} - 1}{2^{i-1}} \cdot \left( \frac{a}{b} - \sum_{j=i}^n c_j \right) + \frac{1}{2^{i-1}} \sum_{j=1}^{i-1} c_j,$$

odkud po úpravě dostaneme

$$\mu_i = \frac{1}{2^i} \cdot \left( \frac{a}{b} - \sum_{j=1}^n c_j \right), i = 1, 2, \dots, n.$$

Nyní již snadno získáme hodnotu individuálního zisku  $i$ -tého článku jako součin ziskové marže podle poslední rovnosti a poptávaného množství podle (2.9):

$$\pi_i = q \cdot \mu_i = \frac{b}{2^n} \cdot \left( \frac{a}{b} - \sum_{i=1}^n c_i \right) \cdot \frac{1}{2^i} \cdot \left( \frac{a}{b} - \sum_{j=1}^n c_j \right) = \frac{b}{2^{n+i}} \cdot \left( \frac{a}{b} - \sum_{i=1}^n c_i \right)^2. \quad \blacksquare$$

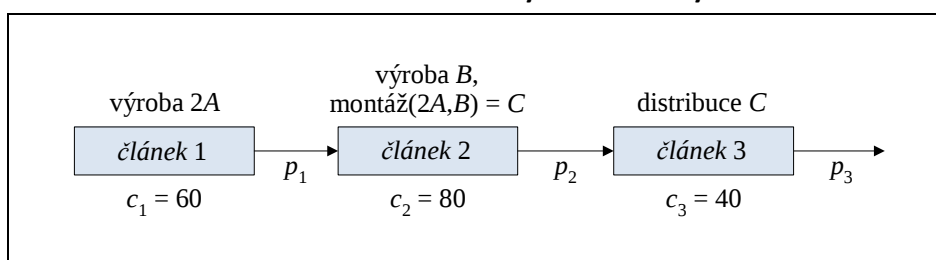
Z tvrzení 2.3 plyne důležitá vlastnost, že zisk každého článku decentralizovaného řetězce v lineárním modelu závisí při dané poptávkové funkci *pouze* na celkových nákladech celého řetězce ( $\sum c_i$ ) a na pořadí daného článku v řetězci. Jinými slovy, rozložení zisku vůbec nezávisí na rozložení nákladů v řetězci. Největší zisk dosahuje článek 1 a pro zisky navazujících článků platí, že  $\pi_{i+1} = \frac{1}{2} \cdot \pi_i$ . Taková situace zároveň svědčí o silných pozitivních externalitách nákladových úspor v rámci řetězce: pokud se libovolnému článku řetězce podaří snížit své jednotkové náklady (např. zlepšením výrobních technologií), zvýší se nejen zisk tohoto článku, ale i zisk všech ostatních článků, a to dokonce ve stejné proporci. Podobně, pokud bychom se z nějakého důvodu rozhodli přesunout některé procesy z nižších článků řetězu na vyšší nebo naopak, zisky jednotlivých článků ani efektivnost celého řetězce se nezmění (pokud budou články vykonávat přesouvané procesy se stejnými jednotkovými náklady). Zmíněné změny budeme ilustrovat na následujících jednoduchých příkladech.

**PŘÍKLAD 2:** Uvažujme sériový dodavatelský řetězec se třemi články. Koncovým produktem řetězce je výrobek  $C$ , který je montován ze dvou druhů polotovarů –  $A$  a  $B$ . Na výrobu jednoho výrobku  $C$  použijeme dva kusy polotovaru  $A$  a jeden polotovar  $B$ . Poptávka po výrobku  $C$  je dána předpisem

$$q = 300 - p_3.$$

Jednotkové náklady na výrobu  $A$  jsou 30 Kč, na výrobu  $B$  60 Kč, montáž dvou kusů  $A$  a jednoho  $B$  do výsledného  $C$  stojí 20 Kč, distribuce  $C$  koncovým spotřebitelům je vyčíslena na 40 Kč za přepravenou jednotku. Obrázek 2-2 popisuje, jak jsou procesy rozděleny mezi jednotlivé články a jaké jsou náklady každého článku na jednu jednotku koncové produkce.

**Obrázek 2-2: Příklad 2 – sériový dodavatelský řetězec**



Podle (2.3) snadno vyčíslíme zisk centralizovaného řetězce jako

$$\pi_{CSC} = \frac{1}{4} \cdot \left( \frac{300}{1} - (60 + 80 + 40) \right)^2 = 3600,$$

podobně zisk decentralizovaného řetězce je podle (2.4) roven 1575 Kč a efektivnost decentralizovaného řetězce je dána jejich poměrem, tedy

$$\varepsilon = \frac{\pi_{DSC}}{\pi_{CSC}} = \frac{1575}{3600} = 0,4375.$$

Efektivnost bychom samozřejmě mohli vyjádřit rovnou užitím (2.11). Podle vztahu (2.12) můžeme dále vypočítat zisky jednotlivých článků v decentralizovaném řetězci, výsledky včetně rovnovážných cen ukazuje následující tabulka.

**Tabulka 2-3: Příklad 2 – rovnovážné hodnoty I**

| $i$     | 1   | 2   | 3   |
|---------|-----|-----|-----|
| $p_i$   | 120 | 230 | 285 |
| $\pi_i$ | 900 | 450 | 225 |

Z tabulky 2-3 je patrné, že individuální zisky jsou v poměru 4 : 2 : 1. Podívejme se nyní, jak se rovnovážné hodnoty změní v případě, že by výrobní náklady na 2 výrobky A (tedy jednotkové náklady článku 1) klesly z 60 Kč na 20 Kč. Výsledky zachycuje tabulka 2-4:

**Tabulka 2-4: Příklad 2 – rovnovážné hodnoty II**

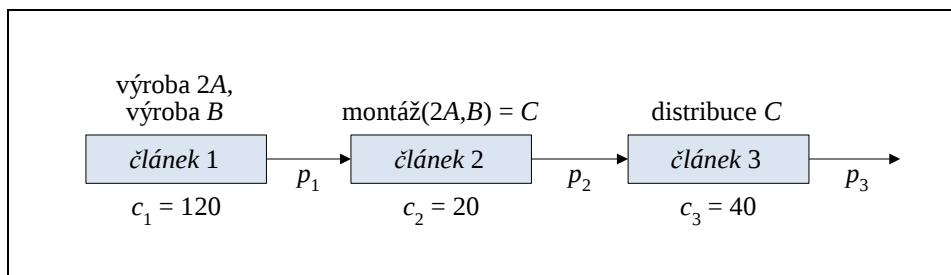
| $i$     | 1    | 2   | 3   |
|---------|------|-----|-----|
| $p_i$   | 100  | 220 | 280 |
| $\pi_i$ | 1600 | 800 | 400 |

Díky snížení jednotkových nákladů došlo v decentralizovaném řetězci k proporcionálnímu zvýšení všech individuálních zisků (přesně o 7/9 původní hodnoty), ze snížených nákladů tedy díky zmíněným externalitám těžily všechny články řetězce. Efektivnost řetězce ovšem zůstala stejná, protože stejnou proporcí, jakou se zvedl zisk všech decentralizovaného řetězce, se zvedl i zisk jeho centralizovaného protějšku (příslušné hodnoty jsou  $\pi_{DSC} = 2800$ ,  $\pi_{CSC} = 6400$ ).

**PŘÍKLAD 3:** Vyjdeme ze stejného počátečního zadání jako v příkladu 2, tj. z řetězce popisovaného obrázkem 2-2 s rovnovážnými hodnotami v tabulce 2-3, a podíváme se, jak by se tyto hodnoty změnily, pokud bychom výrobu polotovaru B přesunuli z článku 2 do článku 1.

Předpokládejme přitom, že oba tyto články vyrábějí polotovar  $B$  se shodnými jednotkovými náklady. Nové rozložení procesů a nákladů v jednotlivých člancích ukazuje obrázek 2-5.

**Obrázek 2-5: Příklad 3 – sériový dodavatelský řetězec**



Rovnovážné hodnoty nového rozložení jsou zachyceny v tabulce 2-6. Porovnáním s původními hodnotami z tabulky 2-3 vidíme, že jediné, co se na výsledcích změnilo, je cena, za kterou prodává svůj výstup článek 1. Touto změnou musí být kompenzován přesun nákladů v rámci řetězce. Individuální zisky a efektivnost řetězce zůstanou však touto změnou nedotčeny.

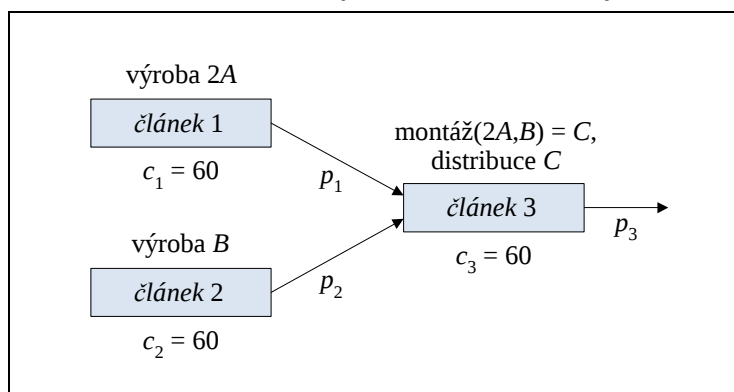
**Tabulka 2-6: Příklad 3 – rovnovážné hodnoty**

| $i$     | 1   | 2   | 3   |
|---------|-----|-----|-----|
| $p_i$   | 180 | 230 | 285 |
| $\pi_i$ | 900 | 450 | 225 |

## 2.2 Paralelní dodavatelské řetězce

Jak jsme viděli na předchozím příkladě, přesunutí výrobních procesů mezi dvěma články v sériovém decentralizovaném řetězci v lineárním modelu nemá na efektivnost vliv. Jiný druh změny organizace výroby v dodavatelském řetězci by však změnu přinesl, jak ukáže následující příklad.

**Obrázek 2-7: Příklad 4 – paralelní dodavatelský řetězec**



**PŘÍKLAD 4:** Změníme nyní původní řetězec z příkladu 2 následujícím způsobem: přesuneme proces montáže z článku 2 do článku 3, a strukturu řetězce změníme tak, jak ji popisuje obrázek 2-7. Taková struktura již představuje řetězec paralelního typu. Vzhledem k tomu, že se celkové jednotkové náklady řetězce nezměnily, je zisk centralizovaného řetězce stejný, jako v případě řetězce sériového. Pokusme se nyní odvodit ceny v decentralizovaném řetězci. Aby článek 3 mohl vyrobit jednotku koncové produkce, musí nakoupit po jedné jednotce od obou svých předchůdců. Zisk třetího článku proto vyjádříme jako

$$\pi_3 = (p_3 - 60 - p_1 - p_2) \cdot (300 - p_3),$$

kde první výraz představuje ziskovou marži a druhý výraz poptávané množství. Při libovolných daných cenách  $p_1$  a  $p_2$  se jedná opět o konkávní funkci jedné proměnné, tedy chce-li třetí článek optimalizovat svůj zisk, bude volit cenu

$$p_3^{opt} = \frac{1}{2} \cdot (360 + p_1 + p_2),$$

kterou získáme z podmínky prvního řádu. Poptávané množství odpovídající této ceně činí

$$q = \frac{1}{2} \cdot (240 - p_1 - p_2),$$

zisk článků 1 a 2 lze tudíž vyjádřit jako

$$\begin{aligned}\pi_1 &= \frac{1}{2} \cdot (p_1 - 60) \cdot (240 - p_1 - p_2), \\ \pi_2 &= \frac{1}{2} \cdot (p_2 - 60) \cdot (240 - p_1 - p_2).\end{aligned}$$

Jaké ceny však budou volit články 1 a 2? Z předchozího vztahu je zřejmé, že pokud jeden z článků zvýší cenu svého výstupu (za účelem zvýšení zisku), sníží tím poptávané množství, a tedy i zisk druhého z článků. Je tomu tak proto, že produkce obou článků je v silně komplementárním vztahu. Stanovení cen v první úrovni decentralizovaného řetězce je potom nekooperativním konfliktem, čili ekonomickou hrou, jejímž racionálním řešením je kombinace cen, která odpovídá Nashově rovnováze. Z definice Nashovy rovnováhy víme, že výsledné ceny musí splňovat vlastnost, že při dané ceně výstupu článku 1 volí článek 2 cenu výstupu maximalizující jeho zisk a naopak. Z podmínky prvního řádu dostaneme ceny maximalizující zisk obou článků při dané ceně článku druhého jako

$$\begin{aligned}p_1^{opt} &= \frac{1}{2} \cdot (300 - p_2), \\ p_2^{opt} &= \frac{1}{2} \cdot (300 - p_1).\end{aligned}$$

Nashova rovnováha představuje takovou kombinaci cen  $p_1$  a  $p_2$ , která splňuje obě podmínky současně. Pro nalezení rovnovážných cen článků 1 a 2 tedy stačí vyřešit uvedenou soustavu dvou lineárních rovnic o dvou proměnných. Poté již snadno dopočítáme rovnovážnou cenu koncové produkce i individuální zisky článků. Výsledky jsou uvedeny v tabulce 2-8. Zisk

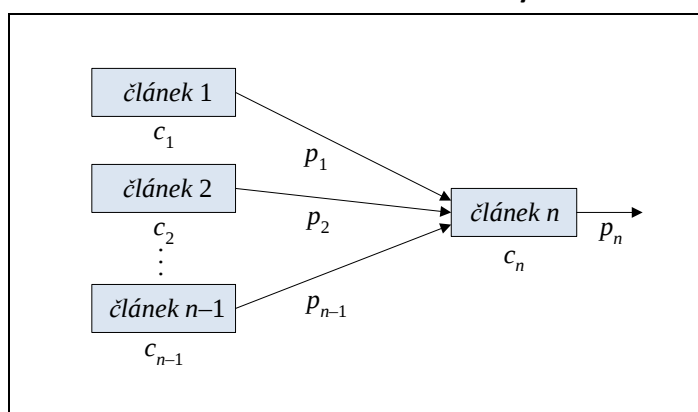
decentralizovaného řetězce se z původních 1575 Kč zvýšil na 2000 Kč, efektivnost řetězce se tedy rovněž zvedla, a to na  $2000 / 3600 = 0,556$ .

**Tabulka 2-8: Příklad 4 – rovnovážné hodnoty**

| $i$     | 1   | 2   | 3   |
|---------|-----|-----|-----|
| $p_i$   | 100 | 100 | 280 |
| $\pi_i$ | 800 | 800 | 400 |

Výsledky příkladu 4 naznačují, že decentralizované řetězce s paralelními články jsou efektivnější, nežli řetězce s řazením sériovým (se stejným počtem článků). Abychom tuto domněnku dokázali, je třeba odvodit vztahy demonstrováné na předchozím příkladu obecně. Uvažujme dodavatelský řetězec s paralelní strukturou, jak ji popisuje obrázek 2-9. Aby mohl článek  $n$  zhotovit jednotku koncové produkce, musí nejprve nakoupit po jednotce výstupu od každého článku  $1, 2, \dots, n-1$ . Při modelování poptávky samozřejmě stále pracujeme s lineární funkcí (2.1).

**Obrázek 2-9: Paralelní dodavatelský řetězec**



Zisk centralizovaného řetězce můžeme vyjádřit zcela stejně jako v modelu řetězce sériového. Chceme-li totiž optimalizovat celkový zisk řetězce, poměřujeme pouze celkové tržby, které jsou dané cenou posledního článku, a celkové náklady, které jsou dány poptávaným množstvím a součtem jednotkových nákladů v celém řetězci; na pořadí článků, resp. struktuře řetězce tedy vůbec nezáleží. Můžeme proto pro zisk centralizovaného řetězce opět využít vztah (2.3), neboli  $\pi_{CSC}^P = \pi_{CSC}^S$ . Situace v decentralizovaném řetězci se ovšem v různých strukturách podstatně liší. Zisk decentralizovaného paralelního řetězce odpovídající rovnovážným cenám popisuje následující tvrzení:

**TVRZENÍ 2.4:** *Decentralizovaný paralelní řetězec dosahuje v lineárním modelu zisku*

$$\pi_{DSC}^P = \frac{2n-1}{n^2} \cdot \frac{b}{4} \cdot \left( \frac{a}{b} - \sum_{i=1}^n c_i \right)^2. \quad (2.13)$$

**DŮKAZ:** Při určení rovnovážných cen decentralizovaného řetězce budeme opět postupovat od konce řetězce, tedy od článku  $n$ . Zisk článku  $n$  vyjádříme jako

$$\pi_n = \left( p_n - c_n - \sum_{i=1}^{n-1} p_i \right) \cdot (a - b \cdot p_n) = -b \cdot p_n^2 + \left( a + b \cdot c_n + b \cdot \sum_{i=1}^{n-1} p_i \right) \cdot p_n - a \cdot \sum_{i=1}^{n-1} p_i.$$

Při daných cenách článků  $1, 2, \dots, n-1$  se jedná o ryze konkávní funkci proměnné  $p_n$  a z podmínky prvního řádu získáme vztah pro optimální cenu výstupu článku  $n$  v závislosti na cenách všech vstupů:

$$p_n^{opt} = \frac{1}{2} \cdot \left( \frac{a}{b} + c_n + \sum_{i=1}^{n-1} p_i \right).$$

Této ceně po dosazení do poptávkové funkce odpovídá poptávané množství

$$q = \frac{b}{2} \cdot \left( \frac{a}{b} - c_n - \sum_{i=1}^{n-1} p_i \right). \quad (2.14)$$

Odtud můžeme pro články z první úrovně řetězce vyjádřit individuální zisk jako

$$\pi_i = (p_i - c_i) \cdot \frac{b}{2} \cdot \left( \frac{a}{b} - c_n - \sum_{j=1}^{n-1} p_j \right), \quad i = 1, 2, \dots, n-1.$$

Při určení rovnovážných cen narazíme i v tomto obecném případě na principiálně stejnou situaci jako v příkladu 4. Články v první úrovni řetězce, tj. články  $1, 2, \dots, n-1$ , svádí prostřednictvím cen svého výstupu souboj o to, kdo si odnese největší část získaných prostředků, přičemž se zde jedná o nekooperativní konflikt (jako v příkladu 4). Jak napovídá teorie her, racionálně uvažující články budou volit ceny na úrovni Nashovy rovnováhy. Kombinace cen, která odpovídá definici Nashovy rovnováhy, musí současně splňovat podmínky optimalizace pro všechny články z první úrovně. Tyto podmínky získáme vynulováním derivace ziskové funkce jednotlivých článků, čímž dostaneme vztahy

$$p_i^{eq} = \frac{1}{2} \cdot \left( \frac{a}{b} + c_i - c_n - \sum_{\substack{j=1, \\ j \neq i}}^{n-1} p_j^{eq} \right), \quad i = 1, 2, \dots, n-1.$$

Pro Nashovy rovnovážné ceny jsme tak obdrželi soustavu o  $n-1$  rovnicích s  $n-1$  neznámými ( $a$ ,  $b$  a jednotkové náklady  $c_i$  jsou parametry soustavy rovnic, neznámé tvoří  $p_i^{eq}$ ), kterou můžeme přepsat jako:

$$\mathbf{A} \cdot \mathbf{p}^{eq} = \mathbf{d}, \quad (2.15)$$

kde

$\mathbf{p}^{eq} = (p_1^{eq}, p_2^{eq}, \dots, p_{n-1}^{eq})^T$  je sloupcový vektor rovnovážných výstupních cen,

$\mathbf{d} = (d_1, d_2, \dots, d_{n-1})^T$  je sloupcový vektor pravých stran,  $d_i = \frac{a}{b} - c_n + c_i$ ,

$\mathbf{A}$  je matice typu  $(n-1) \times (n-1)$ , která obsahuje následující koeficienty:

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 1 & \dots & 1 \\ 1 & 2 & 1 & \dots & 1 \\ 1 & 1 & 2 & \dots & 1 \\ \vdots & & & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 2 \end{pmatrix}. \quad (2.16)$$

Matice  $\mathbf{A}$  je regulární a její inverze  $\mathbf{A}^{-1}$  vypadá následovně:

$$\mathbf{A}^{-1} = \begin{pmatrix} (n-1)/n & -1/n & -1/n & \dots & -1/n \\ -1/n & (n-1)/n & -1/n & \dots & -1/n \\ -1/n & -1/n & (n-1)/n & \dots & -1/n \\ \vdots & & & \ddots & \vdots \\ -1/n & -1/n & -1/n & \dots & (n-1)/n \end{pmatrix}. \quad (2.17)$$

Vynásobením soustavy (2.15) maticí  $\mathbf{A}^{-1}$  zleva získáme explicitní vyjádření ceny výstupu všech článků z druhé úrovně řetězce. Dostaneme postupně:

$$\mathbf{p}^{eq} = \mathbf{A}^{-1} \mathbf{d}, \quad (2.18)$$

abychom mohli přehledněji vyjádřit součin na pravé straně, přepíšeme vektor  $\mathbf{d}$  ve tvaru součinu

$$\begin{pmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{pmatrix} \cdot \left( \frac{a}{b} - c_n, c_1, c_2, \dots, c_{n-1} \right)^T,$$

následně upravíme pravou stranu (2.18) vynásobením matic na

$$\begin{pmatrix} 1/n & (n-1)/n & -1/n & \dots & -1/n \\ 1/n & -1/n & (n-1)/n & \dots & -1/n \\ 1/n & -1/n & -1/n & \dots & -1/n \\ \vdots & & & \ddots & \vdots \\ 1/n & -1/n & -1/n & \dots & (n-1)/n \end{pmatrix} \cdot \left( \frac{a}{b} - c_n, c_1, c_2, \dots, c_{n-1} \right)^T,$$

odkud již snadno vyjádříme

$$p_i^{eq} = \frac{1}{n} \cdot \left( \frac{a}{b} + n \cdot c_i - \sum_{j=1}^n c_j \right), \quad i = 1, 2, \dots, n-1. \quad (2.19)$$

dosadíme-li tyto optimální ceny do (2.14), získáme úroveň poptávaného množství

$$q = \frac{b}{2} \cdot \left( \frac{a}{b} - c_n - \frac{1}{n} \cdot \sum_{i=1}^{n-1} \left( \frac{a}{b} + n \cdot c_i - \sum_{j=1}^n c_j \right) \right),$$

z čehož po úpravě získáme

$$q = \frac{b}{2n} \cdot \left( \frac{a}{b} - \sum_{i=1}^n c_i \right). \quad (2.20)$$

Explicitní vyjádření optimální ceny výstupu posledního článku dostaneme snadno např. dosazením získaného vztahu pro poptávané množství do poptávkové funkce:

$$p_n^{eq} = \frac{1}{2n} \left( (2n-1) \cdot \frac{a}{b} + \sum_{i=1}^n c_i \right). \quad (2.21)$$

Celkový zisk decentralizovaného řetězce získáme z předchozích dvou výsledků jako

$$\pi_{DSC}^P = \left( p_n^{opt} - \sum_{j=1}^n c_j \right) \cdot q = \left( \frac{1}{2n} \left( (2n-1) \cdot \frac{a}{b} + \sum_{i=1}^n c_i \right) - \sum_{j=1}^n c_j \right) \cdot \frac{b}{2n} \cdot \left( \frac{a}{b} - \sum_{i=1}^n c_i \right),$$

odkud již po úpravě vyjde tvrzení. ■

Porovnáním předchozího tvrzení a zisku centralizovaného řetězce (2.3) získáme obecné vyjádření efektivnosti paralelního řetězce v lineárním modelu

$$\varepsilon^P = \frac{\pi_{DSC}^P}{\pi_{CSC}^P} = \frac{2n-1}{n^2}. \quad (2.22)$$

Podobně jako v sériovém řetězci, efektivnost paralelního řetězce v lineárním modelu nezávisí na konkrétním tvaru poptávkové funkce nebo jednotkových nákladů, nýbrž je dána pouze počtem článků řetězce, a s rozsahem řetězce prudce klesá. Porovnáním s efektivností sériového řetězce (2.11) se můžeme přesvědčit, že pro  $n \geq 3$  je efektivnost paralelního řetězce vyšší (pro  $n = 2$  jsou řetězce, a tedy i efektivnost, totožné). Závěr, který jsme uvedli u příkladu 4, má proto opravdu obecnou platnost.

Podívejme se ještě, jak vypadá rozložení zisku decentralizovaného řetězce mezi jednotlivé články. Výsledek shrnuje tvrzení 2.5:

**TVRZENÍ 2.5:** *Zisk posledního článku v decentralizovaném sériovém řetězci v lineárním modelu je roven*

$$\pi_n = \frac{b}{4n^2} \cdot \left( \frac{a}{b} - \sum_{j=1}^n c_j \right)^2, \quad (2.23)$$

*zisk ostatních článků lze vyjádřit jako*

$$\pi_i = \frac{b}{2n^2} \cdot \left( \frac{a}{b} - \sum_{j=1}^n c_j \right)^2, \quad i = 1, 2, \dots, n-1. \quad (2.24)$$

**DŮKAZ:** Pro výpočet zisku jednotlivých článků opět použijeme ziskové marže článků  $\mu$ . Ziskovou marži  $i$ -tého článku pro  $i = 1, 2, \dots, n-1$  zjistíme odečtením nákladů článku od jeho rovnovážné ceny (2.19):

$$\mu_i = (p_i - c_i) = \frac{1}{n} \cdot \left( \frac{a}{b} + n \cdot c_i - \sum_{j=1}^n c_j \right) - c_i = \frac{1}{n} \cdot \left( \frac{a}{b} - \sum_{j=1}^n c_j \right), \quad i = 1, 2, \dots, n-1.$$

Ziskovou marži článku  $n$  můžeme za použití (2.19) a (2.20) vyjádřit jako

$$\mu_n = p_n - c_n - \sum_{i=1}^{n-1} p_i = \frac{1}{2n} \cdot \left( \frac{a}{b} - \sum_{j=1}^n c_j \right).$$

Individuální zisk všech článků již nyní snadno dostaneme jako součin ziskové marže  $\mu_i$  a rovnovážného poptávaného množství  $q$  ze vztahu (2.22), po úpravě vyjde tvrzení. ■

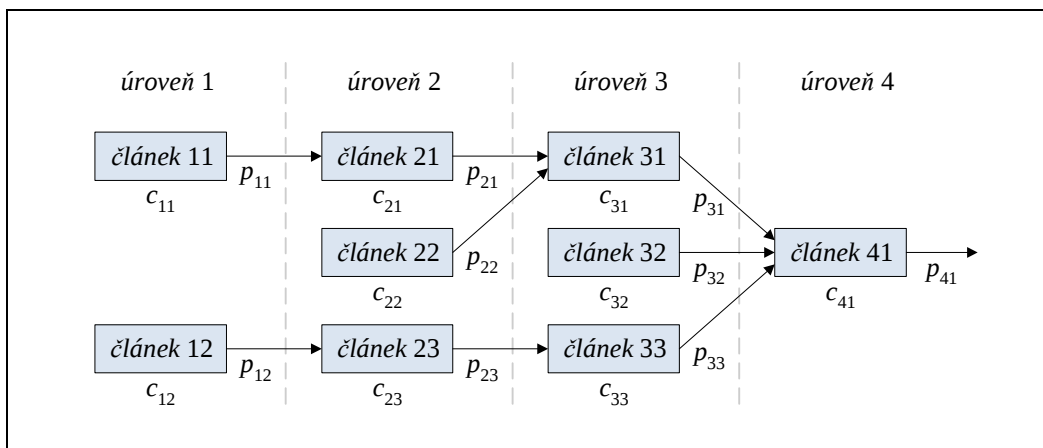
Z tvrzení 2.5 plyne, že zisk jednotlivých článků je, stejně jako v sériovém řetězci, závislý pouze na součtu nákladů v řetězci a na pozici článku v řetězci. Můžeme tedy učinit totožné závěry ohledně nákladových externalit a přeskupení procesů v rámci řetězce jako v sériovém modelu. Celkový zisk decentralizovaného řetězce bude vždy rozdělen tak, že články v první úrovni řetězce (tj. články  $1, 2, \dots, n-1$ ) budou dosahovat stejného zisku a článek koncový bude mít zisk poloviční.

## 2.3 Kombinované dodavatelské řetězce

Obdobným způsobem, jakým jsme odvodili vztahy pro sériové a paralelní řetězce, můžeme zkoumat i řetězce, které obě předchozí struktury kombinují. Při popisu paralelního řetězce byl použit pojem úrovně řetězce, pro popis kombinovaných řetězců potřebujeme tento pojem definovat přesněji: **úrovní řetězce** budeme rozumět množinu všech článků, jejichž výstup musí projít přes stejný počet dalších článků, než se dostane ke koncovému spotřebiteli. Články budeme nadále rozlišovat pomocí dvou indexů, prvním bude číslo úrovně v řetězci a druhým pořadové číslo článku v dané úrovni. Označme celkový počet úrovní řetězce jako  $m$ , počet článků v jednotlivých úrovních pak vyjádříme  $m$ -složkovým vektorem  $\mathbf{n}$ , jehož  $i$ -tá složka  $n_i$  bude představovat počet článků v  $i$ -té úrovni (přičemž koncový článek je ve své

úrovni vždy sám, tedy  $n_m = 1$ ). Obrázek 2-10 znázorňuje příklad řetězce pro  $m = 4$  a  $\mathbf{n} = (2,3,3,1)$ .

**Obrázek 2-10: Příklad kombinovaného dodavatelského řetězce**



Z obrázku je patrné, že samotný vektor  $\mathbf{n}$  neurčuje jednoznačně strukturu řetězce, tj. podobu vazeb mezi články. Jak ukážeme, stačí však i takto hrubý popis struktury řetězce pro určení všech charakteristik decentralizovaného řetězce. Než tak však učiníme, podívejme se na zisk řetězce centralizovaného. Jak bylo uvedeno v diskuzi o řetězcích paralelního typu, výpočet zisku centralizovaného řetězce nezávisí na jeho struktuře. Vztah (2.3), který popisoval zisk sériového (a vlastně i paralelního) centralizovaného řetězce, je ovšem potřeba přeformulovat tak, aby odpovídal nově zavedenému značení kombinovaného řetězce:

$$\pi_{CSC}^C = \frac{b}{4} \cdot \left( \frac{a}{b} - \sum_{i=1}^m \sum_{j=1}^{n_i} c_{ij} \right)^2. \quad (2.25)$$

Zisk decentralizovaného řetězce popisuje následující tvrzení:

**TVRZENÍ 2.6:** *Decentralizovaný kombinovaný řetězec dosahuje v lineárním modelu zisku*

$$\pi_{DSC}^C = \frac{2 \cdot \prod_{i=1}^m (n_i + 1) - 1}{\prod_{i=1}^{m-1} (n_i + 1)^2} \cdot \frac{b}{4} \cdot \left( \frac{a}{b} - \sum_{i=1}^m \sum_{j=1}^{n_i} c_{ij} \right)^2. \quad (2.26)$$

**DŮKAZ:** Nežli začneme s vlastním důkazem, můžeme se přesvědčit, že vztah (2.26) funguje i pro paralelní řetězce. Tomu by tak pochopitelně mělo být, neboť řetězce s paralelní strukturou i řetězce se strukturou sériovou jsou jen speciálními případy řetězců kombinovaných, přičemž paralelní řetězec má počet úrovní  $m = 2$ , všechny články kromě jednoho se nachází v úrovni první, tedy  $\mathbf{n} = (n-1, 1)$ . Dosazením těchto hodnot získáme přesně vztah (2.13) popisující zisk paralelního řetězce (obdobně bychom samozřejmě mohli ověřit platnost dokazovaného tvrzení pro řetězce sériové). Můžeme se proto v následujících úvahách

omezit na řetězce s více než dvěma úrovněmi. Jako už obou předchozích typů řetězců budeme postupovat od konce řetězce. Při odvozování budeme kombinovat přístupy použité v důkazech tvrzení 2.2 a 2.4. Podobně jako v paralelním řetězci, ovšem s novým značením, můžeme vyjádřit cenu, kterou stanoví koncový článek maximalizující zisk, jako

$$p_{m1}^{opt} = \frac{1}{2} \cdot \left( \frac{a}{b} + c_{m1} + \sum_{j=1}^{n_{m-1}} p_{m-1,j} \right),$$

a tomu odpovídající poptávané množství

$$q = \frac{b}{2} \cdot \left( \frac{a}{b} - c_{m1} - \sum_{j=1}^{n_{m-1}} p_{m-1,j} \right). \quad (2.27)$$

Dokážeme nyní platnost pomocného vzorce (2.28), který je analogií k (2.6) z důkazu tvrzení 2.2. Vzorec (2.28) popisuje pro  $i = 1, 2, \dots, m-1$  vztah mezi poptávaným množstvím a cenami článků z  $i$ -té úrovně za podmínky, že všechny navazující články optimalizují zisk:

$$q = \frac{b}{\prod_{j=i+1}^m (n_j + 1)} \cdot \left( \frac{a}{b} - \sum_{j=i+1}^m \sum_{k=1}^{n_j} c_{jk} - \sum_{j=1}^{n_i} p_{ij} \right). \quad (2.28)$$

K důkazu (2.28) použijeme úplnou indukci. Pro  $i = m-1$  nám dává platnost (2.27). Zbývá ověřit indukční krok, což se ukáže jako poměrně zdlouhavé. Cílem následujících dvou stran tedy bude potvrdit, že platí-li (2.28) pro úroveň  $i$ , platí rovněž pro úroveň  $i-1$ . Položíme  $n_0 = 0$  a zavedeme pro  $j = 1, 2, \dots, n_i$  a  $k = 1, 2, \dots, n_{i-1}$  konstanty  $\delta_{jk}$  následujícím způsobem

$$\begin{aligned} \delta_{jk} &= 1, \text{ pakliže v řetězci existuje vazba mezi článkem } ij \text{ a článkem } i-1,k, \\ \delta_{jk} &= 0 \text{ jinak.} \end{aligned}$$

Potom platí, že každý článek z úrovně  $i-1$  má právě jednoho následníka v úrovni  $i$ , čili

$$\sum_{j=1}^{n_i} \delta_{jk} = 1, \quad k = 1, 2, \dots, n_{i-1}, \quad (2.29)$$

$$\sum_{k=1}^{n_{i-1}} \delta_{jk} = 1, \quad j = 1, 2, \dots, n_i. \quad (2.30)$$

Zisk článků z úrovně  $i$  můžeme za platnosti (2.28) pro úroveň  $i$  vyjádřit jako

$$\pi_{ij} = \left( p_{ij} - c_{ij} - \sum_{k=1}^{n_{i-1}} \delta_{jk} \cdot p_{i-1,k} \right) \cdot \frac{b}{\prod_{j=i+1}^m (n_j + 1)} \cdot \left( \frac{a}{b} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} - \sum_{k=1}^{n_i} p_{ik} \right), \quad j = 1, 2, \dots, n_i.$$

Pro články v  $i$ -té úrovni řetězce a tvorbu jejich cen platí v takovém případě totéž, co pro články první úrovně v řetězci paralelním – tvorba cen je nekooperativním konfliktem.

Budeme proto opět hledat Nashovy rovnovážné strategie, které budou řešením soustavy rovnic představujících podmínku maximalizace zisku jednotlivých článků úrovně  $i$  při daných cenách výstupu článků v úrovni  $i-1$ . Rovnice získáme jako vždy vynulováním derivace individuálních ziskových funkcí. Soustava rovnic má následující tvar:

$$p_{ij}^{eq} = \frac{1}{2} \cdot \left( \frac{a}{b} + c_{ij} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} + \sum_{k=1}^{n_{i-1}} \delta_{jk} \cdot p_{i-1,k} - \sum_{\substack{k=1 \\ k \neq j}}^{n_i} p_{ik}^{eq} \right), \quad j = 1, 2, \dots, n_i.$$

Soustavu rovnic přepíšeme do tvaru

$$\mathbf{A} \cdot \mathbf{p}_i^{eq} = \mathbf{d},$$

kde

$\mathbf{p}_i^{eq} = (p_{i,1}^{eq}, p_{i,2}^{eq}, \dots, p_{i,n_i}^{eq})^T$  je sloupcový vektor rovnovážných výstupních cen,

$\mathbf{d} = (d_1, d_2, \dots, d_{n_i})^T$  je sloupcový vektor,  $d_j = \frac{a}{b} + c_{ij} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} + \sum_{k=1}^{n_{i-1}} \delta_{jk} \cdot p_{i-1,k}$ ,

$\mathbf{A}$  je matice typu  $n_i \times n_i$ , jejíž koeficienty mají stejnou strukturu jako v (2.16).

Inverzní matice k matici  $\mathbf{A}$  má tvar

$$\mathbf{A}^{-1} = \frac{1}{n_i + 1} \cdot \begin{pmatrix} n_i & 1 & \dots & 1 \\ 1 & n_i & & 1 \\ \vdots & & \ddots & \\ 1 & 1 & 1 & n_i \end{pmatrix}.$$

Situace je velmi podobná jako v soustavě (2.15) v paralelním řetězci s tím, že na pravé straně jsou navíc ceny článků z nižší úrovně. Za použití podobného postupu jako při řešení soustavy (2.15) a s přihlédnutím k (2.30) lze postupně vyjádřit následující řešení:

$$p_{ij}^{eq} = \frac{1}{n_i + 1} \cdot \left( \frac{a}{b} - \sum_{k=i}^m \sum_{l=1}^{n_k} c_{kl} + (n_i + 1) \cdot c_{ij} + \sum_{k=1}^{n_{i-1}} ((n_i + 1) \cdot \delta_{jk} - 1) \cdot p_{i-1,k} \right). \quad (2.31)$$

Připravíme si ze získaného vztahu (2.31) postupnou úpravou rovnost týkající se rovnovážných cen článků z  $i$ -té úrovně:

$$\begin{aligned} (n_i + 1) \cdot \sum_{j=1}^{n_i} p_{ij}^{eq} &= \sum_{j=1}^{n_i} \left( \frac{a}{b} - \sum_{k=i}^m \sum_{l=1}^{n_k} c_{kl} + (n_i + 1) \cdot c_{ij} + \sum_{k=1}^{n_{i-1}} ((n_i + 1) \cdot \delta_{jk} - 1) \cdot p_{i-1,k} \right) = \\ &= n_i \cdot \left( \frac{a}{b} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} \right) + \sum_{j=1}^{n_i} \left( (n_i + 1) \cdot c_{ij} - \sum_{k=1}^{n_i} c_{ik} \right) + \sum_{k=1}^{n_{i-1}} \sum_{j=1}^{n_k} ((n_i + 1) \cdot \delta_{jk} - 1) \cdot p_{i-1,k} = \end{aligned}$$

$$\begin{aligned}
&= n_i \cdot \left( \frac{a}{b} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} \right) + \sum_{k=1}^{n_i} c_{ik} + \sum_{k=1}^{n_{i-1}} \left( p_{i-1,k} \cdot \sum_{j=1}^{n_i} ((n_i + 1) \cdot \delta_{jk} - 1) \right) = \\
&= n_i \cdot \left( \frac{a}{b} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} \right) + \sum_{k=1}^{n_i} c_{ik} + \sum_{k=1}^{n_{i-1}} p_{i-1,k},
\end{aligned}$$

přičemž poslední rovnost platí díky (2.29), celkem tedy z předchozího máme

$$\sum_{j=1}^{n_i} p_{ij}^{eq} = \frac{1}{n_i + 1} \cdot \left( n_i \cdot \left( \frac{a}{b} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} \right) + \sum_{k=1}^{n_i} c_{ik} + \sum_{k=1}^{n_{i-1}} p_{i-1,k} \right). \quad (2.32)$$

Dosazením (2.32) do (2.28) získáme

$$q = \frac{b}{\prod_{j=i+1}^m (n_j + 1)} \cdot \left( \frac{a}{b} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} - \frac{1}{n_i + 1} \cdot \left( n_i \cdot \left( \frac{a}{b} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} \right) + \sum_{k=1}^{n_i} c_{ik} + \sum_{k=1}^{n_{i-1}} p_{i-1,k} \right) \right),$$

odkud po úpravě dostaneme

$$q = \frac{b}{\prod_{j=i}^m (n_j + 1)} \cdot \left( \frac{a}{b} - \sum_{k=i}^m \sum_{l=1}^{n_k} c_{kl} - \sum_{k=1}^{n_{i-1}} p_{i-1,k} \right)$$

a indukční krok je dokázán a s ním i platnost vztahu (2.28). Dosazením  $i = 0$  (kvůli kterému jsme dodefinovali  $n_0 = 0$ ) pak vyjádříme poptávané množství v decentralizovaném řetězci jako

$$q = \frac{b}{\prod_{j=1}^m (n_j + 1)} \cdot \left( \frac{a}{b} - \sum_{j=1}^m \sum_{k=1}^{n_j} c_{jk} \right), \quad (2.33)$$

dosazením do poptávkové funkce dostaneme koncovou cenu

$$p_{m1}^{eq} = \frac{a}{b} - \frac{1}{\prod_{j=1}^m (n_j + 1)} \cdot \left( \frac{a}{b} - \sum_{j=1}^m \sum_{k=1}^{n_j} c_{jk} \right).$$

Odtud již snadno vyjádříme celkový zisk decentralizovaného řetězce jako součin poptávaného množství a celkové ziskové marže:

$$\pi_{DSC}^C = \frac{b}{\prod_{j=1}^m (n_j + 1)} \cdot \left( \frac{a}{b} - \sum_{j=1}^m \sum_{k=1}^{n_j} c_{jk} \right) \cdot \left( \frac{a}{b} - \frac{1}{\prod_{j=1}^m (n_j + 1)} \cdot \left( \frac{a}{b} - \sum_{j=1}^m \sum_{k=1}^{n_j} c_{jk} \right) - \sum_{j=1}^m \sum_{k=1}^{n_j} c_{jk} \right),$$

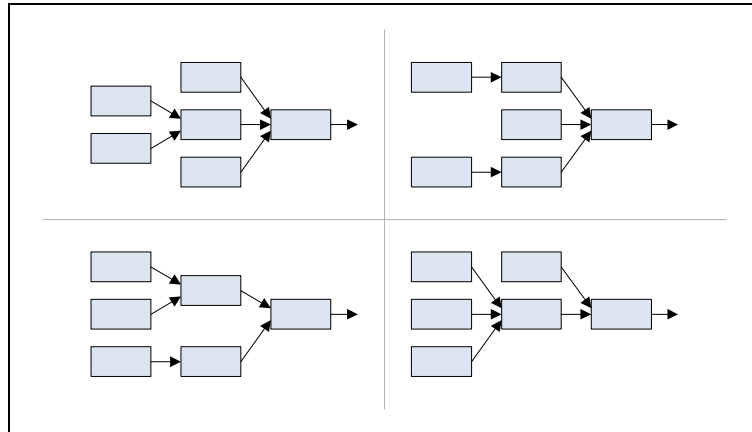
odkud již po úpravě a s přihlédnutím k tomu, že  $n_m = 1$ , vyjde dokazovaný vztah (2.26). ■

Efektivnost kombinovaného řetězce získáme jako podíl zisku decentralizovaného řetězce z tvrzení 2.6 a centralizovaného řetězce ze vztahu (2.25):

$$\varepsilon^C = \frac{2 \cdot \prod_{i=1}^m (n_i + 1) - 1}{\prod_{i=1}^{m-1} (n_i + 1)^2}. \quad (2.34)$$

I v případě decentralizovaného řetězce můžeme vyslovit podobný závěr jako v předchozích případech, totiž že efektivnost řetězce nezávisí na jednotkových nákladech v řetězci ani na parametrech poptávkové funkce. Ze vztahu (2.34) navíc plyne, že efektivnost řetězce je jednoznačně dána počtem článků v jednotlivých úrovních řetězce, konkrétní vazby mezi články z různých úrovní nemají z hlediska efektivnosti žádný vliv. Pokud se navíc počet článků ve dvou různých úrovních řetězce prohodí, celkový zisk a tedy ani efektivnost řetězce se nezmění. Tyto vztahy ilustruje obrázek 2-11. Všechny zobrazené struktury kombinovaného řetězce mají stejnou efektivnost.

**Obrázek 2-11: Stejně efektivní struktury kombinovaného řetězce**



Při podrobnějším pohledu na (2.34) můžeme rovněž zjistit, že efektivnost kombinovaného řetězce klesá rychleji s růstem počtu úrovní řetězce nežli s počtem článků v jednotlivých úrovních. Při daném počtu článků se efektivnost různých struktur kombinovaného řetězce vždy nachází mezi efektivnostmi paralelního a sériového řetězce se stejným počtem článků. Řetězce sériové jsou z uvažovaných struktur tou nejméně efektivní,

řetězce paralelní jsou naopak nejefektivnější a efektivnost řetězce kombinovaného je tím vyšší, čím více využívá vazeb many-to-one, které tvoří paralelní řetězec.

Následující tvrzení ukáže, jak je celkový zisk řetězce rozložen mezi jednotlivé články.

**TVRZENÍ 2.7:** Zisk článku  $ij$  v decentralizovaném kombinovaném řetězci v lineárním modelu je roven

$$\pi_{ij} = \frac{1}{\prod_{k=1}^i (n_k + 1)} \cdot \frac{b}{\prod_{k=1}^m (n_k + 1)} \cdot \left( \frac{a}{b} - \sum_{k=1}^m \sum_{l=1}^{n_k} c_{kl} \right)^2. \quad (2.35)$$

**DŮKAZ:** Pro výpočet zisku jednotlivých článků opět použijeme ziskové marže článků  $\mu$ . Ziskovou marži článku  $ij$  vyjádříme použitím konstanty  $\delta_{jk}$  definované stejně jako v důkazu tvrzení 2.6:

$$\begin{aligned} \mu_{ij} &= p_{ij}^{eq} - c_{ij} - \sum_{k=1}^{n_{i-1}} \delta_{jk} \cdot p_{i-1,k}^{eq} = \\ &= \frac{1}{n_i + 1} \cdot \left( \frac{a}{b} - \sum_{k=i}^m \sum_{l=1}^{n_k} c_{kl} + (n_i + 1) \cdot c_{ij} + \sum_{k=1}^{n_{i-1}} ((n_i + 1) \cdot \delta_{jk} - 1) \cdot p_{i-1,k}^{eq} \right), \end{aligned}$$

za rovnovážnou cenu článku  $ij$  dosadíme z (2.31),

$$\mu_{ij} = \frac{1}{n_i + 1} \cdot \left( \frac{a}{b} - \sum_{k=i}^m \sum_{l=1}^{n_k} c_{kl} + (n_i + 1) \cdot c_{ij} + \sum_{k=1}^{n_{i-1}} ((n_i + 1) \cdot \delta_{jk} - 1) \cdot p_{i-1,k}^{eq} \right) - c_{ij} - \sum_{k=1}^{n_{i-1}} \delta_{jk} \cdot p_{i-1,k}^{eq}$$

a po úpravě dostaneme vztah

$$\mu_{ij} = \frac{1}{n_i + 1} \cdot \left( \frac{a}{b} - \sum_{k=i}^m \sum_{l=1}^{n_k} c_{kl} - \sum_{k=1}^{n_{i-1}} p_{i-1,k}^{eq} \right), \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n_i. \quad (2.36)$$

Všimněme si, že se ze vzorce (2.36) zcela vytratily výrazy  $\delta_{jk}$ , které rozlišují jednotlivé články  $i$ -té úrovně. Dokážeme nyní, že zisková marže jednotlivých článků lze zapsat jako

$$\mu_{ij} = \frac{1}{\prod_{k=1}^i (n_k + 1)} \cdot \left( \frac{a}{b} - \sum_{k=1}^m \sum_{l=1}^{n_k} c_{kl} \right), \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n_i. \quad (2.37)$$

Všimněme si, že tento vztah vlastně říká, že články dané úrovně mají stejnou ziskovou marži (a tedy i stejný zisk). Při důkazu budeme postupovat indukcí podle čísla úrovně. Pro  $i = 1$  je vztah patrný přímo z (2.36). Zbývá dokázat indukční krok z  $i$  k  $i+1$ . Vyjádříme tvar ziskové marže pro libovolný článek  $j$  na úrovni  $i+1$  z (2.36)

$$\mu_{i+1,j} = \frac{1}{n_{i+1} + 1} \cdot \left( \frac{a}{b} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} - \sum_{k=1}^{n_i} p_{ik}^{eq} \right), \quad j = 1, 2, \dots, n_{i+1},$$

dosazením (2.32) za součet rovnovážných cen článků z úrovně  $i$  dostaneme

$$\mu_{i+1,j} = \frac{1}{n_{i+1} + 1} \cdot \left( \frac{a}{b} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} - \frac{1}{n_i + 1} \cdot \left( n_i \cdot \left( \frac{a}{b} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} \right) + \sum_{k=1}^{n_i} c_{ik} + \sum_{k=1}^{n_{i-1}} p_{i-1,k}^{eq} \right) \right),$$

z čehož po úpravě dostaneme

$$\mu_{i+1,j} = \frac{1}{n_{i+1} + 1} \cdot \frac{1}{n_i + 1} \left( \frac{a}{b} - \sum_{k=i+1}^m \sum_{l=1}^{n_k} c_{kl} + \sum_{k=1}^{n_{i-1}} p_{i-1,k}^{eq} \right), \quad j = 1, 2, \dots, n_{i+1}. \quad (2.38)$$

Porovnáme-li výraz na pravé straně rovnosti (2.38) s pravou stranou (2.36), získáme

$$\mu_{i+1,j} = \frac{1}{n_{i+1} + 1} \cdot \mu_{ik}, \quad j = 1, 2, \dots, n_{i+1}, \quad k = 1, 2, \dots, n_i.$$

Z poslední rovnosti plyne, že z platnosti vztahu (2.37) pro  $i$  můžeme vyvodit platnost téhož vztahu pro  $i+1$ , čili indukční krok a s ním i celé pomocné tvrzení (2.37) jsou dokázány. Zisk jednotlivých článků konečně dostaneme jako součin poptávaného množství (2.33) a ziskové marže (2.37), vyjde

$$\pi_{ij} = \frac{b}{\prod_{k=1}^m (n_k + 1)} \cdot \left( \frac{a}{b} - \sum_{k=1}^m \sum_{l=1}^{n_k} c_{kl} \right) \cdot \frac{1}{\prod_{k=1}^i (n_k + 1)} \cdot \left( \frac{a}{b} - \sum_{k=1}^m \sum_{l=1}^{n_k} c_{kl} \right)$$

a po jednoduché úpravě získáme dokazovaný vztah (2.35). ■

Z tvrzení 2.7 vyplývá, podobně jako v sériovém a paralelním řetězci, že rozložení zisku mezi jednotlivé články nezávisí na rozložení nákladů v řetězci. Můžeme tedy opět učinit stejné závěry ohledně pozitivních externalit nákladových úspor a přesunování výrobních procesů v rámci řetězce. Jak bylo naznačeno v průběhu posledního důkazu, všechny články stejné úrovně budou dosahovat stejného zisku, přičemž proporce zisků v jednotlivých úrovních je dána počtem jejich článků. Podobně jako v otázce efektivnosti, i v případě rozložení individuálních zisků můžeme konstatovat, že konkrétní podoba vazeb mezi články v navazujících úrovních nehraje žádnou roli.

Tvrzení 2.7 uzavírá kapitolu, která analyzovala efekt vícenásobné marginalizace v lineárním modelu. Následující kapitola se pokusí odhalit, nakolik se vlastnosti, které jsme v souvislosti s lineárním modelem popsali, zachovávají i v modelech řetězců pracujících s nelineárními poptávkovými a nákladovými funkcemi.



### 3 DODAVATELSKÉ ŘETĚZCE S NELINEÁRNÍMI FUNKCEMI

Při zkoumání efektu vícenásobné marginalizace v dodavatelských řetězcích s nelineárními poptávkovými a nákladovými funkcemi budeme stejně jako v lineárním modelu vycházet z modelu řetězce a chování článků, který je popsán v první kapitole. Pro jednoduchost se v této kapitole omezíme pouze na řetězce sériové a paralelní.

Jakmile začneme uvažovat v roli poptávky a nákladů nelineární funkce, ztratíme takřka okamžitě možnost odvodit analyticky rovnovážné ceny daného řetězce, natož pak jakékoli obecné vztahy (jak tomu bylo v případě lineárního modelu). Důvodem je jednoduchá skutečnost, že při odvozování rovnovážných strategií potřebujeme explicitní funkční vyjádření podmínky maximalizace zisku. Problém ilustruje následující příklad:

**PŘÍKLAD 5:** Uvažujme dodavatelský řetězec se dvěma články, s nulovými výrobními náklady obou článků a funkcí poptávky po koncové produkci ve tvaru

$$q = 300 - (p + 20)^{0,85}.$$

Zisková funkce článku 2 má potom tvar

$$\pi_2 = (300 - (p_2 + 20)^{0,85}) \cdot (p_2 - p_1).$$

Při dané ceně  $p_1$  se jedná o ryze konkávní funkci  $p_2$ , neboť

$$\frac{\partial^2 \pi_2}{\partial p_2^2} = -(p_2 + 20)^{-0,15} \cdot \left( 1,7 - \frac{0,1275}{p_2 + 20} \right) < 0.$$

Maximum této funkce bychom mohli najít z podmínky prvního řádu, která má po úpravě tvar

$$0,15 p_2 - 300 \cdot (p_2 + 20)^{0,15} + 20 + 0,85 p_1 = 0.$$

Řešení této rovnice lze sice pro známou cenu  $p_1$  určit s libovolnou přesností prostřednictvím numerických algoritmů, pro nalezení rovnovážných cen bychom ovšem potřebovali znát funkční vyjádření závislosti ceny  $p_2$  na ceně  $p_1$ , které z podmínky maximalizace zisku explicitně vyjádřit nelze (na rozdíl od lineárního modelu, kde jsme dokázali z podmínky prvního řádu vyjádřit tuto závislost v podobě lineární funkce).

Efekt vícenásobné marginalizace v dodavatelských řetězcích s nelineárními funkcemi budeme proto v této kapitole zkoumat formou výpočetních experimentů pomocí numerických algoritmů, které jsem navrhnul a následně implementoval v programovacím jazyce známého výpočetního prostředí Matlab společnosti Mathworks. Popis těchto algoritmů je předmětem oddílu 3.1. Výsledné spustitelné funkce se nachází na přiloženém CD, zdrojové kódy některých funkcí jsou v tištěné podobě součástí přílohy této práce. Ve stejném adresáři je rovněž k dispozici jednoduché uživatelské rozhraní, ze kterého je možné přehledně nastavit parametry týkající se dodavatelského řetězce a zmíněné algoritmy spustit; stručným popisem tohoto rozhraní se zabývá příloha P.1.

Veškeré algoritmy byly navrženy co nejobecněji tak, aby je bylo možné použít na velkou škálu nelineárních funkcí. Pro účely výpočetních experimentů bylo nutné vybrat konkrétní podobu těchto funkcí. Oddíl 3.2 popíše funkce tvary funkcí, které byly při experimentech použity a které jsou rovněž k dispozici pro snadné spuštění v uživatelském prostředí. Výsledky samotných výpočetních experimentů přinese oddíl 3.3. Kapitulu zakončí oddíl 3.4, ve kterém se ještě zastavíme nad otázkou, jaké změny výsledků bychom se mohli dočkat, pokud bychom jednotlivým článkům umožnili volit výstupní ceny z konečné diskrétní škály.

### 3.1 Numerické algoritmy pro řetězce s nelineárními funkcemi

Pro zkoumání efektu vícenásobné marginalizace v nelineárním prostředí jsou na přiloženém CD k dispozici dvě základní funkce: *equil* pro zkoumání sériových řetězců a *equilpar* pro řetězce paralelní (každá z funkcí v jazyce Matlab je uložena ve stejnojmenném souboru s příponou „.m“, pro spuštění těchto i všech následujících funkcí je třeba mít nainstalováno prostředí Matlab verze 6.1 nebo vyšší). Cílem obou funkcí je provést následující:

- nalézt celkový zisk a cenu koncové produkce v centralizovaném řetězci,
- nalézt rovnovážné ceny a rozložení zisku v decentralizovaném řetězci,
- v případě potřeby zobrazit charakteristiky dodavatelského řetězce v podobě grafů.

Největším problémem je (podobně jako v případě lineárního modelu) nalezení rovnovážných cen (odtud také pramení název obou funkcí). V pododdílech 3.1.1 a 3.1.2 popíšeme, jak se v uvedených funkcích rovnovážné ceny hledají. Než tak učiníme, je třeba definovat některé vlastnosti použitých nelineárních funkcí a zavést potřebné značení.

Jelikož se budeme nadále zabývat pouze sériovými a paralelními řetězci, články řetězce budeme indexovat stejně jako v oddílech 2.1 a 2.2. Budeme dále předpokládat, že funkce poptávky i nákladů jsou reálné spojité nezáporné funkce jedné proměnné. Poptávku bude tedy představovat spojitá funkce, která ceně koncové produkce přiřadí poptávané množství:

$$\delta : R_0^+ \rightarrow R_0^+, q = \delta(p_n), \quad (3.1)$$

nákladová funkce  $i$ -tého článku bude opět spojitá funkce, která přiřadí vyráběnému množství celkové náklady:

$$\gamma_i : R_0^+ \rightarrow R_0^+, tc_i = \gamma_i(q), \quad i = 1, 2, \dots, n. \quad (3.2)$$

Celkové náklady každého článku můžeme potom vyjádřit jako funkci ceny koncové produkce

$$tc_i = \gamma_i(\delta(p_n)), \quad i = 1, 2, \dots, n,$$

zisk celého řetězce vypočteme jako rozdíl celkových tržeb z prodeje a součtu celkových nákladů všech článků:

$$\pi_{SC} = p_n \cdot \delta(p_n) - \sum_{i=1}^n \gamma_i(\delta(p_n)). \quad (3.3)$$

Centralizovaný řetězec bude volit takovou cenu koncové produkce, která ziskovou funkci maximalizuje. Taková cena ovšem samozřejmě existuje pouze tehdy, má-li funkce (3.3) globální maximum. To ovšem můžeme pro praktické příklady dodavatelských řetězců zajistit, neboť zpravidla existuje konečná cena koncové produkce, za kterou si již žádný spotřebitel není ochoten koupit nabízené zboží a hodnota poptávkové funkce pro tuto a každou vyšší cenu je tudíž nulová. Nejnižší cenu, která vynuluje poptávku, budeme označovat nadále jako  $p_{max}$ . Na intervalu  $[p_{max}; +\infty)$  je potom funkce (3.3) konečná a konstantní, na intervalu  $[0; p_{max}]$  je z definice spojitá, a podle Weierstrassovy věty zde tudíž nabývá globálního maxima (viz např. [16]).

Pro maximalizaci funkce (3.3) v prostředí Matlab při hledání ceny stanovené centralizovaným řetězcem jsem použil vestavěné funkce *fminbnd*, která minimalizuje spojitou funkci jedné proměnné na omezeném intervalu pomocí metody zlatého řezu. Vzhledem k požadavku omezenosti intervalu při použití této funkce je proto cenu  $p_{max}$  potřeba zadávat jako jeden z parametrů při spouštění funkcí *equil* a *equilpar*.

### 3.1.1 Algoritmus pro nalezení rovnovážných cen v sériovém řetězci

Při hledání rovnovážných cen v sériovém řetězci budeme rámcově vycházet z postupu, který jsme použili v případě sériových řetězců v lineárním modelu (tj. například v příkladu 1, nebo obecněji na začátku důkazu tvrzení 2.2). Základní myšlenka je následující. Budeme vždy začínat od posledního článku řetězce, který je v přímém styku se spotřebitelem (čili s poptávkovou funkcí). Z podmínky maximalizace zisku vyjádříme optimální cenu posledního článku jakožto funkci ceny článku předcházejícího. Odtud dostaneme poptávané množství jakožto funkci ceny předcházejícího článku, což je vlastně individuální poptávková funkce, které čelí předposlední článek, pokud poslední článek maximalizuje svůj zisk. Pro článek

předposlední nyní opakujeme stejný postup jako pro článek poslední a postupně pokračujeme až k začátku řetězce. Pro první článek řetězce již z podmínky optimalizace zisku dostaneme konkrétní hodnotu rovnovážné ceny a postupným zpětným dosazením do cenových funkcí získáme rovnovážné ceny článků navazujících. Celý algoritmus můžeme zapsat v obecné podobě následujícím způsobem:

**Krok 1:** Formuluj ziskovou funkci článku  $n$ :

$$\pi_n(p_n, p_{n-1}) = (p_n - p_{n-1}) \cdot \delta(p_n) - \gamma_n(\delta(p_n)).$$

**Krok 2:** Nalezni cenovou reakční funkci  $p_n^{opt}$  definovanou předpisem

$$p_n^{opt}(p_{n-1}) = \arg \max_{p_n \in R_0^+} \pi_n(p_n, p_{n-1}).$$

**Krok 3:** Polož  $i = n-1$ .

**Krok 4:** Formuluj funkci  $\psi$  jako složené zobrazení

$$\psi = p_{i+1}^{opt} \circ p_{i+2}^{opt} \circ \dots \circ p_n^{opt} \circ \delta.$$

**Krok 5:** Formuluj ziskovou funkci článku  $i$ :

$$\pi_i(p_i, p_{i-1}) = (p_i - p_{i-1}) \cdot \psi(p_i) - \gamma_i(\psi(p_i)).$$

**Krok 6:** Pokud  $i = 1$ , jdi na krok 9, jinak pokračuj krokem 7.

**Krok 7:** Nalezni cenovou reakční funkci  $p_i^{opt}$  definovanou předpisem

$$p_i^{opt}(p_{i-1}) = \arg \max_{p_i \in R_0^+} \pi_i(p_i, p_{i-1}).$$

**Krok 8:** Vrať se na krok 4.

**Krok 9:** Najdi rovnovážnou cenu prvního článku  $p_1^{eq}$  podle vztahu

$$p_1^{eq} = \arg \max_{p_1 \in R_0^+} \pi_1(p_1, 0).$$

**Krok 10:** Polož  $i = i+1$ .

**Krok 11:** Vypočti rovnovážnou cenu  $i$ -tého článku jako

$$p_i^{eq} = p_i^{opt}(p_{i-1}^{eq}).$$

**Krok 12:** Pokud  $i < n$ , vrať se na krok 10, jinak skonči.

Podívejme se nyní, jak jsou některé z kroků implementovány v podobě funkcí jazyka Matlab. Zdrojový kód všech citovaných funkcí je k dispozici kromě příloženého CD také v tištěné podobě v příloze.

**Kroky 1 a 5:** Ziskové funkce jednotlivých článků jsou formulovány pomocí funkce *profit*. Vstupní parametry této funkce jsou jednak hodnoty proměnných  $p_i$  a  $p_{i-1}$ , jednak parametry popisující poptávkovou a nákladovou funkci. Lze proto tuto funkci použít pro oba kroky 1 a 5 s tím, že v kroku 5 se namísto poptávkové funkce  $\delta$  dosadí funkce  $\psi$  z kroku 4. Funkce *profit* vrací ve skutečnosti zisk článků vynásobený konstantou  $-1$ , a to z toho důvodu, že jej budeme chtít maximalizovat, zatímco vestavěné extremalizační funkce v jazyce Matlab nabízejí výhradně možnost minimalizace.

**Kroky 2 a 7:** Z celého algoritmu jsou tyto dva (vzájemně velmi podobné) kroky ty nejproblematictější. Z popisu těchto kroků vyplývá, že pro každý článek hledáme funkční vztah mezi výstupní cenou, kterou tento článek volí, aby maximalizoval individuální zisk, a jeho vstupní cenou (tj. výstupní cenou článku předchozího). Jinými slovy, funkce  $p_i^{opt}$ , kterou pro každý článek hledáme, by měla každé vstupní ceně  $p_{i-1}$  přiřadit cenu  $p_i$ , která optimalizuje zisk článku  $i$ .

Tato funkce je pro běžné řetězce dobře definovaná, neboť stejnou argumentací, kterou jsme užili v diskuzi o zisku centralizovaného řetězce, bychom mohli ukázat, že zisková funkce na intervalu  $[0; +\infty)$  globálního maxima. Pravdou je, že definice funkce  $p_i^{opt}$  neposkytuje žádný návod k tomu, kterou z cen  $p_i$  si vybrat, pokud globální maximum není ostré. Tento nedostatek by bylo možné odstranit tak, že bychom definici rozšířili v tom smyslu, že v případě většího počtu maximalizujících hodnot  $p_i$  vybereme např. jejich infimum (které vzhledem ke spojitosti ziskové funkce rovněž splňuje podmínku maximalizace). V případě předložené softwarové implementace jsou však tyto úvahy irelevantní, neboť vestavěná funkce jazyka Matlab *fminbnd*, kterou jsem pro nalezení příslušných extrémů použil, již konkrétní hodnotu  $p_i$  vybere sama.

Definice reakční cenové funkce  $p_i^{opt}$  ovšem přináší důležitou otázku týkající se jejího tvaru. Implicitní forma této definice nám totiž prakticky znemožňuje jakékoli závěry ohledně tvaru této funkce vyslovit. Nemůžeme například zaručit, že výsledná funkce bude spojitá. Přitom nespojitost této funkce by nám mohla přinést zásadní problémy při realizaci uvedeného postupu. Jakmile by se tato nespojitost přenesla do ziskové funkce předcházejících článků, ztratili bychom okamžitě možnost nalézt maximum této funkce pomocí standardních algoritmů pro hledání extrémů funkcí spojitých.

Provedl jsem proto několik předběžných výpočetních experimentů, abych tvar funkcí  $p_i^{opt}$  ověřil. Jak se ukazuje, pro všechny druhy nelineárních poptávkových a nákladových funkcí, které budeme v této kapitole zkoumat (viz oddíl 3.2) jsou funkce  $p_i^{opt}$  neklesající a spojitě s výjimkou nejvýš jednoho bodu nespojitosti. Ke skoku v bodu nespojitosti dochází přitom vždy ve chvíli, kdy vstupní cena daného článku vzroste na takovou úroveň, že článek při žádné výstupní ceně není schopen realizovat kladný zisk, začne proto volit cenu tak vysokou, aby vynulovala poptávku, a dosažený zisk je potom nulový. Situaci ilustruje následující příklad.

**PŘÍKLAD 6:** Budeme uvažovat dodavatelský řetězec tvořený dvěma články, kde kladná část poptávkové funkce má hyperbolický tvar

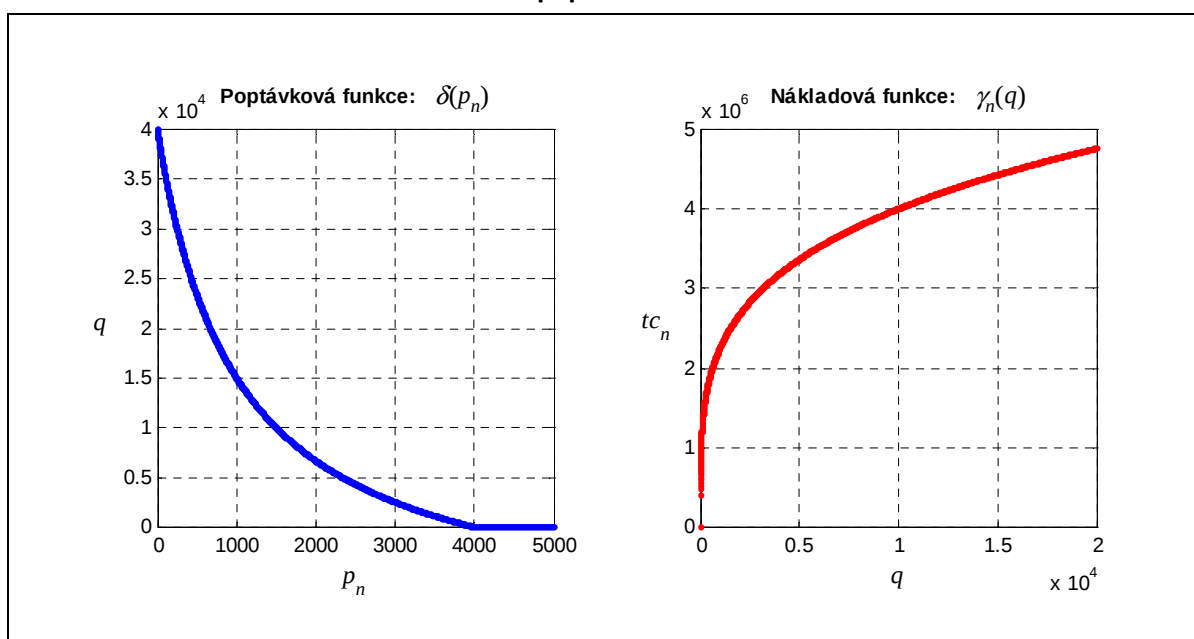
$$\delta(p) = \frac{50.000.000}{1000 + p} - 10.000$$

a nákladová funkce druhého (koncového) článku je dána předpisem

$$\gamma_2(q) = 400.000 \cdot \sqrt[4]{q}.$$

Obě funkce znázorňuje graf 3-1.

**Graf 3-1: Příklad 6 – poptávková a nákladová funkce**



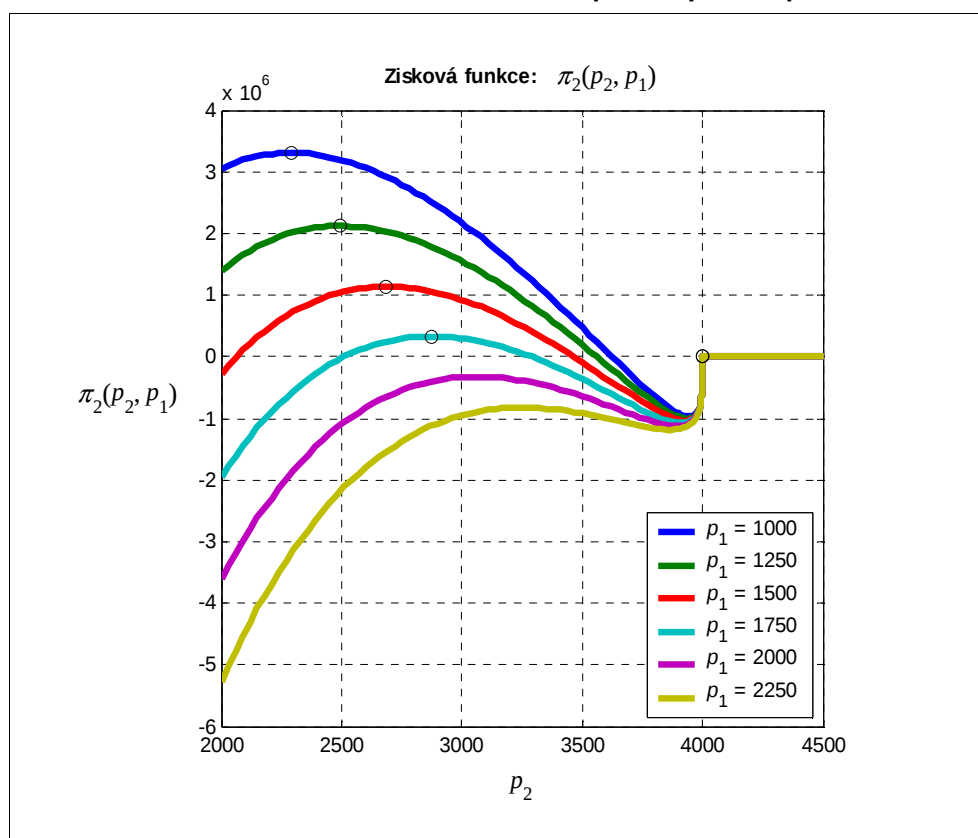
Jak je z grafu 3-1 patrné, pro vysoké hodnoty ceny koncové produkce budeme namísto záporných hodnot, které bychom obdrželi z funkčního předpisu poptávky, uvažovat poptávané množství rovné nule. Cena  $p_{max}$  potom činí 4000 Kč. Můžeme si dále všimnout, že nákladová funkce vykazuje úspory z rozsahu. Zisková funkce druhého článku má při dané úrovni vstupní ceny  $p_1$  tvar

$$\pi_2(p_2) = (p_2 - p_1) \cdot \delta(p_2) - \gamma_2(\delta(p_2)).$$

Graf 3-2 na následující straně ukazuje, jak se mění tvar této ziskové funkce, pokud cena  $p_1$  roste z hodnoty 1000 Kč po přírůstcích 250 Kč postupně až na 2250 Kč. V grafu jsou ukázány hodnoty ziskové funkce pro ceny  $p_2$  z intervalu  $[2000; 4500]$ , na intervalu  $[0; 2000]$  jsou všechny uvedené funkce vesměs rostoucí a na intervalu  $[4500; +\infty)$  jsou konstantní nulové. Na jednotlivých křivkách jsou černými kroužky naznačena globální maxima. Z grafu je patrné, že každá z křivek má však dvě maxima *lokální*. Jedno z nich je tvořeno cenou  $p_{max}$  a z ní

plynoucím nulovým ziskem (stejný výsledek přinese i libovolná vyšší cena). Druhé z nich má hodnotu ceny  $p_2$  nižší a představuje nejlepší cenu, kterou může při daných podmínkách článek 2 volit tak, aby poptávané množství zůstalo kladné. Při nízké ceně vstupu  $p_1$  může dosahovat druhý článek kladného zisku a bude volit lokální maximum spojené s nižší cenou výstupu  $p_2$ . Jak cena  $p_1$  roste, lokální maximum se posunuje doprava (což znamená, že na zvýšenou vstupní cenu musí článek 2 reagovat vyšší cenou  $p_2$ , pokud chce optimalizovat zisk) a jeho hodnota klesá. Pro cenu  $p_1$  mezi 1750 Kč a 2000 Kč se hodnota tohoto lokálního maxima dostane do záporných hodnot, což znamená, že článek 2 již nemůže pro žádnou cenu  $p_2$  dosahovat kladného zisku. Aby se vyhnul ztrátě, volí článek 2 rázem druhé z obou lokálních maxim, tedy cenu  $p_{max}$ , čímž dojde ke zmiňovanému skoku.

**Graf 3-2: Příklad 6 – zisková funkce článku 2 při různých vstupních cenách**



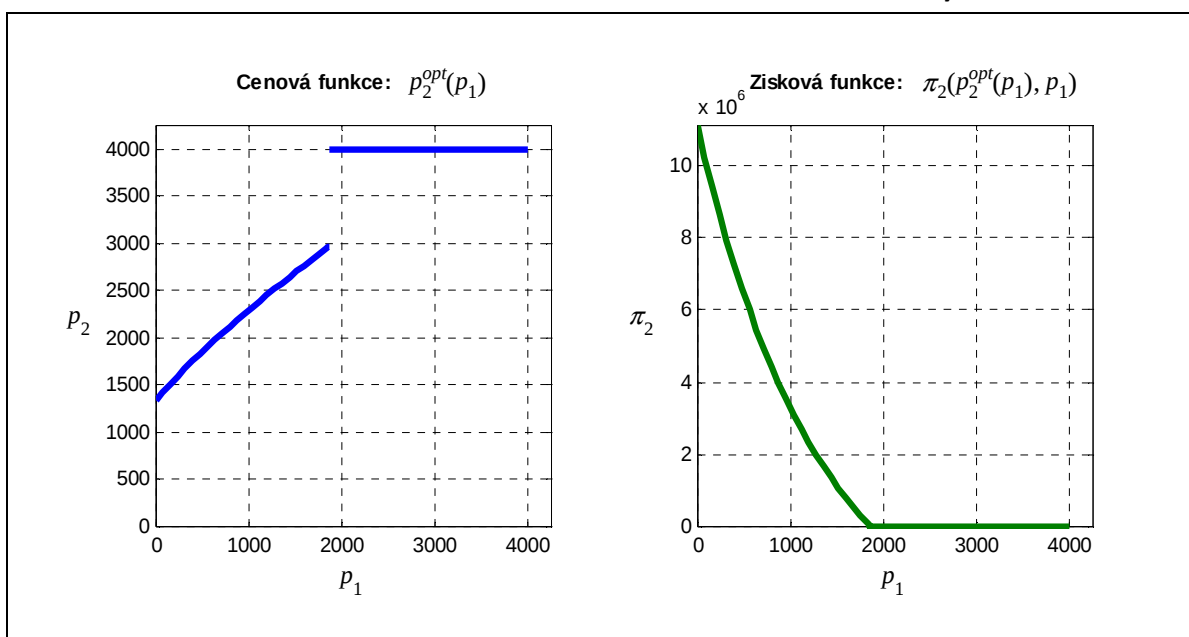
Levá část grafu 3-3 na následující straně ukazuje hodnoty cenové reakční křivky  $p_2^{opt}$  pořízené tak, že pro všechny celočíselné hodnoty  $p_1$  z intervalu  $[0; 4000]$  byla nalezena cena  $p_2$  maximalizující zisk druhého článku. Na grafu je dobře patrný bod nespojitosti (tj. skok) reakční funkce, ke kterému dochází mezi cenami  $p_1 = 1876$  Kč a  $p_1 = 1877$  Kč. V pravé části grafu pak vidíme hodnoty nejvyšší úrovně zisku, jaké může článek 2 při různých hodnotách vstupní ceny dosáhnout (tj. hodnoty spojené výstupní cenou na úrovni hodnoty reakční cenové funkce  $p_2^{opt}$ ). Tato funkce maximální výše zisku je již funkcí spojitou.

Jak již bylo řečeno, obdobných výsledků jako v předchozím příkladě bychom se dočkali i při použití všech dalších tvarů poptávkových a nákladových funkcí, kterými se budeme v této kapitole zabývat. Reakční cenové křivky jsem se proto rozhodl modelovat pomocí funkce, kterou jsem pracovníčně nazval „odříznutým polynomem“ a kterou jsem v jazyce Matlab implementoval ve funkci *polycut*: pro první, rostoucí část reakční křivky je použita aproximace polynomem, a v hodnotě vstupní ceny, při které dochází ke skoku, je tento polynom „odříznut“ a nahrazen konstantní funkcí na úrovni  $p_{max}$ . Matematický předpis celé funkce pro nezápornou cenu  $p$  můžeme zapsat jako

$$\begin{aligned} \text{polycut}(p) &= \sum_{j=0}^m a_j p^j, & \text{pokud } p < p_{\text{cut-off}}, \\ &= p_{\text{max}} & \text{jinak,} \end{aligned}$$

kde  $m$  je stupeň použitého polynomu,  $a_j$  jsou koeficienty u  $j$ -té mocniny proměnné v tomto polynomu a  $p_{\text{cut-off}}$  je hodnota vstupní ceny, při níž dochází ke skoku.

**Graf 3-3: Příklad 6 – cenová reakční funkce a funkce maximální výše zisku**



Nalezení parametrů  $a_j$  a  $p_{\text{cut-off}}$  se provádí v prostředí Matlab pomocí funkce *pricepoly* následujícím způsobem:

- Škála vstupních cen 0 až  $p_{\text{max}}$  je rozdělena na dostatečný počet ekvidistantních hodnot (jejich přesný počet je nastavitelný jedním ze vstupních parametrů funkce).
- Pro všechny tyto vstupní jsou hodnoty funkce  $p_i^{\text{opt}}$  vypočteny přímo z definice, tj. maximalizací ziskové funkce daného článku.
- Podle výsledků se určí provizorní hodnota  $p_{\text{cut-off}}$  jako nejnižší vstupní cena, při které volí článek cenu  $p_{\text{max}}$ . Vzhledem k tomu, že bod nespojitosti  $p_{\text{cut-off}}$  je vhodné

najít co nej přesněji, je jeho výpočet následně ještě zpřesněn desetinásobným zjemněním škály v okolí provizorní hodnoty.

- Funkční hodnoty vstupních cen nižších, nežli je  $p_{cut-off}$ , jsou aproximovány polynomm stupně zadaného v podobě vstupního parametru funkce *pricepoly*. Tím již získáme hodnoty koeficientů  $a_j$  odříznutého polynomu. Pro aproximaci polynomm v prostředí Matlab je využito vestavěné funkce *polyfit*. Pro všechny zkoumané funkce se jako dostačující jeví polynom druhého stupně.

Výstupem z funkce *pricepoly* jsou jak koeficienty  $a_j$  aproximujícího polynomu, tak bod nespojitosti  $p_{cut-off}$ .

**Krok 4:** Funkce  $\psi$  je podle kroku 4 definována jako složení poptávkové funkce a několika reakčních cenových funkcí a představuje závislost poptávaného množství na ceně některého z článků  $1, 2, \dots, n-1$ . V softwarové implementaci je  $\psi$  ztělesněna funkcí *dempoly*. V této funkci se počítá s tím, že funkce  $p_i^{opt}$  mají tvar oříznutého polynomu, a vstupní parametry určí počet a tvar reakčních funkcí, které se s poptávkovou funkcí  $\delta$  skládají.

Ostatní kroky uvedeného algoritmu jsou již, domnívám se, poměrně přímočaré. Ve zdrojovém kódu k funkci *equil*, která tento algoritmus realizuje, a ve funkcích, které *equil* využívá (a které byly výše zmíněny), je případně obsaženo mnoho poznámek, které celý postup komentují.

### 3.1.2 Algoritmus pro nalezení rovnovážných cen v paralelním řetězci

Při tvorbě algoritmu pro paralelní řetězce budeme v mnoha ohledech kopírovat způsob, kterým jsme hledali rovnovážné ceny v lineárním modelu. Zásadním problémem, se kterým se musíme v případě paralelních řetězců potýkat, je nalezení takových cen článků v první úrovni řetězce, které by splňovaly podmínky Nashovy rovnováhy. Připomeňme, že Nashova rovnováha požaduje, aby při daných cenách ostatních článků optimalizoval každý článek prostřednictvím volené ceny svůj zisk. V lineárním modelu bylo možné tyto požadavky shrnout v podobě soustavy lineárních rovnic, která představovala podmínky maximalizace zisku všech článků a o které jsme mohli dokonce prohlásit, že má vždy právě jedno řešení. Nic takového není ovšem v nelineárním modelu možné.

Použitý algoritmus proto hledá Nashovu rovnováhu s použitím informací z funkcí, které jsou analogií k cenovým reakčním křivkám ze sériového řetězce. Myšlenka algoritmu je následující. Zisk posledního článku můžeme vyjádřit jako

$$\pi_n = \left( p_n - \sum_{j=1}^{n-1} p_j \right) \cdot \delta(p_n) - \gamma_n(\delta(p_n)),$$

čili zisk posledního článku je závislý na *součtu* cen článků z první úrovně (nezáleží tedy na výši jednotlivých cen, ze kterých je tento součet složen). Zisk každého článku na první úrovni

řetězce bude proto rovněž závislý na *součtu* cen ostatních článků z první úrovně. Pro články z první úrovně řetězce budeme hledat reakční cenové funkce  $p_i^{opt}$ , které budou nyní vyjadřovat, jakou cenu bude článek  $i = 1, 2, \dots, n-1$  volit při daném součtu cen všech ostatních článků z první úrovně. Zapišeme

$$p_i^{opt} \left( \sum_{\substack{j=1, \\ j \neq i}}^{n-1} p_j \right) = \arg \max_{p_i \in R_0^+} \pi_i \left( p_i, \sum_{\substack{j=1, \\ j \neq i}}^{n-1} p_j \right). \quad (3.4)$$

Kombinace cen článků z první úrovně  $p_i^{eq}$  odpovídající Nashově rovnováze pak musí pro všechna  $i = 1, 2, \dots, n-1$  splňovat

$$p_i^{eq} = p_i^{opt} \left( \sum_{\substack{j=1, \\ j \neq i}}^{n-1} p_j^{eq} \right).$$

Abychom takovou kombinaci našli, pokusíme se najít alternativu k funkci  $p_i^{opt}$ , která bude vyjadřovat funkční závislost mezi zisk optimalizující cenou daného článku a součtem cen *všech* článků z první úrovně (nikoli jen *všech ostatních* článků). Jakmile takovou funkci nalezneme (nazvěme ji  $\tilde{p}_i^{opt}$ ), budeme moci podmínky Nashovy rovnováhy formulovat jako

$$p_i^{eq} = \tilde{p}_i^{opt} \left( \sum_{j=1}^{n-1} p_j^{eq} \right), \quad i = 1, 2, \dots, n-1. \quad (3.5)$$

Sečteme-li nyní tyto podmínky přes všechna  $i$ , získáme nutnou podmínku, kterou musí splňovat součet cen v Nashově rovnováze:

$$\sum_{i=1}^{n-1} p_i^{eq} = \sum_{i=1}^{n-1} \tilde{p}_i^{opt} \left( \sum_{j=1}^{n-1} p_j^{eq} \right).$$

Abychom našli součet cen článků z první úrovně, který odpovídá Nashově rovnováze, stačí nám tedy najít nulovou hodnotu reálné funkce jedné proměnné  $s$  (jako *součet*) dané předpisem

$$f(s) = s - \sum_{i=1}^{n-1} \tilde{p}_i^{opt}(s).$$

Jak za chvíli ukážeme, pro zkoumané tvary poptávkových a nákladových funkcí je  $f$  na intervalu, který je pro nás v jistém slova smyslu relevantní, spojitá, a nalezení nulového bodu na tomto intervalu pomocí numerických algoritmů nebude žádný problém. Dosazením nalezené hodnoty  $s$ , která splňuje  $f(s) = 0$ , namísto součtu cen článků v první úrovni do (3.5) již pak snadno získáme rovnovážné ceny.

Důležitým bodem celého postupu je přechod od funkcí  $p_i^{opt}$ , které jsou definovány pomocí (3.4) a které vyjadřují funkční vztah mezi optimální cenou  $i$ -tého článku z první úrovně a součtem cen *všech ostatních* článků z první úrovně, k funkcím  $\tilde{p}_i^{opt}$ , které popisují vztah mezi optimální cenou jednoho z článků z první úrovně a součtem cen *všech* článků z první úrovně (tj. včetně článku  $i$ ). Označme nyní pro přehlednost pro  $i = 1, 2, \dots, n-1$

$$s_i = \sum_{\substack{j=1, \\ j \neq i}}^{n-1} p_j, \quad (3.6)$$

Chceme-li z  $p_i^{opt}$  vytvořit  $\tilde{p}_i^{opt}$ , musí  $\tilde{p}_i^{opt}$  splňovat pro všechna  $s_i$

$$\tilde{p}_i^{opt}(p_i^{opt}(s_i) + s_i) = p_i^{opt}(s_i). \quad (3.7)$$

Funkce splňující (3.7) ovšem obecně nemusí existovat. Pokud by totiž existovaly dva různé součty cen ostatních článků z první úrovně  $s_{i1}$  a  $s_{i2}$  takové, že

$$p_i^{opt}(s_{i1}) + s_{i1} = p_i^{opt}(s_{i2}) + s_{i2}, \quad (3.8)$$

může vztah (3.7) platit přirozeně pouze buď pro  $s_{i1}$ , nebo pro  $s_{i2}$ . Ukážeme ale, že taková situace nemůže pro zkoumané řetězce nastat. Přesněji řečeno, budeme zkoumat funkci  $g_i$  jedné proměnné  $s_i$  s předpisem

$$g_i(s_i) = p_i^{opt}(s_i) + s_i,$$

a ověříme, že je prostá. V takovém případě už zmiňované nebezpečí nehrozí, neboť potom vztah (3.8) implikuje  $s_{i1} = s_{i2}$ .

Předně si můžeme uvědomit, že pokud bude sklon funkce  $p_i^{opt}$  větší než  $-1$ , pak funkce  $g_i$  bude rostoucí, a tudíž prostá. Skutečnost, že sklon je větší než  $-1$ , budeme demonstrovat následujícím způsobem. Víme, že koncový článek se orientuje pouze podle součtu cen všech článků z první úrovně řetězce. Za podmínky, že se koncový článek volbou výstupní ceny snaží maximalizovat individuální zisk, můžeme proto vyjádřit poptávané množství jako funkci součtu cen článků z prvního řetězce. Označíme-li tuto funkci  $\psi$ , můžeme psát pro  $i$ -tý článek z první úrovně

$$q = \psi\left(\sum_{j=1}^{n-1} p_j\right) = \psi(p_i + s_i).$$

Ziskovou funkci  $i$ -tého článku pak můžeme formulovat jako

$$\pi_i(p_i, s_i) = p_i \cdot \psi(p_i + s_i) - \gamma_i(\psi(p_i + s_i)). \quad (3.9)$$

Představme si, že při daném součtu cen ostatních článků z první úrovně  $s_{i1}$  optimalizuje  $i$ -tý článek svůj zisk prostřednictvím ceny  $p_{i1}$ . Je-li zisková funkce  $i$ -tého článku v bodě  $(p_{i1}, s_{i1})$  diferencovatelná, pak nutně

$$\frac{\partial \pi_i}{\partial p_i}(p_{i1}, s_{i1}) = 0. \quad (3.10)$$

Předpokládejme nyní (pro spor), že sklon funkce  $p_i^{opt}$  je roven  $-1$ . Pokud se součet cen ostatních článků zvýší o  $\varepsilon > 0$ , zareaguje na tuto změnu článek  $i$  tak, že sníží svou výstupní cenu rovněž o  $\varepsilon$ . Zavedeme novou funkci  $\tilde{\pi}_i$  danou předpisem

$$\tilde{\pi}_i(p_i) = \pi_i(p_i - \varepsilon, s_{i1} + \varepsilon). \quad (3.11)$$

Dosazením (3.9) do (3.11) získáme jiné vyjádření funkce  $\tilde{\pi}_i$

$$\begin{aligned} \tilde{\pi}_i(p_i) &= \pi_i(p_i - \varepsilon, s_{i1} + \varepsilon) = \\ &= (p_i - \varepsilon) \cdot \psi(p_i - \varepsilon + s_{i1} + \varepsilon) - \gamma_i \cdot \psi(p_i - \varepsilon + s_{i1} + \varepsilon) = \\ &= p_i \cdot \psi(p_i + s_{i1}) - \gamma_i \cdot \psi(p_i + s_{i1}) - \varepsilon \cdot \psi(p_i + s_{i1}) = \\ &= \pi_i(p_i, s_{i1}) - \varepsilon \cdot \psi(p_i + s_{i1}). \end{aligned}$$

Z předchozího plyne, že pokud je sklon  $p_i^{opt}$  roven  $-1$  a cena  $p_{i1}$  maximalizuje ziskovou funkci  $\pi_i$  pro součet  $s_{i1}$ , pak má v bodě  $p_{i1}$  lokální maximum i funkce  $\tilde{\pi}_i$ , neboť článek  $i$  zde reaguje podle své reakční křivky. Derivaci funkce  $\tilde{\pi}_i$  v bodě  $p_{i1}$  pak můžeme vyjádřit jako

$$\frac{\partial \tilde{\pi}_i}{\partial p_i}(p_{i1}) = \frac{\partial \pi_i}{\partial p_i}(p_{i1}, s_{i1}) - \varepsilon \cdot \frac{\partial \psi}{\partial p_i}(p_{i1} + s_{i1}) = -\varepsilon \cdot \frac{\partial \psi}{\partial p_i}(p_{i1} + s_{i1}), \quad (3.12)$$

přičemž druhá rovnost platí vzhledem k (3.10). Přitom pravá strana je zpravidla pro všechna  $p_{i1}$  a  $s_{i1}$  kladná, neboť derivace  $\partial \psi / \partial p$  je záporná: zvyšují-li články na první úrovni svou výstupní cenu, bude jako reakci na toto zvýšení zdražovat koncovou produkci i poslední článek a poptávané množství se tím bude zmenšovat. Výraz (3.12) má kladnou hodnotu, což znamená, že bod  $p_{i1} - \varepsilon$  se nachází nalevo od lokálního maxima funkce  $\pi_i(p_i, s_{i1} + \varepsilon)$ . Pokud tedy dojde ke zvýšení součtu cen ostatních článků z první úrovně o  $\varepsilon$ , bude článek  $i$  optimalizovat zisk při ceně vyšší nežli  $p_{i1} - \varepsilon$ , a sklon reakční křivky  $p_i^{opt}$  je tudíž větší než  $-1$ , což je spor s naším předpokladem.

Předchozí úvahy nelze pokládat za korektní důkaz tvrzení o sklonu  $p_i^{opt}$ , neboť jsme vycházeli z celé řady předpokladů, jejichž platnost by bylo třeba zajistit: jedním z nich je například diferencovatelnost ziskové funkce  $i$ -tého článku, více pozornosti by si zasluhoval rovněž požadavek záporného sklonu funkce  $\psi$ . Při použití poptávkových a nákladových funkcí, které budeme v této kapitole zkoumat, jsou však tyto předpoklady vesměs splněny a funkce  $\tilde{p}_i^{opt}$  definované v (3.7) tudíž budou existovat.

Postup pro nalezení rovnovážných cen v paralelním řetězci pak vypadá následovně:

**Krok 1:** Formuluj ziskovou funkci článku  $n$ :

$$\pi_n \left( p_n, \sum_{j=1}^{n-1} p_j \right) = \left( p_n - \sum_{j=1}^{n-1} p_j \right) \cdot \delta(p_n) - \gamma_n(\delta(p_n)).$$

**Krok 2:** Nalezni cenovou reakční funkci  $p_n^{opt}$  definovanou předpisem

$$p_n^{opt} \left( \sum_{j=1}^{n-1} p_j \right) = \operatorname{argmax}_{p_n \in R_0^+} \pi_n \left( p_n, \sum_{j=1}^{n-1} p_j \right).$$

**Krok 3:** Formuluj funkci  $\psi$  jako složené zobrazení:

$$p_n^{opt} \circ \delta.$$

**Krok 4:** Polož  $i = 1$ .

**Krok 5:** Formuluj ziskovou funkci článku  $i$  jako

$$\pi_i(p_i, s_i) = p_i \cdot \psi(p_i + s_i) - \gamma_i(\psi(p_i + s_i)).$$

**Krok 6:** Nalezni cenovou reakční funkci  $p_i^{opt}$  definovanou předpisem

$$p_i^{opt}(s_i) = \operatorname{argmax}_{p_i \in R_0^+} \pi_i(p_i, s_i).$$

**Krok 7:** Najdi funkci  $\tilde{p}_i^{opt}$  která splňuje

$$\tilde{p}_i^{opt}(p_i^{opt}(s_i) + s_i) = p_i^{opt}(s_i).$$

**Krok 8:** Pokud  $i < n-1$ , polož  $i = i+1$  a jdi na krok 5.

**Krok 9:** Najdi hodnotu  $s$  splňující

$$f(s) = s - \sum_{i=1}^{n-1} \tilde{p}_i^{opt}(s) = 0.$$

**Krok 10:** Pro  $i = 1, 2, \dots, n-1$  vypočti rovnovážnou cenu  $i$ -tého článku jako

$$p_i^{eq} = \tilde{p}_i^{opt}(s).$$

**Krok 12:** Rovnovážnou cenu článku  $n$  vypočti jako

$$p_n^{eq} = p_n^{opt}(s).$$

Algoritmus je implementován ve funkci *equilpar* a pomocných funkcích, které *equilpar* využívá. Podívejme se nyní na některé kroky podrobněji.

**Kroky 1, 2 a 3:** Tyto kroky jsou víceméně totožné s kroky 1, 2 a 4 v sériovém řetězci. Z hlediska programu je zcela lhostejné, zda budeme jako jeden ze vstupních parametrů uvažovat součet všech cen článků z první úrovně paralelního řetězce nebo pouze jediného článku předchozího (jak tomu bylo u sériového řetězce). Kroky 1, 2 a 3 proto zajišťují po řadě funkce *profit*, *pricepoly* a *dempoly*, které byly popsány v souvislosti se sériovým řetězcem.

**Krok 5:** Zisk článků z první úrovně řetězce počítá funkce *profitpar*. Tato funkce je analogií k funkci *profit* s tím, že zohledňuje paralelní postavení článků z první úrovně. Funkce opět vrací zisk *i*-tého článku vynásobený konstantou  $-1$ .

**Krok 6 a 7:** Oba tyto kroky zajišťuje funkce *pricepolypar*. Pro funkce  $\tilde{p}_i^{opt}$  se stejně jako pro cenové reakční funkce článků v sériovém řetězci využívá aproximace ve tvaru odříznutého polynomu (realizovaného funkcí *polycut*). Nalezení parametrů tohoto polynomu pro *i*-tý článek probíhá následujícím způsobem:

- Je definován vektor součtu cen ostatních článků z první úrovně  $s$ , jehož složky  $s_j$  jsou ekvidistantně rozloženy na intervalu  $[0; p_{max}]$ . Délka vektoru (určující hustotu dělení intervalu) je zadána vstupním parametrem.
- Pro všechna  $s_j$  jsou nalezeny ceny  $p_j$ , které maximalizují ziskovou funkci *i*-tého článku.
- Podobně jako v případě sériového řetězce je zpřesněn odhad součtu cen ostatních článků, při kterém dochází ke skoku výstupní ceny *i*-tého článku na  $p_{max}$ . Označíme tento bod nespojitosti  $s_{cut-off}$ .
- Pro všechna  $s_j$  nižší, nežli je  $s_{cut-off}$ , je závislost mezi hodnotami  $s_j + p_j$  a  $p_j$  aproximována polynomem stupně zadaného v podobě vstupního parametru funkce *pricepolypar* (v roli nezávislé proměnné je zde  $s_j + p_j$ ).

Kromě parametrů odříznutého polynomu pro aproximaci  $\tilde{p}_i^{opt}$  vrací ještě funkce *pricepolypar* nejvyšší hodnotu součtu cen všech článků první úrovně, který ještě přináší kladné poptávané množství za podmínky, že se *i*-tý článek snaží maximalizovat individuální zisk při daných cenách ostatních článků (označme ho  $s_i^{max}$ ), a rovněž nejnižší součet cen, ke kterému může za stejné podmínky dojít (což nastává v případě, že všechny ostatní články z první úrovně volí nulovou výstupní cenu, označíme  $s_i^{min}$ ).

**Krok 9:** Definujeme-li interval  $I$  předpisem

$$I = \bigcap_{i=1}^{n-1} [s_i^{min}; s_i^{max}],$$

pak pro všechna  $s \in I$  a pro všechna  $i = 1, 2, \dots, n-1$  platí, že se nacházíme v té části  $\tilde{p}_i^{opt}$ , kterou jsme aproximovali polynomem (tj. v části před skokem), aproximace funkce  $f$  z kroku 9 bude mít tedy na  $I$  tvar polynomu. Hodnoty mimo tento interval nás přitom nezajímají, neboť buď vyústí v nulovou hodnotu poptávaného množství, nebo jsou příliš nízké, než aby mohly splňovat podmínky Nashovy rovnováhy. Pro nalezení kořenů reálného polynomu

existuje v prostředí Matlab vestavěná funkce *roots*. Z kořenů vybereme ten, který spadá do intervalu  $I$ .

## 3.2 Použité tvary funkcí pro poptávku a náklady

Jak již bylo řečeno, právě popsané algoritmy by měly obstát při použití nejrozumnějších spojitých funkcí v roli poptávky a nákladů. V tomto oddílu budou popsány ty tvary poptávkových a nákladových funkcí, které jsem se rozhodl použít ve výpočetních experimentech, které popisuje oddíl 3.3. Stejně funkce jsou rovněž připraveny pro použití v uživatelském rozhraní pro ovládání parametrů obou algoritmů. Ovládání uživatelského rozhraní je popsáno v příloze P.1.

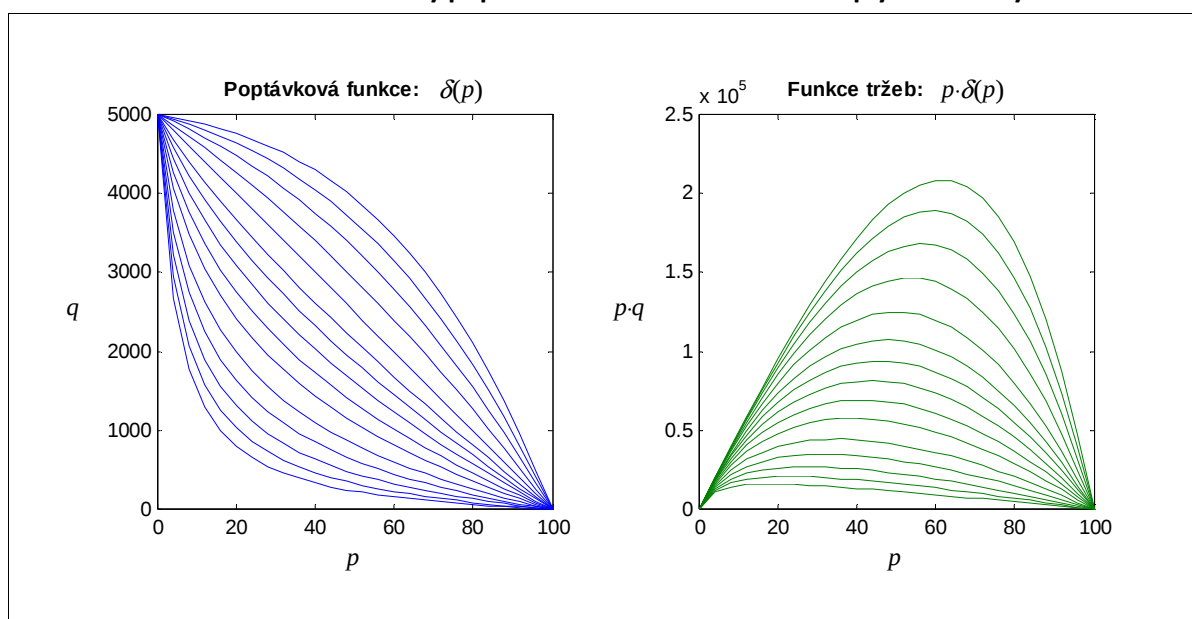
### 3.2.1 Poptávková funkce

Pro modelování poptávky jsem se rozhodl použít pouze jednoho funkčního předpisu, který může ovšem různým nastavením parametrů obsáhnout široké spektrum tvarů poptávkových funkcí. Vztah mezi cenou a poptávaným množstvím, který předepisuje tato poptávková funkce, má podobu

$$\delta(p) = \max \{a \cdot (p + b)^\alpha + d, 0\}, \quad (3.13)$$

kde  $a, b, d$  a  $\alpha$  jsou reálné koeficienty. Definičním oborem této funkce jsou v souladu s definicí (3.1) nezáporná reálná čísla. Podle názvů koeficientů je tato poptávková funkce v dalším textu a v uživatelském prostředí nazývána *abda*. V programové podobě představuje poptávkovou funkci *abda* funkce *dem\_abda*.

**Graf 3-4: Různé tvary poptávkové funkce *abda* a z nich plynoucí tržby**



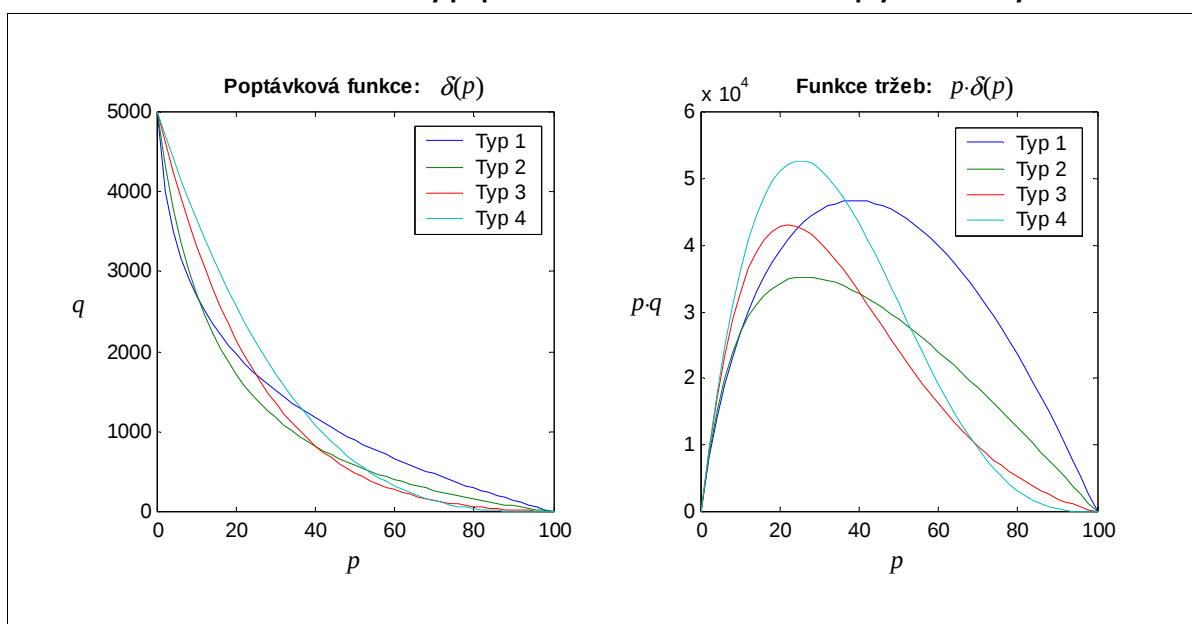
Graf 3-4 vlevo zachycuje tvary poptávkové funkce  $abda$  s různými parametry s tím, že všechny výsledné křivky prochází body  $(100;0)$  a  $(0;5000)$ . Pravá část grafu zobrazuje funkci tržeb, která z jednotlivých tvarů plyne. Funkce tržeb leccos napovídá o výsledném tvaru ziskové funkce posledního článku řetězce.

Z definice poptávkové funkce (3.13) je zřejmé, že ne všechna nastavení parametrů  $a, b, d$  a  $\alpha$  jsou smysluplná. Například pokud  $b = -100$  a  $\alpha = \frac{1}{2}$ , není funkce pro ceny z intervalu  $[0;100)$  vůbec definovaná. V dalším textu budeme zkoumat poptávkové funkce, které jsou klesající a konvexní (na intervalu, na kterém jsou nenulové), což je tvar, který zpravidla reálné poptávkové funkce vykazují (viz např. [8]). Následující tabulka shrnuje čtyři různé typy nastavení parametrů funkce  $abda$ , které vyústí v klesající a konvexní tvar:

**Tabulka 3-5: Typy nastavení parametrů pro konvexní klesající funkci  $abda$**

|              | $a$     | $b$     | $d$     | $\alpha$      |
|--------------|---------|---------|---------|---------------|
| <b>Typ 1</b> | kladné  | kladné  | záporné | záporné       |
| <b>Typ 2</b> | záporné | kladné  | kladné  | $\in (0;1)$   |
| <b>Typ 3</b> | kladné  | záporné | záporné | kladné, sudé  |
| <b>Typ 4</b> | záporné | záporné | kladné  | kladné, liché |

**Graf 3-6: Různé tvary poptávkové funkce  $abda$  a z nich plynoucí tržby**



Popis jednotlivých typů, jak jej podává tabulka 3-5, je pouze velmi hrubý a orientační. Aby měla poptávková funkce požadovaný tvar, nestačí zajistit znaménka koeficientů udávaná v tabulce, je samozřejmě třeba, aby i jejich absolutní hodnoty byly v jistém vztahu. Při nesprávném nastavení parametrů by se jinak mohlo snadno stát, že vyjde poptávková funkce např. konstantní nulová nebo v případě typu 4 třeba konkávní. Funkce odpovídající typu 3 je

zpočátku funkcí klesající, poté klesne na nulu, pro velké vstupní hodnoty začne ale její hodnota opět růst, a pro účely výpočtu je proto třeba omezit definiční obor na nerostoucí část funkce. Toho lze ovšem dosáhnout zvolením vhodného parametru  $p_{max}$ , který je zadáván jako jeden z parametrů při spuštění funkcí pro nalezení rovnovážných cen a který byl popsán v diskuzi o zisku centralizovaného řetězce na začátku kapitoly.

V grafu 3-6 vlevo jsou ukázány příklady poptávkových funkcí všech čtyř typů, které mají opět shodné průsečíky s oběma osami. Rozdíl v průběhu jednotlivých poptávkových funkcí je patrný také z výsledných funkcí tržeb v pravé části grafu. Z grafu je dobře vidět, že funkce *abda* je díky různým možnostem nastavení parametrů funkcí poměrně flexibilní.

### 3.2.2 Nákladové funkce

Pro zkoumání řetězců v nelineárním prostředí jsem připravil tři funkční tvary pro použití v roli nákladových funkcí: jsou jimi nákladová funkce lineární, realizovaná v prostředí Matlab funkcí *cost\_lin*, funkce polynomiální, kterou představuje funkce *cost\_poly*, a funkce mocninná, implementovaná ve funkci *cost\_pow*. Všechny zmíněné funkce uvažují definiční obor i obor hodnot v nezáporných reálných číslech, jak je popisuje definice (3.2). Veškeré záporné hodnoty, kterých by bylo možné dosáhnout zejména chybnou volbou vstupních parametrů, jsou proto při implementaci těchto funkcí nahrazeny nulou.

Lineární nákladová funkce popisuje náklady v tom smyslu, jak jsme s nimi pracovali v případě lineárního modelu. Jedná se tedy o výrobu s konstantními jednotkovými náklady, funkční předpis je

$$\gamma(q) = c \cdot q,$$

jediný parametr  $c \geq 0$  představuje jednotkové náklady.

Mocninná nákladová funkce modeluje náklady podle vztahu

$$\gamma(q) = c_1 \cdot q^{c_2}, \quad (3.14)$$

kde  $c_1$  a  $c_2$  jsou nezáporné reálné koeficienty. Volbou  $c_2$  z intervalu  $(0;1)$  získáme nákladovou funkci, která vykazuje úspory z rozsahu, volbou  $c_2 > 1$  dostaneme naopak funkci s rostoucími mezními (a průměrnými) náklady, která je typická pro krátké období.

Polynomiální nákladová funkce modeluje náklady ve formě polynomu libovolného stupně  $n$ . Funkce *cost\_poly*, která polynomiální nákladovou funkci realizuje v programové podobě, přebírá jako jediný vstupní parametr vektor  $\mathbf{c}$  o  $n+1$  složkách, které představují koeficienty nákladového polynomu podle předpisu

$$\gamma(q) = \sum_{i=0}^n c_{n-i} \cdot q^{n-i}, \quad (3.15)$$

koeficienty je samozřejmě pro všechny praktické aplikace žádoucí volit tak, aby nákladová funkce byla nezáporná a rostoucí. Polynomiální funkce je nejuniverzálnější z uvedených tří typů, neboť jí lze na určitém omezeném intervalu (tj. např. na intervalu relevantním z hlediska poptávky) s libovolnou přesností aproximovat jakoukoli spojitou nákladovou funkcí<sup>1</sup>.

Všechny tři funkční tvary mají společnou tu vlastnost, že se jedná vesměs o funkce spojitě a diferencovatelné. V případě nákladových funkcí bychom mohli v praxi najít nákladové funkce, které jsou nespojitě povahy (způsobené například nutností rozšířit pomocí investic stávající výrobní kapacity). Takové funkce bohužel není možné s navrženými algoritmy pro nalezení rovnovážných cen řetězce použít, neboť použité optimalizační algoritmy jsou bez výjimky navrženy pro extremalizaci spojitých funkcí; lze pro ně ovšem využít algoritmy s diskrétními cenami, kterými se zabývá oddíl 3.4.

### 3.3 Výpočetní experimenty

V druhé kapitole jsme z lineárního modelu odvodili řadu vlastností, které decentralizované dodavatelské řetězce splňují. Výpočetní experimenty se pokusí odhalit, nakolik se tyto vlastnosti týkají i řetězců fungujících v nelineárním prostředí. Zaměříme se přitom na následující z nich:

- S růstem počtu článků prudce klesá efektivnost řetězce, efektivnost řetězce sériového je nižší, než efektivnost paralelního řetězce se stejným počtem článků.
- Články decentralizovaného řetězce dosahují tím většího zisku, v čím nižší úrovni řetězce se nachází. Mezi ziskem článků stejné úrovně nejsou výrazné rozdíly.
- V decentralizovaném řetězci dochází k silným pozitivním externalitám nákladových úspor. Sníží-li jeden článek své výrobní náklady, vzroste individuální zisk všech článků řetězce.
- Zisk řetězce ani jednotlivých článků není závislý na rozložení výrobních procesů mezi články řetězce. Přesuneme-li výrobní proces mezi dvěma články, výsledné zisky se nezmění (pokud samozřejmě tento proces probíhá v obou člancích se stejnými náklady).

Všechny tyto vlastnosti budeme zkoumat ve třech základních scénářích: nejprve budeme uvažovat oddělený vliv nelinearity v poptávkové funkci, tj. zaměříme se na řetězce s nelineární poptávkovou funkcí a lineárními funkcemi nákladovými (pododdíl 3.3.1), poté se podíváme na izolovaný vliv nelinearity v nákladových funkcích, čili budeme zkoumat řetězce s nelineárními nákladovými funkcemi a lineární poptávkou (pododdíl 3.3.2), a nakonec provedeme několik pokusů s řetězci, které pracují v prostředí s nelineárními funkcemi jak v roli poptávky, tak v roli nákladů (pododdíl 3.3.3). Všechny uvedené výsledky se týkají pouze řetězců sériových a paralelních. Lze ovšem očekávat, že vlastnosti kombinovaných

---

<sup>1</sup> Viz např. [16], Weierstrassova věta o aproximaci polynomem.

řetězců budou (podobně jako v lineárním modelu) vymezeny oběma těmito hraničními případy a podle konkrétní struktury řetězce se budou přiklánět k jednomu z nich.

Výpočetní experimenty budou popsány formou příkladů, každý z nich se zaměřuje na určitý typ řetězce a jednu nebo více z uvedených vlastností dodavatelského řetězce. Vzhledem k tomu, že zkoumáme větší počet různých vlastností v řetězcích fungujících v rozličných podmínkách, je třeba v tomto oddílu rozebrat větší množství příkladů. V komentáři u jednotlivých příkladů jsou podrobně diskutovány dílčí výsledky, v pododdílu 3.3.4 jsou pak stručně shrnuty všechny podstatné závěry.

Zdrojové kódy, které spustí výpočet zde popsaných experimentů, zobrazí grafy a výsledky zapíše do paměti v podobě proměnných prostředí Matlab, jsou k dispozici na přiloženém CD v adresáři *Výpočetní experimenty*. Jednotlivé skripty jsou nazvány podle čísla příkladu a podle toho, jaký graf je po jejich spuštění zobrazen. Všechny uvedené experimenty byly provedeny s parametry týkajícími se přesnosti výpočtu *precision* a *degree* nastavenými na hodnoty 100 a 2.

### 3.3.1 Nelinearita v poptávkové funkci

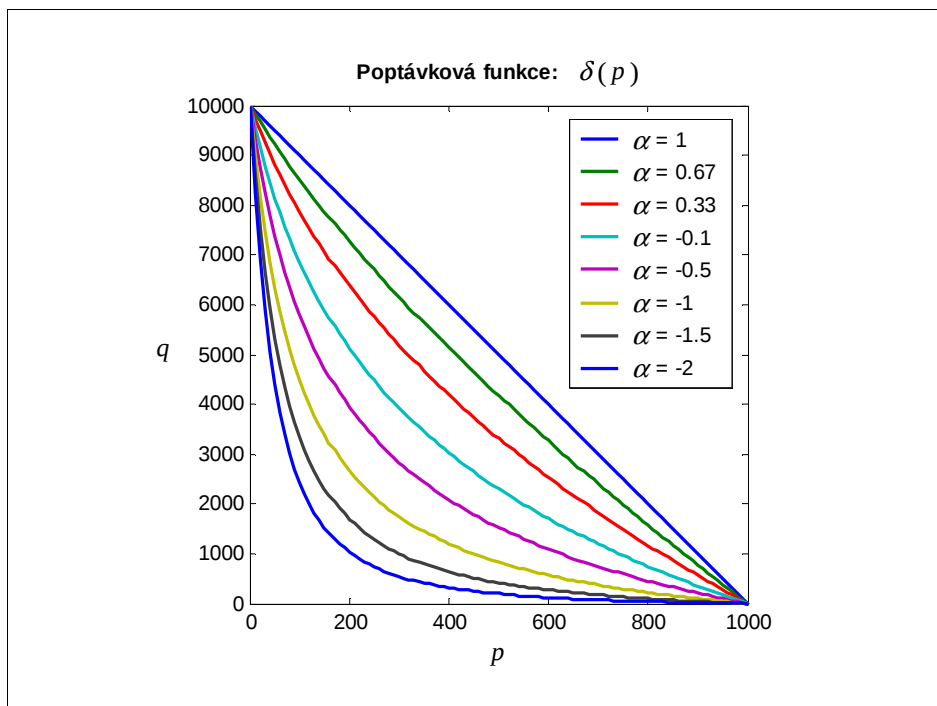
Na začátku tohoto pododdílu se budeme zabývat sériovými řetězci, ve druhé části navážeme řetězci paralelními. V prvním příkladu se zaměříme na první dvě vlastnosti z výše uvedeného výčtu, tj. na efektivnost a rozložení zisků v decentralizovaném řetězci, který prodává koncovou produkci na trhu s nelineární poptávkovou funkcí.

**PŘÍKLAD 7:** V praktických situacích většinou odezva spotřebitelů na změny ceny koncového produktu odpovídá ryze konvexnímu tvaru klesající funkce, spíše než tvaru lineárnímu. V tomto příkladě budeme uvažovat sériový dodavatelský řetězec se třemi články, který budeme vystavovat poptávkovým křivkám, které postupně přechází od lineárního tvaru až k funkci, jejíž směrnice se podél křivky výrazně mění.

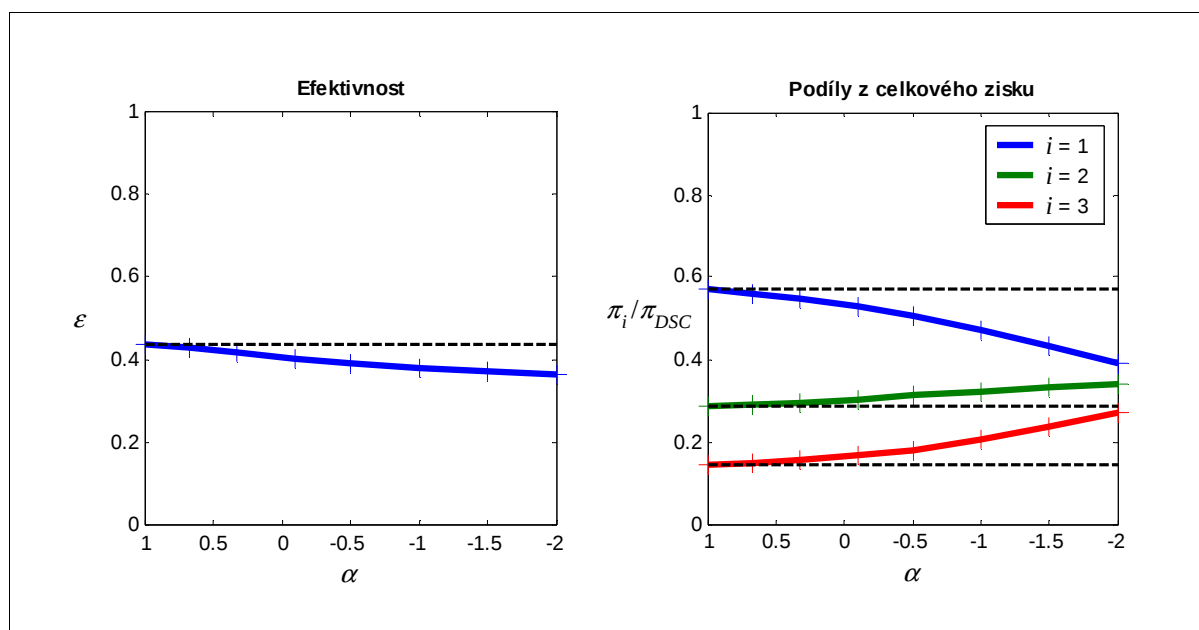
Podobu použitých poptávkových křivek znázorňuje graf 3-7. Všechny uvedené funkce mají za předpis funkci  $abda$  z oddílu 3.2, změna konvexity funkce je docílena pomocí parametru  $\alpha$ , který postupně klesá z 1 až na  $-2$ , jak je uvedeno v legendě grafu (jde tedy o poptávkové funkce  $abda$  typu 1 a 2, použijeme-li označení z pododdílu 3.2.1). Hodnota parametru  $b$  je 100 a parametry  $a$  a  $d$  jsou dopočítány tak, aby všechny křivky procházely průsečíky s osami  $p, q$  s hodnotami  $(0;10.000)$  a  $(1000;0)$ . Pro každou z těchto poptávkových funkcí byly spočítány rovnovážné hodnoty i zisk centralizovaného řetězce v případě, že všechny články vyrábí s jednotkovými náklady 50 peněžních jednotek. Z grafu 3-7 je patrné, že s klesající hodnotou parametru  $\alpha$  u zkoumaných poptávkových funkcí klesá pro každou cenu z intervalu  $[0;1000]$  poptávané množství. Je tedy zřejmé, že v absolutním měřítku se bude výše zisku decentralizovaného a centralizovaného zisku výrazně měnit. Předmětem našeho zájmu však nebude absolutní výše dosahovaného zisku, nýbrž poměr mezi ziskem decentralizovaného

a centralizovaného řetězce (tj. efektivnost řetězce) a rozložení zisku decentralizovaného řetězce mezi jednotlivé články, které budeme vyjadřovat pomocí podílu jednotlivých článků na zisku celého řetězce. Graf 3-8 ukazuje, jak se hodnoty těchto ukazatelů vyvíjely pro různé poptávkové funkce. Čárkovaně jsou v grafu naznačeny příslušné hodnoty z lineárního modelu.

Graf 3-7: Příklad 7 – poptávkové funkce



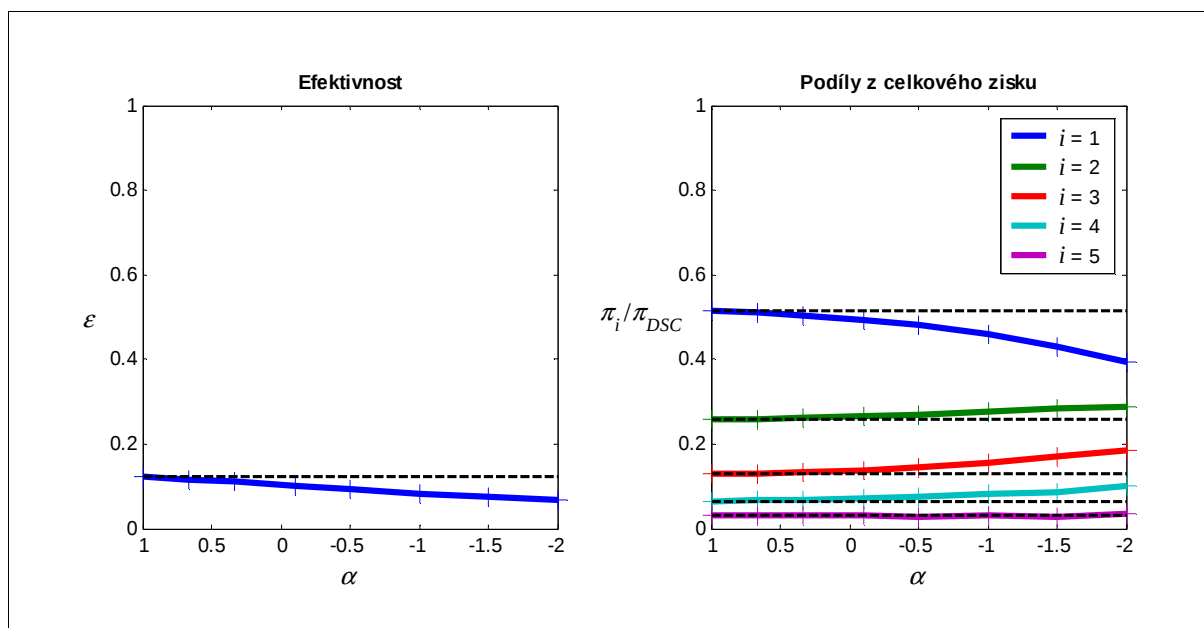
Graf 3-8: Příklad 7 – efektivnost řetězce a rozložení zisku



Není nikterak překvapující, že s tím, jak se tvar poptávkové funkce odchyluje od funkce lineární, mění se i některé ukazatele v dodavatelském řetězci. V pravé části grafu 3-8 vidíme, že s rostoucí konvexitou poptávkové funkce se podíly na individuálním zisku od původního poměru 4:2:1 pro lineární řetězec mění směrem k rovnocenným podílům. Stále však platí závěr z lineárního modelu, že největšího zisku dosahuje první článek a směrem ke konci řetězce zisk článků klesá. Z levé části grafu 3-8 můžeme vyčíst, že efektivnost řetězce se s tvarem poptávky mění daleko méně. S rostoucím parametrem  $\alpha$  klesaly hodnoty efektivnosti z 0,4375 pro lineární model postupně na 0,3639 pro model s hodnotou parametru  $\alpha = -2$ . V následujícím příkladu ověříme, že podobných výsledků bychom se mohli dočkat i v modelu delšího sériového řetězce.

**PŘÍKLAD 8:** V tomto příkladu budeme uvažovat sériový dodavatelský řetězec s pěti články, každý z nich pracuje opět s jednotkovými výrobními náklady 50 peněžních jednotek na kus koncové produkce. Použijeme-li stejné poptávkové funkce jako v předchozím příkladu, získáme hodnoty efektivnosti decentralizovaného řetězce a rozložení zisků v řetězci, které popisuje graf 3-9. Co se týče ziskových podílů, můžeme vyslovit podobný závěr jako v případě řetězce se třemi články. Jediným rozdílem je skutečnost, že tendence k vzájemnému přiblížení individuálních zisků se s prodlužujícím řetězcem zeslabuje. I pro krajní případ, kdy hodnota parametru  $\alpha = -2$ , jsou poměry zisků stále poměrně blízko k lineárnímu modelu. V případě efektivnosti je situace opačná – v řetězci s pěti články došlo k většímu odchýlení od hodnot lineárního modelu, než v řetězci se třemi články. Z původní hodnoty 0,1211 pro lineární model klesla efektivnost řetězce na 0,0672 pro  $\alpha = -2$ , tedy skoro na polovinu.

Graf 3-9: Příklad 8 – efektivnost řetězce a rozložení zisku



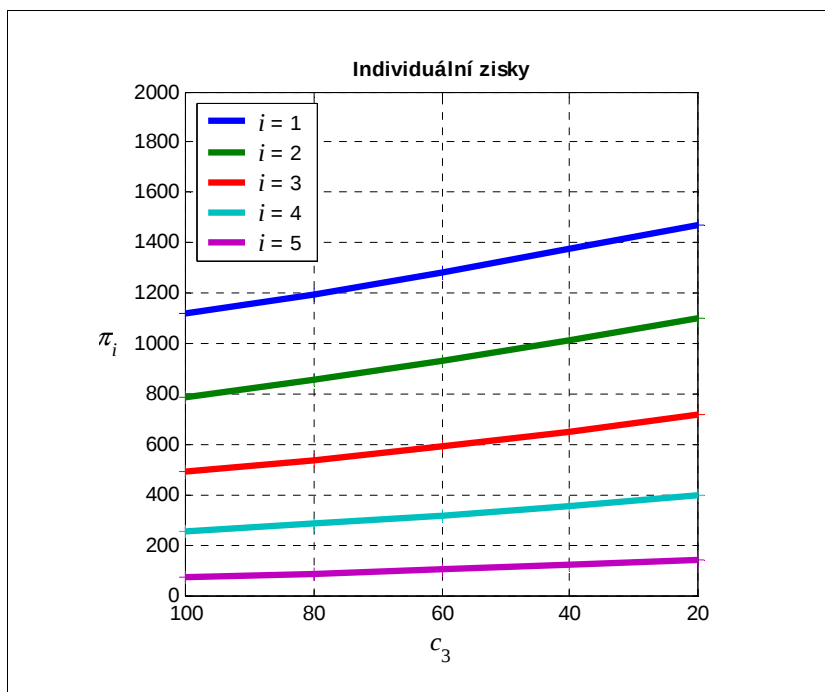
V dalším příkladu budeme zkoumat pozitivní externalitu nákladových úspor. V lineárním modelu nevedly změny v nákladech jednoho článku k žádným změnám v ziskových

podílech v řetězci, a docházelo tudíž k tomu, že pokud např. jeden z článků dosáhl pomocí zlepšení výrobní technologie nižších nákladů, zvedl se zisk nejen tohoto, nýbrž každého článku řetězce, ať už se jedná o článek předcházející nebo navazující.

**PŘÍKLAD 9:** Budeme uvažovat řetězec s pěti články, všechny články vyjma článku třetího budou operovat opět s jednotkovými náklady ve výši 50 peněžních jednotek. Koncová produkce čelí poptávkové funkci z příkladu 7 pro  $\alpha = -2$  (viz graf 3-7). Budeme zkoumat, jak se mění rovnovážné zisky článků a efektivnost decentralizovaného řetězce, pokud třetí článek snižuje své výrobní náklady postupně ze 100 peněžních jednotek na kus koncové produkce až na 20 peněžních jednotek (při výpočtu bylo použito 5 konkrétních hodnot s rozestupem 20 peněžních jednotek).

Vývoj rovnovážného zisku, který odráží různé úrovně nákladů ve třetím článku, zachycuje graf 3-10. Z grafu je dobře patrné, že snížení nákladů třetího článku vyvolává zvýšení zisku všech pěti článků řetězce, k pozitivním externalitám tedy dochází i v případě této nelineární poptávkové funkce. Přitom největších absolutních přírůstků zisku dosahuje nikoli článek třetí, který snižoval své náklady, ale článek první. To je opět podobný závěr jako v případě lineárního řetězce, kde přírůstky zisku musely mít stejnou proporcii jako původní ziskové podíly (viz příklad 2), neboť poměr individuálních zisků byl pro danou strukturu řetězce jednoznačně dán, nehledě na konkrétní parametry řetězce.

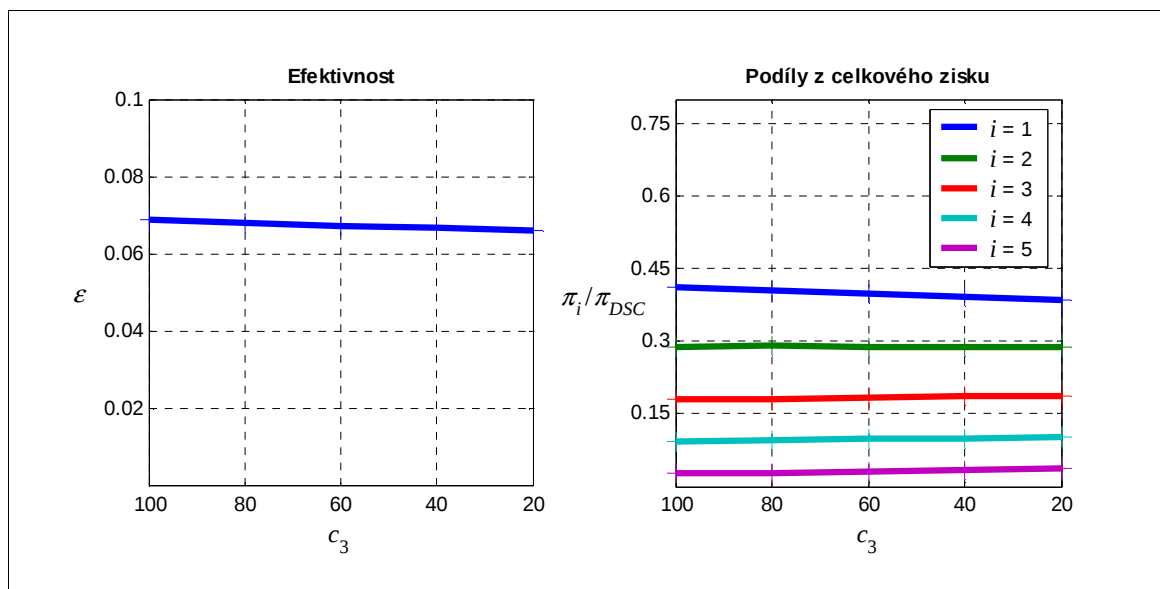
Graf 3-10: Příklad 9 – individuální zisky článků



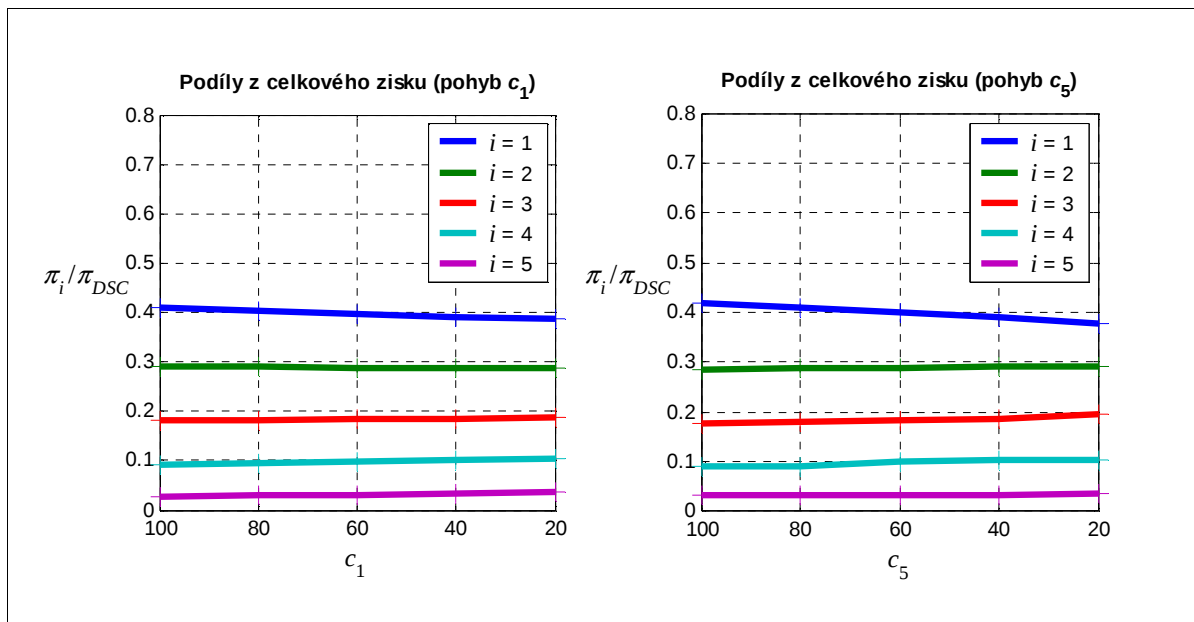
Graf 3-11 na následující straně ukazuje, jak se při změně nákladů třetího článku vyvíjela efektivnost a proporce zisků. Tvar křivky efektivnosti v levé části grafu napovídá, že změna jednotkových nákladů měla na efektivnost jen velmi malý vliv, což je opět v souladu se

závěry z lineárního modelu. Z vývoje podílů na celkovém zisku v pravé části grafu lze usoudit, že přírůstky zisku jsou skoro proporcionální. Můžeme si všimnout, že článkům ke konci řetězce roste zisk o trochu rychleji (ve smyslu proporce), nežli článkům na začátku řetězce.

Graf 3-11: Příklad 9 – efektivnost řetězce a rozložení zisku



Graf 3-12: Příklad 9 – podíly na zisku při nákladových úsporách článků 1 a 5



Na uvedených výsledcích se mnoho nezmění ani v případě, že ke snížení nákladů bude docházet nikoli v prostředku, nýbrž na krajích řetězce. V grafu 3-12 vidíme, jak se vyvíjí podíly na celkovém zisku, pokud by k obdobnému snížení nákladů docházelo namísto třetího článku u článků 1 a 5. Z grafu je patrné, že vývoj ziskových podílů je v obou případech velmi

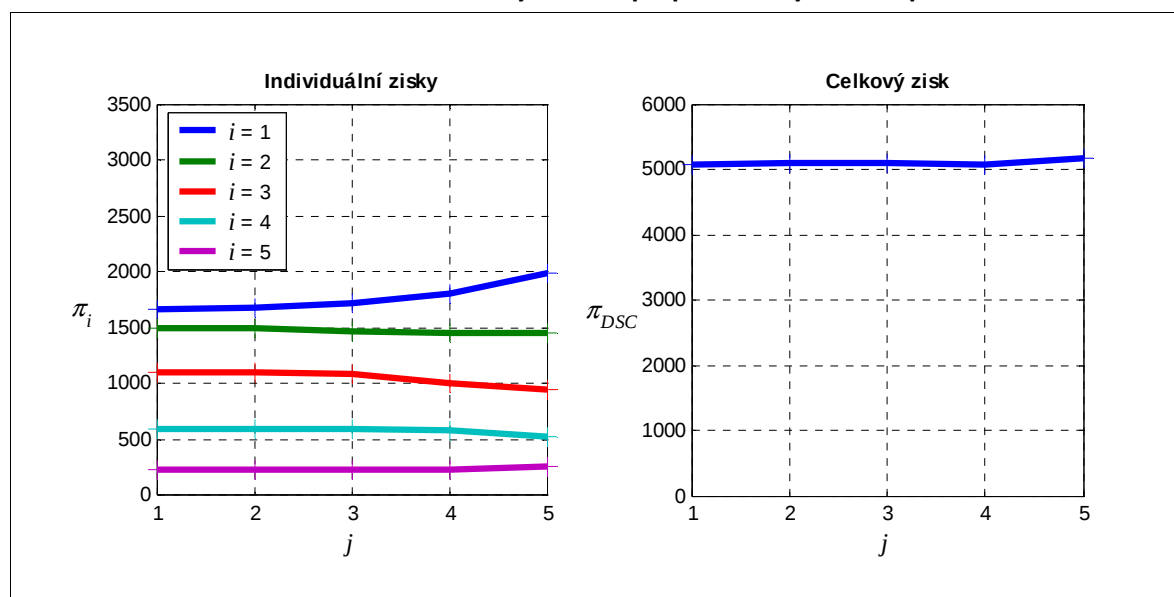
podobný předchozí situaci, kdy se měnily náklady článku třetího. Proporce zisků se při snižujících se nákladech (a tedy zvyšujícím se individuálním zisku) uvažovaného článku mění jen velmi nevýrazně, i zde proto dochází k pozitivním externalitám nákladových úspor v podobném rozsahu jako v lineárním modelu.

Jediným drobným rozdílem mezi všemi třemi případy, který lze z grafu vysledovat, je skutečnost, že s tím, jak se nákladové úspory přesouvají ke konci řetězce, dochází k většímu snižování ziskového podílu článků na začátku řetězce ve prospěch článků na jeho konci. Např. v grafu 3-12 vlevo vidíme, že při nákladových úsporách posledního (pátého) článku se zvyšoval ziskový podíl posledních čtyř článků, zatímco graf napravo ukazuje, že pokud dosáhl snížení nákladů článek první, zvýšil se ziskový podíl pouze u článků 3, 4 a 5, ziskový podíl článku 1 se snižoval ze všech případů nejrychleji.

Zdá se tedy, že rozložení nákladů v řetězci má na rozdíl od lineárního modelu nějaký vliv na rozložení zisku decentralizovaného řetězce. V lineárním modelu jsme ukázali, že pokud přesunujeme nějaký výrobní proces mezi různými články řetězce, zisk celého řetězce ani zisk jednotlivých článků se nezmění (samozřejmě za předpokladu, že všechny články provádí tento výrobní proces se stejnými jednotkovými náklady). Následující příklad ověří, nakolik lze takovou situaci očekávat i v případě řetězce v prostředí s nelineární poptávkou.

**PŘÍKLAD 10:** Uvažujme sériový řetězec s pěti články a poptávkovou funkcí shodnou jako v předchozím příkladu. Výrobní proces s jednotkovými náklady 150 peněžních jednotek budeme postupně přesouvat od prvního článku až ke článku pátému. Index článku, který proces právě provozuje, označíme jako  $j$ . Ostatní výrobní náklady všech článků řetězce budou zanedbatelné (tj. nulové).

**Graf 3-13: Příklad 10 – zisky řetězce při přesunu výrobního procesu**

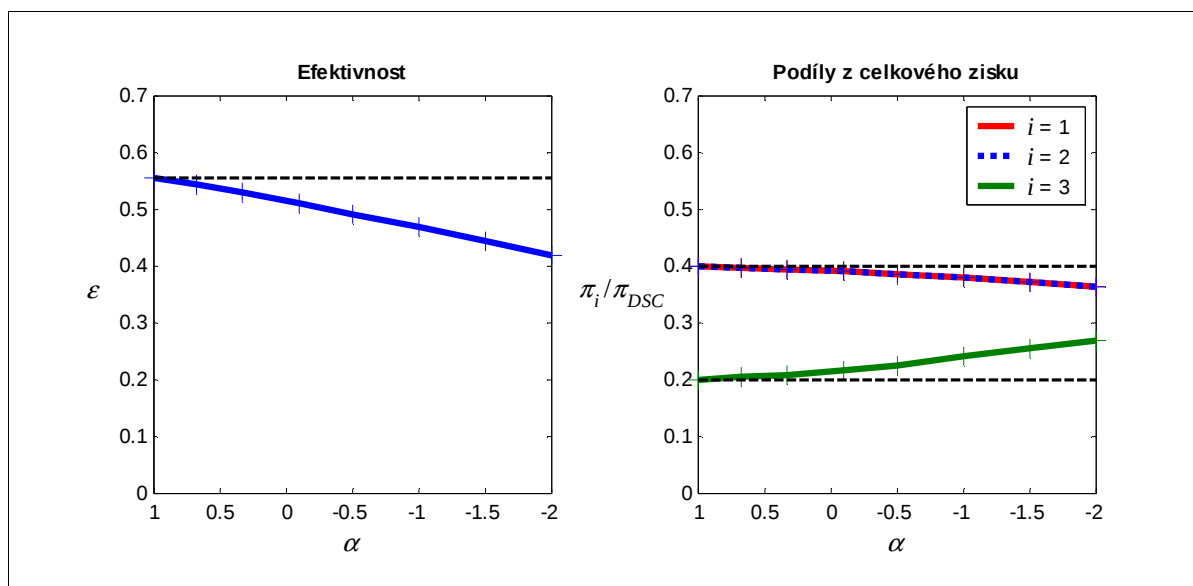


Graf 3-13 znázorňuje, jak se při přesunu procesu vyvíjí individuální zisky a celkový zisk decentralizovaného řetězce. V grafu napravo vidíme, že celkový zisk decentralizovaného řetězce se při přesunu procesu mění jen minimálně. Vzhledem k tomu, že na zisk centralizovaného řetězce nemá přesun procesu žádný vliv, zůstává i efektivnost řetězce pro všechny případy takřka shodná (pohybuje se mezi hodnotami 0,0661 a 0,0717). Větší vliv má přesun nákladů na rozložení zisku mezi jednotlivé články, jak ukazuje graf vlevo. Ani zde však nejsou rozdíly nikterak zásadní. Navíc nelze říci, že by článek, kterému je výrobní proces právě svěřen, dosahoval navzdory vyšším nákladům nižšího zisku než v situaci, kdy má daný proces na starosti článek jiný. Např. článek 5 dosahuje nejvyššího zisku, právě když výrobní proces obstarává on. Je však zřejmé, že poznatek z lineárního modelu ohledně nezávislosti rozložení zisků na rozložení nákladu je v modelu s nelineární poptávkou nutno brát s rezervou.

Obraťme nyní pozornost k řetězcům paralelním. Následující příklady budou analogické k příkladům 7 až 10 s tím, že strukturu řetězce změníme ze sériové na paralelní. Příklad 11 se zabývá dopadem změny tvaru poptávkové funkce (od lineárního po výrazně konvexní) na efektivnost paralelního řetězce a rozložení zisku mezi jednotlivé články.

**PŘÍKLAD 11:** Budeme uvažovat paralelní dodavatelský řetězec se třemi články, každý s jednotkovými náklady ve výši 50, 150 a 50 peněžních jednotek. Funkce poptávky po koncové produkci budou nabývat stejných tvarů, jako v příkladu 7, popisuje je tedy graf 3-7. Vypočteme-li rovnovážné hodnoty pro všechny případy poptávkové funkce, získáme výsledky, které popisuje graf 3-14.

**Graf 3-14: Příklad 11 – efektivnost řetězce a rozložení zisku**

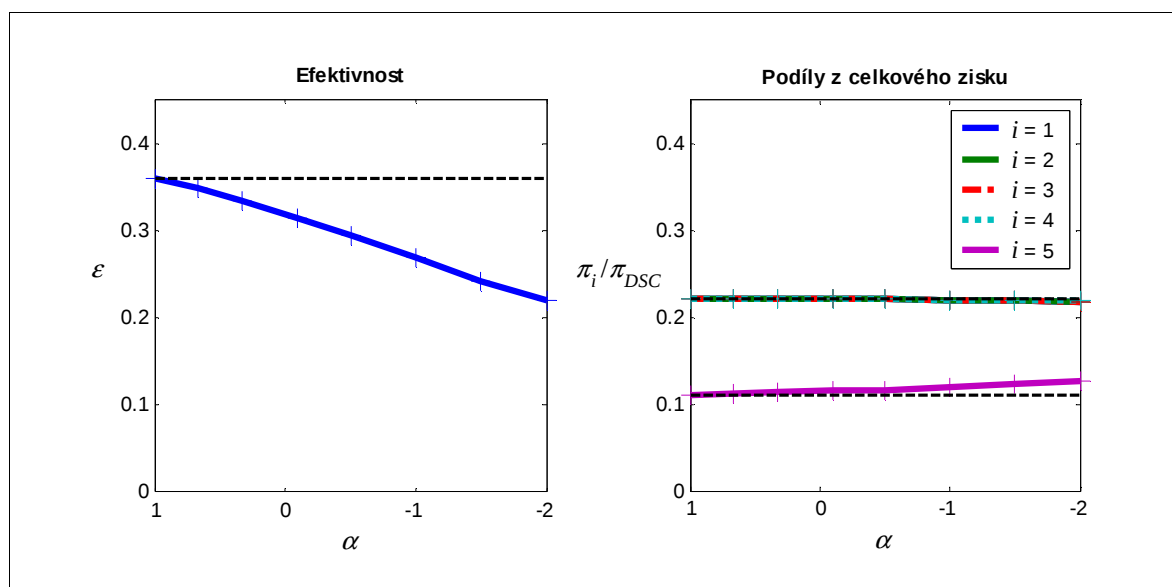


Změna tvaru poptávkové funkce je v grafu vyjádřena (stejně jako v příkladu 7) pomocí snižující se hodnoty parametru  $\alpha$ , který je vynesena na horizontální ose. Čárkovaně jsou

v grafu opět naznačeny hodnoty dosahované v lineárním modelu. Výsledky vykazují obecně stejné trendy jako v sériovém modelu: jak se tvar poptávkové funkce postupně odchyluje od tvaru lineárního, klesá efektivnost decentralizovaného řetězce a ziskové podíly článků se vzájemně přibližují. V grafu napravo si můžeme všimnout, že individuální články v první úrovni řetězce dosahovaly pro daný tvar poptávkové funkce vždy prakticky shodných ziskových podílů (v numerických výsledcích se liší ziskový podíl prvního a druhého článku pro všechny tvary poptávkové funkce až na čtvrtém desetinném místě). Články z první úrovně přitom operují s rozdílnými jednotkovými náklady (50 peněžních jednotek pro článek první a 150 pro článek 2). Zdá se tedy, že v případě lineárních nákladových funkcí je i pro nelineární poptávkovou funkci naplněn závěr z lineárního modelu, že články z téže úrovně dosahují nehledě na rozložení nákladů stejného zisku. Porovnáním grafů 3-14 a 3-8 dále zjistíme, že efektivnost klesla v paralelním modelu výraznějším způsobem (přibližně z 0,56 na 0,42), zatímco tendence k přiblížení ziskových podílů je naopak slabší. V následujícím příkladě ověříme, zda se ke stejným výsledkům dostaneme i při větším počtu článků.

**PŘÍKLAD 12:** Uvažujme paralelní dodavatelský řetězec s pěti články s jednotkovými náklady 30, 60, 90, 120 a 50 jednotek, který čelí stejným funkcím poptávky po koncovém produktu jako řetězec z příkladu 11. Efektivnost decentralizovaného řetězce a rozložení zisků při rovnovážných cenách zobrazuje graf 3-15. Z grafu je patrné, že i v paralelním řetězci se potvrzují závěry ze sériového případu, totiž že při větším počtu článků se ziskové podíly drží blíže k hodnotám z lineárního modelu, zatímco efektivnost řetězce klesá rychleji. Ziskové podíly se dokonce v tomto případě odchýlily od hodnot lineárního modelu jen velmi nevýrazně a opět se potvrzuje, že články na stejné úrovni řetězce dosahují (navzdory různým úrovním nákladů) přibližně stejného zisku.

Graf 3-15: Příklad 12 – efektivnost řetězce a rozložení zisku



Z porovnání hodnot efektivnosti sériových řetězců z příkladů 7 a 8 a jejich paralelních protějšků z příkladů 11 a 12 můžeme usoudit, že paralelní struktura řetězce je i v případě nelineární poptávkové funkce vždy efektivnější nežli struktura sériová. Tabulka 3-16 uvádí souhrn hodnot efektivnosti, které dosahovaly řetězce se třemi články v příkladech 7 a 11 a řetězce s pěti články v příkladech 8 a 12, pro všechny uvažované tvary poptávkových funkcí, jejichž parametr  $\alpha$  je uveden v záhlaví sloupců. Nejen, že je hodnota efektivnosti paralelního řetězce vždy vyšší, než efektivnost řetězce sériového se stejnou poptávkovou funkcí, ale navíc poměr těchto efektivností se s tvarem poptávkové funkce příliš nemění. Poměr efektivnosti sériové a paralelní struktury v lineárním modelu, který se nachází v prvním sloupci tabulky a který můžeme obecně vypočítat pro daný počet článků  $n$  ze vzorců (2.11) a (2.22) jako

$$\frac{2n-1}{n^2} \cdot \frac{2^{2n-2}}{2^n-1}, \quad (3.16)$$

lze tedy přibližně uplatnit i na řetězce s nelineární poptávkovou funkcí.

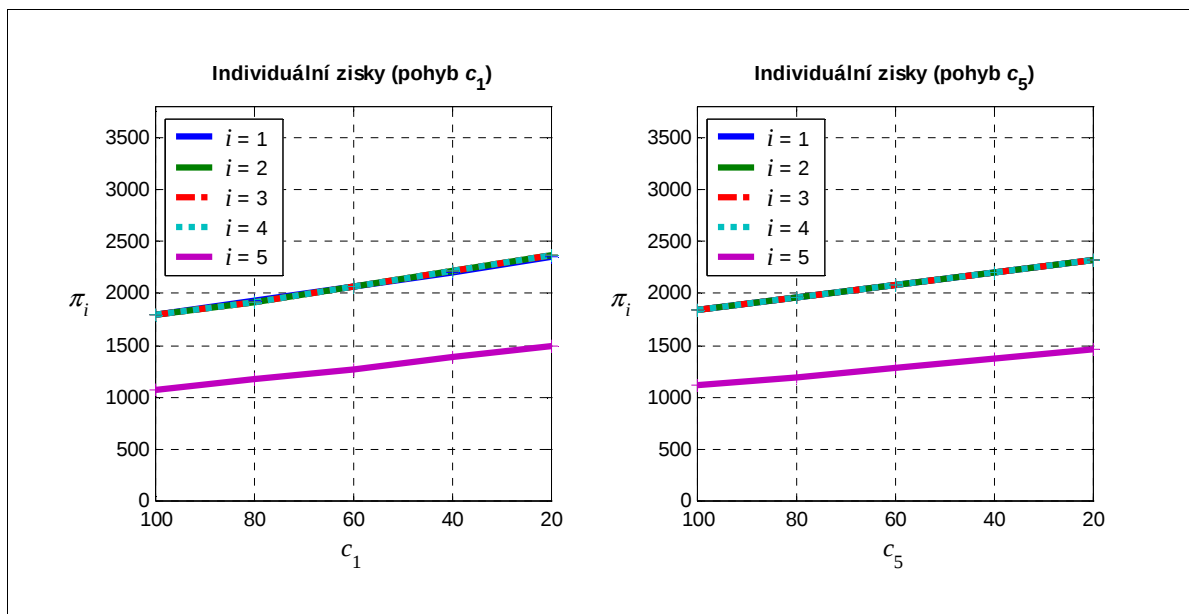
**Tabulka 3-16: Porovnání efektivnosti sériových a paralelních řetězců**

|                       | $\alpha$ |      |      |      |      |      |      |      |
|-----------------------|----------|------|------|------|------|------|------|------|
|                       | 1        | 0,67 | 0,33 | -0,1 | -0,5 | -1   | -1,5 | -2   |
| <b>Příklad 7</b>      | 0,44     | 0,43 | 0,42 | 0,40 | 0,39 | 0,38 | 0,37 | 0,36 |
| <b>Příklad 11</b>     | 0,56     | 0,54 | 0,53 | 0,51 | 0,49 | 0,47 | 0,44 | 0,42 |
| <b>Př. 11 / Př. 7</b> | 1,27     | 1,27 | 1,27 | 1,27 | 1,26 | 1,24 | 1,20 | 1,15 |
| <b>Příklad 8</b>      | 0,12     | 0,12 | 0,11 | 0,10 | 0,09 | 0,08 | 0,07 | 0,07 |
| <b>Příklad 12</b>     | 0,36     | 0,35 | 0,33 | 0,31 | 0,29 | 0,27 | 0,24 | 0,22 |
| <b>Př. 12 / Př. 8</b> | 2,97     | 3,00 | 3,03 | 3,09 | 3,15 | 3,21 | 3,27 | 3,26 |

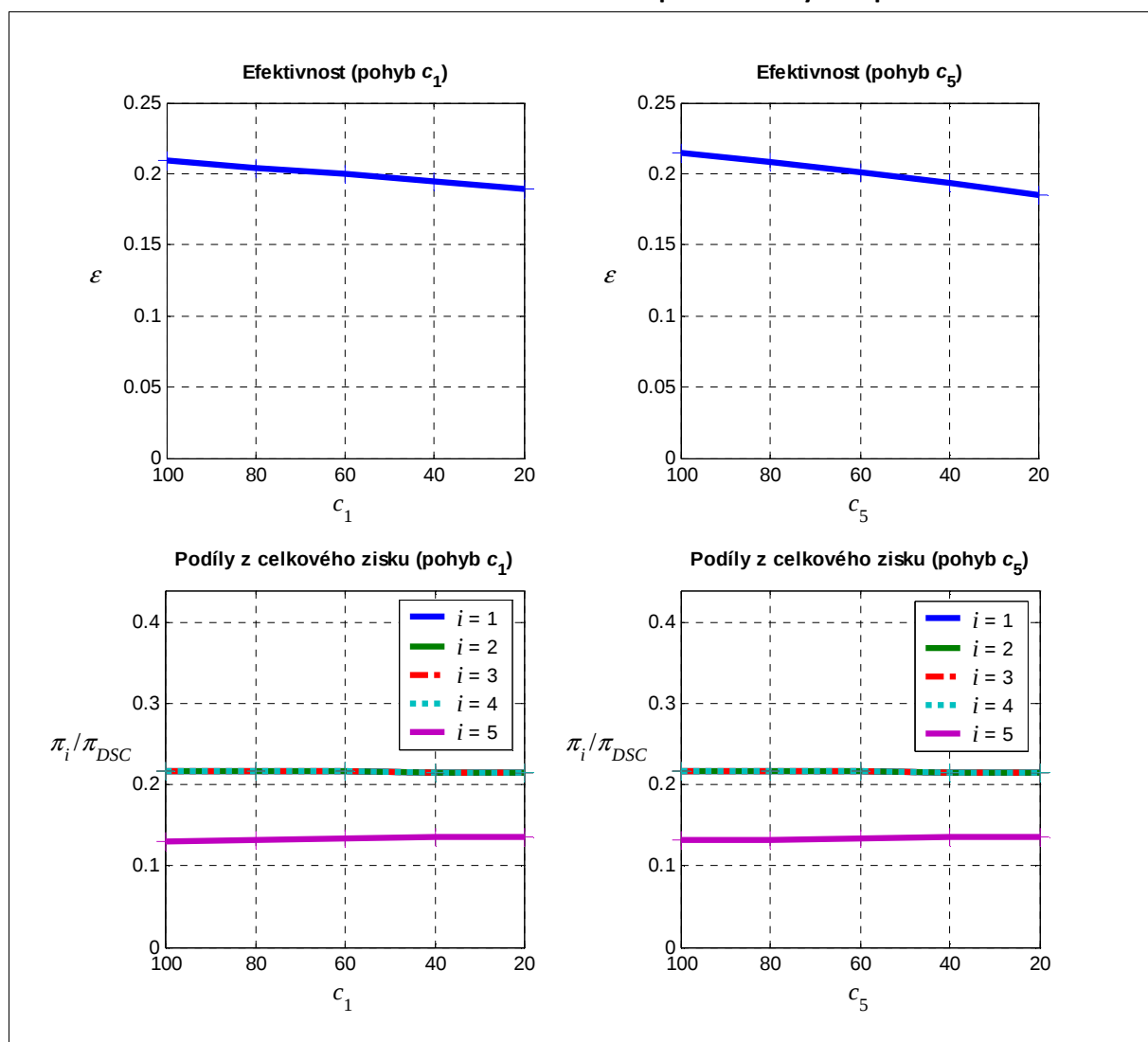
V dalších příkladech se budeme zabývat pozitivními externalitami nákladových úspor a vlivem rozložení nákladů na zisk v paralelních řetězcích, podobně jako v příkladech 9 a 10 u sériových řetězců.

**PŘÍKLAD 13:** Budeme uvažovat paralelní dodavatelský řetězec s pěti články a poptávkou po koncové produkci jako v příkladu 9 (tj.  $\alpha = -2$ ). Všechny články vyrábí s jednotkovými náklady 50 peněžních jednotek na kus koncové produkce, vyjma jednoho, za jehož jednotkové náklady budeme postupně dosazovat hodnoty 100, 80, 60, 40 a 20. Zaměříme se tedy na to, jak se změní situace, když jeden článek dosáhne nákladových úspor. Budeme přitom zkoumat dva případy – změnu nákladů jednoho z článků první úrovně řetězce (články 1 až 4) a změnu nákladů článku koncového. Graf 3-17 znázorňuje vývoj individuálních zisků v případě, pokud se mění náklady prvního, resp. pátého (koncového) článku. V obou případech můžeme potvrdit existenci pozitivních externalit nákladových úspor. Z grafu je navíc patrné, že individuální zisky se v obou případech liší jen minimálně. Zdá se, že v případě paralelního řetězce nezáleží tolik na pozici článku, který dosahuje nákladových úspor, situace je tedy v tomto ohledu lineárnímu model blíží, nežli v případě sériového řetězce.

Graf 3-17: Příklad 13 – efektivnost řetězců a rozložení zisků



Graf 3-18: Příklad 13 – efektivnost a rozložení zisku při nákladových úsporách článku 1 a 5

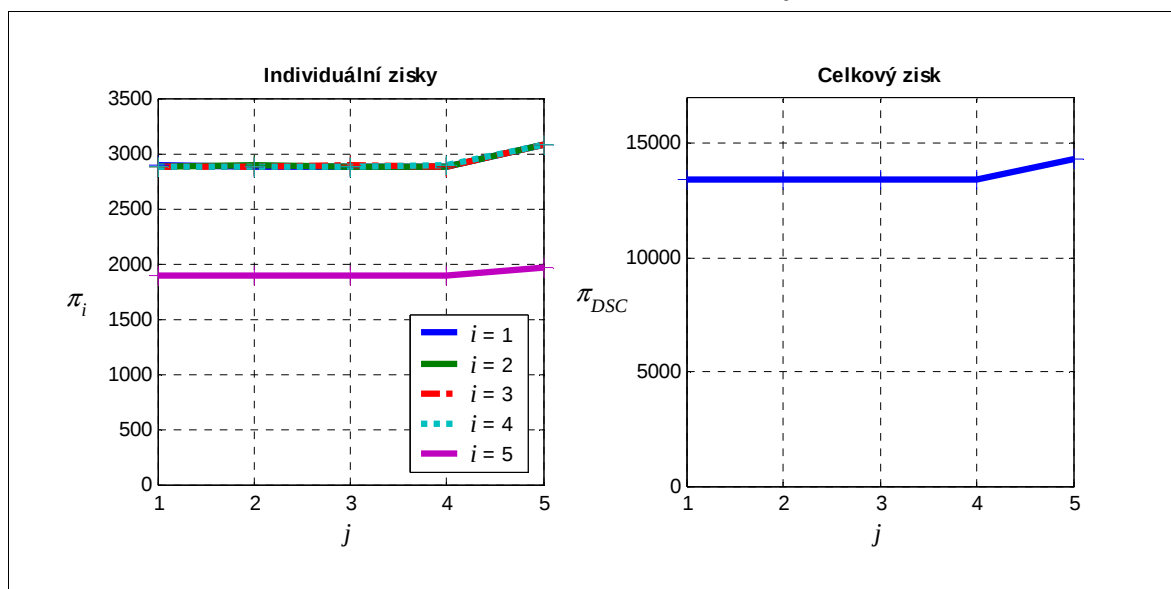


Takový závěr potvrzuje i graf 3-18. V grafu dole vidíme, že ziskové podíly článků paralelního řetězce se vzájemně přiblížily méně, než v obdobném řetězci sériovém (viz graf 3-11 a 3-12). I v paralelním řetězci ovšem došlo s úsporou nákladů k poklesu efektivnosti, která se tak od lineárního modelu vzdaluje.

Poslední příklad tohoto pododdílu se zabývá vlivem přesunu výrobních procesů na individuální zisky a celkový zisk paralelního dodavatelského řetězce.

**PŘÍKLAD 14:** Podobně jako v příkladu 10 budeme uvažovat řetězec s pěti články, mezi kterými přesunujeme výrobní proces s jednotkovými náklady ve výši 150 peněžních jednotek, ostatní náklady všech článků jsou nulové. Funkce poptávky po koncové produkci je opět shodná jako v příkladu 9. Průběh rozložení zisků i celkového zisku při přesunu výrobního procesu zachycuje graf 3-19.

Graf 3-19: Příklad 14 – individuální zisky článků



Při přesunu procesu mezi články první úrovně řetězce nedochází k takřka žádné změně ve výsledných ziscích. Z hlediska celkového zisku není takový výsledek nijak překvapivý, neboť články v první úrovni řetězce jsou z pohledu struktury řetězce ve zcela identické pozici, a je tedy zcela lhostejné, který z nich daný proces obstará. Podstatnější je, že nedochází k výkyvům individuálního zisku – článek, na který výrobní proces připadne, není oproti ostatním ani ve výhodě ani v nevýhodě, všechny ve všech případech dosahují víceméně stejného zisku. Jediné pozorovatelné změny se dočkáme, pokud výrobní proces přemístíme na článek koncový. Podobně jako v sériovém řetězci (příklad 10) dojde ke zvýšení celkového zisku (a efektivnosti řetězce), podíl zisku koncového článku se však v tomto případě sníží.

Nelineární poptávkovou funkci nyní na chvíli opustíme a ukážeme, jaký dopad má přítomnost nelinearity v nákladových funkcích.

### 3.3.2 Nelinearita v nákladových funkcích

Jak bylo řečeno v úvodní části oddílu 3.3, budeme v celém tomto pododdílu uvažovat lineární funkci poptávky po koncové produkci dodavatelského řetězce. Ve všech následujících příkladech bude mít poptávková funkce tvar

$$\delta(p) = 5000 - 10 p_n, \quad (3.17)$$

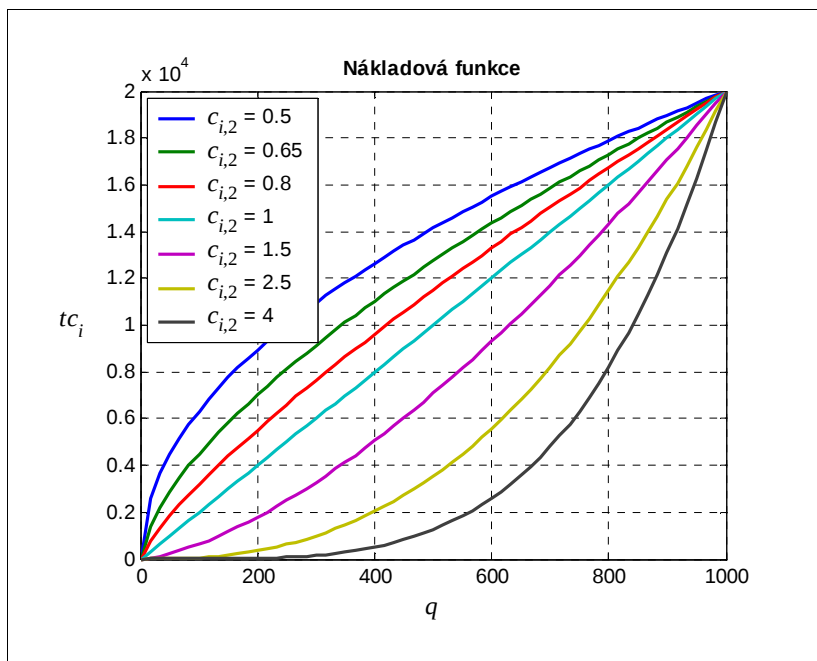
kde  $p_n$  je výstupní cena posledního článku řetězce.

Při použití nelineárních nákladových funkcí budeme rozlišovat především případy, kdy nákladová funkce vykazuje úspory z rozsahu (s růstem objemu výroby rostou průměrné náklady, což vyústí v konkávní tvar nákladové křivky) a případy, kdy průměrné náklady jsou s objemem výroby rostoucí (situace označovaná někdy jako záporné úspory z rozsahu, křivka nákladů je konvexní). Stejně jako v minulém pododdílu budeme nejprve uvažovat modely sériových řetězců, a posléze porovnáme dosažené výsledky s analogickými experimenty pro paralelní řetězce. Následující příklad zkoumá vliv úspor z rozsahu na efektivnost a rozložení zisku v sériovém řetězci.

**PŘÍKLAD 15:** Budeme uvažovat sériový dodavatelský řetězec se čtyřmi články a poptávkovou funkcí (3.17). Všechny články řetězce mají shodnou nákladovou funkci, která má podobu mocninné funkce (3.14). Budeme zkoumat různá nastavení obou parametrů nákladové funkce tak, aby se měnily dosahované úspory z rozsahu. O míře úspor z rozsahu rozhoduje v této funkci parametr  $c_2$ . V následujícím textu je tento parametr označován jako  $c_{i,2}$ , kde  $i$  představuje index článku. V celém příkladu mají všechny články vždy stejné náklady a index  $i$  se proto může zdát nadbytečný, dvouindexové značení však navazuje na značení použité dříve v tomto oddílu a jeho cílem je předejít záměně se značením, které bylo použito pro jednotkové náklady článku 2. Zkoumané hodnoty parametru budou postupně 0,5; 0,65; 0,8; 1; 1,5; 2,5 a 4. Hodnoty zbývajících parametrů jsou dopočítány tak, aby nákladová funkce procházela bodem (1000;20.000).

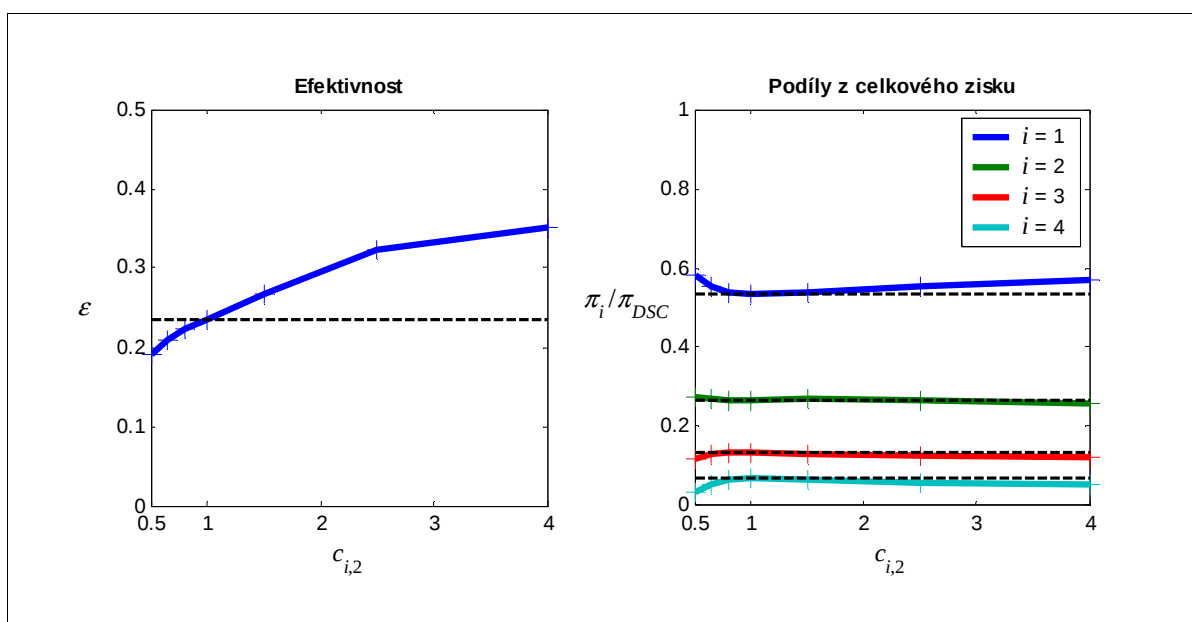
Výsledné tvary poptávkových funkcí na intervalu  $[0;1000]$  znázorňuje graf 3-20. Pro hodnotu parametru  $c_{i,2} < 1$  dosahuje nákladová funkce úspor z rozsahu, pro  $c_{i,2} = 1$  se jedná o lineární funkci a pro  $c_{i,2} > 1$  vykazuje funkce záporné úspory z rozsahu. Z grafu je zřejmé, že pro poptávané množství z intervalu  $(0;1000)$ , kam budou vesměs výsledné hodnoty pro decentralizovaný řetězec spadat, jsou výsledné náklady pro vyšší hodnoty parametru  $c_{i,2}$  nižší, což ovlivní výslednou výši zisku. Nás však bude, stejně jako u předchozích experimentů, spíše než absolutní velikost zisku zajímat proporce jeho rozložení mezi jednotlivé články a jeho poměr k zisku centralizovaného řetězce. Vývoj všech těchto hodnot se změnou parametru  $c_{i,2}$  je zachycen v grafu 3-21. V grafu jsou opět čárkovaně vyneseny hodnoty pro lineární model (kterých je zde dosaženo pro  $c_{i,2}$ ).

Graf 3-20: Příklad 15 – nákladové funkce článků



V grafu napravo vidíme, že velikost zisku jednotlivých článků zachovává při různých tvarech pořadí, které jsme určili pro lineární model. Poměr dosahovaných zisků se poněkud mění – jak pro kladné, tak pro záporné úspory z rozsahu se proporcionální rozdíly mezi zisky jednotlivých článků prohlubují (na rozdíl od výsledků, které jsme získali pro konvexní poptávkové funkce, kde jsme sledovali tendenci k proporcionálnímu přiblížení individuálních zisků).

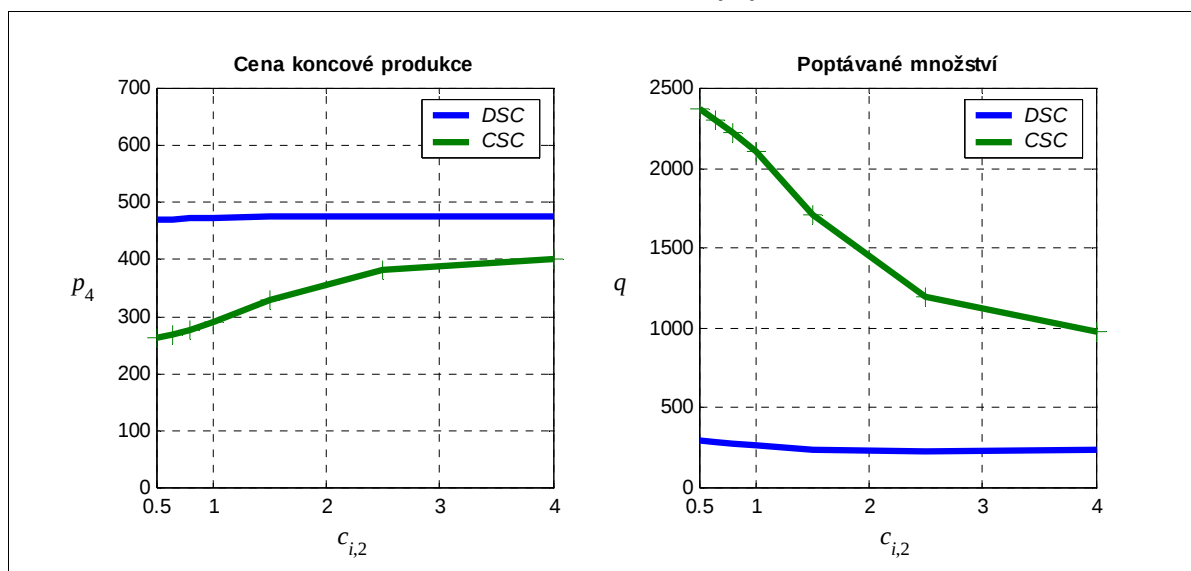
Graf 3-21: Příklad 15 – efektivnost a rozložení zisku



Graf vlevo ukazuje, že s kladnými úsporami z rozsahu dochází ke snižování efektivnosti řetězce a se zápornými úsporami naopak k jejímu zvýšení. Tento jev má poměrně přímočaré

logické odůvodnění. V jádru nízké efektivity decentralizovaných řetězců stojí fakt, že „kompetitivním“ chováním článků v decentralizovaném řetězci je zisková marže a výsledná cena koncové produkce tlačena nad úroveň ceny optimalizující celkový zisk. Vykazují-li nákladové funkce úspory z rozsahu, dochází s růstem ziskové marže ke snížení poptávaného množství a následnému zvýšení průměrných nákladů. Tvar nákladové funkce tedy umocňuje rozdíl v zisku centralizovaného a decentralizovaného řetězce. Naopak v případě záporných úspor z rozsahu nebude volit centralizovaný řetězec cenu koncové produkce o mnoho nižší, než řetězec decentralizovaný, neboť při snižování ceny dochází k růstu průměrných nákladů, které tak kompenzují přírůstek tržeb. Konvexní tvar nákladové funkce tedy stírá rozdíly mezi výsledky obou forem řetězce. Přiblížení cen koncové produkce v centralizovaném a decentralizovaném řetězci jakožto důsledek změny úspor v rozsahu a jeho dopad na poptávané množství dokumentuje graf 3-22 (značení *DSC* a *CSC* představují decentralizovaný a centralizovaný řetězec podobně jako v dříve uvedených vzorcích).

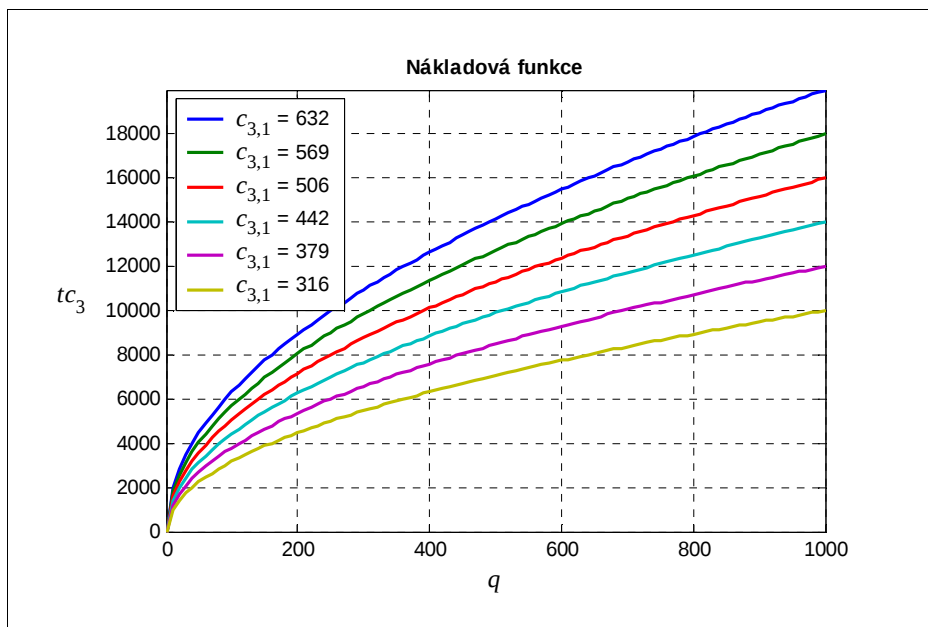
Graf 3-22: Příklad 15 – koncová cena a poptávané množství



V následujícím příkladě se budeme zabývat externalitami nákladových úspor v řetězci, jehož články vyrábí s nákladovými funkcemi vykazujícími úspory z rozsahu.

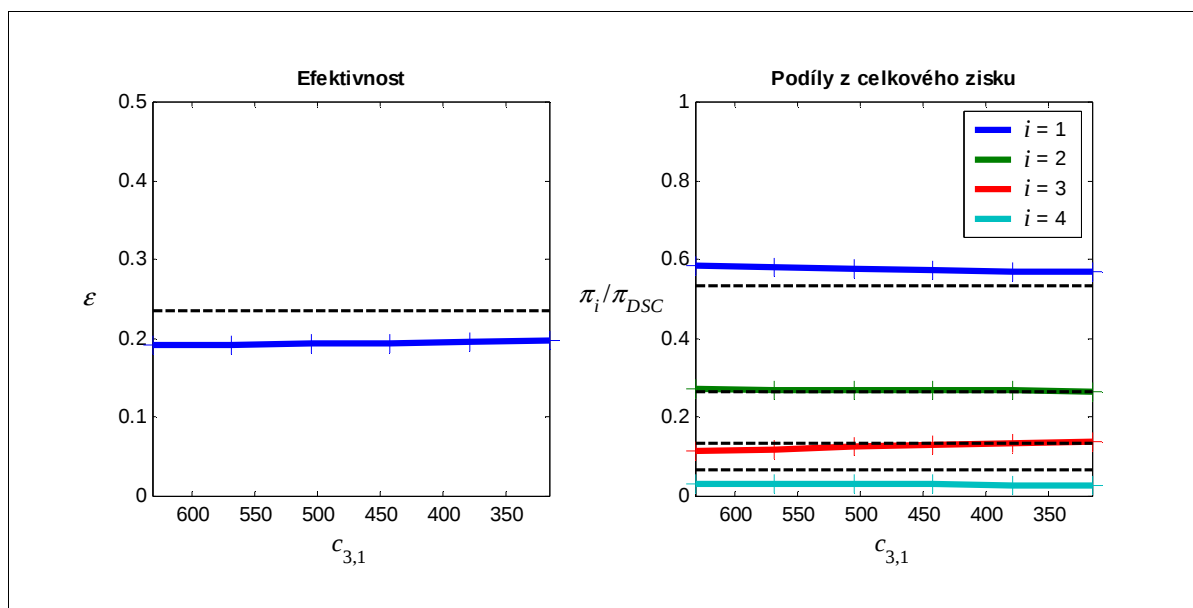
**PŘÍKLAD 16:** Uvažujme opět sériový dodavatelský řetězec s poptávkou po koncové produkci ve tvaru (3.17). Nákladové funkce článků 1, 2 a 4 jsou shodné s nákladovou funkcí z minulého příkladu pro  $c_{i,2} = 0,5$  (v grafu 3.23 znázorněna modře). Nákladová funkce článku 3 má nejprve stejnou podobu, následně však začne třetí článek realizovat nákladové úspory a jeho nákladová funkce se začne posouvat směrem dolů tak, jak to ukazuje graf 3-23. Změny nákladové funkce je dosaženo snižováním prvního parametru nákladové funkce (3.14), tj. zde parametru  $c_{3,1}$ , z původní hodnoty přibližně 632 až na polovičních 316. Druhý parametr nákladové funkce zůstává na úrovni 0,5. Celkové náklady tedy pro všechna vyráběná množství klesají z původní hodnoty postupně až na polovinu.

Graf 3-23: Příklad 16 – nákladová funkce článku 3



Při poklesu nákladů třetího článku dochází okamžitě ke zvýšení zisku centralizovaného i decentralizovaného řetězce. Jak ukazuje graf 3-24, probíhá nárůst těchto zisků takřka se stejnou proporcí, neboť hodnota efektivity se pro všechny hodnoty parametru  $c_{3,1}$  mění jen velmi nevýrazně. V grafu napravo vidíme, že je navíc přibližně zachován i poměr individuálních zisků, dochází tedy k pozitivním nákladovým externalitám, a to dokonce téměř v té podobě, jakou jsme odvodili pro lineární model (tj. proporcionálně).

Graf 3-24: Příklad 16 – efektivnost a rozložení zisku



Z grafu vlevo je patrné, že efektivnost se s klesajícími náklady trochu přibližuje k hodnotě lineárního modelu. To lze vysvětlit podobným způsobem, jako v předchozím příkladě. Při

změně parametru  $c_{3,1}$  se nákladová křivka stává plošší, úspory z rozsahu již nejsou tak intenzivní a dochází tedy k podobnému jevu, jako při přechodu k nákladovým funkcím s nižšími úsporami z rozsahu v předchozím příkladě. Pokud bychom vycházeli z nákladových křivek vykazujících záporné úspory z rozsahu, výsledky by byly obdobné – s klesajícím multiplikativním koeficientem v nákladové křivce by se výsledky přibližovaly lineárnímu modelu (v daném případě by šlo ovšem snižování efektivnosti, neboť ta je v modelech se zápornými úsporami z rozsahu vyšší, než v lineárním modelu).

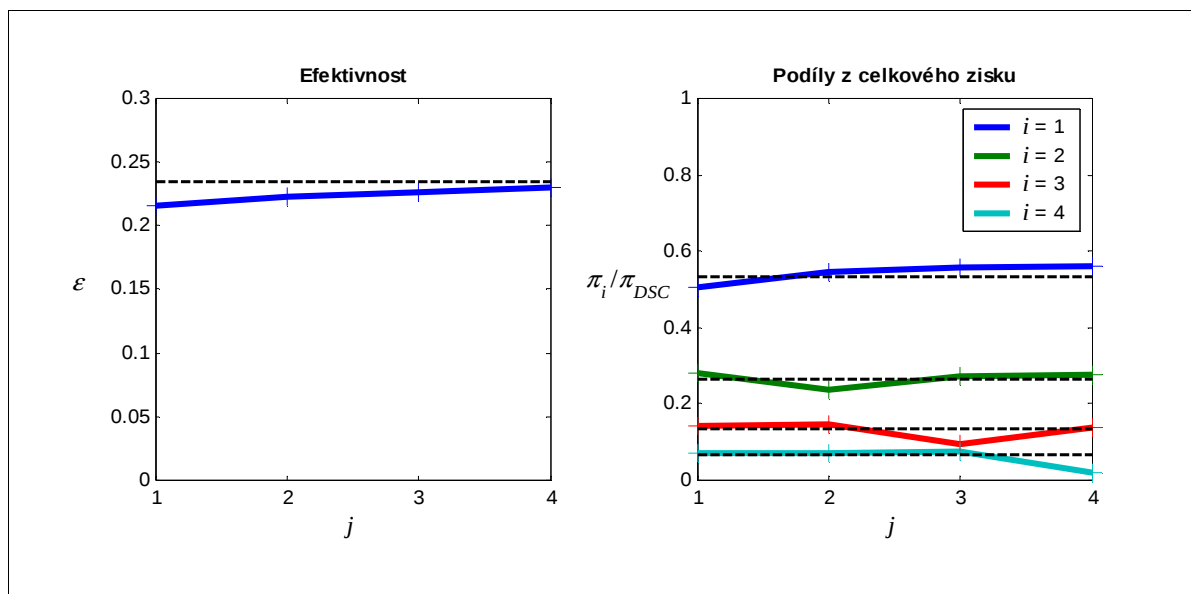
V příkladu 17 budeme testovat vliv rozložení nákladů v řetězci na celkový zisk a jeho rozložení mezi jednotlivé články. Budeme při tom postupovat podobně jako v příkladu 10, tj. budeme přesouvat výrobní proces mezi články řetězce a sledovat dopad na zkoumané ukazatele. Přesouvání výrobní proces ovšem bude v tomto případě prováděn s nelineární nákladovou funkcí (která bude pro všechny články shodná).

**PŘÍKLAD 17:** Uvažujme sériový dodavatelský řetězec se čtyřmi články a poptávkou po koncové produkci ve tvaru (3.17). Mezi články budeme přesouvat výrobní proces, jehož náklady vykazují úspory z rozsahu a jsou dány předpisem

$$tc = 800 \cdot \sqrt{q} \quad (3.18)$$

nehledě na to, který článek proces obstarává. Ostatní náklady článků zanedbáme (tj. položíme rovny nule). Graf 3-25 ukazuje rozložení zisku a efektivnost v jednotlivých případech, index  $j$  na horizontální ose představuje číslo článku, který právě proces zajišťuje.

**Graf 3-25: Příklad 17 – efektivnost a rozložení zisku (úspory z rozsahu)**

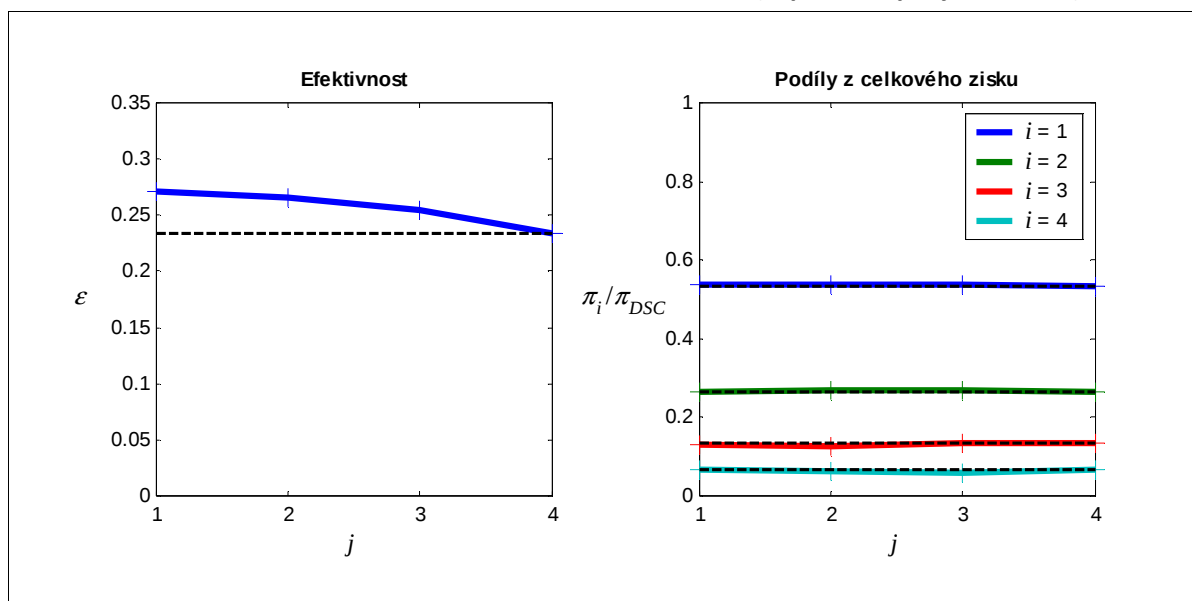


Vzhledem k tomu, že uvažovaný proces vykazuje úspory z rozsahu, je výsledná efektivnost vždy nižší, než v lineárním modelu. V grafu vlevo vidíme, že jak se proces přesunuje ke konci řetězce, efektivnost se k lineárnímu modelu přibližuje. To naznačuje, že povaha nákladů a z ní

plynoucí způsob rozhodování daného článku má tendenci ovlivnit hlavně články, které v řetězci následují. V grafu napravo si můžeme všimnout, že rozložení zisku se s přesunem výrobního procesu mění v tom smyslu, že u článku, který proces zajišťuje, dojde vždy k vychýlení ziskového podílu (a rovněž celkového zisku) směrem dolů. Při použití nákladové funkce s negativními úsporami z rozsahu ovšem takového výsledku nedosáhli. V grafu 3-26 jsou znázorněny hodnoty řetězce v případě, že se přesouvající proces vyznačuje konvexní nákladovou křivkou ve tvaru

$$tc = 0,002 \cdot q^2.$$

**Graf 3-26: Příklad 17 – efektivnost a rozložení zisku (záporné úspory z rozsahu)**



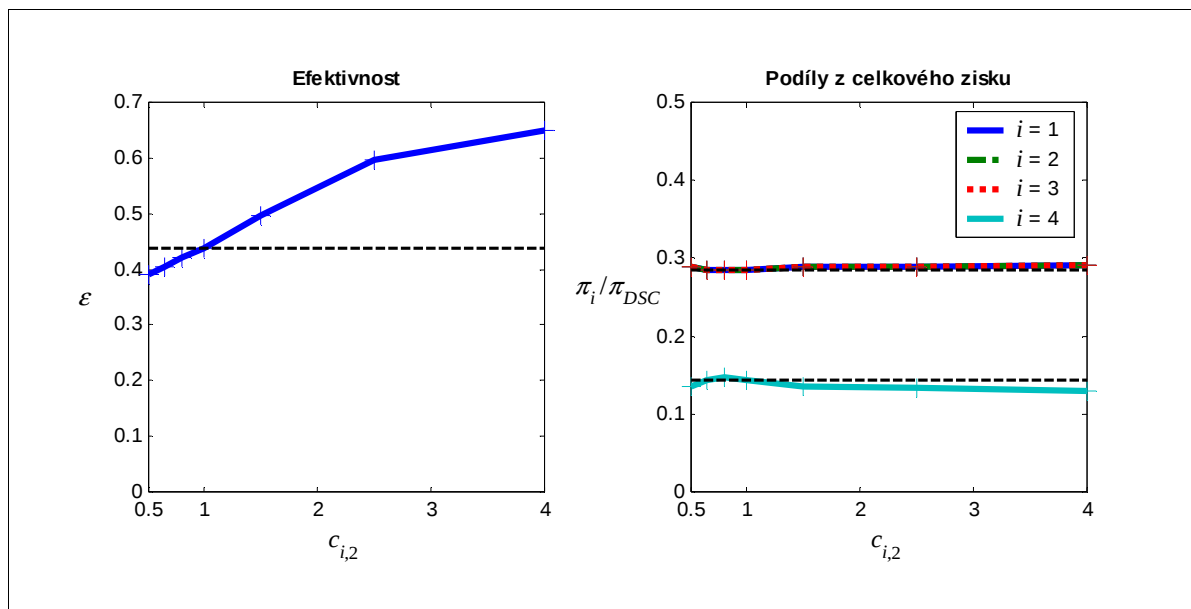
Efektivnost vykazuje podobné vlastnosti jako v předchozím případě, tj. s přesunem procesu ke konci řetězce se blíží k efektivnosti lineárního modelu (nyní ovšem z opačné strany, v souladu se závěry z předchozích příkladů). Ziskové podíly se však tentokrát drží bez výrazných výkyvů na úrovni lineárního modelu. Rozdíl mezi rozložením zisku v obou příkladech lze vysvětlit následujícím způsobem. Prudký nárůst nákladů pro několik prvních jednotek výroby v případě nákladové funkce s úsporami z rozsahu lze přirovnat k fixním nákladům na zahájení výroby. Pokud by byly náklady procesu tvořeny pouze těmito fixními náklady, pak by na rozhodování článku o ceně výstupu neměly takřka žádný vliv, neboť z hlediska optimalizace zisku by vystupovaly jako (při extremalizaci nezajímavá) konstanta. Fixní náklady zahájení výroby mohou toliko ovlivnit úroveň vstupní ceny, při které se článek rozhodne uzavřít výrobu (z důvodu, že není schopen dosahovat kladného zisku). V takovém případě by břemeno zajištění procesu znamenalo pro daný článek pouze povinnost uhradit ze svého zisku tyto fixní náklady, zisk postiženého článku by se tedy snížil o úroveň fixních nákladů. Podobným mechanismem, i když ne v přesně stejném měřítku, dochází k poklesu

zisku postižených článků i v obecnějším případě procesu s nákladovými funkcemi vykazujícími úspory z rozsahu.

Mohli bychom logickou úvahou očekávat, že v případě nákladové funkce se zápornými úsporami z rozsahu bude situace opačná, totiž že článek, který proces vykonává, bude dosahovat většího ziskového podílu. Ačkoli to z grafu není úplně patrné, porovnáním numerických hodnot bychom zjistili, že tomu tak skutečně je – každý článek dosahuje největšího ziskového podílu, právě když proces zajišťuje právě on. Rozdíly jsou v tomto případě ovšem jen velmi nepatrné, ziskové podíly článků se v jednotlivých případech liší o méně než jedno procento. Podobných málo výrazných rozdílů bychom se dočkali i v jiných příkladech. Důvod je ten, že odchýlení od lineárních nákladů, ke kterému může dojít, je výrazně menší, než v případě úspor z rozsahu. Směrnice nákladové funkce (tedy funkce mezních nákladů) je omezená dvěma faktory: na svém počátku musí být nezáporná a v bodě vyráběného množství nemůže být pro zisk optimalizující firmu příliš vysoká (tj. mezní náklady nesmí být vyšší, než mezní příjmy). Tvar nákladové funkce do bodu vyráběného množství se tedy od lineární funkce zpravidla nemůže lišit tolik, jako v případě nákladů s úsporami z rozsahu, kdy může být směrnice na začátku libovolně vysoká a v bodě výroby třeba i velmi nízká.

Podívejme se nyní, jak se nelinearita nákladových funkcí projeví v paralelních řetězcích. Budeme opět vycházet z příkladů pro sériové řetězce (tj. příkladů 15, 16 a 17), abychom mohli snadno porovnat dopad rozdílné struktury řetězce.

**Graf 3-27: Příklad 18 – efektivnost a rozložení zisku**



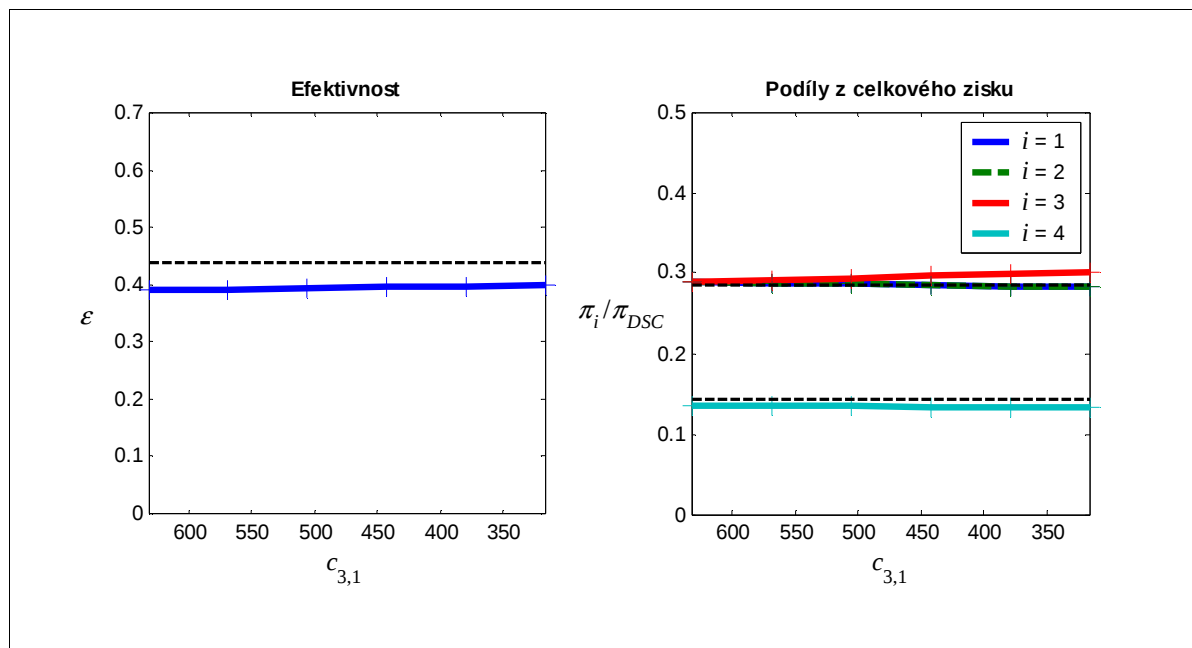
**PŘÍKLAD 18:** Uvažujme paralelní dodavatelský řetězec se čtyřmi články a poptávkovou funkcí ve tvaru (3.17). Budeme sledovat, jak se mění efektivnost řetězce a rozložení zisku mezi jednotlivé články, pokud za nákladové funkce všech článků postupně dosazujeme stejné

funkční tvary, jako v příkladu 15, tj. nákladové funkce zobrazené grafem 3-20. Graf 3-27 zobrazuje získané výsledky, na horizontální ose je podobně jako v příkladu 15 vynesena hodnota druhého parametru nákladové funkce, který určuje, zda je nákladová funkce konkávní, lineární nebo konvexní. Výsledky dosahované v lineárním modelu jsou jako obvykle naznačeny černou čárkovanou čarou. Situace je podobná, jako v případě sériového řetězce. Ziskové podíly jsou opět velmi blízké lineárnímu modelu, s hodnotou parametru  $c_{i,2}$  vzdalující se od jednotky (tj. od lineárních nákladů) se rozdíl mezi ziskovými podíly prohlubují. V grafu nalevo vidíme, že se snižujícími se úsporami z rozsahu dochází opět k růstu efektivnosti. To není nikterak překvapivý výsledek, neboť diskuzi, kterou jsme odůvodnili tento jev pro sériový řetězec, lze beze změny vztáhnout i k řetězci paralelnímu. Výsledky tohoto příkladu dále napovídají, že se tento jev projeví v obou řetězcích v podobném rozsahu.

Příklad 19 ověří platnost závěrů ohledně externalit nákladových úspor, které jsme učinili na základě výsledků příkladu 16.

**PŘÍKLAD 19:** Budeme uvažovat paralelní dodavatelský řetězec se čtyřmi články a poptávkou (3.17). Náklady článků jsou shodné, jako náklady článků se stejnými indexy s příkladu 16. Ve všech nákladových funkcích lze tedy dosáhnout úspor z rozsahu a zkoumáme situaci, kdy článek 3 začne postupně snižovat své náklady pro všechna vyráběná množství tak, jak to popisuje graf 3-23. V grafu 3-28 vidíme, jak se při snižujících nákladech třetího článku vyvíjela efektivnost řetězce a ziskové podíly článků.

Graf 3-28: Příklad 19 – efektivnost a rozložení zisku



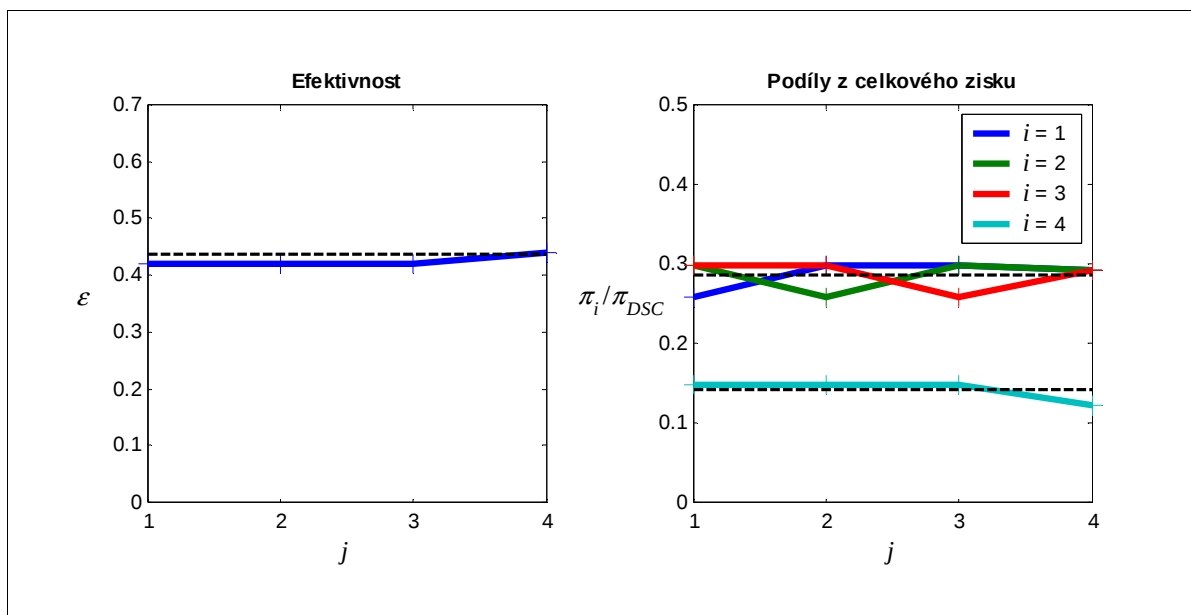
Stejně jako v případě sériového řetězce, i zde dochází k pozitivním externalitám v téměř stejné podobě, jako i lineárního modelu – efektivnost ani ziskové podíly se výrazněji nemění,

což indikuje, že individuální zisky rostou v reakci na pokles nákladů třetího článku přibližně ve stejné proporcii. Lehký nárůst efektivnosti řetězce můžeme vysvětlit analogicky jako u sériového řetězce.

Poslední příklad tohoto pododdílu se věnuje vlivu rozložení nákladů v paralelním řetězci na rozdělení zisku.

**PŘÍKLAD 20:** Uvažujme paralelní dodavatelský řetězec se čtyřmi články a poptávkovou funkcí ve tvaru (3.17). Mezi články budeme přesouvat výrobní proces, který všechny články provádí s náklady podle předpisu (3.18), ostatní náklady článků jsou nulové. Graf 3-25 ukazuje rozložení zisku a efektivnost v jednotlivých případech, index  $j$  na horizontální ose představuje číslo článku, který právě proces zajišťuje. Z grafu je patrné, že pro každý článek znamená úkol zajistit výrobní proces vždy snížení individuálního zisku. To ovšem platí pouze proto, že výrobní proces vykazuje úspory z rozsahu, jak bylo vysvětleno v příkladu se sériovým řetězcem. Pokud bychom pro proces zvolili nákladovou funkci se zápornými úsporami z rozsahu, byly by rozdíly mezi dosahovanými ziskovými podíly jen nepatrné a s opačnou tendencí, tj. s tendencí zvýšit článek, který proces zajišťuje. Stejně jako v sériovém modelu dochází ke zvýšení efektivnosti, pokud se výrobní proces přesunuje ke konci řetězce. Články v první úrovni řetězce mají ovšem ve struktuře řetězce identické postavení, a efektivnost je proto v prvních třech případech zcela totožná.

Graf 3-29: Příklad 20 – efektivnost a rozložení zisku



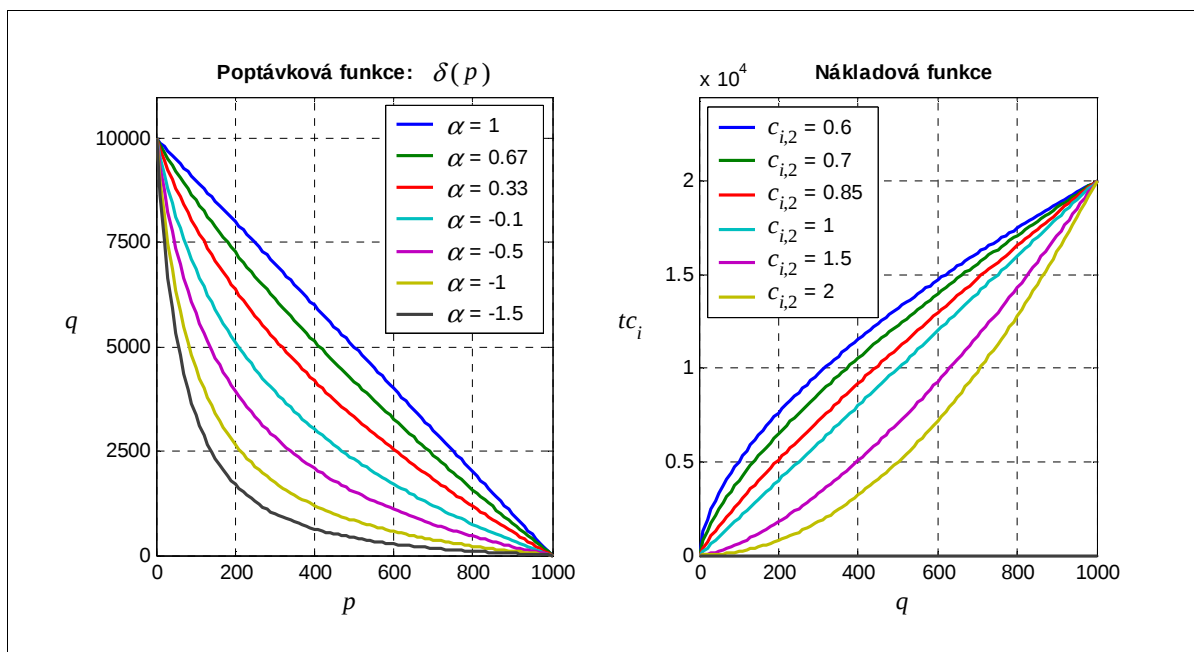
### 3.3.3 Nelinearita v poptávkové funkci i nákladových funkcích

V předchozích pododdílech jsme posuzovali oddělené vlivy nelinearity v poptávkových a nákladových funkcích. V tomto pododdílu provedeme pouze jediný experiment, který ukáže, jakým způsobem se popsané důsledky nelinearity v poptávkových a nákladových

funkcích projeví, pokud mají nelineární tvar nákladové funkce i poptávková funkce zároveň. Následující příklad byl navržen takovým způsobem, aby obsáhl najednou co nejrozumnější nastavení poptávkových a nákladových funkcí. Původním záměrem bylo zkombinovat všechny poptávkové funkce uvažované v příkladu 7 (a jeho variantě pro paralelní řetězec, příkladu 11) a nákladové funkce z příkladu 15 (resp. 18). Některé kombinace uvažovaných funkcí však musely být vyřazeny z toho důvodu, že pro ně neexistovaly rovnovážné ceny (kvůli příliš vysokým nákladům článků).

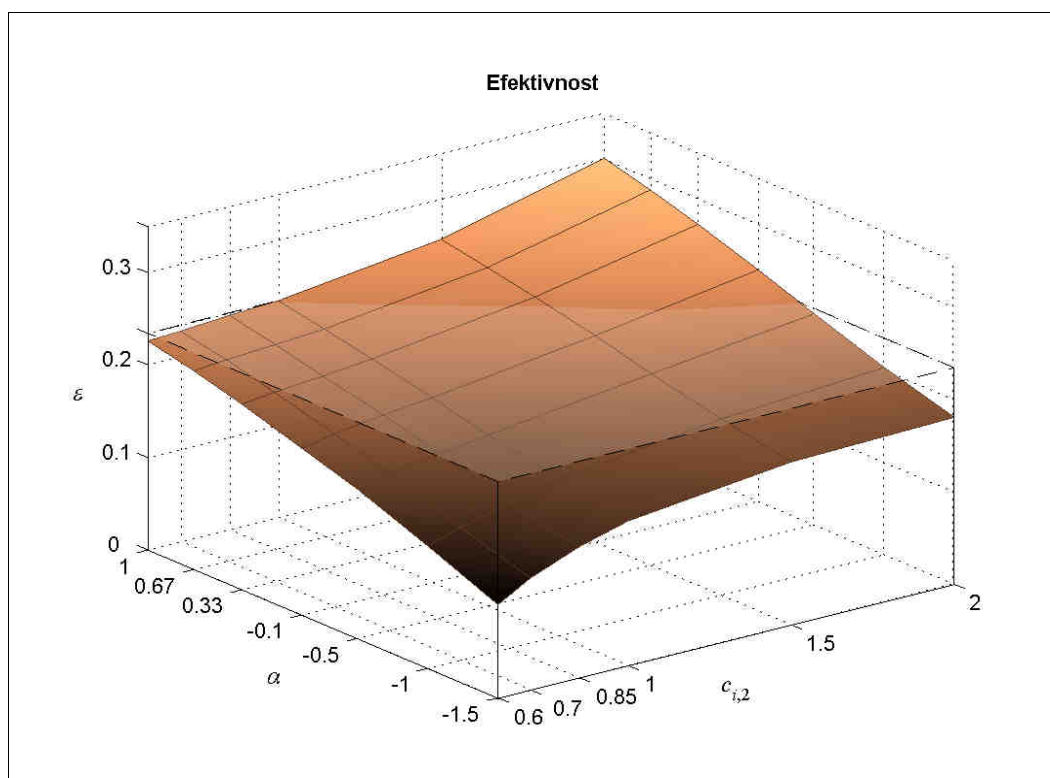
**Příklad 21:** Budeme uvažovat dodavatelský řetězec se čtyřmi články. V roli poptávkových funkcí budeme postupně uvažovat několik různých tvarů funkce  $\delta(p)$  lišících se nastavením parametrů. Jejich konstrukce je podobná jako v příkladu 7, hodnota parametru  $b$  je pro všechny funkce rovna 100, míra prohnutí funkce je kontrolována pomocí parametru  $\alpha$  a zbylé parametry jsou dopočítány tak, aby všechny poptávkové křivky procházely průsečíky s osami  $p, q$  s hodnotami (0;10.000) a (1000;0). Budeme uvažovat sedm různých hodnot parametru  $\alpha$  z intervalu  $[-1,5;1]$ , pro klesající hodnoty se poptávková funkce vzdaluje od funkce lineární. Všechny použité funkce znázorňuje graf 3-30 vlevo. Nákladové funkce všech článků řetězce jsou vždy shodné, za konkrétní funkční tvar budeme dosazovat křivky podobné těm z příkladu 15, které vidíme v grafu vpravo. Jedná se o opět mocninné nákladové funkce, přičemž míra dosahovaných úspor z rozsahu je dána parametrem  $c_{i,2}$ .

**Graf 3-30: Příklad 21 – poptávkové a nákladové funkce při různých hodnotách parametrů**

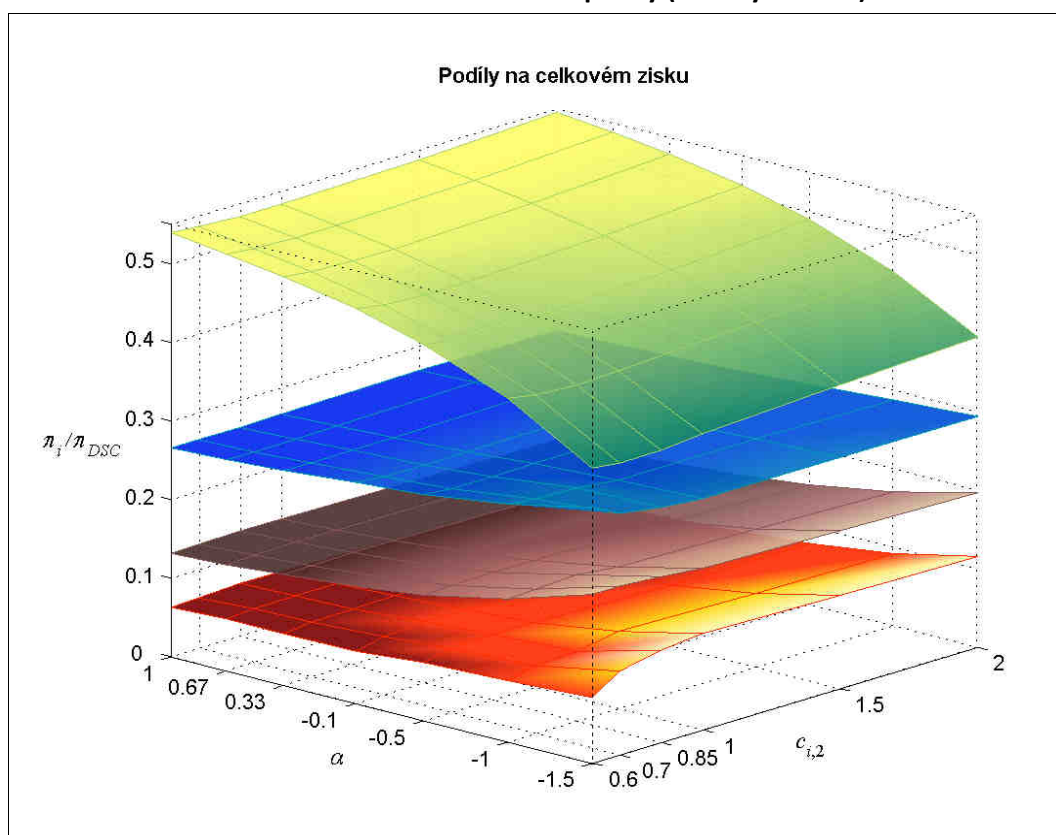


Předpokládejme nejprve, že uvažovaný řetězec má sériovou strukturu. Pro uvedených 7 poptávkových funkcí a 6 funkcí nákladových můžeme zkonstruovat 42 možných scénářů dodavatelského řetězce.

**Graf 3-31: Příklad 21 – efektivnost sériového řetězce**



**Graf 3-32: Příklad 21 – ziskové podíly (sérový řetězec)**

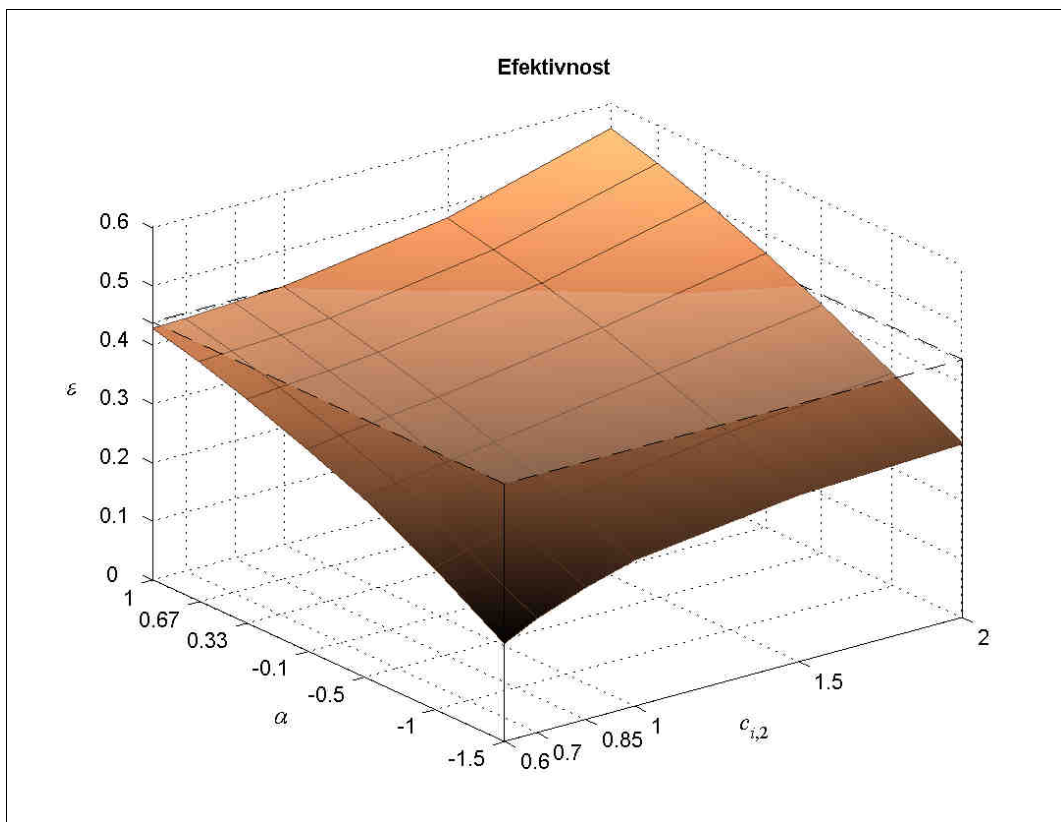


Efektivnost řetězce ve všech těchto případech ukazuje graf 3-31. Na půdorysných osách jsou vyneseny hodnoty parametru  $\alpha$  poptávkové funkce a parametru  $c_{i,2}$  funkce nákladové, vertikální osa měří dosaženou efektivnost. Průsvitná rovina kolmá k vertikální ose vymezená čárkovanými průsečíky s průmětnami představuje efektivnost sériového řetězce se čtyřmi články v lineárním modelu. Můžeme si všimnout, že i v případě kombinace nelineárních funkcí v roli poptávky i nákladů se projevují všechny tendence, které jsme popsali v předchozích experimentech: s rostoucí konvexitou poptávkové funkce efektivnost řetězce klesá a naopak roste se snižujícími se úsporami z rozsahu. Zdá se přitom, že dopad odchýlení tvaru poptávkové funkce od lineárního je ještě zesílen v přítomnosti nelineárních nákladových funkcí (a to jak v případě kladných, tak záporných úspor z rozsahu). Vliv nelinearity nákladových funkcí se s tvarem poptávkové křivky mění následujícím způsobem: při rostoucím prohnutí poptávkové křivky se zvyšuje negativní vliv kladných úspor z rozsahu na efektivnost řetězce a naopak snižuje pozitivní dopad záporných úspor z rozsahu. Některé další experimenty ovšem tento trend jednoznačně nepotvrdily, v některých případech naopak s větším prohnutím poptávkové funkce dopad kladných úspor z rozsahu oslaboval.

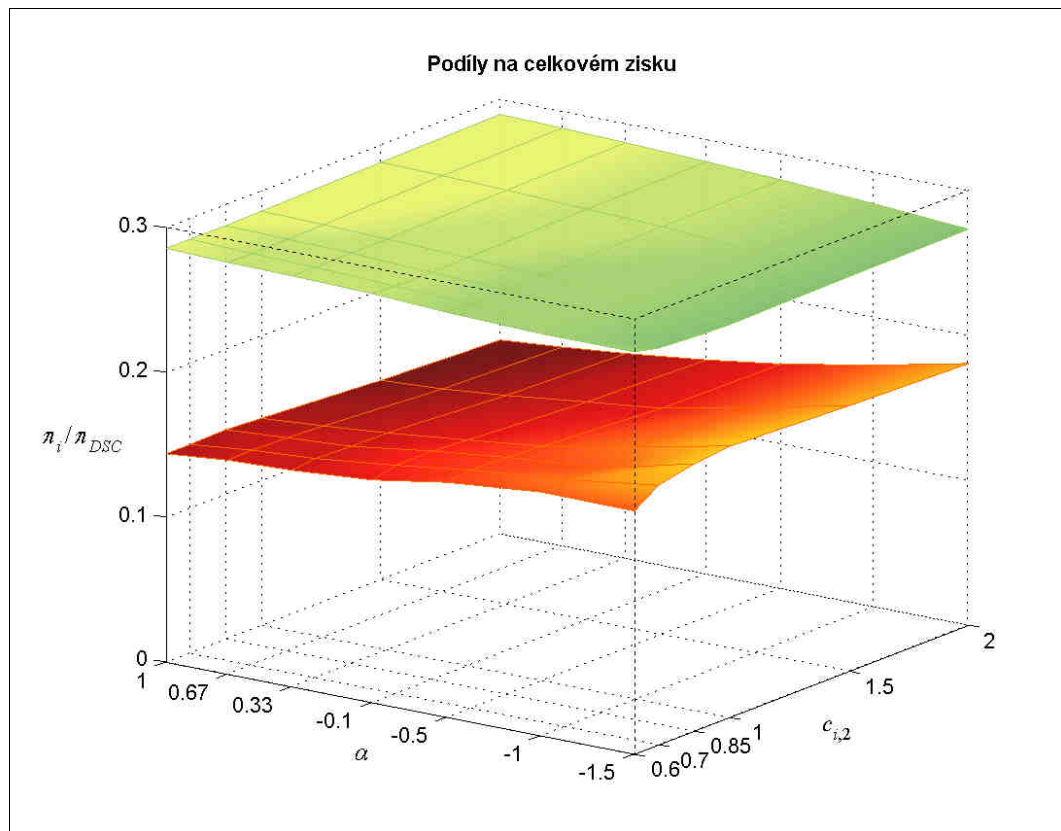
V grafu 3-32 jsou pomocí barevných ploch pro všechny kombinace nákladových a poptávkových křivek zobrazeny ziskové podíly jednotlivých článků. Vrchní zelená plocha zaznamenává podíl článku 1, modrá plocha níže podíl článku 2 atd. Z grafu je patrné, že pořadí ziskových podílů podle velikosti zůstává pro všechny kombinace stejné. Na změnu proporce ziskových podílů má výrazný dopad především tvar poptávkové křivky. Vidíme, že tendence k přiblížení ziskových proporcí s prohnutím poptávkové funkce, kterou jsme sledovali v modelu s lineárními náklady, je zachována i v případě nelineárních nákladových funkcí. Tendence k oddálení ziskových podílů při zvýraznění úspor z rozsahu (kladných i záporných), kterou jsme popsali za podmínek lineární poptávky, je sice rovněž přítomna, je ovšem výrazně slabší, nežli vliv tvaru poptávkové funkce.

Podívejme se nyní, jak vypadají výsledky pro paralelní strukturu řetězce. Graf 3-33 zachycuje efektivnost paralelního řetězce podobným způsobem jako graf 3-31 pro řetězec sériový. Z grafu je patrné, že průběh efektivnosti je velmi podobný sériovému případu, pro efektivnost v paralelním řetězci platí v zásadě všechny závěry, které byly učiněny pro řetězec sériový. Poměr efektivnosti sériového řetězce a jeho paralelního protějšku se v jednotlivých případech neodlišoval od poměru pro lineární řetězec, který dosahuje podle (3.16) hodnoty cca 0,54. Hodnoty dosahované zkoumanými případy se pohybovaly v rozmezí od 0,52 po 0,64. V grafu 3-34 jsou znázorněny ziskové podíly jednotlivých článků. Vzhledem k tomu, že články v první úrovni řetězce mají identické podmínky, dosahují ve všech případech přesně stejného zisku a jejich ziskový podíl představuje horní zelená plocha, zisk článku koncového zachycuje plocha červená. Ani v případě paralelního řetězce se při kombinaci nelineárních poptávkových a nákladových funkcích nijak výrazně nezmění povaha jevů popsaných při odděleném vlivu nelinearity v předchozích pododdílech.

**Graf 3-33: Příklad 21 – efektivnost paralelního řetězce**



**Graf 3-34: Příklad 21 – ziskové podíly (paralelní řetězec)**



### 3.3.4 Shrnutí výsledků výpočetních experimentů

Cílem celého oddílu 3.3, který jsme vytyčili v jeho úvodu, bylo porovnat dodavatelské řetězce fungující v nelineárních podmínkách s řetězci, jejichž vlastnosti popisuje lineární model z předchozí kapitoly. Konkrétní výčet zkoumaných vlastností byl uveden tamtéž. Obecně vzato lze říci, že některé ze sledovaných vlastností byly ve zkoumaných řetězcích lineárnímu modelu možná až překvapivě blízko, jiné se s prvky nelinearity ve specifikaci řetězce od lineárního modelu odchýlily výrazněji, o výsledky lineárního modelu se však zjevně jde i při studiu řetězců v nelineárním prostředí v mnoha ohledech opřít.

Co se týče efektivnosti řetězce, obecně se potvrzuje výsledek lineárního modelu, že efektivnost klesá s počtem článků řetězce a že paralelní řetězce jsou efektivnější, nežli řetězce sériové. Vliv počtu článků na efektivnost řetězce byl explicitně zkoumán pouze porovnáním příkladů 7 a 8, resp. 11 a 12, nicméně výsledky dosahované v ostatních experimentech poukazují na poměrně blízký vztah efektivnosti různých řetězců se stejnou strukturou. Absolutní velikost dosahované efektivnosti v konkrétních případech samozřejmě ovlivňuje tvar poptávkové křivky (s rostoucím prohnutím křivky efektivnost klesá) a křivek nákladových (při kladných úsporách z rozsahu efektivnost rovněž klesá, viz argumentace u příkladu 15), to ovšem nemá výrazný vliv na relativní změnu efektivnosti v případě, že daný řetězec prodloužíme či zkrátíme o určitý počet článků. Experimenty v zásadě potvrzují, že efektivnost řetězce v nelineárním prostředí se obecně vyvíjí s počtem článků podobným způsobem, jako efektivnost v lineárním modelu.

Velmi nevýrazně se i při nejružnějších kombinacích měnil poměr efektivnosti sériového řetězce a jeho paralelního protějšku, jak ukazují především příklady 7 a 11, 8 a 12 a příklad 21. Poměr efektivnosti sériového řetězce a paralelního řetězce z lineárního modelu, který udává vztah (3.16), se tedy zdá být s jistou tolerancí poměrně univerzální (samozřejmě za předpokladu, že nákladové funkce a funkce poptávky jsou v obou řetězcích totožné nebo alespoň velmi podobné).

Ve všech výpočetních experimentech se potvrdilo, že články ve vyšších (koncových) úrovních decentralizovaného řetězce dosahují menšího zisku, nežli články předcházející. Přesný poměr těchto zisků se pak mění především s tvarem poptávkové funkce, a sice v tom smyslu, že s rostoucí konvexitou poptávkové funkce se ziskové podíly navzájem přibližují, přitom tento vliv je patrný především v řetězcích s menším počtem článků (jak ukázaly příklady 7 a 8, 11 a 12 a příklad 21). Články ve stejné úrovni řetězce pak mají tendenci dosahovat stejného zisku v případě, že nákladová funkce nevykazuje úspory z rozsahu. Pokud dosahují některé články úspor z rozsahu, vede to k tlaku na snížení jejich individuálního ziskového podílu ve prospěch ostatních (viz příklady 17 a 20).

Jako jedna z nejstálejších vlastností při měnících se poptávkových i nákladových funkcích se ukazovala přítomnost pozitivních externalit nákladových úspor v rámci řetězce,

kteřé se věnovaly pro různé druhy řetězců příklady 9, 13, 16 a 19. Ve všech případech docházelo při snížení nákladů jednoho z článků ke zvýšení zisku všech článků řetězce, a to vždy přibližně v proporci stávajících individuálních zisků.

Při přesouvání výrobních procesů v rámci řetězce je podstatné, zda se jedná o proces, který vykazuje kladné nebo záporné úspory z rozsahu. Při kladných, příp. žádných úsporách z rozsahu docházelo k lehkému zvýšení zisku řetězce (a jeho efektivnosti) při přesunu procesu ke konci řetězce, v případě záporných úspor z rozsahu se efektivnost naopak snižovala (příklady 10, 14, 17 a 20). Individuální zisky se v případě záporných nebo žádných úspor z rozsahu výrazně neměnily, proces s kladnými úsporami z rozsahu pak z hlediska zisku znevýhodňoval článek, který jej zajišťoval.

K poklesu zisku článků, které vykonávají procesy s úsporami z rozsahu, dochází podobným mechanismem, jako v případě procesů s fixními náklady (jak bylo vysvětleno v diskuzi k příkladu 17). Žádný příklad zahrnující fixní náklady nebyl do tohoto oddílu zahrnut. Výpočetní experimenty, které jsem v této oblasti provedl, potvrzují dříve uvedenou úvahu, že fixní složka nákladů nehraje z hlediska tvorby cen při optimalizaci zisku žádný vliv, a do výsledných cen decentralizovaného řetězce se tudíž nijak nepromítne. Zisk daného článku je pouze ex post snížen o fixní složku nákladů. Fixní náklady mají tedy přirozeně vliv na veškeré ukazatele, které se zabývají proporcemi zisku (tj. i na efektivnost). V modelech s fixními náklady lze však veškeré předchozí závěry aplikovat na situaci před snížením individuálních zisků o fixní náklady. Předchozí závěry o nákladových externalitách a vlivu rozložení nákladů v řetězci se rovněž pochopitelně týkají pouze nákladů variabilních.

### 3.4 Modely s diskrétními cenami

V tomto oddílu se budeme zabývat situací, kdy články nevolí pro svůj výstup ceny z nějakého spojitého intervalu (např.  $[0;1000]$ ), nýbrž pouze z určité konečné diskrétní škály hodnot. Takovou situaci si lze v praxi samozřejmě jednoduše představit, koneckonců u většiny komodit bývá zvykem určovat cenu s přesností nejvýše na dvě desetinná místa v dané měně. Při maximální ceně daného výrobku 10 € tak má prodejce nejvýš 1000 možností, jak cenu stanovit.

Pro nalezení rovnovážných cen a určení efektivnosti řetězce v modelech s diskrétními cenami jsem vyvinul výpočetní algoritmy, které jsou součástí přiloženého CD. Tyto algoritmy pracují s cenami v podobě přirozených čísel, pro ceny s jemnějším či hrubším rozlišením lze příslušným způsobem převést jednotky v parametrech poptávkové funkce. Pro sériové a paralelní řetězce čítající nanejvýš pět článků, které pracují s poptávkovými a nákladovými funkcemi popsány v oddílu 3.2, lze opět využít popsaného uživatelského rozhraní s tím, že v rolovacím seznamu *Pricing options* v rámečku *Approximation settings* zvolíme položku *Discrete (integer)* – viz příloha P.1.

Jednou z výhodných vlastností algoritmů pro diskrétní ceny je však fakt, že nevyžadují (na rozdíl od předchozích postupů) spojitě poptávkové a nákladové funkce. Obecněji řečeno, tvar poptávkových a nákladových funkcí může být takřka libovolný. Kromě samotného posouzení dopadu diskrétních cen tak mohou následující algoritmy sloužit ke zkoumání řetězců s různými poptávkovými a nákladovými funkcí, pro které nelze předchozí algoritmy použít. Při použití dostatečně jemné cenové stupnice pak dosáhneme přibližně stejných výsledků, jako ve spojitém modelu, rizikem v takovém případě je ovšem dlouhý výpočetní čas.

### 3.4.1 Sériové řetězce

Algoritmus pro nalezení rovnovážných cen pro sériové řetězce je implementován ve funkci *equild* a přímo vychází z varianty pro spojitě ceny, kterou rozebíral pododdíl 3.1.1. Zmíníme zde pouze některé body, ve kterých se postup pro diskrétní ceny liší.

**Kroky 2 a 7:** Omezení volby cen na diskrétní hodnoty se projeví především při práci s cenovými reakčními funkcemi, které jsme značili  $p_i^{opt}$  a které byly pro různá  $i$  zavedeny v krocích 2 a 7. Při celočíselných cenách je jak definičním oborem, tak oborem hodnot množina  $\{1, 2, \dots, p_{max}\}$ . Vztah, který v krocích 2 a 7 implicitně definuje cenové reakční funkce, potom můžeme pro  $i = 2, 3, \dots, n$  přepsat takto:

$$p_i^{opt}(p_{i-1}) = \arg \max_{p_i \in \{1, \dots, p_{max}\}} \pi_i(p_i, p_{i-1}). \quad (3.19)$$

Vzhledem k tomu, že hledáme maximum na konečné množině hodnot, není třeba existenci takového maxima hlídat jako v případě spojitých cen. Může se ovšem stát (a nezdá se to opravdu stává), že maximálního zisku lze dosáhnout pomocí různých hodnot výstupní ceny  $p_i$ . Implementovaný algoritmus v takovém případě pro předpis funkce  $p_i^{opt}$  použije nejmenší z takových hodnot. Pokud by se články rozhodly za dané situace volit nějakou vyšší cenu (ačkoli by to nevedlo ke zvýšení zisku), vypadaly by reakční cenové funkce, a tedy i rovnovážné ceny, trochu jinak. Obecně by vznikla tendence volit vyšší cenu výstupu, která by vedla k poklesu celkového zisku řetězce a jeho efektivnosti.

Jelikož funkce  $p_i^{opt}$  má konečný definiční obor, je možné popsat zobrazení pomocí vyčerpávajícího výčtu dvojic obraz-vzor. Není tedy na rozdíl od spojitých funkcí potřeba činit jakékoli předpoklady o funkčním tvaru. Ve funkci *equild* jsou funkce  $p_i^{opt}$  reprezentovány pomocí vektorů o délce  $p_{max}$  (ve zdrojovém kódu jsou tyto vektory označovány jako „mapy“), přičemž  $j$ -tá složka vektoru představuje cenu, kterou bude  $i$ -tý článek volit, pokud jeho vstupní cena je rovna  $j$ . Pro vyjádření ziskové funkce za účelem nalezení maxima je opět použito funkce *profit* a maximalizačním nástrojem je v tomto případě základní vestavěná funkce *max*.

**Krok 4:** V kroku 4 algoritmu pro nalezení rovnovážných cen v sériovém řetězci je zavedena funkce  $\psi$  jako složení poptávkové funkce a několika reakčních cenových funkcí. Tato funkce je posléze využita pro vyčíslení zisku daného článku v případě, že všechny navazující články optimalizují individuální zisk. Při použití diskretních cenových reakčních funkcí je  $\psi$  implementována ve funkci *demmap*.

Použijeme-li pro spuštění algoritmu uživatelského rozhraní, zobrazí se po skončení výpočtu velmi podobný výstup jako v případě spojitého modelu. Rozdíl ve výsledcích oproti modelu se spojitými cenami je dán především hrubostí cenové stupnice a délkou řetězce. Platí samozřejmě, že čím hrubší je cenová stupnice, tím větší je dopad diskretizace cen. V případě délky řetězce se vliv diskretních cen umocňuje skládáním cenových reakčních funkcí, jak ilustruje následující příklad.

**PŘÍKLAD 22:** Budeme zkoumat dva sériové dodavatelské řetězce, jeden o třech a druhý o pěti člancích, které všechny vyrábí s jednotkovými náklady ve výši 5 peněžních jednotek. Funkce poptávky po koncové produkci má lineární tvar

$$q = 1000 - 12,5 \cdot p_n, \quad (3.20)$$

přičemž pro oba řetězce porovnáme případ, že výstupní ceny článků mohou být reálné, a případ cen celočíselných. Tabulka 3-35 shrnuje některé ze získaných výsledků. Na první pohled je patrné, že v případě delšího řetězce se rovnovážné hodnoty mění s přechodem k celočíselným cenám podstatně razantněji. Ačkoli cena koncové produkce, a tedy i celkový zisk, jsou i v případě delšího řetězce v obou situacích velmi podobné, ceny uvnitř řetězce a rozdělení zisku mezi články nemají s výsledky lineárního modelu již v podstatě nic společného. Oproti tomu v řetězci se třemi články jsou výsledky v obou případech srovnatelné. Ceny se liší v řádu desetin peněžní jednotky, shodou okolností byl i přes odlišný celkový zisk dokonce dodržen poměr zisků z lineárního modelu.

**Tabulka 3-35: Příklad 22 – porovnání rovnovážných hodnot při spojitých a diskretních cenách**

|                                 | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $\pi_1$ | $\pi_2$ | $\pi_3$ | $\pi_4$ | $\pi_5$ |
|---------------------------------|-------|-------|-------|-------|-------|---------|---------|---------|---------|---------|
| <b>3 články, diskretní ceny</b> | 37    | 58    | 71    | –     | –     | 3600    | 1800    | 900     | –       | –       |
| <b>3 články, spojitě ceny</b>   | 37,5  | 58,8  | 71,9  | –     | –     | 3300    | 1650    | 825     | –       | –       |
| <b>5 článků, diskretní ceny</b> | 54    | 60    | 66    | 72    | 78    | 1225    | 25      | 25      | 25      | 25      |
| <b>5 článků, spojitě ceny</b>   | 32,5  | 51,3  | 63,1  | 71,6  | 78,3  | 591     | 295     | 148     | 74      | 37      |

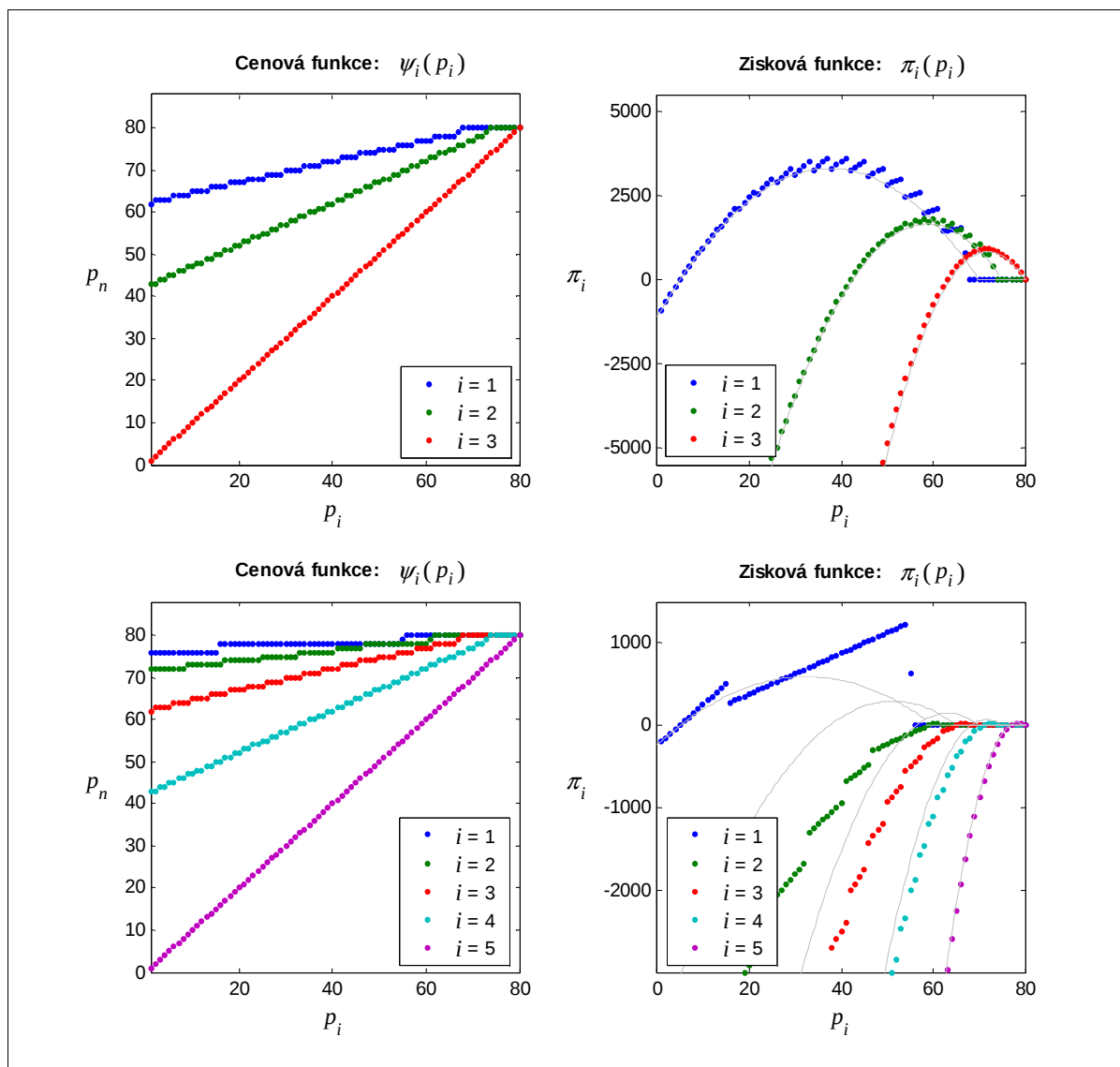
Důvod pro tento jev je dobře patrný z grafu 3-36 na následující straně. Grafy v horní části se týkají řetězce se třemi články, grafy dole řetězce s pěti články. Z grafů vlevo si můžeme udělat obrázek o cenových reakčních křivkách. Funkce  $\psi_i$ , která je v grafech zobrazena, je

funkcí, která se zobrazí po spuštění výpočtu z uživatelského rozhraní. Podrobně je popsána v příloze P.1, je dána předpisem

$$\psi_i = p_{i+1}^{opt} \circ p_{i+2}^{opt} \circ \dots \circ p_n^{opt}$$

a představuje reakci posledního ( $n$ -tého) článku na změnu výstupní ceny  $i$ -tého článku za předpokladu, že články  $i+1, i+2, \dots, n-1$  reagují podle svých individuálních reakčních křivek  $p_i^{opt}$ . S rostoucí délkou řetězce se tato funkce stává pro první články plošší, což při diskrétní cenové škále způsobí opakování některých hodnot (funkce má schodovitý tvar), neboli cena koncové produkce se při některých změnách ceny prvních článků v důsledku diskrétní povahy cen nemění.

Graf 3-36: Příklad 22 – dopad diskrétních cen v různě dlouhých řetězcích



V grafech vpravo vidíme dopad této skutečnosti na individuální ziskové funkce. Zobrazené ziskové funkce jsou opět totožné s výstupem z uživatelského rozhraní, tedy představují zisk daného článku v případě, že článek předchozí volí rovnovážnou cenu a články následující reagují podle svých cenových reakčních funkcí. Tenkou šedou čarou jsou naznačeny ziskové funkce při spojitých cenách. Vidíme, že schody z funkce  $\psi_i$  se přímo promítají do ziskové funkce a dochází tak k jakémusi „roztřepení“ jejího tvaru. V případě řetězce se třemi články jsou schody poměrně nevýrazné a zisková funkce se tak do značné míry drží tvaru funkce ze spojitého modelu. Pro pět článků je však již efekt tak silný, že se zisková funkce prvního článku již svému spojitému protějšku skoro nepodobá. Maxima této funkce je dosahováno pro výrazně odlišnou hodnotu výstupní ceny, což pak zpětně posune i ziskové funkce následujících článků.

### 3.4.2 Paralelní řetězce

V paralelních dodavatelských řetězcích je potřeba algoritmus pro nalezení rovnovážných cen přizpůsobit pro použití s diskrétními cenami podstatně více, nežli tomu bylo u řetězců sériových. Problematickým místem je zde (podobně jako při spojitých cenách) nalezení Nashovy rovnováhy pro cenové strategie článků z první úrovně řetězce. Obecně nelze zaručit, že rovnovážné strategie vůbec existují, všechny provedené experimenty však naznačují, že tomu tak v běžných případech bývá. Na rozdíl od spojitého modelu se zde však velmi často vyskytuje hned několik rovnovážných bodů a nastává otázka, který z nich vybrat nebo zda se uchýlit k popisu rovnováhy pomocí smíšených strategií (viz např. [11]). V implementaci výpočetních algoritmů jsem k tomuto problému přistoupil tak, že výstup funkcí zahrnuje všechny kombinace výstupních cen, které splňují definici Nashovy rovnováhy.

Vzhledem k tomu, že všechny články mají konečný prostor cenových strategií, bylo by možné prostor strategií vyčerpávajícím způsobem prohledat a pro každou kombinaci cen testovat, zda nesplňuje podmínky rovnováhy. Je však zcela evidentní, že v takovém případě dochází velmi rychle ke kombinatorické explozi prohledávaných řešení. Už jen pro paralelní řetězec s pěti články a maximální cenou 1000 peněžních jednotek máme teoreticky  $10^{12}$  možných kombinací cen článků v první úrovni řetězce<sup>1</sup>. Algoritmus, který jsem implementoval v podobě funkce *equilpard*, uplatňuje myšlenku ze spojitého modelu tak, aby bylo možné kombinatorickou explozi v co nejvyšší míře eliminovat a přesto nalézt všechny rovnovážné body.

Zopakujme stručně jádro postupu nalezení Nashových rovnovážných strategií pro ceny článků z první úrovně řetězce v algoritmu pro spojitě ceny z pododdílu 3.1.2.

---

<sup>1</sup> Je nasnadě, že mnohé z těchto strategií bychom mohli předem vyloučit (například ty, kde součet cen je vyšší, než maximální cena), kombinatorické explozi bychom se ale ani tak nevyhnuli.

- Z podmínky maximalizace zisku koncového článku  $n$  se nejprve určí funkce  $\psi$ , která vyjadřuje poptávané množství koncové produkce jakožto funkci součtu cen všech článků první úrovně řetězce (kroky 1, 2 a 3 popsaného algoritmu).
- Pro každý z článků první úrovně se pomocí funkce  $\psi$  a z podmínky maximalizace zisku vytvoří funkce  $p_i^{opt}$ , jež určuje výstupní cenu  $i$ -tého článku, která maximalizuje zisk, v závislosti na součtu cen *všech ostatních* článků první úrovně řetězce (kroky 5 a 6).
- Z funkce  $p_i^{opt}$  je pro každý článek odvozena funkce  $\tilde{p}_i^{opt}$ , která je funkcí součtu cen *všech* článků řetězce (tj. včetně článku  $i$ ). Pro nalezení rovnovážných cen článků první úrovně pak stačí najít takovou hodnotu součtu těchto cen  $s$ , která splňuje

$$s = \sum_{i=1}^{n-1} \tilde{p}_i^{opt}(s).$$

První bod lze provést jen s malými úpravami v podobném duchu, jako tomu bylo u sériového řetězce. Funkce  $\psi$  se pak od spojitého modelu odlišuje tím, že jejím definičním oborem jsou pouze přirozená čísla  $\{1, 2, \dots, p_{max}\}$ . V případě, že zisk koncového článku maximalizuje za daného součtu vstupních cen více hodnot výstupní ceny, počítá algoritmus stejně jako u sériového řetězce s volbou nejnižší z nich.

Větší obezřetnosti k problematice násobných maxim ziskových funkcí je zapotřebí v případě článků z první úrovně řetězce. Pokud bychom se rozhodli počítat s funkcemi  $p_i^{opt}$  dle předpisu (3.19) a jako funkční hodnotu v případě několika maximalizujících výstupních cen vzali tu nejnižší z nich, mohli bychom se připravit o nějaké rovnovážné strategie a následně dojít třeba i k mylnému závěru, že v daném řetězci rovnovážné strategie neexistují.

Přístupu pomocí funkcí  $p_i^{opt}$  a  $\tilde{p}_i^{opt}$  tedy nelze použít, lze se ovšem držet jeho základní myšlenky: pro každou úroveň součtu cen článků první úrovně budeme hledat *všechny* výstupní ceny, při kterých daný článek maximalizuje individuální zisk. Abychom takové kombinace našli, potřebujeme opět nejprve najít *všechny* ceny, které maximalizují zisk při daném součtu cen ostatních článků první úrovně. Jelikož přesný formální zápis celého postupu by byl velmi nepřehledný, budeme postup demonstrovat na konkrétním příkladu.

**PŘÍKLAD 23:** Výpočetní algoritmus pro nalezení Nashovy rovnováhy ukážeme na paralelním řetězci se čtyřmi články se stejnou poptávkovou funkcí jako v příkladu 22 (tj. s funkcí (3.20)). Všechny články operují s jednotkovými náklady 5 jednotek. Jak bylo řečeno výše, na začátku výpočtu se pro každý článek z první úrovně  $i$  a každý součet cen ostatních článků z první úrovně  $j$  hledají výstupní ceny  $p_i$ , které optimalizují zisk. Zajímají nás tedy pro  $i = 1, 2, \dots, n-1$  a  $j = 1, 2, \dots, p_{max}$  všechna řešení úlohy

$$\max_{p_i \in \{1, \dots, p_{max}\}} p_i \cdot \psi(p_i + j) - \gamma_i(\psi(p_i + j)).$$

Pro konkrétní  $i$  a  $j$  lze řešení této úlohy nalézt velmi snadno (díky konečné množině, přes kterou maximum hledáme). Následující výstup z prostředí Matlab představuje výpis řešení úlohy pro druhý článek ( $i = 2$ ) a součty cen ostatních článků z první úrovně (tj. článků 2 a 3) v rozmezí 35 až 45 peněžních jednotek:

|                             |                                    |
|-----------------------------|------------------------------------|
| Soucet cen clanku 1 a 3: 35 | Optimalni cena/y clanku 2: 23      |
| Soucet cen clanku 1 a 3: 36 | Optimalni cena/y clanku 2: 22      |
| Soucet cen clanku 1 a 3: 37 | Optimalni cena/y clanku 2: [21 23] |
| Soucet cen clanku 1 a 3: 38 | Optimalni cena/y clanku 2: 22      |
| Soucet cen clanku 1 a 3: 39 | Optimalni cena/y clanku 2: 21      |
| Soucet cen clanku 1 a 3: 40 | Optimalni cena/y clanku 2: 20      |
| Soucet cen clanku 1 a 3: 41 | Optimalni cena/y clanku 2: [19 21] |
| Soucet cen clanku 1 a 3: 42 | Optimalni cena/y clanku 2: 20      |
| Soucet cen clanku 1 a 3: 43 | Optimalni cena/y clanku 2: 19      |
| Soucet cen clanku 1 a 3: 44 | Optimalni cena/y clanku 2: 18      |
| Soucet cen clanku 1 a 3: 45 | Optimalni cena/y clanku 2: [17 19] |

Pro hodnoty součtu 37, 41 a 45 je namísto jedné optimální ceny článku 2 uveden vektor o dvou složkách, což znamená, že maximálního zisku je dosaženo při obou uvedených cenách. Obě ceny mohou přitom být součástí nějakého rovnovážného bodu, je tedy potřeba pracovat s oběma. Od uvedeného výčtu, který ukazuje možné rovnovážné ceny při daném součtu cen ostatních článků první úrovně, je nyní potřeba přejít ke vztahu kandidátů na rovnovážné ceny a součtu cen všech článků první úrovně. Z prvního řádku výstupu vidíme, že při součtu cen článků 1 a 3 ve výši 35 peněžních jednotek bude druhý článek volit cenu 23, což naznačuje, že pokud budeme hledat rovnovážné ceny při součtu cen všech článků první úrovně  $58 = 23 + 35$ , je cena  $p_2 = 23$  jedním z kandidátů na rovnovážnou cenu. Z druhého a třetího řádku ovšem vyplývá, že stejně tak připadají v úvahu ceny  $p_2 = 22$  a  $p_2 = 21$ , neboť máme  $58 = 22 + 36 = 21 + 37$ . Pokud takovýmto způsobem zohledníme všechny získané optimální ceny, získáme všechny kombinace součtů cen všech článků první úrovně a ceny  $p_2$ , ve kterých může nastat Nashova rovnováha. Jejich kompletní výpis v našem případě je následující:

|                                 |                                    |
|---------------------------------|------------------------------------|
| Soucet cen clanku 1. urovne: 40 | Cena/y clanku 2 : 39               |
| Soucet cen clanku 1. urovne: 42 | Cena/y clanku 2 : [41 40 39 38 37] |
| Soucet cen clanku 1. urovne: 44 | Cena/y clanku 2 : [39 38 37 36 35] |
| Soucet cen clanku 1. urovne: 46 | Cena/y clanku 2 : [37 36 35 34 33] |
| Soucet cen clanku 1. urovne: 48 | Cena/y clanku 2 : [35 34 33 32 31] |
| Soucet cen clanku 1. urovne: 50 | Cena/y clanku 2 : [33 32 31 30 29] |
| Soucet cen clanku 1. urovne: 52 | Cena/y clanku 2 : [31 30 29 28 27] |
| Soucet cen clanku 1. urovne: 54 | Cena/y clanku 2 : [29 28 27 26 25] |
| Soucet cen clanku 1. urovne: 56 | Cena/y clanku 2 : [27 26 25 24 23] |
| Soucet cen clanku 1. urovne: 58 | Cena/y clanku 2 : [25 24 23 22 21] |
| Soucet cen clanku 1. urovne: 60 | Cena/y clanku 2 : [23 22 21 20 19] |
| Soucet cen clanku 1. urovne: 62 | Cena/y clanku 2 : [21 20 19 18 17] |
| Soucet cen clanku 1. urovne: 64 | Cena/y clanku 2 : [19 18 17 16 15] |
| Soucet cen clanku 1. urovne: 66 | Cena/y clanku 2 : [17 16 15 14 13] |
| Soucet cen clanku 1. urovne: 68 | Cena/y clanku 2 : [15 14 13 12 11] |
| Soucet cen clanku 1. urovne: 70 | Cena/y clanku 2 : [13 12 11 10 9]  |
| Soucet cen clanku 1. urovne: 72 | Cena/y clanku 2 : [11 10 9 8 7 6]  |
| Soucet cen clanku 1. urovne: 73 | Cena/y clanku 2 : [7 6]            |

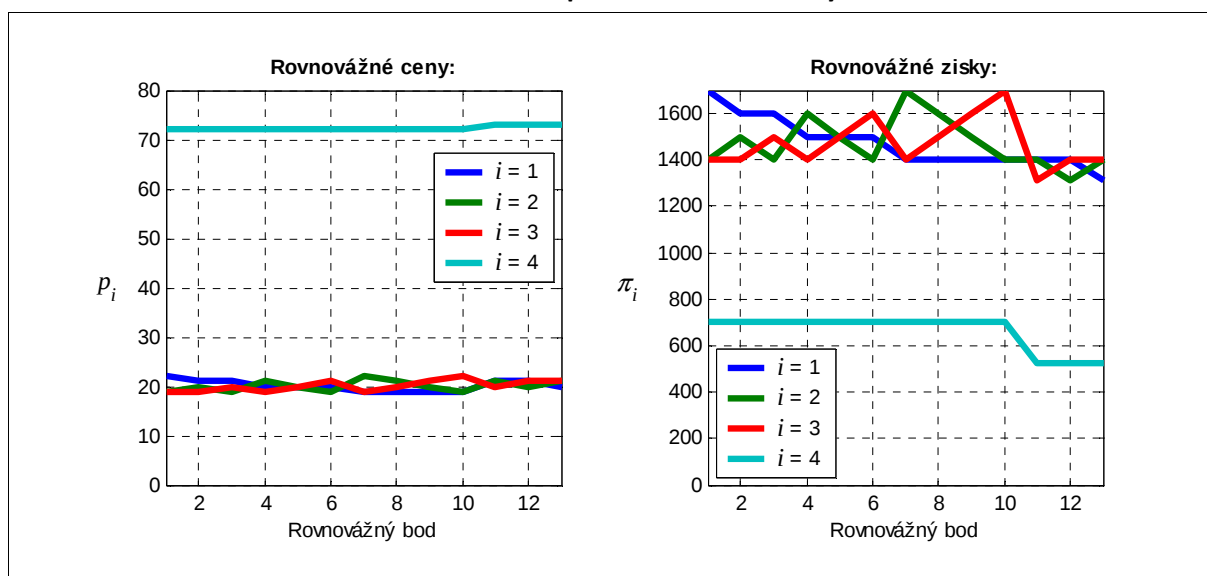
Uvedený seznam je analogií k funkci  $\tilde{p}_i^{opt}$  s tím, že k danému součtu cen článků první úrovně není přiřazena vždy právě jedna hodnota (nejedná se tedy o popis *funkce*), nýbrž výčet

kandidátů, kteří se všichni mohou stát rovnovážnými cenami. Při hledání rovnovážných cen vytvoříme obdobné seznamy pro všechny články první úrovně (v našem případě mají všechny články identické nákladové funkce, seznamy by proto vypadaly úplně stejně jako pro článek 2). Pokud pro nějakou kombinaci cen článků první úrovně platí, že jsou všechny uvedeny ve výčtu kandidátů svého článku pro součet těchto cen, jedná se o ceny rovnovážné. Ve výše uvedeném seznamu, který platí pro všechny články první úrovně, takovou vlastnost splňuje například kombinace  $p_1 = p_2 = p_3 = 20$ , neboť tato cena je v seznamu každého článku uvedena jako kandidát na rovnovážnou cenu při součtu 60 a  $p_1 + p_2 + p_3 = 60$ . Vraťme se nyní k předchozímu výpisu. Vzhledem k tomu, že pro druhý článek je součet cen ostatních v první úrovni roven 40, vidíme, že při ceně  $p_2 = 20$  a daných cenách ostatních článků opravdu optimalizuje článek 2 svůj zisk, podmínka Nashovy rovnováhy je tedy splněna (a zcela stejně vypadá situace u ostatních článků první úrovně). Podobně bychom mohli najít i další rovnovážné kombinace, např.  $p_1 = p_2 = 19$ ,  $p_3 = 22$  nebo  $p_1 = 19$ ,  $p_2 = 20$ ,  $p_3 = 21$ . Jejich kompletní výpis uvádí tabulka 3-37, graficky je pak znázorňuje graf 3-38, který je analogií z grafu, který se vytiskne na obrazovku po spuštění výpočetní procedury z uživatelského rozhraní (srovnej s grafem P-6 v příloze P.1).

Tabulka 3-37: Příklad 23 – různé rovnovážné kombinace cen

|       | Rovnovážný bod č. |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|
|       | 1                 | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 |
| $p_1$ | 22                | 21 | 21 | 20 | 20 | 20 | 19 | 19 | 19 | 19 | 21 | 21 | 20 |
| $p_2$ | 19                | 20 | 19 | 21 | 20 | 19 | 22 | 21 | 20 | 19 | 21 | 20 | 21 |
| $p_3$ | 19                | 19 | 20 | 19 | 20 | 21 | 19 | 20 | 21 | 22 | 20 | 21 | 21 |
| $p_4$ | 72                | 72 | 72 | 72 | 72 | 72 | 72 | 72 | 72 | 72 | 73 | 73 | 73 |

Graf 3-38: Příklad 23 – porovnání rovnovážných bodů



Z principu tvorby seznamů zároveň vyplývá, že kombinace cen, která nesplňuje hledanou vlastnost, že každá z nich je kandidátem v seznamu svého článku pro jejich součet, nemůže být rovnovážnou strategií. Při hledání rovnovážných cen tedy stačí projít všechny součty cen článků první úrovně, které se vyskytují ve všech seznamech současně, a pro každý z nich testovat všechny kombinace příslušných kandidátů. Pokud dají kandidáti dohromady testovaný součet, jedná se o rovnovážnou cenu. Problém kombinatorického nárůstu zkoumaných kombinací cen se tak přesouvá až na úroveň násobnosti kandidátů při jednotlivých součtech. Všechny procedury při konstrukci seznamu jsou totiž co do výpočetního času polynomiálního charakteru. K exponenciálnímu nárůstu výpočetního času s rozsahem úlohy (tj. v tomto případě s počtem článků) dochází až při testování jednotlivých kombinací kandidátů u konkrétního součtu v seznamech. V našem příkladu se pro jeden součet vyskytlo nanejvýš 6 různých kandidátů, pro které je potřeba při třech člancích v první úrovni otestovat  $6^3 = 216$  kombinací rovnovážných cen. Počet různých kandidátů navíc nemá tendenci zvyšovat se s rostoucí maximální cenou a pro otestování dané kombinace je třeba provést jen zcela minimální počet operací (sečíst hodnoty kandidátů a porovnat se součtem v seznamu). Díky těmto příznivým vlastnostem je možné prohledávat i velmi rozsáhlé prostory strategií.

Obecně vzato, v paralelních řetězcích nedochází v důsledku diskretizace cen k takovým odchylkám od výsledků spojitého modelu, jako u řetězců sériových. Je tomu tak proto, že nedochází ke skládání efektu skoků v reakčních cenových funkcích článků. Ziskové funkce článků z prvního okna grafického výstupu proto svým tvarem poměrně věrně kopírují funkce ze spojitého modelu (s tím, že kvůli drobným odchylkám v rovnovážných cenách ostatních článků dochází k mírnému posunutí jednotlivých křivek). S přibývajícím počtem článků se v paralelních řetězcích vliv celočíselných cen nijak výrazně nezesiluje.

## 4 DYNAMICKÉ MODELY VÍCENÁSOBNÉ MARGINALIZACE

Jak bylo vysvětleno v první kapitole, k efektu vícenásobné marginalizace dochází, pokud články řetězce nekoordinují svou cenovou politiku a snaží se maximalizovat individuální zisk způsobem, který je obvyklý pro firmy, které nejsou součástí dodavatelského řetězce – tj. porovnáním (odhadované) funkce poptávky po svém výstupu a funkce nákladové. V této kapitole se pokusíme pomocí počítačové simulace ukázat, jakým způsobem takového chování článků vede k nastolení rovnovážných cen v prostředí, kde jednotlivé články mají zpočátku jen zcela minimální informace o ostatních částech dodavatelského řetězce. Jak název kapitoly napovídá, půjde o model dynamický, v průběhu času budou články na základě odezvy na své výstupní ceny aktualizovat svou představu o funkci poptávky po jejich výstupu a podle toho měnit momentální výstupní cenu.

Prvotní inspirací pro celou tuto kapitolu byl článek [2]. Článek je zaměřen spíše pedagogického, pojednává o experimentech, které prováděli američtí univerzitní pedagogové se svými studenty při výuce problému dvojí marginalizace v hodinách mikroekonomie, hlavním cílem bylo především představit studentům koncept dvojí marginalizace a demonstrovat jeho efekt. Studenti byli rozděleni do několika párů skupin představujících dodavatelské řetězce se dvěma články; jedna skupina v páru vždy představovala výrobce a druhá distributora zboží. Distributor znal funkci poptávky po koncovém produktu (šlo o poptávku lineární), výrobce pouze své výrobní náklady. Experiment probíhal v několika kolech, v každém kole nejprve výrobce oznámil cenu, kterou stanovil pro svůj výstup, a distributor poté určil cenu koncovou a oznámil výrobcí vyšší objemu prodaného množství. Studenti se snažili maximalizovat zisk svých firem na základě svých ekonomických znalostí teorie firmy, a jak popisují autoři článku, s postupujícími koly vesměs docházelo k cenové spirále a ceny obou článků měly jasnou tendenci konvergovat k rovnovážným cenám odvozeným z modelu dvojí marginalizace.

Mým cílem bylo provést podobné experimenty s řetězcí s větším počtem článků a různými strukturami a porovnat, jaký má tvar řetězce dopad na konvergenci k rovnovážným cenám. Z pochopitelných důvodů jsem se rozhodl namísto experimentů s reálnými lidskými bytostmi využít počítačové simulace. Jak je patrné z popsaného experimentu, tvorba cen v řetězcí je výsledkem interakce racionálně uvažujících entit s individuálními cíli a omezenými informacemi o systému, ve kterém se nachází. Dodavatelský řetězec v tomto

pojetí tedy splňuje znaky multiagentního systému a jako takový jej bylo potřeba modelovat. Použité simulační modely proto spadají do kategorie multiagentní simulace. Multiagentní simulace je poměrně moderní a prudce se vyvíjející disciplína, která je zatím ve spojení s ekonomii jako vědní disciplínou v českých krajích poměrně málo využívána<sup>5</sup>. Následující oddíl proto stručně shrne základní myšlenky a východiska této oblasti.

## 4.1 Multiagentní systémy

Historie multiagentních systémů sahá až do 40. let dvacátého století a je spojena s matematikou Von Neumannem a Ulamem, kteří sestrojili tzv. **celulární automaty**. Myšlenky celulárních automatů pak využil anglický matematik John Horton Conway při konstrukci své slavné *Game of Life* (překládáno jako „Hra Život“, „Hra na život“ nebo „Hra Life“, viz [6]) na počátku 70. let. *Game of Life* zkoumá dynamiku systému dvourozměrné mřížky buněk (někdy popisované jako nekonečná šachovnice), každá buňka je v daném okamžiku buď „živá“ nebo „mrtvá“ a celý systém se vyvíjí v diskrétních časových okamžicích podle jednoduchých pravidel:

- *samota*: má-li živá buňka méně než dva živé sousedy (z možných osmi), zemře.
- *přelidnění*: má-li živá buňka více než tři živé sousedy, zemře.
- *reprodukce*: má-li mrtvá buňka právě tři živé sousedy, ožije.

Krátce po uvedení vzbudila *Game of Life* obrovský zájem vědecké i laické veřejnosti. Navzdory velmi jednoduchým a průhledným pravidlům se totiž i jednoduché počáteční konfigurace vyvíjejí nepředvídatelným způsobem – je možné nalézt populace, které se s časem vůbec nemění, populace, které vymírají, populace, které oscilují mezi periodickými stavy, ale také populace, které plynule pohybují napříč mřížkou nebo produkují stále více „potomků“, a rostou tak do nekonečna. Při pohledu na tvar prvotní populace přitom ve většině případů nelze předem budoucí vývoj vůbec odhadnout.

*Game of Life* je tak jednou z nejznámějších manifestací **emergence** – jevu, jehož existence je jedním z hnacích motorů zkoumání multiagentních systémů. Přibližně řečeno, o emergenci hovoříme, pokud systém jako celek vykazuje některé vlastnosti a vzorce chování, které nelze vyvodit z vlastností jeho jednotlivých komponent. Emergentní vlastnosti vznikají interakcí entit, které systém tvoří. Někdy se hovoří o pravidelnostech, které vznikají tzv. spontánním uspořádáním. Podrobnější definice emergence uvádí např. [19] nebo [22], většina definic zmiňuje následující znaky emergentních vlastností:

- *autonomie*: jedná se o vlastnosti, které vznikají ve složitých systémech, ve kterých dochází ke konfrontaci autonomních entit.
- *neschopnost predikce*: vlastnosti nelze předem vyvodit z vlastností částí systému.

---

<sup>5</sup> První a dosud jedinou prací knižního rozsahu v českém jazyce, která se tímto tématem zabývá, je (podle mně dostupných informací) teprve nedávno vyšlý titul [3].

- *nezmenšitelnost*: vlastnosti hovoří o chování systému jako celku a lze je sledovat jen pro systémy o dostatečné velikosti, tj. při velkém počtu entit tvořících systém. Při malém počtu entit se vlastnosti neprojeví.

Četné příklady emergence lze najít v běžném životě. Často zmiňovaným příkladem je mraveniště. Ačkoli jednotliví mravenci pouze individuálně reagují na chemické podněty v podobě pachů, z hlediska systému dochází k projevům chování, které připomíná racionální centralizovanou koordinaci činností. Emergentní vlastností z fyzikální oblasti je například barva. Elementární částice nemají žádnou barvu, tu je možné pozorovat až ve chvíli, kdy se elementární částice uskupí do větších struktur.

Jednou z hlavních motivací pro počítačovou reprezentaci multiagentních systémů je právě snaha v laboratorním prostředí zkoumat a pochopit některé emergentní jevy, ke kterým v praxi dochází (entity, které tvoří multiagentní systém, jsou označovány jako agenti<sup>6</sup>). Vzhledem k tomu, že emergentní jevy není zpravidla možné efektivně podchytit z čistě teoretické roviny, stávají se simulace s multiagentními systémy v těchto situacích důležitým analytickým nástrojem. Velmi často se přitom jedná o jevy socioekonomické, jednotliví agenti pak představují nezávisle se rozhodující jednotlivce, firmy, státní organizace nebo jiné instituce. Jedním z průkopníků v této oblasti se stal Thomas Schelling, pozdější držitel Nobelovy ceny za ekonomii, který ve svém článku [14] studoval pomocí jednoduchého multiagentního modelu přítomnost rasové segregace, ke které dochází i v rasově tolerantních komunitách. Od té doby byly multiagentní systémy použity pro analýzu celé řady nejrůznějších problémů, s cílem buď rozšířit teoretické znalosti (viz např. [7] nebo [12], které spadají do oblasti finančních trhů, kde je nasazení multiagentních systémů velmi časté), nebo pomoci při konkrétních rozhodovacích problémech (viz např. [21], kde je využívána multiagentní simulace pro stanovení optimálních únikových cest ve vytižených budovách). Multiagentních přístupů je využíváno dále i v oblastech, kde nejde o to, emergentní jevy analyzovat, nýbrž jen napodobit – jako tomu je ve filmovém průmyslu a počítačových hrách.

K velkému rozmachu studia multiagentních systémů došlo v devadesátých letech minulého století, kdy přišly první pokusy využít tyto systémy pro řešení složitých úloh v softwarovém inženýrství. Následoval rozsáhlý výzkum v oblasti znalostního inženýrství, která se nazývá **distribovaná umělé inteligence**. Distribuovaná umělá inteligence se zabývá tím, jak využít součinnosti několika autonomních výpočetních systémů (agentů) při řešení konkrétního problému. Problém je nejprve dekomponován na dílčí podproblémy a ty jsou

---

<sup>6</sup> Slovo agent v tomto kontextu se často skloňuje jako podstatné jméno neživotné (podobně jako např. robot, viz např. [9]). Jak bude vysvětleno níže, většina dnes používané terminologie pochází z oblasti softwarového inženýrství, kde se pod označením agent v multiagentním systému skrývá zpravidla program, který má za úkol pouze plnit určitou výpočetní funkci, lze si ho tedy v hmotné podobě představit nejlépe jako nějaký (neživotný) stroj. Oproti tomu v ekonomických aplikacích je agent počítačovým odrazem nějakého živoucího subjektu, člověka nebo skupiny lidí. Životný tvar slova agent zní v tomto případě poněkud přirozeněji, přiklonil jsem se proto k němu.

poté přiřazeny jednotlivým agentům, kteří se je snaží vyřešit a složit dohromady výsledné řešení. Důraz je přitom kladen na komunikaci mezi agenty a koordinaci jejich činností. Vzhledem k náročnosti této oblasti, která kombinuje nejrozličnější disciplíny umělé inteligence, bylo potřeba přesně definovat a klasifikovat různé aspekty této problematiky. Terminologie, která se dnes standardně používá v souvislosti s multiagentními systémy, pochází proto právě odsud. Je však nasnadě, že je třeba dosti odlišné úrovně umělé inteligence proto, abychom stvořili společenství agentů, kteří vykazují známky tvůrčí činnosti a efektivní organizace týmové práce při hledání způsobu řešení náročných výpočetních úloh, než v případě, že nám jde o to, simulovat tržní interakci většího množství nekooperujících ekonomických subjektů s dobře definovanými rozhodovacími problémy (což je situace v drtivé většině dosud provedených experimentů popisujících socioekonomické jevy). Je tedy běžné, že multiagentní modely popisované v ekonomické literatuře nesplňují všechny vlastnosti multiagentních systémů, které požadují standardně používané definice (viz např. [3]).

Multiagentní systém je nejčastěji definován jednoduše jako dynamický systém, ve kterém dochází k interakci autonomních agentů. Kromě autonomie agenta, tj. možnosti řídit své akce individuálně bez zásahu zvnějšku, se dále zpravidla od agenta očekávají všechny nebo některé z následujících vlastností. Agent by měl být<sup>7</sup>:

- *racionální*: jedná tak, aby dosáhl svých cílů.
- *reaktivní*: vnímá okolí a reaguje na jeho změny.
- *sociální*: komunikuje s ostatními agenty.
- *mobilní*: je umístěn v rámci svého prostředí a může umístění případně měnit.
- *proaktivní*: sám se iniciativně pouští do řešení problémů.
- *korektní*: neposkytuje o sobě vědomě nepravdivé informace.
- *benevolentní*: snaží se udělat to, o co je žádán.

Další důležitou vlastností, která se někdy objevuje v definicích multiagentních systémů, je omezená informace ze strany agentů. Často se hovoří o tzv. lokálních znalostech – žádný agent nemá kompletní přehled o celém systému, nýbrž jen o svém (blízkém) okolí.

Pro zajištění uvedených vlastností přichází ke slovu nejrozličnější techniky z repertoáru umělé inteligence: expertní systémy, umělé neuronové sítě, strukturální a příznakové metody rozpoznávání, genetické algoritmy a další. Je zřejmé, že pro některé aplikace mohou být některé z uvedených vlastností irelevantní nebo dokonce nežádoucí. Například v simulacích ekonomických antagonistických konfliktů nás může vysloveně zajímat dopad nekorektního jednání a jednání benevolentní od ekonomických subjektů zpravidla rovněž čekat nemůžeme. Dále je nasnadě, že čím více vlastnostmi se pokusíme agenta vybavit, tím složitější bude model jeho uvažování, což zvyšuje jednak náročnost implementace systému a jednak případný čas potřebný na jeho provozování. Je proto vždy třeba zvážit, jak propracovaný by

---

<sup>7</sup> Přejato z [10].

měl model agentova uvažování být v kontextu plánovaného použití multiagentního systému. Jak uvádí [9], můžeme podle stupně racionality a složitosti rozhodovacích komponent agenta klasifikovat jedním ze čtyř typů:

- *reaktivní*: jedná se o nejjednodušší typ agenta, který pouze reaguje na vnější podněty některým z předem definovaných druhů chování, neuchovává si vnitřní reprezentaci světa a nevyužívá historických údajů pro tvorbu budoucích plánů.
- *deliberativní* (též *intencionální*): tento typ agenta si již na základě zkušeností vytváří symbolickou reprezentaci světa a využívá jí k tvorbě budoucích plánů, zvažuje způsoby, kterak naplnit své cíle.
- *sociální*: sociální agent se vyznačuje vysokou mírou komunikace s ostatními agenty. Ačkoli jistou formu komunikace může praktikovat již i reaktivní agent, v případě sociálního agenta jde o formu komunikace ve vyšším jazyce, která může např. umožnit vzájemné sdílení informací o cílech a plánech agentů a sloužit tak ke kooperaci a koordinaci v rámci systému. Sociální agent má rovněž k dispozici modely chování ostatních agentů, které umožňují předvídat jejich reakce.
- *hybridní*: tento agent může vykazovat vlastnosti různých typů podle nastalé situace.

Vzhledem k různorodosti multiagentních systémů je takřka nemožné sestavit univerzální softwarové nástroje pro jejich implementaci. Snahy sjednotit postupy při návrhu multiagentních systémů se týkají především dílčích oblastí – existují např. některé široce využívané standardizované formáty a protokoly pro komunikaci mezi agenty, jako jsou KQML (*Knowledge Query Manipulation Language*) nebo ACL (*Agent Communication Language*). Často se však i komunikační procesy programují individuálně s ohledem na konkrétní použití.

Pro účely multiagentní simulace existuje řada softwarových balíčků, komerčních i zdarma dostupných, které se lze přizpůsobit na řadu různých problémů z konkrétní (často poměrně úzké) aplikační oblasti, např. AnyLogic nebo SimWalk (viz [21], [23]). Bohužel jsem nenalezl žádný existující software, který by šlo s výhodou použít pro problém zkoumaný v této kapitole, následující simulační modely bylo tedy nutno celé naprogramovat. Jako platformu pro simulační modely jsem opět zvolil prostředí a programovací jazyk Matlab. Stejně jako v případě algoritmů z minulé kapitoly, všechny spustitelné funkce jsou součástí příloženého CD.

## 4.2 Multiagentní model sériového řetězce

Při modelování sériového řetězce pomocí multiagentního systému budeme vycházet z experimentů z článku [2], které byly popsány v úvodu kapitoly. Jednotliví agenti představují články řetězce a jejich cílem je maximalizace individuálního zisku. Jedinou akcí, kterou může agent výši svého zisku ovlivnit, je určení výše své výstupní ceny. Časový vývoj systému bude

probíhat v kolech představujících po sobě jdoucí období. Na začátku každého období rozhodne o své výstupní ceně nejprve agent na začátku řetězce (tj. článek 1) a oznámí tuto cenu agentovi představujícímu navazující článek. Tento agent se na základě informace o ceně vstupu rovněž rozhodne o výši jím požadované výstupní ceny, oznámí tuto cenu článku navazujícímu a celý proces pokračuje až k poslednímu článku řetězce. Z jeho výstupní ceny je posléze konfrontací s poptávkovou funkcí určeno poptávané množství (které se dá na vědomí všem agentům). Pomocí počítačové simulace s vytvořeným modelem budeme zkoumat, zda i v delších dodavatelských řetězcích budou postupem času ceny konvergovat k hodnotám, které jsme v předchozím textu určili jako ceny rovnovážné (jako tomu bylo v experimentech se studenty v roli agentů v dodavatelských řetězcích o dvou článcích v citovaném článku), a případně jaký má vliv délka řetězce na rychlost konvergence.

Při počítačové implementaci simulačního modelu jsem využil objektového přístupu, každý agent je instancí třídy *agent*, jejíž metody se nachází ve funkcích v adresáři *@agent* na přiloženém CD. Tyto funkce obsahují i model uvažování agentů, který popisuje pododdíl 4.2.1. Funkcí, která v prostředí Matlab vytvoří podle zadaných parametrů model sériového řetězce a spustí simulační běh, je funkce *simul*. Podobně jako v případě numerických algoritmů z oddílu 3.1, simulační model lze využít pro řetězce s nejrůznějšími spojitými nákladovými a poptávkovými funkcemi<sup>8</sup>. Pokud si vystačíme s funkcemi definovanými v oddílu 3.2 lze k zadání parametrů a spuštění funkce opět využít připraveného uživatelského rozhraní (viz opět příloha P.1).

#### 4.2.1 Model racionálního uvažování agentů

Při tvorbě modelu racionálního uvažování jsem použil přístup, který je nejbližší nejjednoduššímu z uvedených typů agenta, tj. agentovi reaktivnímu. Jak uvidíme, vykazuje však agent i známky deliberativního chování, neboť si drží jistou reprezentaci vnějšího světa v podobě představy o individuální poptávkové funkci, kterou aktualizuje na základě předchozích zkušeností.

Jak již bylo řečeno, jedinou akcí, kterou se agent v modelu projevuje navenek, je oznámení své výstupní ceny. Model racionálního uvažování má tedy jediný úkol – zajistit, aby volba této ceny vedla k maximalizaci individuálního zisku článku. V okamžiku rozhodování má přitom agent k dispozici následující údaje:

- znalost vstupní ceny (tj. ceny stanovené předchozím článkem v daném kole),
- znalost své nákladové funkce,
- představa o individuální funkci poptávky po svém výstupu.

---

<sup>8</sup> Striktně vzato, poptávková funkce být spojitá nemusí, neboť z hlediska použitých optimalizačních algoritmů je třeba, aby byla spojitá pouze agentova *představa* o poptávkové funkci, což je zaručeno modelem uvažování agenta. Při použití poptávkové funkce se skoky bychom se ovšem dočkat velmi nestabilního chování agentů.

Žádný článek nezná funkci poptávky po koncové produkci (tj. ani článek koncový, což je drobný rozdíl oproti experimentům v [2]) ani náklady ostatních článků. V souladu s tím, jak byly popsány předpoklady modelu vícenásobné marginalizace v první kapitole, chovají se články tak, jako by nebyly součástí dodavatelského řetězce, ale samostatnou firmou na trhu s koncovým zbožím. Porovnávají tedy celkové náklady (které tvoří součet nákupní ceny v daném kole a výrobních nákladů) a celkové příjmy, přičemž budoucí příjmy odhadují na základě své aktuální představy o poptávkové funkci. Při známých vstupních údajích je pak nalezení výstupní ceny jednoduše úloha maximalizace funkce o jedné proměnné.

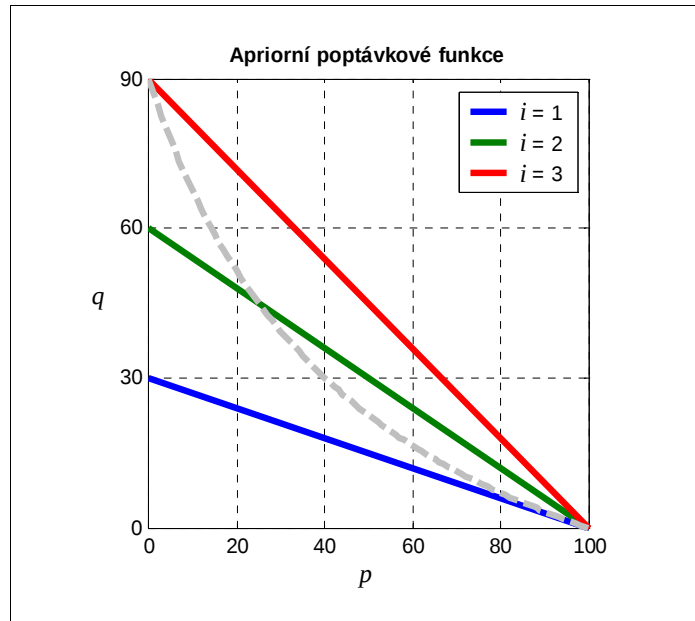
Pro dynamiku systému je klíčový především způsob, jakým agenti přizpůsobují své představy o poptávkové funkci na základě získaných hodnot. Ačkoli simulovaný řetězec může mít podle vstupních parametrů různé nelineární tvary poptávkových funkcí (a tím spíše jsou pak nelineární tvary individuální poptávky po produkci článků na začátku řetězce), pracuje model uvažování agenta pro jednoduchost pouze s lineárními představami o poptávkové funkci. To zpravidla není tak zkreslující, jak by se mohlo na první pohled zdát, neboť články nebudou lineární funkcí nahrazovat celý průběh poptávkové křivky, nýbrž jen tu část, ve které se právě nachází, mají totiž k dispozici jen znalosti minulých hodnot, které bývají vždy lokální povahy. O apriorní představy agentů na začátku simulace se stará funkce *simul*, která model řetězce vytváří, zadávají se pomocí vstupních parametrů  $p_{max}$  a  $q_{max}$ , což jsou vektory o  $n$  složkách (kde  $n$  je počet agentů), jejichž  $i$ -té složky představují průsečík funkce předpokládané  $i$ -tým agentem s osami  $p$  a  $q$ . Při použití uživatelského rozhraní je průsečík s osou  $p$  pro všechny agenty shodný se stejným průsečíkem skutečné křivky poptávky po koncové produkci a průsečík s osou  $q$  je pro  $i$ -tého agenta dán výrazem

$$\frac{i}{n} \cdot q_{max} \cdot (from + r \cdot (to - from)), \quad (4.1)$$

kde  $q_{max}$  představuje poptávané množství koncové produkce při nulové ceně (průsečík skutečné poptávky po koncové produkci s osou  $q$ )  $to$  a  $from$  jsou hodnoty stejnojmenných parametrů pro *Slope estimate* z uživatelského rozhraní a  $r$  je náhodné číslo (tj. nezávislá hodnota náhodné veličiny s rovnoměrným rozdělením na intervalu  $(0;1)$ ). Pomocí parametrů  $to$  a  $from$  je tedy možné vnést náhodný vliv do počátečních podmínek modelu a získávat různé výsledky při opětovném spuštění. Nastavení parametrů  $to$  a  $from$  na hodnotu 1 vyřadí náhodný vliv a prvotní představy agentů pak vypadají podobně jako v příkladu, který zobrazuje graf 4-1. V grafu je šedou čárkovanou čarou naznačena funkce poptávky po koncové produkci, plnými barevnými čarami jsou do grafu zaneseny apriorní představy agentů o jejich individuálních poptávkových křivkách. Na tomto příkladu se třemi články vidíme, že pro jednotlivé články se představa o individuální poptávkové křivce u článků směrem k počátku řetězce naklápí okolo horizontálního průsečíku dolů (tj. stává se plošší). To je v souladu s jevem, ke kterému obecně v decentralizovaných řetězcích dochází (tento jev je mj. důsledkem tvaru cenových reakčních křivek, které popisoval oddíl 3.1).

Logika, podle které agent aktualizuje své představy podle výsledků z minulých kol, vychází z metody částečného přizpůsobení. Pokud kombinace výstupní ceny v minulém kole a výsledného poptávaného množství neleží přesně na předpokládané poptávkové funkci (což nastane skoro vždy), přizpůsobí agent poněkud sklon a polohu předpokládané poptávkové funkce směrem k nově naměřeným hodnotám. Přizpůsobovací mechanismus, který popíšeme v následujících odstavcích, je realizován ve funkci *pricing* (součástí této funkce je i následná volba výstupní ceny).

**Graf 4-1: Apriorní představy agentů o individuálních poptávkových funkcích**



Vzhledem k tomu, že agentem uvažované poptávkové funkce jsou lineární, budeme se při jejich popisu držet značení (2.1) zavedeného pro lineární model. Parametry poptávkové funkce, kterou agent předpokládá v  $t$ -tém kole, tedy označíme jako  $a_t$  a  $b_t$ . V daném čase  $t$  se nejprve proloží přímkou posledními dvěma pozorováními, tj. dvojicemi cena-množství, parametry přímky označíme analogicky jako  $\bar{a}_{t-1}$  a  $\bar{b}_{t-1}$ . Hodnoty  $a_t$  a  $b_t$  jsou pak získány jako konvexní kombinace předchozích představ o poptávkové funkci a údajů získaných z předchozích dvou pozorování:

$$\begin{aligned} a_t &= 0,3\kappa \cdot \bar{a}_{t-1} + (1 - 0,3\kappa) \cdot a_{t-1}, \\ b_t &= 0,3\kappa \cdot \bar{b}_{t-1} + (1 - 0,3\kappa) \cdot b_{t-1}, \end{aligned} \quad (4.2)$$

kde  $\kappa$  je reálné číslo z intervalu  $[0; 1]$ , které představuje souhrn celkem tří korekčních faktorů. Tyto faktory jsou navrženy tak, aby odrážely několik „rozumných“ principů při posouzení významu získaných parametrů  $\bar{a}_{t-1}$  a  $\bar{b}_{t-1}$ . Jednotlivé korekční faktory, ze kterých je koeficient  $\kappa$  vypočítán, jsou:

- $\kappa_1$ , který zohledňuje objem znalostí agenta.
- $\kappa_2$ , který posuzuje důvěryhodnost sklonu  $\bar{b}_{t-1}$ .

- $\kappa_3$ , který posuzuje míru inovace znalostí v posledních dvou pozorováních.

Z těchto faktorů se vyjádří výsledný koeficient  $\kappa$  podle vztahu

$$\kappa = \min\{1, \kappa_1 \cdot \kappa_2 \cdot \kappa_3\}.$$

Při určení hodnoty faktoru  $\kappa_1$  se předpokládá, že objem znalostí agenta je funkcí času. Cílem zahrnutí faktoru  $\kappa_1$  je zvětšit poněkud význam nových pozorování v několika prvních kolech simulace, kdy představy agentů o poptávkové funkci mohou být značně zatíženy jejich apriorními předpoklady. Hodnota  $\kappa_1$  v čase  $t$  je dána předpisem

$$\kappa_1 = 1 + \sqrt{\frac{3}{t}}.$$

V čase  $t = 1$  má  $\kappa_1$  hodnotu přibližně 2,7 a v dalších kolech klesá směrem k hodnotě 1.

Sklon poptávkové křivky, kterým se zabývá  $\kappa_2$ , hraje velkou roli při určování výstupní ceny, jeho odhad pomocí hodnoty  $\bar{b}_{t-1}$  se ovšem může velmi snadno silně vychýlit díky změně chování následujících článků. Pokud například agent v předchozím kole svou výstupní cenu vůbec nezmění, ale koncový článek z nějakého důvodu ano, dojde k tomu, že měřený sklon je vertikální. Je vhodné, aby článek poznal, že vývoj v posledních dvou pozorováních je důsledkem nikoli tvaru jeho individuální poptávkové funkce, nýbrž jejího posunu (způsobeného patrně změnou představ ostatních článků), a vyhnul se aktualizaci představ o sklonu poptávkové křivky. Faktor  $\kappa_2$  zajišťuje, že agenti jsou s přizpůsobováním představ o poptávkové funkci obezřetní v případě, že sledovaný sklon  $\bar{b}_{t-1}$  se výrazně liší od jejich současných představ. Současně je do jeho výpočtu zakódováno apriorní přesvědčení, že sklon poptávkové křivky musí být záporný. Pokud bude hodnota  $\bar{b}_{t-1}$  záporná (tj. sklon přímky bude kladný), nebude agent přímce proložené posledními dvěma pozorováními přikládat žádný význam. Celý vzorec pro výpočet  $\kappa_2$  má podobu

$$\begin{aligned} \kappa_2 &= \min\left\{\sqrt{\frac{\bar{b}_{t-1}}{\bar{b}_{t-1}}}, \sqrt{\frac{\bar{b}_{t-1}}{\bar{b}_{t-1}}}\right\}, & \text{pokud } b > 0, \\ &= 0 & \text{jinak.} \end{aligned}$$

Pokud se sledovaný sklon od předpokládaného hodně liší, je menší z výrazů ve složené závorce blízký nule a představy agenta jsou ovlivněny jen málo.

Faktor  $\kappa_3$  zabraňuje nepříznivému jevu, ke kterému dochází zejména v situaci, kdy se agenti začínají pohybovat v blízkosti rovnovážných cen. Změny cen agenta se jsou velmi malé a může proto snadno docházet k situaci podobné té, která byl popsána v předchozím odstavci. Obecně vzato je přirozené přisuzovat větší význam parametrům  $\bar{a}_{t-1}$  a  $\bar{b}_{t-1}$ , které vznikly z pozorování (tj. bodů, dvojic cena-množství), které u sebe nejsou příliš blízko. Jsou-li body daleko od sebe, dávají lépe tušit vývoj poptávky a navíc není výsledný sklon tolik ovlivněn drobnými změnami v chování následujících článků. Při výpočtu  $\kappa_3$  je proto

poměřována euklidovská vzdálenost obou bodů v rovině s variabilitou ostatních pozorování. Výpočet probíhá podle následujícího předpisu:

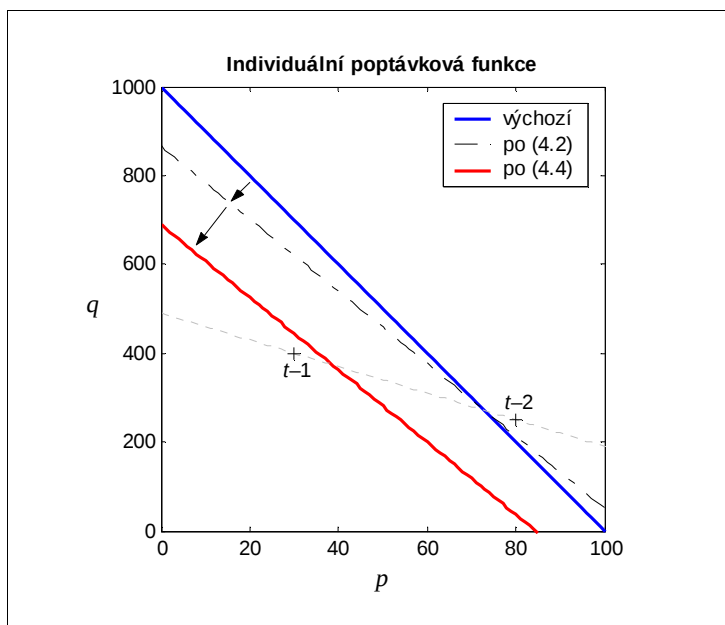
$$\kappa_3 = 4 \sqrt{\frac{(p_{t-1} - p_{t-2})^2 + (q_{t-1} - q_{t-2})^2}{\sigma^2(\mathbf{p}) + \sigma^2(\mathbf{q})}}, \quad (4.3)$$

kde  $\mathbf{p}$  je vektor se složkami  $(p_1, p_2, \dots, p_{t-1})$ , které zachycují ceny volené agentem v jednotlivých kolech, a  $\mathbf{q}$  je vektor se složkami  $(q_1, q_2, \dots, q_{t-1})$ , které představují poptávaná množství, a  $\sigma^2(\mathbf{p})$ , resp.  $\sigma^2(\mathbf{q})$  je rozptyl složek vektoru  $\mathbf{p}$ , resp.  $\mathbf{q}$ .

Z předchozího textu vyplývá, že korekce  $\kappa$  zabraňuje tomu, aby došlo k přecenění vypovídací schopnosti spojnice posledních dvou pozorování. Mohlo by se tak ale stát, že se z přizpůsobovacího mechanismu zcela vynechá poslední pozorování, což by z pohledu agenta neodpovídalo racionálnímu uvažování. Po výpočtu parametrů  $a_t$  a  $b_t$  podle (4.2) je proto ještě upraven parametr  $a_t$  (který určuje polohu poptávkové křivky) tak, aby zohlednil polohu bodu posledního pozorování. Postupuje se při tom tak, že se přímka poptávkových očekávání vertikálně posune směrem k poslednímu pozorování o 80 procent jejich původní vzdálenosti. Označíme-li  $\tilde{a}_t$  takovou hodnotu prvního parametru poptávkové křivky, která zajistí požadované posunutí, vypadá výpočet  $\tilde{a}_t$  následovně:

$$\tilde{a}_t = 0,8 \cdot (q_{t-1} + b_t \cdot p_{t-1}) + 0,2 \cdot a_t. \quad (4.4)$$

**Graf 4-2: Průběh procesu částečného přizpůsobení představ o poptávce**



Graf 4-2 ilustruje průběh přizpůsobovacího procesu. Modrá přímka představuje poptávku, kterou agent očekával v čase  $t-1$ . Dva body, označené  $t-1$  a  $t-2$ , zachycují poslední dvě kombinace cena-množství, šedou přerušovanou čarou je naznačena jejich

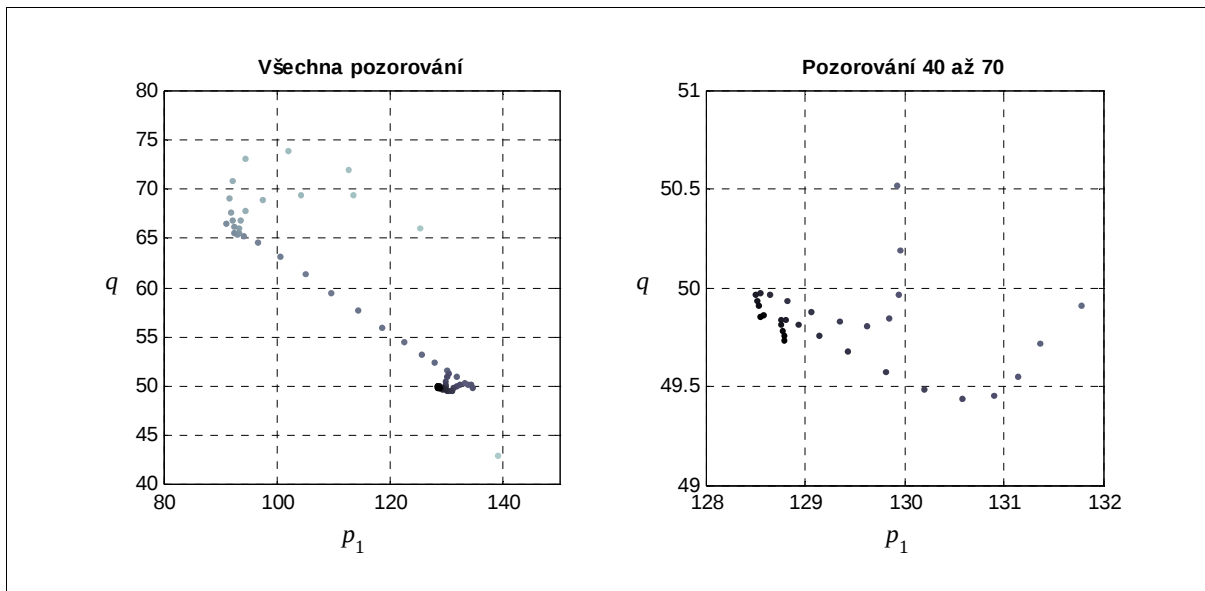
spojnice (která má parametry  $\bar{a}_{t-1}$  a  $\bar{b}_{t-1}$ ). Černá čerchovaná přímka znázorňuje funkci s parametry  $a_t$  a  $b_t$  vypočtenými podle (4.2) a konečně červená přímka je výslednou představou poptávkové funkce, která má s čerchovanou přímkou společný parametr  $b_t$  (je tedy rovnoběžná), ale namísto parametru  $a_t$  má  $\tilde{a}_t$  spočtený podle (4.4). Z grafu je patrné, že sklon poptávkové přímky se změnil méně, než její poloha. To není nikterak překvapivé, vzhledem k tomu, že koeficient přizpůsobení (tj. váha nového pozorování v konvexní kombinaci) je ve vztahu (4.4), který se věnuje právě posunu poptávkové křivky, roven 0,8; oproti tomu v předpisu (4.2), který určuje změnu sklonu, je tato hodnota pouze 0,3, případně ještě snižena korekčním koeficientem  $\kappa$ .

Váhy 0,8 a 0,3 a konkrétní funkční tvar výpočtu korekčního koeficientu jsem zvolil na základě výpočetních experimentů tak, aby co nejlépe odpovídaly racionálnímu uvažování. Proces nastavení výsledných parametrů proběhl následujícím způsobem: v průběhu simulace jsem si nechával vykreslit grafy podobné grafu 4-2 doplněné o bodový diagram všech pozorování a sledoval jsem, zda výsledné poptávkové funkce očekávané jednotlivými agenty odpovídají přibližně optickému úsudku. Pokud tomu tak nebylo, hodnoty nebo funkční tvary jsem vhodným způsobem pozměnil. Konečná podoba modelu racionálního uvažování dává proto dobré výsledky zejména v příkladech, které se podobají těm, na kterých jsem celý model testoval a vyvíjel. Nelze však očekávat, že popsáný model chování bude univerzálně použitelný při libovolných nastaveních parametrů dodavatelského řetězce. Cílem však v tomto případě bylo především navrhnout takový model chování, kterého půjde v konkrétních výpočetních experimentech použít pro simulaci racionálního uvažování agentů, a pro tento účel je navržený model adekvátní.

Uvedený přizpůsobovací mechanismus lze označit za poměrně jednoduchý nástroj formování agentových očekávání, neboť se v daném okamžiku se stávající agentovou představou porovnávají prakticky jen výsledky posledních dvou kol (ostatních pozorování je využito jen okrajově při výpočtu hodnoty  $\kappa_3$  v (4.3)). Při odhadu poptávkové funkce jednotlivými agenty bychom mohli rovněž přímo využívat všechny historické hodnoty a použít některé z běžných statistických metod. Narazili bychom ovšem na několik problémů. Předně si musíme uvědomit, že na počátku simulace jsou představy článků zpravidla velmi nepřesné a zejména pro články na počátku řetězce dochází k zdánlivě zcela nekonzistentním reakcím poptávaného množství na jejich výstupní cenu v důsledku toho, jak se mění rozsah informací článků následujících. Reakce na cenu se tedy stabilizují až se zpřesněním informací na konci řetězce. Pokud bychom tedy chtěli použít pro odhad poptávkové funkce všechna pozorování, tj. celé vektory  $\mathbf{p}$  a  $\mathbf{q}$ , jak byly definovány v (4.3), museli bychom nějakým způsobem tuto skutečnost zohlednit a přisuzovat starším pozorováním obecně menší význam – toho by bylo možné docílit např. stanovením časově orientovaných vah pro jednotlivá pozorování, nebo přímo zapomínáním počátečních pozorování. Tak či onak, mohli bychom v simulačním běhu snadno narazit na problém, který lze označit jako „rovnovážnou krátkozrakost“.

Podstatou problému je skutečnost, že jakmile se informace článků začínají zpřesňovat (vlivem přibývajících pozorování), začnou články volit ceny velmi blízké cenám rovnovážným. Nová pozorování jsou proto velmi blízko u sebe, vykazují tedy nedostatečnou variabilitu pro odhad tvaru poptávkové funkce. Může pak snadno stát, že dojde k velkému vychýlení sklonu očekávané poptávkové funkce. Situaci ilustruje graf 4-3.

**Graf 4-3: Reakce poptávaného množství na výstupní cenu agenta**



Graf představuje kombinace cena-množství, které zaznamenal první článek sériového řetězce se třemi články v simulačním běhu čítajícím 70 kol. Stáří pozorování je znázorněno jeho barvou: pozorování z prvních kol jsou světlá a čím je pozorování aktuálnější, tím má barvu tmavší. V grafu vlevo jsou zanesena všechna pozorování. Můžeme si všimnout, že nová pozorování se kupí přibližně okolo kombinace  $p_1 = 130$ ,  $q = 50$ . Graf vpravo ukazuje detail posledních 31 pozorování, tj. pozorování s pořadovými čísly 40 až 70. Jak vidíme, jejich variabilita je minimální (množství se mění v rozmezí cca jednoho procenta a cena v rozmezí přibližně tří procent) a jejich rozložení o tvaru poptávkové funkce mnoho nevypovídá – zachycuje spíše její posuny v důsledku drobného zpřesnění poptávkových očekávání navazujících článků. Pokud bychom při odhadu parametrů individuální poptávkové funkce vycházeli právě z těchto pozorování (ať už bychom na ně kladli důraz pomocí vah nebo rovnou vynecháním např. první poloviny pozorování), mohl by být výsledný tvar poptávkové funkce značně zkreslený.

Jak již bylo řečeno, jakmile je dokončena aktualizace představ agenta o individuální poptávkové funkci, je možné jednoduše najít cenu, která maximalizuje zisk. Ačkoli by volba takové ceny určitě byla projevem racionality, je do modelu chování agenta vnesena ještě jakási opatrnostní pojistka, která omezuje velikost změny ceny mezi po sobě jdoucími koly deseti procenty její současné hodnoty. To znamená, že pokud se poptávková očekávání změní

tak, že se cena maximalizující zisk zvedne o více než 10 procent její hodnoty z minulého kola, bude agent v daném kole volit cenu přesně o 10 procent vyšší. Jedná se tedy vlastně o opatření zajišťující plynulé zvyšování nebo snižování cen (namísto zdražování nebo zlevňování skokem). Toto opatření přichází ke slovu především v prvních kolech modelu, kde se představy agentů vyvíjejí nejbouřlivěji. Při jeho absenci docházelo občas při simulaci k tomu, že některý z agentů špatně odhadl situaci v navazujících částech řetězce a stanovil cenu tak vysokou, že to vedlo k vynulování poptávky a ukončení výroby v řetězci.

#### 4.2.2 Dynamika cen v sériovém řetězci

S výše popsaným simulačním modelem jsem provedl celou řadu experimentů. Ve většině případů začaly s postupujícím časem ceny (a tedy i individuální zisky) konvergovat směrem k rovnovážným hodnotám. V blízkosti rovnovážných hodnot se vývoj buď zcela zastavil, nebo začaly ceny nepravidelně oscilovat okolo hodnot blízkých hodnotám rovnovážným. Rychlost konvergence byla závislá na konkrétním nastavení parametrů, obecně lze však říci, že v delších řetězcích trvá dosažení rovnovážných hodnot déle, jak dokumentuje následující příklad.

**PŘÍKLAD 24:** Budeme uvažovat sériové řetězce o třech, čtyřech a pěti člancích. Všechny tři řetězce čelí poptávce po koncové produkci ve tvaru

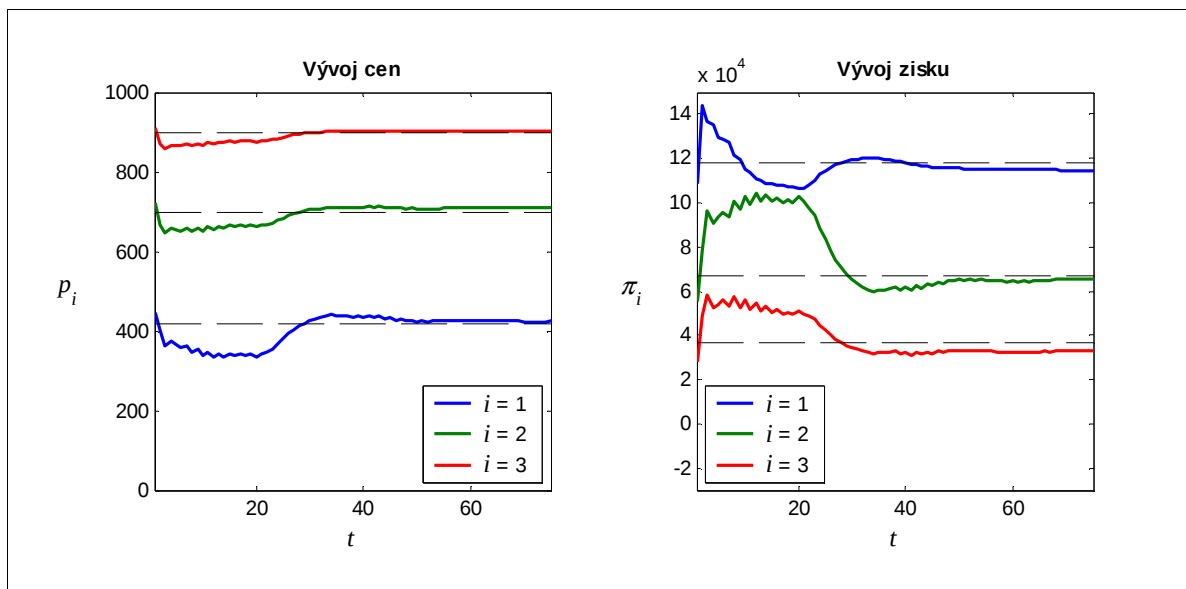
$$q = \frac{7500000}{p + 500} - 5000 \text{ .}$$

Celkové náklady jsou ve všech řetězcích 300 peněžních jednotek na kus koncové produkce a jsou rovnoměrně rozloženy na jednotlivé články (tzn. 100 peněžních jednotek na kus pro články v nejkratším řetězci, 75 v prostředním řetězci a 60 v nejdelším). Se všemi řetězci byl proveden simulační běh s tím, že apriorní poptávková očekávání byla vypočtena automaticky z uživatelského rozhraní, tj. podle (4.1), s parametry *to* a *from* nastavenými na hodnotu 1. V grafech 4-4, 4-5, a 4-6 jsou zaznamenány trajektorie, podle kterých se v jednotlivých případech vyvíjely výstupní ceny článků a výsledné individuální zisky (jedná se o grafy, které se zobrazí po spuštění simulačního běhu v okně s grafickým výstupem). Čárkovanou čarou jsou naznačeny rovnovážné hodnoty vypočítané podle algoritmů z předchozí kapitoly.

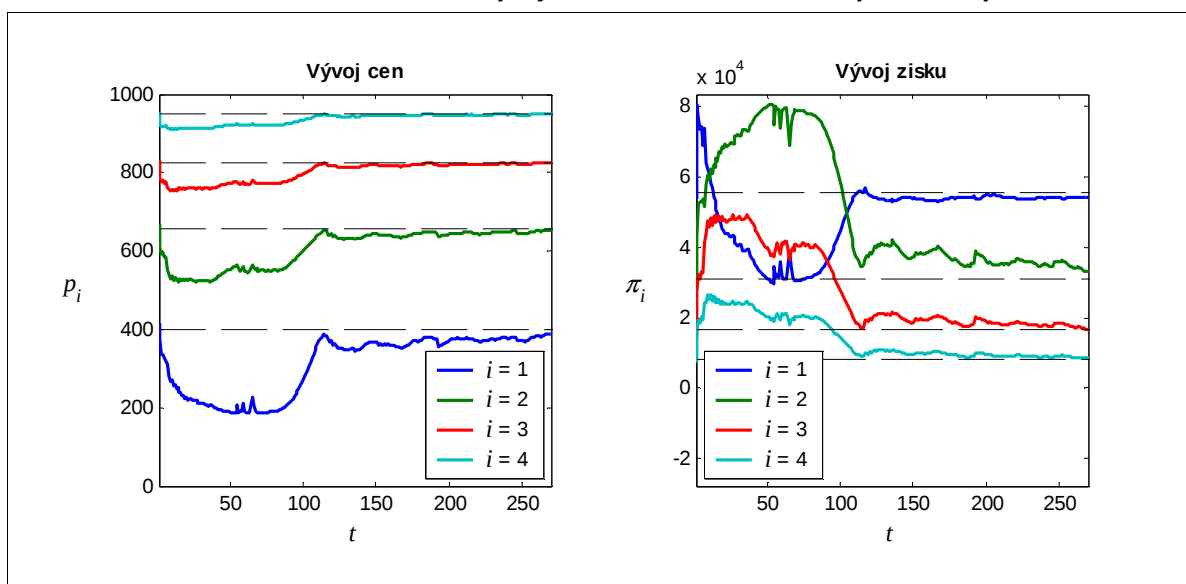
Z grafů vidíme, že ve všech třech případech došlo po určitém čase ke konvergenci k hodnotám velmi blízkým hodnotám rovnovážným. V případě řetězce se třemi články se tak stalo zhruba po třiceti časových obdobích, pro řetězec se čtyřmi články k tomu došlo po více než 100 obdobích a řetězec s pěti články se do blízkosti rovnováhy dostal až po více než 300 kolech. Tyto výsledky tedy naznačují, že délka konvergenčního procesu roste s počtem článků řetězce přibližně exponenciálním způsobem (pro tento příklad by se dala délka konvergence pro řetězec s  $n$  články přibližně vyjádřit jako  $10 \cdot 3^{n-1}$  kol). Podobné výsledky

přinesla i řada dalších experimentů, ačkoli se našly i výjimky, kde se například rychlost konvergence pro čtyři a pět článků tak výrazně nelišila.

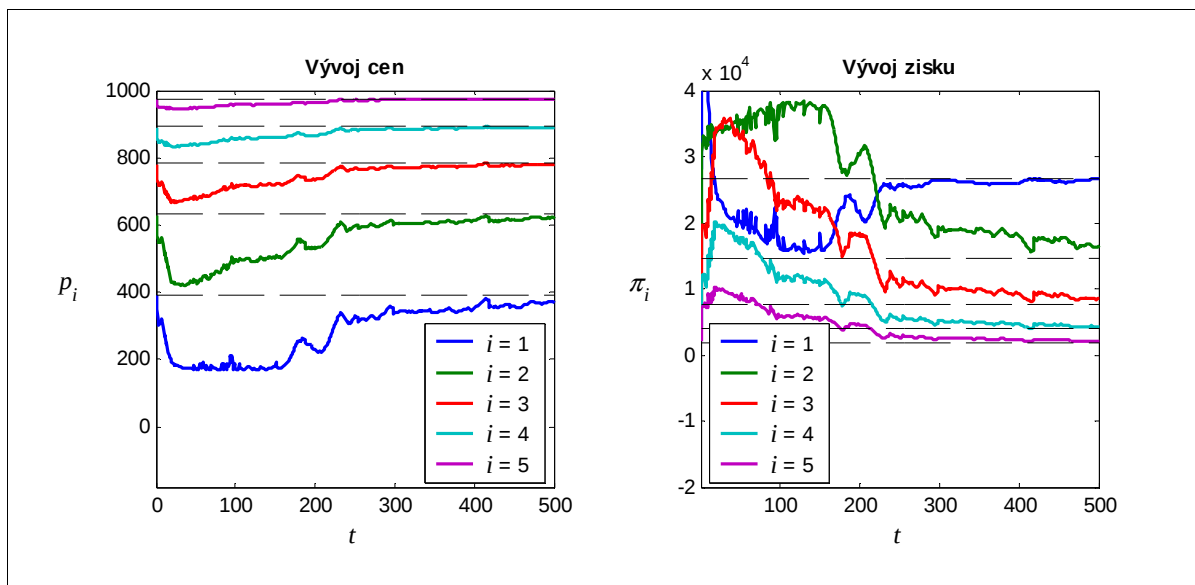
**Graf 4-4: Příklad 24 – vývoj cen a zisků v řetězci se třemi články**



**Graf 4-5: Příklad 24 – vývoj cen a zisků v řetězci se čtyřmi články**



Graf 4-6: Příklad 24 – vývoj cen a zisků v řetězci s pěti články



Výsledky simulací potvrzují, že pokud se články v sériovém řetězci s omezenou informací snaží optimalizovat svůj individuální zisk na základě postulátů základní mikroekonomické teorie firmy, dochází téměř vždy po jistém čase k nastolení rovnovážných cen popsanych v předchozích kapitolách, a to navzdory tomu, že model racionálního chování při odhadu poptávkové funkce je v implementovaném modelu značně zjednodušený. Jak uvidíme v následujícím oddílu, v případě paralelního řetězce je proces konvergence k rovnovážným cenám za použití stejných výchozích podmínek poněkud problematictější.

### 4.3 Multiagentní modely paralelních řetězců

Hlavní rozdíl oproti sériovým řetězcům způsobuje skutečnost, že články v první úrovni řetězce musí při stanovení své výstupní ceny brát v úvahu cenu stanovenou ostatními články z první úrovně řetězce (vzhledem ke komplementární povaze jejich výstupu). Jak bylo popsáno v oddílu 2.1, článek usilující o maximalizaci individuálního zisku bude dle mikroekonomické teorie volit cenu na úrovni Nashových rovnovážných strategií. V modelu s omezenou informací ovšem články nemají dostatek podkladů proto, aby mohli rovnovážné strategie určit. Pokud budou mít agenti v paralelním řetězci k dispozici pouze ty informace, které jsme předpokládali pro sériový řetězec (viz pododdíl 4.2.1), nebudou si předně ani vědomi toho, že se nachází v první úrovni řetězce a že poptávka po jejich výstupu je ovlivněna cenami ostatních článků z první úrovně; konvergence v takovém případě bude pochopitelně daleko méně přímočará, nežli v sériovém modelu. Takovou situaci se zabývá pododdíl 4.3.1, budeme hovořit o **modelu s minimální mírou informace**. V pododdílu 4.3.2 budeme předpokládat, že si články uvědomují, že je poptávka po jejich výstupu spojena s cenami ostatních článků z první úrovně, a proto si i tyto ceny zjistí (**model s rozšířenou informací**).

Oba dva modely jsou implementovány v prostředí Matlab, simulační model s minimální mírou informace spustí funkce *simulpar* a model s rozšířenými informacemi se nachází ve funkci *simulpar2*. V uživatelském rozhraní (viz příloha P.1) jsou modely rozlišeny prostřednictvím rolovacího seznamu *Chain structure*, položka *Parallel* představuje model s rozšířenými informacemi a volba *Parallel no inf.* spustí model s minimální mírou informace.

#### 4.3.1 Dynamika cen v paralelních řetězcích s minimální mírou informace

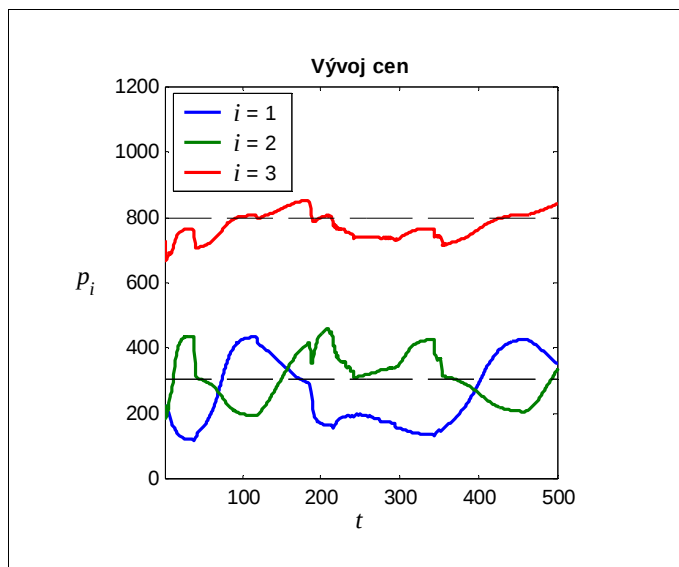
V modelu s minimální mírou informace probíhá simulace skoro stejným způsobem, jako v sériovém modelu, jedinou odlišností je fakt, že všechny články z první úrovně určují svou výstupní cenu nezávisle na začátku každého kola (podobně jako první článek v sériovém řetězci) a koncový článek určuje svou výstupní cenu až poté, co mu nahlásí své výstupní ceny všechny články z první úrovně. Model uvažování jednotlivých agentů je zcela totožný s modelem v sériovém řetězci.

Jak bylo řečeno, články první úrovně si zde jakoby neuvědomují, že poptávané množství jejich výstupu není dáno pouze jejich cenou, nýbrž také cenou výstupu ostatních článků z první úrovně. Vzhledem k tomu, že poptávkovou funkci odhaduje každý agent na základě reakce poptávaného množství na změnu své výstupní ceny, může snadno dojít ke zcela mylným závěrům ohledně tvaru poptávkové funkce v případě, že ostatní články z první úrovně své výstupní ceny mezi jednotlivými koly mění (což se v simulačním běhu děje takřka pokaždé). Články v první úrovni potom zaznamenávají zdánlivě velmi nekonzistentní vývoj kombinací cena-množství. Rozdíl oproti sériovému případu tkví v tom, že se tyto články rozhodují naprosto nezávisle (oproti navazujícím článkům v sériovém řetězci, kde funguje jednosměrný tok informace v podobě volené výstupní ceny), zatímco jejich výsledné hodnoty jsou silně provázány – se změnami cen jednotlivých článků dochází k posunu funkcí poptávky po výstupu ostatních (zatímco v sériovém modelu se funkce poptávky měnila pouze se zpřesňujícími se poptávkovými očekáváními navazujících článků). Změna, kterou bude článek s popsáním způsobem uvažování považovat za projev tvaru poptávkové funkce, může být ve skutečnosti dána jejím posunem. Jen zcela ojediněle se proto stává, že by se představy agentů o jejich individuální poptávce ustálily na nějaké konkrétní podobě. Daleko častěji dochází tomu, že se poptávková očekávání nepřetržitě mění, v důsledku čehož začnou ceny volené jednotlivými články oscilovat okolo rovnovážných hodnot, jak ilustruje následující příklad.

**PŘÍKLAD 25:** Budeme simulovat dynamiku cen v paralelním dodavatelském řetězci se třemi články, stejnou funkcí poptávky po koncové produkci jako v předchozím příkladu a produkčními funkcemi všech článků ve tvaru (3.18), čili  $tc = 800 \cdot \sqrt{q}$ . Graf 4-7 zachycuje vývoj cen v simulačním běhu dlouhém 500 časových období. Čárkovaně jsou opět vyznačeny

rovnovážné hodnoty vypočítané podle algoritmů z minulé kapitoly (přičemž články z první úrovně mají identické podmínky, a tedy i rovnovážné ceny).

**Graf 4-7: Příklad 25 – vývoj cen v paralelním řetězci s minimální informací**



Vidíme, že zde není patrná jakákoli tendence k ustálení cen – ceny neustále poměrně výrazně kmitají, za centrum této oscilace bychom mohli přibližně označit rovnovážné hodnoty. Navzdory tomu, že nedochází ke stabilizování hodnot, nám vypočtené rovnovážné ceny proto stále přinášejí jistou informační hodnotu, ačkoli nelze čekat, že právě jich bude v daný okamžik dosaženo. Další provedené experimenty naznačují, že takové chování je typické pro většinu paralelních řetězců s minimální mírou informace.

#### 4.3.2 Dynamika cen v paralelních řetězcích s rozšířenými znalostmi

V minulém pododdílu jsme ukázali, že díky provázanosti polohy individuálních poptávkových křivek a aktuálních cen v první úrovni nemohou články odhadovat tyto poptávkové křivky pouze na základě reakce poptávaného množství na svou výstupní cenu (a pokud tak činí, dochází k neustálému pohybu mylných poptávkových očekávání a následné oscilaci cen). Budeme proto nadále předpokládat, že články v první úrovni jsou si vědomi svého paralelního postavení a snaží se tudíž zapracovat ceny ostatních článků z první úrovně do svých úvah o poptávce. Pro jednoduchost budeme vycházet se situace, kdy články znají historické ceny, které požadovaly za svůj výstup ostatní články v řetězci.

Model racionálního chování článků v první úrovni bude vycházet z modelu pro sériový řetězec (který byl použit i pro paralelní řetězec s minimální informací). Rozdíl mezi oběma modely spočívá v tom, že články v paralelním řetězci s rozšířenou informací neodhadují funkci poptávky po svém individuálním výstupu, nýbrž po výstupu celé první úrovně; půjde tedy o funkční závislost mezi poptávaným množstvím a *součtem cen* všech článků

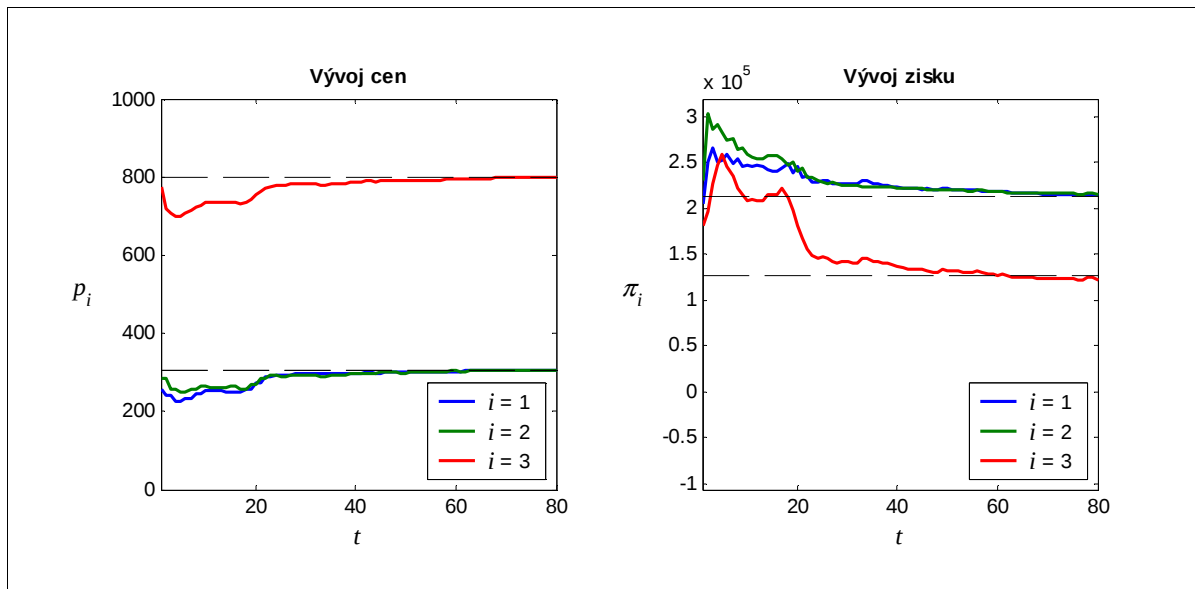
z první úrovně. Vzhledem k tomu, že každý článek zná historické ceny všech ostatních, může probíhat celý proces přizpůsobení poptávkových očekávání v podstatě stejně jako při odhadování individuální poptávky v sériovém modelu. Algoritmus, podle kterého články z první úrovně aktualizují své představy o poptávce, je obsažen ve funkci *pricingpar2* a od algoritmu popsaného v pododdílu 4.2.1 se tedy liší pouze tím, že namísto kombinací výstupní ceny a poptávaného množství uvažuje agent kombinace součtu cen první úrovně v daném čase a výsledného poptávaného množství. Pokud se agent při dané představě o poptávce coby funkci součtu cen první úrovně snaží maximalizovat svůj zisk, potřebuje znát (nebo alespoň odhadnout) ceny, které v aktuálním období ostatní články z první úrovně zvolí. Agenti rozhodující se podle funkce *pricingpar2* pro jednoduchost předpokládají, že ostatní články z první úrovně budou v aktuálním období volit stejnou cenu, jako v období předešlém.

V modelu s rozšířenými informacemi dochází oproti předchozímu modelu k poměrně přímočaré konvergenci ke stabilním hodnotám poblíž rovnovážných cen. Jak uvidíme v následujícím příkladu, je tato konvergence obecně rychlejší nežli v sériových řetězcích. Takový výsledek však není vůbec překvapivý. V sériovém řetězci bylo pro dosažení stabilních rovnovážných hodnot potřeba, aby všechny články řetězce přizpůsobily své představy o jejich individuální poptávce jejímu skutečnému tvaru (hovoříme nyní o funkci individuální poptávky odvozené za předpokladu, že navazující články optimalizují individuální zisk). Pro koncový článek řetězce je toto přizpůsobení velmi rychlé, neboť tento článek čelí přímo dané funkci poptávky po koncové produkci, která je neměnná; po několika málo obdobích získá koncový článek pomocí popsaného procesu částečného přizpůsobení velmi dobrou představu o poptávkové funkci (resp. její lokální lineární aproximaci). Pro články v hloubi řetězce trvá tento proces déle, neboť je nejprve třeba, aby se ustálily poptávkové očekávání všech navazujících článků. Tato skutečnost vede ke zmíněnému exponenciálnímu charakteru délky konvergence.

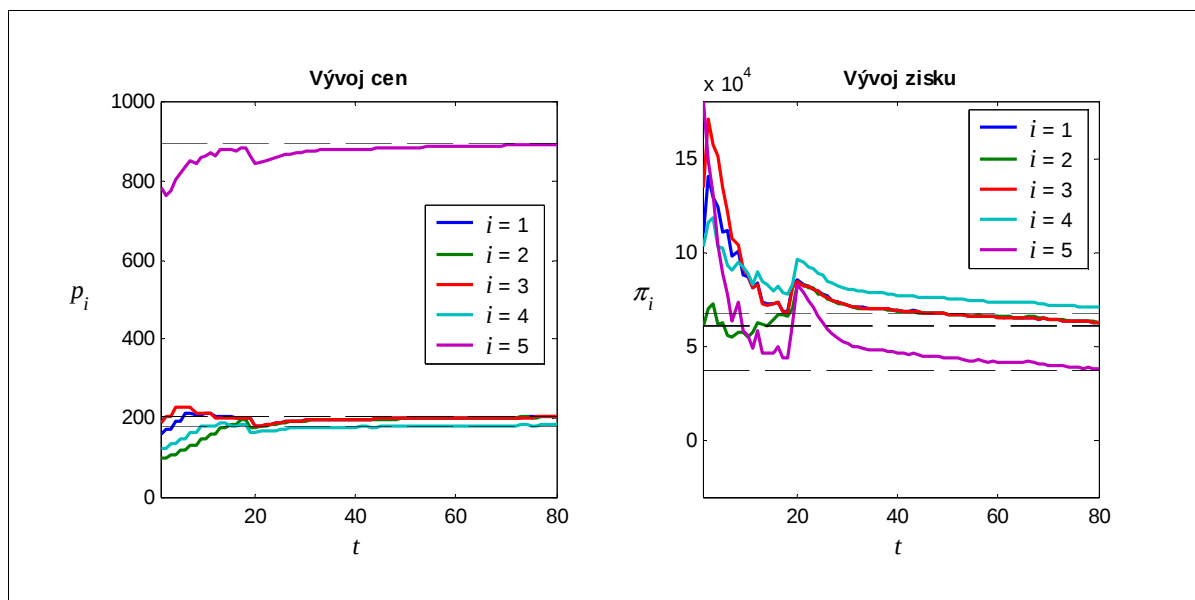
Oproti tomu v paralelním řetězci se dosažení rovnovážných cen skládá ze dvou procesů. Nejprve je třeba dosáhnout realistických odhadů poptávkových funkcí. V tomto případě ovšem jde o vývoj poptávkových očekávání jen ve dvou úrovních; podotkněme, že všechny články v první úrovni odhadují stejnou poptávkovou funkci ze stejných dat, jejich poptávkové očekávání se proto mohou lišit jen vlivem apriorních očekávání, jejichž vliv s postupem času rychle slábne. Tento proces proto probíhá poměrně rychle a nezávisí na počtu článků v paralelním řetězci. Druhým procesem je konvergence cen v první úrovni k Nashově rovnováze. V každém kole mění články své výstupní ceny tak, aby při daném součtu cen ostatních článků maximalizovaly svůj individuální zisk, automaticky se tak vlastně snaží zajistit podmínky kladené na Nashovu rovnováhu. K dosažení cen blízkých Nashově rovnováze proto dochází také poměrně rychle, navíc experimenty naznačují, že s počtem článků v první úrovni řetězce klesá rychlost tohoto procesu jen velmi mírně; dokumentuje to následující příklad.

**PŘÍKLAD 26:** Simulujme dynamiku cen ve stejném dodavatelském řetězci jako v příkladu 25, nyní však v podmínkách rozšířené informace. Graf 4-8 zaznamenává vývoj cen a individuálních zisků v osmdesáti obdobích. Vidíme, že na rozdíl od varianty s minimální mírou informace dochází ke konvergenci k rovnovážným cenám (naznačeným opět čárkovaně) a do bezprostřední blízkosti rovnovážných cen se řetězec dostává již po přibližně 25 kolech.

**Graf 4-8: Příklad 26 – vývoj cen a zisku v řetězci s rozšířenou informací (3 články)**



**Graf 4-9: Příklad 26 – vývoj cen a zisku v řetězci s rozšířenou informací (5 článků)**



Uvažujme nyní stejný dodavatelský řetězec, který se od předchozího liší pouze tím, že má v první úrovni o dva články navíc. Oba tyto články budou mít nulové výrobní náklady. V grafu 4-9 vidíme vývoj cen a zisků za osmdesát období pro tento řetězec (nové články v první úrovni mají rovnovážnou díky nulovým nákladům nižší, nežli články ostatní,

představuje ji nejnižše položená čárkovaná čára). Z grafu je patrné, že délka konvergence se zvedla jen minimálně, což potvrzuje uvedené závěry.

## 4.4 Shrnutí výsledků multiagentních simulací

Výsledky získané pomocí multiagentních modelů popisují, jakým způsobem dochází za předpokladů modelu vícenásobné marginalizace (vytyčených v oddílu 1.2) k nastolení rovnovážných cen. Celkem jsme zkoumali tři varianty multiagentního modelu: model sériového řetězce, model paralelního řetězce s minimální mírou informace a model paralelního řetězce s rozšířenou informací.

V sériových řetězcích docházelo ke konvergenci k rovnovážným cenám v drtivé většině případů. Mezi délkou řetězce a délkou konvergenčního procesu lze často vysledovat přibližně exponenciální závislost. Výskyt nekonvergujících cen se týkal především delších řetězců (5 článků a více), kde je přenos cenového impulzu prvních článků na výsledné poptávané množství zprostředkovaný několika mezilehlými články, a tudíž velmi slabý.

V případě paralelních řetězců jsme získali zcela odlišné výsledky pro varianty s různou mírou informace. Ceny v řetězcích s minimální mírou informace konvergovaly k rovnovážným cenám jen zcela ojediněle. Typicky měly ceny oscilační charakter – kmitaly periodicky okolo rovnovážných hodnot. Nezřídka se ale chovaly poněkud méně předvídatelně, v některých případech se dokonce celý cenový systém zhroutil a došlo k vynulování poptávaného množství. Oproti tomu řetězce s rozšířenou mírou informace se vyznačovaly nejčastějším výskytem konvergence cen ze všech tří zkoumaných variant. Konvergence byla navíc vesměs poměrně rychlá a s prodlužováním řetězce její délka výrazněji nerostla.

Vzhledem k tomu, že v případě konvergence se ceny v sériovém i paralelním modelu ustálily v blízkém okolí hodnot, které byly pomocí postupů z předchozí kapitoly označeny jako rovnovážné, může výstup z multiagentních simulací sloužit i jako ověření správnosti analytických výpočtů rovnovážných hodnot.

## ZÁVĚR

Cílem této práce bylo popsat a kvantifikovat efekt vícenásobné marginalizace v dodavatelských řetězcích, které se liší jednak svou strukturou a jednak prostředím, ve kterém fungují (zejména ve smyslu poptávkových a nákladových funkcí). Při základní definici modelového rámce jsem vycházel z klasických modelů zabývajících se problematikou dvojí ziskové marže – viz např. [5], [1] nebo [13]; tento modelový rámec je popsán v první kapitole. Kromě základního představení marginalizačního efektu a definice měřítka pro jeho dopad (efektivnosti řetězce) jsou zde představeny různé druhy architektur dodavatelského řetězce, kterými se práce zabývá (architektura sériová, paralelní a kombinovaná).

Právě zkoumání odlišností nebo naopak shodných znaků u různých typů architektur řetězce byla věnována poměrně velká pozornost. Je tomu tak především proto, že žádná (mně známá) stávající odborná literatura se nezabývá v podobném analytickém pojetí řetězci, které byly označovány v předchozím textu jako paralelní nebo kombinované. Pro řetězce sériové už byly některé z uvedených výsledků publikovány – nejznámější je patrně poměr mezi ziskovou marží článků sériového řetězce, který je jako obecný princip prvně objasněn v [1] (ač pouze pro řetězce se dvěma články)<sup>9</sup>.

Pro účely zkoumání různých aspektů efektu vícenásobné marginalizace jsem využil tři odlišných metodologických přístupů:

- analýza matematického modelu a následné odvození funkčních vztahů,
- výpočetní experimenty se sestavenými numerickými algoritmy,
- analýza pomocí multiagentní simulace.

První z uvedených přístupů byl nasazen ve druhé kapitole při zkoumání řetězců fungujících v lineárním prostředí. Linearita použitých funkcí umožnila odvodit obecné vzorce pro efektivnost řetězce a některé další ukazatele (např. rovnovážné ceny a zisk). Jak již bylo řečeno, pro sériové řetězce byly již některé z uvedených vztahů publikovány; odvození

---

<sup>9</sup> Některé další práce (např. [4] nebo [17]) se zabývají architekturou, která byla krátce zmíněna v oddílu 1.2 a která by se dala nazvat „oligopolní variantou“ sériového řetězce – jedná se o řetězce sériového typu, ve kterých ovšem v jednotlivých úrovních nefiguruje pouze jediný článek v monopolní pozici, nýbrž několik článků v rovnocenném postavení, které vyrábějí identický výstup ve vzájemně oligopolních podmínkách – formálně vzato se zde tedy objevují vazby typu many-to-one jako u paralelních nebo kombinovaných řetězců, funkčně jde však o zcela rozdílnou architekturu.

jednotlivých tvrzení, které zde předkládám, je ovšem původní a koncipované tak, aby vycházelo ze stejného modelového rámce a vůbec způsobu uvažování, který je použit pro analýzu řetězců s paralelní nebo kombinovanou architekturou<sup>10</sup>; analýza paralelních a kombinovaných řetězců je vesměs původní.

Ze získaných vzorců plyne celá řada poznatků o decentralizovaných řetězcích. Shrňme stručně nejdůležitější z nich (všechny hovoří o *decentralizovaných řetězcích v lineárním modelu*):

- efektivnost řetězce prudce klesá s počtem článků.
- efektivnost sériového řetězce je nižší než efektivnost řetězce paralelního se stejným počtem článků, řetězec kombinovaný se podle konkrétního tvaru pohybuje mezi oběma hraničními možnostmi.
- efektivnost řetězce vůbec nezávisí na parametrech poptávkové funkce ani výši nákladů v řetězci.
- v řetězci dochází k silným pozitivním externalitám nákladových úspor – při změně jednotkových nákladů jednoho článku dojde ke stejnému proporcionálnímu zvýšení zisku všech článků v řetězci.
- rozdělení celkového zisku řetězce mezi jednotlivé články probíhá v proporcích, které jsou určeny výhradně strukturou řetězce, resp. počtem článků v jednotlivých úrovních; není tedy závislé na rozložení nákladů podél řetězce.

Třetí kapitola využívá druhého ze zmíněných metodologických přístupů, tj. výpočetních experimentů s numerickými algoritmy. Důvodem pro použití numerických algoritmů namísto formální analýzy vztahů byl záměr zkoumat řetězce fungující v nelineárním prostředí, kde nelze odvodit obecné vzorce s dobrou vypovídací schopností. Pro jednoduchost byly zkoumány pouze řetězce sériové a paralelní (přičemž lze očekávat, že vlastnosti kombinovaných řetězců se budou podle konkrétního tvaru přiklánět k jedné z těchto variant). Pro sériové a paralelní řetězce jsem zkonstruoval algoritmy pro nalezení rovnovážných cen v nelineárním prostředí a implementoval je v prostředí Matlab, kde jsem rovněž připravil uživatelské rozhraní pro jejich snadné spuštění – vše je součástí přiloženého CD.

Připomeňme ve stručnosti výsledky výpočetních experimentů, které jsou podrobněji shrnuty v oddílu 3.3.4 (vše se opět týká decentralizovaného řetězce):

- řada poznatků z lineárního modelu je velmi dobře přenositelná na řetězce operující v nelineárním prostředí – např. vliv počtu článků nebo architektury řetězce na jeho efektivnost.

---

<sup>10</sup> Např. ve [4] dochází autoři k některým vztahům pro oligopolní sériový řetězec zmiňovaný v předchozí poznámce, ze kterých coby speciální případ opět získáme některé vzorce pro jednoduchý sériový řetězec, prostřednictvím poněkud odlišného modelového aparátu, než který byl použit v této práci (cournotovský přístup rozhodování o objemu výroby v jednotlivých úrovních řetězce namísto rozhodování o cenách).

- přítomnost konvexní poptávkové funkce nebo nákladových funkcí vykazujících úspory z rozsahu má negativní vliv na efektivnost řetězce (ve srovnání s lineárními variantami).
- proporcionální rozložení celkového zisku řetězce mezi jednotlivé články je na rozdíl od lineárního modelu ovlivněno jak parametry poptávkové funkce, tak povahou funkcí nákladových. Seřazení ziskových podílů podle velikosti ovšem ve všech uvedených experimentech odpovídalo pořadí podle lineárního modelu.
- vykazují-li náklady na výrobní proces úspory z rozsahu, lze jeho přesunem směrem ke konci řetězce zpravidla zvýšit celkový zisk (a opačně pro záporné úspory z rozsahu).

V závěru třetí kapitoly byl představen model s diskrétními cenami. Cílem zde bylo zjistit, nakolik jsou předchozí výsledky zkresleny, pokud do úvah týkajících se rozhodování o cenách zahrneme požadavek, že ceny výstupu jednotlivých článků musí být voleny z diskrétní škály. Pro účely analýzy jsem zkonstruoval a implementoval algoritmy pro nalezení rovnovážných cen v podmínkách diskrétních cen. V případě paralelního řetězce bylo třeba vyhnout se kombinatorické explozi při hledání rovnovážné kombinace cen výstupu článků z první úrovně; navržený algoritmus tento problém výrazně potlačuje.

Při výpočetních experimentech se ukazuje, že zkreslení předchozích výsledků v důsledku diskretizace cen je citelné především u sériových řetězců, kde se zejména při větším počtu článků můžeme dopracovat k hodnotám, které nemají s výsledky ze spojitého modelu takřka nic společného. V paralelních řetězcích jsou rozdíly daleko menší; na druhou stranu zpravidla existuje více rovnovážných kombinací cen výstupu článků první úrovně, a výsledné hodnoty potom nejsou zcela jednoznačné.

Poslední kapitola se zabývala otázkou konvergence cen v dodavatelských řetězcích s omezenou informací směrem k rovnovážným hodnotám. Při vymezení modelu pro dvojí (příp. vícenásobnou) ziskovou marži se předpokládá, že pokud se jednotlivé články řetězce snaží maximalizovat svůj individuální zisk bez jakékoli kooperace s ostatními, dojde k cenové spirále, která ceny uvnitř řetězce vytlačí na jejich rovnovážné hodnoty. Cílem této kapitoly bylo zkoumat, zda (a případně po jaké době) k takové situaci dojde, vycházíme-li z prostředí, kde články disponují omezenou informací o poptávkové funkci a člancích ostatních. V zásadě jde tedy o jistou verifikaci dříve získaných výsledků, resp. o ověření jejich relevance.

Použitou analytickou metodou byla v tomto případě multiagentní simulace. V textu je představen simulační model, který jsem implementoval v prostředí Matlab. Získané výsledky se velmi liší podle toho, zda se jedná o sériový nebo paralelní řetězec. V sériovém řetězci docházelo ve většině experimentů k ustálení cen v bezprostřední blízkosti rovnovážných cen, přičemž délka konvergenčního procesu rostla přibližně exponenciálním způsobem s počtem článků. V případě paralelních řetězců bylo třeba rozlišit, zda mají články v první úrovni řetězce nějakou informaci o ostatních člancích řetězce (přesněji informaci o jejich existenci

a historických cenách výstupu – model s rozšířenými znalostmi) nebo nikoli (model s minimální mírou informace).

V případě modelu s minimální mírou informace měla dynamika cen v řetězci oscilační povahu – ceny se neměly tendenci ustalovat na konkrétních hodnotách, spíše neustále kmitaly nahoru a dolů, často ovšem právě okolo rovnovážných cen. V řetězcích s rozšířenými znalostmi docházelo naopak k velmi jednoznačné konvergenci k rovnovážným cenám, jejíž délka se s počtem článků řetězce příliš neměnila.

Domnívám se, že v rámci cílů vytyčených v úvodu práce se mi podařilo za použití různých metod získat poměrně ucelené výsledky. Je však nasnadě, že zpracované téma by šlo dále rozšiřovat o další modelové přístupy. Velmi zajímavou teoretickou oblastí, do které by bylo možné směřovat další výzkum, je například problematika formování dodavatelských řetězců v podmínkách oligopolního postavení článků – koncept popsany v [4] pro sériové řetězce by se dal možná rozšířit pro paralelní a kombinované struktury řetězce. I v oblasti formování řetězců by se dalo patrně využít experimentů s multiagentním modelem.

Další pokračování výzkumu by se mohlo týkat rovněž empirické verifikace získaných výsledků. Předmětem této práce bylo zkoumání dodavatelských řetězců v teoretické rovině. Jistě by bylo zajímavé sledovat, zda (resp. nakolik) k popisovaným jevům fakticky dochází v reálné podnikové praxi – v [1] je provedena taková analýza reálných dat pro model sériového řetězce se dvěma články; podobný přístup by bylo možné použít pro řetězce s bohatší strukturou. Je ovšem zřejmé, že zásadním problémem takového výzkumu by bylo zajištění adekvátní datové základny.

## LITERATURA

- [1] BRESNAHAN, T., REISS, P.: Dealer and Manufacturer Margins. In: *Rand Journal of Economics*, 1985, Vol. 16, No. 2, pp. 253-238.
- [2] BADASYAN, N., GOEREE, J., MORGAN, J., SERVÁTKA, M. a kol.: Vertical Integration of Successive Monopolists: A Classroom Experiment. In: *Caltech Working Papers*, 2003, 18 s.
- [3] CAHLÍK, T. a kol.: *Multiagentní modely v ekonomii*. Praha: Karolinum, 2006, 155 s.
- [4] CORBETT, C., KARMAKAR, U.: Competition and Structure in Serial Supply Chains with Deterministic Demand. In: *Management Science*, 2001, Vol. 47, No. 7, pp. 966-978.
- [5] FIALA, P.: *Modelování dodavatelských řetězců*. Praha: Professional Publishing, 2005, 168 s.
- [6] GARDNER, M.: The Fantastic Combinations of John Conway's New Solitaire Game "Life". In: *Scientific American*. New York, 1970, Issue 223, pp. 120-123.
- [7] HOFFMANN, A., DELRE, S., VON EJE, J., JAGER, W.: Artificial Multi-Agent Stock Markets: Simple Strategies, Complex Outcomes. In: *Advances in Artificial Economics: The Economy as a Complex Dynamic System*. Heidelberg: Springer Verlag, 2006, pp. 167-176.
- [8] KATZ, M., ROSEN, H.: *Microeconomics*. Boston: Irwin/McGraw-Hill, 1998, 3. vydání, 656 s.
- [9] KUBÍK, A.: *Intelligentní agenty*. Brno: Computer Press, 2004, 280 s.
- [10] MAŘÍK, V., ŠTĚPÁNKOVÁ, O., LAŽANSKÝ, J. a kol.: *Umělá inteligence (2)*. Praha: Academia, 1997, 373 s.
- [11] MAÑAS, M.: *Games and economic decisions: lecture notes*. Praha: Oeconomica, 2003, 87 s.
- [12] PALMER, R., ARTHUR, B., HOLLAND, J., LEBARON, B.: An Artificial Stock Market. In: *Journal of Artificial Life and Robotics*, 1999, Vol. 3, No. 1, pp. 27-31.
- [13] RHODES, E., WARREN, J., CARTER, R.: *Supply Chains and Total Product Systems: A Reader*. Malden: Blackwell Publishing, 2006, 527 s.
- [14] SCHELLING, T.: Dynamic Models of Segregation. In: *Journal of Mathematical Sociology*. 1971, pp. 143-146.

- [15] TAYUR, S., ERHUN, F., KESKINOCAK, P.: *Spot Markets for Capacity and Supply Chain Coordination*. 2000, s. 32. Dostupný z WWW:  
<http://citeseer.ist.psu.edu/erhun00spot.html>.
- [16] VESELÝ, J.: *Matematická analýza pro učitele I,II*. Praha: MatfyzPress, 1997, 452 s.
- [17] WATERSON, M.: Vertical Integration, Variable Proportions and Oligopoly. In: *The Economic Journal*, 1982, Vol. 92, No. 365, pp. 129-144.

## INTERNETOVÉ ZDROJE:

- [18] [http://faculty.haas.berkeley.edu/rjmorgan/classroom\\_experiments.htm](http://faculty.haas.berkeley.edu/rjmorgan/classroom_experiments.htm) : Univerzitní internetová stránka jednoho z autorů článku [2], Johna Morgana, na které jsou zveřejněny některé další informace týkající se prováděných experimentů.
- [19] <http://plato.stanford.edu/entries/properties-emergent/> : definice pojmu emergence a její historický vývoj podle autorů projektu *Stanford Encyclopedia of Philosophy*.
- [20] [www.mathworks.com](http://www.mathworks.com) : domovské stránky firmy MathWorks, výrobce software Matlab, který byl použit pro veškeré výpočty v této práci. Je zde k dispozici mimo jiné nápověda k softwaru, která vysvětluje algoritmy použité ve vestavěných funkcích.
- [21] [www.simwalk.com](http://www.simwalk.com) : domovské stránky výrobce softwaru SimWalk na multiagentní simulaci prostupnosti budov a veřejných míst s velkým pěším provozem.
- [22] [www.wikipedia.org](http://www.wikipedia.org) : dnes již klasická volně přístupná internetová encyklopedie Wikipedia. Jsou zde (zatím především v anglickém jazyce) k dispozici velmi dobře zpracované informace o inteligentních agentech, multiagentních modelech a pojmu emergence).
- [23] [www.xjtek.com](http://www.xjtek.com) : domovské stránky výrobce prostředí AnyLogic pro podporu simulačních modelů ekonomických procesů (software nabízí i využití multiagentních přístupů).

## PŘÍLOHY

Následující dvě přílohy, které mají podobu oddílů *P.1* a *P.2*, se obě týkají softwarové implementace úloh z kapitol 3 a 4, která se nachází na přiloženém CD ve složce *Matlab* – podle stejnojmenného prostředí, ve kterém byly všechny funkce naprogramovány. Pro jejich spuštění je třeba mít nainstalován program Matlab verze 6.1 nebo vyšší.

Příloha *P.1* popisuje uživatelské rozhraní, které bylo připraveno pro snadné ovládání funkcí zaměřených na analýzu vícenásobné marginalizace. Uživatelské rozhraní má dvě hlavní funkce: jednak slouží k jednoduchému a přehlednému zadávání parametrů týkajících se dodavatelského řetězce pro použití s připravenými funkcemi, dále pak usnadňuje interpretaci výsledků pomocí grafických a textových výstupů. Z uživatelského rozhraní je možné spouštět jak numerické algoritmy pro nalezení rovnovážných hodnot, které byly popsány ve třetí kapitole (oddíly 3.1 a 3.4), tak i multiagentní simulace z kapitoly 4.

V příloze *P.2* jsou vypsány zdrojové kódy některých funkcí, které realizují algoritmy popsané v kapitolách 3 a 4. Tyto funkce se nachází na přiloženém CD buď přímo v adresáři *Matlab*, nebo jeho podadresáři *@agent*. Příloha *P.2* neobsahuje zdrojové kódy všech funkcí, které se v těchto adresářích nachází a které byly pro účely analýzy vícenásobné marginalizace naprogramovány. Do výpisu jsou zahrnuty jen ty funkce, které byly v předchozím textu zmíněny či podrobněji rozebrány.

Zdrojové kódy všech funkcí začínají nápovědou k dané funkci; v příkazovém okně prostředí Matlab je ji možno vyvolat příkazem „*help*“ následovaným jménem funkce. Kromě nápovědy jsou kódy opatřeny četnými poznámkami. Tyto poznámky jsou zvýrazněny zelenou barvou (v textu i v editoru prostředí Matlab) a umožňují sledovat, jakým způsobem jsou algoritmy popsané v kapitolách 3 a 4 implementovány v jazyce prostředí Matlab. Zdrojové kódy některých funkcí byly zkráceny o poměrně rozsáhlé části zabývající se grafickým výstupem, které nejsou z hlediska implementace uvedených algoritmů podstatné.

Kromě funkcí, které zajišťují vlastní výpočty nebo funkcionality uživatelského rozhraní, jsou na přiloženém CD ještě skripty v jazyce prostředí Matlab obsažené v podadresáři *Výpočetní experimenty* adresáře *Matlab*. Skripty z tohoto podadresáře zachycují všechny výpočetní experimenty ze stejnojmenného oddílu 3.3. Název každého skriptu říká, ke kterému příkladu se skript vztahuje a jaký graf se zobrazí po jeho spuštění.

## P.1 Popis uživatelského rozhraní

Uživatelské rozhraní bylo vytvořeno jakožto grafické uživatelské rozhraní (*graphical user interface*, GUI) v jazyce Matlab verze 6.1. Pro spuštění uživatelského rozhraní je nejprve třeba spustit program Matlab, nastavit (nejjednodušeji v podokně *Current Directory*) složku *Matlab* na přiloženém CD jako aktuální adresář, a poté v hlavním příkazovém podokně (*Command Window*) napsat příkaz „*start*“. Tento příkaz pouze nastaví správný formát výstupu, potlačí některé varovné hlášky prostředí Matlab, které by mohly činit výstup z programu nepřehledným, a spustí uživatelské rozhraní. Vlastní zdrojový soubor uživatelského rozhraní se nachází na přiloženém CD ve složce *Matlab* a nazývá se *DSCEquilibrium.m*.

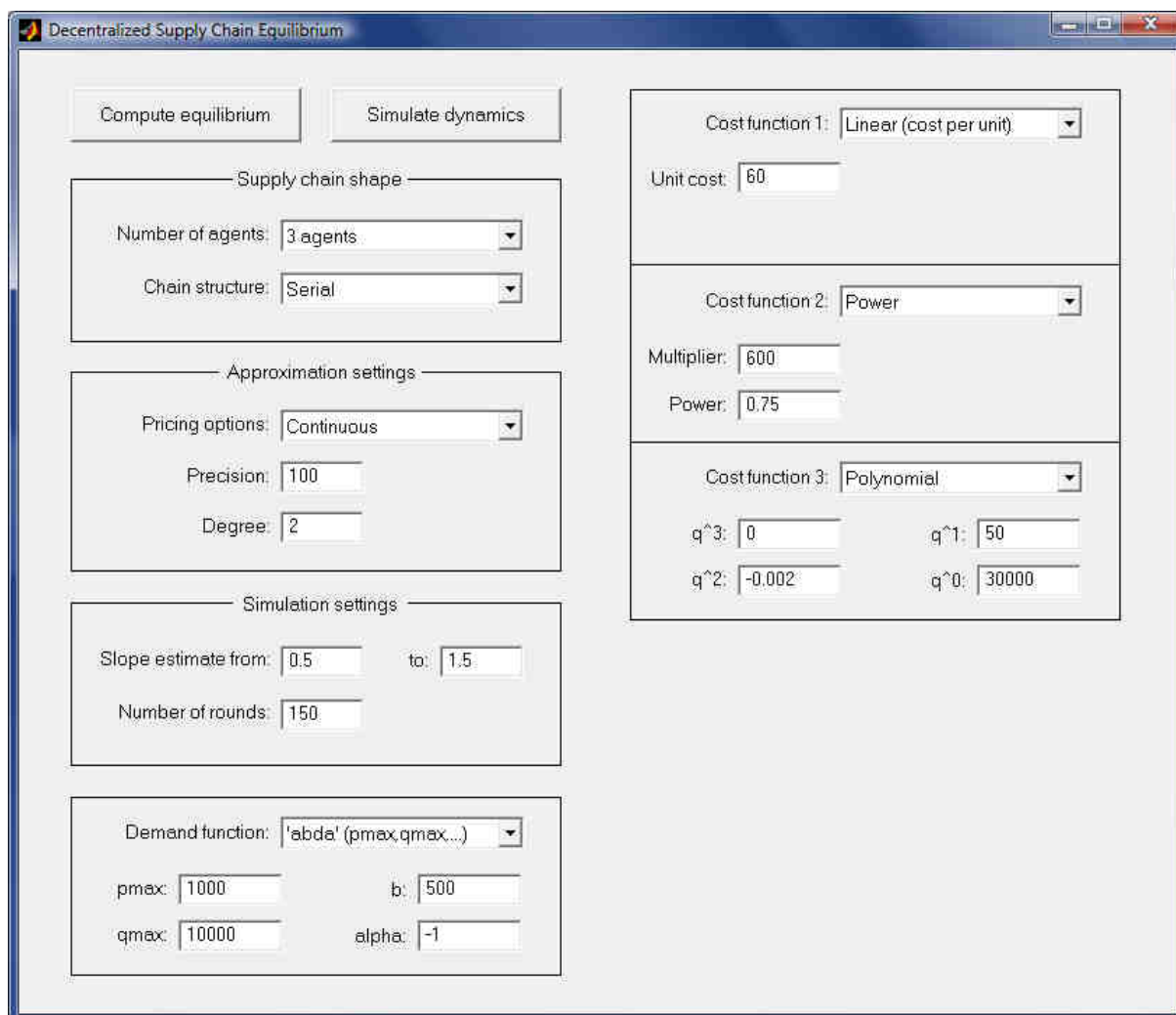
### P.1.1. Zadávání vlastností dodavatelského řetězce a parametrů výpočtu

Po spuštění příkazu *start* se na obrazovce objeví dvě okna – hlavní okno, které zachycuje obrázek P-1, sloužící k vlastnímu zadávání parametrů a ovládání funkcí, a okno s grafy, znázorněné na obrázku P-2, které zobrazuje tvar poptávkové funkce a nákladových funkcí všech článků podle aktuálních hodnot a umožňuje tak průběžnou kontrolu nastavovaných parametrů. Veškeré popisky i výstupy z programů jsou v anglickém jazyce (zejména kvůli problémům s diakritikou v prostředí Matlab).

V pravé části hlavního okna se nachází informace o nákladových funkcích jednotlivých článků. Pro větší přehlednost jsou informace týkající se jednoho článku ohraničené rámečkem, přičemž jsou vždy zobrazeny nákladové funkce jen tolika článků, kolik jich je momentálně uživatelem pro daný řetězec zvoleno. Typ nákladové funkce pro *i*-tý článek je možné volit v rolovacím seznamu nadepsaném *Cost Function i*, kde *i* je číslo článku. K dispozici jsou tři funkční tvary popsané v minulém oddílu, v seznamu jsou nazvány jako *Linear (cost per unit)* – lineární nákladová funkce, *Polynomial* – polynomiální funkce a *Power* – mocninná funkce. Při volbě jedné z možností se v rámečku zobrazí nabídka nastavovaných parametrů. Pro polynomiální funkci je možné nastavit v uživatelském prostředí polynom nanejvýš třetího stupně.

V levé části hlavního okna nahoře se nachází tlačítka pro zahájení výpočtů, numerické algoritmy pro nalezení rovnovážných hodnot se spouští tlačítkem *Compute equilibrium*, multiagentní simulace tlačítkem *Simulate dynamics*. V rámečku pod tlačítky, nazvaném *Supply chain shape*, je možné v rolovacích seznamech volit parametry týkající se tvaru řetězce, kterými jsou počet článků řetězce a struktura řetězce. Počet článků je z grafických důvodů omezen pěti, struktura řetězce může být sériová nebo paralelní. Volba *Parallel no inf.* v seznamu dostupných struktur řetězce se týká dynamického modelu s minimální mírou informace (viz pododdíl 4.3.1), pro výpočet rovnovážných hodnot je tato volba totožná s možností *Parallel*.

Obrázek P-1: Hlavní okno uživatelského rozhraní



Decentralized Supply Chain Equilibrium

Compute equilibrium Simulate dynamics

Supply chain shape

Number of agents: 3 agents

Chain structure: Serial

Approximation settings

Pricing options: Continuous

Precision: 100

Degree: 2

Simulation settings

Slope estimate from: 0.5 to: 1.5

Number of rounds: 150

Demand function: 'abda' (pmax,qmax,...)

pmax: 1000 b: 500

qmax: 10000 alpha: -1

Cost function 1: Linear (cost per unit)

Unit cost: 60

Cost function 2: Power

Multiplier: 600

Power: 0.75

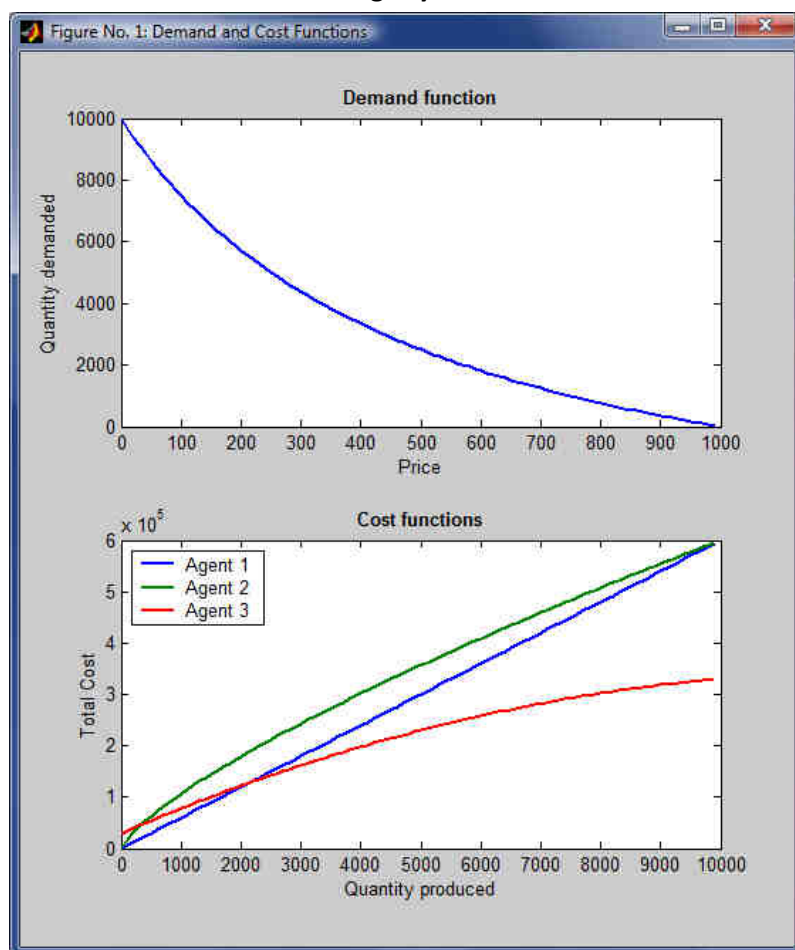
Cost function 3: Polynomial

q<sup>3</sup>: 0 q<sup>1</sup>: 50

q<sup>2</sup>: -0.002 q<sup>0</sup>: 30000

Rámeček *Approximation settings* obsahuje tři parametry, které mají vliv na výpočty při aproximacích cenových reakčních křivek jednotlivých článků během postupu, který byl popsán v oddílu 3.1. První z nich (*Pricing options*) rozhoduje, zda cenové strategie článků jsou spojité nebo diskrétní (jimiž se zabývá oddíl 3.4). Parametr *Precision* určuje, v kolika bodech jsou funkce  $p_i^{opt}$  vyčísleny před tím, než je získanými body proložen aproximující polynom (viz oddíl 3.1). Tento parametr určuje nejen kvalitu získaných koeficientů, ale rovněž přesnost, se kterou je nalezen bod nespojitosti cenových reakčních křivek. Parametr *Degree* udává stupeň polynomu, kterým se reakční křivky aproximují. Při výpočetních experimentech se jako dostačující jevil polynom druhého stupně. Parametr *Precision* je třeba volit s rozvahou, neboť pro každý bod, pro který hodnotu funkce vyčíslujeme, je třeba nalézt pomocí numerického algoritmu minimum funkce jedné proměnné, která je navíc definována takovým způsobem, že se při výpočtu její hodnoty na sebe odkazuje vícero vnořených funkcí. Pokud tedy nastavíme tento parametr příliš vysoko, tak může být zejména pro delší řetězce výpočet poměrně zdlouhavý.

Obrázek P-2: Pomocné grafy v uživatelském rozhraní



Rámeček *Simulation settings* obsahuje, jak název napovídá, sadu parametrů pro spuštění simulačního modelu. Parametr *Slope estimate* se týká počátečních představ článků o jejich individuální poptávkové funkci a podrobněji se jím zabývá oddíl 4.2. Parametrem *Number of rounds* lze nastavit délku simulačního běhu. Žádný parametr z výše komentovaného rámečku *Approximation settings* se sice netýká vlastní simulace, nicméně parametry *Degree* a *Precision* jsou po dokončení simulace použity při výpočtu rovnovážných hodnot, které jsou zobrazeny v grafickém výstupu k simulačnímu běhu. Vzhledem k tomu, že veškeré výpočty se simulačním modelem počítají se spojitými funkcemi v roli nákladů i poptávky, není brána hodnota rolovacího seznamu *Pricing options* vůbec v úvahu a ceny jsou pro účely simulace automaticky brány jako spojitě.

Poslední rámeček v levé části hlavního okna slouží k nastavení parametrů poptávkové funkce. Jak již bylo řečeno, v uživatelském prostředí lze pracovat pouze s funkcí *abda*. Jsou zde však k dispozici dva způsoby, jak parametry funkce *abda* zadávat. Zvolíme-li v rolovacím seznamu položku '*abda*' (*a,b,d,alpha*), budeme zadávat přímo hodnoty parametrů z definice (3.13). Vybereme-li položku '*abda*' (*pmax,qmax,...*), budeme zadávat pouze parametry *b* a *a* a dále hodnotu  $p_{max}$ , která představuje nejnížší cenu nulující poptávku, a množství  $q_{max}$ , které odpovídá nulové ceně (jde vlastně o průsečíky křivky poptávky s osami grafu). Tyto hodnoty

již zbylé parametry  $a$  a  $d$  jednoznačně určují (výpočet parametrů funkce  $abda$  při zadaných hodnotách  $b$ ,  $\alpha$ ,  $p_{max}$  a  $q_{max}$  je zprostředkován funkcí  $param\_abda$ ).

Rozhodneme-li se zadávat přímo parametry  $a$ ,  $b$ ,  $d$  a  $\alpha$ , je třeba zajistit, aby poptávková funkce byla klesající a pro nějakou dostatečně velkou cenu nabývala hodnoty 0. Označíme-li tuto cenu  $p_{max}$ , je třeba rovněž volit parametry tak, aby výsledná funkce byla na celém intervalu  $[0; p_{max}]$  definovaná a spojitá. Pro konvexní klesající funkce lze jako vodítko pro nastavení parametrů použít tabulku 3-5 z oddílu 3.2, průběh poptávkové funkce je možné vizuálně kontrolovat v okně s grafy (viz obrázek P-2).

### P.1.2. Hledání rovnovážných cen pomocí uživatelského rozhraní

Jakmile nastavíme všechny parametry řetězce, spustíme výpočet rovnovážných cen tlačítkem *Compute equilibrium*. Po jeho stisku se nejprve v hlavním příkazovém podokně zobrazí název funkce, která byla spuštěna, spolu se všemi jejími vstupními parametry. Jakmile je výpočet dokončen, vytiskne se v příkazovém podokně výpis výsledků a otevře se okno s grafickým výstupem. V případě, že bychom spustili výpočet při takovém nastavení parametrů, jaké zobrazuje obrázek P-1, objeví se v hlavním příkazovém podokně následující výpis (pro přehlednost zde byl dvakrát zalomen první řádek výpisu):

```
Executing: equil(100,2,1,1000,[@cost_lin @cost_pow @cost_poly @dem_abda],
               'dem',7500000,500,-5000,-1,
               'cost1',60,'cost2',600,0.75,'cost3',[0 -0.002 50 30000])
```

```
-----
agent 1:      price      profit      profitshare
agent 2:      399      153956      0.595
agent 3:      713      83346      0.322
            875      21507      0.083

demand:      454.0666
efficiency:   0.34721
profit CSC:   745395.3993
price CSC:    497.5861
demand CSC:   2518.148
-----
```

Tabulka v horní části výsledkové listiny představuje rovnovážné hodnoty, kterých dosáhnou jednotlivé články (*agents*) v decentralizovaném řetězci. Sloupec *price* zachycuje výsledné ceny, sloupec *profit* výsledný zisk a ve sloupci *profitshare* je vyjádřen podíl jednotlivých článků na celkovém zisku. Jak ceny, tak zisky jsou pro větší přehlednost zaokrouhleny na celá čísla, ziskové podíly pak na tři desetinná místa. Ve spodní části výsledkové listiny je nejprve uvedeno poptávané množství, které bude prodávat decentralizovaný řetězec, dále efektivnost tohoto řetězce, a nakonec několik údajů týkajících se řetězce centralizovaného (CSC): celkový zisk tohoto řetězce, cena koncové produkce a poptávané množství, které této ceně odpovídá. Prvního řádku výpisu, který ukazuje přesnou syntax funkce spuštěné pro nalezení všech výsledných hodnot, lze využít v případě, že chceme dále pracovat s daty, která při výpočtu získáme. Spustíme-li totiž výpočet

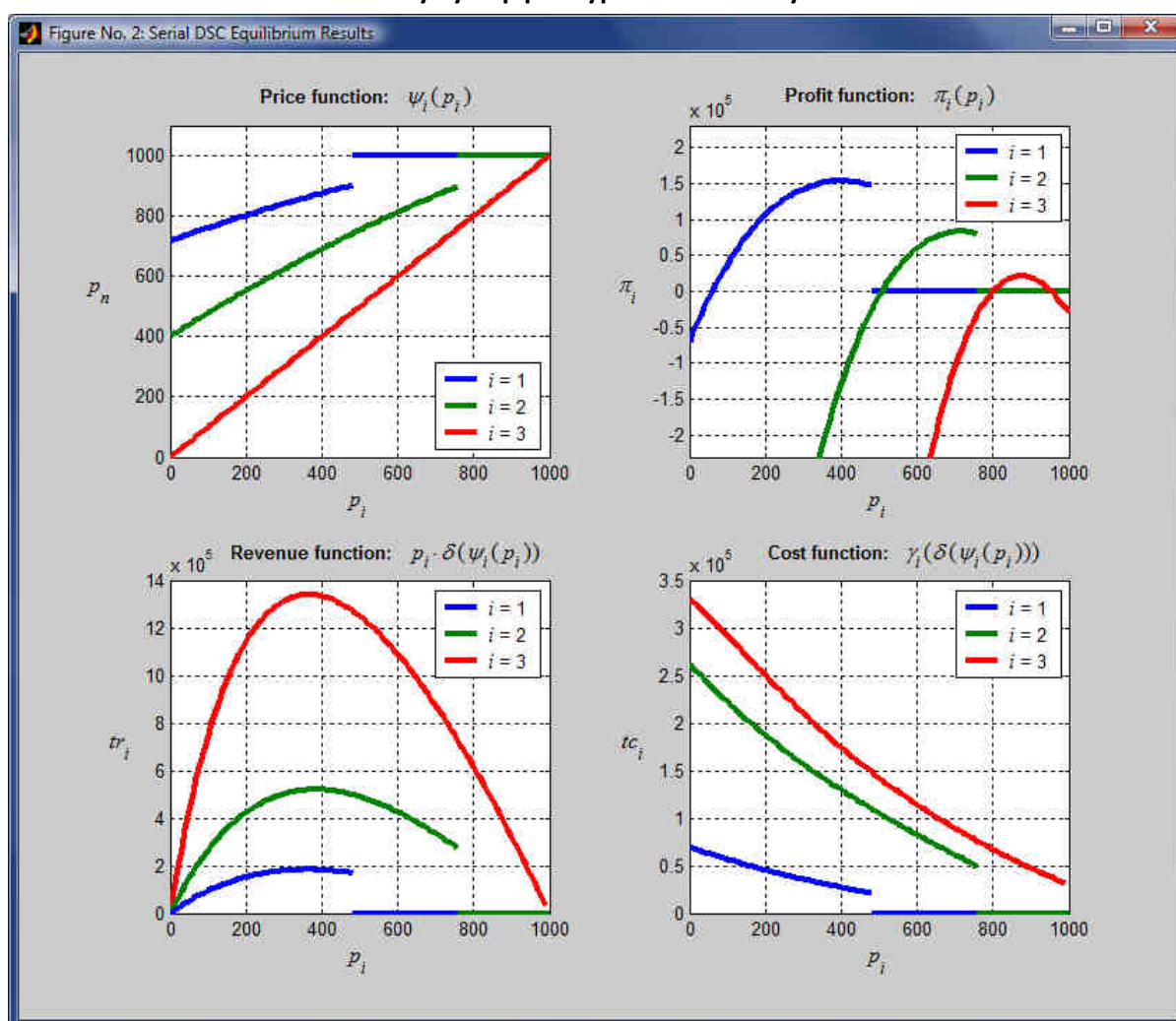
z uživatelského rozhraní, výsledky se pouze v souhrnné podobě vypíší na obrazovku. Chceme-li uložit přesnou hodnotu výsledných proměnných pro další zpracování, je třeba ručně spustit funkci uvedenou v prvním řádku a její výstupní hodnoty zapsat do nějakých proměnných. Funkce *equil* i *equilpar* vracejí veškeré výsledky v pořadí, které je uvedeno ve výsledkové listině (s tím, že první tři hodnoty výstupu tvoří vektory představující sloupce tabulky z výsledkové listiny). Abychom tedy např. uložili výsledky pro stejný řetězec, jako v předchozím příkladě, můžeme do hlavního příkazového podokna napsat příkaz

```
[price,profit,profitshare,demand,efficiency,priceCSC,profitCSC,demandCSC] =
```

a za rovnítko zkopírovat syntax funkce z prvního řádku výstupu z grafického rozhraní.

Obrázek P-3 ukazuje okno s grafickým výstupem, které se otevře po spuštění výpočtu charakteristik dodavatelského řetězce popsaného v hlavním okně uživatelského rozhraní z obrázku P-1. Všechny grafy, které se v okně nachází, se týkají decentralizovaného řetězce. Zobrazované informace se poněkud liší pro řetězce sériové a paralelní. Popíšeme nejprve výstup pro řetězec sériový (který zachycuje i obrázek P-3).

**Obrázek P-3: Grafický výstup pro výpočet rovnováhy v sériovém řetězci**



Graf vlevo nahoře, nadepsaný jako *Price function*, se týká cenových reakčních křivek  $p_i^{opt}$ , které byly popsány v oddílu 3.1. Funkce, nazvané v nadpisu grafu jako  $\psi_i$ , mají pro  $i = 1, 2, \dots, n-1$  předpis

$$\psi_i = p_{i+1}^{opt} \circ p_{i+2}^{opt} \circ \dots \circ p_n^{opt}, \quad (P.1)$$

pro  $i = n$  se jedná o identitu. Funkce  $\psi_i$  tedy udává vztah mezi výstupní cenou, kterou volí  $i$ -tý článek, a cenou koncové produkce  $p_n$  za podmínky, že články  $i+1, i+2, \dots, n-1$  se snaží optimalizovat svůj individuální zisk. Jedná se tedy o reakci ceny volené posledním článkem na cenu stanovenou článkem  $i$ .

Zbylé tři grafy z této reakční funkce vychází. Graf vlevo dole (označený jako *Revenue function*) ukazuje závislost mezi výstupní cenou  $i$ -tého článku a jeho celkovými tržbami, opět za podmínky, že následující články maximalizují individuální zisk. Graf vpravo dole (*Cost function*) znázorňuje velikost celkových nákladů za stejných podmínek a konečně graf vpravo nahoře (*Profit function*) zachycuje závislost mezi cenou  $i$ -tého článku a jeho ziskem v případě, že navazující články maximalizují individuální zisk a předchozí článek volí výstupní cenu na úrovni své rovnovážné ceny. Z tohoto grafu je patrné, že globální maxima ziskových funkcí opravdu odpovídají rovnovážným hodnotám z výsledkové listiny.

Pokud nyní změníme strukturu dosud uvažovaného řetězce z obrázku P-1 v uživatelském rozhraní na paralelní a spustíme výpočet, bude mít výstup v příkazovém podokně následující podobu:

```
Executing: equilpar(100,2,1,1000,[@cost_lin @cost_pow @cost_poly @dem_abda],
                    'dem',7500000,500,-5000,-1,
                    'cost1',60,'cost2',600,0.75,'cost3',[0 -0.002 50 30000])
```

```
-----
agent 1:      price      profit      profitshare
agent 2:      312      145651         0.449
agent 3:      344      128086         0.395
agent 3:      845      50558         0.156

demand:      577.5579
efficiency:   0.43507
profit CSC:   745395.3993
price CSC:    497.5861
demand CSC:   2518.148
-----
```

Forma výpisu je zcela totožná jako v případě sériového řetězce. Použitá funkce se změnila na *equilpar*, z výpisu je však patrné, že syntax vstupních parametrů této funkce je zcela stejná jako v předchozím případě u funkce *equil*. Zcela analogický je rovněž datový výstup z obou funkcí. Pokud bychom chtěli s výslednými hodnotami dále pracovat, lze využít prvního řádku zobrazeného výpisu stejným způsobem, jaký byl popsán výše pro funkci *equil*.

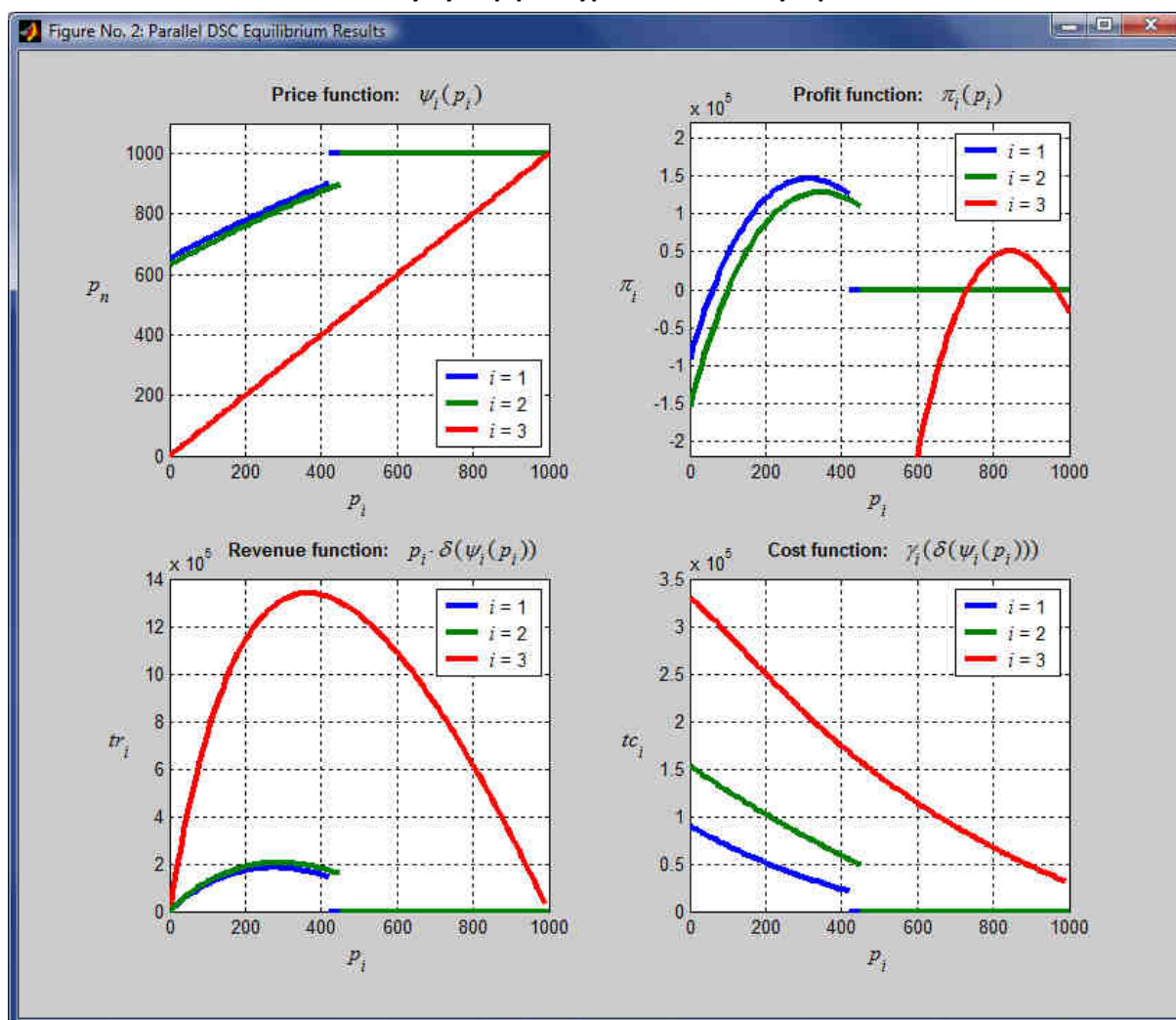
Okno s grafickým výstupem ukazuje obrázek P-4. Cenová reakční funkce článků z první úrovně řetězce, kterou zobrazuje graf vlevo nahoře, zachycuje závislost mezi cenou,

kterou volí daný článek a cenou koncové produkce za podmínky, že všechny ostatní články z první úrovně řetězce volí rovnovážné ceny a koncový článek optimalizuje individuální zisk. Formálně zapsáno (opět s použitím značení z oddílu 3.1):

$$\psi_i(p_i) = p_n^{opt} \left( p_i + \sum_{\substack{j=1, \\ j \neq i}}^{n-1} p_j^{eq} \right). \quad (P.2)$$

Funkce tržeb, nákladů a zisku na zbylých třech grafech odpovídají svým protějškům ze sériového řetězce s tím, že se při jejich konstrukci vychází z této definice funkce  $\psi_i$ . Vždy se tedy jedná o závislost mezi výstupní cenou daného článku a příslušným ukazatelem za podmínky, že všechny články z první úrovně volí rovnovážnou cenu a článek koncový optimalizuje individuální zisk.

**Obrázek P-4: Grafický výstup pro výpočet rovnováhy v paralelním řetězci**



Výstupy z výpočtů poněkud změní, pokud se rozhodneme počítat paralelní model s diskrétními cenami, kterými se zabývá pododdíl 3.4.2. V příkladu 23 z tohoto pododdílu jsme ukázali, že v paralelním modelu s diskrétními cenami může existovat více rovnovážných kombinací cen. Výstup funkcí spuštěných z uživatelského rozhraní nám ukáže výsledky pro všechny rovnovážné kombinace.

Vraťme se nyní ke zmíněnému příkladu 23; uvažujeme zde paralelní dodavatelský řetězec se čtyřmi články, každý z nich vyrábí s jednotkovými náklady 5 peněžních jednotek a poptávka po koncové produkci je ve tvaru

$$q = 1000 - 12,5 \cdot p_n.$$

Spustíme-li z uživatelského rozhraní výpočet, zobrazí se kvůli přítomnosti několika rovnovážných kombinací cen poněkud rozsáhlejší textový výstup než v předchozích příkladech. Veškeré hodnoty týkající se rovnovážného řetězce jsou uspořádány do tabulky, jejíž sloupce zachycují údaje pro jednotlivé rovnovážné body (původní výstup obsahoval 13 sloupců, v následujícím výpisu jsou pro přehlednost zahrnuty pouze první čtyři – rovnovážné ceny pro všech 13 rovnovážných bodů uvádí tabulka 3-37):

```
Executing: equilibpard(1,80,[@cost_lin @cost_lin @cost_lin @cost_lin @dem_abda],
                        'dem', -12.5,0,1000,1,
                        'cost1',5,'cost2',5,'cost3',5,'cost4',5)
```

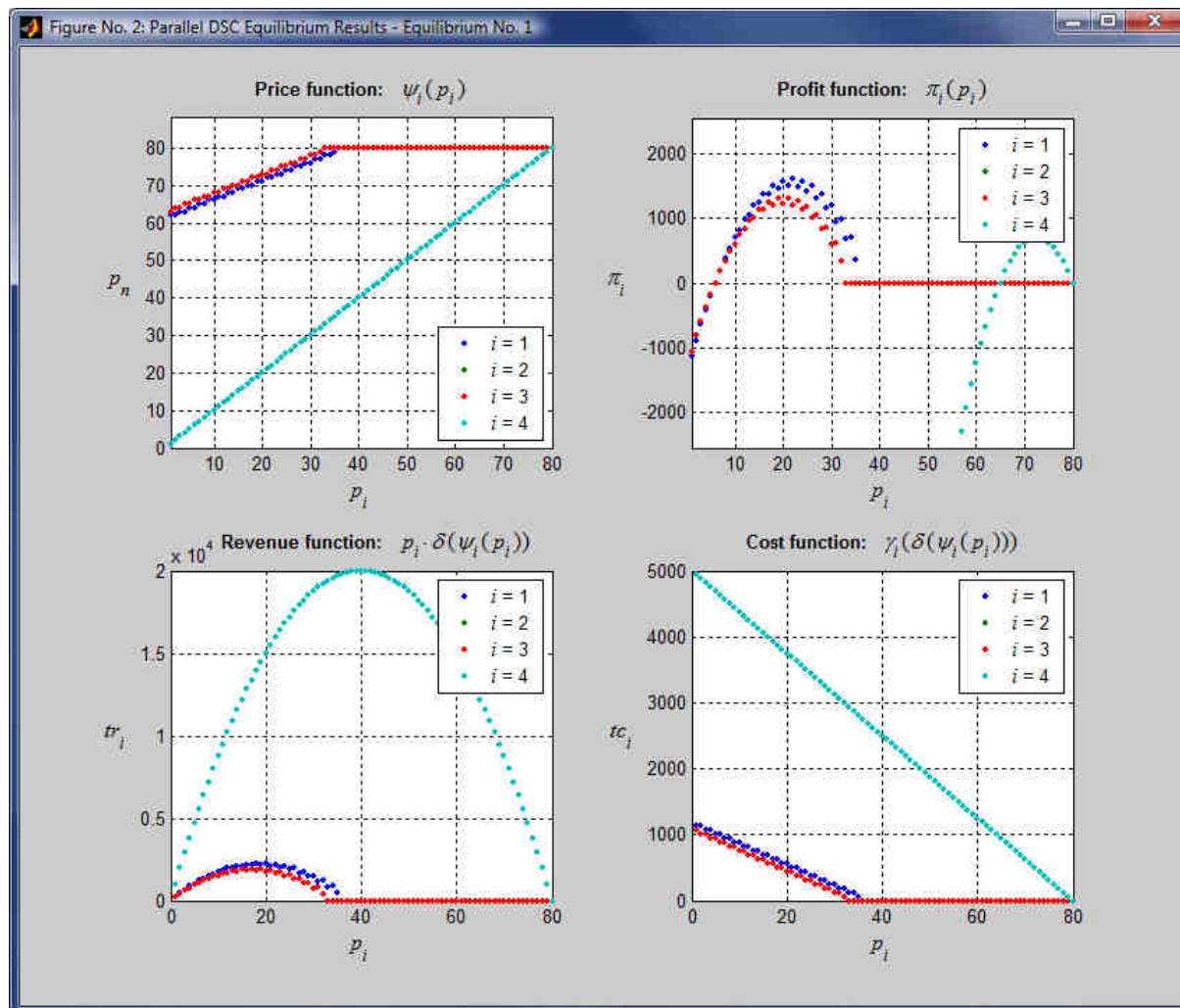
-----  
Multiple Nash equilibria found for 1st-level agents.

| equil. #:      | 1       | 2     | 3     | 4     |
|----------------|---------|-------|-------|-------|
| price 1:       | 22      | 21    | 21    | 20    |
| price 2:       | 19      | 20    | 19    | 21    |
| price 3:       | 19      | 19    | 20    | 19    |
| price 4:       | 72      | 72    | 72    | 72    |
| profit 1:      | 1700    | 1600  | 1600  | 1500  |
| profit 2:      | 1400    | 1500  | 1400  | 1600  |
| profit 3:      | 1400    | 1400  | 1500  | 1400  |
| profit 4:      | 700     | 700   | 700   | 700   |
| profitshare 1: | 0.327   | 0.308 | 0.308 | 0.288 |
| profitshare 2: | 0.269   | 0.288 | 0.269 | 0.308 |
| profitshare 3: | 0.269   | 0.269 | 0.288 | 0.269 |
| profitshare 4: | 0.135   | 0.135 | 0.135 | 0.135 |
| demand:        | 100     | 100   | 100   | 100   |
| efficiency:    | 0.277   | 0.277 | 0.277 | 0.277 |
| profit CSC:    | 18762.5 |       |       |       |
| price CSC:     | 1       |       |       |       |
| demand CSC:    | 987.5   |       |       |       |

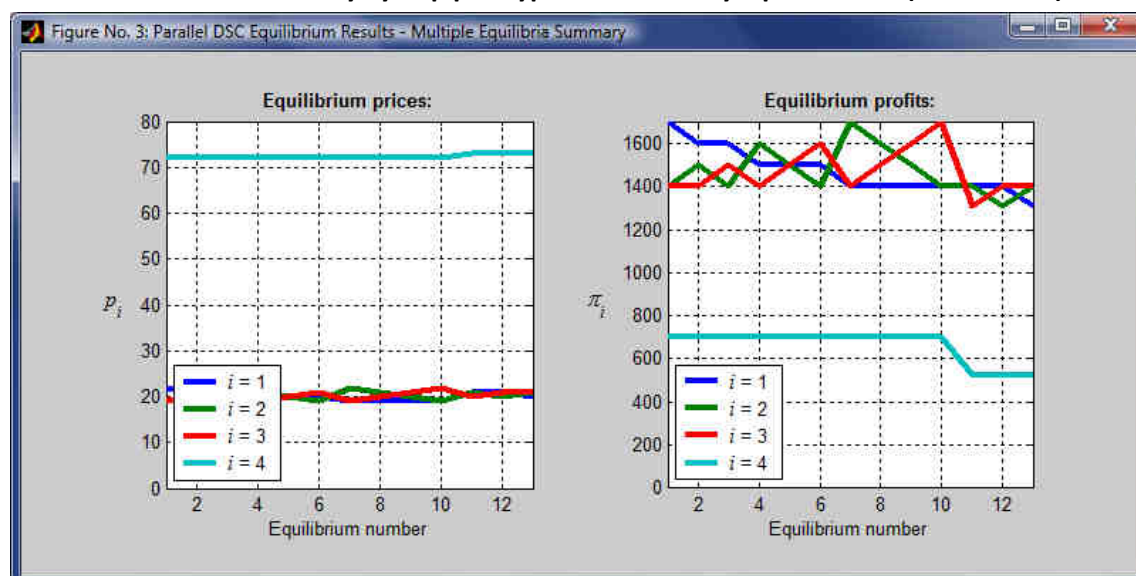
-----

Na rozdíl od modelu se spojitými cenami se po dokončení výpočtu neotevře jedno okno s grafickým výstupem (viz příloha P.1), nýbrž hned okna dvě. První z nich, které znázorňuje obrázek P-5, představuje výstup analogický tomu pro spojitý model s tím, že zobrazuje údaje pro *první rovnovážný bod* – tato skutečnost je uvedena i v názvu okna. Grafy ve druhém okně (na obrázku P-6) slouží k přehlednému porovnání cen a individuálních zisků v různých rovnovážných bodech (jde o stejné grafy, jako ukazuje graf 3-38).

Obrázek P-5: Grafický výstup pro výpočet rovnováhy v příkladu 23 (první okno)



Obrázek P-6: Grafický výstup pro výpočet rovnováhy v příkladu 23 (druhé okno)



### P.1.3. Multiagentní simulace pomocí uživatelského rozhraní

Výstupy po spuštění multiagentní simulace jsou obecně vzato jednodušší a snáze čitelné, nežli výstupy při výpočtu rovnovážných hodnot. Pokud nastavíme v hlavním okně parametry dodavatelského řetězce opět tak, jak to zachycuje obrázek P-1, a spustíme simulaci tlačítkem *Simulate dynamics*, objeví se okno s grafickým výstupem jako na obrázku P-7, a v příkazovém okně se vypíše následující výstup<sup>11</sup> (první řádek výstupu je opět pro přehlednost třikrát zalomen):

```
Executing: simulpar2(100,2,1,150,[5577.1617 6459.6852 14218.1297],1000,
                  [@cost_lin @cost_pow @cost_poly @dem_abda],
                  'dem',7500000,500,-5000,-1,'cost1',60,
                  'cost2',600,0.75,'cost3',[0 -0.002 50 30000])
```

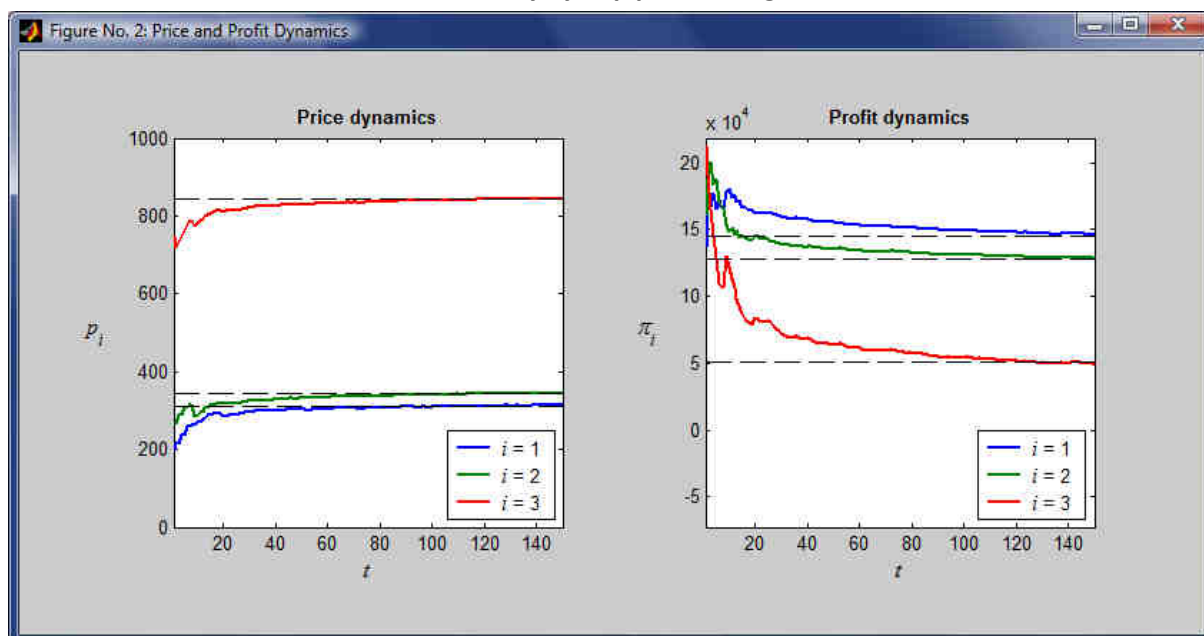
-----  
DSC Simulation results:

| Final Round: | Price | Profit |
|--------------|-------|--------|
| Agent 1:     | 314   | 146297 |
| Agent 2:     | 345   | 128511 |
| Agent 3:     | 845   | 49094  |

| Equilibrium: | Price | Profit |
|--------------|-------|--------|
| Agent 1:     | 312   | 145651 |
| Agent 2:     | 344   | 128086 |
| Agent 3:     | 845   | 50558  |

-----

Obrázek P-7: Grafický výstup pro multiagentní simulaci



Grafy na obrázku P-7 mají stejnou podobu jako grafy z oddílu 4. Barevné křivky na grafu vlevo zachycují vývoj cen stanovených jednotlivými články během simulačního běhu,

<sup>11</sup> Vzhledem k tomu, že při určení počátečních představ článků o poptávkové funkci hraje roli náhoda (viz pododdíl 4.2.1, vzorec (4.1)), nemusí výsledné hodnoty při opakovaném spuštění přesně souhlasit.

který zde trval 150 časových období (kol). Při spuštění simulace z uživatelského rozhraní se současně spustí i výpočet rovnovážných cen, které jsou v grafu vyznačeny černými čárkovanými čarami. Toto zobrazení umožňuje posoudit, zdali a případně jak rychle konvergovaly ceny k rovnovážným hodnotám. Textový výstup zaznamenává pouze ceny z posledního kola simulace (sloupec *Price* v tabulce označené jako *Final Round*) a ceny rovnovážné (v tabulce *Equilibrium*). Všechny hodnoty v textovém výstupu jsou podobně jako v předchozích případech zaokrouhleny na celá čísla.

Graf na obrázku P-7 napravo ukazuje vývoj individuálních zisků, čárkovaně jsou zde opět zaneseny rovnovážné hodnoty. Podobně jako v případě cen se v textovém výpisu nachází pouze zaokrouhlené hodnoty z posledního kola simulačního běhu a hodnoty rovnovážné. Pokud bychom chtěli znát přesné (resp. nezaokrouhlené) hodnoty cen a zisků z posledního kola nebo i kol ostatních, musíme funkci *simulpar2* spustit ručně a její výstup (který tvoří právě vektory hodnot cen a zisků v jednotlivých kolech) uložit do nějakých proměnných. Máme-li v příkazovém okně výstup týkající se zkoumané simulace, stačí napsat např.

```
[prices,profits] =
```

a za rovnítko zkopírovat syntax funkce z prvního řádku výstupu.

Podobného postupu můžeme použít i v případě, že potřebujeme pro vyjádření poptávky nebo nákladů použít jiné funkční tvary, než které nabízí uživatelské rozhraní. V uživatelském rozhraní jsou pro popis dodavatelského řetězce k dispozici pouze funkční tvary, které byly popsány v oddílu 3.2. Pokud bychom se rozhodli použít funkce jiné, je třeba spustit výpočet pomocí některé z funkcí *equil*, *equilpar*, *equild*, *equilpard*, *simul*, *simulpar* nebo *simulpar2* ručně z příkazového okna. Nutno však podotknout, že syntax těchto funkcí je poněkud komplikovaná (jak ostatně ukazuje první řádek výpisu v předchozích příkladech). Je tomu tak proto, že zadávané parametry musejí obsáhnout mimo jiné popis veškerých funkčních vztahů v řetězci.

I v tomto případě lze však uživatelského rozhraní využít jako vodítka pro správné nastavení vstupních parametrů spouštěné funkce. Můžeme spustit nejprve výpočet pro dodavatelský řetězec, který namísto požadované funkce obsahuje nějakou funkci z nabídky, poté použít vypsaného příkazového řádku a dotyčnou funkci v něm nahradit. Komentář k jednotlivým vstupním parametrům je k dispozici ve vestavěné nápovědě ke všem funkcím (nápověda se spouští příkazem „*help*“ následovaným jménem funkce). Následující oddíl uvádí zdrojové kódy těchto funkcí, stejně jako některých dalších funkcí, kterými se práce ve větší či menší míře podrobnosti zabývá.

## P.2 Zdrojové kódy některých funkcí

V této příloze předkládám zdrojové kódy některých funkcí, které realizují algoritmy popsané v kapitolách 3 a 4 a které byly v těchto kapitolách zmíněny či podrobněji rozebrány. Všechny funkce jsou naprogramovány v jazyce Matlab a jsou obsaženy v elektronické podobě na přiloženém CD. Funkce jsou v následujícím textu řazeny abecedně.

Zeleně jsou v textu zvýrazněny některé poznámky k průběhu algoritmů a na začátku každé funkce též nápověda dané funkce, kterou je možné zobrazit v příkazovém okně prostředí Matlab příkazem „*help*“ následovaným příslušným jménem funkce. Zdrojové kódy některých funkcí byly zkráceny o poměrně rozsáhlé části zabývající se grafickým výstupem, které nejsou z hlediska implementace uvedených algoritmů podstatné.

### SEZNAM FUNKCÍ:

|                                  |     |
|----------------------------------|-----|
| Funkce <i>cost_lin</i> .....     | 133 |
| Funkce <i>cost_poly</i> .....    | 134 |
| Funkce <i>cost_pow</i> .....     | 134 |
| Funkce <i>costsum</i> .....      | 134 |
| Funkce <i>dem_abda</i> .....     | 134 |
| Funkce <i>dem_poly</i> .....     | 134 |
| Funkce <i>demmap</i> .....       | 135 |
| Funkce <i>demmappar</i> .....    | 135 |
| Funkce <i>dempoly</i> .....      | 135 |
| Funkce <i>equil</i> .....        | 136 |
| Funkce <i>equild</i> .....       | 137 |
| Funkce <i>equilpar</i> .....     | 138 |
| Funkce <i>equilpard</i> .....    | 140 |
| Funkce <i>polycut</i> .....      | 142 |
| Funkce <i>pricepoly</i> .....    | 142 |
| Funkce <i>pricepolypar</i> ..... | 143 |
| Funkce <i>pricing</i> .....      | 145 |
| Funkce <i>pricingpar</i> .....   | 145 |
| Funkce <i>pricingpar2</i> .....  | 147 |
| Funkce <i>simul</i> .....        | 148 |
| Funkce <i>simulpar</i> .....     | 149 |
| Funkce <i>simulpar2</i> .....    | 150 |

### Funkce *cost\_lin*

```
function cost = cost_lin(q,c)
% COST_LIN Linear cost function.
%
% COST_LIN(q,c) is the total variable cost of production of
% 'q' units with the unit production price of 'c'.

cost = q.*c;
cost = max(cost,zeros(size(cost)));
```

### Funkce *cost\_poly*

```
function cost = cost_poly(q,coeffs)
% COST_POLY Polynomial cost function.
%
% COST_POLY(q,coeffs) is calculated as polyval(coeffs,q).

cost = polyval(coeffs,q);
cost = max(cost,zeros(size(cost)));
```

---

### Funkce *cost\_pow*

```
function cost = cost_pow(q,mult,pow)
% COST_POW Power cost function.
%
% COST_POW(q,mult,pow) results in mult*(q.^pow), unless the result
% is negative, in which case the value is set to zero.

cost = mult*(max(q,zeros(size(q))).^pow);
```

---

### Funkce *costsum*

```
function cost = costsum(q,fcost,varargin)
% COSTSUM Sum of all production costs in a SC.
%
% COSTSUM(q,fcost,varargin) calculates the sum of all agent's
% production costs in a SC. Argument "q" specifies the quantity
% being produced, "fcost" provides function handles to cost
% functions and "varargin" lists the arguments for all cost
% functions, delimited by strings 'cost1','cost2' etc.
% COSTSUM is used for the calculation of a CSC's profit.
% Example syntax:
%
% cost = costsum(1000,[@cost_lin @cost_pow],...
%                  'cost1',50,'cost2',50,0.25)

n = length(fcost);
param = varargin2arr(n,varargin{:});
val = zeros(n,length(q));
for i = 1:n
    val(i,:) = feval(fcost(i),q,varargin{param(i,1):param(i,2)});
end
cost = sum(val);
```

---

### Funkce *dem\_abda*

```
function dem = dem_abda(p,a,b,d,alpha)
% DEM_ABDA Demand function with parameters a,b,d and alpha.
% DEM_ABDA(p,a,b,d,alpha) computes the demand for a supply chain's
% production at price 'p' according to the following formula:
%      dem = a*(p + b)^alpha + d

dem = a.*(p + b).^alpha + d;
dem = max(dem,zeros(size(dem))); % Quantity demanded is non-negative.
```

---

### Funkce *dem\_poly*

```
function dem = dem_poly(p,coeffs)
% DEM_POLY Polynomial demand function.
%
% DEM_POLY(p,coeffs) returns quantity demanded of a good
% at price "p" with the demand function being a polynomial with
% coefficients "coeffs". It's taken for granted that demand functions
```

---

---

```
% exhibit a negative slope; therefore, if the slope at point "p" is
% positive, quantity demanded of 0 is returned.
```

```
dem = polyval(coeffs,p);
dem = max(dem,zeros(size(dem)));
der = polyval(polyder(coeffs),p);
dem(der>0)=0;
```

---

### Funkce *demmap*

```
function dem = demmap(fdem,map,varargin)
% DEMMAP Demand function with price mappings inside a SC.
%
% DEMMAP(fdemand,map,varargin) calculates quantity demanded
% in a SC as a result to various output prices of an agent
% given that succeeding agents maximize their individual profit
% and their integer price mappings are described by "map".
% The demand function for the end production is specified by
% function handle "fdemand" and a list of arguments
% in "varargin".
%
% See also: EQUILD, EQUILDPAR.

[n,pmax] = size(map);
price = 1:pmax;
if n > 1
    for i = 1:(n-1)
        price = map(i,price);
    end
end
dem = feval(fdem,price,varargin{:});
```

---

### Funkce *demmappar*

```
function dem = demmappar(fdem,map,varargin)
% DEMMAPPAR Discrete demand function of 1st level agents in a parallel SC.
%
% DEMMAPPAR(fdem,map,varargin) calculates quantity demanded
% under various settings of 1st level agents' prices in
% a parallel DSC, given that the 2nd level agent optimizes
% their individual profit (and his pricing function is
% represented by the discrete integer mapping "map").
% The demand function for the end production is specified by
% function handle "fdemand" and a list of arguments
% in "varargin".
%
% See also: EQUILD, EQUILDPAR.

pmax = length(map);
price = [1:pmax]'*ones(1,pmax) + ones(pmax,1)*[1:pmax];
price = min(price,pmax);
for i = 1:pmax
    price(i,:) = map(price(i,:));
end
dem = feval(fdem,price,varargin{:});
```

---

### Funkce *dempoly*

```
function dem = dempoly(price,fdemand,pricecoeffs,pmax,varargin)
% DEMPOLY Demand function with polynomial price relations inside a SC.
%
% DEMPOLY(price,fdemand,pricecoeffs,pmax,varargin) calculates quantity
% demanded of a non-cooperative SC, given the price of a downstream
% agent, where the demand function for the end production is
```

---

```
% specified by function handle "fdemand" and a list of arguments
% in "varargin", and the relations between the prices of consecutive
% agents are determined by cut-off polynomials (see POLYCUT) with
% parameters in "pricecoeffs" and "pmax".
%
% See also: POLYCUT, PRICEPOLY, PRICEPOLYPAR.

[n,dummy] = size(pricecoeffs);
if n > 1
    for i = 1:(n-1)
        price = polycut(price,pricecoeffs(i,:),pmax(i),pmax(i+1));
    end
else
    price = polycut(price,pricecoeffs,pmax(1),pmax(2));
end

dem = feval(fdemand,price,varargin{:});
```

---

### Funkce *equil*

```
function [price,profit,profitshare,demand,eff,priceCSC,profCSC,demCSC,pmax] = ...
    equil(precision,degree,ploton,pmax,fcostdem,varargin);
% EQUIL Finds equilibrium prices and profits in a serial DSC.
%
% EQUIL(precision,degree,ploton,pmax,fcostdem,varargin) calculates
% equilibrium characteristics in a serial DSC and, if desired, prints
% out the results in the form of charts. The argument "fcostdem" is an
% array of function handles specifying the cost and demand functions,
% their additional parameters are supplied in the "varargin" argument,
% for valid syntax of "varargin" and correct order of functions in
% "fcostdem" see example below. If a price of end production is known,
% at which the demand is zero (or less than 1), it should be stated
% in "pmax" argument (otherwise, set "pmax" to high enough a number).
%
% EQUIL uses polynomial fitting to explicitly approximate (implicit)
% price functions inside the SC. The "precision" parameter determines
% the number of points at which price functions will be evaluated
% for the sake of fitting, "degree" is the degree of the fitting polynom;
% see also PRICEPOLY.
%
% For plots to be displayed, set "ploton" to 1.
%
% Example syntax:
%
% [p,profit,profitshare] = equil(precision,degree,ploton,pmax,...
%                               [@cost_lin @cost_poly @cost_pow @dem_abda],...
%                               'dem',a,b,d,alpha,...
%                               'cost1',unit_cost1,...
%                               'cost2',[c21 c22 c23],...
%                               'cost3',mult,pow);
%
% See also: PRICEPOLY,COST_LIN,COST_POLY,COST_POW,DEM_ABDA,PARAM_ABDA.

% Parameter analysis.
n = length(fcostdem) - 1;
param = varargin2arr(n,varargin{:});
fdem = fcostdem(n+1);

% Variable initialization.
price = zeros(1,n);
polycoeffs = zeros(n,degree+1);
pmax = pmax*ones(1,n);

% Approximates price functions.
for k = n:-1:2
    if k == n
        [polycoeffs(n-1,:),pmax(n-1)] = ...
```

---

---

```

        pricepoly(precision,degree,pmax(n),fdem,fcostdem(n), ...
            'dem',varargin{param(n+1,1):param(n+1,2)}, ...
            'cost',varargin{param(n,1):param(n,2)});
    else
        [polycoeffs(k-1,:),pmax(k-1)] = ...
            pricepoly(precision,degree,pmax(k),@dempoly,fcostdem(k), ...
                'dem',fdem,polycoeffs(k:n,:),pmax(k:n), ...
                varargin{param(n+1,1):param(n+1,2)}, ...
                'cost',varargin{param(k,1):param(k,2)});
    end
end

% Finds equilibrium prices.
price(1) = fminbnd(@profit,0,1.1*pmax(1),[],0,@dempoly,fcostdem(1), ...
    'dem',fdem,polycoeffs(1:n,:),pmax(1:n), ...
    varargin{param(n+1,1):param(n+1,2)}, ...
    'cost',varargin{param(1,1):param(1,2)});
if price(1) > (1-1/(160*precision))*pmax(1), price(1) = pmax(1); end
for i = 2:n
    price(i) = polycut(price(i-1),polycoeffs(i-1,:),pmax(i-1),pmax(i));
end

% Calculates equilibrium quantity demanded and profits.
demand = feval(fdem,price(n),varargin{param(n+1,1):param(n+1,2)});
profit(1) = demand*price(1) - ...
    feval(fcostdem(1),demand,varargin{param(1,1):param(1,2)});
for i = 2:n
    profit(i) = demand*(price(i) - price(i-1)) - ...
        feval(fcostdem(i),demand,varargin{param(i,1):param(i,2)});
end
profitshare = profit/sum(profit);

% Calculates CSC's optimal price, profit and quantity demanded,
% and DSC's efficiency.
[priceCSC profCSC] = fminbnd(@profit,0,1.1*pmax(n),[],0,fdem,@costsum, ...
    'dem', varargin{param(n+1,1):param(n+1,2)}, ...
    'cost', fcostdem(1:n),varargin{:});
profCSC = -profCSC;
demCSC = feval(fdem,priceCSC,varargin{param(n+1,1):param(n+1,2)});
eff = sum(profit)/profCSC;

```

---

### Funkce *equild*

```

function [price,profit,profitshare,demand,eff,priceCSC,profCSC,demCSC] = ...
    equild(ploton,pmax,fcostdem,varargin);
% EQUILD Finds equilibrium in a serial DSC with discrete prices.
%
% EQUILD(precision,degree,ploton,pmax,fcostdem,varargin) calculates
% equilibrium characteristics in a serial DSC with discrete (i.e.,
% integer) prices and, if desired, displays results in the form
% of graphs. The syntax and output is the same as that of EQUIL
% function (the only difference between the two functions is that
% unlike EQUIL, EQUILD deals with integer prices only).
%
% See also: EQUIL,EQUILPARD.

% Parameter analysis.
n = length(fcostdem) - 1;
param = varargin2arr(n,varargin{:});
fdem = fcostdem(n+1);

% Variable initialization.
price = zeros(1,n);
if pmax ~= fix(pmax), pmax = fix(pmax)+1; end
map = zeros(n,pmax);
map(n,:) = 1:pmax;
o = ones(1,pmax);

```

---

```
% Finds price mapping.
for k = n:-1:2
    dem = feval(@demmap,fdem,map(k:n,:),varargin{param(n+1,1):param(n+1,2)});
    cost = feval(fcostdem(k),dem,varargin{param(k,1):param(k,2)});
    profs = (dem*o).*( [1:pmax]'*o-o'*[1:pmax] )-cost*o;
    [profs,map(k-1,:)] = max(profs);
    map(k-1,profs==0) = pmax;
end

% Finds equilibrium prices.
dem = feval(@demmap,fdem,map,varargin{param(n+1,1):param(n+1,2)});
cost = feval(fcostdem(1),dem,varargin{param(1,1):param(1,2)});
profs = dem.*(1:pmax)-cost;
[profit(1),price(1)] = max(dem.*(1:pmax)-cost);
for i = 2:n
    price(i) = map(i-1,price(i-1));
end

% Calculates equilibrium quantity demanded and profits.
demand = feval(fdem,price(n),varargin{param(n+1,1):param(n+1,2)});
for i = 2:n
    profit(i) = demand*(price(i) - price(i-1)) - ...
        feval(fcostdem(i),demand,varargin{param(i,1):param(i,2)});
end
profitshare = profit/sum(profit);

% Calculates CSC's optimal price, profit and quantity demanded,
% and DSC's efficiency.
profs = -feval(@profit,[1:pmax],0,fdem,@costsum, ...
    'dem', varargin{param(n+1,1):param(n+1,2)}, ...
    'cost',fcostdem(1:n),varargin{:});
[profCSC priceCSC] = max(profs);
demCSC = feval(fdem,priceCSC,varargin{param(n+1,1):param(n+1,2)});
eff = sum(profit)/profCSC;
```

---

### Funkce *equilpar*

```
function [price,profit,profitshare,demand,eff,priceCSC,profCSC,demCSC,pmax] = ...
    equilpar(precision,degree,ploton,pmax,fcostdem,varargin);
% EQUILPAR Finds equilibrium prices and profits in a parallel DSC.
%
% EQUILPAR(precision,degree,ploton,pmax,fcostdem,varargin) calculates
% equilibrium characteristics in a parallel DSC and, if desired, prints
% out the results in the form of charts. The argument "fcostdem" is an
% array of function handles specifying the cost and demand functions,
% their additional parameters are supplied in the "varargin" argument,
% for valid syntax of "varargin" and correct order of functions in
% "fcostdem" see example below. If a price of end production is known,
% at which the demand is zero (or less than 1), it should be stated
% in "pmax" argument (otherwise, set "pmax" to high enough a number).
%
% EQUILPAR uses polynomial fitting to explicitly approximate (implicit)
% price functions inside the SC. The "precision" parameter determines
% the number of points at which price functions will be evaluated
% for the sake of fitting, "degree" is the degree of the fitting polynom;
% see also PRICEPOLY, PRICEPOLYPAR.
%
% For plots to be displayed, set "ploton" to 1.
%
% Example syntax:
%
% [p,profit,profitshare] = equilpar(precision,degree,ploton,pmax,...
%     [@cost_lin @cost_poly @cost_pow @dem_abda],...
%     'dem',a,b,d,alpha,...
%     'cost1',unit_cost1,...
%     'cost2',[c21 c22 c23],...
```

---

---

```

%                                     'cost3',mult,pow);
%
% See also: PRICEPOLY,PRICEPOLYPAR,COST_LIN,COST_POLY,COST_POW,DEM_ABDA.

% Parameter analysis.
n = length(fcostdem) - 1;
param = varargin2arr(n,varargin{:});
fdem = fcostdem(n+1);

% Variable initialization.
price = zeros(1,n);
coeffs_n = zeros(1,degree+1); % n-th agent's price function;
coeffs = zeros(n-1,degree+1); % first-level agents;
minsum = zeros(n-1,1);
maxsum = zeros(n-1,1);

% Approximates price functions.
[coeffs_n,pmax2] = pricepoly(precision,degree,pmax,fdem,fcostdem(n), ...
                             'dem',varargin{param(n+1,1):param(n+1,2)}, ...
                             'cost',varargin{param(n,1):param(n,2)});
for i = 1:n-1
    [coeffs(i,:),minsum(i),maxsum(i)] = ...
        pricepolypar(precision,degree,pmax2,@dempoly,fcostdem(i), ...
                     'dem',fdem,coeffs_n,[pmax2 pmax], ...
                     varargin{param(n+1,1):param(n+1,2)}, ...
                     'cost',varargin{param(i,1):param(i,2)});
end

% Finds equilibrium prices.
sumcoeffs = sum(coeffs);
sumcoeffs(degree) = sumcoeffs(degree) - 1;
equilsum = roots(sumcoeffs);
equilsum = equilsum(equilsum >= max(minsum));
equilsum = equilsum(equilsum <= min(maxsum));
if length(equilsum)==0
    msgbox(' No equilibrium combination of prices found. ', ...
          ' No solution. ', 'warn');
    return
end
if length(equilsum)>1
    msgbox(' Multiple equilibria found, the one with lowest prices is shown.', ...
          ' Multiple equilibria. ', 'warn');
    equilsum = min(equilsum);
end
for i = 1:n-1
    price(i) = polyval(coeffs(i,:),equilsum);
end
price(n) = polycut(equilsum,coeffs_n,pmax2,pmax);

% Calculates equilibrium quantity demanded and profits.
demand = feval(fdem,price(n),varargin{param(n+1,1):param(n+1,2)});
for i = 1:n-1
    profit(i) = demand*price(i) - ...
        feval(fcostdem(i),demand,varargin{param(i,1):param(i,2)});
end
profit(n) = demand*(price(n) - equilsum) - ...
    feval(fcostdem(n),demand,varargin{param(n,1):param(n,2)});
profitshare = profit/sum(profit);

% Calculates CSC's optimal price, profit and quantity demanded,
% and DSC's efficiency.
[priceCSC profCSC] = fminbnd(@profit,0,1.05*pmax,[],0,fdem,@costsum, ...
                             'dem', varargin{param(n+1,1):param(n+1,2)}, ...
                             'cost', fcostdem(1:n),varargin{:});
profCSC = -profCSC;
demCSC = feval(fdem,priceCSC,varargin{param(n+1,1):param(n+1,2)});
eff = sum(profit)/profCSC;
pmax = pmax*ones(1,n);

```

---

```
for i = 1:n-1
    pmax(i) = pmax2 - equilsum + price(i);
end;
```

---

### Funkce *equilpard*

```
function [price,profit,profitshare,demand,eff,priceCSC,profCSC,demCSC] = ...
    equilp(ploton,pmax,fcostdem,varargin);
% EQUILPARD Finds equilibrium in a serial DSC with discrete prices.
%
% EQUILPARD(precision,degree,ploton,pmax,fcostdem,varargin) calculates
% equilibrium characteristics in a serial DSC with discrete (i.e.,
% integer) prices and, if desired, displays results in the form
% of graphs. The syntax and output is the same as that of EQUILPAR
% function (the only difference between the two functions is that
% unlike EQUILPAR, EQUILPARD deals with integer prices only).
%
% See also: EQUILPAR,EQUILD.

% Parameter analysis.
n = length(fcostdem) - 1;
param = varargin2arr(n,varargin{:});
fdem = fcostdem(n+1);

% Variable initialization.
price = zeros(1,n);
equilp = zeros(1,2*n);
if pmax ~= fix(pmax), pmax = fix(pmax)+1; end
mapn = zeros(1,pmax);
map_n = cell(1,pmax);
maxcount_n = zeros(1,pmax);
map = cell(n-1,pmax);
maxcount = zeros(n-1,pmax);
maxprofs = zeros(1,pmax);
o = ones(1,pmax);
p = 1:pmax;

% Finds price mapping and equilibrium prices.
dem = feval(fdem,1:pmax,varargin{param(n+1,1):param(n+1,2)});
cost = feval(fcostdem(n),dem,varargin{param(n,1):param(n,2)});
profs = (dem*o).*([1:pmax]'*o-o'*[1:pmax])-cost*o;
[maxprofs,mapn] = max(profs);
mapn(maxprofs==0) = pmax;
for k = 1:n-1 % k is an agent's number.
    dem = feval(@demmappar,fdem,mapn,...
        varargin{param(n+1,1):param(n+1,2)});
    cost = feval(fcostdem(k),dem,varargin{param(k,1):param(k,2)});
    profs = dem.*([1:pmax]'*o) - cost;
    maxprofs = max(profs); % Profit maximization.
    for i = 1:pmax % i denotes the sum of other 1-st level agents here.
        if maxprofs(i)~=0
            % "pkopt" is the optimal price of k-th agent given the sum i.
            pkopt = p(profs(:,i)==maxprofs(i));
            for j = 1:length(pkopt)
                map{k,i + pkopt(j)}(end+1) = pkopt(j);
            end
        end
    end
    end
    for i = 1:pmax
        maxcount(k,i) = length(map{k,i});
        if ~isempty(map{k,i}) & k == 2
            disp(['Soucet cen clanku 1. urovne: ' int2str(i) ...
                ' Cena/y clanku 2 : ' mat2str(map{k,i}) ]);
        end
    end
end
end
for i = 1:pmax
```

---

---

```

if sum(maxcount(:,i)==zeros(n-1,1))>0,continue,end % No profit, no eq.
leng = [maxcount(:,i)' 1];
ind = zeros(1,n-1);
for j = 1:prod(leng) % Loops across all combinations of prices pkopt.
    pricesum = 0;
    for k = 1:n-1
        if j==1
            ind(k)=1;
        else
            ind(k)=mod(fix((j-1)/prod(leng(k+1:end))), (leng(k))) + 1;
        end
        pricesum = pricesum + map{k,i}(ind(k));
    end
    if pricesum == i % Equilibrium found.
        eqsz = size(equilp);
        equilp(eqsz(1)+1,:) = zeros(1,2*n);
        for k = 1:n-1
            equilp(end,k) = map{k,i}(ind(k));
        end
        equilp(end,n) = mapn(pricesum);
        equilp(end,n+1:2*n-1) = ind;
        equilp(end,2*n) = 1;
    end
end
end

% Calculates DSC's equilibrium quantity demanded and profits, CSC's
% optimal price, profit and quantity demanded, and DSC's efficiency.
if size(equilp,1) == 1 % No equilibrium found.
    price = 0;
    profit = 0;
    profitshare = 0;
    demand = 0;
    eff = 0;
    priceCSC = 0;
    profCSC = 0;
    demCSC = 0;
    return
else
    equilp = equilp(2:end,:);
    match = [];
    for i = 1:size(equilp,1)-1 % Check for identical equilibria.
        for j = i+1:size(equilp,1)
            if isequal(equilp(i,1:n),equilp(j,1:n)) & ...
                equilp(i,2*n) == equilp(j,2:n)
                match(end+1)=j;
            end
        end
    end
    for i = length(match):-1:1
        equilp(match(i),:) = [];
    end
    price = equilp;
    equilp = equilp(:,1:n);
    demand = feval(fdem,equilp(:,n),varargin{param(n+1,1):param(n+1,2)});
    for k = 1:n-1
        profit(:,k) = demand.*equilp(:,k) - ...
            feval(fcostdem(k),demand,varargin{param(k,1):param(k,2)});
    end
    profit(:,n) = demand.*(equilp(:,n) - sum(equilp(:,1:n-1))') - ...
        feval(fcostdem(n),demand,varargin{param(n,1):param(n,2)});
    profitshare = profit./(sum(profit')'*ones(1,n));
    [profCSC priceCSC] = max(feval(@profit,[1:pmax],0,fdem,@costsum, ...
        'dem', varargin{param(n+1,1):param(n+1,2)}, ...
        'cost', fcostdem(1:n),varargin{:}));
    demCSC = feval(fdem,priceCSC,varargin{param(n+1,1):param(n+1,2)});
    eff = (sum(profit')/profCSC)';
end

```

---

### Funkce *polycut*

```
function price = polycut(pin,coeffs,pmaxin,pmaxout)
% POLYCUT Evaluates a cut-off polynom for price functions.
%
% POLYCUT(pin,coeffs,pmaxin,pmaxout) returns the value of
% the polynom specified by "coeffs", if both the input price "pin"
% is less than the cut-off bound "pmaxin" and the result is less than
% the cut-off value "pmaxout". Otherwise, the output is "pmaxout".

price = ones(size(pin));
isunder = find(pin<pmaxin);
isover = find(pin>=pmaxin);

% If input price exceeds "pmaxin" cut-off limit, return "pmaxout";
if ~isempty(isover)
    price(isover) = pmaxout;
end

% Evaluate specified polynom. If the result exceeds "pmaxout",
% drop it down to "pmaxout".
if ~isempty(isunder)
    price(isunder) = min(pmaxout,polyval(coeffs,pin(isunder)));
end
```

---

### Funkce *pricepoly*

```
function [polycoeffs,pmax_imin1,p_imin1,p_i,profit_i] ...
    = pricepoly(precision,degree,pmax,fdem,fcost,varargin)
% PRICEPOLY Finds a cut-off polynom approximating a price function in a SC.
%
% PRICEPOLY(precision,degree,pmax,fdem,fcost,varargin) finds
% coefficients of a cut-off polynom approximating the relation
% between the output price (denoted "p_i") of an agent in a SC
% and his input price ("p_imin1"), given that the demand function
% for the agent's production and the cost function are given by
% the function handles "fdem", "fcost", and additional parameters
% in "varargin".
%
% The "precision" parameter determines the number of points at
% which the price function will be evaluated for the sake of fitting,
% "degree" is the degree of the fitting polynom; the "pmax" argument
% is the minimum price of output at which no units are sold.
% "pmax_imin1" is the cut-off price, i.e. the lowest price of input
% that makes the agent choose price "pmax". For more information on
% cut-off polynoms, see POLYCUT.
%
% Example syntax:
%
% [polycoeffs,pmax(n-1)] = pricepoly(precision,degree,pmax(n), ...
%                                     @dem_abda,@cost_pow, ...
%                                     'dem',a,b,d,alpha, ...
%                                     'cost',mult,pow);
%
% See also: POLYCUT, EQUIL, DEM_ABDA, COST_POW.

% Determines precision parameters.
increment = pmax/precision;
roundcoeff = 1 - 1/(16*precision);

% Initializes variables.
p_i = zeros(1,precision + 11);
profit_i = zeros(1,precision + 11);
p_imin1 = increment*(0:precision + 10);

% Evaluates price function at equidistant discrete points.
```

---

---

```

for j = 1:(precision+1)
    [p_i(j),profit_i(j)] = ...
        fminbnd(@profit,0,1.05*pmax,[],p_imin1(j),fdem,fcost,varargin{:});
    if p_i(j) > roundcoeff*pmax | profit_i(j) >= 0
        p_i(j) = pmax;
        profit_i(j) = 0;
    end
end

% Increases precision (10 times as dense) around the (first) cut-off point.
indexes = 1:(precision+11);
cutoff = min(indexes(p_i == pmax));
if cutoff > 1;
    p_i((cutoff+10):(precision+11)) = p_i(cutoff:(precision+1));
    p_imin1((cutoff+10):(precision+11)) = p_imin1(cutoff:(precision+1));
    profit_i((cutoff+10):(precision+11)) = profit_i(cutoff:(precision+1));
    p_imin1(cutoff:(cutoff+9)) = ones(1,10)*p_imin1(cutoff-1) + ...
        (increment/11)*(1:10);
    for j = cutoff:(cutoff+9)
        [p_i(j),profit_i(j)] = ...
            fminbnd(@profit,0,1.1*pmax,[],p_imin1(j),fdem,fcost,varargin{:});
        if p_i(j) > roundcoeff*pmax | profit_i(j) >= 0
            p_i(j) = pmax;
            profit_i(j) = 0;
        end
    end
end

cutoff = min(indexes(p_i == pmax)) - 1; % Finds new cut-off index.
polycoeffs = polyfit(p_imin1(1:cutoff),p_i(1:cutoff),degree); % Fits a polynom.
if polycoeffs == zeros(1,degree+1)
    polycoeffs = [zeros(1,degree) pmax];
end
pmax_imin1 = p_imin1(cutoff + 1); % Returns the cut-off price.
profit_i = -profit_i; % Makes profit positive.

```

---

### Funkce *pricepolypar*

```

function [polycoeffs,minsum,maxsum,pricesum,price,prof] ...
    = pricepolypar(precision,degree,pmax,fdem,fcost,varargin)
% PRICEPOLYPAR Approximates 1st-level price functions in a parallel SC.
%
% PRICEPOLYPAR(precision,degree,pmax,fdem,fcost,varargin) finds
% coefficients of a cut-off polynom approximating the relation
% between the output price (denoted "price") of an agent in the 1st
% level of a parallel SC and the sum of the prices charged by other
% 1st-level agents ("pricesum"), given that the demand function
% for the agent's cost function and the demand for the aggregate
% output of the 1st level are given by the function handles "fdem",
% "fcost", and additional parameters in "varargin".
%
% The "precision" parameter determines the number of points at
% which the price function will be evaluated for the sake of fitting,
% "degree" is the degree of the fitting polynom; the "pmax" argument
% is the minimum price of output at which no units are sold.
% "pmax_imin1" is the cut-off price, i.e. the lowest price of input
% that makes the agent choose price "pmax". For more information on
% cut-off polynoms, see POLYCUT.
%
% Example syntax:
%
% [polycoeffs,pmax(n-1)] = pricepoly(precision,degree,pmax(n), ...
%     @dem_abda,@cost_pow, ...
%     'dem',a,b,d,alpha, ...
%     'cost',mult,pow);
%
% See also: PRICEPOLY, POLYCUT, EQUILPAR.

```

---

```
% Determines precision parameters.
increment = pmax/precision;
roundcoeff = 1 - 1/(16*precision);

% Initializes variables.
price = zeros(1,precision+1);
prof = zeros(1,precision+1);
pricesum = increment*(0:precision);
cutoff = 0;

% Evaluates price function at equidistant discrete points, then increases
% precision around the cutoff point (see PRICEPOLY).
for j = 1:(precision+1)
    [price(j) prof(j)] = ...
        fminbnd(@profitpar,0,max(eps,1.01*(pmax-pricesum(j))),[],pricesum(j), ...
            fdem,fcost,varargin{:});
    if price(j)+pricesum(j) > roundcoeff*pmax | prof(j) >= 0
        price(j) = pmax;
        prof(j) = 0;
        cutoff = j;
        break;
    end
end
if cutoff == 1
    msgbox(' No equilibrium prices, production cost too high.', ...
        ' No solution!', 'warn');
    return
end
if cutoff == 0
    cutoff = precision+2;
else
    price = [price(1:cutoff-1) zeros(1,10)];
    prof = [prof(1:cutoff-1) zeros(1,10)];
    pricesum = [pricesum(1:cutoff-1) zeros(1,10)];
    pricesum(cutoff:(cutoff+9)) = ...
        ones(1,10)*pricesum(cutoff-1) + (increment/10)*(1:10);
    for j = cutoff:(cutoff+9)
        [price(j) prof(j)] = ...
            fminbnd(@profitpar,0,max(eps,1.01*(pmax-pricesum(j))),[], ...
                pricesum(j),fdem,fcost,varargin{:});
        if price(j)+pricesum(j) > roundcoeff*pmax | prof(j) >= 0
            price(j) = pmax;
            prof(j) = 0;
            cutoff = j;
            break;
        end
    end
end
end

price = price(1:cutoff-1);           % \
pricesum = pricesum(1:cutoff-1);    % } Selects relevant results.
prof = prof(1:cutoff-1);           % /
pricesum = pricesum + price;        % Returns the sums of all 1st-level prices.
maxsum = pricesum(cutoff-1);        % Returns the maximum sum.
minsum = pricesum(1);               % Returns the minimum sum.
prof = -prof;                       % Makes profit positive.
if pricesum(1) ~= min(pricesum),msgbox('pricepolypar: Slope < -1 !'),end

% Approximates the price function.
polycoeffs = polyfit(pricesum,price,degree);
if polycoeffs == zeros(1,degree+1)
    polycoeffs = [zeros(1,degree) pmax];
end
```

---

## Funkce *pricing*

```
function a = pricing(a,time)
% PRICING Agents' pricing logic.

% Parameters:
weight1 = 0.3; % Slope adjustment rate.
weight2 = 0.8; % Position adjustment rate.
weight3 = 0.1; % Price adjustment rate.

switch time
    case 1 % First round (a priori information only).

    case 2 % Second round (little empiric information).
        p = a.priceout;
        q = a.demand;
        c = a.fdem{1};
        p(2) = (p - q/c(1)+a.pmax(1))/2;
        q(2) = 0;
        a.fdem{2} = lfit2(p,q);
        a.pmax(2) = a.pmax(1);

    otherwise % Remaining rounds.

        % Fit last two observations.
        p = a.priceout(time-2:time-1);
        q = a.demand(time-2:time-1);
        c_old = a.fdem{time-1};
        c_new = lfit2(p,q);

        % Weight corrections.
        corr_time = 1 + sqrt(3/time); % Offset corr_slope in early stages.
        if c_new(1) >= 0 % Make sure there's no upward slope,
            corr_slope = 0;
        else % Benefit close slope ratios (inertia).
            corr_slope = sqrt(min(c_old(1)/c_new(1),c_new(1)/c_old(1)));
        end

        % Ignore repeated observations.
        corr_dist = sqrt(dist(p,q)/sqrt(var(a.demand,1)+var(a.priceout,1)));
        % Combine corrections.
        corr = min(1,corr_time*corr_slope*corr_dist);

        % Slope and position of the new demand curve estimate.
        a.fdem{time} = (1-corr*weight1)*c_old + corr*weight1*(c_new);
        q_new = (1-weight2)*polyval(a.fdem{time},p(2)) + weight2*q(2);
        a.fdem{time}(2) = lfit1(p(2),q_new,a.fdem{time}(1));
        a.pmax(time) = a.pmax(time-1);
    end

% Optimize estimated profit.
pricein = a.pricein(time);
priceout = fminbnd(@profit,0,a.pmax(time),[],pricein,@dem_poly,a.fcost{1},...
    'dem',a.fdem{time},'cost',a.fcost{2:end});

% Check the maximum allowable price change.
if time > 1
    diff = priceout - a.priceout(time-1);
    if abs(diff/a.priceout(time-1)) > weight3
        priceout = a.priceout(time-1)*(1 + sign(diff)*weight3);
    end
end
a.priceout(time) = min(a.pmax(time),max(priceout,pricein));
```

---

## Funkce *pricingpar*

```
function a = pricingpar(a,time)
```

---

```
% PRICINGPAR Pricing logic of a 1st-level agent of a parallel SC.
%
% The difference between PRICING and PRICINGPAR is only the
% re-estimation of "a.pmax" in PRICINGPAR. This is because in a
% parallel SC, the maximum allowable price that yields a positive
% demand cannot be stated a priori, since it depends on the prices
% set by other agents.

% Parameters:
weight1 = 0.3; % Slope adjustment rate.
weight2 = 0.8; % Position adjustment rate.
weight3 = 0.1; % Price adjustment rate.

switch time
case 1 % First round (a priori information only).

case 2 % Second round (little empiric information).
    p = a.priceout;
    q = a.demand;
    c = a.fdem{1};
    p(2) = (p - q/c(1)+a.pmax(1))/2;
    q(2) = 0;
    a.fdem{2} = lfit2(p,q);
    a.pmax(2) = p(2);

otherwise % Remaining rounds.

    % Fit last two observations.
    p = a.priceout(time-2:time-1);
    q = a.demand(time-2:time-1);
    c_old = a.fdem{time-1};
    c_new = lfit2(p,q);

    % Weight corrections.
    corr_time = 1 + sqrt(3/time); % Offset corr_slope in early stages.
    if c_new(1) >= 0 % Make sure there's no upward slope,
        corr_slope = 0;
    else % Benefit close slope ratios (inertia).
        corr_slope = sqrt(min(c_old(1)/c_new(1), c_new(1)/c_old(1)));
    end

    % Ignore repeated observations.
    corr_dist = sqrt(dist(p,q)/sqrt(var(a.demand,1)+var(a.priceout,1)));
    % Combine corrections.
    corr = min(1, corr_time*corr_slope*corr_dist);

    % Slope and position of the new demand curve estimate.
    a.fdem{time} = (1-corr*weight1)*c_old + corr*weight1*(c_new);
    q_new = (1-weight2)*polyval(a.fdem{time},p(2)) + weight2*q(2);
    a.fdem{time}(2) = lfit1(p(2),q_new,a.fdem{time}(1));
    a.pmax(time) = -a.fdem{time}(2)/a.fdem{time}(1);
end

% Optimize estimated profit.
pricein = a.pricein(time);
priceout = fminbnd(@profit,0,a.pmax(time),[],pricein,@dem_poly,a.fcost{1},...
    'dem',a.fdem{time},'cost',a.fcost{2:end});

% Check the maximum allowable price change.
if time > 1
    diff = priceout - a.priceout(time-1);
    if abs(diff/a.priceout(time-1)) > weight3
        priceout = a.priceout(time-1)*(1 + sign(diff)*weight3);
    end
end
a.priceout(time) = max(priceout,pricein);
```

## Funkce *pricingpar2*

```

function a = pricingpar2(a,time)
% PRICINGPAR2 Pricing logic of a 1st-level agent of a parallel SC.
%
% Unlike PRICINGPAR, PRICINGPAR2 deals with agents who possess the
% following information about their environment: they know they are
% in the same level of a supply chain as some other agents and they
% know the sum of output prices of these agents in the previous rounds.

% Parameters:
ploton = 0;
weight1 = 0.3; % Slope adjustment rate.
weight2 = 0.8; % Position adjustment rate.
weight3 = 0.1; % Price adjustment rate.

switch time
case 1 % First round (a priori information only).

case 2 % Second round (little empiric information).
    p = a.priceout + a.pricein(2);
    q = a.demand;
    c = a.fdem{1};
    p(2) = (p - q/c(1) - c(2)/c(1))/2;
    q(2) = 0;
    a.fdem{2} = lfit2(p,q);
    a.pmax(2) = a.pmax(1);

otherwise % Remaining rounds.

    % Fit last two observations.
    p = a.priceout(time-2:time-1) + a.pricein(time-1:time);
    q = a.demand(time-2:time-1);
    c_old = a.fdem{time-1};
    c_new = lfit2(p,q);

    % Weight corrections.
    corr_time = 1 + sqrt(3/time); % Offset corr_slope in early stages.
    if c_new(1) >= 0 % Make sure there's no upward slope,
        corr_slope = 0;
    else % Benefit close slope ratios (inertia).
        corr_slope = sqrt(min(c_old(1)/c_new(1), c_new(1)/c_old(1)));
    end

    % Ignore repeated observations.
    corr_dist = sqrt(dist(p,q)/sqrt(var(a.demand,1) + ...
        var(a.priceout + a.pricein(2:end),1)));
    % Combine corrections.
    corr = min(1,corr_time*corr_slope*corr_dist);

    % Slope and position of the new demand curve estimate.
    a.fdem{time} = (1-corr*weight1)*c_old + corr*weight1*(c_new);
    q_new = (1-weight2)*polyval(a.fdem{time},p(2)) + weight2*q(2);
    a.fdem{time}(2) = lfit1(p(2),q_new,a.fdem{time}(1));
    a.pmax(time) = a.pmax(time-1);
end

% Optimize estimated profit.
pricein = a.pricein(time);
priceout = fminbnd(@profitpar,0,max(0,a.pmax(time)-pricein),[],pricein,...
    @dem_poly,a.fcost{1},'dem',a.fdem{time},'cost',a.fcost{2:end});

% Check the maximum allowable price change.
if time > 1
    diff = priceout - a.priceout(time-1);
    if abs(diff/a.priceout(time-1)) > weight3
        priceout = a.priceout(time-1)*(1 + sign(diff)*weight3);
    end
end

```

```
end
a.priceout(time) = max(priceout,0);
```

---

### Funkce *simul*

```
function [prices,profits,eqprice,eqprof] = ...
    simul(precision,degree,ploton,rounds,qmax,pmax,fcostdem,varargin);
% SIMUL Simulates price and profit dynamics in a serial DSC.
%
% SIMUL(precision,degree,ploton,rounds,qmax,pmax,fcostdem,varargin)
% simulates dynamics of prices and profits in a serial DSC with
% (semi-)intelligent agents possessing severely limited information.
% SIMUL first creates a multi-agent system (a collection of 'agent'
% class objects) with properties specified in input parameters and
% then carries out the simulation. The agent's rationing (pricing
% logic) is contained in the PRICING function. SIMUL uses the same
% structure of input arguments to capture the properties of a SC as
% EQUIL function (see EQUIL function), with two additional parameters,
% "rounds" and "qmax". The "rounds" parameter sets the length of the
% simulation run, "qmax" determines the agents' first assumptions
% about the slope of the agent-specific demand function.
%
% Example syntax:
%
% [prices,profits] = simul(precision,degree,ploton,rounds,pmax,qmax...
%                         [@cost_lin @cost_poly @cost_pow @dem_abda],...
%                         'dem',a,b,d,alpha,...
%                         'cost1',unit_cost1,...
%                         'cost2',[c21 c22 c23],...
%                         'cost3',mult,pow);
%
% See also: EQUIL, AGENT, PRICING, SIMULPAR.

% Parameter analysis.
n = length(fcostdem) - 1;
param = varargin2arr(n,varargin{:});
fdem = fcostdem(n+1);

% Retrieve equilibrium characteristics.
[eqprice,eqprof,dummy1,dummy2,dummy3,dummy4,dummy5,dummy6,pmax] = ...
    equil(precision,degree,0,pmax,fcostdem,varargin{:});
clear dummy*;

% Initiate agents.
for j = 1:n
    if length(qmax) == 1
        demcoeffs = lfit2([pmax(j) 0],[0 qmax*j/n]);
    else
        demcoeffs = lfit2([pmax(j) 0],[0 qmax(j)]);
    end
    such(j) = agent(['Agent ' int2str(j)]);

    end

    such(j) = set(such(j),'fdem',1,demcoeffs);
    such(j) = set(such(j),'fcost',fcostdem(j),varargin{param(j,1):param(j,2)});
end

% Simulate dynamics
for i = 1:rounds
    such(1) = set(such(1),'pricein',i,0);
    such(1) = pricing(such(1),i);
    for j = 2:n
        such(j) = set(such(j),'pricein',i,get(such(j-1),'priceout',i));
        such(j) = pricing(such(j),i);
    end
end
```

---

---

```

    dem(i) = feval(fdem,get(such(n),'priceout',i), ...
        varargin{param(n+1,1):param(n+1,2)});
    for j = 1:n
        such(j) = set(such(j),'demand',i,dem(i));
    end
end

for j = 1:n
    such(j) = calcprofits(such(j));
end

prices = zeros(n,rounds);
profits = zeros(n,rounds);
for j = 1:n
    prices(j,:) = get(such(j),'priceout');
    profits(j,:) = get(such(j),'profit');
end

```

---

### Funkce *simulpar*

```

function [prices,profits,eqprice,eqprof] = ...
    simulpar(precision,degree,ploton,rounds,qmax,pmax,fcostdem,varargin);
% SIMULPAR Simulates dynamics in a parallel DSC with no information.
%
% SIMULPAR(precision,degree,ploton,rounds,qmax,pmax,fcostdem,varargin)
% simulates dynamics of prices and profits in a parallel DSC with
% (semi-)intelligent agents possessing severely limited information.
% SIMULPAR first creates a multi-agent system (a collection of "agent"
% class objects) with properties specified in input parameters and
% then carries out the simulation. The agent's rationing (pricing
% logic) is contained in the PRICING function. SIMULPAR uses the same
% structure of input arguments to capture the properties of a SC as
% EQUIL function (see EQUIL function), with two additional parameters,
% "rounds" and "qmax". The "rounds" parameter sets the length of the
% simulation run, "qmax" determines the agents' first assumptions
% about the slope of the agent-specific demand function.
%
% Example syntax:
%
% [prices,profits] = simul(precision,degree,ploton,rounds,pmax,qmax...
%     [@cost_lin @cost_poly @cost_pow @dem_abda],...
%     'dem',a,b,d,alpha,...
%     'cost1',unit_cost1,...
%     'cost2',[c21 c22 c23],...
%     'cost3',mult,pow);
%
% See also: EQUIL, AGENT, PRICING, SIMULPAR.

% Parameter analysis.
n = length(fcostdem) - 1;
param = varargin2arr(n,varargin{:});
fdem = fcostdem(n+1);
weight = 1 - 1/(n-1);

% Retrieve equilibrium characteristics.
[eqprice,eqprof,dummy1,dummy2,dummy3,dummy4,dummy5,dummy6,pmax] = ...
    equilpar(precision,degree,0,pmax,fcostdem,varargin{:});
clear dummy*;

% Initiate agents.
for j = 1:n
    pmax(j) = pmax(j) + weight*(1-fix(j/n))*(eqprice(j)-pmax(j));
    if length(qmax) == 1
        q = qmax*(1 + fix(j/n))/2;
        demcoeffs = lfit2([pmax(j) 0],[0 q]);
    else

```

---

```
        demcoeffs = lfit2([pmax(j) 0],[0 qmax(j)]);
    end
    such(j) = agent(['Agent ' int2str(j)]);
    such(j) = set(such(j),'pmax',1,pmax(j));
    such(j) = set(such(j),'fdem',1,demcoeffs);
    such(j) = set(such(j),'fcost',fcostdem(j),varargin{param(j,1):param(j,2)});
end

% Simulate dynamics
for i = 1:rounds
    sumprice = 0;
    for j = 1:n-1
        such(j) = set(such(j),'pricein',i,0);
        such(j) = pricingpar(such(j),i);
        sumprice = sumprice + get(such(j),'priceout',i);
    end
    such(n) = set(such(n),'pricein',i,sumprice);
    such(n) = pricing(such(n),i);
    dem(i) = feval(fdem,get(such(n),'priceout',i), ...
        varargin{param(n+1,1):param(n+1,2)});
    for j = 1:n
        such(j) = set(such(j),'demand',i,dem(i));
    end
end

for j = 1:n
    such(j) = calcprofits(such(j));
end

prices = zeros(n,rounds);
profits = zeros(n,rounds);
for j = 1:n
    prices(j,:) = get(such(j),'priceout');
    profits(j,:) = get(such(j),'profit');
end
```

---

### Funkce *simulpar2*

```
function [prices,profits,eqprice,eqprof] = ...
    simulpar2(precision,degree,ploton,rounds,qmax,pmax,fcostdem,varargin);
% SIMULPAR2 Simulates dynamics in a parallel DSC with limited information.
%
% SIMULPAR2(precision,degree,ploton,rounds,qmax,pmax,fcostdem,varargin)
% simulates dynamics of prices and profits in a parallel DSC with
% (semi-)intelligent agents possessing severely limited information.
% SIMULPAR first creates a multi-agent system (a collection of "agent"
% class objects) with properties specified in input parameters and
% then carries out the simulation. The agent's rationing (pricing
% logic) is contained in the PRICINGPAR function. SIMULPAR uses the same
% structure of input arguments to capture the properties of a SC as
% EQUIL function (see EQUIL function), with two additional parameters,
% "rounds" and "qmax". The "rounds" parameter sets the length of the
% simulation run, "qmax" determines the agents' first assumptions
% about the slope of the agent-specific demand function.
%
% Example syntax:
%
% [prices,profits] = simul(precision,degree,ploton,rounds,pmax,qmax...
%     [@cost_lin @cost_poly @cost_pow @dem_abda],...
%     'dem',a,b,d,alpha,...
%     'cost1',unit_cost1,...
%     'cost2',[c21 c22 c23],...
%     'cost3',mult,pow);
%
% See also: EQUIL, AGENT, PRICINGPAR, SIMULPAR.
%
% Parameter analysis.
```

---

```

n = length(fcostdem) - 1;
param = varargin2arr(n,varargin{:});
fdem = fcostdem(n+1);

% Retrieve equilibrium characteristics.
[eqprice,eqprof,dummy1,dummy2,dummy3,dummy4,dummy5,dummy6,pmax] = ...
    equilpar(precision,degree,0,pmax,fcostdem,varargin{:});
clear dummy*;
equilsum = sum(eqprice(1:n-1));
pmax2 = sum(eqprice(2:n-1))+pmax(1);

% Initiate agents.
for j = 1:n
    if j==n, p = pmax(n);,else p = pmax2;,end
    if length(qmax) == 1
        q = qmax*(1 + fix(j/n))/2;
        demcoeffs = lfit2([p 0],[0 q]);
    else
        demcoeffs = lfit2([p 0],[0 qmax(j)]);
    end
    such(j) = agent(['Agent ' int2str(j)]);
    if j==n
        such(j) = set(such(j),'pmax',1,pmax(n));
    else
        such(j) = set(such(j),'pmax',1,pmax2);
        such(j) = set(such(j),'pricein',1,(0.9+rand/2)*(equilsum-eqprice(j)));
    end
    such(j) = set(such(j),'fdem',1,demcoeffs);
    such(j) = set(such(j),'fcost',fcostdem(j),varargin{param(j,1):param(j,2)});
end
dem = zeros(1,rounds);

% Simulate dynamics.
% First round.
sumprice = 0;
for j = 1:n-1
    such(j) = pricingpar2(such(j),1);
    sumprice = sumprice + get(such(j),'priceout',1);
end
such(n) = set(such(n),'pricein',1,sumprice);
such(n) = pricing(such(n),1);
dem(1) = feval(fdem,get(such(n),'priceout',1), ...
    varargin{param(n+1,1):param(n+1,2)});
for j = 1:n
    such(j) = set(such(j),'demand',1,dem(1));
end
% Following rounds.
for i = 2:rounds
    pricein = sumprice;
    sumprice = 0;
    for j = 1:n-1
        such(j) = set(such(j),'pricein',i,pricein-get(such(j),'priceout',i-1));
        such(j) = pricingpar2(such(j),i);
        if mod(i,2) == mod(j,2)
            such(j) = set(such(j),'priceout',i,get(such(j),'priceout',i-1));
        end
        sumprice = sumprice + get(such(j),'priceout',i);
    end
    such(n) = set(such(n),'pricein',i,sumprice);
    such(n) = pricing(such(n),i);
    dem(i) = feval(fdem,get(such(n),'priceout',i), ...
        varargin{param(n+1,1):param(n+1,2)});
    for j = 1:n
        such(j) = set(such(j),'demand',i,dem(i));
    end
end
for j = 1:n-1

```

---

```
        such(j) = calcprofitspar(such(j));  
end  
such(n) = calcprofits(such(n));  
  
prices = zeros(n,rounds);  
profits = zeros(n,rounds);  
for j = 1:n  
    prices(j,:) = get(such(j),'priceout');  
    profits(j,:) = get(such(j),'profit');  
end
```