Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE

Bc. Ladislav Štěpánek

Ekonomické aplikace geometrického programování

Katedra pravděpodobnosti a matematické statistiky

Vedoucí diplomové práce:	prof. RNDr. Jitka Dupačová DrSc.
Studijní program:	Matematika
Studijní obor:	Pravděpodobnost, matematická statistika a ekonometrie

Praha 2013

Děkuji především paní profesorce Dupačové za trpělivost, kterou se mnou měla při psaní této práce a její poznámky. Dále bych rád poděkoval své rodině za podporu a také všem, kteří mi pomáhali s kontrolou práce.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 2.8.2013 Ladislav Štěpánek

Název práce: Ekonomické aplikace geometrického programování
Autor: Bc. Ladislav Štěpánek
Katedra (ústav): Katedra pravděpodobnosti a matematické statistiky
Vedoucí bakalářské práce: prof. RNDr. Jitka Dupačová DrSc.
e-mail vedoucího: Jitka.Dupacova@mff.cuni.cz

Abstrakt: Úlohy geometrického programování jsou speciální úlohy nelineárního programování, v nichž účelová funkce a omezení jsou ve tvaru posynomů. V této práci představíme úlohu geometrického programování a uvedeme možné způsoby řešení. V poslední kapitole budeme geometrické programování aplikovat na Cobb-Douglasovu produkční funkci, vytvoříme model s náhodnou poptávkou a uvedeme možná rozšíření této úlohy.

Klíčová slova: geometrické programování, optimalizace, ekonometrie, Cobb-Douglasova produkční funkce, náhodná poptávka

Title: Economic applications of geometric programming
Author: Bc. Ladislav Štěpánek
Department: Department of Prebability and Mathematical Statistics
Supervisor: prof. RNDr. Jitka Dupačová DrSc.
Supervisor's e-mail address: Jitka.Dupacova@mff.cuni.cz

Abstract: Geometric programming is a special case of nonlinear programming, where objective function and constraints are shaped as posynomials. In this work we introduce geometric programming and solving methods. In last chapter we will apply the geometric programming to Cobb-Douglas production function, create a model with random demand and possible extensions of this model.

Keywords: geometric programming, optimization, ekonometrics, Cobb-Douglass production function, random demand

Obsah

Kapitola 1

Úvod

Úlohy geometrického programování jsou speciální úlohy nelineárního programování, v nichž účelová funkce a omezení jsou ve tvaru posynomů.Tento tvar umožňuje efektivní numerické řešení a usnadňuje analýzu stability výsledků.

V úloze geometrického programování minimalizujeme (maximalizujeme) hodnotu posynomiální účelové funkce za platnosti posynomiálních omezení. Tato úloha má uplatnění ve financích (produkční funkce), návrhu polovodičových součástek či obvodů, viz. [2], plánování chodu elektráren, plánování lidských zdrojů či odhadu parametrů při logistické regresi (věrohodnostní rovnice tvoří úlohu geometrického programování). Tyto aplikace budou podrobněji rozvedeny v páté kapitole s důrazem na ekonomickou aplikaci.

Pro řešení úlohy lze používat různé metody. Tyto metody jsou podrobněji popsány ve třetí kapitole - řešení úlohy geometrického programování. Tyto algoritmy buďto řeší přímo úlohu geometrického programování, a nebo k ní přidruženou duální úlohu. Úlohu lze také řešit pomocí obecných algoritmů nelineárního programování, nicméně v takovém případě není využita informace o tvaru úlohy, která řešení zjednodušuje.

Ve čtvrté kapitole se budeme věnovat možným rozšířením této úlohy. Není totiž vždy podmínkou, že existuje řešení. Dále se nám může stát, že jednotlivé parametry neznáme exaktně, ale jsou náhodné. V takovém případě je nutné úlohu upravit do deterministického tvaru. Nakonec ne vždy se podaří formulovat úlohu, která je ve standardním posynomiálním tvaru. I tento případ bude v práci zmíněn a čtenář bude odkázán na příslušnou literaturu.

V poslední kapitole se budeme věnovat ekonomické aplikaci geometrického programování na Cobb-Douglasovy produkční funkce. Nejprve představíme řešení nejjednodušší úlohy a dále budeme sestavovat vlastní model produkce s náhodnou poptávkou. Uvedeme několik přístupů jak úlohu převést na deterministický tvar, vyřešíme příslušné úlohy a provede rozbor řešení. Nakonec navrhneme možná rozšíření těchto přístupů nad rámec geometrického programování.

Kapitola 2

Základní definice a věty

2.1 Posynomy

Úloha geometrického programování má účelovou funkci ve tvaru posynomů (posynomials) či mononomů (monomials).

Definice 1. **Posynom** *je funkce více proměnných tvaru:*

$$f(x_1, x_2, \ldots, x_n) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} \cdots x_n^{a_{nk}}$$

kde všechny proměnné x_i a koeficienty c_k jsou **kladná realná čísla** *a exponenty a_{ik} jsou reálná čísla.*

Poznámka. *Posynomy jsou uzavřené vzhledem ke sčítání, vzájemnému násobení a násobení kladným číslem.*

Definice 2. **Mononom** *je funkce více proměnných tvaru:*

$$f(x_1, x_2, \ldots, x_n) = c_k x_1^{a_{1k}} \cdots x_n^{a_{nk}}$$

kde všechny proměnné x_i a koeficienty c_k jsou **kladná realná čísla** *a exponenty a_{ik} jsou reálná čísla.*

Poznámka. *Kladnost proměnných x_i a koeficientů c_k je nezbytná pro řešitelnost úlohy. V případě, že není striktně uvedeno jinak, tak se předpokládá kladnost optimalizovaných proměnných v celé této práci.*

Poznámka. *Vydělíme-li posynom (mononom) monomonem, tak dostaneme opět posynom (mononom). Toto je užitečné, máme-li omezení tvaru $f(\mathbf{x}) \leq g(\mathbf{x})$, kde g je mononom. Pak lze toto omezení převédst na $f(\mathbf{x})/g(\mathbf{x}) \leq 1$.*

2.2 Úloha geometrického programování a její duální úloha

Nyní již můžeme přistoupit k samotné definici úlohy geometrického programování a také zde uvedeme úlohu k ní duální.

Definice 3. Úlohou geometrického programování *rozumíme:*

$$\min\{g_0(\mathbf{x}), g_k(\mathbf{x}) \leq 1, k = 1, 2, \ldots, K, \mathbf{x} \in \mathbb{R}^n_{++}\} \tag{2.1}$$

kde $g_k(\mathbf{x})$ jsou posynomy pro k=0,1, ..., K. Z definice posynomů je $x_i > 0$ pro i = 1, 2, ..., n.

Poznámka. *Symbolem $\mathbb{R}^n_{++}$ je v této práci označován prostor $(0, \infty)^n$. Tento zápis implikuje kladnost všech členů vektoru.*

Poznámka. *V některé literatuře se ještě uvádí, že součástí úlohy geometrického programování mohou být dodatečná omezení $h_j(\mathbf{x}) = 1$, kde h_j jsou mononomy pro j=1,2, ...,J.*

Pro definování duální úlohy si nyní musíme podrobněji rozebrat tvar funkcí $g_k(\mathbf{x})$, které jsou tvaru:

$$g_k(\mathbf{x}) = \sum_{i \in I_k} c_i \prod_{j=1}^{M} x_j^{a_{ij}} = \sum_{i \in I_k} u_i(\mathbf{x}), k = 0, 1, \cdots, K, \tag{2.2}$$

kde $u_i(\mathbf{x})$ jsou mononomy. Označme celkový počet mononomů Q a buď $\{I_k$, k=0,1, ..., K$\}$ rozklad množiny $\{1, 2, \ldots, Q\}$ do K + 1 disjunktních množin. Potom vzhledem ke struktuře úlohy geometrického programování (2.1) můžeme definovat duální úlohu:

Definice 4. Duální úloha geometrického programování*:*

$$\begin{gathered}\max_{\delta,\lambda} \nu(\delta, \lambda) := \textstyle\prod_{i=1}^{Q} (c_i/\delta_i)^{\delta_i} \prod_{k=1}^{K} {\lambda_k}^{\lambda_k} \\ s.t : \textstyle\sum_{i \in I_0} \delta_i = 1, \delta_i \geq 0, \\ \textstyle\sum_{i=1}^{Q} a_{ij}\delta_i = 0, j = 1, 2, \ldots, M, \\ \textstyle\sum_{i \in I_k} \delta_i = \lambda_k, k = 1, 2, \ldots, K. \end{gathered} \tag{2.3}$$

Věta 1. *Mezi optimálním řešením* $\mathbf{x}^*$ *primární úlohy geometrického programování (2.1) a optimálním řešením* δ^**,* λ^* *duální úlohy (2.3) je následující vztah:*

$$\begin{gathered}\delta_i^* = \tfrac{u_i(\mathbf{x}^*)}{g_0(\mathbf{x}^*)} = \tfrac{u_i(\mathbf{x}^*)}{v(\delta^*,\lambda^*)}, i \in I_0 \\ \delta_i^* = \lambda_k^* u_i(\mathbf{x}^*), i \in I_k, k = 1, 2, \ldots, K. \end{gathered} \tag{2.4}$$

2.3 Cobb-Douglasova produkční funkce

Nakonec je nutné definovat Cobb-Douglasovu produkční funkci, kterou se v této práci budeme podrobně zabývat. Motivaci na pozadí této úlohy si vysvětlíme v poslední části této práce.

Definice 5. Cobb-Douglasovou produkční funkcí *rozumíme funkci ve tvaru:*

$$\tilde{u}(\mathbf{x}) = A \prod_{i=1}^{n} x_i^{\lambda_i}, \tag{2.5}$$

kde $\mathbf{x} \in \mathbb{R}^n_{++}$, $A \in \mathbb{R}^n_{++}$ a $\lambda_i > 0$ pro i=1, 2, ..., n.

Poznámka. *Koeficienty λ_i vyjadřují elasticity jednotlivých složek produkční funkce. Tedy o kolik procent se zvýší produkce při zvýšení vstupu o 1%.*

Kapitola 3

Řešení úlohy geometrického programování

Úlohu geometrického programování lze řešit mnoha způsoby. Existují dva hlavní přístupy. Jako první byl rozvinut přístup, který řeší duální úlohu. Tento přístup má tu výhodu, že duální úloha má velmi jednoduchou strukturu s lineárními omezeními. Nicméně tato metoda se potýká s mnoha problémy v případě, že některá omezení nejsou splněna jako rovnost. I v tomto případě existují postupy, jak získat optimální řešení, nicméně vyvstává potřeba řešit "vedlejší" problémy. Proto se mnoho výzkumníků začalo zabývat přímo řešením primární úlohy. V tomto případě se ovšem člověk musí smířit s řešením silně nelineární úlohy. Tyto algoritmy jsou pak založeny na řešení lineárních úloh buďto s nekonečně mnoha omezeními a nebo s nekonečně mnoha proměnnými. Iterativním postupem je pak tato aproximace zpřesňována, dokud se nedosáhne požadované tolerance.

3.1 Řešení pomocí konvexní transformace

Úloha geometrického programování může být díky svým výhodným vlastnostem přeformulována na úlohu konvexního programování. Nejprve zavedeme exponenciální substituci $z_j = \log(x_j)$, ∀j. Tato substituce je možná vzhledem k předpokládané kladnti všech x_j. Tímto jsou posynomy transformovány do tvaru:

$$h_k(\mathbf{z}) = \sum_{i \in I_k} c_i \exp\{\sum_{j=1}^{M} a_{ij} z_j\}, k = 0, 1, \ldots, K. \tag{3.1}$$

Nyní ještě převedeme člen c_i do exponentu:

$$h_k(\mathbf{z}) = \sum_{i \in I_k} \exp\{\log(c_i) + \sum_{j=1}^{M} a_{ij} z_j\}, k = 0, 1, \ldots, K. \tag{3.2}$$

Vzhledem k pozitivnosti všech proměnných je možné v případě monomiálního tvaru provést zlogaritmování na tvar, což dále zjednodušuje zápis úlohy do optimalizačního softwaru:

$$\log(h_k(\mathbf{z})) = \log(c_i) + \sum_{j=1}^{M} a_{ij} z_j, k = 0, 1, \ldots, K. \tag{3.3}$$

Tímto jsem získali úlohu konvexního programování, kterou lze již snadno řešit pomocí běžně dostupného softwaru. K řešení obvykle bývá použita metoda vnitřního bodu. Narozdíl od původní úlohy je transformovaná užitková funkce konvexní a algoritmus vždy najde optimální řešení, případně zjistí, že neexistuje žádné přípustné řešení. Případ kdy neexistuje žádné přípustné řešení je také velmi zajímavý a vede na nový problém nalezení "nejlepšího"přibližného řešení. Tímto případem se také budeme zaobírat v další části práce.

3.2 Řešení pomocí duální úlohy

Další přístup je řešení pomocí duální úlohy. Duální úloha je lineární úloha, kterou lze řešit pomocí běžně dostupného softwaru. Dále je ovšem nutné se zabývat získáním řešní původní primární úlohy. Máme již řešení duální úlohy (λ^*, δ^*), ale jak již bylo naznačeno v úvodu, ne vždy toho lze jednoduše dosáhnout. Hlavně v případě, že některé omezení nebylo splněno jako rovnost. Ze vztahů mezi primární a duální úlohou (2.4) získáváme následující soustavu rovnic, kterou je třeba řešit:

$$c_i \prod_{j=1}^{M} x_j^{a_{ij}} = \delta_i^* \nu(\delta^*, \lambda^*), i \in I_0 \tag{3.4}$$

$$c_i \prod_{j=1}^{M} x_j^{a_{ij}} = \delta_i^* / \lambda_k^*, i \in I_k, k = 1, 2, \ldots, K. \tag{3.5}$$

Pokud jsou všechny λ_k^* a δ_k^* kladná, tak zlogaritmovaním vztahů (3.4) a (3.5) získáme řešitelnou soustavu M rovnic o M neznámých:

$$\sum_{j=1}^{M} a_{ij}\omega_j = \log[\delta_i \nu(\lambda, \delta)/c_i], i \in I_0, \tag{3.6}$$

$$\sum_{j=1}^{M} a_{ij}\omega_j = \log[\delta_i/(\lambda_k c_i)], i \in I_k, k \geq 1, \lambda_k > 0. \tag{3.7}$$

Jak již bylo napsáno, tak největší problémy nastávají v případě, že některé z omezení nebylo splněno jako rovnost. V [5] je ukázáno, že λ_k jsou ekvivalentní s Lagrangovými multiplikátory. Tedy z podmínek kompementarity, když k-té omezení není splněno jako rovnost, tak automaticky musí být $\lambda_k = 0$. Pak také $\delta_i = 0$ pro $i \in I_k$ z definice duální úlohy (2.3). Tímto pak vyvstává problém s tím, že pojmy jako například $(c_i/\delta_i)^{\delta_i}$ a $\lambda_k^{\lambda_k}$ nejsou definovány. Také kvůli numerickému řešení úlohy vyvstává problém s tím, které z λ_k a δ_i jsou nulové. Je tedy nutné nadefinovat parametry, které určí, jestli je již příslušná hodnota "dostatečně blízko" k 0. Podrobnější informace k řešení těchto problémů je možné najít například v publikacích [5] a [3].

3.3 Řešení primární úlohy jako semi-nekonečné lineární úlohy

Hlavním přístupem při řešení složitějších úloh je jejich vhodná aproximace pomocí úloh jednodušších. S myšlenkou lineární aproximace přišel již Duffin a později ji rozvíjel Gochet. Z tohoto principu vycházeli také J. Rajgopal a D.L. Bricker v publikaci [15], ze které je tento způsob řešení úlohy geometrického programování převzat. V této publikaci je také možné najít důkazy příslušných vztahů.

Původní úlohu (2.1) tedy přeformulujeme na úlohu semi-nekonečné lineární úlohy. Nejprve je však nutné zavést konvexní množiny Q_k, které budou pro každé omezení původní úlohy generovat nekonečné množství lineárních omezení:

$$Q_k = \{\rho \in \mathbb{R}^n \mid \rho_i = 0 \text{ pro } i \notin I_k, \sum_{i=1}^{n} \rho_i = \sum_{i \in I_k} \rho_i, \rho_i \geq 0\} \tag{3.8}$$

a také konvexní funkce A_{kj} a G_k:

$$A_{kj}(\rho) = \sum_{i=1}^{n} a_{ij}\rho_i, \rho \in Q_k = \sum_{i \in I_k} a_{ij}\rho_i, \rho \in Q_k \tag{3.9}$$

$$G_k = (\rho) = \sum_{i=1}^{n} \rho_i \log(c_i/\rho_i), \rho \in Q_k = \sum_{i \in I_k} \rho_i \log(c_i/\rho_i), \rho \in Q_k \tag{3.10}$$

Pak lze již definovat úlohy, která je autory označována jako program P:

Definice 6. Program P:

$$\min \quad \omega_0 \tag{3.11}$$

$$s.t: \quad \textstyle\sum_{j=1}^{m} \omega_j A_{0j}(\rho) + \omega_0 \geq G_0(\rho) \;\; pro \;\; \rho \in Q_0, \tag{3.12}$$

$$\textstyle\sum_{j=1}^{m} \omega_j A_{kj}(\rho) \geq G_k(\rho) \;\; pro \;\; \rho \in Q_k, k = 1, 2, \ldots, K. \tag{3.13}$$

Pro každý posynom z původní úlohy je tedy definována parametrická množina Q_k, která obsahuje nekonečně mnoho prvků. Tedy úloha má skutečně nekonečné množství omezení a K+1 proměnných ω. V [15] je pak dokázána následující věta:

Věta 2. *Předpokládejme, že původní úloha GP je konzistentní a dosahuje svého optima v bodě x^*. Pak vektor ω^*, kde ω_0^*=log $g_0(x^*)$ a ω_j^*= -logx_j^* pro j= 1, 2, ..., K je optimální řešení programu P. Opačně je li vector ω^* optimální pro pro program P, pak vektor x^*, kde x_j^*=exp(-ω_j^*) pro j= 1, 2, ..., K je optimální pro původní úlohu GP a hodnota účelové funkce je exp(ω_0^*).*

Tato věta nám tedy dává dohromady vztah mezi původní úlohou a programem P. Nicméně pro řešení programu P autoři použili úlohu k němu duální (program D). Nyní si zavedeme duální úlohu a pak již budeme moci prezentovat samotný algoritmus řešení.

Definice 7. Program D *Cílem je naleznout množinu vektorů $\{\rho^0, \rho^1, \ldots, \rho^K\}$ z množin $Q_0, Q_1, \ldots, Q_K$ a pozitivní vektor $\lambda = [\lambda_0, \lambda_1, \ldots, \lambda_K]^T$ jako řešení úlohy:*

$$\max V(\lambda, \rho) = \textstyle\sum_{k=0}^{K} G_k(\rho^k)\lambda_k \tag{3.14}$$

$$s.t: \textstyle\sum_{k=0}^{K} \lambda_k A_{kj}(\rho^k) = 0, \tag{3.15}$$

$$\lambda_0 = 1, \tag{3.16}$$

$$\lambda_k \geq 0, k = 1, 2, \ldots, K. \tag{3.17}$$

Z tvaru účelové funkce je patrné, že není lineární v ρ^k a je naopak lineární v λ. Tedy algoritmus probíhá ve dvou krocích. Nejprve se generují množiny vektorů ρ a následně je provedena lineární optimalizace programu D. Pro plné pochopení tohoto algoritmu je ještě nutné uvést větu dávající do vztahu řešení programu P a programu D:

Věta 3. *Předpokládejme, že program P dosahuje svého optima v bodě ω^*. Pak je Program D konzistentní a má optimum v bodě ($\lambda^*, \underline{\rho^*}$) a V($\lambda^*, \underline{\rho^*}$)=$\omega_0^*$. Dále vektor $\underline{\rho^*}$, který splňuje*

$$\underline{\rho_i^*} = \frac{c_i exp(\sum_{j=1}^{K} -\omega_j^* a_{ij})}{\sum_{t\in[k]} c_t exp(\sum_{j=1}^{K} -\omega_j^* a_{ij})} \tag{3.18}$$

pro $i \in [k]$ se nazývá superoptimální.

Věta 4. *Vztah mezi řešením programu D a původní úlohy GP je následující:*

$$\underline{\rho_i^*} = \frac{c_i \prod_{j=1}^{K} x_j^{*a_{ij}}}{\sum_{l\in[k]} c_l \prod_{j=1}^{K} x_j^{*a_{lj}}} = \frac{c_i \prod_{j=1}^{K} x_j^{*a_{ij}}}{g_k(x^*)} \tag{3.19}$$

Z věty (3.19) hlavní výhoda tohoto přístupu oproti řešení duální úlohy geometrického programování. A sice, že všechny výrazy ve větě (3.19) jsou vždy korektně definovány vzhledem k tomu, že $\underline{\rho_i^*}$ je striktně pozitivní. Odpadají tak veškeré problémy z předchozího přístupu.

Nyní již máme připraveny všechny věty a definice, abychom mohli uvést algoritmus:

KROK 1 (inicializace): Definujeme toleranci $\varepsilon > 0$ a zkontrolujeme základní aproximaci pro program D. Vyřešíme úlohu a pomocí (3.18) získáme vektor simplexových multiplikátorů $\omega = (\omega_1, \omega_2, \ldots, \omega_K, \omega_0)$.

KROK 2 (kontrola konvergence): Pomoci (3.19) vypočítáme vektor $\mathbf{x}$, kde x_j = exp(-ω_j). Definujeme množinu L = { k | g_k(x) $\geq$ (1 + ε) pro k = 1, 2, ..., K a $g_0 \geq$ exp(ω_k) + ε }. Pokud L = $\emptyset$, pak jsme nalezli optimální řešení. V opačném případě pokračujeme na krok 3.

KROK 3 (iterace): Pro každé k $\in$ L vyhodnotíme vektor ρ^k, jehož i-tý člen je: $\rho_i^k = c_i$ exp($\sum_{j=1}^{K}$ -$\omega_j a_{ij}$) / $\sum_{s\in I_k} c_s$ exp($\sum_{j=1}^{K} -\omega_j a_{sj}$). Poté do úlohy přidáme nové sloupce [$G_k(\rho^k)$, $A_{k1}(\rho^k)$, ..., $A_{kK}(\rho^k)$, σ]^T, kde $G_k(\rho^k) = \sum_{i\in I_k}(\rho_i^k)$log($c_i/\rho_i^k$); $A_{kj}(\rho^k) = \sum_{i\in I_k} a_{ij}\rho_i^k$ pro každé j = 1, 2, ..., K a $\sigma = 1$ pro k=0 a $\sigma = 0$ jinak. Následně opět vyřešíme program D s přidanými sloupci, získáme nový vektor simplexových multiplikátorů $\omega = (\omega_1, \omega_2, \ldots, \omega_K, \omega_0)$ a jdeme zpátky na krok 2.

Algoritmus pro řešení je podrobněji popsán v [16]. Implementovaný algoritmus je také možné přímo stáhnout na *ftp.pitt.edu/dept/ie/GP/.*

3.4 Shrnutí

Při hledání optimálního řešení úlohy geometrického programování máme volbu mezi konvexní transformací, řešením duální úlohy anebo přímým řešením primární úlohy. Řešení konvexní úlohy je jedním z nejjednodušších přístupů. Nicméně při řešení se žádným způsobem nevyužívá speciální tvar úlohy a řešení tak může vyžadovat více výpočetních prostředků než jiné přístupy. Řešení duální úlohy je jednodušší, nicméně v případě přítomnosti omezení, která nejsou splněna jako rovnost, dochází ke komplikacím. Tyto komplikace je sice možné vyřešit, ale je k tomu nutná konstrukce a řešení dalších podúloh, které jsou velmi složité. Při řešení primární úlohy pak vzniká semi-nekonečná úloha. Ta je obecně náročnější na řešení, nicméně nalezneme řešení vždy a není tak nutné řešit podúlohy v případě, že by algoritmus nenašel řešení přímo. Zajímavým námětem pro další práci by bylo porovnání efektivity jednotlivých algoritmů a přístupů k řešení úlohy geometrického programování.

Kapitola 4

Modifikace úlohy geometrického programování

4.1 Existence řešení

Jak již bylo v předchozí kapitole naznačeno, ne vždy musí existovat přípustná řešení úlohy geometrického programování. V případě, že jsou omezení příliš přísná, nemusí existovat žádný vektor $\mathbf{x}$, který je splňuje. V takovém případě je užitečné umět najít bod, který v nějakém smyslu co nejméně daná omezení porušuje. Můžeme tím získat informaci, jakým způsobem při nejmenších "nákladech" upravit omezení tak, aby již nějaké přípustné řešení existovalo.

Prvním přístupem je získat představu o tom, jak moc jsou omezení porušena. Proto formulujeme úlohu:

$$\begin{gathered}\min s, \\ s.t.\ g_k(\mathbf{x}) \leq s, k = 0, 1, \ldots, K, \\ s \geq 1.\end{gathered} \tag{4.1}$$

V případě, že s=1, tak existuje přípustné řešení původní úlohy. V opačném případě získáme informaci o "míře" porušení zadaných omezení.

Druhou velice zajímavou možností je, formulovat úlohu:

$$
\begin{aligned}
\min \textstyle\sum_{k=1}^{K} s_k, \\
s.t.\ g_k(\mathbf{x}) \leq s_k, k = 0, 1, \dots, K, \\
s_k \geq 1, k = 0, 1, \dots, K.
\end{aligned}
\tag{4.2}
$$

Získáme tak bod $\bar{x}^*$ a vektor $s_1, s_2 \dots s_K$. Tento bod $\bar{x}^*$ většinou splňuje většinu omezení a vektor $s_1, s_2 \dots s_K$ nám dává informaci o tom, jakým způsobem uvolnit omezení pro zajištění existence přípustného a tedy i optimálního řešení úlohy.

4.2 Stochastické geometrické programování

V tomto rozšíření připouštíme možnost, že některé konstanty c_i a/nebo a_{ij} nejsou deterministické, ale náhodné. Pro řešení úlohy je pak nutné znát rozdělení těchto náhodných veličin a úlohu transformovat tak, aby se opět jednalo o standartní úlohu geometrického programování. Nejjednodušší případ je, pokud některá c_i v omezeních mají převážně nekorelované normální rozdělení s $c_i \sim N(Ec_i, \sigma_i^2)$. Pak lze příslušné podmínky přepsat do tvaru:

$$
P((\sum_{i \in I_k} c_i \prod_{j=1} t_j^{a_{ij}}) \geq 1 - \varepsilon_k, k = 1, 2, \dots, K, \tag{4.3}
$$

tyto pravděpodobností omezení je pak možné přepsat do tvaru:

$$
\sum_{i \in I_k} \mathrm{E} c_i \prod_{j=1} t_j^{a_{ij}} + \Phi^{-1}_{(1-\varepsilon_k)} \sqrt{\sum_{i \in I_k} \sigma_i^2 \prod_{j=1} t_j^{2a_{ij}}} \leq 1, k = 1, 2, \dots, K. \tag{4.4}
$$

Toto omezení ještě není standartního posynomiálního tvaru, je tedy potřeba zavédst 2 další optimalizační proměnné t_1 a t_2 a rozdělit toto omezení do následujicích 3 omezení:

$$t_1^{-1} \sum_{i \in I_k} \mathrm{E} c_i \prod_{j=1} t_j^{a_{ij}} \leq 1, \tag{4.5}$$
$$t_2^{-1} \sum_{i \in I_k} \sigma_i^2 \prod_{j=1} t_j^{2a_{ij}} \leq 1, \tag{4.6}$$
$$t_1 \Phi^{-1}_{(1-\varepsilon_k)} t_2^{-1/2} \leq 1. \tag{4.7}$$

Mnohem rozšířenějším případem je, že nejsou deterministické exponenty a_{ij}. Například pokud jsme získali produkční funkci jako řešení regresního modelu. V [10] je navrženo následující řešení:

$$\min\{t_0 : t_0^{-1} g_0(t,\omega) \leq 1, g_k(t,\omega) \leq 1, k = 1, 2, \ldots K, t_0 > 0, t \in \mathbb{R}^n_{++}, \omega \in \mathcal{P}\}, \tag{4.8}$$

kde jsme zavedli novou proměnnou t_0 pro transformování účelové funkce do podmínek a $\mathcal{P}$ je předem specifikovaná pravděpodobnostní množina. Také je navrženo rozdělit jednotlivá omezení na jednotlivé mononomy:

$$u_i(t,\omega)\theta_{ik}^{-1} \leq 1, i = 1, 2, \ldots, Q, k = 0, 1, \ldots, K, \tag{4.9}$$
$$\theta_{ik} > 0, \sum_{i \in I_k} \theta_{ik} = 1, i = 1, 2, \ldots, Q, k = 0, 1, \ldots, K, \tag{4.10}$$

kde jednotlivá θ_{ik} vyjadřují poměrný příspěvek i-tého mononomu ke k-tému posynomu. Úloha je pak řešena ve dvou stupních, kde v prvním stupni optimalizujeme $t_0 > 0$ a θ_{ik}. Ve druhém stupni pak optimalizujeme příslušnou úlohu geometrického programování. Omezení v prvním stupni nemusí být splněna, a proto je nutné doplnit úlohu o penalizační funkci za nesplnění těchto omezení. Jako penalizační funkce je v [10] doporučena logaritmická funkce. Je zde také ukázán explicitní tvar v případě normálního či mnohorozměrného diskrétního rozdělení parametrů c_i a a_{ij}.

4.3 Zobecněné geometrické programování

Tento pojem (generalized geometric programming) je v literatuře nejednotný. Nejčastěji je však používán pro úlohu, kerá má v účelové funkci a v omezeních rozdíl dvou posynomů. Tedy úloha je tvaru:

Definice 8. Úlohou zobecněného gemetrického programování *rozumíme:*

$$\min\{g_0^+(\mathbf{x}) - g_0^-(\mathbf{x}), g_k^+(\mathbf{x}) - g_k^-(\mathbf{x}) \leq 0, k = 1, 2, \ldots, K, \mathbf{x} \in \mathbb{R}^n_{++}\} \quad (4.11)$$

kde $g_k^+(\mathbf{x})$ *a* $g_k^-(\mathbf{x})$ *jsou posynomy pro k=0,1, . . . , K.*

Narozdíl od geometrického programování neexistuje žádná transformace primární ani duální úlohy, tak aby byla konvexní. Úloha tedy může mít více lokálních minim ve kterých se běžný algoritmus řešení zastaví. V [13] je prezentován přístup, který pomocí vhodných transformací původní úlohy a řešení série konvexnich úloh nalezne řešení zobecněné úlohy.

Kapitola 5

Aplikace úlohy geometrického programování

Úloha geometrického programování má široké uplatnění především při řešení technických úloh. Abychom čtenáři nabídli větší přehled o problémech, které lze takto modelovat, tak zde uvedeme několik příkladů s odkazem na příslušnou literaturu.

5.1 Řezání materiálů

Každý stroj má svojí životnost v závislosti na provozních podmínkách. Je velmi důležité umět nastavit vhodné pracovní podmínky tak, aby se dosáhlo co největší efektivity stroje v poměru k rychlosti jeho práce. Tato analýza je důležitá zejména pro firmy, které tyto stroje vyrábějí, neboť pak mohou dosáhnout konkurenční výhody při správném nastavení. Toto nastavení je možné zkoumat za běžného provozu a statisticky vyhodnotit. To je ovšem velmi časově náročné, a proto se v [4] zabývali možností modelování nastavení rychlosti řezání a životnosti řezacích strojů. Modelování probíhalo na základě optimalizování procesu vzhledem k rychlosti otáčení, rychlosti řezání, životnosti stroje a efektivity řezání. Do modelu byly také zahrnuty veškeré náklady na provoz stroje a nutnou údržbu v případě nastavení stroje na vyšší pracovní výkon.

Tento model byl dále ještě rozvinut v [6]. Zde byl doplněn předpoklad, že materiál ani řezací nástroj nemusí být homogenní. Tedy, že životnost řezacího nástroje není konstantní a ani řezaný materiál není vždy homogenní. Následně byla řešena stochastická úloha geometrického programování.

5.2 Design odpařovače

Dalším zajímavým příkladem je získávání pitné vody z vody mořské. Tento příklad je detailně popsán a řešen v [17]. Mořskou vodu lze teoreticky velmi výhodně odsolovat pomocí polopropustné membrány. Bohužel v praxi je tento přístup velice neefektivní. Proto se pitná voda získává za pomocí destilace. Je tedy nutné navrhnout odpařovač, pro který je pak nutné nastavit pracovní parametry. Je především nutné zvolit pracovní teplotu, náklady na amortizaci při daném výkonu a ceny materiálu na výrobu povrchů pro výměnu tepla. Dále je model rozvinut o využití odpadního tepla při kondenzaci k předehřívání vody vstupující do soustavy. Nakonec jsou přidány náklady na chemikálie bránící přílišné krystalizaci soli v kondenzátoru a náklady na pumpování vody v jednotlivých částech soustavy.

5.3 Design cívky a prostorové úlohy

Geometrické programování je také možné použít při navrhování cívek. Cívka má měděné vedení a železné jádro. Je nutné minimalizovat objem jádra a také minimalizovat obalovací měděné vedení.Dalšími parametry, které je třeba optimalizovat jsou odpor, efektivita a maximální kapacita cívky. Tyto omezení jsou také ve tvaru posynomů. Nakonec je také nutné do úlohy zakomponovat elektrické ztráty vlivem odporu. Podrobné vzorce a řešení úlohy je možné najít v [17].

Geometrické programování je obecně velmi užitečným nástrojem pro řešení úloh s prostorovým či plošným uspořádáním, neboť takové úlohy obsahují omezení v posynomiálním tvaru.

5.4 Digitální obvody

V [2] je uveden přístup k plánování rozmístění jednotlivých prvků, jejich parametrů a jejich spojení pomocí vedení. Také je pro jednotlivé prvky nutné nastavit správné napětí. Hlavním cílem v této úloze je navrhnout vhodné parametry tranzistorů, jejich propojení a uspořádání v prostoru. V úloze je nutné brát v úvahu fyzikální vlastnosti materiálů, rychlost signálu (zpoždění v závislosti na délce propojení), možnosti výroby a další parametry. Tyto úlohy jsou velmi rozsáhlé, v dnešních počítačových komponentech jsou až miliardy tranzistorů. Takto rozsáhlé úlohy jsou zatím neřešitelné. V praxi se používá vícekrokový postup, kdy se navrhnou menší funkční celky, které se opět spojují do větších. Jedná se opět o úlohu prostorového uspořádání s dodatečnými požadavky. Některé úlohy není možné modelovat pouze pomocí geometrického programování, ale vedou přibližně na úlohu zobecněného geometrického programování či na úlohu diskrétního geometrického programování. Úloha diskrétniho geometrického programování, kde jsou dodatečná omezení ve tvaru $x_i \in X$, kde X je diskrétní množina stavů.

5.5 Ekonomická aplikace - Cobb Douglasova produkční funkce

Cobb-Douglasova produkční funkce původně vznikla při snaze o modelování národního hospodářství na základě dvou vysvětlujicích proměnných: kapitálu a práce. V roce Paul Douglas a Charles Cobb přišli s funkcí Q = A$L^{\beta}K^{1-\beta}$, kde L je práce, K je kapitál a konstanta A určuje technologickou úroveň a tím efektivitu produkce. Na základě metody nejmenších čtverců určili koeficient $\beta = 0.75$. Tento odhad byl následně potvrzen Národním úřadem pro ekonomický výzkum ve Spojených státech. [18]

Jejich práce se však setkala s kritikou i přes to, že velmi přesně popisovala danou problematiku. Především kvůli tomu, že jejich odhad byl založen na malém množství dat. Průlom přišel až v roce 1947, kdy Paul Douglas aplikoval funkci na údaje ze sčítání lidu a poslal článek prezidentovi Americké ekonomické asociace. Krátce na to ovšem onemocněl a již nemohl plně pokračovat ve výzkumu. Nicméně o 20 let později na jeho práci navázali ekonomové jako Paul Samuelson a Robert Solow. [18]. Cobb-Douglasova funkce se následně stala jedním z prvních nástrojů pro modelování makroekonomikých veličin. [8]

Cobb-Douglasova produkční funkce má široké uplatnění při modelování velkého množství ekonomických a ekonometrických aplikací. Také je velmi vhodná pro modelování produkčních funkcí. Definici (2.5) je také možné přepsat do tvaru:

$$\log(\tilde{u}(\mathbf{x})) = \log(A) + \sum_{i=1}^{L} \lambda_i \log(x_i), \tag{5.1}$$

což je pro ekonometrii běžný způsob odhadování závislostí v regresních modelech. Je tedy zřejmé, že geometrické programování bude mít nepřeberné množství aplikací. Vyvstává zde ovšem zásadní problém s tím, že koeficienty λ_i nejsou známy přesně a je nutné je modelovat jako náhodné veličiny.

V této práci si nejprve ukážeme nejjednodušší tvar produkční funkce a její řešení tak, jak ji modelovali Cobb a Douglas. Následně bude naším cílem tuto funkci rozšířit pro modelování dalších úloh a ukázat jejich řešení.

Kapitola 6

Příklad

6.1 Základní úloha

Jak jsme již v úvodu této kapitoly naznačili, nejprve zde ukážeme řešení jednoduchého příkladu

$$Q = AL^{\beta}K^{1-\beta}. \tag{6.1}$$

Jak již bylo dříve napsáno, tak L je práce a K je kapitál. Q je pak množství produkce. Nyní ještě model doplníme o prodejní cenu P a náklady c_1 za jednotku práce a náklady c_2 az jednotku kapitálu. Celá úloha má pak tvar:

$$\max PQ - c_1L - c_2K, Q = AL^{\beta}K^{1-\beta}. \tag{6.2}$$

Tuto úlohu ještě nelze řešit přímo, neboť není standartního tvaru geometrického programování. Je nutné zvolit, analogicky jako v případě Markowitzova modelu, jeden ze dvou přístupů. Prvním bude, že chceme vyrobit určité množství prokuktu, který označíme jako Q^* s co nejmenšími náklady:

$$\min c_1L + c_2K, \tag{6.3}$$

$$st : AL^{\beta}K^{1-\beta} \geq Q^*. \tag{6.4}$$

Vzledem k pozitivnosti všech členů je možné úlohu přeformulovat do stadartního tvaru geometrického programování:

$$\min c_1L + c_2K, \tag{6.5}$$
$$st : Q^*A^{-1}L^{-\beta}K^{\beta-1} \leq 1. \tag{6.6}$$

Druhým možným přístupem je tak zvané rozpočtové omezení. V tomto případě máme k dispozici rozpočet B a budeme chtít vyrobit co nejmětší množství produktu:

$$\max AL^{\beta}K^{1-\beta}, \tag{6.7}$$
$$st : c_1L + c_2K \leq B. \tag{6.8}$$

Opět upravíme do standartního tvaru tím, že budeme umocňovat účelovou funkci na -1 a upravíme omezení:

$$\min A^{-1}L^{-\beta}K^{\beta-1}, \tag{6.9}$$
$$st : B^{-1}(c_1L + c_2K) \leq 1. \tag{6.10}$$

Řešení této úlohy a rozbor rešení lze nalézt například v [9].

6.2 Sestavení úlohy

V naší úloze zvolíme modelování produkce na základě 3 proměnných: L - práce, K - kapitál a M - materiál nutný k výrobě.Jejich ceny pak budou c_1, c_2 a c_3. Cena výroby tedy bude:

$$\hat{C} = c_1L + c_2K + c_3M. \tag{6.11}$$

Dále budeme v modelu předpokládat, že existují úspory z rozsahu. Tedy celkové nákady výroby se budou snižovat s rozsahem v závislosti na vyrobeném množství Q:

$$C = \hat{C}Q^{-\lambda}, \tag{6.12}$$

kde koeficient λ určuje elasticitu nákladů v závislosti na množství produkovaného množství. Pro modelování je nutné zvolit $0 < \lambda < 1$. λ blízko 0 znamená, že v případě výroby velkého množství jednotek nedojde k významné úspoře. Naopak pro λ blízko 1 dochází k významným úsporám při produkci velkého množství jednotek.

Vyrobené množství pak budeme modelovat pomocí Cobb-Douglasovy produkční funkce ve tvaru:

$$Q = AL^{a_1}K^{a_2}M^{a_3}, \tag{6.13}$$

kde A je škálovací konstanta určujicí technologické možnosti, A > 0.

Do úlohy je také nutné zahrnout omezení pro poměry jednotlivých výrobních faktorů. Je totiž nemožné vyrábět požadované množství jednotek, pokud jsou jednotlivé vstupy ve velké nerovnováze. Například každou jednotku kapitálu nemůže obsluhovat víc než o_1 pracovníků a každá jednotka kapitálu může zpracovat maximálne o_2 jednotek materiálu. Zavedeme následujicí omezení:

$$\tfrac{L}{K} \leq o_1, \tag{6.14}$$
$$\tfrac{K}{M} \leq o_2. \tag{6.15}$$

Pro potřeby geometrického programování je pak třeba tyto omezení přetransformovat do standartního tvaru. Obě omezení o_i jsou kladná je tedy možné je přepsat na tvar:

$$o_1^{-1}\tfrac{L}{K} \leq 1, \tag{6.16}$$
$$o_2^{-1}\tfrac{K}{M} \leq 1. \tag{6.17}$$

Nyní již můžeme sestavit základní úlohu:

$$\max \mathrm{ZISK} = DP - (c_1L + c_2K + c_3M)Q^{-\lambda}, \tag{6.18}$$
$$s.t: \tfrac{Q}{A}L^{-a_1}K^{-a_2}M^{-a_3} \leq 1, \tag{6.19}$$
$$\tfrac{Q}{D} \leq 1, \tag{6.20}$$

$$o_1^{-1}\frac{L}{K} \leq 1, \tag{6.21}$$

$$o_2^{-1}\frac{K}{M} \leq 1, \tag{6.22}$$

kde D je poptávka a P je prodejní cena. Ještě zavedeme omezení, že nemůžeme prodat více než vyrobíme. Poptávka D samozřejmně není deterministická veličina, ale je náhodná. Proto se v dalším textu budeme zabývat způsoby řešení této stotchastické úlohy geometrického programování. Ukážeme si celkem 3 postupy pro vyřešení této úlohy a v závěru shrneme výsledky a vhodnost jednotlivých metod pro řešení. Pro numerické výpočty volíme následující hodnoty jednotlivých parametrů:

- A = 88
- $\lambda = 0.02$
- $c_1 = 0.15$, $c_2 = 0.02$ a $c_3 = 0.07$
- $a_1 = 0.32$ $a_2 = 0.16$ a $a_3 = 0.18$
- $o_1 = 3$ a $o_2 = 2$

Pro výpočty byl použit program GAMS. Vzhledem k velikosti úlohy bylo možné úlohu řešit pomocí demoverze za využití solveru CONOPT3. V demoverzi nebylo možné využít speciální solver pro řešení úlohy geometrického programování. Proto jsme pro výpočty převedli úlohu do konvexního tvaru (3.2) a využili solver pro řešení obecné úlohy nelineárního programování. Více informací o programu GAMS je možné najít například v [19] či [20] či na stránkách www.gams.com. Informace o fungování solveru CONOPT3 lze najít v jeho manuálu [21].

6.3 Řešení pomocí kvantilu

V případě, že nám jde hlavně o splnění požadavků na výrobu, tak je vhodné upravit stávající podmínku na vyrobené množství na pravděpodobnostní podmínku:

$$P(g_k(\mathbf{x}) \leq 1) \geq 1 - \varepsilon_k. \tag{6.23}$$

Tím máme zajištěno, že s 100(1-ε_k)% pravděpodobností dojde k dodání potřebného množství produktu zákazníkovi. Tímto způsobem je také možné upravit všechny podmínky v případě, že kterékoli s koeficientů c_i nejsou známy přesně. V našem případě tedy upravujeme podmínku (6.20) do tvaru:

$$P(D \leq Q) \geq 1 - \varepsilon. \tag{6.24}$$

V tomto případě je nutné znát či odhadnout rozdělení poptávky. V našem příkladě budeme předpokládat, že poptávka D má normální rozdělení se střední hodnotou 1500 a rozptylem 900. Pak lze danou podmínku upravit následovně:

$$P(\frac{D - 1500}{30} \leq \frac{Q - 1500}{30}) \geq 1 - \varepsilon \tag{6.25}$$

$$\Phi(\frac{Q - 1500}{30}) \geq 1 - \varepsilon \tag{6.26}$$

$$\frac{Q - 1500}{30} \geq \Phi^{-1}_{(1-\varepsilon)} \tag{6.27}$$

Nakonec ještě úprava pro potřeby úlohy geometrického programování:

$$(1500 + 30\Phi^{-1}_{(1-\varepsilon)})Q^{-1} \leq 1. \tag{6.28}$$

Funkce Φ^{-1} je kvantil normálního rozdělení N(0, 1). Standartními volbami pro ε jsou hodnoty 1%, 5% či 10%. Hodnoty kvantilů normálního rozdělení N(0,1) jsou pak po řadě 2.33, 1.64 a 1.28. Za těchto podmínek je také možné upravit účelovou funkci. Protože je potřebné množství výroby modelováno v podmínce (6.41), naším úkolem je pouze minimalizovaní nákladů na výrobu. Máme připravené všechny potřebné základy pro formulování konečné úlohy:

$$\min \mathrm{NAKLADY} = (c_1L + c_2K + c_3M)Q^{-\lambda}, \tag{6.29}$$

$$s.t: \tfrac{Q}{A}L^{-a_1}K^{-a_2}M^{-a_3} \leq 1, \tag{6.30}$$

$$(1500 + 30\Phi^{-1}_{(1-\varepsilon)})Q^{-1} \leq 1, \tag{6.31}$$

$$o_1^{-1}\tfrac{L}{K} \leq 1, \tag{6.32}$$

$$o_2^{-1}\tfrac{K}{M} \leq 1, \tag{6.33}$$

následně již pouze nahradíme proměnné L, K, M a Q přes které budeme optimalizovat proměnnými x_1 až x_4 a získáme úlohu geometrického programování ve standartním tvaru:

$$\min \text{NAKLADY} = (c_1x_1 + c_2x_2 + c_3x_3)x_4^{-\lambda}, \tag{6.34}$$

$$s.t : A^{-1}x_1^{-a_1}x_2^{-a_2}x_3^{-a_3}x_4 \leq 1, \tag{6.35}$$

$$(1500 + 30\Phi^{-1}_{(1-\varepsilon)})x_4^{-1} \leq 1, \tag{6.36}$$

$$o_1^{-1}x_1x_2^{-1} \leq 1, \tag{6.37}$$

$$o_2^{-1}x_2x_3^{-1} \leq 1. \tag{6.38}$$

Úlohu dále převedeme do konvexního tvaru substitucí $z_i = \log(x_i)$ a pro další zjednodušení zlogaritmujeme monomiální omezení:

$$\min \text{NAKLADY} = \exp(\log(c_1) + z_1 - \lambda z_4) \tag{6.39}$$

$$+ \exp(\log(c_2) + z_2 - \lambda z_4) + \exp(\log(c_3) + z_3 - \lambda z_4),$$

$$s.t : -\log(A) - a_1z_1 - a_2z_2 - a_3x_3 + z_4 \leq 0, \tag{6.40}$$

$$\log(1500 + 30\Phi^{-1}_{(1-\varepsilon)}) - z_4 \leq 0, \tag{6.41}$$

$$-\log(o_1^{-1}) + z_1 - z_2 \leq 0, \tag{6.42}$$

$$-\log(o_2^{-1}) + z_2 - z_3 \leq 0. \tag{6.43}$$

Tuto konvexní úlohu pak vyřešíme pomocí obecného nelineárního solveru CONOPT3 v GAMSu a získáme následujicí výsledky pro jednotlivé hodnoty ε:

V této úloze jsme chtěli s 100(1-ε) procentní pravděpodobností uspokojit poptávku po našem produktu při znalosti jejího rozdělení. Řešení této úlohy nám pak řiká, jak optimálně rozvrhnout zdroje a dává nám informaci o nákladech na výrobu poptávaného množství. Nicméně již nic neříká o celkovém zisku. Ten je nutné vypočítat z ceny produktu a porovnat s náklady na výrobu. Poté je možné rozhodnout, zda se výroba vyplatí či nikoli. Z řešení v příloze A je také patrné, že jedno z omezení není splněno jako rovnost a bylo by náročné řešit duální úlohu, neboť by zde vycházel jeden nulový Lagrangeův koeficient a úloha by se tím zkomplikovala. Nelineární solver neměl s úlohou

Proměnná	$\varepsilon = 1\%$	$\varepsilon = 5\%$	$\varepsilon = 10\%$
x_1 (práce)	54.982	53.893	53.303
x_2 (kapitál)	159.334	156.179	154.470
x_3 (materiál)	79.599	78.101	77.246
x_4 (vyrobené množství)	1570.266	1549.984	1539.172
Náklady	14.677	14.392	14.241

Tabulka 6.1: Výsledky kvantilového přístupu.

žádné potíže a vyřešil ji v řádu tisícin sekundy. Z toho vyplývá, že pro tuto velikost úlohy není nutné hledat specializovaný software pro její řešení.

6.4 Řešení pomocí scénářů

Dalším možným přístupem je generování scénářů. Mějme tedy konečnou množinu scénářů:

$$\omega = \{\omega_s, s = 1, 2, \ldots, S\}, \tag{6.44}$$

jejichž realizace pak budou

$$\mathbf{D} = \{D_s, s = 1, 2, \ldots, S\}, \tag{6.45}$$

které nastávají s pravděpodobnostmi p_s, přičemž musí být dodrženo, že

$$\sum_{s=1}^{S} p_s = 1. \tag{6.46}$$

V našem příkladu vytvoříme 3 scénáře:

- optimistický: D_1=1200, s pravděpodobností $p_1 = 0.3$
- neutrální: D_2=1500, s pravděpodobností $p_2 = 0.5$
- pesimistický: D_3=1900, s pravděpodobností $p_3 = 0.2$

Scénářu může být i více, ale je nutné si uvědomit, že s každým dalším scénářem se úloha velmi rychle rozrůstá a je tedy potřeba zvážit, kolik scénářů je opravdu nutné počítat a kolik máme k dispozici výpočetních prostředků. Také je možné scénáře formulovat přes více období, čímž se úloha dále rozrůstá. Cílem scénářového přístupu je zvážit všechny možnosti a rozhodnout se pro optimální řešení vzhledem ke všem scénářům a pravděpodobnostem, se kterými nastanou. Ukážeme také, že v případě apriorní znalosti budoucího scénáře bude optimální řešení jiné. Reformulujeme tedy naši úlohu:

$$\max \text{ZISK} = \textstyle\sum_{s\in S} p_s[\min(D_s,Q)P - (c_1L + c_2K + c_3M)Q^{-\lambda}] \quad (6.47)$$

$$= \textstyle\sum_{s\in S}[p_s \min(D_s,Q)P] - (c_1L + c_2K + c_3M)Q^{-\lambda}$$

$$s.t: \tfrac{Q}{A}L^{-a_1}K^{-a_2}M^{-a_3} \le 1, \quad (6.48)$$

$$o_1^{-1}\tfrac{L}{K} \le 1, \quad (6.49)$$

$$o_2^{-1}\tfrac{K}{M} \le 1. \quad (6.50)$$

Pro značení jednotlivých scénářů jsme zvolili dolní index, aby nemohlo dojít k záměně s exponenty. V této úloze je navíc nutné počítat s cenou P=0.07. Vidíme tedy, že naše 3 scénáře značně zkomplikovali úlohu. Úloha není ve standartním tvaru a navíc obsahuje funkci "minimum". Je podstatné zde zmínit, že jednotlivé scénáře nelze řešit odděleně, neboť pak neexistuje jednoduchý způsob pro jejich zkombinování do jednoho výsledku. To si také v této práci ukážeme. Nejprve vyřešíme každý scénář zvlášť a pak výsledky porovnáme s řešením celé scénářové úlohy. V případě řešení jednotlivých scénářů využijeme přístup z minulého příkladu. Pro scénářovou úlohu bude nutné vymyslet sofistikovanější přístup. Je proto nutné zvolit jiný přístup pro přeformulování úlohy na standartní tvar geometrického programování.

Bylo by velmi složité a nepřehledné řešit celou scénářovou úlohu v jedné optimalizační úloze. Navíc by se nám nikdy nepodařilo převést takto formulovanou úlohu na standartní tvar geometrického programování. Pro její řešení by dále byl nutný algoritmus, který dokáže větvit výpočet, což by bylo nutné kvůli přítomnosti funkce "minimum"a také by musel umět řešit úlohu zobecněného geometrického programování. Proto větvení provedeme před výpočtem a jednotlivé případy převedeme na standartní tvar.

Důležité případy jsou, pokud je optimální množství v intervalu [1200, 1500] nebo [1500, 1900]. V případě, že by nebyla produkce rentabilní, tak by bylo vhodné vypočítat ještě úlohu na intervalu [0, 1200]. Produkce na intevalu [1900, ∞) by byla zajimavá, pokud by náklady na výrobu mohli klesat s rozsahem produkce. To by mohla nastat v případě, že by $a_1 + a_2 + a3$ bylo větší než 1, což není náš případ. Vzhledem k tomu, že v žádném scénáři není možné prodat více než 1900 kusů, tak tato možnost není zajímavá a hodnoty 1900 by bylo dosaženo při optimalizaci na intervalu [1500, 1900]. Pro úplnost budou v tabulce zobrazena i řešení modelu na intervalech [0, 1200] a [1900, ∞).

Základní úloha má pak tvar:

$$\max \text{ZISK} = p_1 \min(D_1, Q)P + p_2 \min(D_2, Q)P \tag{6.51}$$
$$+p_3 \min(D_3, Q)P - (c_1 L + c_2 K + c_3 M)Q^{-\lambda}$$
$$s.t: \tfrac{Q}{A} L^{-a_1} K^{-a_2} M^{-a_3} \leq 1, \tag{6.52}$$
$$o_1^{-1} \tfrac{L}{K} \leq 1, \tag{6.53}$$
$$o_2^{-1} \tfrac{K}{M} \leq 1. \tag{6.54}$$

Abychom se zbavili funkcí "minimum", tak nyní rozdělíme úlohu na jednotlivé intervaly. Začneme intervalem [1200, 1500]. Úloha se pak zjednoduší na tvar:

$$\max \text{ZISK} = p_1 D_1 P + (p_2 + p_3)QP \tag{6.55}$$
$$-(c_1 L + c_2 K + c_3 M)Q^{-\lambda}$$
$$s.t: \tfrac{Q}{A} L^{-a_1} K^{-a_2} M^{-a_3} \leq 1, \tag{6.56}$$
$$o_1^{-1} \tfrac{L}{K} \leq 1, \tag{6.57}$$
$$o_2^{-1} \tfrac{K}{M} \leq 1. \tag{6.58}$$

První člen v účelové funkci je konstatní, tedy není v optimalizační úloze důležitý. Pro zbytek pak zavedeme novou proměnnou x_0 a provedeme následující úpravy:

$$s.t: x_0 \leq (p_2 + p_3)QP - (c_1 L + c_2 K + c_3 M)Q^{-\lambda} \tag{6.59}$$
$$\Rightarrow x_0 + (c_1 L + c_2 K + c_3 M)Q^{-\lambda} \leq (p_2 + p_3)QP \tag{6.60}$$
$$\Rightarrow x_0 Q^{-1} P^{-1} (p_2 + p_3)^{-1} \tag{6.61}$$
$$+(c_1 L + c_2 K + c_3 M)Q^{-\lambda-1} P^{-1} (p_2 + p_3)^{-1} \leq 1$$

Tímto jsme získlali omezení ve tvaru posynomu a můžeme definovat konečnou úlohu pro tento interval:

$$\max x_0 \tag{6.62}$$
$$s.t: \tfrac{Q}{A}L^{-a_1}K^{-a_2}M^{-a_3} \leq 1, \tag{6.63}$$
$$o_1^{-1}\tfrac{L}{K} \leq 1, \tag{6.64}$$
$$o_2^{-1}\tfrac{K}{M} \leq 1, \tag{6.65}$$
$$x_0Q^{-1}P^{-1}(p_2+p_3)^{-1} \tag{6.66}$$
$$+(c_1L+c_2K+c_3M)x_4^{-\lambda-1}P^{-1}(p_2+p_3)^{-1} \leq 1,$$

dále nahradíme proměnné L, K, M a Q proměnnými x_1 až x_4 a provedeme konvexní transformaci $z_i = \log(x_i)$, i = 0, 1 , . . . , 4. Úloha geometrického programování v konvexním tvaru je pak:

$$\max z_0 \tag{6.67}$$
$$s.t: z_4 - \log(A) - a_1z_1 - a_2z_2 - a_3z_3 \leq 0, \tag{6.68}$$
$$-\log(o_1) + z_1 - z_2 \leq 0, \tag{6.69}$$
$$-\log(o_2) + z_2 - z_3 \leq 0, \tag{6.70}$$
$$\exp[z_0 - z_4 - \log(P) - \log(p_2+p_3)] \tag{6.71}$$
$$+\exp[\log(c_1)z_1 - (\lambda+1)z_4 - \log(P) - \log(p_2+p_3)]$$
$$+\exp[\log(c_2)z_2 - (\lambda+1)z_4 - \log(P) - \log(p_2+p_3)]$$
$$+\exp[\log(c_3)z_3 - (\lambda+1)z_4 - \log(P) - \log(p_2+p_3)] \leq 1.$$

Tuto úlohu je již opět možné vyřešit v programu GAMS. Výsledky budou uvedeny na konci tohoto příkladu. Nyní se budeme podobným způsobem věnovat intervalu [1500, 1900], kde tvar úlohy bude následující:

$$\max \text{ZISK} = p_1D_1P + p_2D_2P + p_3QP \tag{6.72}$$
$$-(c_1L+c_2K+c_3M)Q^{-\lambda}$$
$$s.t: \tfrac{Q}{A}L^{-a_1}K^{-a_2}M^{-a_3} \leq 1, \tag{6.73}$$
$$o_1^{-1}\tfrac{L}{K} \leq 1, \tag{6.74}$$
$$o_2^{-1}\tfrac{K}{M} \leq 1. \tag{6.75}$$

Obdobně jako na předchozím intervalu jsou první 2 členy účelové funkce (6.72) konstantní a je tedy možné je z úlohy odstranit. Opět zavedeme pomocnou proměnnou x_0 a upravíme účelovou funkci do podmínky:

$$s.t: x_0 \leq p_3QP - (c_1L + c_2K + c_3M)Q^{-\lambda} \tag{6.76}$$

$$\Rightarrow x_0 + (c_1L + c_2K + c_3M)Q^{-\lambda} \leq p_3QP \tag{6.77}$$

$$\Rightarrow x_0Q^{-1}P^{-1}p_3^{-1} \tag{6.78}$$

$$+(c_1L + c_2K + c_3M)Q^{-\lambda-1}P^{-1}p_3^{-1} \leq 1$$

Celá úloha pro zadání no prostředí GAMS, včetně nahrazení proměnných a zavedení substituce, pak bude:

$$\max z_0 \tag{6.79}$$

$$s.t: z_4 - \log(A) - a_1z_1 - a_2z_2 - a_3z_3 \leq 0, \tag{6.80}$$

$$-\log(o_1) + z_1 - z_2 \leq 0, \tag{6.81}$$

$$-\log(o_2) + z_2 - z_3 \leq 0, \tag{6.82}$$

$$\exp[z_0 - z_4 - log(P) - log(p_3)] \tag{6.83}$$

$$+\exp[\log(c_1)z_1 - (\lambda + 1)z_4 - \log(P) - \log(p_3)]$$

$$+\exp[\log(c_2)z_2 - (\lambda + 1)z_4 - \log(P) - \log(p_3)]$$

$$+\exp[\log(c_3)z_3 - (\lambda + 1)z_4 - \log(P) - \log(p_3)] \leq 1.$$

V solveru CONOPT3 v programu GAMS nejprve vypočítáme jednotlivé scénáře. Použijeme úlohu (6.39) - (6.43) z předchozího přístupu a omezení (6.41) nahradíme omezením $\log(D_s) - z_4 \leq 0$. Výsledky jsou uvedeny v tabulce 6.2.

Proměnná	s = 1	s = 2	s = 3
x_1 (práce)	36.60	51.32	73.41
x_2 (kapitál)	106.06	148.71	212.72
x_3 (materiál)	52.98	74.37	106.38
x_4 (vyrobené množství)	1200	1500	1900
náklady	9.82	13.71	19.53
zisk	74.18	91.29	113.47

Tabulka 6.2: Výsledky jednotlivých scénářů.

Následně vypočítáme scénářový model pro jednotlivé intervaly výroby:

Proměnná	Q≤1200	1200≤Q≤1500	1500≤Q≤1900	1900≤Q
x_1 (práce)	36.60	51.32	55.20	73.41
x_2 (kapitál)	106.06	148.71	159.97	212.72
x_3 (materiál)	52.98	74.37	79.99	106.38
x_4 (vyrobené množství)	1200	1500	1574.53	1900
náklady	9.82	13.71	14.74	19.53
předpokládaný zisk	74.18	84.99	85.00	84.774

Tabulka 6.3: Výsledky scénářového přístupu.

Z výsledků je patrné, jak se liší přístup k jednotlivým scénářům a k celé úloze. V případě jednotlivých scénářů se vzhledem k prodejní ceně vždy vyplatí vyrobit maximální množství, které lze prodat. Na druhou stranu v celém scénářovém modelu je nutné počítat s tím, že se nám nemusí vyplatit celých 1900 jednotek, neboť pak zbytečně investujeme do výroby jednotek, které se v závistlosti na nastalém scénáři nemusí prodat. Navíc by bylo nutné přebytečné jednotky uložit do skladiště a docházelo by k dalšímu zvyšování nákladů, které nejsou v naší úloze započteny. Model doporučuje při stávajicí ceně vyrobit 1574.53 jednotek za využití zdrojů uvedených v tabulce. Toto množství by se významně měnilo při změně ceny, což si může každý vyzkoušet pomocí zdrojových kódů pro GAMS, které jsou k této práci přiloženy. Podrobné výsledky, pouze pro optimální řešení, je uvedeno v příloze B.

6.5 Řešení pomocí modelování poptávky

Posledním řešením, které si zde představíme je založeno na myšlence modelování poptávky D na základě jiných proměnných v modelu [14]. Hlavní proměnnou, na které budou záviset prodeje bude cena P, která již nebude pevná, ale bude to jedna z rozhodovacích proměnných. Pro další možnosti ovlivnění poptávky pak přidáme ještě náklady na marketing, kterými bude možné vyvážit zvýšenou cenu. Nakonec ještě přidáme nákladový člen, který bude zohledňovat rostoucí požadavky na kvalitu a marketing při neúměrném růstu ceny. Zadefinujeme tedy funkci pro poptávku jako:

$$D = \alpha P^{-\gamma} N^{\theta}, \tag{6.84}$$

$$R = gP^2, \tag{6.85}$$

kde P je cena, α je konstanta pro rozsah trhu, $\gamma > 1$ je elasticita poptávky, N jsou náklady na reklamu a $0 < \theta < 1$ je efekt reklamy. R je pak člen usměrňujicí náklady při zvyšování ceny, g je škálovací konstanta a α škáluje velikost poptávky.

Z výše uvedeného vyplývá, že nám přibyde jedno omezení a také se podstatným způsobem změní účelová funkce. Získáme úlohu:

$$\max \text{ZISK} = DP - (c_1 L + c_2 K + c_3 M)Q^{-\lambda} - ND - gP^2, \tag{6.86}$$

$$s.t: \tfrac{Q}{A} L^{-a_1} K^{-a_2} M^{-a_3} \leq 1, \tag{6.87}$$

$$\tfrac{Q}{D} \leq 1, \tag{6.88}$$

$$o_1^{-1} \tfrac{L}{K} \leq 1, \tag{6.89}$$

$$o_2^{-1}\tfrac{K}{M} \leq 1, \tag{6.90}$$
$$D\alpha^{-1}P^{\gamma}N^{-\theta} \leq 1. \tag{6.91}$$

Toto není standartní úloha geometrického programování, hlavně proto, že funkce (6.86) stále obsahuje znaménko mínus. Zavedeme novou proměnnou x_0 a provedeme tedy následující úpravy:

$$\max x_0 = \min x_0^{-1} \tag{6.92}$$

$$s.t: x_0 \leq QP - (c_1L + c_2K + c_3M)Q^{-\lambda} - NQ - gP^2 \tag{6.93}$$
$$\Rightarrow x_0 + (c_1L + c_2K + c_3M)Q^{-\lambda} + NQ + gP^2 \leq QP \tag{6.94}$$
$$\Rightarrow x_0Q^{-1}P^{-1} + (c_1L + c_2K + c_3M)Q^{-\lambda-1}P^{-1} \tag{6.95}$$
$$+NP^{-1} + gQ^{-1}P \leq 1$$

následně již pouze nahradíme proměnné L, K, M, Q, N, P a D, přes které budeme optimalizovat po řadě proměnnými x_1 až x_7 a získáme úlohu geometrického programování ve standartním tvaru:

$$\min x_0^{-1}, \tag{6.96}$$
$$s.t: A^{-1}x_1^{-a_1}x_2^{-a_2}x_3^{-a_3}x_4 \leq 1, \tag{6.97}$$
$$x_4x_7^{-1} \leq 1, \tag{6.98}$$
$$o_1^{-1}x_1x_2^{-1} \leq 1, \tag{6.99}$$
$$o_2^{-1}x_2x_3^{-1} \leq 1, \tag{6.100}$$
$$x_0x_4^{-1}x_6^{-1} + (c_1x_1 + c_2x_2 + c_3x_3)x_4^{-\lambda-1}x_6^{-1} \tag{6.101}$$
$$+x_5x_6^{-1} + gx_4^{-1}x_6 \leq 1,$$
$$\alpha^{-1}x_5^{-\theta}x_6^{\gamma}x_7 \leq 1. \tag{6.102}$$

Pro výpočty pak použijeme následující konvexní tvar se substitucí $z_i = \log(x_i)$ pro i = 0, 2, ..., 7, substituci lze provést vzhledem ke kladnosti všech proměnných:

$$\max z_0, \tag{6.103}$$
$$s.t: -\log(A) - a_1z_1 - a_2z_2z_3 + z_4 \leq 0, \tag{6.104}$$

$$z_4 - z_7 \leq 0, \tag{6.105}$$

$$-\log(o_1) + z_1 - z_2 \leq 0, \tag{6.106}$$

$$-\log(o_2) + z_2 - z_3 \leq 0, \tag{6.107}$$

$$\exp(z_0 - z_4 - z_6) + exp[\log(c_1) + z_1 - (\lambda + 1)z_4 - z_6] \tag{6.108}$$

$$+\exp[\log(c_2) + z_2 - (\lambda + 1)z_4 - z_6] \tag{6.109}$$

$$+\exp[\log(c_3) + z_3 - (\lambda + 1)z_4 - z_6] \tag{6.110}$$

$$+\exp(z_5 - z_6) + \exp[\log(g) - z_4 + z_6] \leq 1, \tag{6.111}$$

$$-\log(\alpha) - \theta z_5 + \gamma z_6 + z_7 \leq 0. \tag{6.112}$$

Pro výpočet pak volíme hodnoty dodatečných konstant následovně:

- $\alpha = 1\ 500$
- $\gamma = 1.8$
- $\theta = 0.12$
- g $= 10^{-7}$

Tuto úlohu pak vyřešíme nelineárním solverem CONOPT3 v programu GAMS:

x_1 (práce)	x_2 (kapitál)	x_3 (materiál)	x_4 (vyrobené množství)
3948.14	11441.47	5721.58	26370.47
x_5 (marketing)	x_6 (cena)	x_7 (poptávka)	x_0 (zisk)
0.00997	0.15	26370.47	2683.83

Tabulka 6.4: Výsledky úlohy při modelování poptávky.

V této úloze jsme modelovali poptávku za pomocí ceny a nákladů na marketing. Řešení nám tedy dává přesný návod pro nastavení ceny našeho produktu, optimální náklady na reklamu a také je hned patrné, jaký je předpokládaný zisk v případě platnosti tohoto modelu. Z výsledků je tedy na první pohled zřejmé, že se výroba vyplatí. Vzhledem k vyrobenému množství se také uplatnily úspory z rozsahu, které násobí cenu výroby koeficientem $Q^{-\lambda} = 0.816$. Také je vidět poměrně malý vliv reklamy na prodané množství. Jedno z omezení opět není splněno jako rovnost a tedy nelze doporučit řešení pomocí duální úlohy. Čas řešení je téměř nulový a tedy není nutné použít specializovaný solver. Podrobný výpis řešení je v příloze C.

6.6 Možná rozšíření

Tyto modely lze dále rozšiřovat. Na tomto místě si uvedeme stručný návrh některých z nich.

Prvním možným rozšířením by bylo přidání dalších zdrojů do produkční funkce. Úloha by se tím v podstatě nezměnila, jen by bylo nutné vypočtávat více proměnných.

Dalším rozšířením by bylo připuštění nedeterministických koeficientů a_i. Tím by se velmi zkomplikovala podmínka:

$$A^{-1}x_1^{-a_1}x_2^{-a_2}x_3^{-a_3}x_4 \leq 1, \tag{6.113}$$

na maximální vyrobené množství za použití danného množství surovin. Více informací o řešení této úlohy lze nalézt například v [10].

Dále je možné podstatně rozšířit úlohu se scénářovým přístupem. Prvním rozšířením by mohlo být zavedení skladu. Tedy penalizace za neprodání všech vyrobených jednotek. V této úloze by se pak objevil další nepříjemný člen nákladů na skladování:

$$-\min(0, Q - D_s)J, \tag{6.114}$$

kde J jsou náklady na skladování. Přidáním tohoto členu by model dával konzervativnější (nižší) doporučený počet vyrobených jednotek. Druhým přirozeným rozšířením by byla optimalizace přes více období. Nicméně obtížnost úlohy by velmi rychle rostla o nutné větvení v důsledku přítomnosti funkcí "minimum".

Do úlohy s modelováním poptávky by bylo možné přidat další členy zohledňujicí konkurenci, zisk či ztrátu tržního podílu či věrnost zákazníků ke značce. Více o tomto přístupu je možné se dočíst například v [14].

Nakonec je možné úlohu rozšířit o více produktů. Tyto produkty lze dále provázat, například tak, že jeden produkt bude potřeba k výrobě jiného, budou různé vstupy pro různé produkty a různé skladovací náklady na jejich skladování. V tomto směru umožňuje geometrické programování počítat optimální využití skladovacího prostoru a náklady budou účtovány za 1 metr čtvereční.

Kapitola 7

Závěr

V této práci jsme čtenáře seznámili s úlohou geometrického programování. Udělali jsme přehled možných způsobů řešení této úlohy a porovnali jsme jejich výhody a nevýhody včetně představení algoritmů pro jejich řešení. Také jsme udělali přehled možných rozšíření této úlohy a odkázali jsme zvídavé čtenáře na příslušnou literaturu.

Geometrické programování má mnoho technických aplikací. Jejich přehled je také součástí této práce. Naším hlavním cílem bylo zkoumání ekonomických aplikací. Pro tento účel jsme vybrali Cobb-Douglasovy produkční funkce. Představili jsme původní koncept autorů a také jsme dali tyto funkce do souvislosti s regresními modely.

Hlavní částí práce je produkční model s náhodnou poptávkou. Pro náš model jsme použili 3 vstupy: práci, kapitál a materiál. Výrobní funkce je v Cobb-Douglasově tvaru. Hlavním přínosem této práce je prezentování způsobů, jak převédst úlohu na deterministický tvar a úprav pro převod těchto úloh do standartního tvaru geometrického programování. Nakonec jsme úlohu transformovali do konvexního tvaru a použili obecný nelineární solver CONOPT3 v programu GAMS. Součástí příloh této práce jsou příslušné zdrojové kódy a také jejich kompletní řešení.

První přístup (kapitola 6.3) k řešení pomocí kvantilů je velmi jednoduchý. Chceme vyrobit dostatečné množství produktu pro uspokojení poptávky se 100(1-ε) procentní pravděpodobností. Výstupem modelu je optimální množství jednotlivých vstupů a náklady na výrobu, které jsme v této úloze minimalizovali. Také jsme analýzou výsledků vyloučili možnost použití duální úlohy k řešení tohoto problému.

V druhém přístupu (kapitola 6.4) jsme využili scénářový přístup. Vytvořili jsme 3 možné scénáře a ukázali jsme, jak se řešení jednotlivých scénářů liší od řešení celé úlohy při uvažování všech scénářů najednou a pravděpodobností, se kterou nastanou. Výsledek dává doporučení pro optimální objem výroby a také optimální využití jednotlivých vstupů.

V posledním přístupu (kapitola 6.5) jsme se snažili modelovat poptávku po produktu na základě nastavení optimální ceny a přidali jsme také možnost ovlivnění poptávky pomocí reklamy. Tento model dává doporučení pro tvorbu ceny, vyrobené množství a také využití jednotlivých zdrojů.

Všechny modely bylo možné převést na standartní tvar geometrického programování a velmi rychle vyřešit pomocí demoverze programu GAMS, pro rozsáhlejší úlohy by bylo nutné tento program zakoupit. Produkční funkce v Cobb-Douglasově tvaru jsou častým výstupem z regresních modelů. Z toho vyplývá, že geometrické programování je pro ekonomické aplikace velmi zajímavým nástrojem.

Literatura

[1] Stephen P. Boyd et al.: *A tutorial on geometric programming. Optimization and Engineering*, 2007, pp. 67-127

[2] Stephen P. Boyd, Seung-Jean Kim, Dinesh D. Patil, Mark A. Horowitz: *Digital Circuit Optimization via Geometric Programming*, Operations Research,Vol. 53, No. 6, November–December 2005, pp. 899–932 issn 0030-364X

[3] R.S. Dembo, *Dual to primal conversion in geometric programming*, Journal of Optimization Theory and Applications, vol. 26, pp. 243-252, 1978b.

[4] S.S.K. Deepak: *A Geometric Programming Model for Production Rate Optimiation of Turning Process with Experimental Validation* International Journal of Engineering Research and Applications, Vol. 2, Issue 5, Septemeter-October 2012, pp. 1544-1549

[5] R.J. Duffin, E.L. Peterson, C. Zener, *Geometric Programming - Theory and Applications*, John Wiley: New York, 1967.

[6] J. Dupačová: *Stochastic geometric programming with an application*, Kybernetika, volume 46 (2010), number 3, pp. 374-386.

[7] J. G.Ecker, *Geometric Programming: Methods, Computations and Applications*. SIAM Review, Vol. 22, No. 3 (Jul., 1980), pp. 338-362, Published by: Society for Industrial and Applied Mathematics, Article Stable URL: http://www.jstor.org/stable/2030321

[8] Jesus Filipe and Gerard Adams (2005). *The Estimation of the Cobb Douglas Function*. Eastern Economic Journal 31 (3): 427–445.

[9] I. Guney, E. OZ: , *An application of geometric programming*, International Journal of Electronics; Mechanical and Mechatronics Engineering, Vol.2 Num. 2, pp. 157-161

[10] R. Jagannathan: , *A stochastic geometric programming problem with multiplicative recourse*, Oper. Res. Lett. 9 (1990), pp. 99-104

[11] J. Kyparisis: *Sensitivity analysis in geometric programming: Theory and Computations.* Ann. Oper. Res. 27, 1990, pp. 39–64

[12] M. Luptáčik: *Geommetric programming. Method and applications.* OR Spektrum 2, 1981, pp. 129-143

[13] Costas D. Maranas. Christodoulos A. Floudas, *Global optimization in generalized geometric programming, Computers & Chemical Engineering*, Volume 21, Issue 4, 20 December 1997, pp. 351-369, ISSN 0098-1354, 10.1016/S0098-1354(96)00282-7 (http://www.sciencedirect.com/science/article/pii/S0098135496002827)

[14] F.G. Nezami, M.B. Aryanezhad a S.J.Sadjadi, *Determining Optimal Demand Rate and Production Decisions: A Geometric Programming Approach*

[15] J. Rajgopal, D.L Bricker, *Posynomial geometric programming as a special case of semi-infinite linear programming*, Journal of Optimization Theory and Applications, vol. 66, pp. 455-475, 1990.

[16] J. Rajgopal, D.L. Bricker, *Solving Posynomial Geometric Programming Problems via Generalized Linear Programming*

[17] C. Zenner, *Engineering design by geometric programming*, Jon Wiley & Sons Inc., 1971, ISBN 0-471-98200-8, pp. 87-82

[18] *The Cobb-Douglas Production Function Once Again: Its History, Its Testing, and Some New Empirical Values.* Journal of Political Economy 84 (5): 903–916. October 1976.

[19] GAMS Development Corporation: GAMS - The Solver Manual, 2008: *http://www.gams.com/dd/docs/solvers/alphaecp.pdf*

[20] GAMS Development Corporation: GAMS - A User's Guide, 2008: *http://www.gams.com/dd/docs/bigdocs/GAMSUsersGuide.pdf*

[21] Manuál solveru CONOPT: *http://www.gams.com/dd/docs/solvers/conopt.pdf*

Seznam tabulek

Přílohy

Řešení úlohy z kapitoly 6.3

```
C O N O P T 3   version 3.15D
Copyright (C)   ARKI Consulting and Development A/S
                Bagsvaerdvej 246 A
                DK-2880 Bagsvaerd, Denmark


The model has 5 variables and 5 constraints
with 14 Jacobian elements, 4 of which are nonlinear.
The Hessian of the Lagrangian has 4 elements on the diagonal,
3 elements below the diagonal, and 4 nonlinear variables.

** Optimal solution. Reduced gradient less than tolerance.


CONOPT time Total                            0.001 seconds
  of which: Function evaluations             0.000 =  0.0%
            1st Derivative evaluations       0.000 =  0.0%
            Directional 2nd Derivative       0.000 =  0.0%


                      LOWER     LEVEL     UPPER    MARGINAL

---- EQU nakladyeq      .         .         .        1.000
---- EQU podminka1    -INF      4.477     4.477    -21.578
---- EQU podminka2    -INF     -7.339    -7.339    -21.293
```

```
---- EQU podminka3      -INF     -1.064     1.099      EPS
---- EQU podminka4      -INF      0.693     0.693    -0.785

  nakladyeq  Rovnice nákladů
  podminka1  Výrobní funkce
  podminka2  Požadavek na výrobu
  podminka3  Omezení na max. počet pracovníků na jednotku kapitálu
  podminka4  Omezení na zpracovaný materiál jednotkou kapitálu

                       LOWER     LEVEL     UPPER    MARGINAL

---- VAR naklady        -INF     14.241     +INF       .

  naklady  Nákladová funkce

---- VAR z  Promenne

     LOWER     LEVEL     UPPER    MARGINAL

L     -INF      3.976     +INF       .
K     -INF      5.040     +INF       .
M     -INF      4.347     +INF       .
Q     -INF      7.339     +INF       .


**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
```

```
                           0  UNBOUNDED
                           0     ERRORS


EXECUTION TIME       =        0.047 SECONDS      2 Mb
```

Řešení úlohy z kapitoly 6.4

```
    C O N O P T 3   version 3.15D
    Copyright (C)   ARKI Consulting and Development A/S
                    Bagsvaerdvej 246 A
                    DK-2880 Bagsvaerd, Denmark


    The model has 5 variables and 6 constraints
    with 18 Jacobian elements, 9 of which are nonlinear.
    The Hessian of the Lagrangian has 4 elements on the diagonal,
    3 elements below the diagonal, and 4 nonlinear variables.

 ** Optimal solution. Reduced gradient less than tolerance.


 CONOPT time Total                               0.001 seconds
   of which: Function evaluations                0.000 =  0.0%
             1st Derivative evaluations          0.000 =  0.0%
             2nd Derivative evaluations          0.000 =  0.0%
             Directional 2nd Derivative          0.000 =  0.0%
```

```
--- DICOPT: No discrete variables found
--- DICOPT: Returning NLP solution

      The model does not contain any discrete variables.
      The NLP solution is returned to GAMS.
      Advice: use an NLP solver for this model.


                         LOWER     LEVEL     UPPER    MARGINAL

---- EQU ziskeq          77.700    77.700    77.700     1.000
---- EQU podminka1        -INF      4.477     4.477    22.338
---- EQU podminka3        -INF     -1.064     1.099      EPS
---- EQU podminka4        -INF      0.693     0.693     0.812
---- EQU podminka5        -INF     14.743 1.0000E+5       .
---- EQU podminka7        -INF   1574.529  1900.000       .

  ziskeq  Rovnice zisku
  podminka1  Výrobní funkce
  podminka3  Omezení na max. počet pracovníků na jednotku kapitálu
  podminka4  Omezení na zpracovaný materiál jednotkou kapitálu
  podminka5  Sledování nákladů
  podminka7  Omezení výroby

                         LOWER     LEVEL     UPPER    MARGINAL
```

```
---- VAR zisk            -INF     85.000     +INF        .

  zisk  Nákladová funkce

---- VAR z  Promenne

     LOWER     LEVEL     UPPER    MARGINAL

L     -INF      4.011     +INF        .
K     -INF      5.075     +INF        .
M     -INF      4.382     +INF        .
Q     -INF      7.362     +INF  -2.416E-9


**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED
                             0     ERRORS


EXECUTION TIME       =        0.000 SECONDS
```

Řešení úlohy z kapitoly 6.5

```
    C O N O P T 3   version 3.15D
    Copyright (C)   ARKI Consulting and Development A/S
```

```
                    Bagsvaerdvej 246 A
                    DK-2880 Bagsvaerd, Denmark


   The model has 9 variables and 7 constraints
   with 22 Jacobian elements, 7 of which are nonlinear.
   The Hessian of the Lagrangian has 7 elements on the diagonal,
   10 elements below the diagonal, and 7 nonlinear variables.

** Optimal solution. Reduced gradient less than tolerance.


CONOPT time Total                            0.055 seconds
  of which: Function evaluations             0.000 =  0.0%
            1st Derivative evaluations       0.000 =  0.0%
            2nd Derivative evaluations       0.000 =  0.0%
            Directional 2nd Derivative       0.000 =  0.0%

                        LOWER     LEVEL     UPPER    MARGINAL

---- EQU ziskeq           .         .         .        1.000
---- EQU podminka1      -INF      4.477     4.477      0.562
---- EQU podminka2      -INF        .         .        0.816
---- EQU podminka3      -INF     -1.064     1.099       EPS
---- EQU podminka4      -INF      0.693     0.693      0.020
---- EQU podminka5      -INF      1.000     1.000      1.469
---- EQU podminka6      -INF      7.313     7.313      0.816
```

```
  ziskeq  funkce zisku
  podminka1  Výrobní funkce
  podminka2  Požadavek na výrobu
  podminka3  Omezení na max. počet pracovníků na jednotku kapitálu
  podminka4  Omezení na zpracovaný materiál jednotkou kapitálu
  podminka5  Transformace ziskové funkce
  podminka6  Model poptávky

                      LOWER     LEVEL     UPPER    MARGINAL

---- VAR zisk          -INF      7.895     +INF       .

  zisk  zisk

---- VAR z  Promenne

        LOWER     LEVEL     UPPER    MARGINAL

ZISK     -INF      7.895     +INF       .
L        -INF      8.281     +INF       .
K        -INF      9.345     +INF  -3.41E-10
M        -INF      8.652     +INF       .
Q        -INF     10.180     +INF       .
N        -INF     -4.608     +INF  -6.83E-10
P        -INF     -1.900     +INF       .
D        -INF     10.180     +INF       .
```

```
**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED
                             0     ERRORS

EXECUTION TIME       =        0.000 SECONDS      2 Mb
```