tanulmányok

157/1984

MTA Számitástechnikai és Automatizálási Kutató Intézet Budapest

MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

A GEOMETRIAI PROGRAMOZÁS FEJLŐDÉSEI ÉS MEGOLDÁDSI MÓDSZEREI

Irta:

HO NGOC LUAT

Tanulmányok 157/1984

A kiadásért felelős:

DR. VÁMOS TIBOR

Főosztályvezető:

PRÉKOPA ANDRÁS

ISBN 963 311 174 9
ISSN 0324-2951

Hozott anyagról sokszorosítva

8414935 MTA Sokszorosító, Budapest. F. v.: dr. Héczey Lászlóné

TARTALOM

A dolgozat a geometriai programozás 22 éves fejlődési folyamatának áttekintése kiván lenni. Első részében a geometriai programozás elméletének fejlődésével foglalkozunk. Ebből a szempontból az eltelt időszak két különböző szakaszra osztható: az első időszakra az jellemző, hogy ekkor a pozinomos geometriai programozással foglalkoztak, a második szakaszban pedig a szignomos geometriai programozás elméleti kérdéseit is tárgyalják.

A dolgozat első részében ezen kivül még a geometriai programozás alkalmazási lehetőségeiről is emlitést teszünk.

A dolgozat második része a geometriai programozási feladat megoldására alkalmas nemlineáris programozási módszerekkel és a speciálisan geometriai programozási feladatra kidolgozott különböző eljárásokkal foglalkozik.

Külön kitérünk ebben a részben a geometriai programozási feladat olyan adottságainak ismertetésére, amelyeket a feladat megoldásánál figyelembe kell venni. Javaslatot teszünk az emlitett nehézségek elkerülésére és ismertetjük erre vonatkozó számitógépes tapasztalatokat.

1. A GEOMETRIAI PROGRAMOZÁS ELMÉLETÉNEK TÖRTÉNETE

I. A POZINOMOS GEOMETRIAI PROGRAMOZÁS

i-1. A feladat megfogalmazása:

Az eredeti egy geometriai programozási feladatnak a következő feladatot nevezték:

Primál geometriai programozási feladat (PGP):

$$g_0(t) = \sum_{i\in[0]} c_i\, t_1^{a_{i1}}.t_2^{a_{i2}}\ldots t_m^{a_{im}} \to min!$$

(1) feltéve, hogy

$$g_k(t) = \sum_{i\in[k]} c_i\, t_1^{a_{i1}}.t_2^{a_{i2}}\ldots t_m^{a_{im}} \leq 1,\ k=1,2,\ldots,p$$

ahol $[k] = \{m_k,\ m_k+1,\ldots,n_k\},\ k=0,1,\ldots,p$

$$m_0=i,\ m_1=n_0+1,\ m_2=n_1+1,\ldots,n_p=n.$$

$A=(a_{ij})$ egy valós mátrix (exponens mátrix)
$\underline{c}$ pozitiv n-dimenziós vektor.
A $g_k(.)$ függvényeket *pozinom*nak nevezzük.

Duál geometriai programozási feladat (DGP):

$$V(\underline{\delta}) = \left[\prod_{i=1}^{n} \left(\frac{c_i}{\delta_i}\right)^{\delta_i}\right] \prod_{k=1}^{p} \lambda_k^{\lambda_k} \to max!$$

feltéve, hogy

(2)

$$\sum_{i\in[0]} \delta_i = 1$$ (normalitási feltétel)

$$\sum_{i=1}^{n} a_{ij}\delta_i = 0,\ j=1,\ldots,m$$ (ortogonalitási feltétel)

ahol $\lambda_k = \sum_{[k]} \delta_i$, $k=1,2,\dots,p$; A és $\underline{c}$ a primál feladatban levő mátrix és együttható vektor.

A fenti (1) geometriai programozási feladatot először 1961-ben Zener fogalmazta meg [89], és ebben az évben Duffin-Zener [28] e cikkben példákat adott geometriai programozási feladatra. A geometriai programozás dualitási kérdéseivel Duffin és Peterson együtt foglalkozott [30]. 1967-ben megjelent az eddigi eredmények összefoglalása, Duffin-Peterson-Zener [33] könyve. A könyv korrektan tárgyalja a geometriai programozást és dualitási problémakörét, a geometriai programozás alkalmazását és a geometriai programozás uj tárgyalását is tartalmazza.

A geometriai programozási feladatpárral nagyon fontos az a tény, hogy a pozinomok általában nem konvex függvények, például $g(t)=t^{\frac{1}{2}}$, tehát közvetlenül nem látható az, hogy a geometriai programozási feladat konvex vagy nem konvex. Azonban a PGP feladat áttranszformálható egy olyan vele ekvivalens feladatra, amelyről belátható, hogy konvex programozási feladat. Legyen

$$t_j = e^{z_j} \qquad (j=1,2,\dots,n)$$

akkor a PGP feladat ekvivalens a következő feladattal:

$$g_o(\underline{z}) = \sum_{[0]} c_i \exp\left(\sum_{j=1}^{m} a_{ij} z_j\right) \to \min$$

(3) feltéve, hogy

$$g_k(\underline{z}) = \sum_{[k]} c_i \exp\left(\sum_{j=1}^{m} a_{ij} z_j\right) \leq 1, \quad (k=1,2,\dots,p)$$

ahol, amint az belátható, $g_k(.)$ konvex függvény $(\underline{c} > \underline{o})$. Igy a PGP feladat minden lokális minimuma szintén globális minimum.

A PGP egy másik (4) ekvivalens feladatát kapjuk, ha az $\underline{x}=A\underline{z}$ transzformációt alkalmazzuk (3)-ra:

$$\sum_{[0]} c_i \exp(x_i) \rightarrow \min!$$

feltéve, hogy

(4)

$$\sum_{[k]} c_i \exp(x_i) \leq 1, \quad k=1,2,\ldots,p.$$

$$\underline{x} = A\underline{z}.$$

Ez tulajdonképpen egy szeparálható programozási feladat, általános elméleti kérdéseivel Peterson [74] foglalkozik; néhány speciális alakja pedig Ecker [35]; Heymann [48]; Rijckaert [76] és [56] dolgozataiban található. A (4) feladatban az látható, hogy ha A mátrix rangja kisebb mint m (a primál változók száma), akkor néhány z_j változót nullára tehetünk, igy néhány t_j változó 1-gyel egyenlő. Tehát ennél a néhány fix változónál a PGP olyan, hogy az exponens mátrixsza m-rangu (teljes rangu). Ezért a továbbiakban feltehető, hogy az A exponens mátrix teljes rangu.

Azt mondjuk, hogy egy geometriai programozási feladat (duál vagy primál) *konzisztens*, ha van legalább egy megengedett megoldása, azaz olyan pont létezik, amely teljesiti a feladat összes feltételeit. A PGP-ről azt mondjuk, hogy szuperknzisztens, ha létezik egy $\bar{\underline{t}}$ megengedett megoldás, amelyre

$$g_k(\bar{\underline{t}}) < 1, \quad k=1,2,..,p.$$

PGP degenerált feladat ha a PGP megengedett megoldáshalmazában (feltéve, hogy ez nem üres), az optimális megoldás keresése folyamán, egy $c_i t_1^{a_{i1}} \ldots t_m^{a_{im}}$ tényező tart nullához, miközben $\underline{t}$ vektor tart optimális megoldáshoz. Ha egy PGP nem degenerált, akkor kanonikusnak nevezzük. A degeneráltságra vagy kanonikusságra jellemző, fontos állitás [33]-ban az, hogy a PGP kanonikus akkor és csak akkor ha létezik egy olyan $\bar{\underline{\delta}}$, hogy $A^T\bar{\underline{\delta}} = \underline{0}$ és $\bar{\underline{\delta}} > \underline{0}$. Nyilvánvaló az, hogy ha PGP kanonikus, akkor DGP konzisztens. Ha PGP degenerált akkor töröljük el azt az i-edik $c_i t_1^{a_{i1}} \ldots t_m^{a_{im}}$ tényezőt, amelyre $\delta_i=0$ (δ megengedett megoldás

DGP számára), akkor a megkapott un. redukált feladat már kanonikus [79]. [33]-ban Duffin-Peterson-Zener megmutatta, hogy ennek az ekvivalens redukált feladatnak optimális megoldás az eredeti PGP feladatnak is. Tehát a dualitási tételt egyszerüség kedvéért kimondjuk a kanonikus feladatra.

Dualitási tétel [33]. Ha PGP kanonikus és szuperkonzisztens, akkor

(i) a DGP feladatvan egy $\underline{\delta}^*$ maximuma

(ii) a $V(\underline{\delta}^*)$ optimális érték egyenlő a PGP optimális értékével

(iii) a PGP $\underline{t}^*$ optimális megoldására teljesül:

$$c_i t_1^{*a_{i1}} \cdot t_2^{*a_{i2}} \cdots t_m^{*a_{im}} = \begin{cases} \delta_i^* . V(\underline{\delta}^*), & i \in [0] \\ \delta_i^*/\lambda_k^*, & i \in [k], \lambda_k^* > 0. \end{cases} \tag{5}$$

és

$$\lambda_k^*[1-g_k(t^*)] = 0, \quad k\ 1,2,\ldots,p. \tag{6}$$

Megjegyezzük, hogy a dualitási tételnek feltétele biztositja, hogy a PGP felveszi minimumát valamely $\underline{t}^*$ pontban [33]. Az (5) reláció módot ad arra, hogy meghatározzuk a primál feladat optimális megoldását a duál feladat optimális megoldása ismeretében és forditva. Ez a meghatározás egyértelmü, hiszen a $\log(t_j)$ változóban (5) egy olyan lineáris egyenletrendszer, amelyre A együttható mátrix teljes rangu. A (6) komplementaritási reláció szerint, ha $g_k(t^*)<1$, akkor $\lambda_k^*=0$ ilyen esetben (5) egyenletrendszer nem adja egyértelmüen a $\underline{t}^*$ optimális megoldást. Erre az esetre vonatkozóan [33], [4] megmutatja, hogy kisegitő feladatokat kell megoldani a primál feladat optimális megoldása meghatározására.

Az $n-(m+1)$ számot, ahol n a duál változók száma, ez szintén a primál feladatban levő összes $c_i t_1^{a_{i1}} \ldots t_m^{a_{i2}}$ tényezők száma és m a primál változók száma ez szintén az ortogonalitási feltételek száma, a geometriai programozási feladat *nehézségi fok*ának nevezzük.

i-2. Egy példa a pozinomos geometriai programozásra

Tekintsük a következő szállitóhajó problémát [42]. Tegyük fel, hogy van egy nagy szállitóhajó, ami tele van a szálitmánynyal. A szállitmányt ki kell pakolnunk a kikötőbe - mondjuk - kis hajókkal. A kis hajókat előre kell gyártanunk. Nyilvánvaló, hogy minél nagyobb a hajó kapacitása (tonna), annál költségesebb a gyártása. Ezen kivül ha a sok lóerejü motort szerelnénk fel, akkor többet kell fizetni, mint kisebb lóerejü motor beszerzése esetében. Milyen hajókat gyártsunk, hogy annak a költsége, hogy hajókat gyártunk és a szállitmányt a hajókkal partra kipakoljuk, minimális legyen?

Ha a hajók száma t_1, kapacitásuk t_2, akkor az egyszeri kapacitás $t_1.t_2$. A költség első része arányos $t_1.t_2$-vel, legyen c_1 a költségi tényező, akkor ez $c_1.t_1.t_2$ lesz. A költség másik része a motor beszerzési költsége, ami arányos $t_2^{\frac{2}{3}}.t_3^3$-mal, ahol t_3 a hajó forgalmi sebessége. Legyen c_2 a második költségi tényező. Akkor minimalizálni kell

$$g_0(t) = c_1 t_1 t_2 + c_2 t_1 t_2^{2/3} t_3^3$$

függvényt.

A következő feltételeknek kell teljesülniük:

i) Egy hajó szállitási hatékonyságának bizonyos alsó korlát felett kell lennie. A hatékonyság arányos a hajó kapacitásával és a hasznos idővel, ez utóbbi viszont a hajó sebességével és egy hasznossági tényezővel (t_4) arányos, vagyis

$$t_1 t_2 t_3 t_4 \geq c_3$$

ami felirható

$$g_1(t) = {}^{c_3}/_{t_1 t_2 t_3 t_4} \leq 1$$

alakban is, ahol c_3 adott alsó korlát.

ii) A hasznossági tényező a tengeren tartozkódó időnek és az egész szállitási menet idejének relativ értéke. A szállitás közben igyekszünk a

$$\text{hasznossági tényező} \leq \frac{\text{utazási idő}}{(\text{utazási idő}+\text{várakozási idő})}$$

egyenlőtlenség teljesülését betartani. A várakozási idő tulajdonképpen a hajó kapacitásának explicit függvénye, az utazási idő pedig arányos ${}^{1/}t_3$-mal. Igy ez a feltétel alábbi alakban felirható:

$$g_2(t) = c_4t_2t_3t_4 + c_5t_4 \leq 1$$

ahol c_4, c_5 arányossági tényezők.

Összefoglalva,

$$\min g_o(t) = c_1t_1t_2 + c_2t_1t_2^{2/3}t_3^3$$

feltéve, hogy

$$g_1(t) = c_3t_1^{-1}t_2^{-1}t_3^{-1} \leq 1 \tag{7}$$

$$g_2(t) = c_4t_2t_3t_4 + c_5t_4 \leq 1$$

$$t_j > 0, \quad j=1,2,3,4.$$

Ennél a feladatnál $[0] = \{1,2\}$; $[1] = \{3\}$; $[2] = \{4,5\}$, az exponens mátrix pedig

$$A = \begin{array}{c} \left(\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 1 & \frac{2}{3} & 3 & 0 \\ -1 & -1 & -1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{array}\right) \end{array} \begin{array}{l} \} [0] \\ \\ \} [1] \\ \} [2] \\ \\ \end{array}$$

Igy (7) feladatnak duál feladata a következő:

$$\max \{v(\delta) = \prod_{i=1}^{5} \left(\frac{c_i}{\delta_i}\right)^{\delta_i} \lambda_1^{\lambda_1} \lambda_2^{\lambda_2}\}$$

feltéve, hogy

$$(8) \qquad \begin{aligned} \delta_1 + \delta_2 \quad &= 1 \\ \delta_1 + \delta_2 - \delta_3 \quad &= 0 \\ \delta_1 + \tfrac{2}{3}\delta_2 - \delta_3 + \delta_4 \quad &= 0 \\ 3\delta_2 - \delta_3 + \delta_4 \quad &= 0 \\ \delta_4 + \delta_5 &= 0 \\ \delta_i \geqq 0,\ i=1,\dots,5 & \end{aligned}$$

ahol $\lambda_1 = \delta_3,\ \lambda_2 = \delta_4 + \delta_5.$

A (8) lineáris egyenletrendszernek van egyetlen egy megoldása:

$$\delta_1^* = \frac{7}{10};\quad \delta_2^* = \frac{3}{10};\quad \delta_3^* = 1;\quad \delta_4^* = \frac{1}{10};\quad \delta_5^* = \frac{9}{10}$$

Igy

$$V(\delta^*) = 2{,}5496 \cdot c_1^{7/10} c_2^{3/10} c_3 c_4^{1/10} c_5^{9/10}.$$

A $\underline{c}$ vektor ismeretében az (5) egyenletrendszert alkalmazva meg lehet határozni a primál feladat optimális megoldását. Ebben az esetben egyértelmüen tudjuk megkapni az optimális megoldást, hisz $\underline{\delta}^* > \underline{0}$, vagyis kanonikus feladatról van szó. A $\underline{\delta}^*$ vektort azért volt könnyü meghatározni, mert $(5-(4+1))$ nulla nehézségi foku geometriai programozási feladat volt a (7) feladat.

Sokkal bonyolultabb a feladat, amikor $n-(m+1) > 0$. Ebben az esetben a DGP feladat visszavezethető egy $n-(m+1)$ változótól függő feladatra, tudniillik a $\underline{\delta}$ vektor ugy irható [33]:

$$\delta_i = b_i^o + \sum_{j=1}^{n-(m+1)} b_i^j\, y_j \qquad (i=1,2,\dots,n)$$

ahol $\underline{b}^o$ normalitási vektor teljesiti mind a normalitási feltételt mind az ortogonalitási feltételeket; a $\underline{b}^j$ nullitási vektorok pedig a DGP feltételi egyenletek homogén lineáris egyenletrendszerének a megoldásai. Igy a DGP feladat következőképpen felirható:

$$V(\underline{y}) = \prod_{i=1}^{n} \left({}^{c_i}/_{\delta_i(\underline{y})} \right)^{\delta_i(\underline{y})} \prod_{k=1}^{p} \lambda_k^{\lambda_k} \to max!$$

feltéve, hogy

(9) $$\underline{\delta}(\underline{y}) = \underline{b}^o + \sum_{j=1}^{n-(m+1)} \underline{b}^j y_j \geq \underline{0}$$

ahol $\lambda_k = \sum_{i \in [k]} \delta_i(\underline{y}),\ k=1,2,\dots,p.$

Itt látható, hogy $\underline{\delta}(y)$ azon a tartományon van, amelyet a $\underline{b}^j$ $(j=1,2,\dots,n-(m+1))$ vektorok által alkotott altér $\underline{b}^o$-val való eltolásának és R_+^n pozitiv ortánsnak a metszete ad.

2. SZIGNOMOS GEOMETRIAI PROGRAMOZÁS

2.1 A feladat megfogalmazása

A pozinomos geometriai programozás számos müszaki probléma leirására alkalmas. Azonban a valóságban sok fontos müszaki probléma esetén a pozinomos geometriai programozás-beli c_i-re vonatkozó pozitivitási feltétel nem teljesül. Ezért vált szükségessé az olyan geometriai programozási probléma vizsgálata, ahol a c_i-k akármilyen előjelüek lehetnek. Ezeket a geometriai programozási problémákat nevezzük szingomos geometriai prograramozási (SGP) feladatnak.

A szignomos geometriai programozás fogalmát először Passy-Wilde [71], ezt követően Blau-Wilde [10] vezették be. Annak primál alakja az alábbi:

$$f_o(\underline{t}) = \sum_{i\in[0]} u_i(\underline{t}) \to min!$$

feltéve, hogy

(10)
$$f_k(\underline{t}) = \sum_{i\in[k]} u_i(\underline{t}) \leq 1,\ k=1,2,\dots,p$$

$$f_k(\underline{t}) = \sum_{i\in[k]} u_i(\underline{t}) \geq 1,\ k=p+i,\dots,q.$$

$$\underline{t} > \underline{0}$$

ahol $u_i(\underline{t}) = c_i t_1^{a_{i1}} \dots t_m^{a_{im}}$,

A (nxm) valós mátrix, $\underline{c}$ n-dimenziós *valós* vektor és $[0]$, $[k]$ diszjunkt indexhalmazok - mint (1) feladatnál -, amelyek együttesen a $\{1,2,\dots,n\}$ halmazt adják.

A (10) feladat duál feladatát bonyolultsága miatt nem közöljük inkább ismertetjük [32] gondolatmenetét, ami szerint a szignomokat visszavezetjük a pozinomokra és igy a SGP egy ekvivalens pozinomos GP feladatát adhatjuk meg. És tárgyaljuk az utóbbi feladatot.

A szignomos geometriai programozási feladat feltételei általában három csoportra oszthatók:

$$f(\underline{t}) \leq -1,\ f(\underline{t}) \leq 0,\ f(\underline{t}) \leq 1,$$

ahol $f(\underline{t})$ a feladatban levő szignomok. Át fogjuk transzformálni a fenti feltételeket az alábbi két fajtára

(11) $$g(\underline{t}) \leq 1,\quad g(\underline{t}) \geq 1$$

ahol $g(\underline{t})$ már egy pozinom.

a) Tekintsük az $f(\underline{t}) \leq -1$ esetet.

- Ha $f(\underline{t})$ pozinom, akkor a PSGP (10) nem konzisztens.
- Ha $f(\underline{t})$ pozinom -1-szerese, akkor legyen $g(\underline{t})=-f(\underline{t})$ és ez a feltétel $g(\underline{t}) \geq 1$ alakban irható.
- Végül ha $f(\underline{t})=f_1(\underline{t})-f_2(\underline{t})$, ahol $f_1(\underline{t})$ és $f_2(\underline{t})$ pozinomok, akkor vezessük be a t_{m+1} pozitiv változót a következő módon. Legyen t_{m+1} olyan pozitiv szám, amelyre

$$1 + f_1(\underline{t}) \leq t_{m+1} \leq f_2(\underline{t})$$

teljesül, ahol $\underline{t}$ a (10)-nek egy megengedett megoldása. Ha $\underline{t}$ megengedett megoldás nincs, akkor (10) feladat nem konzisztens. Igy bevezetve a

$$g_1(\underline{t}, t_{m+1}) := t_{m+1}^{-1} [1+f_1(\underline{t})]$$

és

$$g_2(\underline{t}, t_{m+1}) := t_{m+1}^{-1} \; f_2(\underline{t})$$

függvényeket, ebben az esetben

$$g_1(\underline{t}, t_{m+1}) \leq 1$$

$$g_2(\underline{t}, t_{m+1}) \geq 1$$

feltételek már (11) alakuak.

b) Tekintsük most az $f(\underline{t}) \leq 0$ esetet.

- Ha $f(\underline{t})$ pozinom, akkor (10) feladat nem konzisztens.
- Ha $f(\underline{t})$ egy pozinom -1-szerese, akkor ez a feltétel kihagyható.
- Végül az az eset, amikor $f(\underline{t})$ kifejezhető két pozinomnak a különbségével, az a) esethez hasonlóan tárgyalható.

c) Tekintsük végül az $f(\underline{t}) \leq 1$ esetet.

- Ha $f(\underline{t})$ pozinom, akkor ez már (ii) alaku.
- Ha $f(t)$ egy pozinom -1-szerese, akkor ez a feltétel kihagyható. Amennyiben $f(\underline{t})$ kifejezhető két pozinom különbségével, akkor mint az a) esetben visszavezethető két (11) alaku feltételre.

Ezek után tegyük fel, hogy feladatunk a következőképpen felirható:

$$(12) \qquad \min f_o(\underline{t})$$

feltéve, hogy

$$(13) \qquad \begin{array}{ll} g_k(\underline{t}) \leq 1 & k=1,2,\dots,p \\ g_k(\underline{t}) \geq 1 & k=p+1,\dots,q \\ \underline{t} > \underline{0} & \end{array}$$

ahol $g_k(\cdot)$ pozinomok.

Ha az $f_o(\underline{t})$ függvény egy pozinom, akkor (12)-(13) egy olyan pozinomos PGP feladat, melynél nem csak $g_k(\underline{t}) \leq 1$ alaku feltételek vannak, hanem a $g_k(\underline{t}) \geq 1$ alaku feltételek is szerepelnek. Ezt a pozinomos PGP feladatot *forditott pozinomos primál geometriai programozási feladatnak* (röviden RPGP: Reversed Primal Geometric Program) nevezzük.

Ha $f_o(\underline{t})$ nem egy pozinomos, akkor egy t_o uj pozitiv változót bevezetünk és ezzel a célfüggvényt át fogjuk transzformálni egy (11) alaku feltétellé. Az áttranszformálás a célfüggvénynek a (13) feltételek melleti infimuma előjelétől függ.

- Ha az infimum előjele pozitiv, akkor (12)-(13) feladatunk azzá válik, hogy minimalizáljuk a t_o változót feltéve, hogy $f_o(\underline{t}) \leq t_o$ és (13)-beli feltételek fennállása. Ez az első áttranszformáció.

- Ha most az infimum előjele negativ, akkor a (12)-(13) feladat ugy fogalmazható meg, hogy maximalizáljuk a t_o változót feltvéte, hogy $f_o(\underline{t})+t_o \leq 0$ és (13)-beli feltételek teljesülése. Ez un. második áttranszformáció.

Igy az $f_o(\underline{t}) \leqq t_o$ vagy $f_o(\underline{t})+t_o \leqq 0$ feltételt az a), b) vagy c) esethez hasonlóan visszavezethetjük a (11) alaku feltételekre.

Ami az infimum előjelét illeti, azt előre nem nagyon lehet megtudni. Igy bizonyos találgatás szükséges az infimum előjelének megismerésére, és ennek alapján eldöntjük, hogy melyik áttranszformáció alkalmas. Ha az első áttranszformációt végeztük és a feladat infimuma nulla, akkor a második áttranszformáció alkalmazásával megtudjuk, hogy vajon az infimum nem-e egy negativ érték. Ha már az elsőként a második áttranszformációt választottuk és a feladat nem konzisztens, akkor az első áttranszformáció elvégzésével megtudjuk, hogy vajon nem pozitiv-e az infimum.

Tehát az eddig tárgyaltak szerint az eredeti primál signomos geometriai programozási feladatot átalakithatjuk a következő forditott pozinomos primál geometriai programozási feladatra:

<u>RPGP:</u> $\min g_o(\underline{t})$

feltéve, hogy

(14)
$$g_k(\underline{t}) \geqq 1 , \qquad k=1,2,\dots,p$$
$$g_k(\underline{t}) \leqq 1 , \qquad k=p+1,\dots,q$$
$$\underline{t} > 0$$

ahol

$$g_k(\underline{t}) = \sum_{[k]} u_i(\underline{t}) , \quad k=0,1,\dots,p,\dots,q$$

és

$$u_i(\underline{t}) = \begin{cases} c_i t_i^{a_{i1}} . t_2^{a_{i2}} \dots t_m^{a_{im}} & i \in [k], \ k=0,1,\dots,p \\ c_i t_1^{-a_{i1}} . t_2^{-a_{i2}} \dots t_m^{-a_{im}} & i \in [k], \ k=p+1,\dots,q. \end{cases}$$

a_{ij} valós számok, c_i-k pozitivak.

Jelöljük

$$P = \{1,2,\dots,p\}$$
$$R = \{p+1,\dots,q\}$$

és

$$[K] = \bigcup_{k\in K} [k], \qquad K \subseteq \{0\} \cup P \cup R.$$

továbbá

$$K(\underline{\delta}) := \{k\in K \mid \lambda_k \neq 0\} \qquad K \subseteq \{0\} \cup P \cup R$$
$$[K](\underline{\delta}) := \{i\in [K] \mid \delta_i \neq 0)\} \quad K \subseteq \{0\} \cup P \cup R.$$

Akkor a RPGP duál feladata következőképpen felirható:

<u>RDGP:</u> Keresendő az alábbi függvénynek a maximuma:

$$V(\underline{\delta}) = \left\{ \left[\prod_{[0](\delta)} \left({}^{c_i}/_{\delta_i} \right)^{\delta_i} \right] \left[\prod_{[R](\delta)} \left({}^{c_i}/_{\delta_i} \right)^{\delta_i} \right] \left[\prod_{[R](\delta)} \left({}^{c_i}/_{\delta_i} \right)^{-\delta_i} \right] \right\}$$

$$\times \left\{ \left[\prod_{P(\delta)} \lambda_k^{\lambda_k} \right] \left[\prod_{R(\delta)} \lambda_k^{-\lambda_k} \right] \right\}.$$

feltéve, hogy

$$\delta_i \geq 0, \quad i \in \{1,2,\dots,n\}$$
$$\lambda_o = 1 \tag{15}$$
$$\sum_{i=1}^{n} a_{ij}\, {}_i = 0, \quad j=1,2,\dots,m.$$

ahol

$$\lambda_k = \sum_{[k]} \delta_i, \qquad k\in\{0,1,\dots,q\}$$

A RDGP feladat esetében fontos az, hogy $log V(\underline{\delta})$ a δ_i, $i \in [0] \cup [P]$ változóknak a konkáv függvénye és konvex a δ_i, $i \in [R]$ változókra nézve [32]. Ebben az esetben nem minden lokális megoldás optimális megoldás. Azokat a megoldásokat, melyekre teljesülnek az (5)-(6) optimális feltételek, nevezzük *egyensulyi megoldásnak*. Pontosabban, a RPGP feladatnak egy $\underline{t}^*$ megengedett megoldását primál egyensulyi megoldásnak nevezzük, ha létezik a RDGP feladatnak egy olyan δ^* megengedett megoldása, amelyre

$$\text{(16)} \quad \begin{aligned} \delta_i^* \, g_o(\underline{t}^*) &= u_i(\underline{t}^*), \quad i \in [0] \\ \delta_i^* &= \lambda_k^* \, u_i(\underline{t}^*), \quad i \in [k], \; k \in P \cup R. \end{aligned}$$

Ebben az esetben $\underline{\delta}^*$ megoldást duál egyensulyi megoldásnak nevezzük.

A geometriai programozás fõlemmája [33], ami a GP konvexitási tulajdonságát felhasználva biztositja azt a fontos tényt, hogy a primál célfüggvény értéke egy megengedett megoldásnál mindig nem kisebb mint a duál célfüggvény értéke bármely duál megengedett megoldásnál. Az egyenlõség akkor és csak akkor áll fenn, ha a primál megengedett és a duál megengedett megoldás teljesiti az (5) optimalitási feltételt.

Az RDGP feladatnak a nemkonvexitási tulajdonsága miatt, egy egyensulyi megoldáspárra nem mindig optimális megoldás. De arra tényre hivatkozunk, hogy egy általános nemlineáris programozási feladatnak bizonyos regularitási feltétel melletti lokális megoldása mindig kritikus megoldása a feladatra feliró Lagrange-problémának. Igy egy primál lokális megoldás primál egyensulyi megoldás is. Tehát a szignomos geometriai programozási feladat optimális megoldásai az egyensulyi megoldások között vannak.

A továbbiakban a geometriai programozásnak néhány uj és fontos alakját ismertetjük, amelyet gyakran felhasználnak a geometriai programozási feladatnak megoldása meghatározására.

2.2 A geometriai programozás lineáris alakja

Ha a PGP feladat olyan, hogy minden pozinoma egy tagu tényező (monom), azaz $g_k(\underline{t})=u_k(\underline{t})$, $k=0,1,\ldots,p$, akkor ennek a logaritmus vétele után egy ekvivalens lineáris programozási feladatot kapunk. A lineáris programozási feladatban $\log t_j$ a változó. Ezt a lehetőséget Duffin [29] tovább fejlesztette az általános GP-re oly módon, hogy a pozinomokat a geometriai egyenlőtlenség segitségével "süritjük" monomokra. A geometriai egyenlőtlenség ugy felirható:

$$(17)\qquad \sum_{i\in[k]} u_i \geq \prod_{[k]} \left(\frac{u_i}{\varepsilon_i}\right)^{\varepsilon_i}, \quad k=0,1,\ldots,p$$

ahol $u_i>0$, $\varepsilon_i\geqq 0$, $i=1,2,\ldots,n$ és $\sum_{[k]} \varepsilon_i=1$, $k=0,1,\ldots,p$.

Igy adott egy PGP és $\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_n$ nem negativ számok - olyanok, hogy $\sum_{[k]} \varepsilon_i=1$, $k=0,1,\ldots,p$ - esetén tekintsük a következő feladatot:

<u>LPGP:</u> $\min \overline{g_0}(\underline{t})$

feltéve, hogy

$$(18)\qquad \bar{g}_k(\underline{t}) \leqq 1, \quad k=1,2,\ldots,p$$

$$\underline{t} > \underline{0}$$

ahol

$$\bar{g}_k(\underline{t}) = \prod_{[k]} \left(\frac{u_i(\underline{t})}{\varepsilon_i}\right)^{\varepsilon_i}$$

vagyis

$$\bar{g}_k(\underline{t}) = \bar{c}_k t_1^{\bar{a}_{k1}} . t_2^{\bar{a}_{k2}} \ldots t_m^{\bar{a}_{km}} =: \bar{u}_k(\underline{t})$$

ahol

$$\bar{c}_k = \prod_{[k]} \left(\frac{c_i}{\varepsilon_i}\right)^{\varepsilon_i}$$

$$\bar{a}_{kj} = \sum_{[k]} \varepsilon_i a_{ij}, \qquad (j=1,2,\ldots,m)$$

Ennek a (18) feladatnak a duális feladata a következő:

LDGP: $$\max \bar{v}(\underline{\lambda}) = \prod_{k=0}^{p} \bar{c}_k^{\lambda_k}$$

$$\lambda_k \geqq 0, \quad k=1,2,\ldots,p$$

$$\lambda_o = 1$$

$$\sum_{k=0}^{p} \lambda_k \bar{a}_{kj} = 0 \qquad j=1,2,\ldots,m$$

Ha logaritmust veszünk két feladatra, akkor egy olyan feladatpárt kapunk, amely a $\log t_j$ és λ_k változókra nézve, mint primál és duál változó, egy lineáris programozási feladatpár.

Meg kell emliteni, hogy LPGP csak egy közelitő megoldást ad PGP feladatnak, tudniillik, ha $\underline{t}$ megengedett megoldása PGP-nek akkor

$$1 \geqq g_k(\underline{t}) \geqq \bar{g}_k(\underline{t})$$

egyenlőtlenségből következik, hogy $\underline{t}$ is megengedett megoldása a LPGP feladatnak. Az ε_i alkalmas megválasztásával a LPGP optimális megoldása egyaránt optimális megoldása a PGP feladatnak. Igaz a

Tétel [29]. Ha PGP superkonzisztens, a primál célfüggvény felveszi M infimumát valamely $\underline{t}^*$-nál. Akkor ha

$$\varepsilon_i = \frac{u_i(\underline{t}^*)}{\sum_{[k]} u_i(\underline{t}^*)}, \quad k=0,1,\ldots,p$$

megválasztással a LPGP felveszi optimális értékét $(\bar{M})\underline{t}^*$-ban és $\bar{M} = M$.

Általában $M \leqq \bar{M}$. Ez a süritési technika alkalmazható a RPGP feladatra is. Ha például két feltételünk van

$$G_1(\underline{t}) \leq 1$$
$$G_2(\underline{t}) \geq 1,$$

akkor a süritési technikát alkalmazva $G_2(\underline{t}) \geq 1$ feltételre, egy uj feladatot kapunk, ennek optimális értéke legyen $\bar{M}$, valamely $\bar{\underline{t}}$ pontnál. Nyilvánvaló, hogy a $G_2(\underline{t}) \geq 1$ feltétel átváltozott a $\bar{G}_2(\underline{t}) \geq 1$ feltételre, vagyis $[\bar{G}_2(\underline{t})]^{-1} \leq 1$, mert $\bar{G}_2(\underline{t})$ egy monom. Alkalmazzuk tovább a süritési technikát $\varepsilon_i = u_i(\bar{\underline{t}})/G_2(\bar{t})$ megválasztással, akkor az uj feladat optimális értéke legyen $\bar{\bar{M}}$, melyre igaz a

$$M \leq \bar{\bar{M}} \leq \bar{M}$$

egyenlőtlenség.

2.3 Kiegészitő geometriai programozási feladat

Egy kiegészitő geometriai programozási feladatnak nevezünk egy olyan geometriai programozási feladatot, amelyben a függvények mind a pozinomok racionális függvénye. Igy egy kiegészitő GP következőképpen irható:

$$R_o(\underline{t}) \to min!$$

feltéve, hogy

$$R_k(\underline{t}) \leq 1, \qquad k=1,2,\ldots,p$$
$$\underline{t} > \underline{0}$$

ahol

$$(19) \qquad R_k(\underline{t}) = \frac{A_k(\underline{t}) - B_k(\underline{t})}{C_k(\underline{t}) - D_k(\underline{t})}, \qquad k=0,1,\ldots,p$$

és A_k, B_k, C_k, D_k mind pozinomok és bármelyik hiányozhat a (19) kifejezésből.

Vonjuk feltételek közé a $R_o(\underline{t})$ függvényt is olyan módon, hogy bevezetünk egy uj t_o változót, amelyre

$$t_o \leqq R_o(\underline{t})$$

egyenlőtlenséget irunk, azaz $\dfrac{R_o(\underline{t})}{t_o} \leqq 1$

Ezzel a kiegészitő GP igy irható:

$$t_o \to min!$$

feltéve, hogy

$$\frac{A(\underline{t}) - B(\underline{t})}{C(\underline{t}) - D(\underline{t})} \leqq 1 \qquad (20)$$

$$t = (t_o, t_1, \ldots, t_m) > 0$$

A (20) feltétel akkor értelmes, ha $C(\underline{t}) - D(\underline{t})$ állandó előjelü, azaz ha $C(\underline{t}) - D(\underline{t})$ pozitiv valamely $\underline{t}$ megengedett megoldásnál akkor pozitiv az egész megengedett megoldáshalmazon is. Igy annak megfelelően, hogy $C(\underline{t}) - D(\underline{t})$ pozitiv vagy negativ, a feltétel átrendezhető

$$\frac{A(t) + D(t)}{C(t) + B(t)} \leqq 1$$

vagy

$$\frac{A(\underline{t}) + D(\underline{t})}{C(\underline{t}) + B(\underline{t})} \geqq 1$$

alakra.

A kiegészitő GP, tehát, standard alakja a következő:

<u>KGP:</u> $t_o \to min!$

feltéve, hogy

$$\frac{P_k(\underline{t})}{Q_k(\underline{t})} \leq 1, \quad k=0,1,\ldots,p$$

$$\underline{t} = (t_o, t_1, \ldots, t_m) > \underline{0}$$

ahol $P_k(\underline{t}) = \sum\limits_{[k]} u_i(\underline{t})$; $Q_k(\underline{t}) = \sum\limits_{[k]} w_i(\underline{t})$.

Az Avrie-Williams [4] algoritmusa a KGP feladat megoldása meghatározására, ugy szól: tekintsünk egy $\bar{\underline{t}}$ megengedett megoldást akkor minden k index esetén legyen

$$\varepsilon_i = \frac{w_i(\bar{\underline{t}})}{Q_k(\bar{\underline{t}})},$$

igy $\sum\limits_{[k]} \varepsilon_i = 1.$

Ezzel a (17) egyenlőtlenség ugy irható:

$$Q_k(\underline{t}) = \sum_{[k]} w_i(\underline{t}) \geq \prod_{[k]} \left(\frac{w_i(\underline{t})}{w_i(\bar{\underline{t}})} Q_k(\bar{\underline{t}})\right)^{\frac{w_i(\bar{\underline{t}})}{Q_k(\bar{\underline{t}})}} =: \bar{Q}_k(\underline{t}, \bar{\underline{t}}). \tag{21}$$

Könnyen meggyőződhető, hogy $\bar{Q}_k(\underline{t}, \bar{\underline{t}})$ egy monom. Igy a KGP feladatot a következő menet szerint megoldjuk: egy $\underline{t}^{(1)}$ megengedett megoldással kezdünk. Helyettesitjük a $Q_k(\underline{t})$ pozinomokat a (21) monomokkal, megoldjuk ezt a PGP feladatot, ami tulajdonképpen ekvivalens egy lineáris programozási feladattal. Megkapjuk a $\underline{t}^{(2)}$ uj megengedett megoldást és ezzel kezdhetjük az uj lépést ... kapunk egy $\{\underline{t}^{(l)}\}$ sorozatot, ahol $\underline{t}^{(l+1)}$ optimális megoldása a következő PGP_l feladatnak:

<u>PGP_l:</u> $t_o \to min!$

feltéve, hogy

$$\frac{P_k(\underline{t})}{\bar{Q}_k(t,t^{(l)})} \leq 1, \quad k=0,1,\ldots,p$$

$$t = (t_o, t_1, \ldots, t_m) > 0$$

Itt látható, hogy a PGP_l megengedett megoldáshalmaza, igy az optimális megoldáshalmaza is részhalmaza a KGP megengedett megoldáshalmazának. A $\{t^{(l)}\}$ sorozat limesze - mint Avrie-Williams [4] bizonyitotta, hogy létezik - a KGP egy lokális megoldását adja. Érdekes az, hogy a PGP_l feladat nehézségi foka kisebb lett az eredeti standard feladaténál.

2.4 A geometriai programozás egy matematikai változata

Az elmult időszakban a geometriai programozási probléma megfogalmazása elterjedt az olyan általános megfogalmazásra is, mely hasonló bármely matematikai programozási feladat megfogalmazásához. A geometriai programozás - ilyen szempontból - általános elméleti kérdéseinek tárgyalása összefoglalóan található Peterson [74] cikkében. Ez az általános megfogalmazás nagy hatással gyakorol az alkalmazás sok területére. Érdemes itt egy speciális megfogalmazást megismertetni és annak nézány alkalmazási lehetőségét.

Az *egyszerü általános geometriai programozás*: Minimalizáljuk a $g(\underline{x})$ függvényt feltéve, hogy:

$$\underline{x} \in X \cap C$$

ahol X konvex kup az R^n-ben, $g(\cdot)$ konvex függvény a zárt konvex C halmazon.

A (3) feladat éppen ilyen formába irható, ahol $\underline{z}$ vektor változó az $X \equiv R^n$ konvex kupból van C pedig a feltételi konvex függvények értelmezési tartományának a közös része. E feladat un. primál feladat. Annak a duál feladata a következő:

$$\min g^*(\underline{y})$$

feltéve, hogy $\underline{y} \in C^* \cap Y$

ahol $$g^*(\underline{y}) = \sup_{\underline{x} \in C} (\underline{y}^T\underline{x} - g(\underline{x}))$$

és $$C^* = \{\underline{y} \in R^n \mid \sup_{\underline{x} \in C} (\underline{y}^T\underline{x} - g(\underline{x})) < \infty\}$$

$$Y = \{\underline{y} \in R^n \mid \underline{y}^T\underline{x} \geq 0 \text{ minden } \underline{x} \in X\}$$

Ennél a feladatnál az optimális feltételek az alábbiak:

$$g(\underline{x}) + g^*(\underline{y}) = 0$$

$$\underline{y}^T\underline{x} = 0$$

$$\underline{x} \in C \cap X, \quad \underline{y} \in C^* \cap Y$$

$$\underline{x} \in \delta g^*(\underline{y}), \quad \underline{y} \in \delta g(\underline{x})$$

ahol $\delta g(\underline{x})$ jelöli a $g(\underline{x})$ függvény $\underline{x}$-beli subgradiensét

$$\delta g(\underline{x}) = \{\underline{y} \mid g(\underline{z}) \geq g(\underline{x}) + \underline{y}^T(\underline{z}-\underline{x}), \text{ minden } \underline{z} \in C\}.$$

Nyilván, ha $g(\cdot)$ differenciálható, akkor

$$\underline{x} = \nabla g^*(\underline{y}), \quad \underline{y} = \nabla g(\underline{x}).$$

Ennek a formának egy alkalmazása az elosztási probléma [80]. Tegyük fel, hogy van N fajta késztermékünk. Mindegyik fajtából van bizonyos mennyiség adott alsó és felső korláttal. Mindegyik fajtához hozzárendelünk egy un. hasznossági függvényt, ami monoton növő konkáv. Kérdés az, hogy bizonyos feltétel mellett milyen eloszlása legyen a késztermékeknek, hogy a hasznosságuk maximális legyen, azaz

$$\max \sum_{i=1}^{N} Q_i(x_i)$$

feltéve, hogy

$$\sum_{i=1}^{N} x_i \leqq B \tag{22}$$

$$a_i \leqq x_i \leqq b_i, \quad i=1,2,\dots,N$$

ahol $Q_i(\cdot)$ differenciálható növő konkáv függvény minden pozitiv x_i-re, B egy konstans, $0 \leqq a_i \leqq b_i$ adottak $i=1,2,\dots,N$. A feltevésünk az, hogy a (22) feladatnak van legalább egy megengedett megoldása.

Hogy jobban össze tudjuk hasonlitani e feladatot az általános alakkal, irjuk át az alábbi formába:

$$\min - \sum_{i=1}^{N} Q_i(x_i)$$

feltéve, hogy

$$\underline{x} \in X \cap C$$

ahol

$$C = \{(\underline{x}^T, \lambda)^T \mid a_i \leqq x_i \leqq b_i, \; i=1,\dots,N; \; \lambda=B\}$$

$$X = \{(\underline{x}^T, \lambda)^T \mid \lambda - \sum_{i=1}^{n} x_i \geqq 0\}$$

Tehát a λ uj változó bevezetésével X egy konvex kup, C konvex zárt halmaz. Határozzuk meg a célfüggvény konjugált függvényét:

$$\sup_{(\underline{x}^T,\lambda)^T \in C} (\underline{y}^T \underline{x} + v\lambda + \sum_{i=1}^{n} Q_i(x_i))$$

$$= Bv + \sum_{i=1}^{N} \sup_{a_i \leqq x_i \leqq b_i} (y_i x_i + Q_i(x_i))$$

$$= Bv + \sum_{i=1}^{N} Q_i^{*}(y_i)$$

ahol

$$Q_i^*(y_i) = \begin{cases} a_i y_i + Q_i(a_i) & ha \quad y_i + \nabla Q_i(a_i) < 0 \\ b_i y_i + Q_i(a_i) & ha \quad y_i + \nabla Q_i(b_i) > 0 \\ \sup\limits_{x_i} (y_i z_i + Q_i(x_i)) & egyébként \end{cases}$$

Az X konvex kup igy irható

$$X = \{(x^T, \lambda)^T \mid A(x^T, \lambda) \geqq 0\}$$

ahol $A = (-1,-1,\dots,-1,1)$ $1x(N+1)$- mátrix.
Ekkor

$$Y = \{(\underline{y}^T, v)^T \mid (y^T, v) = A\alpha,\ \alpha \geqq 0\},$$

azaz

$$y^T = -\alpha(1,1,\dots,1)$$

$$v = \alpha$$

$$\alpha \geqq 0$$

Ezek szerint a $Q_i^*(.)$ függvény igy irható:

$$\begin{cases} -a_i\alpha + Q_i(a_i) & ha \quad \nabla Q_i(a_i) < \alpha \\ -b_i\alpha + Q_i(b_i) & ha \quad \nabla Q_i(b_i) > \alpha \\ \sup\limits_{x_i} (-\alpha x_i + Q_i(x_i)) & egyéb. \end{cases}$$

És annak a duál feladata:

$$\min_{\alpha \geqq 0} \{B\alpha + \sum_{i=1}^{N} Q_i^*(-\alpha)\}$$

Vagyis duál feladat egy egyváltozós programozási feladat.

A problémának van egy másik érdekes alkalmazása is a hidraulikus hálókra, ez röviden igy fogalmazható meg: meghatározandó vizhálózat esetén az egyes csövekben folyó viz mennyisége vagy nyomása, ha ismert a befolyó és kifolyó mennyiség vagy a forrási feszültség és a nyelő feszültség. Részletes elemzés található [50]-ben.

2.5 Sztochasztikus geometriai programozás

A Dantzig-Madansky féle kétlépcsős sztochasztikus lineáris programozás mintájára Avriel-Wilde [5] megfogalmazta a kétlépcsős sztochasztikus geometriai programozási feladatot. A véletlen szerepét itt a $\underline{c}$ együttható vektor játsza. Amellett a feltétel mellett, hogy $\underline{c}$ vektor minden komponense korlátos valószinüségi eloszlás, azaz létezik egy λ_i, amelyre

$$0 < c_i \leq \lambda_i \,, \quad i=1,2,\ldots,n,$$

és a $\underline{c}$ minden értékére (realizálására), amely a fenti feltételt teljesiti, a GP szuperkonzisztens és a célfüggvény felveszi optimális értékét. Ebben az esetben a GP transzformálható egy ekvivalens konvex programozási feladatra. A legegyszerübb esetben - nulla nehézségi fok - Stark [81] vizsgálta a duál célfüggvény viselkedését. Ekkor a célfüggvény csupán a $\underline{c}$ vektornak a függvénye (1.2 példa). E vizsgálat általánositásával Paul-Eller [73] foglalkoztak. Továbbá stabilis vizsgálatok találhatók [22]-ben.

A geometriai programozás számos alkalmazása található Duffin-Peterson-Zener [33] könyvében, Wilde-Beightler [87], Federovicz [41] Avriel-Rijckaert és Wilde [6] , Beightler [9], Kláfsziki [55] munkáiban. A fizikai müszaki alkalmazások számos példája található Zener [90] könyvében, Schinzinger [82] Unklesaby-Creggton és Staats [85], Ecker-Wiebking [37], Duffin-Zener [34], Mancini-Piziali [61], Dinkel-Kochenberger [24], Morris [63], Neghabat-Stark [64],Templeman [83] munkáiban.

Továbbá a kémiai-müszaki alkalmazások a [33]-, [49]-, [72]-, [18]-ban; a környezetvédelmi alkalmazások a [36]-, [62]-. [86]-, és [27]-ben; a gazdasági modelltervezési alkalmazások a [26]-, [53]-, és [65]-ben; az ügyviteli problémák alkalmazása a [57]-, [7]-, [67]-, [70]-ben; a magfizikai alkalmazások a [12]- és [86]-ban találhatók. Az egyéni döntési problémák - mint a közlekedési [51], megbizhatósági [41], flotta-építészet [42]- alkalmazása is sorra került.

II. A GEOMETRIAI PROGRAMOZÁS MEGOLDÁSMÓDSZEREI

Ebben a részben egy áttekintést adunk a különböző eljárásokról, programcsomagokról, amelyeket a GP feladat megoldására dolgoztak ki. Mivel a GP feladat egyuttal nemlineáris programozási feladat, igy kipróbálhatunk egy nemlineáris programcsomagot is a GP feladat megoldására. Egy összehasonlitást is adunk a leghatékonyabb nemlineáris programozási algoritmusok és a legjobb GP algoritmusok között a futási idő függvényében.

Részletesen tárgyaljuk a két legjobb geometriai programozási algoritmust.

GGP-algoritmus. Az algoritmus megoljda a szignomos PGP feladatot, vagy az RPGP feladatot a közelitő pozinomos GP feladatok megoldásaival. Az algoritmus [2] felhasználja a kiegészitő GP eredményeit, továbbá alkalmazza a süritési technikát. Igy minden feltétel logaritmus transzformációval átalakitható egy lineáris egyenlőtlenséggé. Ezután a Kelley féle metsző siktechnikával a lineáris programozási feladatok egy sorozatát oldja meg, mig nem kapunk egy megengedett megoldást a közelitő pozinomos GP feladatra. Ekkor a szignomos PGP feladatnak egy uj közelitését kapjuk. Folytatjuk az eljárást, a megkapott közelitő megoldássorozat limesze adja az eredeti feladat optimális megoldását. Ennek az eljárásnak az első programcsomagját Dembo [15] dolgozta ki, Dembo és Avriel [16], [2] fejlesztették tovább.

A kiegészitő GP ötletét felhasználva Templeman [83] kidolgozott egy eljrást, ami abban különbözik még a fenti eljárástól, hogy nem a közelitő PGP feladatot oldja meg, hanem annak a duál feladatát a Fletcher - Reeves féle konjungált gradiens irányok módszerével.

Primál süritési algoritmus. Mint emlitettük, minden szignomos PGP átalakitható vele ekvivalens RPGP feladattá. Továbbá a Duffin süritési technikával [29] minden $g_k(t) \geq 1$,

$k=p+1,\dots,q$ feltételt átalakitunk $g_k(\underline{t},\underline{\varepsilon}) \leq 1$, $k=p+1,\dots,q$ egyenlőtlenségé a számtani-mértani egyenlőtlenség alkalzamásával. Ezután ismét alkalmazva a süritési technikát, minden pozinom egy monommá válik. Igy logaritmus-transzformációval kapunk egy lineáris egyenletrendszert. Ennek megoldásával eldönthetjük, hogy szükséges-e kiszámitani uj egyensulyozó tényezőket és ezzel kezdődhet az uj iteráció. Az eljárás [7]-ben található GPKTC névvel.

A fenti két algoritmus a leghatékonyabb algoritmusok közé tartozik, amelyeket a GP feladat megoldása meghatározására dolgoztak ki. Ezeket fogjuk néhány feladaton tesztelni.

Tekintsük át röviden még a primál módszereket, duál módszereket és a szingomos módszereket.

1. PRIMÁL MÓDSZEREK

Egy pozinomos PGP feladat egyuttal egy nemlineáris programozási feladat is, számitástechnikai szempontból is az. Hiszen a pozinomok jól definiáltak és könnyen ki is számithatók. Ha például a

$$g(\underline{t}) = \sum_{i=1}^{r} c_i t_1^{a_{i1}} \cdot t_2^{a_{i2}} \dots t_m^{a_{im}}$$

függvényt kell deriválni $\underline{\bar{t}} > \underline{0}$ pontban, akkor

$$\nabla_j g(\underline{\bar{t}}) = \sum_{i=1}^{r} \frac{a_{ij}}{\bar{t}_j} c_i \bar{t}_1^{a_{i1}} \bar{t}_2^{a_{i2}} \dots \bar{t}_m^{a_{im}} \qquad (j=1,2,\dots,m)$$

Igy összesen r szorzást és r osztást kell végezni. A PGP feladat konvex programozási feladat a $\log t_j = z_j$ (lásd (3)) változó szerint.

Sajnos, a $\underline{t} > \underline{0}$ feltétel által adott tartomány nem kompakt, igy nem is minimum keresésről van szó, hanem infimum keresésről. Amikor egy változó tart nullához vagy néhány értéke negativvá válik, akkor a pozonomoknak egy vagy több tényezője $(u_i(\underline{t}))$ nemdefiniálhatóvá válik. Emiatt csak olyan nemlineáris programozási algoritmus alkalmazható a PGP feladat megoldására, ami biztositja, hogy a változók pozitivak maradjanak. Egyszerü mód lenne, az, hogy $t_j \geq \varepsilon$, ahol $\varepsilon > 0 = (j\ 1, 2, \ldots, m)$ feltétellel helyettesitjük a pozitivitási feltételeket. Ebben az esetben nehéz helyzetbe kerülünk, ha a GP degenerált, azaz néhány tényezőnek nullává kell válnia az optimális megoldásnál.

Tehát legcélszerübb az lenne, hogy ha sikerülne kidolgozni valami gradiens-irányokon alapuló módszert, ami megoldja a fenti nehéz pozitivitási követelést, hiszen a gradiens számitása nagyon könnyü. Sőt egyszerü transzformáció után egy ekvivalens konvex programozási feladatot kapunk.

A PGP feladat szintén megoldható a dekompoziciós technikával. Mint (4)-ben látható, a szeparálható PGP $m+n$ változóju ($x=Az$), ha az eredeti PGP m változóju volt. Azonban e feltételek speciális strukturája miatt lehetne külön algoritmust kidolgozni. Eddig még csak ketten próbálkoztak ezzel: Reklaitis és Wilde [75] kidolgozták a Wilde-Beghtler [87] módszerét; Gochet és Smeers [45] kifejlesztették a metszősik módszert és megmutatták, hogy az általuk kidolgozott metszősik-eljárás konvexprogramozási feladatok esetében jobb mint a Kelley [54]-féle módszer. [23] -ban Dinkel, Elliott, Kochenberger négy különböző metszősik-módszert vizsgáltak a PGP feladatra. Ecker és Zoracki [38] kidolgoztak egy módszert arra, hogy a PGP feladatot hogyan lehet közeliteni egy lineáris programozási feladattal egy előre adott pontosság szerint. Erre felhasználták azt, hogy minden pozinomos PGP feladat átalakitható olyan pozinomos PGP feladattá, melyben minde pozinomnak csak két tényezője van $(U_1(\underline{t}) + U_2(\underline{t}))$.

Az uj algoritmusokat a süritési technika alkalmazásával fejlesztette tovább Passy [68], Pascual és Ben-Israel [66]. A [2]-ben Avriel, Dembo és Passy azokat az algoritmusokat ismertetik a pozinomos és szignomos GP feladatokra, melyekben a pozinomokat süritési technikával monomokra átalakitották.

2. DUÁL MÓDSZEREK

Tekintsük az alábbi un. szeparálható DGP feladatot (SDGP):

$$\max V(\underline{\delta},\underline{\lambda}) = \sum_{i=1}^{n} \delta_i . \log\left(\frac{c_i}{\delta_i}\right) + \sum_{k=1}^{p} \lambda_k \log \lambda_k$$

feltéve, hogy

$$\sum_{[0]} \delta_i = 1$$

(23)

$$A^T \underline{\delta} = \underline{0}$$

$$B\underline{\delta} - \underline{\lambda} = \underline{0}$$

$$\underline{\delta} \geqq \underline{0}$$

ahol $A-(mxn)$, $B-(pxn)$ mátrix.

Ez az SDGP nyilvánvalóan ekvivalens DGP feladattal. Csakhogy a SDGP feladatnak $n+p$ változója van, mig a DGP feladatnak csak n változója volt, másrészt SDGP konvex programozási feladat.

A DGP, az SDGP és a redukált DGP (lásd (9)) feladatoknak van néhány tulajdonságuk, melyek meggátolják a nemlineáris programozási algoritmusok alkalmazását. A legszembetünőbb a nem-deriválhatóság a nullánál. A nem differenciálhatóság feloldására a következő lehetőségek ismertek (Dembo [20]).

a) Korlátozzuk a $\underline{\delta}$ változó vektort ugy, hogy $\delta_i \geqq \varepsilon > 0$, $i=1,\dots,n$ legyen, ahol ε tetszőlegesen kicsi pozitiv szám. Ezzel elhárítható a nem-differenciálhatóság. De a számitás során sok algoritmus "cikk-cakkozik" a $\delta_i^*=\varepsilon$, $i\in[k]$ (valamely k-ra) értékénél. Célszerü lenne tehát minden $\delta_i^*=\varepsilon$ helyett $\delta_i^*=0$ választása. Ekkor viszont az uj $\underline{\delta}^*$ nem megengedett megoldás, hiszen nem teljesiti a normalitási és/vagy ortogonalitási feltételt. Ecker [39] módositotta Beck-Ecker [8] algoritmusát ugy, hogy ezt a nehézséget el tudjuk háritani. Inditsuk az algoritmust egy $\bar{\underline{\delta}}>0$ megengedett megoldással. Ha ilyen $\bar{\underline{\delta}}$ nem létezik, akkor a feladat nem konzisztens, igy az I. részben emlitettek szerint adott $\bar{\underline{\delta}}$ megengedett megoldás esetén a (9) feladat feltételei igy irhatók

(24) $\underline{y} + B\underline{z} = \underline{b}$

ahol $\underline{y}$ és $\underline{z}$ a $\underline{\delta}$ vektornak egy felbontása, amelyet ugy is csinálhatunk, hogy $\underline{y}$ tartalmazza a $\underline{\delta}$ legnagyobb komponenseit. $\underline{y}$ a bázis változó és $\underline{z}$ a nem bázis változó. Ha a z_i változót változtatjuk, akkor az y_j meghatározható ugy, hogy a (24) egyenlet teljesüljön. A z_i változása a következővel jellemezhető:

(25) $r_i(\underline{\bar{\delta}}) = G\, z_i(\underline{\bar{\delta}}) - \Sigma G y_j(\underline{\bar{\delta}})\, b_{ji}$

ahol b_{ji} a B mátrix j-edik során és i-edik oszlopján fekszik, és

(26) $G\delta_i(\underline{\bar{\delta}}) = \left.\dfrac{\partial V}{\partial \delta_i}\right|_{\underline{\bar{\delta}}} = log\left(\dfrac{c_i \lambda_k}{\delta_i}\right) \quad i \in [k], \quad k=0,1,\dots,p.$

Ha $r_i > 0$ akkor z_i változót lehet még növelni, miközben a $V(\underline{\delta})$ célfüggvény növekszik, és ha $r_i < 0$, akkor z_i változót csökkenteni kell (nullára), hogy eközben a $V(\underline{\delta})$ célfüggvény növekedjen. A nem bázis változó megválasztása meghatározza a célfüggvény növekedési irányát. Igy az iránymenti maximalizálással meghatározható a legjobb $\underline{\delta}^*$ és ez legyen a következő lépés kezdő megoldása. [88]-ban Zangwill ad pontos szabályokat a nem bázis vektor megválasztására ugy, hogy az eljárás konvergáljon.

A Beck-Ecker féle algoritmus módositása az, hogy a duál változók bizonyos csoportjára (bizonyos $[k]$-hoz tartozó index) megengedhetjük, hogy nullához tartson ugy, hogy a gradiens vektor jól definiálható maradjon. Ez a módositás azon az észrevételene alapszik, hogy $G\delta_i$ $(i \in [k])$ marad változatlanul, ha λ_k/δ_i $(i \in [k])$ hányados (lásd (26)) nem változik. Igy ha létezik egy olyan $[k]$ csoportja a δ^* változónak, hogy $r_i(\underline{\delta}^*) < 0$, $(i \in [k])$, akkor minden ilyen $\delta_i^* (i \in [k])$ változót csökkentünk ugy, hogy λ_k^*/δ_i^* maradjon változatlanul $(i \in [k])$. Ez a döntési szabály meghatározza azt az irányt, mely mentén a célfüggvény növekszik, miközben a csoport elemeit nullára csökkenthetjük. De néha valamely lépésnél ugyanaz az indexcsoport is tartalmaz olyan indexet, melynek megfelelő redukált gradiens-komponens (r_i) is pozitiv. Igy e változó növekvése (nullától) a célfüggvény növekvését eredményezi. Emiatt az algoritmus nem müködne, hiszen a gradiens komponense nem lenne definiálható. E nehézség leküzdésére Beck-Ecker [8] mutatja azt, hogy egy ilyen indexcsoport esetén lehet konstruálni egy $\underline{\delta}^o$ megengedett megoldást, melynél a célfüggvény értéke ugyanaz mint a szóban forgó megoldásnál, de ebben az indexcsoportban mindegyik redukált gradiens komponense (25) vagy pozitiv, vagy negativ, vagy nulla. Ha mindegyik komponens pozitiv, akkor az indexcsoport elemei növelhetők ugy, hogy a λ_k^o/δ_i^o hányados ne változzék és ezzel növelhetjük a célfüggvényt. Ha mindegyik komponens nem pozitiv, akkor az legyen nulla.

b) Egy másik lehetőség az, hogy ha valamely $\delta_i \leq \varepsilon$, akkor a $\delta_i log(c_i/\delta_i)$ kifejezést egy kvadratikus alakkal közelitjük, azaz

$$\delta_i \; log(c_i/\delta_i) \simeq \alpha \, \delta_i^2 + \beta\delta_i \qquad 0 \leq \delta_i \leq \varepsilon$$

ahol $\alpha = -{}^1/_\varepsilon$; $\beta = log({}^{c_i}/_\varepsilon) + 1$.

Ennek a közelitésnek előnye az, hogy a $V(\delta,\lambda)$ célfüggvény szintén folytonosan differenciálható miközben δ_i tart nullához, mivel α és β-t már ugy választottuk, hogy $\delta_i \; log(c_i/\delta_i)$ deriváltjának az értéke és $\alpha\delta_i^2 + \beta\delta_i$ deriváltjának az értéke egyenlő egymással $\delta_i = \varepsilon$-nál. Az ε érték megválasztása elég egyszerü. Hiszen a mi filozófiánk az, hogy ha az algoritmus valamely lépésénél $\delta_i = 0$, $i \in [k]$, akkor annak a deriváltja ne legyen negativ, mert akkor a következő lépés már rosszabb lesz. Mivel $\beta = log(c_i/\varepsilon) + 1$, ez pozitiv marad, ha $\varepsilon < c_i e (log e = 1)$. Igy például ugy választható

$$\varepsilon = min\,\{0.9.c_i e,\; 10^{-5}\}.$$

Az első a DGP feladatra kidolgozott algoritmus Frank-tól [43] származott. Jól ismert még Blau-Wilde [11] algoritmusa, amely a DGP Kuhn-Tucker féle szükséges és elégséges optimalitási feltételeti oldja meg. Ezzel rokon Rijckaert-Martens [78] algoritmusa.

Az algoritmusok egy osztálya a redukált DGP (8) feladatok megoldására alkalmas, Bradley [13], Templeman-Wilson [84]. Dinkel, Kochenberger és McCarl [25] egy módositott Newton-Raphson- féle algoritmust, Jefferson [52] egy módositott Newton féle algoritmust fejlesztett ki.

3. A DUÁL ÉS PRIMÁL MÓDSZER MEGVÁLASZTÁSA

Gyakran felvetődik a kérdés, hogy a geometriai programozási feladatokat duál módszerrel vagy primál módszerrel oldjuk meg. Vagyis mikor alkalmazzuk a duál módszert és mikor a primál módszert. A legegyszerübb döntés, akkor, amikor például a DGP nagyon egyszerü (nulla nehézségi fok), vagy a PGP egyszerübb mint DGP.

A lineáris programozásban ilyen döntés könnyen meghozható, de a geometriai programozásban ez a kérdés sokkal bonyolultabb. Kézenfekvő volna a számitási eredmények alapján dönteni, mint Rijckaert-Martens [78] és Dembo [20] csinálták. Azonban - mint lineáris programozásban - van olyan probléma, melyre mind a duál módszer mind a primál módszer alkalmazható. Például tekintsünk egy feltétel nélküli GP feladatot, annak (3)-transzformációja után következő konvex szeparálható feladathoz jutunk

Primál feladat: $$\min_{\underline{y},\underline{x}} \log \sum_{i=1}^{n} c_i \exp(y_i)$$

feltéve, hogy

$$\underline{y} - A\,\underline{x} = \underline{0}$$

Duál feladat: $\max \sum_{i=1}^{n} \delta_i \log (c_i/\delta_i)$

feltéve, hogy

$$\sum_{i \in [0]} \delta_i = 1$$

$$A \underline{\delta} = \underline{0}$$

$$\underline{\delta} \geq \underline{0}$$

Itt a két célfüggvény szerkezete majdnem ugyanaz. Tehát a különbség a felételekben rejlik, hiszen a primál feltételek között nincs egyenlőtlenség, és a célfüggvény mindenütt differenciálható, mig a duál célfüggvény nullánál nem értelmezhető.

A feltételes GP feladatok esetén a probléma ugy tünik forditva van. A PGP tartalmaz sok nemlineáris feltételt, mig a DGP nem (kivéve $\underline{\delta} \geq \underline{0}$ feltételt). Dembo [20]-ban javasolja, amikor a nehézségi fok valamely feladatnál nagyon nagy, akkor a duál módszert használjuk. Ennek kettős az indoklása, egyrészt a sok nemlineáris feltételt nehezebb kezelni mint a lineáris feltételeket; másrészt mint ezt sokan ki is próbálták, a DGP feladatot meg lehet oldani Newton féle algoritmusokkal, ami szerint mindegyik lépésnél megoldunk egy négyzetes lineáris egyenletrendszert, amelynek a nagyságát a $(n-(m+1))$ nehézségi fok vagy a primál változók száma (m) határozza meg attól függően, hogy melyik szám a legkisebb. Sajnos ez a javaslat nem mindig alkalmazható, amiért ezt 4-ben indokolni fogjuk.

4. A SZIGNOMOS MÓDSZEREK.

Mint 2.1-ben emlitettük, egy szignomos GP mindig visszavezethető egy vele ekvivalens RGP (forditott geometriai programozás) feladatra. Ekkor a süritési technikát alkalmazva kaphatunk egy közelitő pozinomos GP feladatot. Tehát az első gondolat az, hogy oldjuk meg a szignomos GP feladatot közelitő pozinomos feladatokon keresztül.

Az Avriel-Williams féle eljárást alkalmazva Avriel-Dembo-Passy [2] kidolgoztak egy nagyon hatékony algoritmust. [31]-ben Duffin-Peterson megmutatták, hogy hogyan kell használni az aritmetikai-harmonikai egyenlőtlenséget az Avrie féle eljárásra. Dinkel, Kochenberger és McCarl [25] Avriel [4] módszerére az aritmetikai-geometriai egyenlőtlenséget alkalmazták.

Az általános nemlineáris programozási algoritmusokat is sokan próbálták felhasználni a pozinomos és szignomos GP feladatok megoldására. [20]-ban Dembo használta az általános redukált gradiens irányok módszerét (GRG) [59], [1] ; a büntetéses függvény módszerét (SUMT) és a kiterjesztett gradiens irányok módszerét (EGPM) [53] . A nemlináris algoritmusok és a GP algoritmusok futási idejének hányadosát 8 feladaton vizsgálták. A teszt problémák [17]-ben találhatók. Az alábbi táblázat tartalmazza a főbb adatokat és a normalizált futási időre vonatkozó eredményeket.

1. Tábla. A NLP algoritmusok és a GP algoritmusok futási idő hányadosa:

probléma	tipus	változó	nemlin. felt.	$\frac{\text{legjobb NLP}}{\text{legjobb GPKTC}}$	$\frac{\text{legjobb NLP}}{\text{legjobb GGP}}$	$\frac{\text{legjobb NLP}}{\text{legjobb GP}}$
1A	pozinomos	12	3	79.6	17.0	79.6
1B	"	12	3	1.0	0.2	1.0
2	szignomos	5	6	0.2	5.0	5.0
3	"	7	14	1.3	1.4	1.4
4A	"	8	4	1.7	0.1	1.7
4B	"	8	4	1.8	0.2	1.8
4C	pozinomos	9	5	1.7	1.9	1.9
5	szignomos	8	6	1.5	0.4	1.5
6	"	12	13	1.6	1.8	1.8
7	"	16	19	1.4	4.0	4.0
8A	pozinomos	7	4	0.8	1.8	1.8
8B	"	7	4	1.0	2.0	2.0
8C	"	7	4	1.8	3.3	3.3

Itt érdekes az, hogy a legjobb GP algoritmus mindig gyorsabb, mint a legjobb NLP algoritmus. De figyelembe kell venni azt, hogy itt csak nyolc feladat van. Az általánosabb esetben, azaz többfeladat, több változó esetén hogy reagál ez az összehasonlitás, látni fogjuk e szakasz végén.

A számitási eredmények elemzése megtalálható még Rijckaert [77]-ben és Schinzinger [82]-ben.

Lasdon, Ratner és Jain [60] saját általánositott redukált gradiens módszerüket (GRG) használták a [17] és [77]-ben megadott teszt problémák megoldására. Kiderült, hogy a GRG programcsomag ugyanolyan hatékony mint a GP programcsomag, habár ennek a GRG programcsomagnak nem sikerült megoldania két [17]-beli problémát. Fiacco és Ghaenni kidolgoztak néhány sikeres eljárást a SUMT módszer alkalmazásával a pozinomos GP feladatok egy speciális csoportjára.

Van néhány eljárás, ami a szignomos probléma globális minimumát keresi meg. [40]-ben Falk javasol egy korlátozás-szétválasztás elvén alapuló algoritmust, ami a szignomos feladat globális minimumához konvergál. [46]-ban Gochet és Smeers vizsgálnak egy korlátozás-szétválasztási algoritmust a foridott (Reversed GP) feladatra, ami a fordiott $(g_k(t) \geq 1)$ feltételeket közelitő feltételekkel helyettesiti. Passy vizsgált egy hasonló eljárást [69]-ben. [44]-ben Fattler, Reklaitis, Sin, Root és Ragsdell egy ujabb tanulmányt ismertetnek a geometriai programozási technikák elemzésére. A tanulmányban 10 programcsomagot használtak fel (4 nemlienáris programcsomag és 6 geometriai programozási programcsomag) öt különböző feladattipusra: (1), (2), (3), (4), és (9)-re (lásd I. rész). A nemlineáris programcsomagok között legfontosabb a GRG algoritmus (aminek neve továbbiakban OPT, hiszen nem a Lasdon [60] algoritmusát, hanem az Abadie [1] által fejlesztett algoritmust használták fel). Az eredeti algoritmus [47]-ben található. A tanulmány 42 különböző teszt problémákon alapul, a változók, a feltételek száma a következőképpen változott

$2 \leq$ *primál változók* ≤ 30
$8 \leq$ *primál tényezők* ≤ 197
$1 \leq$ *feltételek* ≤ 73
$1 \leq$ *nehézségi fok* ≤ 166

A szerzők a következő eredményekre jutottak:

Általában a duál módszerek akkor lesznek annyira jók mint a (3) tipusu konvex primál feladatokra kidolgozott módszerek, ha a nehézségi fok kicsi *(<15)* és sok feltétel teljesül egyenlőséggel az optimális megoldásnál. A GGP és GPKTC programcsomagok versenyképessége nem éri el az OPT programcsomagét. A jövőben az OPT-féle algoritmus nagy valószinüséggel a leghatékonyabb lehet ha a (3) tipusu konvex primál feladatnak uj specializálodására is. A (4) tipusu transzformált primál feladatokra épült programcsomagok nem olyan taékonyak mint a (3) konvex primál feladatokra kidolgozott programcsomagok, bár az előbbiek szeparálható feladatok. A pozinomos GP nehézsége - a computer időre mérve - a teszt változói számának exponeciális függvénye és egyenesen arányos a feltételekben levő összes tényezők számával.

Dembo [20] azt javasolja, hogy néhány olyan elméleti kérdést kellene megvizsgálni, aminek alapján a primál változók merev *(>0)* feltételeit könnyebben lehet kezelni. Ezzel hatékonyabbá tehetők a primál módszerek.

IRODALOMJEGYZÉK

[1] Abadie,J., and Guigou, J., Numerical Experiments with the GRG Method, Integer and nonlinear Programming, Edited by J.Abadie North Holland, Publishing Company, Amsterdam, Holland, 1970.

[2] Avriel, M., and Bunting, M., Reducing posynomial programs, SIAM J. Appl. Math., 27/1974/ pp. 629-640.

[3] Avriel, M., Dembo, R., and Passy, U., Solution of generalized geometric programs, Internat. J. Numer. Methods Engrg., 9/1975/ pp. 149-169.

[4] Avriel, M. and Williams, A.C., Complementary geometric programming, SIAM J. Appl. Math., 19/1970/, pp. 125-141.

[5] Avriel, M., and Wilde, D.J., Stochastic geometric programming, Proceeding of the International Mathematical Programming Sympsoium, Edited by H.W. Kuhn, Priceton University Press, Princeton, New Jersey, pp. 73-91, 1970.

[6] Avriel, M., Rijckaert, M.J., and Wilde, D.J., Editors, Optimization and Design, Prentice Hall, Englewood Cliffs, New Jersey, 1973.

[7] Balachandran, V., and Gensch, D., Solving the marketing -mix problem using geometric programming, Management Sci., 21/1974/ pp. 160-171.

[8] Beck, P.A., and Ecker, J.C., A modified concave Simplex algorithm for geometric programming. J.O.T.A., 15/1975/ pp. 189-202.

[9] Beightler, C.S., and Phillips, D.T., Applied geometric programming, John Wiley and Sons, New York, New York 1976.

[10] Blau, G., Wilde, D.J., Second order characterization of generalized posynomial programs, Paper Pressented at the Princeton International Symposium on Math Prog. 1967.

[11] ________, A lagragian algorithm for equality constrained generalized posynomial optimizatio, AICh.E.J., 17/1971/ pp. 235-240.

[12] Bouchey, G.D., Beightler, C.S., and Koen, B.V., Optimization of nuclear systems by geometric programming, Nuclear Science and Engineering, 44 /1971/, pp. 267-272.

[13] Bradley, J., The development of posynomial programming algorithm with applications. Ph.D. thesis, Dept. of Computer Science Dublin Univ., Dublin, Ireland, 1975.

[14] Dawkings, G.S., Mcinnis, B.C., and Moonat, S.K., Solution to geometric programming problems by transformation to convex programming problems, Intern. J. of Solid Structures, Vol. 10, pp. 135-136, 1974.

[15] Dembo, R.S., GGP- A Computer program for soling generalized geometric programming problems, Users, Manual, Rep. 72159, Dept. of chemical Engineering, Technion, Israel Institute of Technology, Israel, March 1972.

[16] ______, The solution of complementary geometric programming problems, M.Sc. Thesis, Technion, Israel Institute of Technology Haifa, Israel, 1972.

[17] ______, A set of geometric programming test problems and their solutions, Math, Programming, 10/1976/, pp. 1972-214.

[18] _______, Optimal design of a membrane separation process, Applied geometric programming, C. Beightler and D. Phillips, eds., John Wiley, New York, 1976.

[19] _______, Some real-world applications of GP, Applied geometric programming, Edited by C.S. Beightler and D.T. Phillips, Prentice hall, Englewood Cliffs, New Jersey, 1976.

[20] _______, Current state of the art of algorithms and computer software for GP, J.O.T.A., 26/1978/, pp. 149-184.

[21] ______, Second order algorithms for the geometric programming dual, Part I: Analysis, Tech. Rep. 22, Yale School of Organization and Management, New Haren, CT, 1978.

[22] ______, Sensitivity analysis in geometric programming, J.O.T.A., Vol. 37, No.I, /1978/, pp. 1-21.

[23] Dinkel, J.J., Elliott, W.H., and Kochenberger, G.A., Computational aspects of cutting plane algorithms for geometric programming problems. Math. Programming, 13/1977/, pp. 200-220.

[24] Dinkel, J.J., and Kochenberger, G.A., A cofferdam design optimization, Ibid., 6/1974/, pp. 114-116.

[25] Dinkel, J.J., Kochenberger, G.A., and McCarl, B., An approach to the numerical solution of geometric programming, Ibid., 7/1974/, pp. 181-190.

[26] Dinkel, J.J., Kochenberger, G.A., and Seppala, Y., On the solution of regional planning models via geometric programming, Environmental Planning, 5/1973/, pp. 397-408.

[27] Dinkel, J.J., Kochenberger, G.A., and Wong, S.N., Entropy maximization and geometric programming, Environmental and Planning, 9/1977/, ppl. 419-427.

[28] Duffin, R., and Zener, C., Optimization of engineering proglems, Westinghouse Engineer, Vol.24, pp. 154-164, 1961.

[29] Duffin, R., Linearizing geometric programming, SIAM Review, Vol. 12.pp. 211-227, 1970.

[30] Duffin, R., and Peterson, E.L., Duality theory for geometric programming, SIAM J. Appl. Math., 14/1966/, pp. 1307-1349.

[31] _____, Reversed geometric programming treated by harmonic means, Indiana Univ. Math. J., 22/1972/, pp. 531--550.

[32] ____, Geometric programming with signomials, J.O.T.A. 11/1973/ pp. 3-35.

[33] Duffin, R., Peterson, E.L., and Zener, C., Geometric programming, John Wiley, New York, 1967.

[35] Ecker, J.G., Decomposition in separable GP, J.O.T.A., Vol. 9. N.3. 1972.

[34] Duffin, R., and Zener, C., Geometric programming and the Darwin-Fowler method in Statistical mechanics, J. Physical Chemistry, 74/1970/, pp. 2419-2423.

[36] _____, A geometric programming model for optimal allocation of stream-dissolved oxygen, Management Sci., 21/1975/, pp. 658-668.

[37] Ecker, J.G., and Wiebking, R.D., Optimal design of a dry-type natural-draft cooling tower by geometric programming, J.O.T.A., 26/1978/, pp. 305-324.

[38] Ecker, J.G., and Zorack, M.J., An easy primal method for geometric programming, Management Sci., 23/1976/, pp. 71-77.

[39] Ecker, J.G., Geometric programming: methods, computation and application, SIAM review, 22/1980/, pp. 338-362.

[40] Falk, J.E., Global solutions of signomial problems, Tech. Rep. T-274, George Washington Univ., Program in Logistcs, Washington, DC. 1973.

[41] Federowich, A.J., and Mazumdar, M., Use of geometric programming to maximize reliability achieved by redundancy, Operations Res. 16, 1968, pp. 948-954.

[42] Folker, J.S., Ship optimization and design, Optimization and design M. Avriel, M.J. Rijchaert and D.J. Wilde, eds., Prentice-Hall, Englewood Cliffs, NJ, 1973, pp. 221-226.

[43] Frank, C.J., An algorithm for geometric programming, Recent Advances in Optimization Techniques, A. Lavi and T. Vogl, eds. John Wiley, New York, 1966, pp. 145-162.

[44] Fattler, J.E., Reklaitis, G.V., Sin, Y.T., Root, R.R., and Ragsdell, K.M., On the computational utility of posynomial GP solution methods, Math. Programming 22/1982/, pp. 163. 201.

[45] Gochet, W., Loute, E., and Solow, D., Comparative computer results of three algorithms for solving prototype geometric programming problems, Cahiers Centre Étude Recherce Opér., 16/1974/, 461-486.

[46] Gochet, W., and Smeer, Y., A branch andboudn method for reversed geometric programming Core Discussion Paper 7511, Catholic Univ of Louvain, 1975, Operations Res., /1980/.

[47] Gabriele, G.A., and Ragsdell, K.N., OPT: a nonlinear programming code in FORTRAN IV, The Modern Design Serie, Vol. I., Purdue Research Foundation /West Lafayette, IN, 1976/.

[48] Heymann, M., and Avriel, M., On a decomposition for a special class of GP problems, J.O.T.A., 6/1969/, pp. 392.

[49] Hellincky, L.J., and Rijchaert, M.J., Optimal capacities of production facilities: An application of GP, Canad. J. Chemical Engrg. 50/1972/, pp. 148-150.

[50] Hall, M.A., Hydraulic network analysis using generalized geometric programming, Networks, Vol.6, pp. 105-131, 1976.

[51] Jefferson, T.R., Geometric programming with an application to transportation planning, Ph.D.Thesis, Noerthwester Univ., Eranston IL, 1972.

[52] Jefferson, T.R., Manual for the geometric programming code GPROG /CDC/ VERSION2, Report 1974/OR/2, Mechanical and industrial Engineering, Univ. of New South Wales, Australie, 1974.

[53] Kádas, S., An application of geometric programming to regional economies, Paper of the Regional Science Association, 34/1975/, pp. 95-106.

[54] Kelley, J.E., The cutting-plane method for solving convex programs, SIAM J. Appl. Math., 8/1960/, pp. 703-712.

[55] Klafsziky, E., Kandidátusi értekezés, Budapest, 1973.

[56] Kochenberger, G.A., Woolsey, R.E.D., and McCarle, B.A., On the solution of GP via separable programming, Operations Res. Quart. Vol. 24/1973/, pp. 285-296.

[57] Kochenberger, G.A., Inventory model optimization by geometric programming, Decision Science, 2/1971/, pp. 193-205.

[58] Kreuser, J.L., and Rosen, J.B., GPM/GPMNLC extended gradient projection method nonlinear programming subroutines, University of Wisconsin, Academic Computer Center, 1971.

[59] Lasdon, L.S., Warren, A.D., Ratner, M.W., and Jain, A., GRG system documentation, Cleveland State Univ., Technical Memorandum No. CIS-75-01, 1975.

[60] Lasdon, L.S., Ratner, M.W., and Jain, A., Solving geometric programming using GRG: results anc comparisions, J.O.T.A., 26/1978/, pp. 253-265.

[61] Mancinic, L.J., and Piziali, R.A., Optimal design of helical springs, By GP, J. Engreg. Optimization, 2 /1976/, pp. 83-95.

[62] McNamara, J.R., An optimization model for orgion water quality management. Water Resources Research, 12/1976/, pp. 658-668.

[63] Morris, A.J., A primal-dual method for minimum weight design of statically determinate structures with several system of load, Internat. J. Mech. Sci., 16/1974/, pp. 801-807.

[64] Neghabat, F., and Stark, R.M., A cofferdam design optimization, Math. Programming, 2 /1973/, pp. 263-276.

[65] Nijkamp, P., and Palinck, J.H.P., Inerregional model of environmental papers of the Regional Science Associatio, 31/1971/, 51-71.

[66] Pascual, L.D., and Ben-Izrael, A., Constraided maximization of posynomials by GP, J.O.T.A., 5/1970/, pp. 73-80.

[67] Passy, U., Modular design, an appliaction of structured GP, Operadtions Res., 18/1970/, pp. 441-453.

[68] ____, Generalized weighted mean programming, SIAM J., Appl. Math. 20/1971,/ pp. 763-778.

[69] ____, Signomial GP: Determining the globalminimum. Mathematical Programs for Activity Analysis, P.Van Moeseke, ed., North-Holland Amsterdam, 1974.

[70] _____, Nonlinear assignment problems tread by GP, Operations Res. 19/1971/, pp. 1675-1690.

[71] Passy, U., and Wilde, D.J., Generalized psynomial optimizations, SIAM J. Appl. Math., 15/1967/, pp. 1344-1356.

[72] _____, A GP algorithm for solving chemical equilibrium problems, SIAM review, 16/1968/, pp. 363.

[73] Paul, M.E., and Startk, R.M., On the distribution of the optimal value for a class of stochastic, GP, Nava Res. Logist. Quart. 27/1980/, pp. 549-571.

[74] Peterson, E.L., Geometric programming- a Survey, SIAM Review 18 1976, pp. 1-51.

[75] Reklaitis, G.V., and Wilde, D.J., GP via a primal auxiliary problem, AIIE Trans., 6/1974/, pp. 308-317.

[76] Rijckaert, M.J., On the application of decomposition techniques in GP, Decomposition of Large-scale Problems, D.M. Himmuelblau, ed., 1973.

[77] _____, Comparisons of generalized geometric programming algorithms Ibid., 26/1978/, pp. 205-242.

[78] Rijckaert, M.J., and Martens, X.M., A condensation method for generalized GP, Math. Programming, II/1976/ pp. 89-93.

[79] Rockafeller, R.T., Convex programming problems wiht linearly constrained duals, Nonlinear Programming, Edited by J.B.Rosen, et at, Acad., Press, New York, New York, 1970.

[80] Scott, C.H., Jefferson, T.R., and Kerdvonbundit, P., Geometric programming duality applied to the resource allocation problem, INFOR, 17/1978/, pp. 133-137.

[81] Stark, R.M., On zero-degree stochastic geometric programs, J.O.T.A. 23/1977/, pp. 167-187.

[82] Schinzinger, R., Optimization in electro-magnetic system design, Recent Advances in Opt. Tech, A. lavi and. T. Vogl, eds., John Wiley New York, 1966.

[83] Templeman, A.B., Structural design for minimum cost using the method of GP. Proceedings of the Institute of Civil Engineering 46/1970/, pp. 459-470.

[84] Templeman, A.B., Wilson, A.J., and Winterbottom, S.K., SIGNOT- a computer code for solving signomial geometric programming problems, Res. Rep., Dept. of Civil Engineering, Univ. of Liverpool, Liverpool, England 1972.

[85] Unklesbay, K., Staats, G.E., and Greghton, D.L., Optimal design of joirnal bearing, Internat. J. Engrg. Sci., 11/1973/, 973-983.

[86] Wyman, F.P., The use of geometric programming in the design of an Algerian water conveyance system, Interfaces, 8/1978/, pp.1-6.

[87] Wilde, D.J., and Beightler, C.S., Foundations of optimization, Prentica-Hall. Englewood Cliffs, NJ, 1967.

[88] Zangwill, W.I., The convex simplex method, Management Sci. 14. 1967. pp. 221-238.

[89] Zener, C., A mathematical aid in optimizing engineering design, Proc. Nat. Acad. Sci. U.S.A., 47/1961/, pp. 537-539.

[90] _____, Engineering design by geometric programming, John Wiley and Sons, New York, New York, 1971.