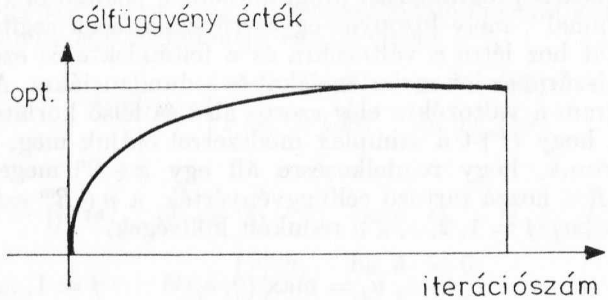
Somos Endre

# Felső korlát az optimális megoldásra a szimplex módszernél és a Dantzig-Wolfe dekompozíciós algoritmusnál

Nagyméretű lineáris programozási feladatok szimplex módszerrel történő megoldásakor gyakran a következő jelenség figyelhető meg: bár az aktuális megoldáshoz tartozó célfüggvény érték már az optimum közelébe került, még igen sok iterációt kell végrehajtani addig, míg az algoritmus eléri az optimumot (1. ábra).

A gyakorlatban ez ellen az egyik védekezési mód az, hogy nem várják meg, míg az algoritmus optimális megoldást ad, hanem megelégszenek egy olyan




1. *ábra*

megengedett megoldással, amelyhez tartozó célfüggvény érték várhatóan már az optimum közelében van. A szimplex módszer azonban közvetlenül nem ad arra becslést, hogy ilyen módon mekkora hibát követünk el. Így általában azt is nehéz megállapitani, mikor kaptunk az optimumhoz elég közeli megoldást, vagyis mikor érdemes megállni. Ezért lényeges, hogy az optimumértékre egy aránylag jó felső becslés álljon rendelkezésre.

A továbbiakban először a [3]-ban leírt módszert ismertetjük, mely az általános lineáris programozási feladatra ad egy egyszerű, és egy ennél jobb, módosított felső korlátot, majd ezt az elvet alkalmazva a Dantzig—Wolfe dekompozíciós algoritmus egy olyan változatát állítjuk elő, mely némi többletmunka árán az eddig ismertnél jobb felső korlátot ad.

## Egy egyszerű felső korlát

Tekintsük a következő általános lineáris programozási feladatot:

$$(\mathrm{P})\ \max cx$$
$$Ax = b,$$
$$x \geqslant 0,$$

ahol $c, x \in R^n$, $b \in R^m$ és $A \in R^{m \times n}$. Legyen ennek optimumértéke $v(P)$. Egészítsük ki a feladatot az

$$s \leqslant x \leqslant t$$

feltételekkel, ahol $t$ komponensei nem feltétlenül végesek. A kiegészített feladatot nevezzük $P(s, t)$-nek, és tegyük fel, hogy

$$v(P) = v\big(P(s, t)\big),$$

vagyis $s$ és $t$ olyan korlátok, amelyek nem befolyásolják a feladat optimumértékét.

Megjegyzendő, hogy ilyen korlátok nem mindig állnak rendelkezésre, viszont a feltételekből és az esetleg meglevő korlátokból előállíthatók [1]. (A LIPROS lineáris programozási programcsomag például el van látva olyan ,,előtétprogrammal'', mely bizonyos egyszerű számítások segítségével alsó és felső korlátokat hoz létre a változókra és a feltételekre és ezek segítségével megpróbálja kiszűrni az inkonzisztenciákat és redundanciákat. A tapasztalatok szerint a program a változókra elég szoros alsó és felső korlátokat ad.)

Tegyük fel, hogy $(P)$-t a szimplex módszerrel oldjuk meg, és a megoldás során ott tartunk, hogy rendelkezésre áll egy $\bar{x} \in R^n$ megengedett bázismegoldás, $\bar{z} \in R$ a hozzá tartozó célfüggvényérték, a $\bar{p} \in R^m$ szimplex szorzók és a $\bar{c}_j = c_j - \bar{p}a_j$, $j = 1, 2, \ldots, n$ redukált költségek.

Legyen

$$\bar{u}_j = \max\{0, \bar{c}_j\} \qquad j = 1, \ldots, n$$
$$-\bar{v}_j = \min\{0, \bar{c}_j\} \qquad j = 1, \ldots, n$$

Ekkor

$$\bar{z} \leqslant v(P) = v\big(P(s, t)\big) \leqslant \bar{p}b + \bar{u}t - \bar{v}s$$

mivel $(\bar{p}, \bar{u}, \bar{v})$ a $P(s, t)$ feladat duálisának egy megengedett megoldása.

## Módosított felső korlát

Az optimális célfüggvény érték egyszerű felső korlátja úgy keletkezett, hogy a $P(s, t)$ feladat duálisának egy megengedett megoldását határoztuk meg. Módosított felső korlátot úgy fogunk kapni, hogy a duális feladat megengedett tartományát úgy szűkítjük le, hogy egy könnyen megoldható feladatot kapjunk.

A $P(s, t)$ feladat duálja a következő:

$$(D)\ \min pb + ut - vs$$
$$pA + u - v \geqslant c$$
$$u, v \geqslant 0$$

A $(D)$ feladatban azzal a megkötéssel élünk, hogy adott $g, d \in R^m$ mellett

$$p \in \{p \in R^m \mid p = g + \Theta d, \Theta \in R\}$$

($g$ és $d$ választására később még visszatérünk).

Ezt $(D)$-be helyettesítve a következő feladatot kapjuk:

$$(RD) \min K + r\Theta + ut - vs$$

$$h\Theta + u - v \geqslant c^*$$

$$u, v \geqslant 0$$

ahol $K = gb, r = db, h = dA, c^* = c - gA$.

Az $(RD)$ feladat könnyen megoldható, ugyanis optimális megoldása a $\Theta$ függvényében explicit módon felírható. Ennek érdekében először rögzítsük $\Theta$-t, így $(RD)$ $n$ darab kisebb feladatra esik szét:

$$\big(j(\Theta)\big) \min u_j t_j - v_j s_j$$

$$u_j - v_j \geqslant c_j^* - h_j\Theta$$

$$u_j, v_j \geqslant 0$$

Belátható, hogy $v\big(j(\Theta)\big)j = 1, \ldots, n$ szakaszonként lineáris konvex függvény, melynek legfeljebb egy töréspontja van (2. ábra).

$$v\big(j(\Theta)\big) = \begin{cases} t_j(c_j^* - h_j\Theta)\,, & \text{ha } j \in I(\Theta) \\ s_j(c_j^* - h_j\Theta)\,, & \text{ha } j \notin I(\Theta) \end{cases}$$

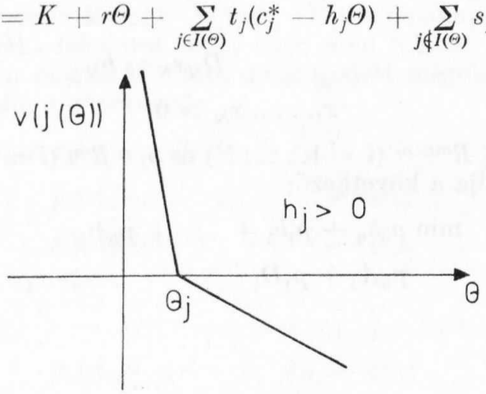
ahol

$$I(\Theta) = \{j \mid \Theta_j \leqslant \Theta \text{ és } h_j < 0 \text{ vagy } \Theta_j > \Theta \text{ és } h_j \geqslant 0\}$$

$$\Theta_j = \begin{cases} c_j^*/h_j\,, & \text{ha } h_j \neq 0 \\ +\infty\,, & \text{ha } h_j = 0 \end{cases}$$

Ezen függvények segítségével felírható az $(RD)$ feladat optimumértéke, mely $\Theta$ függvényében szintén szakaszonként lineáris és konvex lesz, töréspontjai pedig a $\Theta = \Theta_j$, $(j = 1, \ldots, n)$ pontokban adódnak.

$$v\big(RD(\Theta)\big) = K + r\Theta + \sum_{j \in I(\Theta)} t_j(c_j^* - h_j\Theta) + \sum_{j \notin I(\Theta)} s_j(c_j^* - h_j\Theta)$$




2. *ábra*

Mivel ebből könnyen felírható a $v(RD(\Theta))$ függvény akár jobb, akár bal oldali deriváltja, $v(RD) = \min_\Theta v(RD(\Theta))$ egyszerű kereséssel meghatározható.

Hátravan még $g$ és $d$ megválasztása. Elvileg természetesen tetszőlegesen választhatók, tekintetbe kell azonban venni, hogy szükség van a $h = dA$ és a $c^* = c - gA$ mennyiségekre, melyeknek kiszámítása igen munkaigényes lehet, hiszen az egész $A$ mátrixszal kell szorozni. Ezért érdemes úgy választani őket, hogy a $gA$ és $dA$ mennyiségek a korábbi számításokból rendelkezésre álljanak. $d$ és $g$-nek két — ezen szempontokat figyelembe vevő — lehetséges választása a következő:

a) $d = \bar{p}, g = 0$.

Ebben az esetben a $h = \bar{p}A$ vektort a szimplex módszer során amúgy is meg kell határozni, $gA$ pedig a nullvektor lesz. Ezzel a választással $v(RD(\Theta))$ a $\Theta = 1$ helyen az egyszerű felső korlátot adja, tehát $\Theta$ szerint minimalizálva ennél rosszabb eredményt nem kaphatunk.

b) Tegyük fel, hogy az előző iterációban $d'$ és $g'$ választással a minimum $\Theta = \bar{\Theta}$-nál adódott. Legyen

$$d = \bar{p},\ g = g' + \bar{\Theta}d'$$

Ebben az esetben $h = \bar{p}A$ az előzőhöz hasonlóan a szimplex módszerből, $gA$ pedig az előző iteráció adataiból áll rendelkezésre. Mivel $v(RD(\Theta))$ a $\Theta = 0$ választással az előző iterációban kapott korlátot adja, az ilyen módon meghatározott felső korlátok monoton csökkenő sorozatot alkotnak.

## Egyszerű felső korlát a Dantzig—Wolfe-algoritmus esetén

Legyen a megoldandó feladat a következő:

$$(Q)\qquad \begin{aligned} \max c_1x_1 + \ldots + c_Nx_N & \\ A_1x_1 + \ldots + A_Nx_N &= b_0 \\ D_1x_1 \qquad\qquad &= b_1 \\ \vdots \qquad\qquad & \quad \vdots \\ D_Nx_N &= b_N \\ x_1, \ldots, x_N &\geqq 0 \end{aligned}$$

ahol $c_i, x_i \in R^{n_i}$, $A_i \in R^{m_i \times n_i}$ $(i = 1, \ldots, N)$ és $b_i \in R^{m_i}$ $(i = 0, 1, \ldots, N)$.

A $(Q)$ feladat duálja a következő:

$$(QD)\qquad \begin{aligned} \min p_0b_0 + p_1b_1 + \ldots + p_N\mathrm{b}_N & \\ p_0A_1 + p_1D_1 \qquad\qquad &\geqq c_1 \\ \vdots \qquad\qquad & \quad \vdots \\ p_0A_N + \qquad\qquad p_ND_N &\geqq c_N \end{aligned}$$

Legyen $\mathfrak{D}_i = \{x_i \in R^{n_i} \mid D_i x_i = b_i,\ x_i \geqslant 0\} \neq \emptyset$, $\bar{x}_{ij}$, $j \in J_i$ a $\mathfrak{D}_i$ halmaz extremális pontjai, $\bar{\bar{x}}_{ik}$, $k \in K_i$ pedig $\mathfrak{D}_i$ extremális irányai. Ezek felhasználásával $(Q)$ átírható egy ekvivalens alakba, melyet extremális feladatnak nevezünk.

$$(E)\qquad \max \sum_{i=1}^{N} \Big( \sum_{j \in J_i} \lambda_{ij}(c_i \bar{x}_{ij}) + \sum_{k \in K_i} \mu_{ik}(c_i \bar{\bar{x}}_{ik}) \Big)$$

$$\sum_{i=1}^{N} \Big( \sum_{j \in J_i} \lambda_{ij}(A_i \bar{x}_{ij}) + \sum_{k \in K_i} \mu_{ik}(A_i \bar{\bar{x}}_{ik}) \Big) = b_0$$

$$\sum_{j \in J_i} \lambda_{ij} = 1 \qquad (i = 1, \ldots, N)$$

$$\lambda_{ij} \geqslant 0 \qquad i = 1, \ldots, N\ j \in J_i$$

$$\mu_{ij} \geqslant 0 \qquad i = 1, \ldots, N\ k \in K_i$$

Mivel az összes extremális pont és irány explicit módon való előállítása túl nagy munkát igényelne, az algoritmus során általában $(E)$-nek csak egy olyan része áll rendelkezésre, mely nem tartalmaz minden oszlopot. Egy iterációban először ezt a szűkebb feladatot oldjuk meg. Ennek optimális duál megoldása $(\bar{p}_0, \bar{r})$, ahol $\bar{p}_0 \in R^{m_0}$, $\bar{r} \in R^N$. További oszlopokat ezután az alábbi részfeladatokból kaphatunk:

$$(S_i)\qquad \max\,(c_i - \bar{p}_0 A_i)x_i$$

$$D_i x_i = b_i$$

$$x_i \geqslant 0$$

Ha mindegyik $(S_i)$ feladat megoldása korlátos, az optimális duál megoldásokat jelöljük $\bar{p}_i$-vel. Ekkor

$$v(Q) \leqslant \bar{p}_0 b_0 + \bar{p}_1 b_1 + \ldots + \bar{p}_N b_N\,.$$

Mivel könnyen látható, hogy $(\bar{p}_0, \bar{p}_1, \ldots, \bar{p}_N)$ kielégíti $(QD)$ feltételeit, tehát egy duál-megengedett megoldás. Természetesen ha valamely $i$-re $(S_i)$ megoldása nem korlátos, végtelen felső korlát adódik.

## Módosított felső korlát a Dantzig—Wolfe-algoritmus esetén

A módosított felső korlátot hasonló ötlettel lehet megkonstruálni, mint a szimplex módszer esetén. Mivel a Dantzig—Wolfe dekompozíciós algoritmus valójában az extremális feladatot oldja meg, ezen feladat duáljának fogjuk egy aránylag könnyen meghatározható megengedett megoldását megkeresni.

Az $(E)$ feladat duálja a következő:

$$(ED)\qquad \min\, p_0 b_0 + r_1 + \cdots + r_N$$

$$\begin{array}{lll}
p_0(A_1 \bar{x}_{1j}) + r_1 & \geqslant c_1 \bar{x}_{1j} & j \in J_1 \\
p_0(A_1 \bar{\bar{x}}_{1k}) & \geqslant c_1 \bar{x}_{1k} & k \in K_1 \\
\vdots & \vdots & \vdots \\
p_0(A_N \bar{x}_{Nj}) + \quad r_N & \geqslant c_N \bar{x}_{Nj} & j \in J_N \\
p_0(A_N \bar{\bar{x}}_{Nk}) & \geqslant c_N \bar{x}_{Nk} & k \in K_N
\end{array}$$

Az $(ED)$ feladatban azzal a megkötéssel élünk, hogy adott $g, d \in R^{m_0}$ esetén

$$p_0 \in \{p_0 \in R^{m_0} \mid p_0 = g + \Theta d, \Theta \in R\}$$

Ha a $p_0 = g + \Theta d$ képletet behelyettesítjük $(ED)$-be és $\Theta$-t rögzítjük, $(ED)$ $N$ darab kisebb feladatra esik szét:

$$\begin{aligned} &(i(\Theta)) \ \min r_i \\ &r_i \geqslant \big(c_i - (g + \Theta d)A_i\big)\bar{x}_{ij} \qquad && j \in J_i \\ &0 \geqslant \big(c_i - (g + \Theta d)A_i\big)\bar{\bar{x}}_{ik} \qquad && k \in K_i \end{aligned}$$

Ez a következő ekvivalens alakba írható:

$$\begin{aligned} &r_i = \max_{j \in J_i} \big(c_i - (g + \Theta d)A_i\big)\bar{x}_{ij} \\ &\big(c_i - (g + \Theta d)A_i\big)\bar{\bar{x}}_{ij} \leqslant 0 \qquad && k \in K_i \end{aligned}$$

E feladat feltételei azt jelentik, hogy a $\mathfrak{D}$ halmaznak ne legyen olyan extremális iránya, amelynek mentén a $c_i - (g + \Theta d)A_i$ függvény növekszik. Ha ilyen van, felső korlátként végtelent kapunk. Ha ilyen nincs, vagyis erre a célfüggvényre nézve a $\mathfrak{D}$ halmaz korlátos, $\big(i(\Theta)\big)$ optimális célfüggvényértéke $(r_i)$ nem változik, ha a feladatot a következő alakba írjuk:

$$\begin{aligned} &(i'(\Theta)) \ r_i = \max \big(c_i - (g + \Theta d)A_i\big)x_i \\ &D_i x_i = b_i \\ &x_i \geqslant 0 \end{aligned}$$

Legyen $g = \bar{p}_0$, ahol $\bar{p}_0$ az algoritmus során utoljára megoldott extremális feladat optimális duál megoldása. Ekkor $\Theta = 0$-ra ezt a feladatot amúgy is meg kell oldani, majd parametrikus programozással meghatározható, hogy $r_i$ hogyan változik a $\Theta$ függvényében.

Könnyen belátható, hogy $r_i(\Theta)$ szakaszonként lineáris, konvex függvény, melynek töréspontjait, és ezen töréspontokban a függvényértékeket a parametrikus programozás szolgáltatja. Ezekből viszont explicit módon felírható az $r_i(\Theta)$ függvény.

Mivel

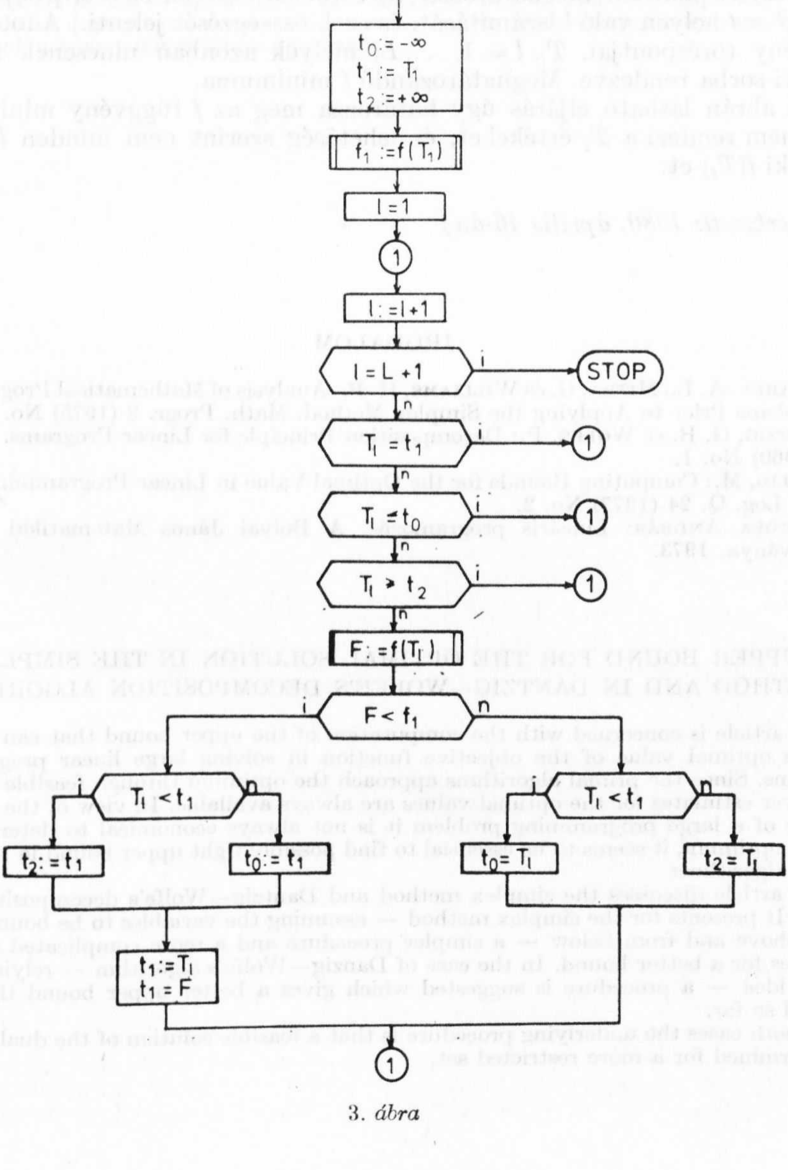
$$v\big(ED(\Theta)\big) = (g + \Theta d)\, b_0 + \sum_{i=1}^{N} r_i(\Theta),$$

vagyis szakaszonként lineáris konvex függvények összege, ezért a $v\big(ED(\Theta)\big)$ függvény maga is szakaszonként lineáris és konvex lesz, és töréspontjai is megegyeznek az $r_i(\Theta)$ függvények töréspontjaival. Így e pontok kiértékelésével a $v\big(ED(\Theta)\big)$ függvény minimuma meghatározható (lásd a Függelékben).

Amint várható is volt, a Dantzig—Wolfe dekompozíciós algoritmusra a módosított felső korlát kiszámítása lényegesen nagyobb munkát igényel, mint az egyszerű lineáris programozási feladat esetén, hiszen a parametrikus programozás is, és a $v\big(ED(\Theta)\big)$ függvény minimumának meghatározása is elég sok számítást igényelhet.

Ezért a módosított felső korlátot nem érdemes minden iterációban kiszámítani, hanem csak időnként, esetleg adaptív módon, az algoritmus részeredményeitől függően. Egy lehetőség például, hogy ezt a számítást csak akkor végezzük el, mikor bizonyos egyéb kritériumok (pl. a célfüggvény növekedési ütemének csökkenése) alapján úgy tűnik, hogy nem érdemes további iterációkat végezni, a jelenleg rendelkezésre álló megengedett megoldás már „elég jó" célfüggvényértéket ad.

Ha a felső korlát számítását sűrűbben kívánjuk elvégezni, további számításokat lehet megtakarítani azáltal, hogy a $v(ED(\Theta))$ függvény minimumának csak egy közelítésével dolgozunk. Például a parameterikus programozást




3. *ábra*

csak egy $S \leq \Theta \leq T$ intervallumban végezzük, vagy a $v(ED(\Theta))$ függvény minimumának meghatározásakor nem veszünk figyelembe minden töréspontot. (Esetleg csak azokat, amelyeknél az egyes $r_i(\Theta)$ függvények minimuma volt.)

## Függelék

A $v(ED(\Theta))$ függvény minimumának meghatározására szolgáló rutinnak a következő feladatot kell megoldania: tudjuk, hogy az $f$: $R \to R$ függvény szakaszonként lineáris, konvex, és rendelkezésre áll egy olyan rutin, mely tetszőleges $t$ pontban meghatározza $f(t)$-t. (Jelen esetben ez a $v(r_i(\Theta))$ függvények $\Theta = t$ helyen való kiszámítását, és ezek összegzését jelenti.) Adottak az $f$ függvény töréspontjai, $T_l$, $l = 1, \ldots, L$, melyek azonban nincsenek nagyság szerinti sorba rendezve. Meghatározandó $f$ minimuma.

A 3. ábrán látható eljárás úgy határozza meg az $f$ függvény minimumát, hogy nem rendezi a $T_l$ értékeket, és lehetőség szerint nem minden $l$-re számítja ki $f(T_l)$-et.

*(Beérkezett: 1980. április 16-án)*

## IRODALOM

1. Brearly, A. L., Mitra, G. és Williams, H. P.: Analysis of Mathematical Programming Problems Prior to Applying the Simplex Method. Math. Progr. 8 (1975) No. 1.
2. Dantzig, G. B. és Wolfe, P.: Decomposition Principle for Linear Programs. Op. Res. 8 (1960) No. 1.
3. Kallio, M.: Computing Bounds for the Optimal Value in Linear Programming. Naval Res. Log. Q. 24 (1977) No. 2.
4. Prékopa András: Lineáris programozás. A Bolyai János Matematikai Társulat kiadványa. 1973.

## UPPER BOUND FOR THE OPTIMAL SOLUTION IN THE SIMPLEX METHOD AND IN DANTZIG–WOLFE'S DECOMPOSITION ALGORITHM

The article is concerned with the computation of the upper bound that can be given for the optimal value of the objective function in solving large linear programming problems. Since the primal algorithms approach the optimum through feasible solutions the lower estimates for the optimal values are always available. In view of the fact that in case of a large programming problem it is not always economical to determine the precise optimum, it seems to be essential to find possibly tight upper bound in the course of the algorithms.

The article discusses the simplex method and Dantzig–Wolfe's decomposition algorithm. It presents for the simplex method — assuming the variables to be bounded both from above and from below — a simpler procedure and a more complicated one, that provides for a better bound. In the case of Danzig–Wolfe's algorithm — relying on the above idea — a procedure is suggested which gives a better upper bound than these applied so far.

In both cases the underlying procedure is that a feasible solution of the dual program is determined for a more restricted set.

## ВЕРХНИЙ ПРЕДЕЛ ОПТИМАЛЬНОГО РЕШЕНИЯ У СИМПЛЕКСНОГО МЕТОДА И ДЕКОМПОЗИЦИОННОГО АЛГОРИТМА ДАНЦИГА-ВОЛЬФА

Настоящая статья занимается вычислением верхнего предела целевой функции оптимального решения в случае задач линейного программирования больших размеров. Так как алгоритмы прямого решения приближаются к оптимуму через допустимые решения, всегда имеется оценка нижнего предела оптимума. Имея в виду, что при решении задач большого размера часто не экономично определять точной оптимум, определение верхнего предела как можно точнее кажется очень важным в алгоритмах.

В статье подробно обсуждаются симплексный метод и декомпозиционный алгоритм Данцига-Вольфа. Предполагая, что для отдельных переменных известны верхние и нижние пределы, в статье дается один простой и один более сложный метод, которые обеспечивают лучшие пределы. Для алгоритма Данцига-Вольфа на основе вышеупомянутой идеи излагается метод, который определяет верхний предел лучше, чем раньше.

В обоих случаях основной идеей является то, что определяется допустимое решение обратной задачи в более узком интервале.