

Egy közelítő eljárás lineáris programozási feladatok megoldására

A lineáris programozási feladatokat numerikusan általában a szimplex módszer valamelyik változatával oldják meg (bár ismeretesek egyéb módszerek is). Ez természetesen nem véletlen. A szimplex módszernek sok előnyös tulajdonsága van: véges (és gyakorlatilag elfogadható) számú lépésben megadja mind a primál, mind a duál optimális megoldást (vagy jelzi, hogy a feladatnak nincs megoldása), nem igényli a feladat egy lehetséges megoldásának ismeretét, és igen hatékony olyan számítássorozatok esetén, ahol az egyes feladatok csak kismértékben különböznek egymástól.

Van azonban a szimplex módszernek egy olyan tulajdonsága, amely nagyobb méretű feladatok esetén már nehézségeket okozhat: a feltételek számával megegyező rendű inverzsorozat¹ kiszámítását igényli, ahol a sorozat egyes tagjait a megelőző tag felhasználásával számítjuk ki. Nagyobb méretűknél² egyetlen inverz kiszámításához is igen sok műveletre van szükség: ez a kerekítési hibák felhalmozódásához vezethet. Fokozza a problémát, hogy az egymást követő inverzek számítási módja miatt bármelyik inverz hibáját „öröklik” az utána következő inverzek.

A kerekítési hibák felhalmozódásának kivédésére sok módszer ismeretes: dupla pontosságú műveletek alkalmazása, újrainvertálás, stb. A nagyobb pontosságnak természetesen ára van: megnő az időigény, dupla pontosságú műveletek alkalmazása esetén nagyobb lesz a memóriaigény. Így a feladat méreteinek növekedésével elérünk egy olyan határhoz, amelyen túl már e módszerek felhasználása sem segít.

Ezért indultak meg — és folynak jelenleg is — olyan kutatások, amelyek célja az, hogy bizonyos (a gyakorlatban gyakran előforduló) feladattípusokra ezek speciális szerkezetét kihasználó, a feltételek számánál lényegesen kisebb méretű inverzet (vagy inverzeket) alkalmazó algoritmusokat szerkesszenek. Így születtek meg a különböző dekompozíciós eljárások és az ezek speciális esetének tekinthető egyedi és általánosított felsőkorlát-technikák. (Ez utóbbiak igen hasznos módszereknek bizonyultak. Kevésbé mondható el ugyanez az általános dekompozíciós eljárásokra: ezek a gyakorlati tapasztalatok szerint nagyon sok iterációt igényelnek, emiatt a kerekítési hibák felhalmozódnak.)

¹ A gyakorlatban általában a módosított szimplex-módszer különböző változatait alkalmazzák.

² Hogy mekkora feladatot tekintünk „nagy”-nak, ez viszonylagos. A felhasználandó gép jellemzőitől függ: a belső- és a háttérmemóriák kapacitásától, az elérési időktől, a műveleti sebességtől. A hazánkban jelenleg található legjobb gépeket alapul véve a „közepes” és „nagy” feladatok határa kb. 700 – 800 feltételnél van.

Végső soron azonban ezen eljárások esetében is az inverz méretei határozzák meg a megoldható feladat nagyságát.

Ezért elképzelhető, hogy nagyméretű feladatok esetén a szimplex módszerrel eredményesen versenyezzenek olyan eljárások, amelyek nem alkalmaznak inverzet és kevésbé érzékenyek a kerekítési hibákra. Különösen előnyösek az utóbbiak, ha ki tudják használni, ha e matrix nagyon „üres”, és azt is, ha rendelkezésre áll egy, az optimálishoz viszonylag közel eső, kiinduló program. A gyakorlatban mindkét eset gyakran előfordul.

Az Országos Tervhivatal egyik kidolgozás alatt levő modelljének³ megoldására széleskörű kutatómunka folyik. Ennek keretében több ilyen algoritmus áll gépi kipróbálás alatt. Ezek közé tartozik az alábbi eljárás is.

A feladat

Az egyszerűbb tárgyalás kedvéért csak olyan lineáris programozási feladatokkal foglalkozunk, ahol minden feltétel „ \leq ” alakban van előírva (egyetlen feltételnek sem kell egyenlőségre teljesülnie). Ismeretes, hogy bármely lineáris programozási feladat megfogalmazható ilyen formában.

Megoldandó tehát az

$$(1) \quad \begin{cases} Ax \leq b \\ c^*x \rightarrow \max \end{cases}; \quad A = \begin{bmatrix} a_1^* \\ \vdots \\ a_m^* \end{bmatrix}; \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}; \quad c = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}; \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

feladat. (Az esetleges nemnegativitási feltételeket nem kezeljük megkülönböztetett módon, ezek is benne foglaltatnak az (1) feltételi rendszerét alkotó „ m ” egyenlőtlenségben.)

A továbbiakban (1) lehetséges megoldásainak halmazát X -szel, optimális megoldásainak halmazát pedig X^0 -lal jelöljük.

Szükségünk lesz még (1) duálisára is:

$$(1a) \quad \begin{cases} u \geq 0 \\ u^*A = c^* \\ u^*b \rightarrow \min \end{cases}$$

Jelöljük (1a) lehetséges megoldásainak halmazát U -val, optimális megoldásainak halmazát U^0 -lal.

A közelítő feladat

(1)-hez a következő „közelítő feladatot” rendeljük:

$$(2) \quad \Phi_s(x) = c^*x - s \varphi^*(x) \varphi(x) \rightarrow \max,$$

³ Lásd: Báger—Morva—Szabó: A közleplejárati népgazdasági terv programozása, Közgazdasági Szemle, 1969. 7—8. sz.

ahol: s megfelelően nagyra választott pozitív *konstans*, továbbá

$$(2a) \quad \varphi(x) = \begin{bmatrix} \varphi_1(x) \\ \vdots \\ \varphi_m(x) \end{bmatrix}, \text{ ahol } \varphi_i(x) = \begin{cases} a_i^*x - b_i & \text{ha } a_i^*x - b_i > 0 \\ 0 & \text{egyébként} \end{cases}; \quad i = 1, 2, \dots, m.$$

(2) optimális megoldásainak halmazát X^0 -al fogjuk jelölni.

Könnyen belátható, hogy $\Phi_s(x)$ konkáv és mindenütt folytonosan differenciálható függvénye x -nek, továbbá

$$(3) \quad g_s(x) = \text{grad } \Phi_s(x) = [c^* - 2s \varphi^*(x) A]^*.$$

Egyébként E_n -nek⁴ minden olyan poliedrikus tartományában, amelyben $Ax - b$ egyetlen komponense sem vált előjelet, $\Phi_s(x)$ kvadratikus függvénye x -nek. Ezen tartományok száma véges (legfeljebb 2^m), teljesen lefedik E_n -t és két ilyen tartomány legfeljebb a határán érintkezik egymással.

Összefüggések az eredeti és a közelítő feladat között

Az eljárás lényege: az (1) feltételes szélsőértékfeladat helyett a (2) feltétel nélküli feladatot oldjuk meg. (Tehát egy SUMT típusú eljárásról⁵ van szó.)

A közelítés jogosságát a következő (itt nem bizonyított) tétel igazolja:

1. tétel: Tegyük fel, hogy X^0 nem üres és korlátos. Tekintsünk egy szigorúan monoton növekvő, végtelenhez tartó, pozitív $\{s_k\}_1^\infty$ sorozatot. Ekkor

a) $X_{s_k}^0 \neq \emptyset$ ($k = 1, 2, \dots$),

továbbá tetszőleges $S = \{x_{s_k}\}$ sorozat esetén ($x_{s_k} \in X_{s_k}^0$),

$$b) \quad c^*x_{s_k} \geq c^*x_{s_{k+1}} \text{ és } \lim_{k \rightarrow \infty} c^*x_{s_k} = c^*x^0 \quad (x^0 \in L^0),$$

$$c) \quad q^*(x_{s_k}) q(x_{s_k}) \geq q^*(x_{s_{k+1}}) q(x_{s_{k+1}}) \text{ és } \lim_{k \rightarrow \infty} q^*(x_{s_k}) q(x_{s_k}) = 0,$$

d) S -nek legalább egy torlódási pontja van, és bármely torlódási pontja optimális megoldása (1)-nek.

A tétel speciális esetként következik McCormick és Fiacco nemlineáris feladatokra vonatkozó tételéből ([3], 104. o.).

E tétel alapján [3] a következő eljárást ajánlja (55. o.): $k = 1$ -től kezdve oldjuk meg a $\Phi_{s_k}(x) \rightarrow \max$ feladatokat. Haladjunk ilyen módon addig, amíg a kapott megoldás elég jól közelíti (1) megoldását. (A korábbi feladatok megoldásából következtetni tudunk a következő feladat megoldására.)

McCormick és Fiacco egy véges eljárást is ad (2) megoldására ([3], 180. o.), amelynek során a változók számával megegyező rendű inverzekkel kell dolgozni.

⁴ E_n -nel az n -dimenziós euklideszi teret jelöljük.

⁵ Lásd: [2], [3] és [4]. Ezek az eljárások bizonyos matematikai programozási (feltételes szélsőérték-számítási) feladatok közelítő megoldására alkalmasak. Lényegük: képezünk egy olyan függvényt, amelynek feltétel nélküli szélsőértékhelye közelítő megoldása lesz az adott programozási feladatnak. Ennek a függvénynek az egyik tagja a megoldandó feladat célfüggvénye, másik tagja pedig azt méri (valamilyen mérték [„segédfüggvény”] szerint), hogy „milyen messze” vannak az egyes programok a lehetséges megoldások halmazának határától.

A dolgozat egyrészt az 1. tétel általánosításával foglalkozik, másrészt olyan összefüggésekkel, amelyek lehetővé teszik, hogy bizonyos információk birtokában egyetlen feltétel nélküli feladat megoldásával előre adott pontosságú közelítő megoldást kapjunk (1)-re. Tárgyal továbbá egy (általában véges), invertálást nem igénylő eljárást (2) megoldására, végül rámutat arra, hogy a dualitást felhasználva (1) megoldására több hasonló eljárás alkalmazható.

A továbbiakban gyakran fel fogjuk használni a következő lemmát ([3], 98. o.):

1. lemma: $x_s \in X_s^0$ akkor és csak akkor, ha $2s\varphi(x_s) \in U$

Bizonyítás: Minthogy $\Phi_s(x)$ konkáv és differenciálható, $g_s(x_s) = 0$ szükséges és elégséges ahhoz, hogy $x_s \in X_s^0$ fennálljon. Ebből (3)-at és (2a)-t felhasználva a lemma közvetlenül következik.

2. tétel: (2) megoldhatóságának szükséges és elégséges feltétele, hogy U ne legyen üres.

Bizonyítás: A szükségesség az 1. lemmából következik. Az elégségesség belátásához tekintsük a

$$(4) \quad \begin{cases} Ax - y \leq b \\ c^*x - sy^*y \rightarrow \max \end{cases}$$

feladatot. Mivel

$$\Phi_s(x) = \max_y \{c^*x - sy^*y \mid Ax - y \leq b\},$$

nyilvánvaló, hogy X_s^0 akkor és csak akkor nem üres, ha (4)-nek van optimális megoldása. (4) célfüggvénye kvadratikus, ezért az optimum létezéséhez elégséges a célfüggvény korlátossága. Szorozzuk be (4) feltételi rendszerének mindkét oldalát egy tetszőleges $u \in U$ -val. Ekkor az

$$(5) \quad \begin{cases} c^*x - u^*y \leq u^*b \\ c^*x - sy^*y \rightarrow \max \end{cases}$$

feladathoz jutunk. Minthogy $u \geq 0$, (4) bármely lehetséges megoldása lehetséges megoldása lesz (5)-nek is. Így (5) optimális célfüggvényértéke (ha létezik) legalább akkora, mint (4)-é. Viszont (5) bármely lehetséges megoldására fent áll, hogy

$$c^*x - sy^*y \leq u^*b + u^*y - sy^*y \leq \max_y \{u^*b + u^*y - sy^*y\} = u^*b + u^*u/4s.$$

A továbbiakban fel fogjuk használni a fenti bizonyításból következő

$$(6) \quad \Phi_s(x) \leq u^*b + u^*u/4s \quad (u \in U)$$

egyenlőtlenséget.

3. tétel: Ha (1)-nek van optimális megoldása, akkor

$$(7) \quad 2s\varphi^*(x_s)\varphi(x_s) \leq c^*x_s - c^*x^0 \leq s\varphi^*(x_s)\varphi(x_s) + u^{0*}u^0/4s,$$

ahol $x_s \in X_s^0$, $x^0 \in X^0$ és $u^0 \in U^0$.

Bizonyítás: Felhasználva az 1. lemmát kapjuk, hogy

$$(8) \quad c^*x_s - c^*x^0 = 2s\varphi^*(x_s)(Ax_s - Ax^0) = 2s\varphi^*(x_s)[(Ax_s - b) + (b - Ax^0)].$$

Mint ahogy (2.a) miatt $\varphi^*(x_s) (Ax_s - b) = \varphi^*(x_s) \varphi(x_s)$, (8)-ből következik, hogy

$$2s \varphi^*(x_s) \varphi(x_s) \leq cx_s - cx^0.$$

A második egyenlőtlenség bizonyításához tekintsük (6)-ot. Legyen $u = u^0$; így $b^*u^0 = c^*x^0$, és

$$(9) \quad \Phi_s(x_s) = c^*x_s - s \varphi^*(x_s) \varphi(x_s) \leq c^*x^0 + \frac{1}{4s} u^{0*}u^0,$$

amiből állításunk már következik.

(7)-et a következőképpen is megfogalmazhatjuk:

$$(10) \quad 2s \varphi^*(x_s) \varphi(x_s) \leq c^*x_s - c^*x^0 \leq \frac{1}{2s} u^{0*}u^0,$$

ami nyilvánvalóan igaz, mivel (7) alapján

$$s \varphi^*(x_s) \varphi(x_s) \leq \frac{1}{4s} u^{0*}u^0.$$

4. tétel: Ha (1)-nek van optimális megoldása, akkor

$$(11) \quad 2s \varphi^*(x_s) b - u^{0*}b \leq \frac{1}{4s} u^{0*}u^0 - s \varphi^*(x_s) \varphi(x_s) \leq \frac{1}{4s} u^{0*}u^0.$$

Bizonyítás: Felhasználva, hogy

$$2s \varphi^*(x_s) b = 2s \varphi^*(x_s) [Ax_s - \varphi(x_s)] = c^*x_s - 2s \varphi^*(x_s) \varphi(x_s),$$

(9) alapján állításunk következik.

A közelítő feladat megoldása

Mivel (2) megoldása nem más, mint egy differenciálható konkáv függvény feltétel nélküli maximum-helyének meghatározása, kézenfekvőnek látszik valamilyen gradiens-módszert alkalmazni. Mivel $\Phi_s(x)$ tartományonként kvadratikus, hasznos eszköz lehet a kvadratikus függvények feltétel nélküli szélsőérték-helyének meghatározására szolgáló ún. „konjugált gradiensek módszere”⁶ egy, az adott függvény természetének megfelelően módosított változata.

Az egyik lehetséges változat — vázlatosan — a következő:

1. iteráció: tetszőleges $x_0 \in E_n$ -ből kiindulva x_1 -nek a

$$\max \{ \Phi_s(x) \mid x = x_0 + t g_s(x_0) \}$$

feladat x_0 -hoz legközelebb eső optimális megoldását választjuk.

⁶ Lásd: [1], 195–200. o., vagy [6]. A konjugált gradiensek módszere iteratív algoritmus, amely tetszőleges x_0 kezdőpontból indulhat, és egyre jobb függvényértékeket szolgáltató pontokon keresztül haladva véges számú lépésben eléri egy konkáv kvadratikus $Q(x)$ függvény maximumhelyét. x_i -vel jelölve az i -edik közelítést, x_{i+1} -et az $x_i + td_i$ egyenes legjobb függvényértéket adó pontjaként határozza meg, ahol $d_0 = g_0$, és $d_i = -g_i^* + (g_i^*g_i/g_{i-1}^*g_{i-1}) d_{i-1}$ ($g_i = \text{grad } Q(x_i)$).

Áttérve az $i + 1$ -edik iterációra ($i = 1, 2, \dots$):

$i + 1$ -edik iteráció: x_{i+1} -nek a

$$\max \{ \Phi_s(x) \mid x = x_i + t d_i \}$$

feladat x_i -hez legközelebb eső optimális megoldását választjuk, ahol:

$$d_i = \begin{cases} g_s(x_i) + \frac{g_s^*(x_i) g_s(x_i)}{g_s^*(x_{i-1}) g_s^*(x_{i-1})} d_{i-1}, & \text{ha} \\ (a_j^* x_i - b_j)(a_j^* x_{i-1} - b_j) \geq 0 & \text{minden } j\text{-re, és } \sum_{j=1}^n \varphi_j(x_i) > 0, \\ g_s(x_i) & \text{egyébként.} \end{cases}$$

Tehát tulajdonképpen arról van szó, hogy valahányszor $\Phi_s(x)$ új kvadratikus függvény alakját ölti fel, újra kezdjük (most már az új függvényre) a konjugált gradiens módszert. Az eljárás monoton és $\Phi_s(x)$ egy maximumhelyéhez konvergált, ha $X_s^0 \neq \emptyset$.

Az eljárás véges, ha $a_j^* \hat{x} \neq b_j$ ($j = 1, 2, \dots, m$), ahol: $\hat{x} = \lim_{i \rightarrow \infty} x_i$. Ekkor ugyanis létezik \hat{x} -nek olyan $K(\hat{x})$ környezete, hogy

$$(12) \quad \Phi_s(x) = c^* x - s \sum_{\varphi_j(x) > 0} (a_j^* x - b_j)^2 \quad \text{ha } x \in K(\hat{x}),$$

tehát ezen x -ekre $\Phi_s(x)$ egyszerű kvadratikus függvény; továbbá létezik olyan „ M ” küszöbszám, hogy $x_i \in K(\hat{x})$, ha $i \geq M$.

Igy a fent vázolt algoritmus $i \geq M$ esetén megegyezik a (12)-re alkalmazott konjugált gradiens módszerrel. Ez az eljárás pedig véges számú (legfeljebb n) lépésben ad eredményt.

Az eljárás néhány változata

A fentiek alapján többféle eljárás képzelhető el lineáris programozási feladatok megoldására:

1. A legegyszerűbbnek a következő megoldás látszik: a megoldandó feladat duálisából indulunk ki, erre írjuk fel és oldjuk meg a közelítő feladatot.

A gyakorlatban mindig tudunk megbízható felső becslést adni a primál feladat optimális megoldásának egyes komponenseire, így (11) alapján s megválasztható úgy, hogy az adott pontosság biztosítva legyen. Az 1. lemma és a 4. tétel alapján az eljárás olyan lehetséges megoldást szolgáltat a primál feladatra, amelynek célfüggvény-értéke legfeljebb előre megadott nagysággal tér el az optimálistól. Ezt a változatot akkor célszerű alkalmazni, ha meg tudunk adni egy, a duális feladat optimális megoldásától nem túl messze eső kiinduló megoldást. (Ennek nem kell lehetséges megoldásnak lennie.)

2. Ha meg tudunk adni egy, a primál feladat optimális megoldásától nem túl messze eső kiinduló megoldást, akkor a primál feladatra oldjuk meg a közelítő feladatot.

Amennyiben a duális feladatra vonatkozó megfelelő információk rendelkezésre állnak, a (10) összefüggés alapján s előre megadható úgy, hogy a kapott közelítő primál megoldás mind a célfüggvény-értéket, mind az egyes feltételek megsértésének mértékét tekintve eleget tegyen az előre megadott pontossági követelményeknek.

Ha ezen túlmenően még lehetséges megoldást is akarunk a primál feladatra kapni, a következőképpen járhatunk el: megoldjuk a primál feladatra vonatkozó közelítő feladatot: ezután a feltételei rendszert bővítjük egy, a célfüggvény — (10)-ből vagy (11)-ből meghatározott — alsó korlátját előíró feltétellel, c helyébe pedig 0-t írunk, és erre a feladatra oldjuk meg a közelítő feladatot. Ehhez a korábban kapott közelítő megoldás jó kiinduló megoldást szolgáltat.

3. Ha sem a primál, sem a duál feladatra nem állnak rendelkezésre megfelelő információk (ez a gyakorlatban általában nem fordul elő), akkor tetszőlegesen s mellett megoldjuk a közelítő feladatot mind a primál, mind a duál feladatra. Ha a pontosság nem kielégítő, nagyobb s -sel megismételjük az eljárást. A (10) és (11) összefüggések biztosítják, hogy így előbb-utóbb elérjük a megfelelő pontosságú megoldást.

4. Megadható olyan eljárás is, amely a primál, illetve duál feladat optimális megoldásához konvergál: a

$$\begin{aligned} -x &\leq 0 \\ Ax &\leq b \\ -u &\leq 0 \\ -A^*u &\leq -c \\ -c^*x + b^*u &\leq 0 \end{aligned}$$

egyenlőtlenség-rendszerhez a (2a)-hoz hasonló módon megkonstruáljuk a $\varphi(x; u)$ függvényeket, és megoldjuk a

$$\varphi^*(x; u) \varphi(x; u) \rightarrow \min$$

feladatot. Amennyiben az optimális célfüggvényérték 0, megkaptuk a primál és duál feladat egy-egy optimális megoldását.

A 2. változat gépi kipróbálása folyamatban van. A kísérletek eredményéről másutt számolunk majd be.

(Beérkezett: 1970. január 23.)

IRODALOM

1. BODEWIG, E.: Matrix Calculus. Amsterdam, 1959. North-Holland.
2. FIACCO, A. V. and Mc'CORMICK, G. P.: The Sequential Unconstrained Minimization Technique for Nonlinear Programming. Management Science, Vol. 10, No. 2, January, 1964. 360—366. p.
3. FIACCO, A. V. and Mc'CORMICK, G. P.: Nonlinear Programming: Sequential Unconstrained Minimization Techniques. John Wiley and Sons, 1968.
4. FORGÓ, F.: Egy módszer nemlineáris programozási problémák közelítő megoldására. Szigma, 1959. 1. szám, 67—74. o.
5. KREKÓ, B.: Nemlineáris programozás (kiadás alatt).
6. RALSTON, A.: Bevezetés a numerikus analízisbe. Budapest, 1969. Műszaki Könyvkiadó.

AN APPROXIMATION METHOD FOR THE SOLUTION OF LINEAR PROGRAMMING PROBLEMS

In our country a wide-range research work is going on with the aim to work out algorithms and computer programmes enabling the solution of large linear programming problems. The procedure described in the article belongs to those attempts which expect to overcome the computational difficulties involved in the treatment of a large inverse by employing algorithms of „non-simplex” type.

The essence of our procedure is that instead of the linear programming problem (1) we solve the unrestricted extreme value problem (2). For the solution of (2) we can apply gradient methods, whereby the rounding-off errors cause less difficulty than in the case of the simplex method. A version of the method of conjugate gradients can also be applied solving (2) generally in a finite number of steps. The procedure yields an approximation to the solution of (1) and (1a). The formula (10) and (11) [in which x^0 , u^0 and x_s are the optimal solutions to (1), (1a) and (2)] enable us — to change s so that a prescribed accuracy be ensured, if we can estimate the primal or dual optimal solution's components.

A computer programme has been prepared for the procedure and the trial computations are under way.

ПРИБЛИЖЕННОЕ РЕШЕНИЕ ДЛЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В нашей стране в настоящее время происходит широкое исследование для выработки алгоритма и машинных программ, которые дают возможность решить задачи линейного программирования большого размера. Метод, излагаемый в статье, относится к той группе исследований, которая надеется решить проблемы расчёта в связи с обращением обратной матрицей большого размера, с помощью алгоритмов «не-симплекс» характера.

Сущность метода состоит в том, что вместо задачи линейного программирования (1) решает экстремальную задачу без ограничений (2). Для решения задачи (2) можем употреблять градиентные методы, таким образом ошибки округления вызывают меньше проблем, чем при симплекс методом. Можно употреблять такой вариант «метода сопряжённых градиентов», который решает задачу (2) обычно в конечном числе шагов. Этот метод даёт приближённое решение задач (1) и (1a). Формулы (10) и (11) (в которых x^0 , u^0 и x_s являются оптимальным решением задач (1), (1a) и (2)) дают возможность, зная порядок компонентов решения прямой или двойственной задачи, изменить s таким образом, чтобы заранее заданная точность была обеспечена.

На этот метод составлена машинная программа, идут предварительные расчёты.