

## РЕШАВАЊЕ ЗАДАЧИ СО СИМПЛЕКС МЕТОД ВО ЕХСЕЛ И МАТНЕМАТИСА

---

Марија Љубеновска <sup>1</sup>

Билјана Златановска <sup>1</sup>

Линеарното програмирање (оптимизација) е математичко моделирање на практични проблеми со линеарна функција на целта којашто треба да се оптимизира (максимизира или минимизира). Зборот „програмирање“ се однесува на наоѓање на најдобриот (оптимален) план, т.е. одредување на вредности на променливите така што функцијата на целта да добие најдобра (оптимална) вредност при што се задоволени ограничувањата на задачата. Линеарното програмирање наоѓа примена во решавање на практични задачи од разни области. Во областа на економијата можат да се разгледуваат задачи за оптимизација на инвестиции; во областа на земјоделието – оптимизација на производството во земјоделското стопанство при дадени природноекономски услови; во областа на индустријата – оптимизација на производство на различни производи, оптимизација на транспорт на производите, итн. За овој вид математичко моделирање најмногу се заинтересирани компаниите кои во него гледаат алатка за успешност на пазарот, намалување на трошоците при производство и транспорт на своите производи и зголемување на профитот ([1,5]).

Најчесто формулацијата на една задача на линеарното програмирање ([1]) се состои од преточување на проблемот во практична задача за наоѓање решение  $(x_1, x_2, \dots, x_n)$  на системот неравенки

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\geq b_2 \\ \dots &\dots \end{aligned} \tag{1}$$

$$\begin{aligned} a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\geq b_m \\ x_j &\geq 0, j = 1, \dots, n \end{aligned} \tag{2}$$

за кои функцијата

$$z = c_1x_1 + c_2x_2 + \dots + c_nx_n \tag{3}$$

прима максимална или минимална вредност.

Неравенствата (1) се нарекуваат *ограничувања* на задачата на линеарното програмирање. Условите (2) се *услови за ненегативност* на променливите. А (3) се нарекува *функција на целта* којашто треба да се оптимизира. За  $n$ -ката вредности  $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$  велíme дека е *допустливо решение* или *програма*, ако ги исполнува ограничувањата (1) и условите за ненегативност (2).

Доколку функцијата на целта (3) треба да се максимизира, тогаш допустливото решение или програмата  $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$  е *максимална* ако

$$c_1\bar{x}_1 + c_2\bar{x}_2 + \dots + c_n\bar{x}_n \geq c_1x_1 + c_2x_2 + \dots + c_nx_n$$

за секое друго допустливо решение  $(x_1, x_2, \dots, x_n)$ . Максималната програма  $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$  се нарекува и *оптимална* или *оптимално допустливо решение*.

Аналогно за минимизација на функцијата на целта (3), допустливото решение или програмата  $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$  е *минимална* ако

$$c_1\bar{x}_1 + c_2\bar{x}_2 + \dots + c_n\bar{x}_n \leq c_1x_1 + c_2x_2 + \dots + c_nx_n$$

за секое друго допустливо решение  $(x_1, x_2, \dots, x_n)$ .

Множеството од сите допустливи решенија или програми се нарекува *допустлива област* на задачата на линеарното програмирање. Со други зборови, да се реши задача на линеарното програмирање, значи да се најде оптимално решение или да се покаже отсуство на (конечна) оптимална програма или да се покаже дека задачата нема допустливи решенија. Задачата е *недопустлива*, ако нејзината допустлива област е празно множество, *решлива* ако постои (конечна) оптимална програма, а *нерешлива* ако постои програма, но не постои (конечна) оптимална програма.

За поедноставување на една ваква задача користиме скратени форми на запис ([1], [2]): матрична или векторска форма. Во овој труд ќе ја користиме матричната форма за претставување на задачата на линеарното програмирање (ЛП - задача). Матричниот облик на задачата (1) – (3) е

$$\begin{aligned} \max(\min) z &:= c^T x \\ Ax &\geq b \\ x &\geq 0 \end{aligned} \tag{4}$$

каде што

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix}, \quad o = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}, \quad c = \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{bmatrix}.$$

Задачата на линеарно програмирање може да ја претставиме и во следниот облик

$$\begin{aligned} \max(\min) z &:= c^T x \\ Ax &= b \\ x &\geq o \end{aligned} \tag{5}$$

каде што  $A, x, b, o$  и  $c$  се зададени во матрична форма како погоре.

Можеме да забележиме дека трансформацијата на една задача на линеарното програмирање од еден во друг облик може многу лесно да се направи согласно [1, 2, 5].

Основите на линеарното програмирање ги поставил советскиот математичар Леонид Канторович (Леонид Виталјевич Канторович, 1912 – 1986) во 1939 година, чии први модели се користени во дрвната индустрија. Во текот на Втората светска војна, Л. Канторович работел за советската војска во делот на воената логистика на оптимизација на проблеми поврзани со воените операции. Придонес во развојот на линеарното програмирање има и холандско-американскиот математичар и економист Тјалинг Чарлс Копманс (Tjalling Charles Kompan, 1910 – 1985). Канторович и Компанс во 1975 година ја делат Нобеловата награда за економија за нивната работа во линеарното програмирање.

Сепак, научниот свет за прв пат се запознава со решавање на практични проблеми со користење на линеарното програмирање во 1947 година преку работата на Џорџ Б. Данциг (George Bernard Dantzig, 1914 – 2005), кога го објавува симплекс алгоритмот, како значаен чекор за проширување на употребата на линеарното програмирање. Затоа, Џорџ Б. Данциг се смета за основоположник на симплекс методот ([1,2,3,5]).

Наједноставниот облик на задача на линеарното програмирање е задача на линеарното програмирање од две ненегативни променливи  $x_1$  и  $x_2$ . Таа се состои од функција на целта која треба да се минимизира

или максимизира од облик  $z = ax_1 + bx_2$ , каде што  $a$  и  $b$  се реални броеви коишто не можат да бидат истовремено нула. Притоа, променливите  $x_1$  и  $x_2$  се вклучени во ограничувањата во вид на систем од линеарни равенки или неравенки.

Во овој труд ќе се задржиме на решавање на конкретна задача со користење на симплекс методот согласно [1,2,3,5,7]. Решението на задачата ќе биде најдено според симплекс методот изложен во [5]. Развојот на различни ИТ алатки кај компјутерите и нивната достапност во училишта, овозможува добивање на брзо решение на математичките проблеми како и решавање на задачи и од областа на линеарното програмирање. Решението на задачата ќе биде презентирано во Excel и во математичкиот пакет Mathematica и ќе биде направена споредба помеѓу решавање на задачата со и без софтверски пакети за што ќе биде извлечен соодветен заклучок.

Трудот е наменет за надарени ученици од средно образование, со предзнаења од предметот Линеарна алгебра и аналитичка геометрија.

## 1. СИМПЛЕКС МЕТОД

Пред да се образложи симплекс методот, да воведеме некои поими, кои се среќаваат во натамошниот текст. Поимите ќе бидат воведени согласно [1].

Нека е даден системот линеарни равенки  $Ax = b$ , каде  $A$  е  $m \times n$  матрица и  $\text{rang}(A) = \text{rang}(A, b) = m < n$  (значи системот има бесконечно многу решенија). Нека  $B$  е подматрица на  $A$  составена од  $m$  линеарно независни колони (барем една таква подматрица постои бидејќи  $\text{rang}(A) = m$ ). Овие вектори образуваат база  $B$  на множеството вектор-колони на  $A$ . Ако со  $R$  се означи подматрицата образувана од колоните на  $A$ , коишто не се во базата, тогаш  $A$  може да се запише како блок матрица,  $A = [B \ R]$  (со соодветна смена на променливи, може да се доведе базата  $B$  да биде составена од првите  $m$  колони на матрицата  $A$ ). Ако со  $X_B$  се означи векторот образуван од компонентите на  $x$  соодветни на базните колони, а со  $X_R$  се означи векторот образуван од компонентите на  $x$  соодветни на останатите колони, тогаш системот  $Ax = b$  може да се претстави во облик

$$[B \ R] \begin{bmatrix} X_B \\ X_R \end{bmatrix} = b$$

каде што компонентите на  $X_B$  се нарекуваат *базни променливи*, а компонентите на  $X_R$  се *слободни променливи*. Ако на слободните променливи им се даде вредност нула, тогаш дадениот систем се сведува на системот  $BX_B=b$ , кој има едно единствено решение  $X_B=B^{-1}b$ , каде што  $B^{-1}$  е инверзна матрица на матрицата  $B$ . Решението на почетниот систем  $\begin{bmatrix} X_B \\ 0 \end{bmatrix}$  се нарекува *базно решение* соодветно на базата  $B$ .

Симплекс методот е итеративна постапка којашто овозможува насочено испитување на базните решенија со познато почетно базно решение. Во секој нареден чекор се наоѓа базно решение за кое функцијата на целта дава поголема или најмалку еднаква вредност (при максимизација) или функцијата на целта дава помала или најмногу еднаква вредност (при минимизација). После конечен број на чекори се добива оптимално решение или се добива сознание за неограниченост на функцијата на целта на допустливата област, а со тоа и непостоење на решението. Со други зборови, симплекс методот ни овозможува донесување оптимални одлуки во сложени ситуации.

За да се примени симплекс методот потребно е задачата на линеарното програмирање да биде во облик (5), при што матрицата  $A$  треба да содржи  $m$  базни колони и векторот  $b \geq 0$ . Симплекс методот кој тука ќе биде изложен е според [5].

Нека задачата на линеарното програмирање е дадена во облик (5) само за максимизација, т.е

$$\begin{aligned} \max z &:= c^T x \\ Ax &= b \\ x &\geq 0 \end{aligned} \tag{6}$$

Оптималното решение на задачата (6) е ненегативно решение  $x \geq 0$  кое го задоволува следното равенство

$$\begin{bmatrix} 1 & -c^T \\ 0 & A \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}, \tag{7}$$

каде што  $z$  е оптималната вредност на функцијата на целта. Нека  $B$  е база на матрицата  $A$  од ред  $m \times n$ ,  $x$  е векторот на основните променливи и  $C_B$  е вектор-колона составена од коефициентите на функцијата на целта пред базните променливи. Тогаш, (7) преминува во

$$\begin{bmatrix} 1 & C_B^T B^{-1} \\ 0 & B^{-1} \end{bmatrix} \begin{bmatrix} 1 & -c^T \\ 0 & A \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} = \begin{bmatrix} 1 & C_B^T B^{-1} \\ 0 & B^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ b \end{bmatrix}, \text{ т.е.}$$

$$\begin{bmatrix} 1 & C_B^T B^{-1} A - c^T \\ 0 & B^{-1} A \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} = \begin{bmatrix} C_B^T B^{-1} b \\ B^{-1} b \end{bmatrix}.$$

За  $P_j$ ,  $j$ -тиот вектор-колона на матрица  $A$  за променлива  $x_j$  и за  $X_B$ , вектор-колона составена од  $m$ -те базни променливи соодветни на базата  $B$ , се добива Табела 1:

Основни променливи	$x_j$	Решение
$z$	$C_B^T B^{-1} P_j - c_j$	$C_B^T B^{-1} b$
$X_B$	$B^{-1} P_j$	$B^{-1} b$

**Табела 1.** Симплекс табела поврзана со променлива  $x_j$  ([5]).

Задачите на линеарното програмирање во овој труд, ќе бидат решени со ревидираниот симплекс метод каде што итеративните чекори се исти како кај симплекс методот, само што направените пресметки се базирани на матрично оперирање согласно [5].

Алгоритмот на ревидираниот симплекс метод за решавање на задачата (7) ќе биде даден во чекори.

**Алгоритам. (Симплекс метод), [5].**

**Чекор 0:** За почетното базно решение  $X_B$ ,  $B$  и  $C_B$  се соодветно неговите придружени база и вектор на целта;

**Чекор 1:** Се пресметува инверзна матрица  $B^{-1}$  на матрицата  $B$ .

**Чекор 2:** За секоја променлива  $x_j$ , која преку  $P_j$  не е влезена во матрицата  $B$ , се пресметува  $z_j - c_j = C_B^T B^{-1} P_j - c_j$ . Ако  $z_j - c_j \geq 0$  за сите слободни променливи  $x_j$  тогаш се прекинува решавањето и оптималното решение е дадено со  $X_B = B^{-1} b, z = C_B^T X_B$  согласно Табела 1. Во спро-

тивно, се применува условот за оптималност, т.е. се одредува влезната променлива  $x_j$  со најмала негативна вредност  $z_j - c_j$ .

**Чекор 3:** Се пресметуваат  $B^{-1}P_j$ . Ако сите елементи  $B^{-1}P_j$  се негативни или нули тогаш постапката завршува (задачата е неограничена, односно функцијата на целта е неограничена). Во спротивен случај, се пресметува  $B^{-1}b$ . Тогаш за сите позитивни елементи на  $B^{-1}P_j$  се наоѓа најмалиот од количниците дефинирани со

$$\frac{(B^{-1}b)_r}{(B^{-1}P_j)_r} = \min_i \left\{ \frac{(B^{-1}b)_i}{(B^{-1}P_j)_i} \mid (B^{-1}P_j)_i > 0 \right\},$$

па променливата  $x_r$  поврзана со најмалиот количник е излезна променлива.

**Чекор 4:** Од постоечка база  $B$ , се формира нова база преку заменување на излезниот вектор  $P_r$  со влезниот вектор  $P_j$ . Се преминува на Чекор 1 за да отпочне нова итерација.

## 2. РЕШАВАЊЕ НА ЛП-ЗАДАЧА СО СИМПЛЕКС МЕТОД

Во овој дел, ние ќе дадеме еден проблем кој ќе го преточиме во задача на линеарно програмирање во облик (6).

**Задача.** Се изработуваат кошули и панталони во една конфекциска фабрика. Дадено е дека:

1. За да се изработи една кошула потребни се: 1 час сечење и 5 часови шиенење;
2. За изработка на еден пар панталони потребни се: 4 часови сечење и 3 часови шиенење;
3. Дневно, машината за сечење има капацитет за максимум 28 часови работа, а машината за шиенење има капацитет за максимум 55 часови работа (на пр., 4 работника работат по 7 часови дневно во одделот за сечење);
4. Бројот на панталони кои може да се произведат за еден ден е за два повеќе од бројот на кошули;
5. Фабриката од секоја кошула има 260 ден. добивка, а од секое поединечно парче панталони добивката е 570 ден.

Да се најде бројот на кошули и панталони кои ја максимизираат добивката. Колку изнесува максималната добивка (профит)?

**Решение:** Воведуваме променливи  $x_1$  - број на кошули и  $x_2$  - број на панталони. Според дадените услови за производство, го составуваме системот од ограничувачки неравенства

$$x_1 + 4x_2 \leq 28$$

$$5x_1 + 3x_2 \leq 55$$

$$x_2 - x_1 \leq 2$$

и ги додаваме условите за ненегативност (бидејќи станува збор за број на парчиња кои се произведени)

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Функцијата на целта која треба да се максимизира е

$$z_1 := 260x_1 + 570x_2.$$

Оваа функција на целта достигнува максимална вредност тогаш кога и функцијата на целта

$$z := 26x_1 + 57x_2$$

достигнува максимална вредност. Затоа, ЛП-задачата која треба да ја решиме е

$$\max z := 26x_1 + 57x_2$$

$$x_1 + 4x_2 \leq 28$$

$$5x_1 + 3x_2 \leq 55$$

$$x_2 - x_1 \leq 2$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Вака поставената задача на линеарно програмирање може да се реши и со геометриски пристап, [1,2,3,5].

Системот од ограничувања го трансформираме во облик (6) со воведување на ненегативни изедначувачки променливи  $x_3, x_4, x_5$ ,

$$x_1 + 4x_2 + x_3 = 28$$

$$5x_1 + 3x_2 + x_4 = 55$$

$$-x_1 + x_2 + x_5 = 2$$

$$x_j \geq 0, j = 1, 2, 3, 4, 5, 6.$$



Од системот ограничувачки равенства, матриците  $A$  и  $b$  имаат облик:

$$A = \begin{bmatrix} 1 & 4 & 1 & 0 & 0 \\ 5 & 3 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 28 \\ 55 \\ 2 \end{bmatrix}.$$

За поголема прегледност при пишувањето, матрицата  $A$  ја запишуваме како  $A = (P_1, P_2, P_3, P_4, P_5)$  каде што  $P_j, j=1,2,3,4,5$  е  $j$ -тиот вектор-колона на матрица  $A$ .

**Итерација 0:** За почетна база се зема матрицата  $B_0 = (P_3, P_4, P_5)$  и соодветниот вектор на коефициенти на целта е  $C_{B_0}^T = [0 \ 0 \ 0]$ . Па, почетното базно решение е

$$X_{B_0} = [x_3 \ x_4 \ x_5]^T, X_{B_0} = B_0^{-1}b = [28 \ 55 \ 2]^T,$$

а соодветната вредност на функцијата на целта е  $z = C_{B_0}^T X_{B_0} = 0$ .

Пресметката за оптималност:

$$C_{B_0}^T \cdot B_0^{-1} = [0 \ 0 \ 0]$$

$$[z_j - c_j]_{j=1,2} = C_{B_0}^T B_0^{-1}(P_1, P_2) - [c_1 \ c_2] = [-26 \ -57]$$

дава дека  $P_2$  е влезен вектор.

Од пресметката за можност

$$X_{B_0} = [28 \ 55 \ 2]^T$$

$$B_0^{-1} \cdot P_2 = [4 \ 3 \ 1]^T$$

$$\min\left\{\frac{28}{4}, \frac{55}{3}, \frac{2}{1}\right\} = \frac{2}{1}$$

гледаме дека на најмалиот количник му соодветствува променливата  $x_5$ , од каде што добиваме дека  $P_5$  е излезен вектор.

За поголема прегледност, горе добиените резултати ги претставуваме во Табела 2:

Основни променливи	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	Решение
$z$	-26	-57	0	0	0	0
$x_3$		4				28
$x_4$		3				55
$x_5$		1				2

Табела 2. Симплекс табела за Итерација 0.

**Итерација 1:** За база се зема  $B_1 = (P_3, P_4, P_2)$  и соодветниот вектор на коефициенти на целта е  $C_{B_1} = [0 \ 0 \ 57]$ . Тогаш, следното базно решение е  $X_{B_1} = [x_3 \ x_4 \ x_2]^T$ ,  $X_{B_1} = B_1^{-1}b = [20 \ 49 \ 2]^T$ , а соодветната вредност на функцијата на целта е  $z = C_{B_1}^T X_{B_1} = 114$ .

Пресметката за оптималност

$$C_{B_1}^T \cdot B_1^{-1} = [0 \ 0 \ 57]$$

$$[z_j - c_j]_{j=1,5} = C_{B_1}^T B_1^{-1}(P_1, P_5) - [c_1 \ c_5] = [-83 \ 57]$$

дава дека  $P_1$  е влезен вектор.

Од пресметката за можност

$$X_{B_1} = [20 \ 49 \ 2]^T$$

$$B_1^{-1} \cdot P_1 = [5 \ 8 \ -1]^T$$

$$\min\left\{\frac{20}{5}, \frac{49}{8}\right\} = \frac{20}{5}$$

гледаме дека на најмалиот количник му соодветствува променливата  $x_3$ , од каде добиваме дека  $P_3$  е излезен вектор.

Резултатите од првата итерација се дадени во Табела 3:

Основни променливи	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	Решение
$z$	-83	0	0	0	57	114
$x_3$	5					20
$x_4$	8					49
$x_2$	-1					2

Табела 3. Симплекс табела за Итерација 1.

**Итерација 2:** За база се зема  $B_2 = (P_1, P_4, P_2)$  и соодветниот вектор на коефициенти на целта е  $C_{B_2}^T = [26 \ 0 \ 57]$ . Тогаш, следното базно решение е  $X_{B_2} = [x_1 \ x_4 \ x_2]^T$ ,  $X_{B_2} = B_2^{-1}b = [4 \ 17 \ 6]^T$  а соодветната вредност на функцијата на целта е  $z = C_{B_2}^T X_{B_2} = 446$ .

Пресметката за оптималност

$$C_{B_2}^T \cdot B_2^{-1} = \begin{bmatrix} 83 & 0 & -47 \\ 5 & & 5 \end{bmatrix}$$

$$[z_j - c_j]_{j=3,5} = C_{B_2}^T B_2^{-1} (P_3, P_5) - [c_3 \quad c_5] = \begin{bmatrix} 83 & -47 \\ 5 & 5 \end{bmatrix}$$

дава дека  $P_5$  е влезен вектор.

Од пресметката за можност

$$X_{B_2} = [4 \quad 17 \quad 6]^T$$

$$B_2^{-1} \cdot P_5 = \begin{bmatrix} -\frac{4}{5} & \frac{17}{5} & \frac{1}{5} \end{bmatrix}^T$$

$$\min\{5, 30\} = 5$$

гледаме дека на најмалиот количник му соодветствува променливата  $x_4$ , од каде добиваме дека  $P_4$  е излезен вектор. Резултатите од втората итерација се дадени во Табела 4:

Основни променливи	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	Решение
$z$	0	0	83/5	0	-47/5	446
$x_1$					-4/5	4
$x_4$					17/5	17
$x_2$					1/5	6

Табела 4. Симплекс табела за Итерација 2.

**Итерација 3:** За базен програм база се зема  $B_3 = (P_1, P_5, P_2)$  и соодветниот вектор на коефициенти на целта е  $C_{B_3}^T = [26 \quad 0 \quad 57]$ . Тогаш, следното базно решение е  $X_{B_3} = [x_1 \quad x_5 \quad x_2]^T$ ,  $X_{B_3} = B_3^{-1}b = [8 \quad 5 \quad 5]^T$ , а соодветната вредност на функцијата на целта е  $z = C_{B_3}^T X_{B_3} = 493$ .

Пресметката за оптималност

$$C_{B_3}^T \cdot B_3^{-1} = \begin{bmatrix} 207 & 47 & 0 \\ 17 & 17 & \end{bmatrix}$$

$$[z_j - c_j]_{j=3,4} = C_{B_3}^T B_3^{-1} (P_3, P_4) - [c_3 \quad c_4] = \begin{bmatrix} 207 & 47 \\ 17 & 17 \end{bmatrix}$$

дава дека  $X_{B_3} = [x_1 \ x_5 \ x_2]^T = [8 \ 5 \ 5]^T$  е оптимален и овде пресметките завршуваат.

Резултатите од третата итерација се дадени во Табела 5:

Основни променливи	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	Решение
$z$	0	0	207/17	47/17	0	<b>493</b>
$x_1$						<b>8</b>
$x_5$						<b>5</b>
$x_2$						<b>5</b>

**Табела 5.** Симплекс табела за Итерација 3.

Од Табела 5 се гледа дека, решението на задачата на линеарното програмирање е  $x_1 = 8, x_2 = 5$  и за функцијата на целта е  $z = 493$ . Тогаш  $z_1 = 10z = 4930$ . Вредностите на изедначувачките променливи не се од интерес на задачата. Добивката се максимизира со производство на 8 кошули и 5 парчиња на пантолони со максималната добивка од 4930 ден.

**Забелешка 1.** Во Табелите 2, 3, 4 и 5 сите полиња не се пополнети. Ние ги пополнивме само полињата кои се од интерес на итерацијата. Како останатите полиња може да се пополнат, може да проверите со помош на интернет апликацијата [8], која нуди можност и за трансформација на внесената задача во облик погоден за примена на симплекс методот.

**Забелешка 2.** Покрај матричниот облик на симплекс методот кој беше изложен во овој дел, тој може да се дефинира и со помош на елементарни трансформации на ЛП-таблици, [4], [6]. На тој начин се избегнува директното барање на инверзна матрица.

### 3. РЕШАВАЊЕ НА ЛП-ЗАДАЧИ ВО EXCEL

Да го покажеме решавањето на задачата со симплекс методот во Excel. За да можеме задачата да ја решиме во Excel потребно е да ја имаме алатката Solver, која може да решава системи равенки и нера-

венки. Доколку не е активна оваа алатка, треба претходно да се активира.

### Чекори за решавање на задачата во Excel:

- Се отвора работен лист во Excel и се запишува текст во вид на коментар во полињата. Тие помагаат да се внесат точните информации (Слика 1).
- Се внесуваат следните формули:
  - Во полето E8 се запишува формулата  $=C5+4*E5$ ;
  - Во полето E9 се запишува формулата  $=5*C5+3*E5$ ;
  - Во полето E10 се запишува формулата  $=E5-C5$ ;
  - Во полето E13 се запишува формулата  $=26*C5+57*E5$ .
- Се избира Solver;
- Се селектира Max.
- Се додаваат функцијата на целта и ограничувањата според упатствата на Solver.
- Се одбира методот Simplex LP за решавање на проблемот. Се клика Solve.

Како изгледа решението на задачата на линеарно програмирање во Excel, пред и по кликањето на Solve, е дадено со Слика 1 и Слика 2, соодветно:

	A	B	C	D	E	F	G	H	I
1									
2		x1-број на кошули							
3		x2-број на панталони							
4									
5		x1		x2					
6									
7		Ограничувања							
8		x1+4x2<=28			<=		28		
9		5x1+3x2<=55			<=		55		
10		x2-x1<=2			<=		2		
11									
12									
13		Максимизира 26x1+57x2							
14									

Слика 1. Симплекс табела во Excel пред стартувањето на Solver.

	A	B	C	D	E	F	G	H	I
1									
2		x1-број на кошули							
3		x2-број на панталони							
4									
5		x1	8 x2		5				
6									
7		Ограничувања							
8		x1+4x2<=28			28 <=		28		
9		5x1+3x2<=55			55 <=		55		
10		x2-x1<=2			-3 <=		2		
11									
12									
13		Максимизира 26x1+57x2			493				
14									

Слика 2. Симплекс табела во Excel после стартувањето на Solver .

Од Слика 2 се гледа дека, се добива истото решение како и во решението на задачата без компјутер т.е.  $x_1=8$ ,  $x_2=5$  и  $z=493$ , од каде  $z_1 = 10z = 4930$ . Добивката се максимизира со производство на 8 кошули и 5 парчиња на панталони и максималната добивка изнесува 4930 денари.

**Забелешка 3.** За задачи кои се недопустливи, Excel дава одговор дека Solver не може да најде можно (возможно) решение. За задачи кои се нерешливи, дава одговор за неограничен проблем.

#### 4. РЕШАВАЊЕ НА ЛП-ЗАДАЧИ ВО MATHEMATICA

Решавањето на задача на линеарно програмирање во математичкиот пакет Mathematica може да биде на повеќе начини.

Со Mathematica, користејќи го симплекс методот можеме да ја решиме задачата табелирајќи ги резултатите. Така, Табелите 2, 3, 4 и 5 може да се добијат со Mathematica како излез на програмата дадена во [7]. Решението на задача на линеарно програмирање може да се добие и само со користење на една наредба. Една опција за решавање на задачата со Mathematica е даден на Слика 3:

```
(Debug) In[1]:=
  LinearProgramming[{-26, -57},
    {{1, 4}, {5, 3}, {-1, 1}, {1, 0}, {0, 1}},
    {{28, -1}, {55, -1}, {2, -1}, {0, 1}, {0, 1}},
    Method -> "Simplex"]

(Debug) Out[1]=
  {8, 5}

(Debug) In[2]:=
  f[x_, y_] := 26 * x + 57 * y

(Debug) In[3]:=
  z = f[8, 5]

(Debug) Out[3]=
  493
```

Слика 3. Симплекс метод во Mathematica.

Се забележува дека наредбата како излез ни нуди решение за променливите, додека оптималната вредност на функцијата на целта треба да ја пресметаме дополнително. Синтаксата на оваа наредба може да се најде на веб страната [9]. Друга опција е користење на наредбата Maximize. За нашата задача на линеарно програмирање решението во Mathematica со оваа наредба е претставено на Слика 4:

```
(Debug) In[4]:=
  Maximize[{26 x + 57 y, x + 4 y ≤ 28 && 5 x + 3 y ≤ 55 && y - x ≤ 2},
    {x, y}]

(Debug) Out[4]=
  {493, {x -> 8, y -> 5}}
```

Слика 4. Решение на ЛП-задачата во Mathematica.

Повторно, се добива истото решение на задачата, како и со користење на претходните два начини т.е.  $x_1=8$ ,  $x_2=5$  и  $z=493$ , па  $z_1 = 10z = 4930$ . Значи, добивката се максимизира со производство на 8 кошули и 5 парчиња на пантолони и максималната добивка е 4930 ден.

**Забелешка 4.** За задачи кои се недопустливи, Mathematica дава одговор дека, не може да се најде можно решение кое ги задоволува

дадените ограничувања. Додека за нерешливи задачи го добиваме одговорот дека проблемот е неограничен.

## 5. ЗАКЛУЧОК

Од претходно изложеното се гледа дека сите три начини даваат ист резултат. Во зависност од потребите, корисниците на симплекс методот може да изберат еден од горе наведените начини според следниве препораки:

– Пресметувањето без компјутер е комплексно и одзема многу време за да се дојде до посакуваното решение. Сепак, доколку е важно да се научи текот на постапката, итерациите и нивното прегледно табелирање, тогаш добивањето на решението без компјутер е незаменливо. Доколку сакаме само табелирање, тогаш тоа е овозможено со користење на програми кои можат да се најдат на библиотечната архива на Wolfram, [7]. Со користење на овие програми, може брзо и лесно да се дојде до решение;

– Доколку ни е важно само оптималното решение на задачата, тогаш Симплекс методот во Excel или во Mathematica е добар избор.

## ЛИТЕРАТУРА

- [1] Д. Л. Карчицка (во соработка со Д. П. Коробар), *Основи на линеарно програмирање*, Библиотека “Математичка школа”, Скопје, 1974.
- [2] Д. Л. Карчицка, М. Кон-Поповска, Д.К. Таневска “*Увод во линеарното програмирање*”, учебник за III година на образовниот профил статистичар-планер од математичко-информатичка струка, Просветно дело, Скопје, 1985.
- [3] И. Стојковска, *Линеарно програмирање (геометриски пристап)*, збир на статии од математичкото списание “Нумерус”, 2010-2011.
- [4] И. Стојковска, *Линеарно програмирање - теорија и методи* (интерна скрипта), Институт за математика, ПМФ, Скопје, 2012.



- [5] X. A. Таха, *Операциони истражувања*, Магор, Скопје 2010 (преводи на Владата на Р. Македонија).
- [6] V. Vujcic, M. Asic, N. Milicic, *Matematičko programiranje*, Matematički institut, Beograd, 1980.
- [7] A Simplex command for Mathematica and examples of its use, <http://library.wolfram.com/infocenter/MathSource/437/>
- [8] Simplex Calculator, <http://simplex.tode.cz/en/>
- [9] Linear Programming in Mathematica, <https://reference.wolfram.com/language/ref/LinearProgramming.html>

<sup>1</sup> Универзитет “Гоце Делчев”, Штип  
Факултет за Информатика  
ул. “Крсте Мисирков” бр. 10-А, Штип, Р. Македонија  
e-mail: [marija.101967@student.ugd.edu.mk](mailto:marija.101967@student.ugd.edu.mk),  
[biljana.zlatanovska@ugd.edu.mk](mailto:biljana.zlatanovska@ugd.edu.mk),

Примен: 30.01.2019

Поправен: 25.05.2019

Одобен: 27.05.2019

Објавен на интернет: 7.06.2019