УНИВЕРЗИТЕТ У НОВОМ САДУ
ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
ДЕПАРТМАН ЗА МАТЕМАТИКУ И
ИНФОРМАТИКУ

Александра Јовановић

Линеарно програмирање и анализа осетљивости

Мастер рад

Ментор:

др Сања Рапајић

Нови Сад, 2018

Садржај

1 Увод **4**

2 Линеарно програмирање **6**
2.1 Појам линеарног програмирања 6
2.2 Општи облик проблема линеарног програмирања и његова својства 8
2.2.1 Математички модел ЛП 8
2.2.2 Стандардни облик ЛП 9
2.2.3 Хиперраван 11
2.2.4 Графичка метода 14

3 Симплекс метода **17**
3.1 Услови решивости проблема линеарног програмирања . . . 17
3.2 Секундарни симплекс алгоритам 26
3.3 Примарни симплекс алгоритам 31
3.4 Дуалност у линеарном програмирању 36

4 Анализа осетљивости **42**
4.1 Основни појмови 43
4.2 Анализа осетљивости и програмски пакети 48
4.2.1 LINDO 48
4.2.2 LINGO 56
4.2.3 Solver 63
4.2.4 Matlab 69

5 Закључак **77**

6 Литература **78**

Велику захвалност, у првом реду, дугујем својој менторки др Сањи Рапајић за труд и велико залагање приликом реализације рада. Захваљујем се и осталим члановима комисије, као и свим професорима, који су ми предавали током студија. Велику захвалност, такође, дугујем колегама на послу, а посебно директору Леу Панићу, на огромној подршци и разумевању током овог периода. Посебно се захваљујем својим пријатељима, који су веровали у мене и проживљавали са мном све моје успоне и падове, а највећу заслугу за оно што сам постигла приписујем својој породици и момку, јер су увек били ту за мене и научили ме правим вредностима.

Нови Сад, 2018. *Александра Јовановић*

1 Увод

Линеарно програмирање је област математике која се бави специјалним типом проблема оптимизације унутар задатих ограничења. Многе одлуке менаџмента подразумевају покушај оптималног искоришћења ресурса организације. На пример произвођач жели да одреди како да искористи количине сировина уз највећи профит, пословођа како да распореди задани посао између својих запослених тако да посао буде одрађен у најкраћем могућем року. Циљ ових и сличних проблема јесте оптимизација, односно максимизација корисности и минимизација трошкова уз задата ограничења.

Подручје примене линеарног програмирања је широко: производња, транспорт и дистрибуција, маркетинг, телекомуникације, финансијско улагање, планирање итд, а формулисати реални проблем као проблем линеарног програмирања захтева тимски рад стручњака из више подручја.

Линеарно програмирање спада у једну од најзначајнијих и у пракси највише примењиваних метода операционих истраживања чији развој почиње тридесетих година 20. века када је Леонид Канторович уводи као методу решавања проблема планирања производње. У САД-у је линеарно програмирање развијено током Другог Светског рата за проблеме војне логистике, као што је оптимизација превоза војске и опреме конвоја.

Један од разлога распрострањене примене линеарног програмирања у пракси је и постојање ефикасних процедура за решавање проблема великих димензија. Међу њима је најпознатија Симплекс метода, коју је развио Данцинг (*Dancing*) 1947. Симплекс метода представља један универзалан алгоритам који се може применити за решавање разних облика проблема линеарног програмирања без обзира на број променљивих параметара који учествују у математичком моделу. У овом раду бавићемо се примарним и секундарним симплекс алгоритмом, као и анализом осетљивости, којом се може испитати како се мењају поједини показатељи оптималности у случају јединичних промена количине ресурса.

Временом су развијени одговарајући програмски пакети који имплементирају Симплекс методу. Наиме, тим софтверима добија се

оптимално решење проблема и вредност функције циља у њему, као и анализа осетљивости, којом се испитује до које границе се могу мењати улазни параметри проблема, а да то не утиче на структуру оптималног решења. У раду ће бити представљени неки програмски пакети за брзо решавање проблема линеарног програмирања великих димензија.

2 Линеарно програмирање

2.1 Појам линеарног програмирања

Оптимизација је научна област која одређује најбоље решење одређеног математички дефинисаног проблема.

Постоје различите методе оптимизације које се могу сврстати у групе према различитим критеријумима. Са становишта присуства ограничења постоје методе за:

(а) оптимизацију без ограничења и

(б) оптимизацију са ограничењима.

Општи облик проблема оптимизације се састоји од функције циља чија се оптимална вредност тражи и скупа ограничења који дефинише допустиви скуп решења. Дакле, у питању је оптимизација са ограничењима при чему је потребно наћи оптималну вредност функције циља на допустивом скупу.

Проблем линеарног програмирања (ЛП проблем) је проблем у коме је функција циља линеарна и сва ограничења су задата линеарним функцијама.

Дефиниција 1. *Функција $f(x_1, x_2, \ldots, x_n)$ је линеарна функција ако и само ако за произвољни скуп константи $c_1, c_2, \ldots, c_n$ функција f може да се запише на следећи начин*

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} c_i x_i = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n.$$

Дефиниција 2. *Проблем линеарног програмирања је оптимизациони проблем који се састоји у тражењу максималне или минималне вредности линеарне функције циља на неком одређеном скупу ограничења, такође задатим линеарним математичким изразима.*

Дефиниција 3. *Допустиво решење проблема ЛП је свако решење које припада допустивом скупу.*

Дефиниција 4. *Оптимално решење проблема ЛП је оно допустиво решење у коме функција циља достиже оптималну вредност.*

Проблем минимизације (максимизације) може се записати у облику

$$\min(\max) J(u)$$

$$u \in U$$

$$J \colon U \subseteq \mathbb{R}^n \to \mathbb{R},$$

где је функција циља $J(u)$ дефинисана на скупу $U \subseteq \mathbb{R}^n$, а скуп U се назива допустив скуп или скуп ограничења.

Код ЛП проблема функција циља $J(u)$ је линеарна и скуп U је задат линеарним ограничењима.

Поступак решавања проблема ЛП има 4 фазе:

1. дефинисање проблема,
2. формулација математичког модела који репрезентује проблем,
3. избор и примена методе за решавање проблема/модела,
4. доношење одлука.

Основни предуслов за решење проблема ЛП је прецизно дефинисање проблема. Циљеви морају јасно да се дефинишу како бисмо формулисали математички модел, што представља други корак у процесу решавања проблема ЛП. Математички модел се састоји од функције циља, чија се оптимална вредност тражи, и скупа ограничења, који дефинише допустив скуп решења. У зависности од облика математичког модела бирамо методу за решавање проблема ЛП, а применом одговарајуће методе долазимо до решења на основу кога се доноси одлука.

2.2 Општи облик проблема линеарног програмирања и његова својства

2.2.1 Математички модел ЛП

Општи облик проблема линеарног програмирања може се записати на следећи начин

$$\text{максимизирати (минимизирати) } J(u) = c_1u_1 + c_2u_2 + \cdots + c_nu_n$$

при ограничењима

$$\begin{gathered} a_{11}u_1 + a_{12}u_2 + \cdots + a_{1n}u_n \lesseqqgtr b_1 \\ a_{21}u_1 + a_{22}u_2 + \cdots + a_{2n}u_n \lesseqqgtr b_2 \\ \vdots \\ a_{m1}u_1 + a_{m2}u_2 + \cdots + a_{mn}u_n \lesseqqgtr b_m \\ u_1, u_2, \ldots, u_n \geq 0 \end{gathered}$$

или сажетије као

$$\begin{gathered} \max(\min) J(u) = \textstyle\sum_{j=1}^{n} c_j u_j \\ \textstyle\sum_{j=1}^{n} a_{ij}u_j \lesseqqgtr b_i, i = 1, 2, \ldots, m \\ u_j \geq 0, j = 1, 2, \ldots, n, \end{gathered}$$

где су a_{ij}, c_j, b_i, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ задати реални бројеви.

Функција $J(u)$ се назива функцијом циља, док се m ограничења називају једноставно ограничењима проблема линеарног програмирања. Област која поред ограничења ЛП обухвата и услове ненегативности назива се допустива област проблема линеарног програмирања.

Напомена: Користимо скраћенице за минимум и максимум, *min* и *max*.

2.2.2 Стандардни облик ЛП

Претпоставимо да имамо n променљивих, тада стандардни облик проблема линеарног програмирања изгледа овако

$$\max(\min) J(u) = c_1u_1 + c_2u_2 + \cdots + c_nu_n$$
$$a_{11}u_1 + a_{12}u_2 + \cdots + a_{1n}u_n = b_1$$
$$a_{21}u_1 + a_{22}u_2 + \cdots + a_{2n}u_n = b_2$$
$$\vdots$$
$$a_{m1}u_1 + a_{m2}u_2 + \cdots + a_{mn}u_n = b_m$$
$$u_1, u_2, \ldots, u_n \geq 0.$$

Ако дефинишемо

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \quad c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

онда стандардну форму можемо записати у матричном облику на следећи начин

$$\max(\min)\ c^T u$$
$$Au = b$$
$$u \geq 0,$$

где је $A = [a_{ij}]_{m\times n}$ матрица ограничења, $u = (u_1, u_2, \ldots, u_n)^T$ вектор колона са n променљивих чије се решење тражи у овом моделу и b_i је слободни члан i-тог ограничења, $i = 1, 2, \ldots, m$, а b је вектор ограничења.

Додатно ћемо претпоставити да је $b \geq 0$, тј. да су сви слободни чланови $b_i, i = 1, 2, \ldots, m$ ненегативни.

Дефиниција 5. *Нека су x,$y \in \mathbb{R}^n$ и нека је x=$(x_1, x_2, ..., x_n)^T$ и y=$(y_1, y_2, ..., y_n)^T$. Скаларни производ вектора x и y дефинише се на следећи начин*

$$\langle x, y\rangle \overset{\text{деф}}{=} x_1y_1 + x_2y_2 + ...x_ny_n.$$

2.2.2.1 Свођење на стандардни облик

Нека је дат проблем ЛП у облику

$$\begin{aligned} \min\ J(u) = \langle c, u\rangle = c^T u \\ Au \leq b \\ u \geq 0, \end{aligned} \qquad (1)$$

где је

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix},$$

$b = (b_1, b_2, ..., b_m)^T$, $u = (u_1, u_2, ..., u_n)^T$, $c = (c_1, c_2, ..., c_n)^T$.

Ако неједнакости у систему $Au \leq b$ допунимо до једнакости тако што i-тој неједначини додамо ненегативну променљиву $u_{n+i} \geq 0$, добијамо једначине

$$a_{i1}u_1 + a_{i2}u_2 + ... + a_{in}u_n + u_{n+i} = b_i, i = 1, 2, ..., m,$$

односно добијамо систем

$$Cx = b,$$

где је

$$x = (u_1, u_2, ..., u_n, u_{n+1}, ..., u_{n+m})^T,$$

$$C = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & 0 & 0 & \dots & 1 \end{bmatrix}.$$

Ако функцију циља дефинишемо са

$$J_k(x) = J(u) = \langle c, u\rangle = c_1u_1 + c_2u_2 + ... + c_nu_n + 0 \cdot u_{n+1} + ... + 0 \cdot u_{n+m}$$

добијамо проблем

$$\begin{aligned} &\min J_k(x) \\ &Cx = b \qquad (2) \\ &x \geq 0. \end{aligned}$$

Проблем (2) је стандардна форма проблема (1).

Теорема 2.1. *[11] Нека је дат проблем (1) и њему придружени проблем (2). Тада свако решење проблема (1) генерише решење проблема (2) и обрнуто, свако решење проблема (2) генерише решење проблема (1).*

Доказ. Ако је $u_* = (u_1, u_2, ..., u_n)^T$ решење проблема (1) оно задовољава систем једначина $Au_* + d = b$, за неко $d \geq 0$. Тада је $x_* = (u_1, u_2, ..., u_n, d_1, d_2, ..., d_m)^T$ решење проблема (2). Наиме, ако би постојало x из допустивог скупа проблема (2) тако да је $J_k(x) \leq J_k(x_*)$ следило би да је $J(\widetilde{x}) \leq J(u_*)$, где је $\widetilde{x} = (u_1, u_2, ..., u_n)^T$, а то би било супротно претпоставци да је u_* решење проблема (1).

Нека је $x_* = (u_1, u_2, ..., u_n, u_{n+1}, ..., u_{n+m})^T$ решење проблема (2). Како је $x_* \geq 0$ из $Cx_* = b$, следи

$$Au_* \leq b, u_* \geq 0,$$

где је $u_* = (u_1, u_2, ..., u_n)^T$. Пошто је функција циља иста у оба случаја закључујемо да је

$$J_k(x_*) = J(u_*).$$

Ако би постојало допустиво решење u проблема (1) за које је $J(u) < J(u_*)$, тада би компоненте тог решења допуњене са компонентама вектора $d = Au - b$ представљале нови вектор који припада допустивом скупу проблема (2), за који функција $J_k(x)$ добија мању вредност од $J_k(x_*)$, што представља контрадикцију. Дакле, u_* је решење проблема (1). $\square$

2.2.3 Хиперраван

Хиперраван дефинишемо на следећи начин

$$\Gamma = \{x \in \mathbb{R}^n, \langle c, x \rangle = \gamma\}, c \in \mathbb{R}^n, \gamma \in \mathbb{R}.$$

За $n = 1$ то је тачка, за $n = 2$ права, а за $n = 3$ раван.

Сада дефинишимо хиперполупросторе:

$$\Gamma^- = \{x \in \mathbb{R}^n, \langle c, x\rangle < \gamma\}, c \in \mathbb{R}^n, \gamma \in \mathbb{R},$$
$$\Gamma^+ = \{x \in \mathbb{R}^n, \langle c, x\rangle > \gamma\}, c \in \mathbb{R}^n, \gamma \in \mathbb{R},$$
$$\bar{\Gamma}^- = \{x \in \mathbb{R}^n, \langle c, x\rangle \leq \gamma\}, c \in \mathbb{R}^n, \gamma \in \mathbb{R},$$
$$\bar{\Gamma}^+ = \{x \in \mathbb{R}^n, \langle c, x\rangle \geq \gamma\}, c \in \mathbb{R}^n, \gamma \in \mathbb{R}.$$

Дефиниција 6. *За скуп U кажемо да је конвексан ако за свако $u, v \in U$ и $\alpha \in [0, 1]$ важи*

$$\alpha u + (1 - \alpha)v \in U.$$

Теорема 2.2. *Пресек конвексних скупова је конвексан скуп.*

Доказ. Нека су $U_i, i = 1, 2, ..., n$ конвексни скупови. Треба да покажемо да је скуп $\bigcap_{i=1}^n U_i$ конвексан скуп.

Претпоставимо да $u, v \in \bigcap_{i=1}^n U_i$. Из овог следи да $u, v \in U_i, \forall i$. Како су U_i конвексни скупови, онда следи $\alpha u + (1 - \alpha)v \in U_i, \forall i$. Даље је $\alpha u + (1 - \alpha)v \in \bigcap_{i=1}^n U_i$, што значи да је пресек конквексних скупова конвексан скуп. □

Теорема 2.3. $\Gamma, \Gamma^-, \Gamma^+, \bar{\Gamma}^-$ *и* $\bar{\Gamma}^+$ *су конвексни скупови.*

Доказ. Нека $x, y \in \Gamma$. Треба показати да за $\alpha \in [0, 1]$ важи

$$\alpha x + (1 - \alpha)y \in \Gamma.$$

Из $x, y \in \Gamma$ следи да је $\langle c, x\rangle = \gamma$ и $\langle c, y\rangle = \gamma$, па је

$$\langle c, \alpha x + (1 - \alpha)\, y\rangle = \alpha \langle c, x\rangle + (1 - \alpha) \langle c, y\rangle = \alpha\gamma + (1 - \alpha)\, \gamma = \gamma.$$

Дакле, из претходне једнакости следи $\alpha x + (1 - \alpha)\, y \in \Gamma$, што имплицира да је скуп Γ конвексан.

Аналогно се доказује да су скупови $\Gamma^-, \Gamma^+, \bar{\Gamma}^-$ и $\bar{\Gamma}^+$ конвексни. □

Теорема 2.4. *Скупови*

$$M_1 = \{x \in \mathbb{R}^n, Ax \leq b\}, A \in \mathbb{R}^{m,n}, b \in \mathbb{R}^m,$$
$$M_2 = \left\{x \in \mathbb{R}^n, \bar{A}x = \bar{b}\right\}, \bar{A} \in \mathbb{R}^{m,n}, \bar{b} \in \mathbb{R}^m,$$
$$M_3 = \{x \in \mathbb{R}^n, \alpha_i \leq x_i \leq \beta_i, i = 1, 2, \ldots, n\}.$$

су конвексни скупови.

Доказ. Нека $u, v \in M_1$, треба да покажемо: $\alpha u + (1-\alpha)v \in M_1$, $\alpha \in [0, 1]$.

Из $u, v \in M_1$ следи $Au \leq b$ и $Av \leq b$.

Како је

$$A(\alpha u + (1-\alpha)v) = \alpha Au + (1-\alpha)Av \leq \alpha b + (1-\alpha)b = b,$$

следи

$$\alpha u + (1-\alpha)v \in M_1,$$

што значи да је M_1 конвексан скуп.

Аналогно се доказује да су M_2 и M_3 конвексни скупови. □

Теорема 2.5. *Нека је* $U = \left\{u \in \mathbb{R}^n, Au \leq b, \bar{A}u = \bar{b}, u_i \geq 0, i \in I \subseteq \{1, 2, ..., n\}\right\}$, $A \in \mathbb{R}^{m,n}, b \in \mathbb{R}^m, \bar{A} \in \mathbb{R}^{s,n}, \bar{b} \in \mathbb{R}^s$. *Скуп* U *је конвексан.*

Доказ. Доказ следи директно на основу Теореме 2.2 и Теореме 2.4. □

Дефиниција 7. *Конвексни полиедар је непразан пресек неких затворених хиперполупростора, тј.*

$$K = \bigcap_{i=1}^{m} \overline{\Gamma}_i^- \neq \emptyset,$$

где је $\bar{\Gamma}_i^- = \{u \in \mathbb{R}^n, \langle a^i, u\rangle \leq b_i\}, i = 1, \ldots, m, a^i = (a_{i1}, \ldots, a_{in})^T, b_i \in \mathbb{R}$.

Дефиниција 8. v *је врх конвексног полиедра* $K = \bigcap_{i=1}^{m} \bar{\Gamma}_i^- \neq \emptyset$ *ако се налази у пресеку бар* n $(n \leq m)$ *дефиниционих хиперравни* Γ_i *које дефинишу полиедар, где је* $\Gamma_i = \{u \in \mathbb{R}^n, \langle a^i, u\rangle = b_i\}$, $i = 1, 2, \ldots, m$.

Теорема 2.6. *Линеарна функција циља над ограниченим конвексним полиедром (допустивим скупом) достиже екстремну вредност (максимум или минимум) у неком врху. Уколико она достиже свој минимум (максимум) у два или више врхова, онда она достиже минимум (максимум) у свакој конвексној комбинацији тих врхова. Уколико је конвексни полиедар неограничен, може се десити да линеарна функција циља нема екстремну вредност, односно да је неограничена, а ако има екстремну вредност, она се достиже у неком врху.*

Теорема 2.7 (Теорема о решивости проблема ЛП)**.** *Проблем ЛП нема решење ако је допустив скуп празан или ако је функција циља неограничена на допустивом скупу. У супротном ЛП проблем има решење и оно је или јединствено или има бесконачно много решења која представљају конвексну комбинацију врхова (решења).*

2.2.4 Графичка метода

Ако је у питању ЛП проблем са две променљиве он се може једноставно решити графичком методом. Дакле, један начин за решавање ЛП проблема је графичка метода, која ће бити илустрована на следећем примеру.

Пример 1. *Марија се бави грнчарством и прави шоље и тањире. Да би се направила једна шоља потребно је 6 минута, док је за тањир потребно само 3 минута. При прављењу шоље потроши се 75g глине, док се за тањир утроши 100g глине. Марија има 20 сати на располагању за израду свих производа и 25kg глине, а зарада која се оствари износи 2 EUR по свакој шољи и 1.5 EUR по тањиру. Она жели да зна колико шоља, односно тањира треба да направи како би остварила максималан профит.*

Решење:
Постављени циљ на који модел треба да одговори је: *Колико шоља, односно тањира треба Марија да направи да би остварила максималан профит?*

Означимо са x непознату количину шоља и са y непознату количину тањира које треба произвести, а са $J(x, y)$ означимо укупан профит од

продаје ових количина, што нам представља функцију циља.

Како су нам познате зараде по шољи и по тањиру функција циља формално се може приказати као

$$J(x, y) = 2x + 1.5y.$$

Скуп ограничења се састоји од ограничења времена и ограничења материјала. Што се тиче ограничења времена знамо да је време потребно да се направи једна шоља 6 минута, дакле да би направили x шоља потребно је $6x$ минута. Слично за тањире, потребно је $3y$ минута.
Обзиром да је укупно на располагању 20 сати, ово ограничење се може записати као

$$6x + 3y \leq 1200.$$

Аналогно, ограничење материјала може се приказати као

$$75x + 100y \leq 25000.$$

Осим тога, пошто количине производа не могу бити негативне, тј. број шоља и тањира не може бити мањи од нуле, важи $x \geq 0$ и $y \geq 0$.

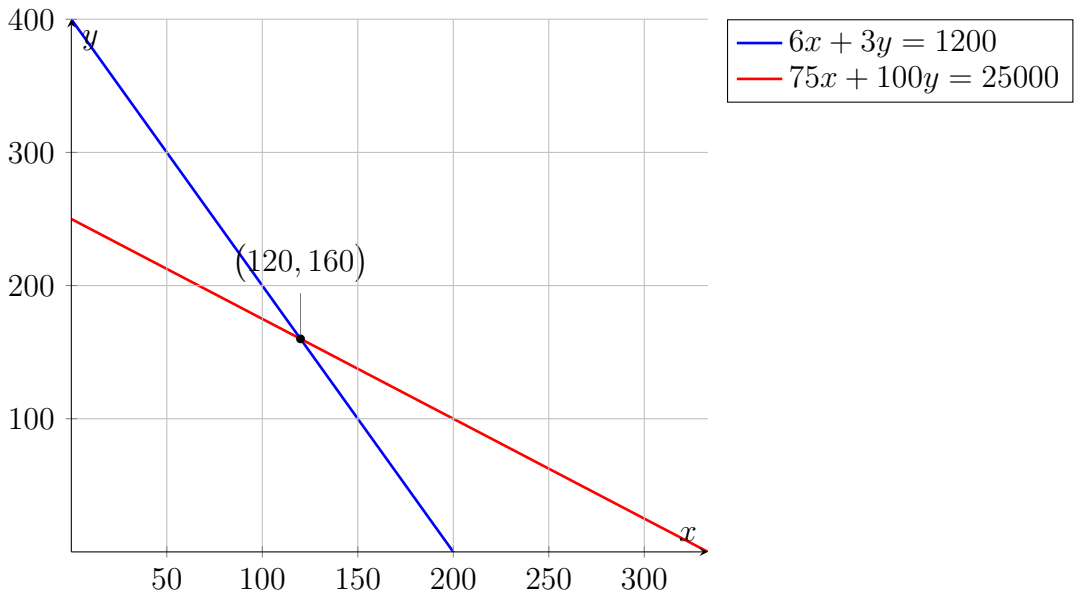
Сада се дефинисани проблем може моделирати као проблем ЛП на следећи начин

$$\max J(x, y) = 2x + 1.5y$$

$$6x + 3y \leq 1200$$
$$75x + 100y \leq 25000$$
$$x \geq 0, y \geq 0.$$

Као што је поменуто, овако једноставан проблем, односно проблем ЛП са две променљиве лако можемо решити графичком методом.

Прво цртамо ограничења како бисмо одредили допустиви скуп. Затим на том скупу тражимо решење у ком функција циља достиже максималну вредност.



Слика 2.1: Решење проблема из Примера 1 графичком методом

Јасно нам је да је допустив скуп решења пресек ограничења $6x + 3y \leq 1200$, $75x + 100y \leq 25000$, $x \geq 0$ и $y \geq 0$.

Врхови овог допустивог скупа су $(0, 0)$, $(200, 0)$, $(120, 160)$ и $(0, 250)$. Пошто је допустив скуп ограничен, на основу теореме о решивости проблема ЛП закључујемо да проблем има решење и управо тачка $(120, 160)$ је она у којој функција циља $J(x, y) = 2x + 1.5y$ достиже максимум, $J(120, 160) = 480$. Тачка максимума је обележена на слици и она представља врх допустивог скупа у коме се постиже максимум функције циља.

Решење нам говори да Марија треба да направи 120 шоља и 160 тањира како би остварила максималан профит од 480 EUR уз задата ограничења као што су време и материјал које има на располагању.

3 Симплекс метода

Симплекс метода је општа метода за решавање проблема ЛП и спада у итеративне методе. Алгоритам за решавање проблема се базира на томе да се полази од неког допустивог решења ЛП проблема, које се у низу корака побољшава док се не дође до оптималног решења. Битно је напоменути да уколико решење постоји, до њега се долази у коначном броју корака.
У овом поглављу су приказани секундарни и примарни симплекс алгоритми. Алгоритам који претпоставља познавање једног врха допустивог скупа је секундарни симплекс алгоритам, а онај који налази почетни врх за секундарни алгоритам или долази до закључка да проблем нема решења, односно да је скуп ограничења празан, је примарни симплекс алгоритам.

Најпре ћемо се упознати са основним дефиницијама и теоремама које ћемо користити за решавање проблема ЛП.

У будуће ћемо посматрати проблем минимизације, а проблем максимизације се лако може свести на проблем минимизације множењем функције циља са (-1).

3.1 Услови решивости проблема линеарног програмирања

Нека је дат проблем

$$\begin{gathered} \min J(u) = \langle c, u \rangle \,, \\ u \in U = \{u \in \mathbb{R}^n, u \geqslant 0, Au = b\} \,, \\ A \neq 0,\ A \in \mathbb{R}^{m,n},\ rang(A) = r,\ b \in \mathbb{R}^m, c \in \mathbb{R}^n. \end{gathered} \tag{3}$$

Дефиниција 9. *[11] Тачка v је* ***врх конвексног скупа*** *U ако је презентација*

$$v = \alpha u_1 + (1-\alpha)u_2, 0 < \alpha < 1, u_1, u_2 \in U$$

могућа само за $u_1 = u_2$.

Другим речима, тачка v је врх конвексног скупа U ако не постоје две различите тачке скупа U такве да је v њихова конвексна комбинација.

Дефиниција 10. *[11] Тачка v је* ***допустиво базично решење*** *проблема (3) ако постоји r линеарно независних вектора колона матрице A, $A_{j1}, A_{j2}, \ldots, A_{jr}$ тако да је*

$$v_{j1}A_{j1} + v_{j2}A_{j2} + \cdots + v_{jr}A_{jr} = b.$$

Дефиниција 11. *[11] Вектори $A_{j1}, A_{j2}, \ldots, A_{jr}$ су* ***вектори базе*** *допустивог базичног решења v. Матрица $B = (A_{j1}, A_{j2}, \ldots, A_{jr})$ је* ***база*** *допустивог базичног решења v.*

Дефиниција 12. *[11] Компоненте $v_{j1}, v_{j2}, \ldots, v_{jr}$ су* ***базичне компоненте*** *вектора v, а остале компоненте су* ***небазичне*** *и једнаке су нули.*

Дефиниција 13. *[11] Ако су све базичне компоненте вектора v позитивне, тада је v* ***недегенерисано*** *допустиво базично решење. Ако постоји нека базична компонента која је једнака нули онда је v* ***дегенерисано*** *допустиво базично решење.*

Дефиниција 14. *[11] Проблем (3) је* ***дегенерисан (сингуларан)*** *ако има бар једно дегенерисано допустиво базично решење. У супротном проблем је* ***недегенерисан (регуларан)****.*

Напомена: *У раду наводимо алгоритам Симплекс методе за проблеме без дегенерације.*

Теорема 3.1. *[11] Тачка v је врх скупа U проблема (3) ако и само ако је v допустиво базично решење проблема (3).*

Доказ. Показујемо прво смер ($\Rightarrow$). Нека је $rang(A) = r$. Показаћемо да позитивним компонентама врха v одговарају линеарно независни вектори. Без ограничења општости можемо претпоставити да су првих

k компоненти врха v позитивне, $v_i > 0, i = 1, 2, \ldots, k$ и $v_i = 0$, i=$k+1, k+2, \ldots n$. Тада је

$$\sum_{i=1}^{k} A_i v_i = b.$$

Претпоставимо супротно, да су вектори $A_i, i = 1, 2 \ldots, k$, линеарно зависни. Дакле, постоје бројеви $\beta_i, i = 1, 2, \ldots, k$, такви да је

$$\sum_{i=1}^{k} \beta_i A_i = 0,$$

а да при томе сви β_i нису једнаки нули. Помножимо $\sum_{i=1}^{k} \beta_i A_i = 0$ са $\epsilon > 0$ и саберимо то са $\sum_{i=1}^{k} A_i v_i = b$, а затим то поновимо са $-\epsilon$.

Добијамо

$$(v_1 + \epsilon\beta_1) A_1 + (v_2 + \epsilon\beta_2) A_2 + ... + (v_k + \epsilon\beta_k) A_k = b,$$

$$(v_1 - \epsilon\beta_1) A_1 + (v_2 - \epsilon\beta_2) A_2 + ... + (v_k - \epsilon\beta_k) A_k = b.$$

Уводећи ознаке

$$x = (v_1 + \epsilon\beta_1, v_2 + \epsilon\beta_2, ..., v_k + \epsilon\beta_k, 0, ..., 0)^T,$$

$$y = (v_1 - \epsilon\beta_1, v_2 - \epsilon\beta_2, ..., v_k - \epsilon\beta_k, 0, ..., 0)^T,$$

$x, y \in \mathbb{R}^n$, лако се види да је $Ax = b$ и $Ay = b$ односно, систем једначина у (3) има два решења x и y. Уз услов $v_i \pm \epsilon\beta_i \geq 0, i = 1, 2, ..., k$, та решења ће увек бити могућа решења проблема (3). За довољно мало ϵ важи да је $x \geq 0$,$y \geq 0$. Како је

$$v = \frac{1}{2}x + \frac{1}{2}y,$$

значи да је v конвексна комбинација две различите тачке из скупа U. Ово је немогуће ако је v врх скупа. На основу те противречности изводимо закључак да су вектори $A_i, i = 1, 2, ..., k$, линеарно независни, односно да је v базично решење. Пошто је $rang(A) = r$ тих вектора може бити највише r, $k \leq r$. Ако је $r = k$ допустиво базично решење v је недегенерисано, а ако је $k < r$ оно је дегенерисано.

Сада показујемо смер ($\Leftarrow$). Нека је $rang(A) = r$. Претпоставимо да је v

допустиво базично решење, а да није врх допустивог скупа. То значи да се v може написати као

$$v = \alpha x + (1 - \alpha)\, y,$$

где је $0 < \alpha < 1$, $x, y \in U$ и $x \neq y$. Пошто је $n - r$ компоненти тачке v једнако нули (небазичне компоненте), те исте компоненте биће једнаке нули и за таче x и y. Без умањења општости можемо узети да су последњих $n - r$ компоненти за све три тачке једнаке нули. Важи дакле

$$A_1 x_1 + A_2 x_2 + ... + A_r x_r = b,$$

$$A_1 y_1 + A_2 y_2 + ... + A_r y_r = b,$$

где су $A_i, i = 1, 2, ..., r$, линеарно независни вектори (A_i - вектори колоне матрице A). Ако одузмемо претходне једначине, добијамо

$$A_1 (x_1 - y_1) + A_2 (x_2 - y_2) + ... + A_r (x_r - y_r) = 0.$$

Због линеарне независности вектора следи да је $x_i - y_i, i = 1, 2, ..., r$, што је у супротности са претпоставком. Дакле, v је врх скупа U. □

Теорема 3.2. *[11] Уколико функција циља проблема (3) достиже минимум у тачки u_*, онда постоји врх v скупа U тако да је $J(u_*)$=$J(v)$.*

Доказ. Нека су $v_i, i = 1, 2, ..., k$, врхови скупа U. Тада је

$$u_* = \sum_{i=1}^{k} \lambda_i v_i, \sum_{i=1}^{k} \lambda_i = 1, \lambda_i \geq 0, i = 1, 2, ..., k,$$

па је

$$J(u_*) = \langle c, u_* \rangle = \langle c, \sum_{i=1}^{k} \lambda_i v_i \rangle = \sum_{i=1}^{k} \lambda_i \langle c, v_i \rangle \geq \langle c, v_p \rangle \sum_{i=1}^{k} \lambda_i = J(v_p)$$

где је $\langle c, v_p \rangle = \min_{1 \leq i \leq k} \langle c, v_i \rangle$. Како је $J(u_*)$ минимална вредност функције на U, следи $J(u_*) = J(v_p)$, тј. функција циља достиже минималну вредност у врху v_p. □

Теорема 3.3. *[11] Ако функција циља проблема (3) достиже минимум у $u_1^*, u_2^*, \ldots, u_k^*$, онда она достиже минимум у свакој конвексној комбинацији тих тачака.*

Доказ. Нека је $J(u_i) = m, i = 1, 2, ...k$ и

$$x = \sum_{i=1}^{k} \lambda_i u_i, \sum_{i=1}^{k} \lambda_i = 1, \lambda_i \geq 0, i = 1, 2, ...k.$$

Тада је

$$J(x) = \langle c, x \rangle = \langle c, \sum_{i=1}^{k} \lambda_i u_i \rangle = \sum_{i=1}^{k} \lambda_i \langle c, u_i \rangle = m \sum_{i=1}^{k} \lambda_i = m.$$

Дакле, за решавање проблема ЛП довољно је посматрати само врхове скупа ограничења (што ће бити приказано у Симплекс методи). □

Теорема 3.4. *[11] Нека је у проблему (3) $rang(A) = r = m$ и нека скуп U нема дегенерисаних врхова. Нека је v врх са базом $A_1, A_2, \ldots, A_r$, а*

$$A_j = \gamma_{1,j} A_1 + \gamma_{2,j} A_2 + ... + \gamma_{r,j} A_r, j = 1, 2, ..., n,$$

разлагање вектора A_j по векторима базе.

Означимо

$$Z_j = \gamma_{1,j} c_1 + \gamma_{2,j} c_2 + ... + \gamma_{r,j} c_r, j = 1, 2, ..., n.$$

Ако постоји $j = \{1, 2, ..., n\}$ тако да је $\Delta_j = Z_j - c_j > 0$, тада врх v није решење проблема (3).

Доказ. Пошто је v врх важи

$$v = (v_1, v_2, ..., v_r, 0, ..., 0)^T,$$

$$v_1 A_1 + v_2 A_2 + ... + v_r A_r = b,$$

$$J(v) = \langle v, c \rangle = c_1 v_1 + c_2 v_2 + ... + c_r v_r.$$

Показаћемо да постоји $x \in U$ тако да је $J(x) < J(v)$, тј. да v није решење проблема.

Ако помножимо $A_j = \gamma_{1,j}A_1 + \gamma_{2,j}A_2 + ... + \gamma_{r,j}A_r$ са $\epsilon > 0$ и то одузмемо од $v_1A_1 + v_2A_2 + ... + v_rA_r = b$ добија се

$$(v_1 - \epsilon\gamma_{1,j})\, A_1 + (v_2 - \epsilon\gamma_{2,j})\, A_2 + ... + (v_r - \epsilon\gamma_{r,j})\, A_r + \epsilon A_j = b.$$

На сличан начин из $Z_j = \gamma_{1,j}c_1 + \gamma_{2,j}c_2 + ... + \gamma_{r,j}c_r, j = 1, 2, ..., n$ и $J(v) = \langle v, c\rangle = c_1v_1 + c_2v_2 + ... + c_rv_r$ добија се

$$(v_1 - \epsilon\gamma_{1,j})\, c_1 + (v_2 - \epsilon\gamma_{2,j})\, c_2 + ... + (v_r - \epsilon\gamma_{r,j})\, c_r + \epsilon c_j = J(v) - \epsilon\,(Z_j - c_j)\,.$$

Нека је

$$x = (v_1 - \epsilon\gamma_{1,j}, v_2 - \epsilon\gamma_{2,j}, ..., v_r - \epsilon\gamma_{r,j}, 0, ..., 0, \epsilon, 0, ..., 0)^T \in \mathbb{R}^n,$$

где је ϵ на j-том месту, и нека је ϵ изабрано тако да је $x \geq 0$. Тада је x могуће решење проблема (3) и важи

$$J(x) = J(v) - \epsilon\,(Z_j - c_j) < J(v).$$

Дакле, v није оптимално решење, што је и требало доказати. □

Теорема 3.5. *[11] Нека је проблем (3) регуларан и нека је $rang(A)$=r=m. Нека је u_* допустиво базично решење. Тачка u_* ће бити оптимално решење проблема (3) ако је $\Delta_j \leq 0, j = 1, 2, ..., n$, где је Δ_j одређено у претходној теореми.*

Доказ. Следи на основу Теореме 3.4. □

Дефиниција 15. *[11] Величине Δ_j, дате у Теореми 3.4., називају се оцене врха v (допустивог базичног решења v).*

Теорема 3.6. *[11] Нека је дат врх v скупа U у проблему (3). Нека је $B = (A_1, A_2, ..., A_r)$ матрица базе врха v и $rang(B) = r = m$. Тада се базичне компоненте произвољне тачке $u \in U$ и вредност функције циља у тој тачки могу изразити преко њених небазичних компоненти и врха v на следећи начин*

$$\bar{u} = \bar{v} - \sum_{i \in P} B^{-1}A_iu_i,$$

$$J(u) = J(v) - \sum_{i \in P} \Delta_i u_i,$$

где је $\bar{u}$ вектор базичних компоненти тачке u, а P је скуп индекса небазичних компоненти.

Доказ. Можемо узети да је $B = (A_1, A_2, ..., A_r)$, па је онда $P = \{r+1, r+2, ..., n\}$.

Систем $Au = b$ се може записати у облику

$$B\bar{u} + A_{r+1}u_{r+1} + ... + A_n u_n = b,$$

где је $\bar{u} = (u_1, u_2, ..., u_r)^T$ вектор базичних компоненти тачке u. Множећи овај систем са B^{-1} (по дефиницији базе, матрица B је регуларна) добија се

$$B^{-1}Au = (E, G)\, u = \bar{u} + \sum_{i \in P} B^{-1}A_i u_i = B^{-1}b = \bar{v},$$

где је E јединична матрица, $G = \{\gamma_{i,j}\}_{i=1,j=r+1}^{r,n}$, а (E, G) матрица која се добије кад се испред матрице G допише јединична матрица одговарајућег формата. Заправо $\gamma_{i,k}$ је i-та компонента вектора $B^{-1}A_k, k \in P$. Дакле, систем $Au = b$ можемо записати као

$$(E, G)\, u = \bar{v},$$

јер је по дефиницији врха $\bar{v} = B^{-1}b$. Даље је

$$\bar{u} = \bar{v} - G\tilde{u},$$

где је $\tilde{u} = (u_{r+1}, u_{r+2}, ..., u_n)^T$, што даје

$$\bar{u} = \bar{v} - \sum_{i \in P} B^{-1}A_i u_i.$$

Ако у $J(u) = \langle \bar{c}, \bar{u} \rangle + \sum_{i=r+1}^{n} c_i u_i$ заменимо $\bar{u}$, добија се

$$J(u) = \langle \bar{c}, \bar{v} \rangle = J(v) - \sum_{i=r+1}^{n} \left(\left\langle \bar{c}, B^{-1}A_i \right\rangle - c_i \right) u_i = J(v) - \sum_{i=r+1}^{n} \Delta_i u_i,$$

где је

$$\Delta_i = Z_i - c_i, Z_i = \left\langle \bar{c}, B^{-1}A_i \right\rangle.$$

□

Теорема 3.7. *[11] Користећи*

$$\bar{u} = \bar{v} - \sum_{i \in P} B^{-1} A_i u_i,$$

$$J(u) = J(v) - \sum_{i \in P} \Delta_i u_i,$$

показати да важи:

1. *Ако је* $\Delta_i = \langle \bar{c}, B^{-1} A_i \rangle - c_i \leq 0, i = r+1, r+2, ..., n,$ *тада је врх* v *решење проблема (3).*

2. *Ако постоји* k*, тако да је* $\Delta_k > 0, \gamma_{i,k} \leq 0, i = 1, 2, ..., r,$ *тада проблем (3) нема решење, јер је функција циља неограничена са доње стране на скупу* U*.*

Доказ. 1. За произвољно $u \in U$ важи

$$J(u) = \langle \bar{c}, \bar{u} \rangle + \sum_{i=r+1}^{n} c_i u_i \geq \langle \bar{c}, \bar{u} \rangle + \sum_{i=r+1}^{n} \left\langle \bar{c}, B^{-1} A_i \right\rangle u_i.$$

Пошто је $\bar{u} = \bar{v} - \sum_{i \in P} B^{-1} A_i u_i$ следи

$$J(u) \geq \langle \bar{c}, \bar{u} + \sum_{i=r+1}^{n} B^{-1} A_i u_i \rangle = \langle \bar{c}, \bar{v} \rangle = J(v).$$

2. Нека је $B = (A_1, A_2, ...A_r)$. Према $\bar{u} = \bar{v} - \sum_{i \in P} B^{-1} A_i u_i$ је

$$u_i = v_i - \sum_{k=r+1}^{n} \gamma_{i,k} u_k, i = 1, 2, ..., r.$$

Нека је u тачка која има све набазичне компоненте једнаке нули осим компоненте u_k, тада

$$u_i = v_i - \gamma_{i,k} u_k, i = 1, 2, ..., r,$$

$$u_{r+1} = u_{r+2} = ... = u_{k-1} = u_{k+1} = u_{k+2} = ... = u_n = 0.$$

Ако су сви $\gamma_{i,k} \leq 0$, онда је $u \geq 0$. Систем $\bar{u} = \bar{v} - \sum_{i \in P} B^{-1} A_i u_i$ је еквивалентан са системом $Au = b$, јер је добијен из њега множењем

десне и леве стране са B^{-1}, где је B матрица базе, одакле следи $u \in U$.

На основу $J(u) = J(v) - \sum_{i \in P} \Delta_i u_i$ следи $J(u) = J(v) - \Delta_k u_k$. Пошто је $\Delta_k > 0$, ако пустимо да $u_k \to +\infty$, u остаје у скупу U и добијамо да је функција циља неограничена на скупу U ($J(u) \to -\infty$) тј. проблем нема решење. $\square$

Дакле, ако претпоставимо следеће:

1. скуп U проблема (3) нема дегенерисаних врхова и v је недегенерисани врх скупа U са базичним компонентама $v_1, v_2, \ldots, v_r$,

2. $A_1, A_2, \ldots, A_r$ чине базу врха v и $B = (A_1, A_2, \ldots, A_r)$ представља матрицу базе,

3. оцене врха v су дефинисане на следећи начин

$$\Delta_j = \left\langle \bar{c}, B^{-1} A_j \right\rangle - c_j, j = 1, 2, \ldots, n,$$

где је $\bar{c} = (c_1, c_2, \ldots, c_r)^T$,
$B^{-1} A_j = e_j, j = 1, 2, \ldots, r$, и
$B^{-1} A_j = (\gamma_{1,j}, \gamma_{2,j}, \ldots, \gamma_{r,j})^T, j = r+1, r+2, \ldots, n$,

тада важи:

(a) $\Delta_j = 0, j = 1, 2, \ldots, r$,

(b) Ако је $\Delta_j \leq 0, j = r+1, r+2, \ldots, n$, онда је врх v оптимално решење проблема (3),

(c) Ако је $\Delta_j < 0, j = r+1, r+2, \ldots, n$, онда је врх v јединствено оптимално решење проблема (3),

(d) Ако је $\Delta_j \leq 0, j = r+1, r+2, \ldots, n$, и ако $\exists k \in \{r+1, r+2, \ldots, n\}$ тако да је $\Delta_k = 0$, онда проблем (3) има бесконачно много решења,

(e) Ако $\exists k \in \{r+1, r+2, \ldots, n\}$ тако да је $\Delta_k > 0$, онда врх v није оптимално решење проблема (3),

(f) Ако $\exists k \in \{r+1, r+2, \ldots, n\}$ тако да је $\Delta_k > 0$ и $\gamma_{i,k} \leq 0$, $i = 1, 2, \ldots, r$, онда проблем (3) нема решења јер функција циља није ограничена са доње стране, тј. $J(u) \to -\infty$.

3.2 Секундарни симплекс алгоритам

Нека је дат ЛП проблем без дегенерације облика

$$\min J(u) = \sum_{j=1}^{n} c_j u_j$$

$$\begin{gathered} u_1 + \gamma_{1,r+1}u_{r+1} + \gamma_{1,r+2}u_{r+2} + ... + \gamma_{1,n}u_n = b_1 \\ u_2 + \gamma_{2,r+1}u_{r+1} + \gamma_{2,r+2}u_{r+2} + ... + \gamma_{2,n}u_n = b_2 \\ \vdots \\ u_r + \gamma_{r,r+1}u_{r+1} + \gamma_{r,r+2}u_{r+2} + ... + \gamma_{r,n}u_n = b_r \\ u_1, u_2, \ldots, u_n \geq 0 \\ b_1, b_2, ..., b_r > 0, \end{gathered}$$

или еквивалентно записано

$$\begin{gathered} \min \langle c, u \rangle, c, u \in \mathbb{R}^n \\ u \in U = \{u \in \mathbb{R}^n, u \geq 0, Au = b\}, b > 0, \end{gathered}$$

где је $b = (b_1, b_2, ..., b_r)^T$, $A \in \mathbb{R}^{m,n}$, $rang(A) = r$ и

$$A = \begin{bmatrix} 1 & 0 & \ldots & 0 & \gamma_{1,r+1} & \gamma_{1,r+2} & \cdots & \gamma_{1,n} \\ 0 & 1 & \ldots & 0 & \gamma_{2,r+1} & \gamma_{2,r+2} & \cdots & \gamma_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 & \gamma_{r,r+1} & \gamma_{r,r+2} & \cdots & \gamma_{r,n} \end{bmatrix}.$$

Тачка $v = (b_1, b_2, ..., b_r, 0, 0, \ldots, 0)^T \in \mathbb{R}^n$ је недегенерисан врх скупа $U = \{u \in \mathbb{R}^n, u \geq 0, Au = b\}$. Произвољна тачка $u \in U$ има базичне компоненте $u_1, u_2, ..., u_r$ и небазичне компоненте $u_{r+1}, u_{r+2}, ..., u_n$.

3.2.0.1 Алгоритам

Корак 1 Формирање симплекс табеле која одговара врху v (Табела 3.1),

	u_{r+1}	$\cdots$	u_{k-1}	u_k	u_{k+1}	$\cdots$	u_n	
u_1	$\gamma_{1,r+1}$	$\cdots$	$\gamma_{1,k-1}$	$\gamma_{1,k}$	$\gamma_{1,k+1}$	$\cdots$	$\gamma_{1,n}$	v_1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
u_{s-1}	$\gamma_{s-1,r+1}$	$\cdots$	$\gamma_{s-1,k-1}$	$\gamma_{s-1,k}$	$\gamma_{s-1,k+1}$	$\cdots$	$\gamma_{s-1,n}$	v_{s-1}
u_s	$\gamma_{s,r+1}$	$\cdots$	$\gamma_{s,k-1}$	$\gamma_{s,k}$	$\gamma_{s,k+1}$	$\cdots$	$\gamma_{s,n}$	v_s
u_{s+1}	$\gamma_{s+1,r+1}$	$\cdots$	$\gamma_{s+1,k-1}$	$\gamma_{s+1,k}$	$\gamma_{s+1,k+1}$	$\cdots$	$\gamma_{s+1,n}$	v_{s+1}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
u_r	$\gamma_{r,r+1}$	$\cdots$	$\gamma_{r,k-1}$	$\gamma_{r,k}$	$\gamma_{r,k+1}$	$\cdots$	$\gamma_{r,n}$	v_r
	Δ_{r+1}	$\cdots$	Δ_{k-1}	Δ_k	Δ_{k+1}	$\cdots$	Δ_n	$J(v)$

Табела 3.1: Симплекс табела која одговара врху v

где су оцене врха Δ_j одређене као у претходним теоремама. Прелазимо на следећи корак.

Корак 2 Уколико је $\Delta_j \leq 0$ за $j = r+1, r+2, ..., n$, онда је v оптимално решење проблема ($J(v) = J_*$). У противном прелазимо на *Корак 3*.

Корак 3 Одређујемо

$$\Delta_k = \max_{r+1 \leq j \leq n} \Delta_j$$

и прелазимо на следећи корак.

Корак 4 Посматрамо колону изнад Δ_k. Ако важи $\gamma_{i,k} \leq 0$ за све $i = 1, 2, ..., r$, онда проблем нема решења, тј. $J \to -\infty$. У супротном идемо на *Корак 5*.

Корак 5 Ако је $\gamma_{i,k} > 0$ за неко $i = 1, 2, ..., r$, онда тражимо пивот почетне симплекс табеле на следећи начин: пивот је оно $\gamma_{s,k}$ за које важи да

$$\frac{v_s}{\gamma_{s,k}} = \min_{i \in I_k} \frac{v_i}{\gamma_{i,k}}$$

где је $I_k = \{i, 1 \leq i \leq r, \gamma_{i,k} > 0\}$.

Корак 6 Елементарним базним трансформацијама на почетну симплекс табелу добија се нова симплекс табела која ће дефинисати нови врх w, за који важи $J(w) \leq J(v)$.

Променљива u_s излази из базе, а небазична компонента u_k улази у базу.

Дакле, произвољна тачка $u \in U$ ће имати следеће базичне компоненте

$$u_1, u_2, ..., u_{s-1}, u_k, u_{s+1}, ..., u_r,$$

и небазичне

$$u_{r+1}, u_{r+2}, ..., u_{k-1}, u_s, u_{k+1}, ..., u_n,$$

а табела која одговара врху w је

	u_{r+1}	$\cdots$	u_{k-1}	u_s	u_{k+1}	$\cdots$	u_n	
u_1	$\gamma'_{1,r+1}$	$\cdots$	$\gamma'_{1,k-1}$	$\gamma'_{1,s}$	$\gamma'_{1,k+1}$	$\cdots$	$\gamma'_{1,n}$	v'_1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
u_{s-1}	$\gamma'_{s-1,r+1}$	$\cdots$	$\gamma'_{s-1,k-1}$	$\gamma'_{s-1,s}$	$\gamma'_{s-1,k+1}$	$\cdots$	$\gamma'_{s-1,n}$	v'_{s-1}
u_k	$\gamma'_{k,r+1}$	$\cdots$	$\gamma'_{k,k-1}$	$\gamma'_{k,s}$	$\gamma'_{k,k+1}$	$\cdots$	$\gamma'_{k,n}$	v'_k
u_{s+1}	$\gamma'_{s+1,r+1}$	$\cdots$	$\gamma'_{s+1,k-1}$	$\gamma'_{s+1,s}$	$\gamma'_{s+1,k+1}$	$\cdots$	$\gamma'_{s+1,n}$	v'_{s+1}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
u_r	$\gamma'_{r,r+1}$	$\cdots$	$\gamma'_{r,k-1}$	$\gamma'_{r,s}$	$\gamma'_{r,k+1}$	$\cdots$	$\gamma'_{r,n}$	v'_r
	Δ'_{r+1}	$\cdots$	Δ'_{k-1}	Δ'_s	Δ'_{k+1}	$\cdots$	Δ'_n	$J(w)$

Табела 3.2: Симплекс табела која одговара врху w

Формирање нове табеле се изводи по следећим правилима:

1. u_s и u_k замене места,
2. пивот се трансформише у свој инверзни елемент

$$\gamma'_{k,s} = \frac{1}{\gamma_{s,k}},$$

3. преостали елементи у врсти којој припада пивот деле се са пивотом, тј.

$$\gamma'_{k,j} = \frac{\gamma_{s,j}}{\gamma_{s,k}}, j = r+1, ..., k-1, k+1, ...n,$$

$$v'_k = \frac{v_s}{\gamma_{s,k}},$$

4. преостали елементи у колони којој припада пивот деле се са пивотом и множе са (-1), тј.

$$\gamma_{i,s}^{'} = -\frac{\gamma_{i,k}}{\gamma_{s,k}}, i = 1, 2, ..., s-1, s{+}1, ...r,$$

$$\Delta_s^{'} = -\frac{\Delta_k}{\gamma_{s,k}},$$

5. преостали елементи у табели се трансформишу по правилу "правоугаоника", тј.

$$\gamma_{i,j}^{'} = \frac{\gamma_{i,j}\gamma_{s,k} - \gamma_{i,k}\gamma_{s,j}}{\gamma_{s,k}},$$

$$v_i^{'} = \frac{v_i\gamma_{s,k} - v_s\gamma_{i,k}}{\gamma_{s,k}},$$

$$\Delta_j^{'} = \frac{\Delta_j\gamma_{s,k} - \gamma_{s,j}\Delta_k}{\gamma_{s,k}},$$

$$j = r{+}1, ..., k-1, k{+}1, ...n,$$

$$i = 1, 2, ..., s-1, s{+}1, ..., r,$$

$$J(w) = \frac{J(v)\gamma_{s,k} - v_s\Delta_k}{\gamma_{s,k}},$$

6. w је нови врх, при чему је

$$w_i = v_i^{'}, i = 1, 2, ..., s-1, k, s{+}1, ..., r,$$

$$w_i = 0, i = r{+}1, ..., k-1, s, k{+}1, ...n.$$

Корак 7 Враћамо се на *Корак 2* и поступак се понавља са новом табелом.

Пример 2. *Решити следећи проблем*

$$\max x + 6y$$
$$x + y \leq 400$$
$$x \leq 200$$
$$y \leq 300$$
$$x, y \geq 0.$$

Решење: Да бисмо решили овај проблем морамо прво да га сведемо на облик такав да можемо применити секундарни симплекс алгоритам. Функцију циља множимо са (-1) да бисмо имали проблем минимизације, а ограничењима ћемо додати тзв. изравнавајуће променљиве. Еквивалентан проблем сада изгледа овако

$$\min -x - 6y$$
$$x + y + d_1 = 400$$
$$x + d_2 = 200$$
$$y + d_3 = 300$$
$$x, y, d_1, d_2, d_3 \geq 0.$$

Заправо, функција циља је $J(x, y, d_1, d_2, d_3) = -x - 6y + 0d_1 + 0d_2 + 0d_3$.

Почетно допустиво базично решење тј. врх допустивог скупа је $(0, 0, 400, 200, 300)^T$, а почетна вредност функције циља је 0.

Сада табеларно решавамо проблем пратећи кораке алгоритма. Прво формирамо симплекс табелу и питамо се да ли је она оптимална. Табела је оптимална ако су у последњем реду сви елементи $\leq$ 0. Како су сви позитивни бирамо највећи тј. тражимо $\max\{1, 6\}$ и посматрамо колону изнад тог броја. Дакле, добили смо да y улази у базу. Остаје да одредимо пивот и која променљива излази из базе, а то радимо поступком минимума количника који је објашњен у кораку 5 секундарног симплекс алгоритма. Тражимо $\min\left\{\frac{400}{1}, \frac{300}{1}\right\}$ и видимо да је пивот елемент 1, заокружен у табели. Сада елементарним базним трансформацијама долазимо до друге табеле и видимо да она није оптимална, аналогно настављамо и долазимо до треће табеле која има

све негативне елементе у последњем реду и закључујемо да је она оптимална.

Дакле, оптимално решење нашег проблема је врх са компонентама

$$x = 100,$$

$$y = 300,$$

а вредност функције циља у том врху износи 1900.

		x	y ↓	
	d_1	1	1	400
	d_2	1	0	200
←	d_3	0	1	300
	-J	1	6	0

		x ↓	d_3	
←	d_1	1	-1	100
	d_2	1	0	200
	y	0	-1	300
	-J	1	-6	-1800

	d_1	d_3	
x	1	-1	100
d_2	-1	1	100
y	0	-1	300
-J	-1	-5	-1900

Табела 3.3: Решење Примера 2 секундарним симплекс алгоритмом

3.3 Примарни симплекс алгоритам

Уколико је облик ЛП такав да се не може применити секундарни симплекс алгоритам јер се не зна почетни врх допустивог скупа, онда се прво примењује примарни симплекс алгоритам који одређује почетни врх v или долази до закључка да проблем нема решења јер је допустив скуп празан.

Нека је проблем ЛП задат у следећем облику

$$\begin{gathered}
\min J(u) = c_1u_1 + c_2u_2 + \cdots + c_nu_n \\
a_{11}u_1 + a_{12}u_2 + \cdots + a_{1n}u_n = b_1 \\
a_{21}u_1 + a_{22}u_2 + \cdots + a_{2n}u_n = b_2 \\
\vdots \\
a_{m1}u_1 + a_{m2}u_2 + \cdots + a_{mn}u_n = b_m \\
u_1, u_2, \ldots, u_n \geq 0,
\end{gathered}$$

или еквивалентно написано

$$J = \langle c, u \rangle \to \min$$

$$u \in U = \{u \in \mathbb{R}^n, u \geq 0, Au = b, A \in \mathbb{R}^{m,n}, b \in \mathbb{R}^m, b > 0\}.$$

3.3.0.1 Алгоритам

Корак 1 Правимо помоћни проблем који придружујемо нашем почетном проблему. Тај проблем се формира тако што се сваком ограничењу почетног проблема дода по једна вештачка променљива, а функција циља је збир тих вештачких променљивих, тј. придружени проблем је облика

$$\begin{gathered}\min J_1 = u_{n+1} + u_{n+2} + ... + u_{n+m} \\ a_{11}u_1 + a_{12}u_2 + \cdots + a_{1n}u_n + u_{n+1} = b_1 \\ a_{21}u_1 + a_{22}u_2 + \cdots + a_{2n}u_n + u_{n+2} = b_2 \\ \vdots \\ a_{m1}u_1 + a_{m2}u_2 + \cdots + a_{mn}u_n + u_{n+m} = b_m \\ u_1, u_2, \ldots, u_{n+m} \geq 0,\end{gathered}$$

или еквивалентно написано

$$\min J_1(z) = u_{n+1} + u_{n+2} + ... + u_{n+m}$$

$$z = (u_1, ..., u_n, u_{n+1}, ..., u_{n+m})^T \in \mathbb{R}^{n+m}, w = (u_{n+1}, ..., u_{n+m})^T$$

$$z \in Z = \left\{z \in \mathbb{R}^{n+m}, z = (u, w)^T, z \geq 0, Cz = b, C = (A, E) \in \mathbb{R}^{m,m+n}, b \in \mathbb{R}^m\right\},$$

$rang(C) = m$. Променљиве $u_{n+1}, u_{n+2}, ..., u_{n+m}$ су вештачке променљиве.

Корак 2 Овај придружени проблем решавамо секундарним симплексом. Нека је z_* решење помоћног проблема. Прелазимо на следећи корак.

Корак 3 Ако је $J_1(z_*) > 0$, онда почетни проблем нема решење, јер је тада допустив скуп почетног проблема празан. У супротном идемо на *Корак 4*.

Корак 4 Сада је $J_1(z_*) = 0$ и $z^* = (v, 0)^T, v = (v_1, v_2, ..., v_n)^T$, где је z_* врх скупа Z, па следи да је првих n компоненти врх скупа U, тј. v је врх скупа U.

Формирамо симплекс табелу за врх v на основу симплекс табеле врха z_*.

(a) У табели врха z_* избришемо последњу врсту.

(b) Посматрамо пресек врста које одговарају вештачким променљивима $u_{n+1}, u_{n+2}, ..., u_{n+m-r}$, које су у бази, и колона небазичних променљивих $u_{r+1}, u_{r+2}, ..., u_n$. Ако пресек не постоји онда идемо на (c), а уколико постоји идемо на (d).

(c) Како нема више вештачких променљивих у бази, база врха z^* се поклапа са базом врха v. Из табеле избацимо колоне које одговарају вештачким променљивим и испунимо последњу врсту величинама $\Delta_i = \sum_{s=1}^{r} c_s \gamma_{s,i} - c_i, i = r+1, r+2, ..., n$ и $J(v)$. Сада на то примењујемо секундарни симплекс алгоритам.

(d) Посматрамо пресек и уочавамо позитивни елемент $\gamma_{n+p,k}$, k=r+1, r+2, ..., n. Узимамо да нам је тај елемент пивот и примењујемо елементарне трансформације. Остајемо у истом врху, али ће се променити база. Из базе сада излази u_{n+p} и брише се колона која одговара вештачкој променљиви u_{n+p}, а улази у базу u_k. Поступак понављамо док не избацимо све вештачке променљиве из базе и тада идемо на (e) или понављамо док више не буде позитивних елемената у пресеку и тада идемо на (f).

(e) Враћамо се на (c), јер у бази нема више вештачких променљивих.

(f) Ако постоји врста у којој су сви елементи једнаки нули, онда је бришемо. После тога, посматраном пресеку одговарају само једначине код којих су сви коефицијенти негативни или нула. Променљиве које стоје уз те коефицијенте анулирамо, а колоне које одговарају анулираним променљивима се изостављају из даљег разматрања.

	u_1	u_2	$\dots$	u_r	u_{r+1}	$\dots$	u_n	u_{n+1}	$\dots$	u_{n+m-r}	$u_{n+m-r+1}$	$\dots$	u_{n+m}	
u_1	1	0	$\dots$	0	$\gamma_{1,r+1}$	$\dots$	$\gamma_{1,n}$	0	$\dots$	0	$\gamma_{1,n+m-r+1}$	$\dots$	$\gamma_{1,n+m}$	v_1^*
u_2	0	1	$\dots$	0	$\gamma_{2,r+1}$	$\dots$	$\gamma_{2,n}$	0	$\dots$	0	$\gamma_{2,n+m-r+1}$	$\dots$	$\gamma_{2,n+m}$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
u_r	0	0	$\dots$	1	$\gamma_{r,r+1}$	$\dots$	$\gamma_{r,n}$	0	$\dots$	0	$\gamma_{r,n+m-r+1}$	$\dots$	$\gamma_{r,n}$	v_r^*
u_{n+1}	0	0	$\dots$	0	$\gamma_{n+1,r+1}$	$\dots$	$\gamma_{n+1,n}$	1	$\dots$	0	$\gamma_{n+1,n+m-r+1}$	$\dots$	$\gamma_{n+1,n+m}$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
u_{n+m-r}	0	0	$\dots$	0	$\gamma_{n+m-r,r+1}$	$\dots$	$\gamma_{n+m-r,n}$	0	$\dots$	1	$\gamma_{n+m-r,n+m-r+1}$	$\dots$	$\gamma_{n+m-r,n+m}$	0
	Δ_1	Δ_2	$\dots$	Δ_r	Δ_{r+1}	$\dots$	Δ_n	Δ_{n+1}	$\dots$	Δ_{n+m-r}	$\Delta_{n+m-r+1}$	$\dots$	Δ_{n+m}	

Табела 3.4: Симплекс табела врха z_*

Корак 5 Применимо секундарни симплекс алгоритам почев од *Корака 2.* Када добијемо оптимално решење, на место анулираних променљивих ставимо нуле и тако добијемо $u_* = (u_1, u_2, ..., u_n)^T$ решење почетног проблема.

Примену примарног симплекс алгоритма показаћемо кроз најопштији облик линеарног програмирања.

Пример 3. *Решити следећи проблем*

$$\begin{gathered}\min J = x_1 + x_2\\ -x_1 \leq -1\\ x_1 - x_2 \geq 0\\ x_2 \leq 2\\ x_1, x_2 \geq 0.\end{gathered}$$

Решење:
Најпре морамо да средимо ограничења, прво ограничење помножимо са (-1) како би са десне стране неједнакости имали ненегативан члан. Затим, желимо да ограничења буду типа једнакости, како би проблем имао облик на који можемо да применимо алгоритам. Због тога, првом и другом ограничењу одузимамо вештачке променљиве, а трећем ограничењу додајемо вештачку променљиву. Сада проблем изгледа овако

$$\begin{gathered}\min J = x_1 + x_2\\ x_1 - d_1 = 1\\ x_1 - x_2 - d_2 = 0\\ x_2 + d_3 = 2\\ x_1, x_2, d_1, d_2, d_3 \geq 0,\end{gathered}$$

и тај проблем знамо да решимо помоћу примарног симплекс алгоритма.
Први корак је да проблему придружимо следећи помоћни проблем

$$\min J_1 = v_1 + v_2 + v_3$$
$$x_1 - d_1 + v_1 = 1$$
$$x_1 - x_2 - d_2 + v_2 = 0$$
$$x_2 + d_3 + v_3 = 2$$
$$x_1, x_2, d_1, d_2, d_3 \geq 0$$
$$v_1, v_2, v_3 \geq 0.$$

Табеларно решавамо проблем, тако што секундарним симплекс алгоритмом избацујемо вештачке променљиве из базе, с тим да, као што је речено у алгоритму, колону која прати ту променљиву бришемо. У четвртом кораку смо избацили последњу вештачку променљиву и добили смо да је табела оптимална по J_1 , али табела није оптимална по J, па зато настављамо са применом секундарног симплекс алгоритма како бисмо добили оптимално решење почетног проблема. Већ у наредном кораку добијамо оптималну табелу, па је оптимално решење $x_* = (1, 0)^T$, а функција циља у оптималном решењу је $J = 1$.

		x_1	x_2	d_1	d_2	d_3	
	v_1	1	0	-1	0	0	1
←	v_2	1	-1	0	-1	0	0
	v_3	0	1	0	0	1	2
	J_1	2	0	-1	-1	1	3
	J	-1	-1	0	0	0	0

		x_2	d_1	d_2	d_3	
←	v_1	1	-1	1	0	1
	x_1	-1	0	-1	0	0
	v_3	1	0	0	1	2
	J_1	2	-1	1	1	3
	J	-2	-2	-1	0	0

		d_1	d_2	d_3	
	x_2	-1	1	0	1
	x_1	-1	0	0	1
←	v_3	1	-1	1	1
	J_1	1	-1	1	1
	J	-4	1	0	2

		d_1	d_2	
←	x_2	-1	1	1
	x_1	-1	0	1
	d_3	1	-1	1
	J_1	0	0	0
	J	-4	1	2

	d_1	x_2	
d_2	-1	1	1
x_1	-1	0	1
d_3	0	1	2
J	-3	-1	1

Табела 3.5: Решење Примера 3 примарним симплекс алгоритмом

3.4 Дуалност у линеарном програмирању

Сваки проблем линеарног програмирања, тзв. примал, има свој дуални облик - дуал. Познавање релација између примала и дуала је од велике важности јер даје интересантне економске интерпретације код анализе осетљивости, што ћемо видети касније.

Нека је дат стандарни проблем максимума ЛП

$$\begin{gathered}\max J_p(u) = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m \\ x_1, x_2, \ldots, x_n \geq 0.\end{gathered} \tag{4}$$

Дуал ово проблема је

$$\begin{gathered}\min J_d(u) = b_1d_1 + b_2d_2 + \cdots + b_md_m \\ a_{11}d_1 + a_{21}d_2 + \cdots + a_{m1}d_m \geq c_1 \\ a_{12}d_1 + a_{22}d_2 + \cdots + a_{m2}d_m \geq c_2 \\ \vdots \\ a_{1n}d_1 + a_{2n}d_2 + \cdots + a_{mn}d_m \geq c_n \\ d_1, d_2, \ldots, d_m \geq 0.\end{gathered}$$

Матрична форма примарног и дуалног проблема ЛП

Примал:

$$\begin{aligned}&\max c^Tx \\ &Ax \leq b \\ &x \geq 0\end{aligned}$$

Дуал:

$$\begin{aligned}&\min d^Tb \\ &dA \geq c \\ &d \geq 0\end{aligned}$$

где су $A \in \mathbb{R}^{m,n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$ и $x \in \mathbb{R}^n, x = (x_1, x_2, \ldots, x_n)^T$ променљиве у прималу, а $d \in \mathbb{R}^m, d = (d_1, d_2, \ldots, d_m)^T$ променљиве у дуалу.

Нека је допустив скуп почетног проблема U=$\{x \in \mathbb{R}^n, x \geq 0, Ax \leq b\}$, а нека је V=$\{d \in \mathbb{R}^m, d \geq 0, dA \geq c\}$ допустив скуп дуалног проблема.

Теорема 3.8. *[11][Теорема слабе дуалности] Ако је x допустиво решење примала и d допустиво решење дуала, онда важи*

$$\langle c, x\rangle \leq \langle b, d\rangle, \forall x \in U, \forall d \in V.$$

Доказ. Како је x допустиво решење примала, онда важи $Ax \leq b$. Из овог следи

$$\langle d, Ax\rangle \leq \langle b, d\rangle .$$

Из друге претпоставке да је d допустиво решење дуала, тј. да је $dA \geq c$, следи

$$\langle x, dA\rangle \geq \langle c, x\rangle .$$

Даље је

$$\langle c, x\rangle \leq \langle x, dA\rangle = \left\langle A^T d, x\right\rangle = \langle d, Ax\rangle \leq \langle b, d\rangle ,$$

чиме је доказано тврђење. □

Теорема 3.9. *[11][Теорема јаке дуалности] За сваки пар примал - дуал важи једно од следећих тврђења:*

(a) Примал и дуал имају оптимално решење. Нека је x^* оптимално решење примала и d^* оптимално решење дуала, тада важи $\langle c, x^*\rangle = \langle b, d^*\rangle$ тј. $J_p(x^*) = J_d(d^*)$.

(b) Ни један проблем нема допустиво решење. Ако су допустиви скупови примала и дуала празни, тј. ако је $U = \varnothing$ и $V = \varnothing$, тада ни примал ни дуал немају оптимално решење.

(c) Један проблем нема допустиво решење, а други проблем има допустиво решење, али нема оптимално решење. Ако је допустив скуп једног од проблема празан, а функција циља није ограничена на другом проблему, тада ни примал ни дуал немају оптимално решење.

Теорема 3.10. *[11] Нека је x^* допустиво решење примала и d^* допустиво решење дуала. Ако важи $\langle c, x^*\rangle = \langle b, d^*\rangle$, онда је x^* оптимално решење примала и d^* оптимално решење дуала.*

Доказ. На основу теореме о слабој дуалности следи

$$\langle c, x\rangle \leq \langle b, d\rangle, \forall x \in U, \forall d \in V.$$

Одакле, даље следи

$$\langle c, x\rangle \leq \langle b, d^*\rangle = \langle c, x^*\rangle \Rightarrow \langle c, x\rangle \leq \langle c, x^*\rangle, \forall x \in U,$$

то значи да је x^* оптимално решење примала. Аналогно се показује да је d^* оптимално решење дуала. □

Теорема 3.11. *[11][Теорема слабе комплементарности] Нека је x^* допустиво решење примала и d^* допустиво решење дуала. Ако важе следећа два услова:*

(a) $\langle b - Ax^*, d^* \rangle = 0,$

(b) $\langle d^*A - c, x^* \rangle = 0,$

онда је x^* оптимално решење примала и d^* оптимално решење дуала.

Теорема 3.12. *[11][Теорема јаке комплементарности] Нека и примал и дуал имају бар једно допустиво решење, тада постоји бар један пар оптималних решења x^* и d^* тако да важи:*

(a) $b - Ax^* + d^* > 0,$

(b) $d^*A - c + x^* > 0.$

Теорема 3.13. *[11] За примални задатак у облику стандардног проблема максимума важи да је дуал његовог дуала полазни проблем примал.*

Доказ. Нека је дат проблем максимума (примал) и његов дуал.

Примал:

$$\begin{aligned} &\max c^T x \\ &Ax \leq b \\ &x \geq 0. \end{aligned}$$

Дуал:

$$\begin{aligned} &\min d^T b \\ &dA \geq c \\ &d \geq 0, \end{aligned}$$

где су $A \in \mathbb{R}^{m,n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$ и $x \in \mathbb{R}^n, x = (x_1, x_2, \ldots, x_n)^T$ променљиве у прималу, а $d \in \mathbb{R}^m, d = (d_1, d_2, \ldots, d_m)^T$ променљиве у дуалу.

Трансформишимо дуал у облику стандардног проблема максимума:

Дуал:

$$\begin{aligned} &\min d^T b \\ &dA \geq c \\ &d \geq 0, \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} &\max -d^T b \\ &-dA \leq -c \\ &d \geq 0. \end{aligned}$$

Одредимо сада дуал дуала.

Дуал дуала:

$$\begin{gathered}\min -c^T x \\ -Ax \geq -b \\ x \geq 0,\end{gathered}$$

Овај запис је еквивалентан полазном проблему тј. прималу. □

Пример 4. *Одредити дуал ако је примал следећег облика:*

$$\begin{gathered}\max J_p = 30x_1 + 20x_2 + 10x_3 \\ 8x_1 + 4x_2 + x_3 \leq 48 \\ 5x_1 + 3x_2 + 2x_3 \leq 31 \\ 3x_1 + 2x_2 + x_3 \leq 19 \\ x_2 \leq 3 \\ x_1, x_2, x_3 \geq 0.\end{gathered}$$

Решење:

$$\begin{gathered}\min J_d = 48d_1 + 31d_2 + 19d_3 + 3d_4 \\ 8d_1 + 5d_2 + 3d_3 \geq 30 \\ 4d_1 + 3d_2 + 2d_3 + d_4 \geq 20 \\ d_1 + 2d_2 + d_3 \geq 10 \\ d_1, d_2, d_3, d_4 \geq 0.\end{gathered}$$

3.4.0.1 Специфичан тип задатка ЛП

Специфичан тип задатка линеарног програмирања има следећи облик

$$\min \langle c, x \rangle$$

$$Ax \geq b$$

$$x \geq 0, b \geq 0, c \geq 0.$$

Овакав проблем је једноставније решити преко дуала.
Да бисмо одредили његов дуал најпре овом облику придружимо проблем облика проблема стандардног максимума

$$\max - \langle c, x \rangle$$
$$-Ax \leq -b$$
$$x \geq 0, b \geq 0, c \geq 0.$$

Овај облик је еквивалентан почетном облику. Сада одређујемо његов дуал

$$\min \langle -b, d \rangle$$
$$-dA \geq -c$$
$$d \geq 0, c \geq 0, b \geq 0.$$

Множењем функције циља и ограничења са (-1) добијамо крајњи облик дуала

$$\max \langle b, d \rangle$$
$$dA \leq c$$
$$d \geq 0, c \geq 0, b \geq 0.$$

Овај проблем се једноставно решава секундарним симплекс алгоритмом.

3.4.0.2 Правила за превођење најопштијег облика ЛП у дуални проблем

Многи реални проблеми се не могу написати у стандардном облику максимума (4). Често ограничења могу бити у облику једнакости, облику неједнакости са знаком $\geq$, такође променљиве не морају бити ненегативне (везане променљиве) као до сада, већ могу бити слободне (могу имати произвољну вредност из скупа $\mathbb{R}$). Према томе, сада наводимо правила која се користе за одређивање дуала најопштијег облика ЛП проблема.

1. Коефицијенти у функцији циља примала постају слободни чланови у ограничењима дуала, а слободни чланови у ограничењима примала постају коефицијенти у функцији циља дуала,

2. Ако је матрица A матрица ограничења примала, онда је матрица A^T матрица ограничења дуала,

3. Ако примал проблем максимума, онда је дуал проблем минимума, и обрнуто,

4. Сваком ограничењу типа неједнакости $\leq$ у прималу, одговара везана ненегативна променљива у дуалу,

5. Сваком ограничењу типа једнакости у прималу одговара слободна променљива у дуалу,

6. Свакој везаној променљиви у прималу одговара ограничење типа неједнакости са истим знаком у дуалу,

7. Свакој слободној променљиви у прималу одговара ограничење типа једнакости у дуалу.

Дакле, за најопштији облик примала, његов дуални проблема је

Примал:		Дуал:
$\max J_p(x) = \sum_{j=1}^{n} c_j x_j$		$\min J_d(d) = \sum_{i=1}^{m} b_i d_i$
$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, i = 1, 2, ..., k$		$d_i \geq 0, i = 1, 2, ..., k$
$\sum_{j=1}^{n} a_{ij} x_j = b_i, i = k{+}1, k{+}2, ..., m$	$\Rightarrow$	$d_i \in \mathbb{R}, i = k{+}1, k{+}2, ..., m$
$x_j \geq 0, j = 1, 2, ..., l$		$\sum_{i=1}^{m} a_{ij} d_i \geq c_j, j = 1, 2, ..., l$
$x_j \in \mathbb{R}, j = l{+}1, l{+}2, ..., n$		$\sum_{i=1}^{m} a_{ij} d_i = c_j, j = l{+}1, l{+}2, ..., n$

Пример 5. *Одредити дуал ако је примал следећег облика*

$$\begin{gathered}
\max J_p(x) = 4x_1 + 3x_2 \\
2x_1 + 2x_2 = 5 \\
2x_1 - x_2 \geq 3 \\
x_1 - x_2 \leq 1 \\
x_1 \geq 0 \\
x_2 \in \mathbb{R}.
\end{gathered}$$

Решење:

Примал:		Дуал:
$\max J_p(x) = 4x_1 + 3x_2$		$\min J_d(d) = -3d_1 + d_2 + d_3$
$-2x_1 + x_2 \leq -3$		$d_1 \geq 0$
$x_1 - x_2 \leq 1$	$\Rightarrow$	$d_2 \geq 0$
$2x_1 + 2x_2 = 5$		$d_3 \in \mathbb{R}$
$x_1 \geq 0$		$-2d_1 + d_2 + d_3 \geq 4$
$x_2 \in \mathbb{R}$		$d_1 - d_2 + 2d_3 = 3$

4 Анализа осетљивости

Проналажење оптималног решења за ЛП проблем је важно, али од велике важности је знати да ли се и како мења оптимално решење при промени улазних параметара. Тиме се бави анализа осетљивости, којом се испитује до које границе се могу мењати улазни параметри проблема, а да то не утиче на структуру оптималног решења, односно којом се може испитати како се мењају поједини показатељи оптималности у случају јединичних промена количине ресурса.

Како ЛП има широку примену у менаџменту при доношењу одлука које би довеле до остварења унапред дефинисаних циљева, анализа осетљивости пружа корисне информације. Да бисмо формулисали проблем ЛП морамо да знамо вредности улазних параметара, међутим често се дешава да менаџмент не зна прецизне вредности у тренутку када се модел формира или знају да се информације могу променити у будућности. Тада се користе апроксимације, које представљају само полазну тачку за даљу анализу.

Анализа осетљивости испитује утицај промене улазних параметара ЛП проблема на његово оптимално решење добијено симплекс методом, тј. задатак анализе осетљивости јесте да утврди колико је оптимално решење осетљиво на промене његових параметара. Промена параметара може да доведе до измене скупа базичних променљивих у оптималном решењу, а може да доведе до промене оптималних вредности променљивих и функције циља, при чему тада база остаје иста. Друга ситуација има значајну предност из разлога што се за одређивање нових вредности променљивих и функције циља може користити решење полазног проблема, тј. нема потребе поново решавати проблем, већ се једноставно примене правила за њихово израчунавање.

Основна улога анализе осетљивости је да за сваки улазни параметар одреди интервал у оквиру којег његова вредност може да се креће, а да се оптимална база проблема не промени. Она омогућава да се директно прецизно установи како се мењају подаци у оптималној симплекс табели када се промене неки улазни параметри почетног проблема.

Дакле, анализа осетљивости је систематска студија која говори о томе колико су осетљива решења на промене у подацима. Поставља

се питање како би промене коефицијената функције циља проблема ЛП или промене десних страна ограничења промениле оптимално решење.

4.1 Основни појмови

Као што је поменуто, анализа осетљивости бави се испитивањем утицаја промене параметара проблема ЛП на његово оптимално решење. Циљ анализе осетљивости је да утврди колико је модел осетљив на поменуте промене. Графичка метода се обично користи само за решавање проблема ЛП са две променљиве, па ћемо на наредном примеру илустровати анализу осетљивости графичком методом. Испитаћемо ефекте промене вредности улазних параметара на оптимално решење проблема.

Пример 6. *Нека је дат проблем*

$$\begin{gathered}\max J(x_1, x_2) = 4x_1 + 3x_2\\ 4x_1 + 2x_2 \leq 60\\ 2x_1 + 4x_2 \leq 48\\ x_1, x_2 \geq 0.\end{gathered}$$

(a) Решити проблем симплекс методом и графички.

(b) Пронаћи опсег оптималности коефицијената у функцији циља.

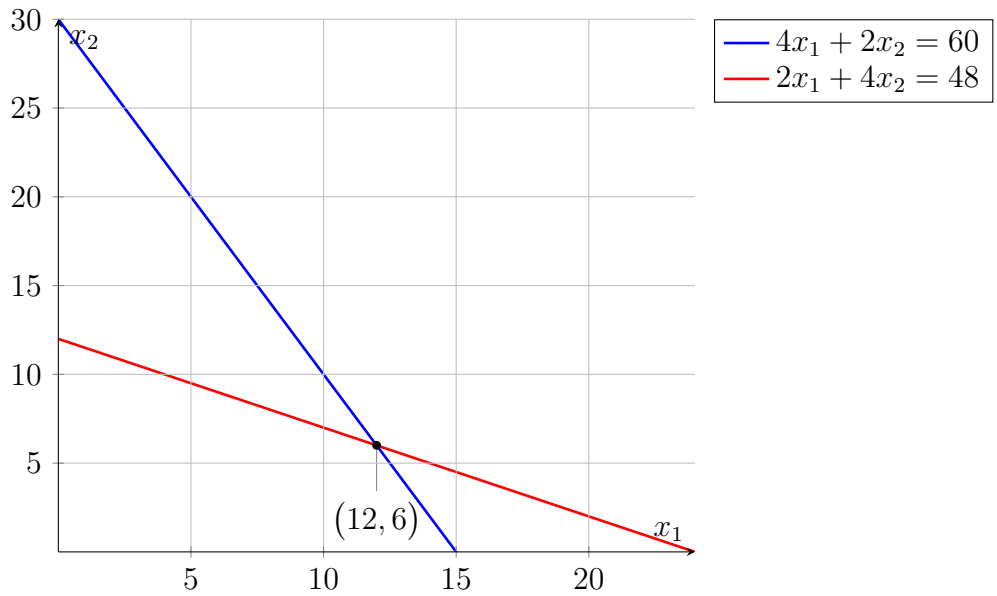
(c) Пронаћи опсег оптималности слободних чланова.

(d) Објаснити израз "Shadow Prices".

Решење

(a) **Графичка метода**:

Као што је раније објашњено, најпре се нацрта допустив скуп проблема. Како је он ограничен, на основу теореме о решивости проблема ЛП следи да се оптимално решење проблема налази у неком врху допустивог скупа.



Слика 4.1: Решење Примера 6 графичком методом

Оптимално решење датог проблема је врх (12,6), тј. $x_1 = 12$ и $x_2 = 6$, док је функција циља у оптималном решењу $J = 66$.

Симплекс метода:

$$
\begin{array}{c|c|cc|c}
 & & \downarrow & & \\
 & & x_1 & x_2 & \\
\hline
\leftarrow & d_1 & 4 & 2 & 60 \\
 & d_2 & 2 & 4 & 48 \\
\hline
 & -J & 4 & 3 & 0
\end{array}
\qquad
\begin{array}{c|c|cc|c}
 & & & \downarrow & \\
 & & d_1 & x_2 & \\
\hline
 & x_1 & \frac{1}{4} & \frac{1}{2} & 15 \\
\leftarrow & d_2 & -\frac{1}{2} & 3 & 18 \\
\hline
 & -J & -1 & 1 & -60
\end{array}
\qquad
\begin{array}{c|cc|c}
 & d_1 & d_2 & \\
\hline
x_1 & \frac{1}{3} & -\frac{1}{6} & 12 \\
x_2 & -\frac{1}{6} & \frac{1}{3} & 6 \\
\hline
-J & -\frac{5}{6} & -\frac{1}{3} & -66
\end{array}
$$

Табела 4.1: Решење Примера 6 добијено табеларно симплекс методом

Симплекс методом добијамо исто решење.

(b) Потребно је одредити интервале коефицијената у функцији циља тако да добијено оптимално решење проблема остане непромењено.

За проблем облика

$$\max J(x_1, x_2) = c_1x_1 + c_2x_2$$
$$a_{11}x_1 + a_{12}x_2 \leq b_1$$
$$a_{21}x_1 + a_{22}x_2 \leq b_2$$
$$x_1, x_2 \geq 0$$

интервали оптималности се добијају на следећи начин:

Корак 1 Одредити нагибе (коефицијенте праваца) за ограничења која дефинишу проблем. Нека су то вредности А и В, тј. $A = -\frac{a_{11}}{a_{12}}$ и $B = -\frac{a_{21}}{a_{22}}$.

Корак 2 Фиксирамо c_2, тј. доделимо му његову вредност из модела и онда решимо неједначине: $A \leq -\frac{c_1}{c_2} \leq B$ по c_1. На тај начин добија се интервал оптималности за c_1.

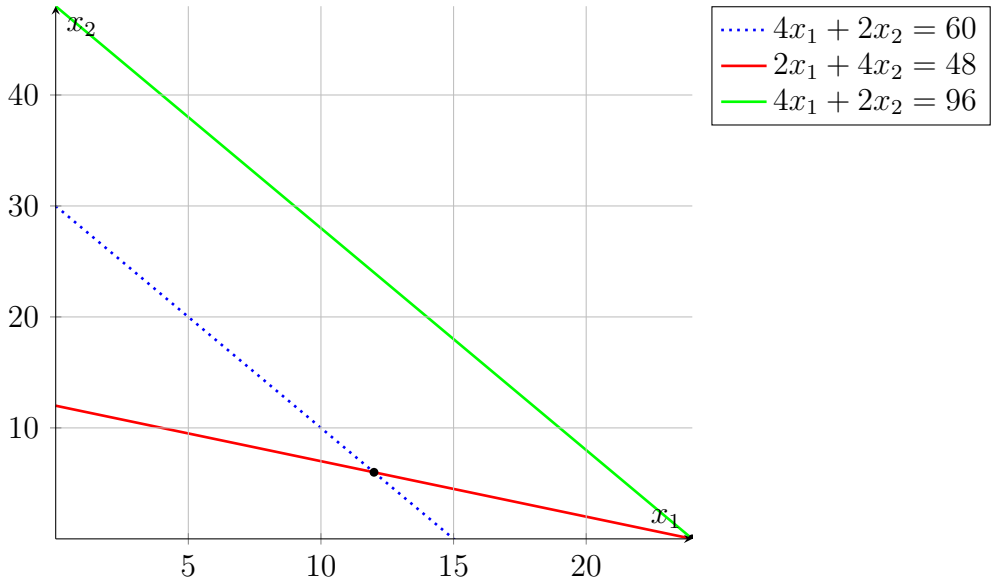
Корак 3 Фиксирамо c_1, тј. доделимо му његову вредност из модела и онда решимо неједначине: $A \leq -\frac{c_1}{c_2} \leq B$ по c_2. Овако се добија интервал оптималности за c_2.

Вратимо се на наш проблем. Фиксирајмо $c_2 = 3$ и одредимо нагиб функције циља проблема, као и нагибе ограничења датог проблема. Нагиб функције циља је $-\frac{c_1}{3}$, нагиб првог ограничења је $-\frac{4}{2}$ и нагиб другог ограничења износи $-\frac{2}{4}$. Ако је $-\frac{c_1}{3} < -\frac{4}{2}$, односно ако је $c_1 > 6$, онда текућа база више није оптимална. Такође, ако $-\frac{c_1}{3} > -\frac{2}{4}$, односно $c_1 < \frac{3}{2}$ текућа база неће бити више оптимална. Дакле, текућа база остаје оптимална за $\frac{3}{2} \leq c_1 \leq 6$.

Фиксирањем коефицијента који стоји уз прву променљиву у функцији циља, $c_1 = 4$, аналогним поступком, добијамо да оптимално решење проблема остаје исто ако се коефицијент c_2 креће у интервалу [2,8], тј. ако је $2 \leq c_2 \leq 8$.

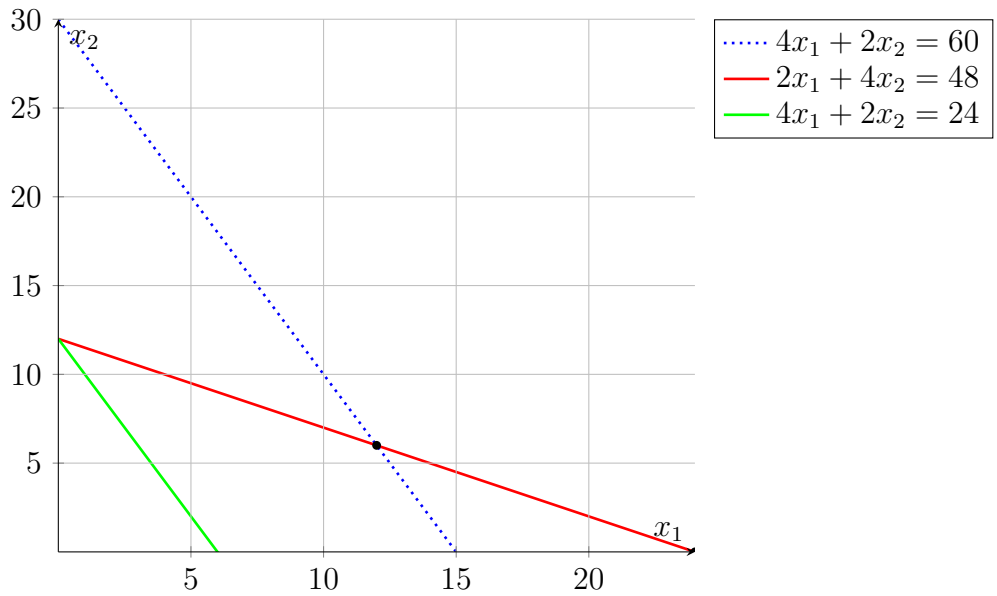
(c) Промену слободних чланова тј. промену десне стране ограничења (или познатије на енглеском као Right-Hand Side, скраћено RHS) показаћемо графички. Посматраћемо прво ограничење и анализирати шта ће се десити када дође до промене десне стране ограничења, односно када променимо b_1.

Промена b_1 помера границу првог ограничења паралелно његовој тренутној позицији. Текуће оптимално решење се налази у пресеку правих које су границе оба ограничења. Ако променимо b_1, докле год је тачка пресека граница оба ограничења допустива, она ће бити оптимално решење.



Слика 4.2: Решење Примера 6 графичком методом уз промену десне стране првог ограничења са 60 на 96

На слици 4.2 видимо да ако је $b_1 > 96$ ограничења ће се сећи у тачки која није допустива, текућа база више неће бити оптимална, па закључујемо да 96 представља горњу границу опсега оптималности слободног члана b_1.



Слика 4.3: Решење Примера 6 графичком методом уз промену десне стране првог ограничења са 60 на 24

Аналогно, уколико је $b_1 < 24$, пресек граница два ограничења неће бити допустива тачка, што значи да 24 представља доњу границу опсега оптималности слободног члана b_1.

(d) *Shadow Prices* се надовезују на промену десне стране ограничења, односно испитује се како промена у десној страни ограничења мења оптималну вредност фунције циља проблема ЛП. *Shadow Prices* за неко ограничење је вредност за коју се оптимална вредност функције циља побољшава (увећа ако је у питању проблем максимизације или смањи ако је у питању проблем минимизације), ако се десна страна ограничења (*RHS*) повећа за 1.
Напомена: Ова дефиниција се примењује ако се десна страна ограничења мења у опсегу оптималности решења, тј. ако промена десне стране ограничења оставља текућу базу оптималном.

Ако повећамо десну страну првог ограничења за Δ $(60+\Delta)$, претпостављајући да текућа база остаје оптимална, оптимално решење проблема ЛП је $x_1 = 12 + \frac{1}{3}\Delta$, $x_2 = 6 - \frac{1}{6}\Delta$. Тада је вредност функције циља у оптималном решењу

$$4x_1 + 3x_2 = 4(12 + \frac{1}{3}\Delta) + 3(6 - \frac{1}{6}\Delta) = 66 + \frac{5}{6}\Delta.$$

Дакле, докле год текућа база остаје оптимална, јединично повећање десне стране првог ограничења датог проблема ће повећати оптималну вредност функције циља за $\frac{5}{6}$, тј. за 0.833.

4.2 Анализа осетљивости и програмски пакети

Ако ЛП има више од две променљиве, опсег оптималности за десну страну ограничења и за коефицијенте функције циља не може се одредити графичком методом. Ови интервали се могу израчунати ручним прорачунима, али то често захтева доста времена. Осим тога, реални ЛП проблеми су углавном проблеми великих димензија, па их је решити ручно скоро немогуће. Због тога се користе одређени програмски пакети за решавање проблема ЛП. Временом су развијени многобројни софтвери који се користе у ту сврху, а неки од њих су бесплатни за коришћење.

У овом делу рада фокусираћемо се на примену програмских пакета за решавање ЛП проблема међу којима су најпознатији *LINDO*, *LINGO*, *Solver* и *Matlab*.

4.2.1 LINDO

LINDO (Linear Interactive and Descrete Optimizer) је софтверски алат развијен од стране *Linus Schrage* и намењен је за решавање проблема линеарног, квадратног и целобројног програмирања. Специфичне области примене, где се *LINDO* показао одличним, су производња, дистрибуција производа, управљање залихама итд. Програм се веома лако користи и разуме, и намењен је за решавање како једноставних ЛП проблема, тако и реалних проблема ЛП у индустрији са десетинама хиљада ограничења и стотине хиљада варијабли.

Синтакса модела захтева следеће:

1. функцију циља,
2. најмање једну променљиву и
3. најмање једно ограничење.

У прозор који се отвара при покретању програма уносимо проблем који треба да решимо. На пример ако је проблем облика

$$\min 2x + 3y$$
$$4x + 3y \leq 10$$
$$3x + 5y \leq 12$$
$$x, y \geq 0,$$

управо тако га уносимо и у програм

```
min  2x+3y
st
4x+3y<=10
3x+5y<=1.
```

Основне карактеристике програма:

1. назив промељиве не сме да садржи више од 8 карактера и мора почети словом, након почетног слова може бити било шта сем $!,), +, -, =, <$ и $>$,
2. *LINDO* препознаје следеће операторе: $+, -, =, <, >$ где је $<$ исто што и $\leq$ и $>$ исто што и $\geq$,
3. препоручује се да све операције унете у програм буду написане с лева на десно,
4. коментари могу бити било где у моделу након узвичника,
5. *LINDO* подразумева да су променљиве ненегативне, уколико му се не нагласи другачије ($x \in \mathbb{R} \Leftrightarrow$ FREE x, $x \in \mathbb{Z}_+ \Leftrightarrow$ GIN x).

Примену овог програмског пакета објаснићемо на следећем примеру.

Пример 7. *Омиљени доручак петочлане породице садржи хлеб, еурокрем и млеко. Цене намирница су следеће: килограм хлеба кошта 2.8 новчаних јединица, литра млека кошта 3.8, а килограм еурокрема 35.76 новчаних јединица.*

У следећој табели су приказане нутритивне вредности појединих намирница:

Намирнице	Угљени хидрати (g)	Беланчевине (g)	Ca (mg)	Fe (mg)	B_1 (mg)	B_2 (mg)
Хлеб	52,1	8,7	77	2,4	4,46	0,34
Млеко	4,7	3,3	119	-	0,04	0,16
Крем	26,7	12,5	6,4	79	0,05	0,35

Табела 4.2: Нутритивне вредности намирница које се користе за доручак

Минимална количина потребних састојака за доручак је приказана у следећој табели:

Угљени хидрати	Беланчевине	Ca	Fe	B_1	B_2
116,75	37,725	149,1	26,45	2,0775	0,7825

Табела 4.3: Минимална количина потребних састојака за доручак

Од различитих намирница које се користе за доручак треба одредити коју количину намирница треба купити тако да трошкови набавке буду најмањи, уз захтев да минималне количине потребних састојака буду задовољене. Задатак решити у LINDO-у.

Решење: На основу добијених информација у задатку, најпре ћемо поставити структуру оброка табеларно.

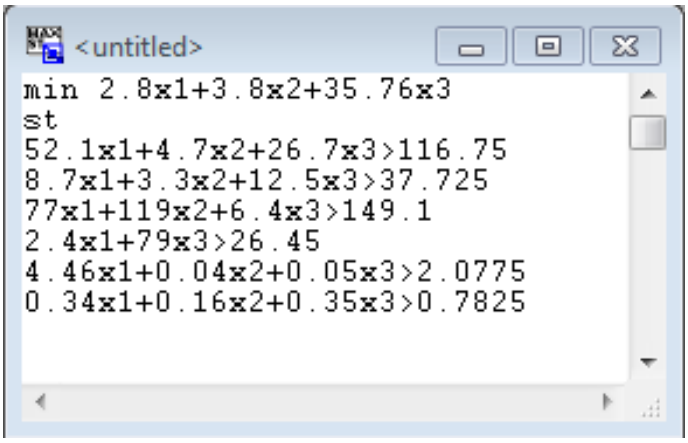
	Хлеб	Млеко	Крем	Мин. кол. састојака
Угљени хидрати	52,1	4,7	26,7	116,75
Беланчевине	8,7	3,3	12,5	37,725
Ca	77	119	6,4	149,1
Fe	2,4	-	79	26,45
B_1	4,46	0,04	0,05	2,0775
B_2	0,34	0,16	0,35	0,7825
Јединична цена	2,8	3,8	35,76	
Кол. артикла у оброку	x_1	x_2	x_3	

Табела 4.4: Структура оброка

Математички модел проблема има следећи облик

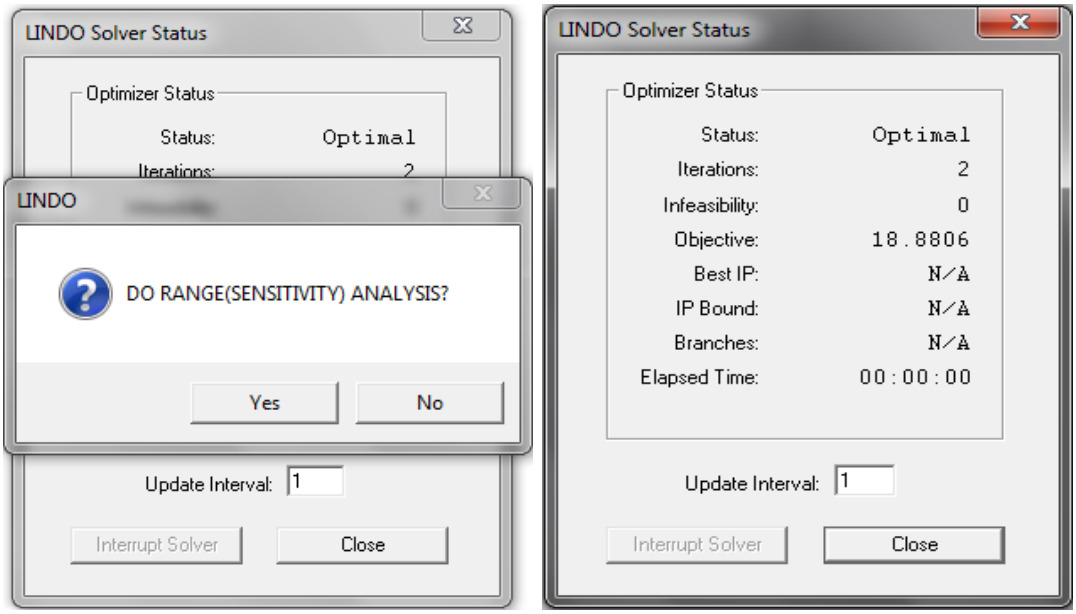
$$\min 2.8x_1 + 3.8x_2 + 35.76x_3$$
$$52.1x_1 + 4.7x_2 + 26.7x_3 \geq 116.75$$
$$8.7x_1 + 3.3x_2 + 12.5x_3 \geq 37.725$$
$$77x_1 + 119x_2 + 6.4x_3 \geq 149.1$$
$$2.4x_1 + 79x_3 \geq 26.45$$
$$4.46x_1 + 0.04x_2 + 0.05x_3 \geq 2.0775$$
$$0.34x_1 + 0.16x_2 + 0.35x_3 \geq 0.7825$$
$$x_1, x_2, x_3 \geq 0.$$

Уносимо модел у програм.



Слика 4.4: Унос проблема у *LINDO*

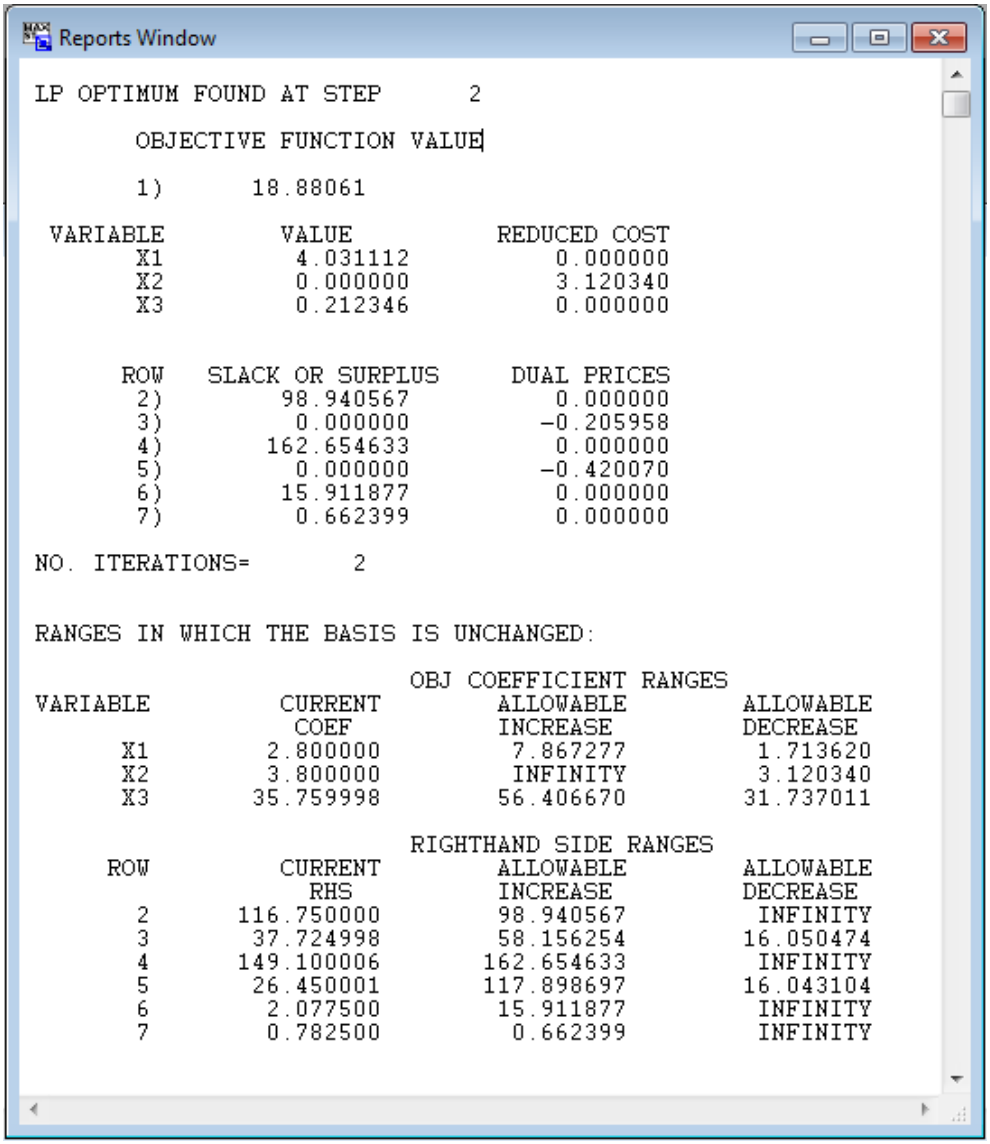
Одабиром опције *Solve* програм нам поставља питање да ли желимо анализу осетљивости (DO RANGE (SENSITIVITY) ANALYSIS?). Кликом на Yes отвара нам се прозор *Solver Status*.



Слика 4.5: Статус решења проблема из Примера 7

У том прозору видимо да је добијено оптимално решење у две итерације и да је вредност функције циља у оптималном решењу 18.8806.

Кликом на дугме *Close* отвара се решење у прозору Reports.



Reports Window

```
LP OPTIMUM FOUND AT STEP      2

        OBJECTIVE FUNCTION VALUE

        1)      18.88061

 VARIABLE        VALUE          REDUCED COST
       X1         4.031112          0.000000
       X2         0.000000          3.120340
       X3         0.212346          0.000000


      ROW   SLACK OR SURPLUS     DUAL PRICES
       2)        98.940567          0.000000
       3)         0.000000         -0.205958
       4)       162.654633          0.000000
       5)         0.000000         -0.420070
       6)        15.911877          0.000000
       7)         0.662399          0.000000

NO. ITERATIONS=       2


RANGES IN WHICH THE BASIS IS UNCHANGED:

                              OBJ COEFFICIENT RANGES
VARIABLE         CURRENT        ALLOWABLE        ALLOWABLE
                  COEF          INCREASE         DECREASE
      X1        2.800000        7.867277         1.713620
      X2        3.800000        INFINITY         3.120340
      X3       35.759998       56.406670        31.737011

                              RIGHTHAND SIDE RANGES
     ROW         CURRENT        ALLOWABLE        ALLOWABLE
                   RHS          INCREASE         DECREASE
       2      116.750000       98.940567         INFINITY
       3       37.724998       58.156254        16.050474
       4      149.100006      162.654633         INFINITY
       5       26.450001      117.898697        16.043104
       6        2.077500       15.911877         INFINITY
       7        0.782500        0.662399         INFINITY
```

Слика 4.6: Извештај о решењу проблема Примера 7, са анализом осетљивости у програму *LINDO*

У првом делу извештаја видимо поново вредност функције циља у оптималном решењу, као и оптимално решење (x_1, x_2, x_3)=$(4.031112, 0, 0.212346)$.

Из резултата се закључује да ће трошкови набавке, у износу од 18.88061 новчаних јединица, бити најмањи ако се за доручак искористи

само 4.031112 килограма хлеба, 0 литара млека и 0.212346 килограма еурокрема, што задовољава минималне количине потребних састојака.

Добијени резултати могу послужити као оријентир у анализи трошкова те помоћи у одређивању оптималног финансирања и на тај начин смањити трошкове и бацање хране свести на минимум.

REDUCED COST нам даје информацију о томе како ће промена коефицијената функције циља уз небазичну променљиву променити оптимално решење. REDUCED COST је за сваку базичну променљиву нула, а за небазичну променљиву представља износ за који ће оптимална вредност функције циља да се умањи, ако је у питању проблем максимума, односно увећа, ако је у питању проблем минимума, ако се за доручак искористи једна јединица посматраног производа.
Посматрано из другог угла, за било коју небазичну променљиву x_k, REDUCED COST је вредност за коју се коефицијент функције циља уз x_k мора смањити, ако је у питању проблем минимума, односно повећати ако је у питању проблем максимума, да би проблем ЛП имао оптимално решење у којем је x_k базична променљива. Наш пример је проблем минимума, па важи следеће: ако се коефицијент функције циља за небазичну променљиву x_k смањи за своју редуковану цену (REDUCED COST), тада ће проблем ЛП-а имати алтернативна оптимална решења - бар једно у коме је x_k базична променљива, и бар једно у коме x_k није базична променљива. Ако се коефицијент функције циља за небазичну променљиву x_k смањи за више од његове редуковане цене, тада било које оптимално решење проблема ЛП-а ће имати x_k као базичну променљиву и $x_k > 0$.
У нашем примеру добијамо за другу променљиву да REDUCED COST износи 3.120340, што значи да ће се за тај износ повећати функција циља уколико би укључили једну литру млека у доручак.

Напомена: Ако нека базична променљива има вредност нула у свакој од колона VALUE и REDUCED COST реч је о дегенерацији.

SLACK OR SURPLUS се односи на допунске променљиве, односно представља вредност допунских променљивих $d_1, d_2, d_3, d_4, d_5, d_6$. Закључујемо да су базичне променљиве d_1, d_3, d_5, а то значи да је искоришћено више од потребне минималне количине тих састојака. На пример, у реду 4) који се односи на треће ограничење имамо да је вредност 162.654633, што значи да је Ca у доручку за 162.654633 веће у односу на минималну потребну количину која треба да се у њему налази.

DUAL PRICES, или како се у литератури често називају SHADOW PRICES, за i-то ограничење представља износ за који би се вредност функције циља у оптималном решењу побољшала (тј. повећала у проблему максимизације, а смањила у проблему минимизације) ако се десна страна ограничења повећа за 1. На пример, ако у другом ограничењу десну страну повећамо за 1, функција циља ће се смањити за -0.205958, тј. повећати за 0.205958.
Напомена: Ограничење типа $\geq$ увек ће имати непозитивну вредност за DUAL PRICES, ограничење типа $\leq$ увек ће имати ненегативну вредност за DUAL PRICES,а вредност DUAL PRICES за ограничење типа једнакости може бити позитивна, негативна или нула.

За било које ограничење типа неједнакости производ вредности допунске променљиве и његове вредности за DUAL PRICES мора бити нула.

Сада прелазимо на други део који се зове RANGES IN WHICH THE BASIS IS UNCHANGED, тј. на интервале у којима база остаје непромењена. Прво се приказују интервали за коефицијенте функције циља. Видимо да за сваки коефицијент у продужетку имамо ALLOWABLE INCREASE, који представља износ за који максимално можемо да повећамо коефицијент и ALLOWABLE DECREASE, који представља износ за који највише можемо да га смањимо, а да база остане непромењена. INFINITY значи да не постоји доња или горња граница.

Аналогно објашење је и за десну страну ограничења (RIGHT-HAND-SIDE). За свако ограничење дат је опсег вредности за десну страну унутар којег текућа база остаје оптимална. Део ALLOWABLE INCREASE показује износ за који се слободан члан ограничења може повећати тако да текућа база остане оптимална. Слично, део ALLOWABLE DECREASE показује износ за који се слободан члан ограничења може смањити тако да текућа база остане оптимална.

За ограничење са позитивном допунском променљивом, њена вредност је повезана са опсегом вредности за десну страну тог ограничења унутар кога текућа база остаје оптимална (погледати табелу 4.5).

Тип ограничења	AI за десну страну	AD за десну страну
$\leq$	$= \infty$	=вредност за допунске променљиве
$\geq$	=вредност за допунске променљиве	$= \infty$

Табела 4.5: Опсег вредности за десну страну ограничења са позитивном допунском променљивом

Напомена: скраћеница AI је за ALLOWABLE INCREASE, а скраћеница AD за ALLOWABLE DECREASE.

Приметимо да за било које ограничење са позитивном допунском променљивом, оптимална вредност функције циља и вредности променљивих остају непромењене унутар опсега вредности за десну страну ограничења. На пример ако посматрамо треће ограничење, вредност његове допунске променљиве је 162.654633, па на основу табеле знамо да је опсег вредности десне стране ограничења (162.654633,∞). Ако се десна страна ограничења мења у том опсегу, текућа база остаје непромењена, као и функција циља у оптималном решењу.

4.2.2 LINGO

LINGO је интерактивни програмски пакет који се може користити за решавање проблема линеарног, целобројног и нелинеарног програмирања. Може се применити у сличним ситуацијама као и *LINDO*, разлика је у томе што *LINGO* пружа флексибилност у погледу изражавања модела.

За разлику од *LINDO*-а, *LINGO* дозвољава заграде и варијабле са десне стране ограничења, тј. ограничења се могу писати у оригиналном облику.

Програм обезбеђује велику библиотеку математичких и статистичких функција и већу могућност читања података из екстерних датотека и радних листова.

Као и *LINDO*, *LINGO* се може користити за интерактивно решавање проблема директним уносом са тастатуре или за решавање проблема помоћу датотека креираних на другим местима, на пример може бити део интегрисаног програма који садржи прилагођени k$\widehat{o}$д и *LINGO* библиотеке за оптимизацију. Ми ћемо се фокусирати на решавање проблема интерактивно.

Унос модела у *LINGO*-у је једноставан, упишу се подаци о моделу онако како бисмо га и ручно решавали.

LINGO прати синтаксу сличну синтакси *LINDO* програма. Потребно је имати функцију циља, једну или више променљивих и једно или више ограничења. Међутим, за разлику од *LINDO*-а, ограничења у *LINGO*-у не претходе никаквим посебним условима као што је *Subject to* или *Such that*, тј. скраћено *st*.

Карактеристике програма:

1. Изјаве се завршавају са ";",
2. *LINGO* располаже следећим математичким операторима: ^, *, /, +, -, <, >, =, <= и >=.
   Напомена: За означавање множења неопходна је звездица,
3. Заграде могу да се користе за одређивање редоследа математичких операција,
4. Назив променљиве може садржати до 32 карактера.

Наредни пример приказује примену програмског пакета *LINGO*.

Пример 8. *Један политичар покушава да победи на изборима. Његов округ има три различите врсте области: градске, приградске и сеоске. Ове области имају редом 100000, 200000 и 50000 регистрованих гласача. Да би он остварио успех, жели да освоји већину гласова у свакој од ове три области. Он је частан и никада не би подржао политику у коју не верује. Међутим, јасно му је да одређена питања могу имати веће ефекте у неким местима. Примарни циљеви су изградња путева, контрола наоружања, субвенционисање фарми и увођење таксе на гориво намењене побољшању јавног превоза. У складу са истраживањима може се проценити колико може гласова освојити или изгубути у свакој популацији уколико утроши 1000 EUR на промоцију сваког питања. Ове информације су дате у табели 4.5.*

Политика	**Град**	**Приградско насеље**	**Село**
Изградња путева	-2	5	3
Контрола наоружања	8	2	-5
Субвенционисање фарми	0	0	10
Таксе на гориво	10	0	-2

Табела 4.6: Ефекти политике на гласаче

У овој табели свако поље означава број гласова (у хиљадама) у градској, приградској и сеоској области који може бити освојен уколико се на промоцију одређеног питања уложи 1000 EUR. Негативне вредности означавају да ће гласови бити изгубљени. Задатак је да се одреди минимална количина новца коју он треба да потроши да би освојио 50000 градских, 100000 приградских и 25000 сеоских гласова. Проблем решити у LINGO-у.

Решење: Уводимо 4 променљиве:

- x_1 је број хиљада евра потрошених на промоцију изградње путева
- x_2 је број хиљаде евра потрошених на промоцију контроле наоружања
- x_3 је број хиљаде евра потрошених на промоцију субвенција за фарме
- x_4 је број хиљаде евра потрошених на промоцију такси на гориво

Захтев да освоји најмање 50000 градских гласова можемо записати као

$$-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50.$$

Слично, захтеви за освајањем најмање 100000 приградских гласова и 25000 сеоских гласова могу се записати као

$$5x_1 + 2x_2 + 0x_3 + 0x_4 \geq 100$$

и

$$3x_1 - 5x_2 + 10x_3 - 2x_4 \geq 25.$$

Било који скуп променљивих x_1, x_2, x_3 и x_4 који задовољава претходне неједнакости чини стратегију која ће освојити довољан број гласова у свакој од области. Да бисмо трошкове свели на минимум, желимо да минимизујемо израз

$$x_1 + x_2 + x_3 + x_4.$$

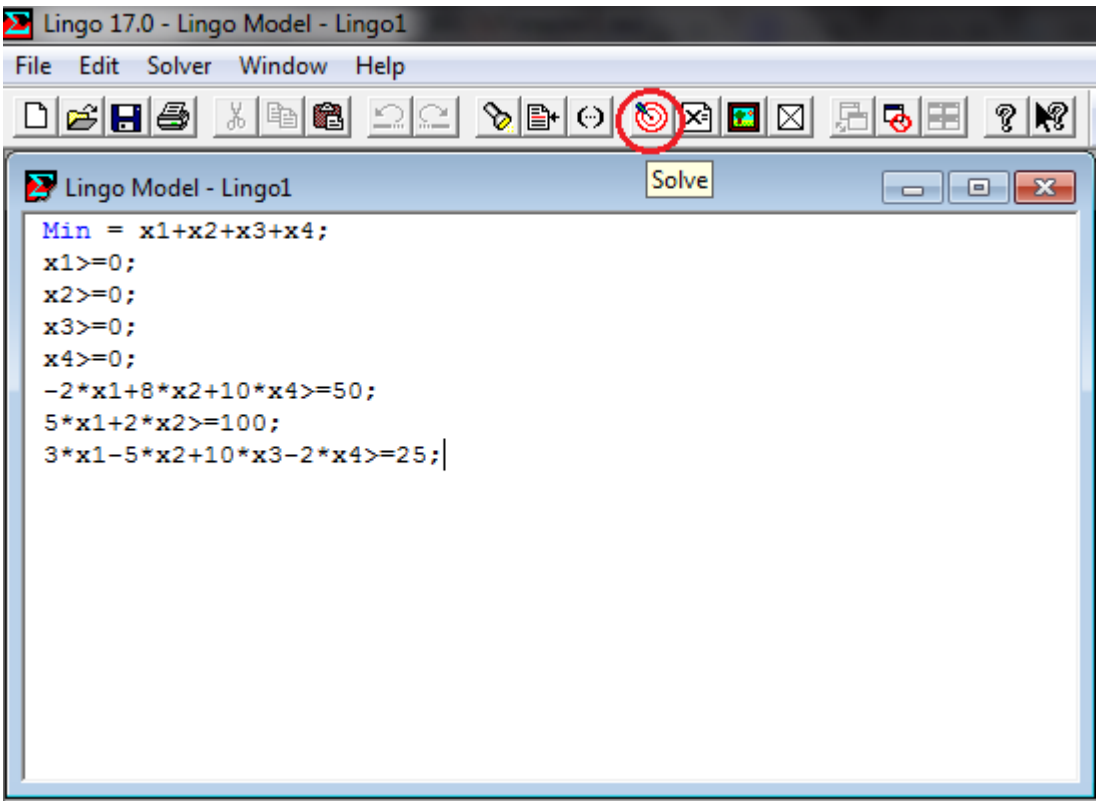
Иако је негативна реклама чест случај у политичким кампањама, негативни трошкови на рекламу нису могући. Из тог разлога захтевамо да

$$x_i \geq 0, i = 1, 2, 3, 4$$

Овај проблем записујемо као

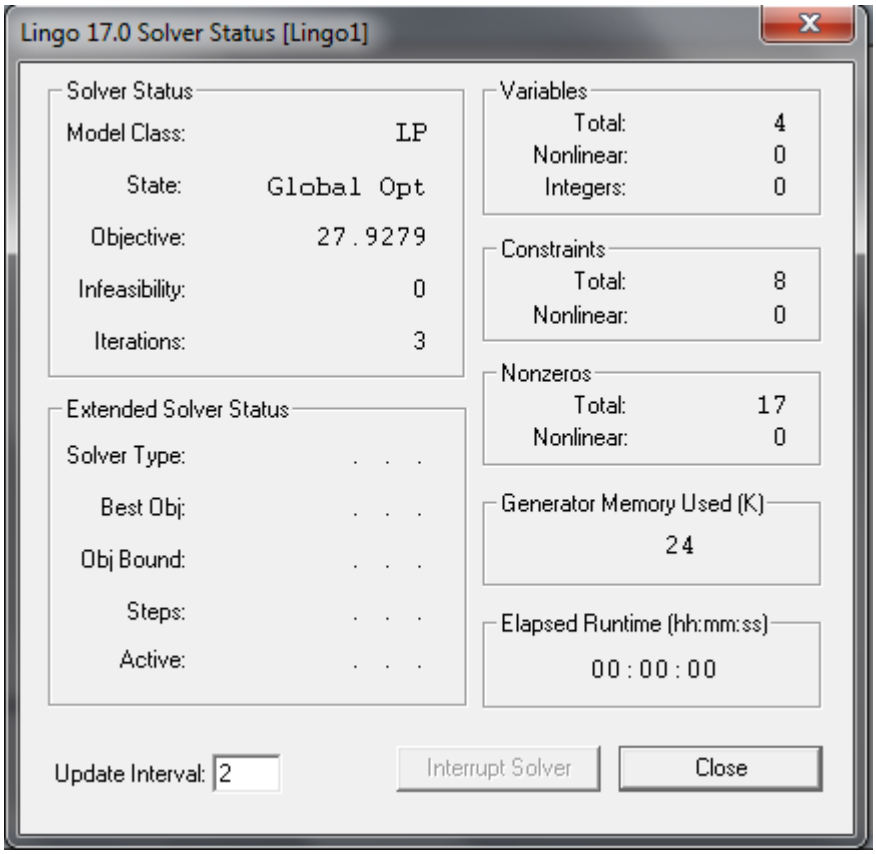
$$\min x_1 + x_2 + x_3 + x_4$$
$$-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50$$
$$5x_1 + 2x_2 + 0x_3 + 0x_4 \geq 100$$
$$3x_1 - 5x_2 + 10x_3 - 2x_4 \geq 25$$
$$x_1, x_2, x_3, x_4 \geq 0.$$

Ради решавања проблема треба урадити следеће: из менија *Solver* бира се команда Solve.



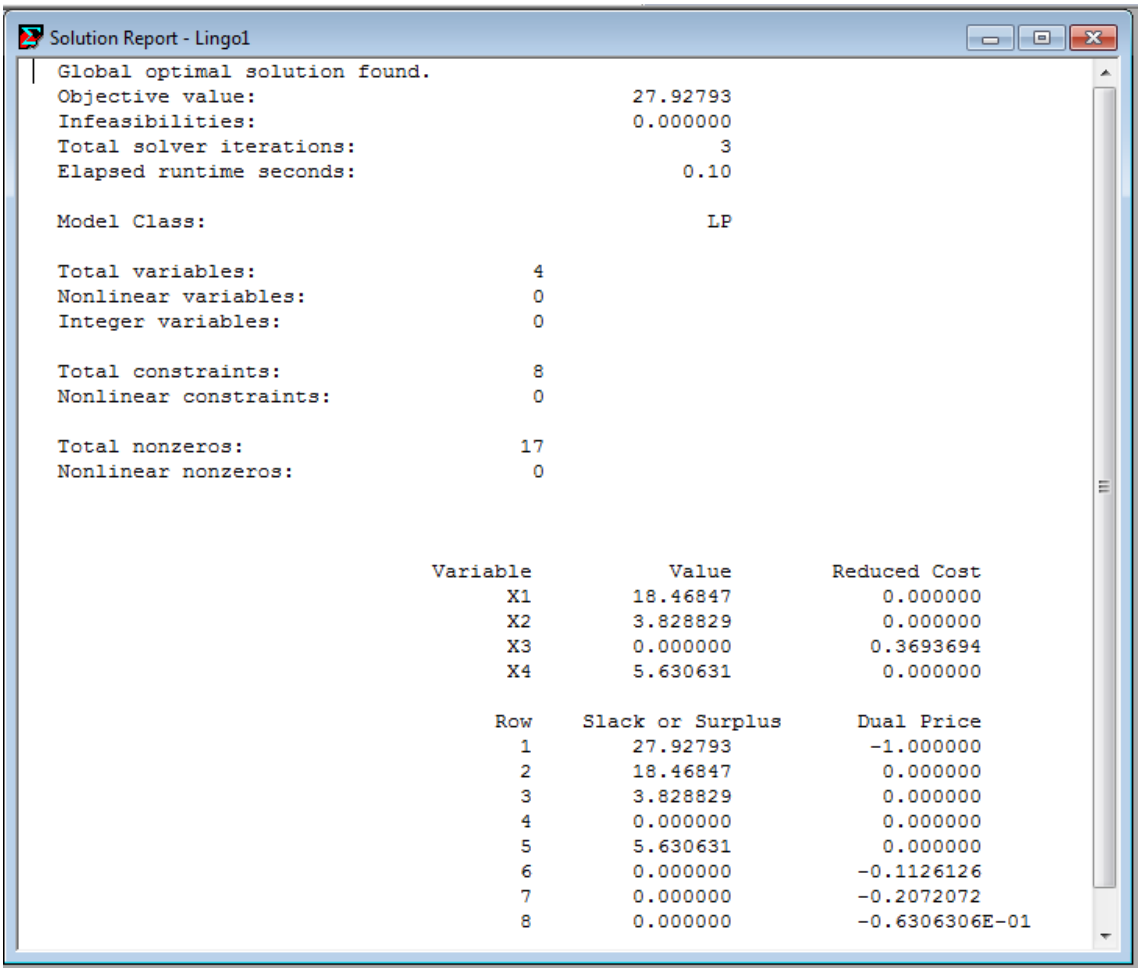
Слика 4.7: Унос проблема у *LINGO*

На екрану ће се појавити статус након извшене Solve команде.



Слика 4.8: Статус решења проблема из примера 8

У прозору *LINGO Solver Status* су приказане опште информације као на пример да је добијено решење оптимално и приказана је вредност функције циља у оптималном решењу. Прозор треба затворити и отворити прозор *Solution Report* у којем се налазе резултати анализе као што приказује слика 4.9.



Solution Report - Lingo1

```
Global optimal solution found.
Objective value:                              27.92793
Infeasibilities:                              0.000000
Total solver iterations:                             3
Elapsed runtime seconds:                          0.10

Model Class:                                        LP

Total variables:                     4
Nonlinear variables:                 0
Integer variables:                   0

Total constraints:                   8
Nonlinear constraints:               0

Total nonzeros:                     17
Nonlinear nonzeros:                  0


                              Variable           Value        Reduced Cost
                                    X1        18.46847            0.000000
                                    X2        3.828829            0.000000
                                    X3        0.000000           0.3693694
                                    X4        5.630631            0.000000

                                   Row    Slack or Surplus      Dual Price
                                     1        27.92793           -1.000000
                                     2        18.46847            0.000000
                                     3        3.828829            0.000000
                                     4        0.000000            0.000000
                                     5        5.630631            0.000000
                                     6        0.000000          -0.1126126
                                     7        0.000000          -0.2072072
                                     8        0.000000          -0.6306306E-01
```

Слика 4.9: Извештај о решењу проблема из примера 8

Оптимално решење проблема је $x_1 = 18.46847, x_2 = 3.828829, x_3 = 0$ и $x_4 = 5.630631$. Оно нам говори да политичар треба да потроши 18.5 хиљада евра на промоцију изградње путева, 3.8 хиљаде евра на промоцију контроле наоружања и 5.6 хиљада евра на промоцију такси на гориво, а да не треба да улаже у промоцију субвенција за фарме, како би освојио 50000 градских, 100000 приградских и 25000 сеоских гласова.

У извештају *Solution Report* (Слика 4.9) налазе се и *Reduced Cost*, *Slack or Surplus* и *Dual Price*, који су детаљно објашњени кроз претходни пример решен у програмском пакету *LINDO*.

Reduced Cost нам даје информацију о томе како ће промена коефицијента функције циља уз небазичну променљиву променити оптимално решење, па тако за небазичну променљиву x_3 *Reduced Cost* износи 0.369 што значи да ако политичар одлучи да ипак потроши хиљаду евра на промоцију субвенција за фарме, онда ће се трошкови увећати за 369 евра.

Slack or Surplus представљају вредности допунских променљивих и у нашем примеру су све једнаке нули.

Како је у питању проблем минимума *Dual Price* за i-то ограничење дефинише се као вредност за коју би се функција циља побољшала, тј. смањила, при јединичном повећању његове десне стране. Вредности у колони *Dual Price* су негативног знака из разлога што су у питању ограничења типа неједнакости $\geq$.

Да би се извршила анализа осетљивости треба изабрати команду *Range* из менија. Тада ће се поново појавити екран *Reports window* али у проширеном облику са анализом осетљивости, односно интервалима у којима се могу кретати коефицијенти функције циља, као и елементи десне стране ограничења.

Range Report - Lingo1

Ranges in which the basis is unchanged:

Objective Coefficient Ranges:

Variable	Current Coefficient	Allowable Increase	Allowable Decrease
X1	1.000000	0.4100000	0.7000000
X2	1.000000	0.2800000	0.1640000
X3	1.000000	INFINITY	0.3693694
X4	1.000000	0.1863636	0.3181818

Righthand Side Ranges:

Row	Current RHS	Allowable Increase	Allowable Decrease
2	0.000000	18.46847	INFINITY
3	0.000000	3.828829	INFINITY
4	0.000000	0.000000	INFINITY
5	0.000000	5.630631	INFINITY
6	50.00000	85.00000	40.32258
7	100.0000	89.28571	32.69231
8	25.00000	17.00000	28.40909

Слика 4.10: Извештај о анализи осетљивости проблема из Примера 8

Коефицијенти у функцији циља могу да се повећавају или смањују. Извештај показује колико се они појединачно могу повећати или смањити при чему су остали коефицијенти функције циља константни, а да се при томе не мења оптимална база. На пример коефицијент уз x_1 је у функцији циља 1, то видимо и у извештају (Слика 4.10.) у колони *Current Coefficient*, у продужетку имамо *Allowable Increase* и *Allowable Decrease* за тај коефицијент, и они износе 0.41 и 0.70, редом. То значи да се коефицијент који стоји уз x_1 може кретати у интервалу (0.30,1.41) а да се не промени оптимална база, с тим да остали коефицијенти остају непромењени.

Овај извештај нам указује и на могуће промене десне стране неједнакости (RHS), повећање или смањење појединачно, а да се при томе не мења оптимална база. На пример, ако посматрамо ограничење $-2x_1 + 8x_2 + 10x_4 \geq 50$ видимо да је његова десна страна 50, што можемо видети и у извештају у колони *Current RHS*, у наставку имамо *Allowable Increase* и *Allowable Decrease* за десну страну ограничења, који износе 85 и 40.32258, редом. То значи да ако се десна страна ограничења повећа највише за 85 или смањи највише за 40.32258 имаћемо исту оптималну базу.

4.2.3 Solver

Excel је програм који поседује широк спектар могућности. Solver обезбеђује додатне функције и могућност да се проблеми реше уз помоћ табела. Користи се за решавање проблема линеарног и нелинеарног програмирања.

Да бисмо активирали *Solver* бирамо *Add-Ins* у *Excel Options*. Из опадајуће листе бирамо *Excel Add-ins*, па затим клик на *Go*, након тога чекирамо *Solver Add-in* и кликом на *OK* додајемо *Solver* у картицу *Data*.

Кроз следећи пример ћемо видети како се проблем ЛП решава у *Solver*-у. У питању је проблем оптимизације производње.

Пример 9. *Одредити оптималне количине четири производа фабрике која жели да оствари максималан профит при датим условима. На тржишту постоји интересовање за сва четири производа, али проблем представљају ограничења по питању расположивог времена на*

машинама које врше обраду у процесу производње. Проблем се своди на максимизирање профита који се описује функцијом циља

$$\max 750x_1 + 710x_2 + 1250x_3 + 1350x_4,$$

где су приказани профити за четири производа. Ограничавајући фактори су приказани неједначинама

$$2056x_1 + 1433x_2 + 1754x_3 + 2658x_4 \leq 70000$$

$$590x_1 + 54x_2 + 486x_3 + 353x_4 \leq 12000$$

$$581x_1 + 70x_2 + 207x_3 + 520x_4 \leq 12000$$

$$587x_1 + 63x_2 + 548x_3 + 68x_4 \leq 11000$$

$$330x_1 + 761x_3 + 546x_4 \leq 4000$$

$$213x_1 + 21x_3 + 21x_4 \leq 2000$$

где су са леве стране неједнакости приказане потребна, а са десне стране лимитирана времена операција на машинама. За потребе производње ова четири производа, разматрано је шест различитих обрада.

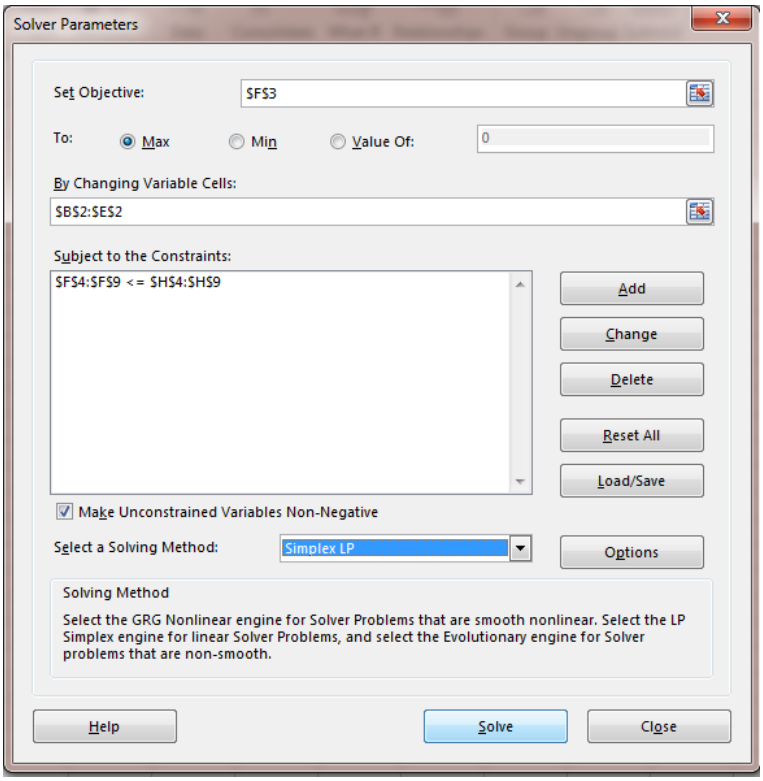
Одредити оптимално решење проблема у Solver-у.

Решење: Код *Solver*-а најважније је добро дефинисати модел на радном листу. На новом, празном радном листу прво се уносе имена променљивих. У следећем кораку се уносе коефицијенти и формуле ограничења у облику табеле. Максималан број ограничења је 500. Након тога се у произвољну ћелију уноси формула функције циља (F3). Група ћелија, која учествују у израчунавању вредности ограничења и функције циља, су променљиве ћелије. Број променљивих ћелија не сме да буде већи од 200. *Solver* прилагођава вредности у променљивим ћелијама да би задовољио ограничења и дао жељени резултат за ћелију циља. *Solver* не дозвољава коришћење више од 1000 ћелија укупно током решавања једног проблема.

	A	B	C	D	E	F	G	H
1		x_1	x_2	x_3	x_4			
2		1	1	1	1			
3	функција циља	750	710	1250	1350	4060		RHS
4	ограничења	2056	1433	1754	2658	7901	<=	70000
5		590	54	486	353	1483	<=	12000
6		581	70	207	520	1378	<=	12000
7		587	63	548	68	1266	<=	11000
8		330	0	761	546	1637	<=	4000
9		213	0	21	21	255	<=	2000

Слика 4.11: Унос података у *Solver*-у

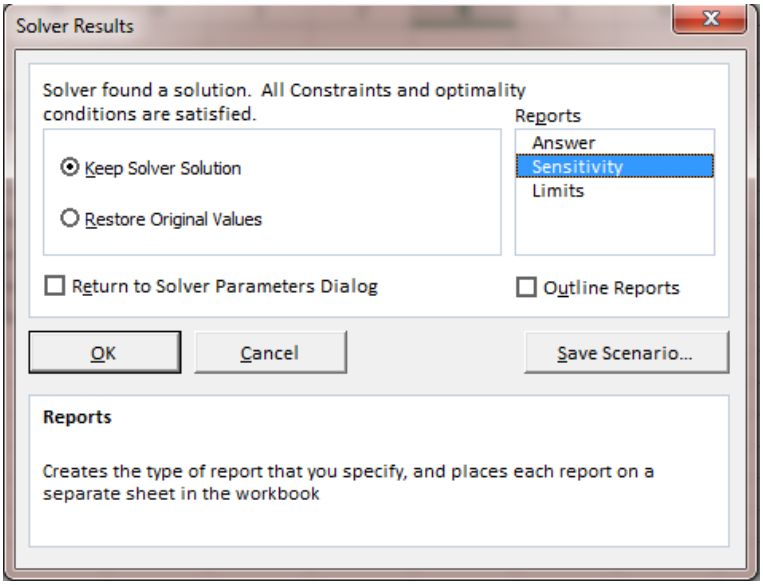
Након уноса података, Solver се може покренути из менија *Microsoft Excela*. Са кликом на *Data* бира се функција *Solver*.



Слика 4.12: Избор параметара за решавање проблема из Примера 9 у *Solver*-у

Отвара се дијалог прозор *Solver Parameters* у коме се одређује ћелија функције циља (Set Target Cell) и ћелије променљивих вредности (By Changing Cells). Променљиве ћелије морају директно или индиректно бити у вези са ћелијом функције циља. Са кликом на *Add* могу се унети ћелије ограничења (Subject to the Constraints). Поред претходних корака, треба да се одабере ставка *Min*.

Може се још кликнути на *Options* како би се унеле додатне опције модела, на пример да се користи линеарно програмирање, те да за количине нису дозвољене негативне вредности итд.



Слика 4.13: Информација о решењу и могућност одабира типа извештаја у *Solver*-у

Након клика на дугме OK у прозору *Solver Options*, програм се враћа на почетни прозор. За решавање проблема потребно је притиснути дугме *Solve*, након чега се отвори нови дијалог прозор *Solver Results* где треба изабрати једну од ових могућности

1. *Keep Solver Solution* – да би програм вредности решења сачувао на радном листу.

2. *Restore Original Values* – да би програм вратио оригиналне вредности, вредности које су биле пре клика на Solver дугме.

И даље смо у прозору *Solver Results* где сада бирамо врсту извештаја од понуђених:

1. *Answer* – генерише извештај о оригиналним и коначним вредностима променљивих у моделу и сачува га као нови радни лист у Excel радној књизи.

2. *Sensitivity* – извештај о променама променљивих у моделу.

3. *Limits* – извештај о достизању граница појединих променљивих.

Извештај ће се појавити на новом листу у Excel фајлу.

Microsoft Excel 15.0 Sensitivity Report
Worksheet: [Book1]Sheet1
Report Created: 9/23/2018 8:32:13 PM

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
B2	x1	0	-433.8718658	750	433.8718658	1E+30
C2	x2	42.41490014	0	710	311.2371722	246.0140894
D2	x3	5.256241787	0	1250	1E+30	334.8827481
E2	x4	0	-240.2706708	1350	240.2706708	1E+30

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
F4	ограничења	70000	0.495464061	70000	123907.6612	60780.55191
F5		4844.938116	0	12000	1E+30	7155.061884
F6		4057.08506	0	12000	1E+30	7942.91494
F7		5552.559208	0	11000	1E+30	5447.440792
F8		4000	0.500599259	4000	8803.591382	4000
F9		110.3810775	0	2000	1E+30	1889.618922

Sensitivity Report 1 | Sheet1

Слика 4.14: Решење и анализа осетљивости

	A	B	C	D	E	F	G	H
1		x_1	x_2	x_3	x_4			
2		0	42.4149	5.256242	0			
3	функција циља	750	710	1250	1350	36684.88		RHS
4	ограничења	2056	1433	1754	2658	70000	<=	70000
5		590	54	486	353	4844.938	<=	12000
6		581	70	207	520	4057.085	<=	12000
7		587	63	548	68	5552.559	<=	11000
8		330	0	761	546	4000	<=	4000
9		213	0	21	21	110.3811	<=	2000
10								

Sensitivity Report 1 | Sheet1

Слика 4.15: Решење проблема из примера 9

У радном листу *Sheet1* (слика 4.15) у ћелијама променљивих вредности појављује се решење проблема, па из добијених резултата видимо да фирма треба да произведе 42.4149 комада другог артикла и 5.256242 трећег артикла како би остварила максималан профит у износу од 36684.88, а да су при томе задовољени сви задати услови.

Нови радни лист (слика 4.14), *Sensitivity Report 1*, представља извештај о анализи осетљивости проблема. У овом извештају се налазе изрази са којима смо се сусретали у извештајима о анализи осетљивости код програмских пакета LINDO и LINGO, као што су *Reduced Cost*, *Objective Coefficient*, *Allowable Increase/Decrease*, *Shadow Price*, *Constraint R.H.Side*.

У извештају о анализи осетљивости се налазе две табеле *Variable Cells*, која се односи на променљиве и коефицијенте који се налазе уз те променљиве у функцији циља, и табела *Constraints*, која се односи на ограничења проблема, а поготово на десну страну ограничења.

Посматрамо прву табелу *Variable Cells* и у колони *Final Value* поново видимо да је оптимално решење $x_1 = 0, x_2 = 42.4149, x_3 = 5.256242$ и $x_4 = 0$. У наставку се налази колона *Reduced Cost*, чије су вредности нула за базичне променљиве, а за небазичне променљиве представља износ за који се мења функција циља при производњи једне јединице посматраног производа. На пример, уколико се предузеће одлучи да произведе један комад првог артикла, функција циља ће се променити за -433,87.

Наредна колона је *Objective Coefficient*, у којој се налазе коефицијенти који стоје уз променљиве у функцији циља, а у продужетку се налазе колоне *Allowable Increase* и *Allowable Decrease*, које показују колико се коефицијенти појединачно могу повећати или смањити при чему су остали коефицијенти функције циља константни, а да се при томе не мења оптимална база.

У другој табели (*Constraints*), налази се колона *Final Value* у којој су вредности леве стране ограничења у оптималном решењу. У продужетку је колона *Shadow Price* (*Dual Price*). Када је у питању проблем максимума, *Shadow Price* за i-то ограничење се дефинише као вредност за коју би се функција циља повећала при јединичном повећању његове десне стране. Затим, након колоне *Shadow Price*, следи колона *Constraint R.H.Side* која представља вредности десне стране ограничења, а онда су у наставку колоне *Allowable Increase* и *Allowable Decrease*, које показују колико се десна страна сваког ограничења појединачно може повећати или смањити, а да се при томе не мења оптимална база.

4.2.4 Matlab

Matlab је још један од програма који има широк спектар могућности,а ми ћемо се фокусирати на решавање проблема ЛП.

Подсетимо се прво основних операција и дефиниција за разумевање решавања проблема ЛП у *Matlab*-у:

- Операција транспоновања означена је суперскриптом T (апостроф у *Matlab*-у)

$$\begin{bmatrix}1 & 2 & 3\end{bmatrix}^T = \begin{bmatrix}1\\2\\3\end{bmatrix}, \begin{bmatrix}1\\2\\3\end{bmatrix}^T = \begin{bmatrix}1 & 2 & 3\end{bmatrix}, \begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\end{bmatrix}^T = \begin{bmatrix}1 & 4\\2 & 5\\3 & 6\end{bmatrix}$$

- Вектор врсте: $\begin{bmatrix}1 & 2 & 3\end{bmatrix}$, вектор колоне: $\begin{bmatrix}1\\2\\3\end{bmatrix}$.

Дефиниција 16. *Нека је c вектор колона димензије n, b вектор колона димензије m и A матрица димензија $m \times n$. Проблем линеарног*

програмирања може бити проблем минимума

$$\min c^T x$$

или проблем максимума

$$\max c^T x$$

при ограничењима

$$Ax \leq b,$$

где је x вектор колона димензије n.

У општем случају проблем ЛП може имати и ограничења типа једнакости.

Дефиниција 17. *Нека је c вектор колона димензије n, b вектор колона димензије m, b_{eq} вектор колона димензије k и нека су A и A_{eq} матрице димензија $m \times n$ и $k \times n$, редом. Проблем линеарног програмирања може бити проблем минимума*

$$\min c^T x$$

или проблем максимума

$$\max c^T x$$

при ограничењима

$$Ax \leq b$$

$$A_{eq}x = b_{eq}$$

где је x вектор колона димензије n.

Matlab подразумева да је у питању проблем минимума, па уколико је наш почетни проблем проблем максимума потребно да је да га трансформишемо у проблем минимума.

За проблеме ЛП користимо команду *linprog*. Пре коришћења команде, сва ограничења морају бити у облику $\leq$ и $=$. Комплетан формат *linprog* наредбе је:

$$[x, fval, exitflag, output, lambda] = linprog\,(c, A, b, A_{eq}, b_{eq}, lb, ub, x_0, options)\,;$$

- x представља оптимално решење почетног проблема;
- $fval$ представља оптималну вредност функције циља;
- $exitflag$ описује излазни услов;

- *output* садржи информације о процесу оптимизације;
- *lambda* преставља Лагранжов множитељ решења x и корисно је за анализу осетљивости;
- c су коефицијенти функције циља, у векторском формату;
- A су коефицијенти ограничења типа неједнакости, у матричном формату;
- b је вектор десне стране ограничења типа неједнакости;
- A_{eq} су коефицијенти ограничења типа једнакости, у матричном формату;
- b_{eq} је вектор десне стране ограничења типа једнакости;
- lb и ub представљају доњу и горњу границу решења x;
- x_0 је почетна вредност решења x, ако се остави празно $[\,]$, онда се претпоставља да је почетна вредност нула вектор;
- *options* садржи параметре који описују како *linprog* треба покренути.

Међутим, проблем можда не укључује све параметре у *linprog* команди, у том случају се користе празни вектори $[\,]$ уместо неупотребљених параметара.

- Ако проблем има само ограничења облика неједнакости, користи се следећа команда:

  $$[x, fval, exitflag, output, lambda] = linprog\,(c, A, b, [\,], [\,], lb, [\,], [\,], options)\,;$$

- Ако проблем, поред ограничења облика неједнакости, има ограничења типа једнакости, користи се следећа команда:

  $$[x, fval, exitflag, output, lambda] = linprog\,(c, A, b, A_{eq}, b_{eq}, lb, [\,], [\,], options)\,;$$

Манипулисањем параметра *options* одређујемо како треба да се покрене *linprog*-а. На пример, ако желимо да *linprog* користи дуал симплекс алгоритам, прво урадимо следеће:

$options = optimoptions('linprog',' Algorithm',' dual-simplex',' Display',' iter');$

Параметар *Display* говори *liprog* да ли да укључи или онемогући приказивање итерације алгоритма на екрану. Може да се онемогући приказ екрана користећи *off* уместо *iter*.

Пример 10. *Одабрати портфолио пакет из скупа алтернативних инвестиција. Потребно је максимизирати очекивани поврат инвестиција, с тим да су очекиване годишње стопе повраћаја познате (дате су у табели 4.7).*

Инвестиције	Очекивана годишња стопа повраћаја (%)
Акција А - производни сектор	15.4 %
Акција Б - производни сектор	19.2%
Акција В - сектор хране и пића	18.7%
Акција Г - сектор хране и пића	13.5%
Заједнички фонд Е	17.8%
Заједнички фонд К	16.3%

Табела 4.7: Очекиване годишње стопе повраћаја

На основу расположивог капитала, политике компаније, ликвидности, трајања привредног живота инвестиција, менаимент је поставио одређене захтеве:

1. *Укупан расположиви износ је 90000 EUR.*
2. *Износ који се улаже у акције производног сектора не треба да буде већи од 50% од укупног расположивог износа. Исто важи и за сектор хране и пића.*
3. *Износ акције производног сектора са већим приносом је мањи или једнак 80% од укупног расположивог износа за улагање у акције сектора производње. Такође, износ акције сектора хране и пића са већим приносом је мањи или једнак 80% од укупног расположивог износа за улагање у акције сектора хране и пића.*
4. *Износ расположив са улагање у акцију Б је мањи или једнак 10% од укупног расположивог износа за улагање у акције.*
5. *Износ за улагање у заједничке фондове је мањи или једнак 25% од износа акција у производном сектору.*

Проблем решити у Matlab-у.

Решење: Уводимо шест променљивих:

- x_1 - уложени износ у акцији А производног сектора.
- x_2 - уложени износ у акцији Б производног сектора.

- x_3 - уложени износ у акцији В сектора хране и пића.
- x_4 - уложени износ у акцији Г сектора хране и пића.
- x_5 - уложени износ у заједнички фонд Е.
- x_6 - уложени износ у заједнички фонд К.

Математички модел проблема је облика

$$\max 0.154x_1 + 0.192x_2 + 0.187x_3 + 0.135x_4 + 0.178x_5 + 0.163x_6$$
$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 90000$$
$$x_1 + x_2 \leq 45000$$
$$x_3 + x_4 \leq 45000$$
$$-0.8x_1 + 0.2x_2 \leq 0$$
$$0.2x_3 - 0.8x_4 \leq 0$$
$$-0.1x_1 + 0.9x_2 - 0.1x_3 - 0.1x_4 \leq 0$$
$$-0.25x_1 - 0.25x_2 + x_5 + x_6 \leq 0$$
$$x_i \geq 0, i = 1, 2, 3, 4, 5, 6.$$

Прво дефинишемо векторе и матрице који су улазни параметри, а затим бирамо опцију да *linprog* користи симплекс алгоритам.

```
>> c=-[0.154 0.192 0.187 0.135 0.178 0.163]';
b=[90000 45000 45000 0 0 0 0]';
A=[1 1 1 1 1 1;
 1 1 0 0 0 0;
0 0 1 1 0 0;
-0.8 0.2 0 0 0 0;
0 0 0.2 -0.8 0 0;
-0.1 0.9 -0.1 -0.1 0 0;
-0.25 -0.25 0 0 1 1];
lb=[0 0 0 0 0 0];
```

Слика 4.16: Унос података

```
>> options=optimoptions('linprog', 'Algorithm', 'simplex', 'Display', 'iter')

options =

  linprog options:

   Options used by current Algorithm ('simplex'):
   (Other available algorithms: 'active-set', 'interior-point')

   Set by user:
    Algorithm: 'simplex'
      Display: 'iter'

   Default:
    Diagnostics: 'off'
        MaxIter: '100*numberOfVariables'
         TolFun: 1.0000e-06
```

Слика 4.17: Одабир опције

```
>> [x, fval, exitflag, output, lambda]=linprog(c,A,b,[],[],lb,[],[],options)
The default starting point is feasible, skipping Phase 1.

Phase 2: Minimize using simplex.
      Iter             Objective                Dual Infeasibility
                         f'*x                     A'*y+z-w-f
       0                     0                     0.414761
       1                     0                     0.396027
       2                     0                      1.03365
       3                     0                      7.32322
       4              -9132.43                     0.330259
       5               -9418.5                        0.289
       6              -15400.8                            0
Optimization terminated.

x =

   1.0e+04 *

    2.7900
    0.8100
    3.6000
    0.9000
    0.9000
         0


fval =

  -1.5401e+04


exitflag =

     1


output =

         iterations: 6
          algorithm: 'simplex'
       cgiterations: []
            message: 'Optimization terminated.'
    constrviolation: 1.8190e-10
      firstorderopt: 2.3551e-11


lambda =

    ineqlin: [7x1 double]
      eqlin: [0x1 double]
      upper: [6x1 double]
      lower: [6x1 double]
```

Слика 4.18: Решење

Проблем је решен помоћу команде *linprog*. Добијени извештај садржи оптимално решење и вредност функције циља у оптималном решењу. Наведени резултати говоре да максимални очекивани поврат инвестиција у износу од 15400 EUR, се остварује ако уложимо у акцију А 27900 EUR, у акцију Б 8100 EUR, у акцију В 36000 EUR, у акцију Г

9000 EUR, у заједнички фонд Е 9000 EUR, а у заједнички фонд К не треба улагати.

Приметимо да се у извештају налази и податак *lambda* помоћу ког добијамо *DUAL PRICES* и *REDUCED COST*. Заправо, *lambda.ineqlin* представља *DUAL PRICES*, а *lambda.lower REDUCED COST*.

```
>> shadow_price_inequilities=lambda.ineqlin

shadow_price_inequilities =

    0.1618
         0
    0.0186
         0
    0.0520
    0.0380
    0.0162
```

```
>> lambda.lower

ans =

         0
         0
         0
         0
         0
    0.0150
```

Слика 4.19: Shadow prices и Reduced Cost

Као и до сада, *REDUCED COST* се односи на небазичне променљиве и дефинише се као износ за који би оптимална вредност функције циља била умањена ако би уложили једну јединицу посматране акције или фонда. Како је x_6 једина небазична променљива, *REDUCED COST* за ту променљиву износи 0.0150, што значи да ако би ипак уложили једну јединицу у заједнички фонд К, профит би био мањи од оптималног за 0.0150.

Пошто је у питању проблем максимума, *DUAL PRICES* за i-то ограничење дефинише се као вредност за коју би се вредност функције циља побољшала тј. повећала, при јединичном повећању његове десне стране.

5 Закључак

Линеарно програмирање је математички метод за распоређивање ограничених ресурса на планиране активности на најбољи могући начин, с обзиром на унапред дефинисан циљ. Задатак линеарног програмирања је да се одреди максимална или минимална вредност линеарне функције циља, при унапред датим ограничавајућим условима, који су описани преко линеарних релација.

У раду је представљен проблем ЛП као и графичка и симплекс метода које се користе за решавање ЛП проблема. Осим тога, рад се бави и анализом осетљивости којом се испитује до које границе се могу мењати улазни параметри проблема а да то не утиче на структуру оптималног решења, односно којом се може испитати како се мењају поједини показатељи оптималности у случају јединичних промена количина ресурса.

За праксу значајни проблеми ЛП су проблеми великих димензија који изискују примену разних софтвера. Стога су представљени програмски пакети за брзо решавање ЛП проблема великих димензија као што су *LINDO*, *LINGO*, *Solver* и *Matlab*. Обрађени примери решени су применом наведених софтвера и анализирана су њихова решења.

Програмски пакети *LINDO* и *LINGO* су специјализовани за решавање искључиво проблема математичког програмирања. Они не захтевају много времена за учење, проблеми се веома једноставно уносе у програм, а решење се добија у виду извештаја који се лако може разумети. За разлику од њих *Solver* и *Matlab* су програми који имају широк спектар других могућности, па обзиром на то, унос података, читање резултата и тумачење анализе осетљивости у њима је захтевније.

6 Литература

[1] S. Bastani, *Solving Linear and Integer Programs in Matlab*, Lund University, Department of Electrical and Information Technology, 2018

[2] И. Брајдић, Ј. Боговић, *Примјена линеарног програмирања у проблему прехране у дому за незбринуту дјецу у Славонском броду*, Факултет за туристички и хотелски менаџмент, Ријека, 2004

[3] G. Hadley, *Linear programming*, Addison - Wesley Pub. Co., 1962

[4] Ђ. Илић, Ј. Радуловић, Б. Микарић, *Компјутерска примена симплекс методе у решавању проблема оптимизације производње*, Висока пословна школа струковних студија "Проф. др Радомир Бојковић", Крушевац, 2005

[5] М. Ивановић, *Операциона истраживања - вежбе*, Математички факултет, Београд, 2014

[6] V. Kostoglou, *Linear programming - Case studies and solutions*, Alexander Technological Educational Institute of Thessaloniki, Thessaloniki, 2018

[7] С. Крчевинац, М. Чангаловић, В. Ковачевић - Вујичић, М. Мартић, М. Вујошевић, *Операциона истраживања I*, Београд, 2006

[8] Schrage, Linus E, *Linear, integer and quadratic programming with LINDO*, Palo Alto, CA: Scientific Press, 1986

[9] Т. Симовић, *Оптимизација портфолија хартија од вредности на Београдској берзи на основу теорије Харија Марковица*, Универзитет Сингидунум, Београд, 2012

[10] Б. Стојановић, *Алгоритамске стратегије*, Природно - математички факултет, Крагујевац, 2017

[11] К. Сурла, З. Лозанов - Црвенковић, *Операциона истраживања*, Природно - математички факултет, Нови Сад, 2002

[12] В. Вујчић, М. Ашић, Н. Миличић, *Математичко програмирање*, Математички институт, Београд, 1980

[13] W. L. Winston, *Operations research: Applications and Algorithms, 4th. Edition*, Duxbury Press, 2003

[14] `http://www.mathworks.com/help/optim/ug/intlinprog.html`

[15] `https://www.solver.com`

БИОГРАФИЈА

Александра Јовановић је рођена 03.09.1994. године у Нишу. Завршила је основну школу "Карађорђе" у Горњем Матејевцу, а потом је уписала средњу школу "Економска школа Ниш", смер банкарски службеник. Средњу школу је завршила 2013. године са одличним успехом и уписала основне академске студије примењене математике на Природно-математичком факултету у Новом Саду. Положила је све испите предвиђене наставним планом и 2016. године завршила основне академске студије. Исте године је уписала мастер студије примењене математике. Закључно са септембарским роком 2018. године положила је све испите и стекла услов за одбрану мастер рада. Од септембра 2018. запослена је у компанији ДДОР Нови Сад у Дирекцији за стратешко планирање, менаџмент контролу и извештавање.

УНИВЕРЗИТЕТ У НОВОМ САДУ
ПРИРОДНО-МАТЕМАТИЧКИ ФАКУЛТЕТ
КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број:
РБР

Идентификациони број:
ИБР

Тип документације: *монографска документација*
ТД

Тип записа: *текстуални штампани материјал*
ТЗ

Врста рада: *мастер рад*
ВР

Аутор: *Александра Јовановић*
АУ

Ментор: *др Сања Рапајић*
МН

Наслов рада: *Линеарно програмирање и анализа осетљивости*
НР

Језик публикације: *српски (ћирилица)*
ЈП

Језик извода: *с/е*
ЈИ

Земља публиковања: *Република Србија*
ЗП

Уже географско подручје: *Војводина*
УГП

Година: *2018.*
ГО

Издавач: *ауторски репринт*
IZ

Место и адреса: *Нови Сад, Трг Доситеја Обрадовића 4*
МА

Физички опис рада: *(6/80/15/12/20/0/0)(број поглавља/број страна/број литерарних цитата/број табела/број слика/број графика/број прилога)*
ФО

Научна област: *математика*
НО

Научна дисциплина: *операциона истраживања*
НД

Кључне речи: *линеарно програмирање, симплекс метода, анализа осетљивости*
ПО

УДК

Чува се: *у библиотеци Департмана за математику и информатику, Природно-математичког факултет, у Новом Саду.*
ЧУ

Важна напомена:
ВН

Извод: *У мастер раду је анализирана универзална метода за решавање проблема линеарног програмирања - Симплекс метода и приказана је анализа осетљивости којом се испитује како промене улазних параметара утичу на структуру оптималног решења. Осим тога представљена је примена савремених софтвера за брзо решавање проблема линеарног програмирања великих димензија, као што су LINDO, LINGO, Solver и Matlab.*
ИЗ

Датум прихватања теме од стране НН већа: *03. септембар 2018.*
ДП

Датум одбране:
DO

Чланови комисије:
КО

Председник: *др Ненад Теофанов, редовни професор Природно-математичког факултета у Новом Саду*
Ментор: *др Сања Рапајић, ванредни професор Природно-математичког факултета у Новом Саду*
Члан: *др Милица Жигић, доцент Природно-математичког*

факултета у Новом Саду

UNIVERSITY OF NOVI SAD
FACULTY OF SCIENCE
KEY WORDS DOCUMENTATION

Accession number:
ANO

Identification number:
INO

Document type: *monograph type*
DT

Type of record: *printed text*
TR

Contents code: *Master thesis*
CC

Author: *Aleksandra Jovanović*
AU

Mentor: *Prof. Dr. Sanja Rapajić*
MN

Title: *Linear programming and sensitivity analysis*
XI

Language of text: *Serbian (cyrillic)*
LT

Language of abstract: *s/e*
LA

Country of publication: *Republic of Serbia*
CP

Locality of publication: *Vojvodina*
LP

Publication year: *2018.*
PY

Publisher: *author's reprint*
PU

Publ. place: *Novi Sad, Trg Dositeja Obradovića 4*
PP

Physical description:
(6/80/15/12/20/0/0)(chapters/pages/quotations/tables/pictures/graphics/enclosures)
PD

Scientific field: *mathematics*
SF

Scientific discipline: *operation research*
SD

Key words: *linear programming, simplex method, sensitivity analysis*
UC

Holding data: *Department of Mathematics and Informatics' Library, Faculty of Sciences, Novi Sad*
HD

Note:
N

Abstract: *The general method for solving LP problem - Simplex method is proposed in this thesis. The sensitivity analysis, which examines how changes of input parameters effect on optimal solution, is also presented. Furthermore, application of modern software for solving LP problems of large dimensions like LINDO, LINGO, Solver and Matlab is presented.*
AB

Accepted by the Scientific Board on: *03 September 2018*
ASB

Defended:
DE

Thesis defend board:
DB

President: *dr Nenad Teofanov, full professor at Faculty of Sciences in Novi Sad*
Mentor: *dr Sanja Rapajić, associate profesor at Faculty of Sciences in Novi Sad*
Member: *dr Milica Žigić, assistant professor at Faculty of Sciences in Novi Sad*