Izbrani algoritmi

Linearno programiranje

Damjan Strnad

Linearno programiranje

- naj bo *f* linearna kriterijska funkcija *n* spremenljivk
- podane so tudi omejitve v obliki linearnih neenačb z istimi spremenljivkami, ki omejujejo konveksno območje v *n*-dimenzionalnem prostoru
- iskanje minimuma ali maksimuma funkcije *f* pri podanih omejitvah imenujemo **linearno programiranje** (LP) ali **linearna optimizacija**, samo definicijo problema pa **linearni program**
- LP ima veliko aplikacij v ekonomiji, proizvodnji, delitvi virov, ...

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

krit. funkcija: $f\left(x_{1,}x_2\right)=29x_1+45x_2$

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

dobiček po zabojniku K

krit. funkcija: $f(x_1, x_2) = 29 x_1 + 45 x_2$

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

št. zabojnikov K na uro

krit. funkcija: $f(x_1, x_2) = 29x_1 + 45x_2$

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

dobiček po zabojniku L

krit. funkcija: $f\left(x_{1,} x_2\right) = 29 x_1 + 45 x_2$

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

št. zabojnikov L na uro

krit. funkcija: $f\left(x_{1,} x_2\right) = 29x_1 + 45x_2$

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

krit. funkcija: $f(x_1, x_2) = 29x_1 + 45x_2$

omejitve: $2x_1 + 8x_2 \leq 60$

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

krit. funkcija: $f(x_1, x_2) = 29x_1 + 45x_2$

minut po zabojniku K na stroju M_1

omejitve: $2x_1 + 8x_2 \leq 60$

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

krit. funkcija: $f(x_1, x_2) = 29x_1 + 45x_2$

št. zabojnikov K na uro

omejitve: $2x_1 + 8x_2 \leq 60$

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

krit. funkcija: $f(x_{1,} x_2) = 29 x_1 + 45 x_2$

minut po zabojniku L na stroju M_1

omejitve: $2 x_1 + 8 x_2 \leq 60$

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

krit. funkcija: $f(x_1, x_2) = 29x_1 + 45x_2$

št. zabojnikov L na uro

omejitve: $2x_1 + 8x_2 \leq 60$

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

krit. funkcija: $f\left(x_{1,} x_2\right) = 29\, x_1 + 45\, x_2$

minut zasedenosti stroja M_1 v uri

omejitve: $2\, x_1 + 8\, x_2 \leq 60$

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

krit. funkcija: $f(x_1, x_2) = 29x_1 + 45x_2$

omejitve:

$$2x_1 + 8x_2 \le 60$$

$$4x_1 + 4x_2 \le 60$$ ← enako sklepanje za stroj M_2

Linearno program. – zgled

- V proizvodnji dveh vrst zabojnikov K in L uporabljamo dva stroja M_1 in M_2. Pri proizvodnji zabojnika K potrebujemo stroj M_1 dve minuti in stroj M_2 štiri minute. Podobno, zabojnik L zasede stroj M_1 osem minut in stroj M_2 štiri minute. Dobiček pri proizvodnji zabojnika K je 29 €, pri proizvodnji zabojnika L pa 45 €. Sestavi načrt proizvodnje (št. proizvedenih zabojnikov K in L na uro), ki maksimizira dobiček.
- Linearni program*:

krit. funkcija: $f(x_1, x_2) = 29x_1 + 45x_2$

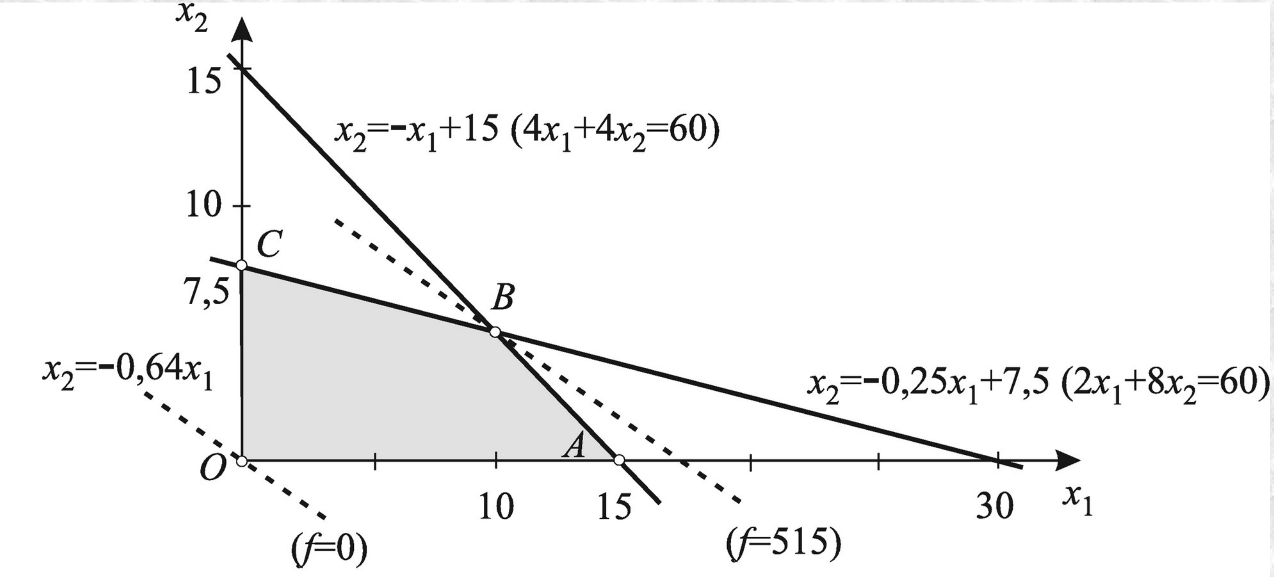
omejitve:

$$
\begin{aligned}
2x_1 + 8x_2 &\leq 60 \\
4x_1 + 4x_2 &\leq 60 \\
x_1 &\geq 0 \\
x_2 &\geq 0
\end{aligned}
$$

omejitve nenegativnosti

Linearno program. – zgled

- grafična rešitev*:
  - omejitve izrišemo kot premice, od katerih vsaka razdeli ravnino na dve polravnini, tako da samo ena vsebuje **dopustne rešitve** (feasible solution)
  - presek veljavnih polravnin določa konveksno **dopustno regijo** (feasible region) ali **simpleks** (simplex)
  - enačba kriterijske funkcije določa vzporedne premice enakovrednih rešitev
  - optimalna rešitev (če obstaja) je vedno oglišče ali rob simpleksa



Linearno program. – zgled

- grafična rešitev*:
  - omejitve izrišemo kot premice, od katerih vsaka razdeli ravnino na dve polravnini, tako da samo ena vsebuje **dopustne rešitve** (feasible solution)
  - presek veljavnih polravnin določa konveksno **dopustno regijo** (feasible region) ali **simpleks** (simplex)
  - enačba kriterijske funkcije določa vzporedne premice enakovrednih rešitev
  - optimalna rešitev (če obstaja) je vedno oglišče ali rob simpleksa

- grafično reševanje je uporabno samo za 2D primere, v višjih dimenzijah potrebujemo algoritem za iskanje optimuma

Splošni linearni program

$$f(x_1,\ldots,x_n)=c_1x_1+c_2x_2+\ldots+c_nx_n=\sum_{j=1}^{n}c_jx_j$$

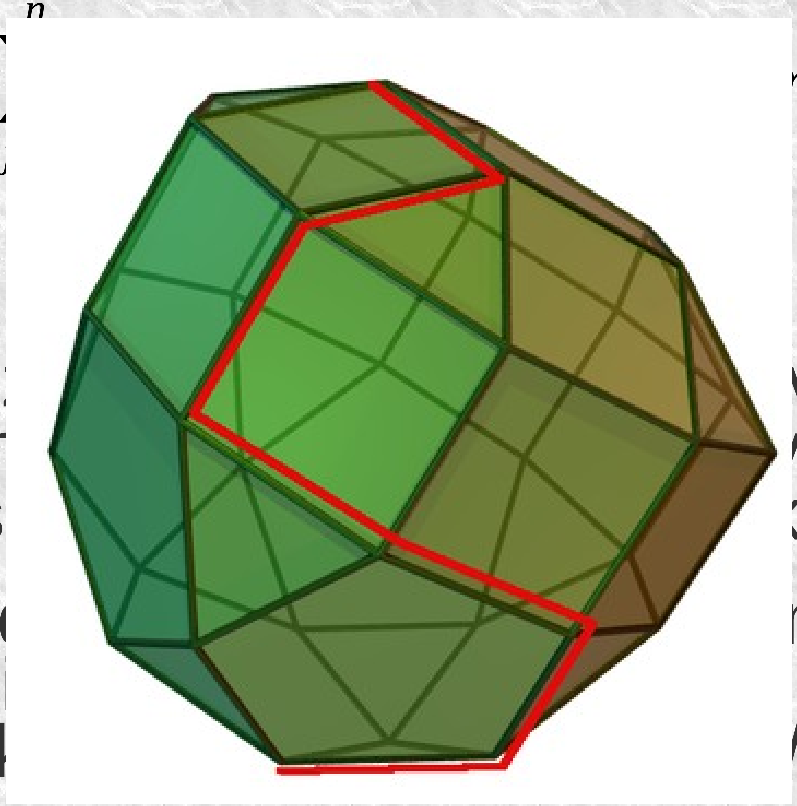
$$\sum_{j=1}^{n}a_{ij}x_j\le b_i,\qquad i=1,2,\ldots,m$$

$$x_j\ge 0,\qquad j=1,2,\ldots,n$$

- v višjih dimenzijah je dopustna regija večdimenzionalni konveksni simpleks, optimalna rešitev pa je eno od njegovih oglišč, robov ali mejnih ploskev (hiperravnin)
- dokazano je, da iz vsakega oglišča simpleksa, ki ni optimalna rešitev, poteka vsaj en rob, vzdolž katerega kriterijska funkcija monotono narašča/pada
- z izbiro poljubnega začetnega oglišča in pomikanjem vzdolž robov v boljše sosede lahko dosežemo globalni optimum => **algoritem simpleks***

Splošni linearni program

$$f(x_{1,}\ldots,x_n)=c_1x_1+c_2x_2+\ldots+c_nx_n=\sum_{j=1}^{n}c_jx_j$$



- v višjih dimenz večdimenzionalni konveksni sim pa je eno od njegovih ogliš ev (hiperravnin)
- dokazano je, mpleksa, ki ni optimalna reš vzdolž katerega kriterijska funk pada
- z izbiro poljubnega začetnega oglišča in pomikanjem vzdolž robov v boljše sosede lahko dosežemo globalni optimum => **algoritem simpleks***

Standardna oblika LP

- standardna oblika LP:
  - podani so vektor $\mathbf{c}=[c_j]$ (koeficienti krit. fun.), vektor $\mathbf{b}=[b_i]$ in matrika $A=[a_{ij}]$ (koeficienti omejitev), $i=1,...,m$; $j=1,...,n$
  - iščemo vektor $\mathbf{x}=[x_j]$, $j=1,...,n$, ki <u>maksimizira</u> $\mathbf{c}^T\mathbf{x}$ ob omejitvah $\mathbf{Ax}\leq\mathbf{b}$ in $\mathbf{x}\geq 0$ (vsak element $\mathbf{x}$ je ≥ 0)*
- LP ni v standardni obliki, če:
  - iščemo rešitev, ki <u>minimizira</u> $\mathbf{c}^T\mathbf{x}$
  - nekatere spremenljivke nimajo omejitve nenegativnosti
  - obstajajo omejitve enakosti
  - obstajajo omejitve neenakosti z znakom $\geq$ namesto $\leq$

Standardna oblika LP

- standardna oblika LP:
  - podani so vektor **c**=[c_j] (koeficien… **b**=[b_i] in matrika A=[a_{ij}] (koeficien… …n; j=1,...,n
  - iščemo vektor **x**=[x_j], j=1,...,n, ki … omejitvah **Ax**≤**b** in **x**≥0 (vsak element **x** je ≥0)*

Maksimiziraj

$\mathbf{c}^T\mathbf{x}$

ob pogoju

Ax≤**b** in

x≥0

- LP ni v standardni obliki, če:
  - iščemo rešitev, ki minimizira $\mathbf{c}^T\mathbf{x}$
  - nekatere spremenljivke nimajo omejitve nenegativnosti
  - obstajajo omejitve enakosti
  - obstajajo omejitve neenakosti z znakom ≥ namesto ≤

Standardna oblika LP

- standardna oblika LP:
  - podani so vektor $\mathbf{c}=[c_j]$ (koeficienti krit. fun.), vektor $\mathbf{b}=[b_i]$ in matrika $A=[a_{ij}]$ (koeficienti omejitev), $i=1,...,m$; $j=1,...,n$
  - iščemo vektor $\mathbf{x}=[x_j]$, $j=1,...,n$, ki <u>maksimizira</u> $\mathbf{c}^T\mathbf{x}$ ob omejitvah $\mathbf{Ax}\leq\mathbf{b}$ in $\mathbf{x}\geq 0$ (vsak element $\mathbf{x}$ je ≥ 0)*
- LP ni v standardni obliki, če:
  - iščemo rešitev, ki <u>minimizira</u> $\mathbf{c}^T\mathbf{x}$
  - nekatere spremenljivke nimajo omejitve nenegativnosti
  - obstajajo omejitve enakosti
  - obstajajo omejitve neenakosti z znakom $\geq$ namesto $\leq$

Pretvorba v standardno obliko

Minimiziraj

$$3x_1 - 2x_2$$

ob pogoju

$$x_1 + 4x_2 = 10$$

$$x_1 - x_2 \leq 5$$

$$x_1 \geq 0$$

Pretvorba v standardno obliko

- minimizacijski problem pretvorimo v maksimizacijskega s spremembo predznaka koeficientov **c***

Minimiziraj

$$3x_1 - 2x_2$$

ob pogoju

$$x_1 + 4x_2 = 10$$

$$x_1 - x_2 \leq 5$$

$$x_1 \geq 0$$

Pretvorba v standardno obliko

- minimizacijski problem pretvorimo v maksimizacijskega s spremembo predznaka koeficientov **c***

Maksimiziraj

$$-3x_1 + 2x_2$$

ob pogoju

$$x_1 + 4x_2 = 10$$

$$x_1 - x_2 \leq 5$$

$$x_1 \geq 0$$

Pretvorba v standardno obliko

- minimizacijski problem pretvorimo v maksimizacijskega s spremembo predznaka koeficientov **c***
- vsako spremenljivko x_i brez omejitve nenegativnosti zamenjamo z $x_i' - x_i''$ in dodamo pogoja $x_i' \geq 0$ ter $x_i'' \geq 0$*

Maksimiziraj

$$-3x_1 + 2x_2$$

ob pogoju

$$x_1 + 4x_2 = 10$$

$$x_1 - x_2 \leq 5$$

$$x_1 \geq 0$$

Pretvorba v standardno obliko

- minimizacijski problem pretvorimo v maksimizacijskega s spremembo predznaka koeficientov **c***
- vsako spremenljivko x_i brez omejitve nenegativnosti zamenjamo z $x_i' - x_i''$ in dodamo pogoja $x_i' \geq 0$ ter $x_i'' \geq 0$*

Maksimiziraj

$$-3x_1 + 2x_2' - 2x_2''$$

ob pogoju

$$x_1 + 4x_2' - 4x_2'' = 10$$

$$x_1 - x_2' + x_2'' \leq 5$$

$$x_1, x_2', x_2'' \geq 0$$

Pretvorba v standardno obliko

- minimizacijski problem pretvorimo v maksimizacijskega s spremembo predznaka koeficientov **c***
- vsako spremenljivko x_i brez omejitve nenegativnosti zamenjamo z $x_i' - x_i''$ in dodamo pogoja $x_i' \geq 0$ ter $x_i'' \geq 0$*
- vsako omejitev enakosti $f(\mathbf{x})=b$ nadomestimo z neenakostima $f(\mathbf{x}) \geq b$ in $f(\mathbf{x}) \leq b$*

Maksimiziraj

$$-3x_1 + 2x_2' - 2x_2''$$

ob pogoju

$$x_1 + 4x_2' - 4x_2'' = 10$$

$$x_1 - x_2' + x_2'' \leq 5$$

$$x_1, x_2', x_2'' \geq 0$$

Pretvorba v standardno obliko

- minimizacijski problem pretvorimo v maksimizacijskega s spremembo predznaka koeficientov **c***
- vsako spremenljivko x_i brez omejitve nenegativnosti zamenjamo z $x_i' - x_i''$ in dodamo pogoja $x_i' \geq 0$ ter $x_i'' \geq 0$*
- vsako omejitev enakosti $f(\mathbf{x})=b$ nadomestimo z neenakostima $f(\mathbf{x}) \geq b$ in $f(\mathbf{x}) \leq b$*

Maksimiziraj

$$-3x_1 + 2x_2' - 2x_2''$$

ob pogoju

$$x_1 + 4x_2' - 4x_2'' \geq 10$$
$$x_1 + 4x_2' - 4x_2'' \leq 10$$
$$x_1 - x_2' + x_2'' \leq 5$$
$$x_1, x_2', x_2'' \geq 0$$

Pretvorba v standardno obliko

- minimizacijski problem pretvorimo v maksimizacijskega s spremembo predznaka koeficientov **c***
- vsako spremenljivko x_i brez omejitve nenegativnosti zamenjamo z $x_i' - x_i''$ in dodamo pogoja $x_i' \geq 0$ ter $x_i'' \geq 0$*
- vsako omejitev enakosti $f(\mathbf{x})=b$ nadomestimo z neenakostima $f(\mathbf{x}) \geq b$ in $f(\mathbf{x}) \leq b$*
- vsako omejitev neenakosti oblike $f(\mathbf{x}) \geq b$ pretvorimo v neenakost oblike $f(\mathbf{x}) \leq b$ tako, da jo pomnožimo z -1*

Maksimiziraj

$$-3x_1 + 2x_2' - 2x_2''$$

ob pogoju

$$x_1 + 4x_2' - 4x_2'' \geq 10$$
$$x_1 + 4x_2' - 4x_2'' \leq 10$$
$$x_1 - x_2' + x_2'' \leq 5$$
$$x_1, x_2', x_2'' \geq 0$$

Pretvorba v standardno obliko

- minimizacijski problem pretvorimo v maksimizacijskega s spremembo predznaka koeficientov **c***
- vsako spremenljivko x_i brez omejitve nenegativnosti zamenjamo z $x_i' - x_i''$ in dodamo pogoja $x_i' \geq 0$ ter $x_i'' \geq 0$*
- vsako omejitev enakosti $f(\mathbf{x})=b$ nadomestimo z neenakostima $f(\mathbf{x}) \geq b$ in $f(\mathbf{x}) \leq b$*
- vsako omejitev neenakosti oblike $f(\mathbf{x}) \geq b$ pretvorimo v neenakost oblike $f(\mathbf{x}) \leq b$ tako, da jo pomnožimo z -1*

Maksimiziraj

$$-3x_1 + 2x_2' - 2x_2''$$

ob pogoju

$$\begin{aligned} -x_1 - 4x_2' + 4x_2'' &\leq -10 \\ x_1 + 4x_2' - 4x_2'' &\leq 10 \\ x_1 - x_2' + x_2'' &\leq 5 \\ x_1, x_2', x_2'' &\geq 0 \end{aligned}$$

Ohlapna oblika LP

- dobimo jo s pretvorbo iz standardne oblike LP:
  - vsako omejitev oblike $\sum_{j=1}^{n} a_{ij} x_j \le b_i$ pretvorimo z uvedbo nove spremenljivke x_{n+i} v par omejitev $x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j$ in $x_{n+i} \ge 0$*
  - nove spremenljivke imenujemo **osnovne**, njihove indekse zapišemo v množico B
  - ostale spremenljivke imenujemo **neosnovne**, njihove indekse zapišemo v množico N*
  - velja $|N|=n$, $|B|=m$, $N \cup B = \{1,\ldots,m+n\}$

Ohlapna oblika LP

- dobimo jo s pretvorbo iz standardne oblike LP:
  - vsako omejitev oblike $\sum_{j=1}^{n} a_{ij} x_j \le b_i$ pretvorimo z uvedbo nove spremenljivke x_{n+i} v par omejitev $x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j$ in $x_{n+i} \ge 0$*
  - Maksimiziraj
  -
  - ob pogoju
  -

Maksimiziraj		Maksimiziraj
$-3x_1 + 2x_2 - 2x_3$		$-3x_1 + 2x_2 - 2x_3$
ob pogoju	$\Rightarrow$	ob pogoju
$-x_1 - 4x_2 + x_3 \le -10$		$x_4 = -10 + x_1 + 4x_2 - x_3$
$x_1 - 5x_3 \le 6$		$x_5 = 6 - x_1 + 5x_3$
$x_1, x_2, x_3 \ge 0$		$x_1, x_2, x_3, x_4, x_5 \ge 0$

Ohlapna oblika LP

- dobimo jo s pretvorbo iz standardne oblike LP:
  - vsako omejitev oblike $\sum_{j=1}^{n} a_{ij} x_j \leq b_i$ pretvorimo z uvedbo nove spremenljivke x_{n+i} v par omejitev $x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j$ in $x_{n+i} \geq 0$*
  - nove spremenljivke imenujemo **osnovne**, njihove indekse zapišemo v množico B
  - ostale spremenljivke imenujemo **neosnovne**, njihove indekse zapišemo v množico N*
  - velja $|N|=n$, $|B|=m$, $N \cup B = \{1,\ldots,m+n\}$

Ohlapna oblika LP

- dobim
  - vsa … edbo
    nov … $a_{ij} x_j$
    in x

Maksimiziraj

$$-3x_1+2x_2-2x_3$$

ob pogoju $\Rightarrow$ N={1,2,3}, B={4,5}

$$x_4=-10+x_1+4x_2-x_3$$
$$x_5=6-x_1+5x_3$$
$$x_1,x_2,x_3,x_4,x_5\geq 0$$

  - nove spremenljivke imenujemo **osnovne**, njihove indekse zapišemo v množico B
  - ostale spremenljivke imenujemo **neosnovne**, njihove indekse zapišemo v množico N*
  - velja $|N|=n$, $|B|=m$, $N\cup B=\{1,\ldots,m+n\}$

Ohlapna oblika LP

- kompakten zapis ohlapne oblike:
  - za vrednost kriterijske funkcije uporabimo oznako z,
  - omejitev nenegativnosti pa eksplicitno ne zapisujemo, prav tako ne izpisujemo „maksimiziraj“ in „ob pogoju“
  - kriterijski funkciji dodamo konstantni člen v, iz katerega ob zaključku reševanja razberemo optimalno vrednost kriterijske funkcije*

Ohlapna oblika LP

- kompakten zapis ohlapne oblike:
  - za vrednost kriterijske funkcije uporabimo oznako *z*,
  - omejitev nenegativnosti pa eksplicitno ne zapisujemo, prav tako ne izpisujemo „maksimiziraj“ in „ob pogoju“
  - kriterijski funkciji dodamo konstantni člen *v*, iz katerega ob zaključku reševanja razberemo optimalno vrednost kriterijske funkcije*

Maksimiziraj

$$-3x_1+2x_2-2x_3$$

ob pogoju

$$x_4=-10+x_1+4x_2-x_3$$

$$x_5=6-x_1+5x_3$$

$$x_1,x_2,x_3,x_4,x_5\geq 0$$

$$\Rightarrow$$

$$z=v-3x_1+2x_2-2x_3$$

$$x_4=-10+x_1+4x_2-x_3$$

$$x_5=6-x_1+5x_3$$

Ohlapna oblika LP

- kompakten zapis ohlapne oblike:
  - za vrednost kriterijske funkcije uporabimo oznako z,
  - omejitev nenegativnosti pa eksplicitno ne zapisujemo, prav tako ne izpisujemo „maksimiziraj“ in „ob pogoju“
  - kriterijski funkciji dodamo konstantni člen v, iz katerega ob zaključku reševanja razberemo optimalno vrednost kriterijske funkcije*
  - ohlapna oblika je med postopkom reševanja v vsakem trenutku enolično podana z množico $(N,B,\mathbf{A},\mathbf{b},\mathbf{c},v)$*

Ohlapna oblika LP

$$z = 14 - \frac{x_2}{2} + \frac{x_4}{3} - 2x_5$$

$$x_1 = -1 + \frac{3x_2}{7} - \frac{x_5}{8}$$

$$x_3 = 6 - \frac{5x_2}{4} + x_4 - \frac{2x_5}{3}$$

$$x_6 = 11 - \frac{x_2}{4} - \frac{3x_4}{8} + \frac{4x_5}{11}$$

$\Rightarrow$

$N = \{2,4,5\}$, $B = \{1,3,6\}$, $v = 14$

$\mathbf{b} = [\ b_1\ b_3\ b_6\ ]^T = [\ -1\ \ 6\ \ 11\ ]^T$

$\mathbf{c} = [\ c_2\ c_4\ c_5\ ]^T = [\ -1/2\ \ 1/3\ \ -2\ ]^T$

$$\mathbf{A} = \begin{bmatrix} a_{12} & a_{14} & a_{15} \\ a_{32} & a_{34} & a_{35} \\ a_{62} & a_{64} & a_{65} \end{bmatrix} = \begin{bmatrix} -{}^{3}\!/_{7} & 0 & {}^{1}\!/_{8} \\ {}^{5}\!/_{4} & -1 & {}^{2}\!/_{3} \\ {}^{1}\!/_{4} & {}^{3}\!/_{8} & -{}^{4}\!/_{11} \end{bmatrix}$$

– ohlapna oblika je med postopkom reševanja v vsakem trenutku enolično podana z množico (N,B,$\mathbf{A}$,$\mathbf{b}$,$\mathbf{c}$,v)*

Metoda simpleks

- iterativna metoda, podobna GE (procedura `SIMPLEX`*)
- inicializacija: tvorimo ohlapno obliko LP, ki ima dopustno **osnovno rešitev** (procedura `INITIALIZE_SIMPLEX`*):
  - osnovno rešitev dobimo, če neosnovne spremenljivke postavimo na 0 in izračunamo vrednosti osnovnih sprem.
- v vsaki iteraciji algoritma pretvorimo trenutno ohlapno obliko v ekvivalentno ohlapno obliko (procedura `PIVOT`*)
- vrednost kriterijske funkcije v naslednji iteraciji je večja ali kvečjemu enaka kot v prejšnji iteraciji
- osnovna rešitev zaključne ohlapne oblike je optimalna
- metoda simpleks v najslabšem primeru dela v eksponentnem času, a je v praksi pogosto hitrejša od metod s polinomsko zahtevnostjo

Metoda simpleks

```
SIMPLEX(A,b,c)
    (N,B,A,b,c,v) ← INITIALIZE_SIMPLEX(A,b,c)
    while nek indeks j∈N ima c_j > 0 do
        izberi indeks e∈N, pri katerem je c_e > 0
        for vsak indeks i∈B do
            if a_ie > 0 then
                Δ_i ← b_i / a_ie
            else
                Δ_i ← ∞
        izberi indeks l∈B, ki minimizira Δ_i
        if Δ_l = ∞ then
            return "neomejen"
        else
            (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,e)
    for i ← 1 to n do
        if i∈B then
            x'_i ← b_i
        else
            x'_i ← 0
    z = v
    return (x'_1,x'_2,…,x'_n;v)
```

- iterativn … IMPLEX*)
- inicializ … a dopustno **osnovn** … SIMPLEX*):
  - osno… menljivke posta… vnih sprem.
- v vsaki … ohlapno obliko v … ura PIVOT*)
- vrednos… ji je večja ali kvečjem…
- osnovn… optimalna
- metoda … v ekspone… trejša od metod s…

Metoda simpleks

- iterativna metoda, podobna GE (procedura `SIMPLEX`*)
- inicializacija: tvorimo ohlapno obliko LP, ki ima dopustno **osnovno rešitev** (procedura `INITIALIZE_SIMPLEX`*):
  - osnovno rešitev dobimo, če neosnovne spremenljivke postavimo na 0 in izračunamo vrednosti osnovnih sprem.
- v vsaki iteraciji algoritma pretvorimo trenutno ohlapno obliko v ekvivalentno ohlapno obliko (procedura `PIVOT`*)
- vrednost kriterijske funkcije v naslednji iteraciji je večja ali kvečjemu enaka kot v prejšnji iteraciji
- osnovna rešitev zaključne ohlapne oblike je optimalna
- metoda simpleks v najslabšem primeru dela v eksponentnem času, a je v praksi pogosto hitrejša od metod s polinomsko zahtevnostjo

Metoda simpleks

- iterativna metoda, podobna GE (procedura SIMPLEX*)
- inici … opustno osn … PLEX*):
  - o … jivke p … sprem.
- v vs … apno obli … PIVOT*)
- vred … večja ali kve …
- osn … malna
- met … eksponentnem času, a je v praksi pogosto hitrejša od metod s polinomsko zahtevnostjo

INITIALIZE-SIMPLEX(A,b,c)

  $k \leftarrow$ indeks minimalnega b_i

  if $b_k \geq 0$ then

    return ({1,2,…,n},{n+1,…,n+m},A,b,c,0)

  tvori pomožni linearni program L_{aux}

  (N,B,A,b,c,v) $\leftarrow$ ohlapna oblika za L_{aux}

  $l \leftarrow n+k$

  (N,B,A,b,c,v) $\leftarrow$ PIVOT(N,B,A,b,c,v,l,0)

  iteriraj zanko while v SIMPLEX do optimalne rešitve za L_{aux}

  if osnovna rešitev za L_{aux} je $x_0 = 0$ then

    return ohlapna oblika z originalno krit. f. in odstranjeno x_0

  else

    return “nerešljiv”

Metoda simpleks

- iterativna metoda, podobna GE (procedura `SIMPLEX`*)
- inicializacija: tvorimo ohlapno obliko LP, ki ima dopustno **osnovno rešitev** (procedura `INITIALIZE_SIMPLEX`*):
  - osnovno rešitev dobimo, če neosnovne spremenljivke postavimo na 0 in izračunamo vrednosti osnovnih sprem.
- v vsaki iteraciji algoritma pretvorimo trenutno ohlapno obliko v ekvivalentno ohlapno obliko (procedura `PIVOT`*)
- vrednost kriterijske funkcije v naslednji iteraciji je večja ali kvečjemu enaka kot v prejšnji iteraciji
- osnovna rešitev zaključne ohlapne oblike je optimalna
- metoda simpleks v najslabšem primeru dela v eksponentnem času, a je v praksi pogosto hitrejša od metod s polinomsko zahtevnostjo

Metoda simpleks

- iterativna me … edura `SIMPLEX`*)
- inicializacija: … P, ki ima dopustno **osnovno reš** … `LIZE_SIMPLEX`*):
  - osnovno re … ne spremenljivke postavimo … sti osnovnih sprem.
- v vsaki iterac … renutno ohlapno obliko v ekviv … procedura `PIVOT`*)
- vrednost krite … ji iteraciji je večja ali kvečjemu en…
- osnovna reši … like je optimalna
- metoda simp … ru dela v eksponentne … gosto hitrejša od metod s polin…

PIVOT(N,B,A,b,c,v,l,e)

    $b'_e \leftarrow b_l / a_{le}$
    for vsak $j \in N-\{e\}$ do
        $a'_{ej} \leftarrow a_{lj} / a_{le}$
    $a'_{el} \leftarrow 1 / a_{le}$
    for vsak $i \in B-\{l\}$ do
        $b'_i \leftarrow b_i - a_{ie} b'_e$
        for vsak $j \in N-\{e\}$ do
            $a'_{ij} \leftarrow a_{ij} - a_{ie} a'_{ej}$
        $a'_{il} \leftarrow -a_{ie} a'_{el}$
    $v' \leftarrow v + c_e b'_e$
    for vsak $j \in N-\{e\}$ do
        $c'_j \leftarrow c_j - c_e a'_{ej}$
    $c'_l \leftarrow -c_e a'_{el}$
    $N' = N-\{e\} \cup \{l\}$
    $B' = B-\{l\} \cup \{e\}$
    return (N',B',A',b',c',v')

Metoda simpleks

- iterativna metoda, podobna GE (procedura `SIMPLEX`*)
- inicializacija: tvorimo ohlapno obliko LP, ki ima dopustno **osnovno rešitev** (procedura `INITIALIZE_SIMPLEX`*):
  - osnovno rešitev dobimo, če neosnovne spremenljivke postavimo na 0 in izračunamo vrednosti osnovnih sprem.
- v vsaki iteraciji algoritma pretvorimo trenutno ohlapno obliko v ekvivalentno ohlapno obliko (procedura `PIVOT`*)
- vrednost kriterijske funkcije v naslednji iteraciji je večja ali kvečjemu enaka kot v prejšnji iteraciji
- osnovna rešitev zaključne ohlapne oblike je optimalna
- metoda simpleks v najslabšem primeru dela v eksponentnem času, a je v praksi pogosto hitrejša od metod s polinomsko zahtevnostjo

Metoda simpleks – ideja

- povečanje vrednosti kriterijske funkcije *z* v naslednji iteraciji:
  - izberemo neosnovno spremenljivko x_i, katere povečanje iz 0 poveča tudi vrednost *z* (t.j. koeficient c_i mora biti pozitiven)
  - x_i povečamo za najmanjšo vrednost, pri kateri ena izmed osnovnih spremenljivk x_j postane enaka 0
  - izvedemo pretvorbo ohlapne oblike, pri čemer zamenjamo vlogi x_i (postane osnovna spremenljivka) in x_j (postane neosnovna spremenljivka) – tej operaciji rečemo **pivotiranje**, x_i je **vstopna spremenljivka**, x_j pa **izstopna spremenljivka**

Metoda simpleks – zgled 1

- Naloga: Tisoč evrov želimo investirati v tri oblike naložb, in sicer kratkoročne rizične naložbe A, dolgoročne rizične naložbe B in dolgoročne varne naložbe C. Cene enote naložbe in pričakovani dobički po enoti so v tabeli. Zaradi zavarovanja vrednost rizičnih naložb ne sme presegati 75 % vrednosti varnih naložb. Prav tako morajo zaradi davčnih omejitev dolgoročne naložbe predstavljati vsaj 70 % vseh naložb. Poiščite optimalno strukturo portfelja (št. kupljenih enot posamezne naložbe, ki je lahko tudi realna vrednost), ki maksimizira pričakovani dobiček.

Naložba	Cena enote	Pričakovani dobiček
A	7	6
B	5	4
C	4	2

Metoda simpleks – zgled 1

- spremenljivke:

Naložba	Cena enote	Pričakovani dobiček
A	7	6
B	5	4
C	4	2

Metoda simpleks – zgled 1

- spremenljivke:
  - x_A, x_B, x_C – število kupljenih enot naložbe A, B oz. C

Naložba	Cena enote	Pričakovani dobiček
A	7	6
B	5	4
C	4	2

Metoda simpleks – zgled 1

- spremenljivke:
  - x_A, x_B, x_C – število kupljenih enot naložbe A, B oz. C
- kriterijska funkcija:

Naložba	Cena enote	Pričakovani dobiček
A	7	6
B	5	4
C	4	2

Metoda simpleks – zgled 1

- spremenljivke:
  - x_A, x_B, x_C – število kupljenih enot naložbe A, B oz. C
- kriterijska funkcija:
  - pričakovani dobiček = $6x_A + 4x_B + 2x_C$
  - želimo maksimizirati

Naložba	Cena enote	Pričakovani dobiček
A	7	6
B	5	4
C	4	2

Metoda simpleks – zgled 1

- spremenljivke:
  - x_A, x_B, x_C – število kupljenih enot naložbe A, B oz. C
- kriterijska funkcija:
  - pričakovani dobiček = $6x_A + 4x_B + 2x_C$
  - želimo maksimizirati
- omejitve:

Naložba	Cena enote	Pričakovani dobiček
A	7	6
B	5	4
C	4	2

Metoda simpleks – zgled 1

- spremenljivke:
  - x_A, x_B, x_C – število kupljenih enot naložbe A, B oz. C
- kriterijska funkcija:
  - pričakovani dobiček = $6x_A + 4x_B + 2x_C$
  - želimo maksimizirati
- omejitve:
  - sredstva: $7x_A + 5x_B + 4x_C \leq 1000$

Naložba	Cena enote	Pričakovani dobiček
A	7	6
B	5	4
C	4	2

Metoda simpleks – zgled 1

- spremenljivke:
  - x_A, x_B, x_C – število kupljenih enot naložbe A, B oz. C
- kriterijska funkcija:
  - pričakovani dobiček = $6x_A + 4x_B + 2x_C$
  - želimo maksimizirati
- omejitve:
  - sredstva: $7x_A + 5x_B + 4x_C \leq 1000$
  - zavarovanje: $7x_A + 5x_B \leq 0.75 \cdot 4x_C$

Naložba	Cena enote	Pričakovani dobiček
A	7	6
B	5	4
C	4	2

Metoda simpleks – zgled 1

- spremenljivke:
  - x_A, x_B, x_C – število kupljenih enot naložbe A, B oz. C
- kriterijska funkcija:
  - pričakovani dobiček = $6x_A + 4x_B + 2x_C$
  - želimo maksimizirati
- omejitve:
  - sredstva: $7x_A + 5x_B + 4x_C \leq 1000$
  - zavarovanje: $7x_A + 5x_B \leq 0.75 \cdot 4x_C$
  - davčna: $5x_B + 4x_C \geq 0.7 \cdot (7x_A + 5x_B + 4x_C)$

Naložba	Cena enote	Pričakovani dobiček
A	7	6
B	5	4
C	4	2

Metoda simpleks – zgled 1

- linearni program – originalna oblika:

Maksimiziraj

$$6x_A + 4x_B + 2x_C$$

ob pogoju

$$7x_A + 5x_B + 4x_C \leq 1000$$

$$7x_A + 5x_B \leq 3x_C$$

$$5x_B + 4x_C \geq 4.9x_A + 3.5x_B + 2.8x_C$$

$$x_A, x_B, x_C \geq 0$$

Metoda simpleks – zgled 1

- linearni program – standardna oblika:

Maksimiziraj

$$6x_A + 4x_B + 2x_C$$

ob pogoju

$$7x_A + 5x_B + 4x_C \leq 1000$$
$$7x_A + 5x_B - 3x_C \leq 0$$
$$4.9x_A - 1.5x_B - 1.2x_C \leq 0$$
$$x_A, x_B, x_C \geq 0$$

Metoda simpleks – zgled 1

- linearni program – ohlapna oblika:

$$z = 6x_A + 4x_B + 2x_C$$
$$x_1 = 1000 - 7x_A - 5x_B - 4x_C$$
$$x_2 = -7x_A - 5x_B + 3x_C$$
$$x_3 = -4.9x_A + 1.5x_B + 1.2x_C$$

Metoda simpleks – zgled 1

- linearni program – ohlapna oblika:

$$z = 6x_A + 4x_B + 2x_C$$
$$x_1 = 1000 - 7x_A - 5x_B - 4x_C$$
$$x_2 = -7x_A - 5x_B + 3x_C$$
$$x_3 = -4.9x_A + 1.5x_B + 1.2x_C$$

N={A,B,C}, B={1,2,3}, v=0

$\mathbf{b}$=[1000 0 0]^T, $\mathbf{c}$=[6 4 2]^T

$$\mathbf{A} = \begin{bmatrix} 7 & 5 & 4 \\ 7 & 5 & -3 \\ 4.9 & -1.5 & -1.2 \end{bmatrix}$$

Metoda simpleks – zgled 1

- linearni program – ohlapna oblika:

$$z = 6x_A + 4x_B + 2x_C$$
$$x_1 = 1000 - 7x_A - 5x_B - 4x_C$$
$$x_2 = -7x_A - 5x_B + 3x_C$$
$$x_3 = -4.9x_A + 1.5x_B + 1.2x_C$$

N={A,B,C}, B={1,2,3}, v=0

$\mathbf{b}=[1000\ 0\ 0]^T$, $\mathbf{c}=[6\ 4\ 2]^T$

$$\mathbf{A}=\begin{bmatrix} 7 & 5 & 4 \\ 7 & 5 & -3 \\ 4.9 & -1.5 & -1.2 \end{bmatrix}$$

- osnovna rešitev ohlapne oblike je $\mathbf{x} = (x_A,x_B,x_C,x_1,x_2,x_3) = (0,0,0,1000,0,0)$ in **je dopustna**

Metoda simpleks – zgled 1

$$z = 6x_A + 4x_B + 2x_C$$

$$x_1 = 1000 – 7x_A – 5x_B – 4x_C$$

$$x_2 = -7x_A – 5x_B + 3x_C$$

$$x_3 = -4.9x_A + 1.5x_B + 1.2x_C$$

Metoda simpleks – zgled 1

$$z = 6x_A + 4x_B + 2x_C$$

$$x_1 = 1000 – 7x_A – 5x_B – 4x_C$$

$$x_2 = -7x_A – 5x_B + 3x_C$$

$$x_3 = -4.9x_A + 1.5x_B + 1.2x_C$$

- izberemo neosnovno spremenljivko, katere povečanje poveča z

Metoda simpleks – zgled 1

$$z = 6x_A + 4x_B + 2x_C$$
$$x_1 = 1000 - 7x_A - 5x_B - 4x_C$$
$$x_2 = -7x_A - 5x_B + 3x_C$$
$$x_3 = -4.9x_A + 1.5x_B + 1.2x_C$$

- izberemo neosnovno spremenljivko, katere povečanje poveča z
- izberemo lahko sicer tudi x_A ali x_B, ampak bi njuno povečanje povzročilo kršitev omejitev (ugotovili bi, da ju smemo „povečati“ za največ 0)

Metoda simpleks – zgled 1

$$z = 6x_A + 4x_B + 2x_C$$
$$x_1 = 1000 – 7x_A – 5x_B – 4x_C$$
$$x_2 = -7x_A – 5x_B + 3x_C$$
$$x_3 = -4.9x_A + 1.5x_B + 1.2x_C$$

- izberemo neosnovno spremenljivko, katere povečanje poveča z
- izberemo lahko sicer tudi x_A ali x_B, ampak bi njuno povečanje povzročilo kršitev omejitev (ugotovili bi, da ju smemo „povečati“ za največ 0)
- izberimo torej x_C, ki postane vstopna spremenljivka

Metoda simpleks – zgled 1

$$z = 6x_A + 4x_B + 2x_C$$
$$x_1 = 1000 – 7x_A – 5x_B – 4x_C$$
$$x_2 = -7x_A – 5x_B + 3x_C$$
$$x_3 = -4.9x_A + 1.5x_B + 1.2x_C$$

- izberemo neosnovno spremenljivko, katere povečanje poveča z
- izberemo lahko sicer tudi x_A ali x_B, ampak bi njuno povečanje povzročilo kršitev omejitev (ugotovili bi, da ju smemo „povečati“ za največ 0)
- izberimo torej x_C, ki postane vstopna spremenljivka
- omejitev nenegativnosti za x_1 omejuje povečanje x_C na 250, omejitvi za x_2 in x_3 pa povečanja x_C ne omejujeta

Metoda simpleks – zgled 1

$$z = 6x_A + 4x_B + 2x_C$$
$$x_1 = 1000 – 7x_A – 5x_B – 4x_C$$
$$x_2 = -7x_A – 5x_B + 3x_C$$
$$x_3 = -4.9x_A + 1.5x_B + 1.2x_C$$

- izberemo neosnovno spremenljivko, katere povečanje poveča z
- izberemo lahko sicer tudi x_A ali x_B, ampak bi njuno povečanje povzročilo kršitev omejitev (ugotovili bi, da ju smemo „povečati“ za največ 0)
- izberimo torej x_C, ki postane vstopna spremenljivka
- omejitev nenegativnosti za x_1 omejuje povečanje x_C na 250, omejitvi za x_2 in x_3 pa povečanja x_C ne omejujeta
- x_1 predstavlja najtesnejšo omejitev in zato postane izstopna spremenljivka

Metoda simpleks – zgled 1

$$z = 6x_A + 4x_B + 2x_C$$
$$x_1 = 1000 - 7x_A - 5x_B - 4x_C$$
$$x_2 = -7x_A - 5x_B + 3x_C$$
$$x_3 = -4.9x_A + 1.5x_B + 1.2x_C$$

- najtesnejšo omejitev preoblikujemo tako, da izrazimo vstopno spremenljivko:

$$x_C = 250 - \frac{7}{4} x_A - \frac{5}{4} x_B - \frac{1}{4} x_1 = 250 - 1.75\, x_A - 1.25\, x_B - 0.25\, x_1$$

Metoda simpleks – zgled 1

$$z = 6x_A + 4x_B + 2x_C$$
$$x_1 = 1000 - 7x_A - 5x_B - 4x_C$$
$$x_2 = -7x_A - 5x_B + 3x_C$$
$$x_3 = -4.9x_A + 1.5x_B + 1.2x_C$$

- najtesnejšo omejitev preoblikujemo tako, da izrazimo vstopno spremenljivko:

$$x_C = 250 - \frac{7}{4}x_A - \frac{5}{4}x_B - \frac{1}{4}x_1 = 250 - 1.75\,x_A - 1.25\,x_B - 0.25\,x_1$$

- dobljen izraz za x_C vstavimo v enačbo za kriterijsko funkcijo in enačbe ostalih omejitev; rezultat pivotiranja je nova ohlapna oblika

Metoda simpleks – zgled 1

$$z = 500 + 2.5x_A + 1.5x_B – 0.5x_1$$
$$x_C = 250 – 1.75x_A – 1.25x_B – 0.25x_1$$
$$x_2 = 750 – 12.25x_A – 8.75x_B – 0.75x_1$$
$$x_3 = 300 – 7x_A – 0.3x_1$$

- najtesnejšo omejitev preoblikujemo tako, da izrazimo vstopno spremenljivko:

$$x_C = 250 - \frac{7}{4}x_A - \frac{5}{4}x_B - \frac{1}{4}x_1 = 250 - 1.75\,x_A - 1.25\,x_B - 0.25\,x_1$$

- dobljen izraz za x_C vstavimo v enačbo za kriterijsko funkcijo in enačbe ostalih omejitev; rezultat pivotiranja je nova ohlapna oblika

Metoda simpleks – zgled 1

$$z = 500 + 2.5x_A + 1.5x_B - 0.5x_1$$
$$x_C = 250 - 1.75x_A - 1.25x_B - 0.25x_1$$
$$x_2 = 750 - 12.25x_A - 8.75x_B - 0.75x_1$$
$$x_3 = 300 - 7x_A - 0.3x_1$$

- najtesnejšo omejitev preoblikujemo tako, da izrazimo vstopno spremenljivko:

$$x_C = 250 - \frac{7}{4}x_A - \frac{5}{4}x_B - \frac{1}{4}x_1 = 250 - 1.75\,x_A - 1.25\,x_B - 0.25\,x_1$$

- dobljen izraz za x_C vstavimo v enačbo za kriterijsko funkcijo in enačbe ostalih omejitev; rezultat pivotiranja je nova ohlapna oblika
- osnovna rešitev nove ohlapne oblike je $\mathbf{x} = (x_A,x_B,x_C,x_1,x_2,x_3) = (0,0,250,0,750,300)$ in ima vrednost $z = 500$

Metoda simpleks – zgled 1

$$z = 500 + 2.5x_A + 1.5x_B - 0.5x_1$$
$$x_C = 250 - 1.75x_A - 1.25x_B - 0.25x_1$$
$$x_2 = 750 - 12.25x_A - 8.75x_B - 0.75x_1$$
$$x_3 = 300 - 7x_A - 0.3x_1$$

- v naslednji iteraciji izberimo npr. x_A kot vstopno spremenljivko (edina alternativna izbira je x_B)

Metoda simpleks – zgled 1

$$z = 500 + 2.5x_A + 1.5x_B – 0.5x_1$$
$$x_C = 250 – 1.75x_A – 1.25x_B – 0.25x_1$$
$$x_2 = 750 – 12.25x_A – 8.75x_B – 0.75x_1$$
$$x_3 = 300 – 7x_A – 0.3x_1$$

- v naslednji iteraciji izberimo npr. x_A kot vstopno spremenljivko (edina alternativna izbira je x_B)
- omejitev x_C dopušča povečanje x_A za 250/1.75=142.857, omejitev x_2 za 750/12.25=61.224 in omejitev x_3 za 300/7=42.857

Metoda simpleks – zgled 1

$$z = 500 + 2.5x_A + 1.5x_B – 0.5x_1$$
$$x_C = 250 – 1.75x_A – 1.25x_B – 0.25x_1$$
$$x_2 = 750 – 12.25x_A – 8.75x_B – 0.75x_1$$
$$x_3 = 300 – 7x_A – 0.3x_1$$

- v naslednji iteraciji izberimo npr. x_A kot vstopno spremenljivko (edina alternativna izbira je x_B)
- omejitev x_C dopušča povečanje x_A za 250/1.75=142.857, omejitev x_2 za 750/12.25=61.224 in omejitev x_3 za 300/7=42.857
- najtesnejša omejitev je x_3, ki postane izstopna spremenljivka

Metoda simpleks – zgled 1

$$z = 500 + 2.5x_A + 1.5x_B - 0.5x_1$$
$$x_C = 250 - 1.75x_A - 1.25x_B - 0.25x_1$$
$$x_2 = 750 - 12.25x_A - 8.75x_B - 0.75x_1$$
$$x_3 = 300 - 7x_A - 0.3x_1$$

- v naslednji iteraciji izberimo npr. x_A kot vstopno spremenljivko (edina alternativna izbira je x_B)
- omejitev x_C dopušča povečanje x_A za 250/1.75=142.857, omejitev x_2 za 750/12.25=61.224 in omejitev x_3 za 300/7=42.857
- najtesnejša omejitev je x_3, ki postane izstopna spremenljivka
- izrazimo x_A iz omejitve za x_3: $x_A = 42.857 - 0.043x_1 - 0.143x_3$

Metoda simpleks – zgled 1

$$z = 607.143 - 1.25x_1 - 0.357x_3 + 1.5x_B$$
$$x_C = 175 + 0.275x_1 + 0.25x_3 - 1.25x_B$$
$$x_2 = 225 + 2.295x_1 + 1.75x_3 - 8.75x_B$$
$$x_A = 42.857 - 0.43x_1 - 0.143x_3$$

- v naslednji iteraciji izberimo npr. x_A kot vstopno spremenljivko (edina alternativna izbira je x_B)
- omejitev x_C dopušča povečanje x_A za 250/1.75=142.857, omejitev x_2 za 750/12.25=61.224 in omejitev x_3 za 300/7=42.857
- najtesnejša omejitev je x_3, ki postane izstopna spremenljivka
- izrazimo x_A iz omejitve za x_3: $x_A = 42.857 - 0.043x_1 - 0.143x_3$
- vstavimo novo enačbo v ostale in dobimo ohlapno obliko, katere osnovna rešitev $\mathbf{x} = (42.857,0,175,0,225,0)$ ima vrednost $z = 607.143$

Metoda simpleks – zgled 1

$$z = 607.143 - 1.25x_1 - 0.357x_3 + 1.5x_B$$
$$x_C = 175 + 0.275x_1 + 0.25x_3 - 1.25x_B$$
$$x_2 = 225 + 2.295x_1 + 1.75x_3 - 8.75x_B$$
$$x_A = 42.857 - 0.43x_1 - 0.143x_3$$

- v zadnji iteraciji izberimo kot vstopno spremenljivko x_B, ki edina še lahko poveča vrednost z

Metoda simpleks – zgled 1

$$z = 607.143 - 1.25x_1 - 0.357x_3 + 1.5x_B$$
$$x_C = 175 + 0.275x_1 + 0.25x_3 - 1.25x_B$$
$$x_2 = 225 + 2.295x_1 + 1.75x_3 - 8.75x_B$$
$$x_A = 42.857 - 0.43x_1 - 0.143x_3$$

- v zadnji iteraciji izberimo kot vstopno spremenljivko x_B, ki edina še lahko poveča vrednost z
- omejitev x_C dopušča povečanje x_B za 175/1.25=140, omejitev x_2 za 225/8.75=25.714 in omejitev x_A za ∞

Metoda simpleks – zgled 1

$$z = 607.143 - 1.25x_1 - 0.357x_3 + 1.5x_B$$
$$x_C = 175 + 0.275x_1 + 0.25x_3 - 1.25x_B$$
$$x_2 = 225 + 2.295x_1 + 1.75x_3 - 8.75x_B$$
$$x_A = 42.857 - 0.43x_1 - 0.143x_3$$

- v zadnji iteraciji izberimo kot vstopno spremenljivko x_B, ki edina še lahko poveča vrednost z
- omejitev x_C dopušča povečanje x_B za 175/1.25=140, omejitev x_2 za 225/8.75=25.714 in omejitev x_A za ∞
- najtesnejša omejitev je x_2, ki postane izstopna spremenljivka

Metoda simpleks – zgled 1

$$z = 607.143 - 1.25x_1 - 0.357x_3 + 1.5x_B$$
$$x_C = 175 + 0.275x_1 + 0.25x_3 - 1.25x_B$$
$$x_2 = 225 + 2.295x_1 + 1.75x_3 - 8.75x_B$$
$$x_A = 42.857 - 0.43x_1 - 0.143x_3$$

- v zadnji iteraciji izberimo kot vstopno spremenljivko x_B, ki edina še lahko poveča vrednost z
- omejitev x_C dopušča povečanje x_B za 175/1.25=140, omejitev x_2 za 225/8.75=25.714 in omejitev x_A za ∞
- najtesnejša omejitev je x_2, ki postane izstopna spremenljivka
- izrazimo x_B iz omejitve za x_2:

$$x_B = 25.714 + 0.262x_1 - 0.114x_2 + 0.2x_3$$

Metoda simpleks – zgled 1

$$z = 645.714 - 0.857x_1 - 0.171x_2 - 0.057x_3$$
$$x_C = 142.857 - 0.052x_1 + 0.142x_2$$
$$x_B = 25.714 + 0.262x_1 - 0.114x_2 + 0.2x_3$$
$$x_A = 42.857 - 0.43x_1 - 0.143x_3$$

- v zadnji iteraciji izberimo kot vstopno spremenljivko x_B, ki edina še lahko poveča vrednost z
- omejitev x_C dopušča povečanje x_B za 175/1.25=140, omejitev x_2 za 225/8.75=25.714 in omejitev x_A za ∞
- najtesnejša omejitev je x_2, ki postane izstopna spremenljivka
- izrazimo x_B iz omejitve za x_2:

$$x_B = 25.714 + 0.262x_1 - 0.114x_2 + 0.2x_3$$

- nova ohlapna oblika ima osnovno rešitev $\mathbf{x}$ = (42.857,25.714,142.857,0,0,0) in vrednost z = 645.714

Metoda simpleks – zgled 1

$$z = 645.714 - 0.857x_1 - 0.171x_2 - 0.057x_3$$
$$x_C = 142.857 - 0.052x_1 + 0.142x_2$$
$$x_B = 25.714 + 0.262x_1 - 0.114x_2 + 0.2x_3$$
$$x_A = 42.857 - 0.43x_1 - 0.143x_3$$

- vrednosti kriterijske funkcije v zadnji ohlapni obliki več ne moremo povečati s povečanjem nobene neosnovne spremenljivke
- optimalna rešitev problema je nakup 42.857 enot naložbe A, 25.714 enot naložbe B in 142.857 enot naložbe C

Metoda simpleks – zgled 2

- Imejmo naslednji LP v ohlapni obliki:

$z = 2x_1 - x_2$
$x_3 = 2 - 2x_1 + x_2$
$x_4 = -4 - x_1 + 5x_2$

$N=\{1,2\}$, $B=\{3,4\}$, $v=0$
$\mathbf{b}=[2\ \ -4]^T$
$\mathbf{c}=[2\ \ -1]^T$

$$\mathbf{A}=\begin{bmatrix} 2 & -1 \\ 1 & -5 \end{bmatrix}$$

- osnovna rešitev ohlapne oblike je $\mathbf{x} = (x_1,x_2,x_3,x_4) = (0,0,2,-4)$ in **ni dopustna** (x_4 krši omejitev nenegativn.)
- za pravilno inicializacijo ohlapne oblike LP poskrbi procedura `INITIALIZE_SIMPLEX`

Inicializacija simpleksa – zgled 2

- procedura `INITIALIZE_SIMPLEX` kot vhod sprejme standardno obliko LP in ugotovi, ali je LP rešljiv
- če je LP rešljiv, procedura vrne ohlapno obliko z dopustno osnovno rešitvijo, ki predstavlja izhodišče za izvajanje metode `SIMPLEX`
- če vhodni LP nima ohlapne oblike z dopustno osnovno rešitvijo, jo poiščemo s *pomožnim LP* (auxiliary LP):
  - uvedemo dodatno spremenljivko x_0
  - kriterijsko funkcijo postavimo na $z = -x_0$
  - desni strani vsake omejitve v ohlapni obliki LP dodamo člen $+x_0$*
- originalni LP je rešljiv natanko tedaj, ko ima optimalna rešitev pomožnega LP vrednost $z=0$

Inicializacija simpleksa – zgled 2

- procedu… kot vhod sprejme standard… e LP rešljiv
- če je LP… ono obliko z dopustno osnovno… dišče za izvajanje metode SIMPLEX
- če vhodni LP nima ohlapne oblike z dopustno osnovno rešitvijo, jo poiščemo s *pomožnim LP* (auxiliary LP):
  - uvedemo dodatno spremenljivko x_0
  - kriterijsko funkcijo postavimo na $z = -x_0$
  - desni strani vsake omejitve v ohlapni obliki LP dodamo člen $+x_0$*
- originalni LP je rešljiv natanko tedaj, ko ima optimalna rešitev pomožnega LP vrednost z=0

Originalni LP (ohlapna oblika):

$$z = 2x_1 - x_2$$
$$x_3 = 2 - 2x_1 + x_2$$
$$x_4 = -4 - x_1 + 5x_2$$

Inicializacija simpleksa – zgled 2

- procedu … kot vhod sprejme standard … e LP rešljiv
- če je LP … pno obliko z dopustno osnovno … dišče za izvajanje metode …

Originalni LP (ohlapna oblika):

$$z = 2x_1 – x_2$$
$$x_3 = 2 – 2x_1 + x_2$$
$$x_4 = -4 – x_1 + 5x_2$$

- če vhodni LP nima ohlapne oblike z dopustno osnovno rešitvijo, … *LP* (auxiliary LP):
  - uvede … x_0
  - kriteri … $z = -x_0$
  - desni … apni obliki LP dodamo člen $+x_0$*

Pomožni LP (ohlapna oblika):

$$z = -x_0$$
$$x_3 = 2 – 2x_1 + x_2 + x_0$$
$$x_4 = -4 – x_1 + 5x_2 + x_0$$

- originalni LP je rešljiv natanko tedaj, ko ima optimalna rešitev pomožnega LP vrednost z=0

Inicializacija simpleksa – zgled 2

- na pomožnem LP izvedemo pivotiranje z x_0 kot vstopno in x_{n+k} kot izstopno spremenljivko, kjer je k indeks pri katerem je vrednost b_k najmanjša

$$
\begin{aligned}
z &= -x_0 \\
x_3 &= 2 - 2x_1 + x_2 + x_0 \\
x_4 &= -4 - x_1 + 5x_2 + x_0
\end{aligned}
$$

Inicializacija simpleksa – zgled 2

- na pomožnem LP izvedemo pivotiranje z x_0 kot vstopno in x_{n+k} kot izstopno spremenljivko, kjer je k indeks pri katerem je vrednost b_k najmanjša

$$
\begin{aligned}
z &= -x_0 \\
x_3 &= 2 - 2x_1 + x_2 + x_0 \\
x_4 &= -4 - x_1 + 5x_2 + x_0
\end{aligned}
$$

- najnižji b_k je pri k=2 (b_2=-4), zato pivotiramo z x_0 in x_{2+2}=x_4

Inicializacija simpleksa – zgled 2

- na pomožnem LP izvedemo pivotiranje z x_0 kot vstopno in x_{n+k} kot izstopno spremenljivko, kjer je k indeks pri katerem je vrednost b_k najmanjša

$$
\begin{aligned}
z &= -x_0 \\
x_3 &= 2 - 2x_1 + x_2 + x_0 \\
x_4 &= -4 - x_1 + 5x_2 + x_0
\end{aligned}
$$

- najnižji b_k je pri k=2 (b_2=-4), zato pivotiramo z x_0 in x_{2+2}=x_4
- $x_0 = 4 + x_1 - 5x_2 + x_4$

Inicializacija simpleksa – zgled 2

- na pomožnem LP izvedemo pivotiranje z x_0 kot vstopno in x_{n+k} kot izstopno spremenljivko, kjer je k indeks pri katerem je vrednost b_k najmanjša

$$
\begin{aligned}
z &= -4 - x_1 + 5x_2 - x_4 \\
x_0 &= 4 + x_1 - 5x_2 + x_4 \\
x_3 &= 6 - x_1 - 4x_2 + x_4
\end{aligned}
$$

- najnižji b_k je pri k=2 (b_2=-4), zato pivotiramo z x_0 in x_{2+2}=x_4
- $x_0 = 4 + x_1 - 5x_2 + x_4$

Inicializacija simpleksa – zgled 2

- na pomožnem LP izvedemo pivotiranje z x_0 kot vstopno in x_{n+k} kot izstopno spremenljivko, kjer je k indeks pri katerem je vrednost b_k najmanjša

$$
\begin{aligned}
z &= -4 - x_1 + 5x_2 - x_4 \\
x_0 &= 4 + x_1 - 5x_2 + x_4 \\
x_3 &= 6 - x_1 - 4x_2 + x_4
\end{aligned}
$$

- najnižji b_k je pri k=2 (b_2=-4), zato pivotiramo z x_0 in x_{2+2}=x_4
- $x_0 = 4 + x_1 - 5x_2 + x_4$
- osnovna rešitev dobljene ohlapne oblike je vedno dopustna; $\mathbf{x}$=(4,0,0,6,0), z=-4

Inicializacija simpleksa – zgled 2

- na dobljeni ohlapni obliki pomožnega LP izvedemo metodo simpleks, da najdemo optimalno rešitev

$$
\begin{aligned}
z &= -4 - x_1 + 5x_2 - x_4 \\
x_0 &= 4 + x_1 - 5x_2 + x_4 \\
x_3 &= 6 - x_1 - 4x_2 + x_4
\end{aligned}
$$

Inicializacija simpleksa – zgled 2

- na dobljeni ohlapni obliki pomožnega LP izvedemo metodo simpleks, da najdemo optimalno rešitev

$$
\begin{aligned}
z &= -4 - x_1 + 5x_2 - x_4 \\
x_0 &= 4 + x_1 - 5x_2 + x_4 \\
x_3 &= 6 - x_1 - 4x_2 + x_4
\end{aligned}
$$

- kot vstopno spremenljivko izberemo x_2

Inicializacija simpleksa – zgled 2

- na dobljeni ohlapni obliki pomožnega LP izvedemo metodo simpleks, da najdemo optimalno rešitev

$$z = -4 - x_1 + 5x_2 - x_4$$
$$x_0 = 4 + x_1 - 5x_2 + x_4$$
$$x_3 = 6 - x_1 - 4x_2 + x_4$$

- kot vstopno spremenljivko izberemo x_2
- omejitev pri x_0 dopušča povečanje x_2 za 4/5=0.8, omejitev pri x_3 pa za 6/4=1.5

Inicializacija simpleksa – zgled 2

- na dobljeni ohlapni obliki pomožnega LP izvedemo metodo simpleks, da najdemo optimalno rešitev

$$
\begin{aligned}
z &= -4 - x_1 + 5x_2 - x_4 \\
x_0 &= 4 + x_1 - 5x_2 + x_4 \\
x_3 &= 6 - x_1 - 4x_2 + x_4
\end{aligned}
$$

- kot vstopno spremenljivko izberemo x_2
- omejitev pri x_0 dopušča povečanje x_2 za 4/5=0.8, omejitev pri x_3 pa za 6/4=1.5
- najtesnejša omejitev je x_0, ki postane izstopna spremenljivka

Inicializacija simpleksa – zgled 2

- na dobljeni ohlapni obliki pomožnega LP izvedemo metodo simpleks, da najdemo optimalno rešitev

$$z = -4 - x_1 + 5x_2 - x_4$$
$$x_0 = 4 + x_1 - 5x_2 + x_4$$
$$x_3 = 6 - x_1 - 4x_2 + x_4$$

- kot vstopno spremenljivko izberemo x_2
- omejitev pri x_0 dopušča povečanje x_2 za 4/5=0.8, omejitev pri x_3 pa za 6/4=1.5
- najtesnejša omejitev je x_0, ki postane izstopna spremenljivka
- $x_2 = 0.8 - 0.2x_0 + 0.2x_1 + 0.2x_4$

Inicializacija simpleksa – zgled 2

- na dobljeni ohlapni obliki pomožnega LP izvedemo metodo simpleks, da najdemo optimalno rešitev

$$
\begin{aligned}
z &= -x_0 \\
x_2 &= 0.8 - 0.2x_0 + 0.2x_1 + 0.2x_4 \\
x_3 &= 2.8 + 0.8x_0 - 1.8x_1 + 0.2x_4
\end{aligned}
$$

- kot vstopno spremenljivko izberemo x_2
- omejitev pri x_0 dopušča povečanje x_2 za 4/5=0.8, omejitev pri x_3 pa za 6/4=1.5
- najtesnejša omejitev je x_0, ki postane izstopna spremenljivka
- $x_2 = 0.8 - 0.2x_0 + 0.2x_1 + 0.2x_4$

Inicializacija simpleksa – zgled 2

- na dobljeni ohlapni obliki pomožnega LP izvedemo metodo simpleks, da najdemo optimalno rešitev

$$z = -x_0$$
$$x_2 = 0.8 – 0.2x_0 + 0.2x_1 + 0.2x_4$$
$$x_3 = 2.8 + 0.8x_0 – 1.8x_1 + 0.2x_4$$

- dodatno povečanje z ni več mogoče, zato je optimalna rešitev pomožnega LP enaka **x**=(0,0,0.8,2.8,0) in ima vrednost z=0 ⇒ originalni LP je rešljiv

Inicializacija simpleksa – zgled 2

- na dobljeni ohlapni obliki pomožnega LP izvedemo metodo simpleks, da najdemo optimalno rešitev

$$z = -x_0$$
$$x_2 = 0.8 - 0.2x_0 + 0.2x_1 + 0.2x_4$$
$$x_3 = 2.8 + 0.8x_0 - 1.8x_1 + 0.2x_4$$

- tvorimo začetno ohlapno obliko za metodo simpleks:
  - iz zadnje ohlapne oblike za pomožni LP odstranimo x_0 (uporabimo x_0=0) in povrnemo originalno kriterijsko funkcijo*

Inicializacija simpleksa – zgled 2

- na dobljeni ohlapni obliki pomožnega LP izvedemo metodo simpleks, da najdemo optimalno rešitev

$$z = 2x_1 - x_2$$
$$x_2 = 0.8 + 0.2x_1 + 0.2x_4$$
$$x_3 = 2.8 - 1.8x_1 + 0.2x_4$$

- tvorimo začetno ohlapno obliko za metodo simpleks:
  - iz zadnje ohlapne oblike za pomožni LP odstranimo x_0 (uporabimo x_0=0) in povrnemo originalno kriterijsko funkcijo*

Inicializacija simpleksa – zgled 2

- na dobljeni ohlapni obliki pomožnega LP izvedemo metodo simpleks, da najdemo optimalno rešitev

$$z = 2x_1 - x_2$$
$$x_2 = 0.8 + 0.2x_1 + 0.2x_4$$
$$x_3 = 2.8 - 1.8x_1 + 0.2x_4$$

- tvorimo začetno ohlapno obliko za metodo simpleks:
  - iz zadnje ohlapne oblike za pomožni LP odstranimo x_0 (uporabimo x_0=0) in povrnemo originalno kriterijsko funkcijo*
  - po potrebi v enačbo kriterijske funkcije vstavimo enačbe osnovnih spremenljivk*

Inicializacija simpleksa – zgled 2

- na dobljeni ohlapni obliki pomožnega LP izvedemo metodo simpleks, da najdemo optimalno rešitev

$$z = -0.8 + 1.8x_1 - 0.2x_4$$
$$x_2 = 0.8 + 0.2x_1 + 0.2x_4$$
$$x_3 = 2.8 - 1.8x_1 + 0.2x_4$$

- tvorimo začetno ohlapno obliko za metodo simpleks:
  - iz zadnje ohlapne oblike za pomožni LP odstranimo x_0 (uporabimo x_0=0) in povrnemo originalno kriterijsko funkcijo*
  - po potrebi v enačbo kriterijske funkcije vstavimo enačbe osnovnih spremenljivk*

Inicializacija simpleksa – zgled 2

- na dobljeni ohlapni obliki pomožnega LP izvedemo metodo simpleks, da najdemo optimalno rešitev

$$z = -0.8 + 1.8x_1 - 0.2x_4$$
$$x_2 = 0.8 + 0.2x_1 + 0.2x_4$$
$$x_3 = 2.8 - 1.8x_1 + 0.2x_4$$

- tvorimo začetno ohlapno obliko za metodo simpleks:
  - iz zadnje ohlapne oblike za pomožni LP odstranimo x_0 (uporabimo x_0=0) in povrnemo originalno kriterijsko funkcijo*
  - po potrebi v enačbo kriterijske funkcije vstavimo enačbe osnovnih spremenljivk*
- nova ohlapna oblika ima dopustno osnovno rešitev **x**=(0,0.8,2.8,0) z vrednostjo z=-0.8

Implementacija simpleks

```
SIMPLEX(A,b,c)
    (N,B,A,b,c,v) ← INITIALIZE_SIMPLEX(A,b,c)
    while nek indeks j∈N ima c_j > 0 do
        izberi indeks e∈N, pri katerem je c_e > 0
        for vsak indeks i∈B do
            if a_ie > 0 then
                Δ_i ← b_i / a_ie
            else
                Δ_i ← ∞
        izberi indeks l∈B, ki minimizira Δ_i
        if Δ_l = ∞ then
            return "neomejen"
        else
            (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,e)
    for i ← 1 to n do
        if i∈B then
            x'_i ← b_i
        else
            x'_i ← 0
    z = v
    return (x'_1,x'_2,…,x'_n;v)
```

Implementacija simpleks

```
SIMPLEX(A,b,c)
    (N,B,A,b,c,v) ← INITIALIZE_SIMPLEX(A,b,c)
    while nek indeks j∈N ima c_j > 0 do
        izberi indeks e∈N, pri katerem je c_e > 0
        for vsak indeks i∈B do
            if a_ie > 0 then
                Δ_i ← b_i / a_ie
            else
                Δ_i ← ∞
        izberi indeks l∈B, ki minimizira Δ_i
        if Δ_l = ∞ then
            return “neomejen”
        else
            (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,e)
    for i ← 1 to n do
        if i∈B then
            x'_i ← b_i
        else
            x'_i ← 0
    z = v
    return (x'_1,x'_2,…,x'_n;v)
```

tvorba ohlapne oblike z dopustno osnovno rešitvijo

Implementacija simpleks

```
SIMPLEX(A,b,c)
    (N,B,A,b,c,v) ← INITIALIZE_SIMPLEX(A,b,c)
    while nek indeks j∈N ima c_j > 0 do
        izberi indeks e∈N, pri katerem je c_e > 0
        for vsak indeks i∈B do
            if a_ie > 0 then
                Δ_i ← b_i / a_ie
            else
                Δ_i ← ∞
        izberi indeks l∈B, ki minimizira Δ_i
        if Δ_l = ∞ then
            return "neomejen"
        else
            (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,e)
    for i ← 1 to n do
        if i∈B then
            x'_i ← b_i
        else
            x'_i ← 0
    z = v
    return (x'_1,x'_2,…,x'_n;v)
```

glavna zanka metode

Implementacija simpleks

```
SIMPLEX(A,b,c)
    (N,B,A,b,c,v) ← INITIALIZE_SIMPLEX(A,b,c)
    while nek indeks j∈N ima c_j > 0 do
        izberi indeks e∈N, pri katerem je c_e > 0
        for vsak indeks i∈B do
            if a_ie > 0 then
                Δ_i ← b_i / a_ie
            else
                Δ_i ← ∞
        izberi indeks l∈B, ki minimizira Δ_i
        if Δ_l = ∞ then
            return “neomejen”
        else
            (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,e)
    for i ← 1 to n do
        if i∈B then
            x'_i ← b_i
        else
            x'_i ← 0
    z = v
    return (x'_1,x'_2,…,x'_n;v)
```

izbira vstopne spremenljivke

Implementacija simpleks

```
SIMPLEX(A,b,c)
    (N,B,A,b,c,v) ← INITIALIZE_SIMPLEX(A,b,c)
    while nek indeks j∈N ima c_j > 0 do
        izberi indeks e∈N, pri katerem je c_e > 0
        for vsak indeks i∈B do
            if a_ie > 0 then
                Δ_i ← b_i / a_ie
            else
                Δ_i ← ∞
        izberi indeks l∈B, ki minimizira Δ_i
        if Δ_l = ∞ then
            return “neomejen”
        else
            (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,e)
    for i ← 1 to n do
        if i∈B then
            x'_i ← b_i
        else
            x'_i ← 0
    z = v
    return (x'_1,x'_2,…,x'_n;v)
```

določitev izstopne spremenljivke

Implementacija simpleks

```
SIMPLEX(A,b,c)
    (N,B,A,b,c,v) ← INITIALIZE_SIMPLEX(A,b,c)
    while nek indeks j∈N ima c_j > 0 do
        izberi indeks e∈N, pri katerem je c_e > 0
        for vsak indeks i∈B do
            if a_ie > 0 then
                Δ_i ← b_i / a_ie
            else
                Δ_i ← ∞
        izberi indeks l∈B, ki minimizira Δ_i
        if Δ_l = ∞ then
            return "neomejen"
        else
            (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,e)
    for i ← 1 to n do
        if i∈B then
            x'_i ← b_i
        else
            x'_i ← 0
    z = v
    return (x'_1,x'_2,…,x'_n;v)
```

izstop, če je LP neomejen

Implementacija simpleks

```
SIMPLEX(A,b,c)
    (N,B,A,b,c,v) ← INITIALIZE_SIMPLEX(A,b,c)
    while nek indeks j∈N ima c_j > 0 do
        izberi indeks e∈N, pri katerem je c_e > 0
        for vsak indeks i∈B do
            if a_ie > 0 then
                Δ_i ← b_i / a_ie
            else
                Δ_i ← ∞
        izberi indeks l∈B, ki minimizira Δ_i
        if Δ_l = ∞ then
            return “neomejen”
        else
            (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,e)
    for i ← 1 to n do
        if i∈B then
            x'_i ← b_i
        else
            x'_i ← 0
    z = v
    return (x'_1,x'_2,…,x'_n;v)
```

pivotiranje

Implementacija simpleks

```
SIMPLEX(A,b,c)
    (N,B,A,b,c,v) ← INITIALIZE_SIMPLEX(A,b,c)
    while nek indeks j∈N ima c_j > 0 do
        izberi indeks e∈N, pri katerem je c_e > 0
        for vsak indeks i∈B do
            if a_ie > 0 then
                Δ_i ← b_i / a_ie
            else
                Δ_i ← ∞
        izberi indeks l∈B, ki minimizira Δ_i
        if Δ_l = ∞ then
            return "neomejen"
        else
            (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,e)
    for i ← 1 to n do
        if i∈B then
            x'_i ← b_i
        else
            x'_i ← 0
    z = v
    return (x'_1,x'_2,…,x'_n;v)
```

tvorba optimalne rešitve

Implementacija simpleks

```
SIMPLEX(A,b,c)
    (N,B,A,b,c,v) ← INITIALIZE_SIMPLEX(A,b,c)
    while nek indeks j∈N ima c_j > 0 do
        izberi indeks e∈N, pri katerem je c_e > 0
        for vsak indeks i∈B do
            if a_ie > 0 then
                Δ_i ← b_i / a_ie
            else
                Δ_i ← ∞
        izberi indeks l∈B, ki minimizira Δ_i
        if Δ_l = ∞ then
            return “neomejen”
        else
            (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,e)
    for i ← 1 to n do
        if i∈B then
            x'_i ← b_i
        else
            x'_i ← 0
    z = v
    return (x'_1,x'_2,…,x'_n;v)
```

- obstaja možnost ciklanja med dvema enakovrednima ohlapnima oblikama LP

Implementacija simpleks

```
SIMPLEX(A,b,c)
    (N,B,A,b,c,v) ← INITIALIZE_SIMPLEX(A,b,c)
    while nek indeks j∈N ima c_j > 0 do
        izberi indeks e∈N, pri katerem je c_e > 0
        for vsak indeks i∈B do
            if a_ie > 0 then
                Δ_i ← b_i / a_ie
            else
                Δ_i ← ∞
        izberi indeks l∈B, ki minimizira Δ_i
        if Δ_l = ∞ then
            return “neomejen”
        else
            (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,e)
    for i ← 1 to n do
        if i∈B then
            x'_i ← b_i
        else
            x'_i ← 0
    z = v
    return (x'_1,x'_2,…,x'_n;v)
```

- obstaja možnost ciklanja med dvema enakovrednima ohlapnima oblikama LP
- rešitev: pri izbiri indeksov *e* in *l* med več enakovrednimi možnostmi izberemo najmanjšega

Implementacija simpleks

```
INITIALIZE-SIMPLEX(A,b,c)
   k ← indeks minimalnega b_i
   if b_k ≥ 0 then
      return ({1,2,…,n},{n+1,…,n+m},A,b,c,0)
   tvori pomožni linearni program L_aux
   (N,B,A,b,c,v) ← ohlapna oblika za L_aux
   l ← n+k
   (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,0)
   iteriraj zanko while v SIMPLEX do optimalne rešitve za L_aux
   if osnovna rešitev za L_aux je x_0 = 0 then
      return ohlapna oblika z originalno krit. f. in odstranjeno x_0
   else
      return “nerešljiv”
```

Implementacija simpleks

```
INITIALIZE-SIMPLEX(A,b,c)
   k ← indeks minimalnega b_i
   if b_k ≥ 0 then
      return ({1,2,…,n},{n+1,…,n+m},A,b,c,0)
   tvori pomožni linearni program L_aux
   (N,B,A,b,c,v) ← ohlapna oblika za L_aux
   l ← n+k
   (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,0)
   iteriraj zanko while v SIMPLEX do optimalne rešitve za L_aux
   if osnovna rešitev za L_aux je x_0 = 0 then
      return ohlapna oblika z originalno krit. f. in odstranjeno x_0
   else
      return “nerešljiv”
```

osnovna rešitev je dopustna

Implementacija simpleks

```
INITIALIZE-SIMPLEX(A,b,c)
    k ← indeks minimalnega b_i
    if b_k ≥ 0 then
        return ({1,2,…,n},{n+1,…,n+m},A,b,c,0)
    tvori pomožni linearni program L_aux
    (N,B,A,b,c,v) ← ohlapna oblika za L_aux
    l ← n+k
    (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,0)
    iteriraj zanko while v SIMPLEX do optimalne rešitve za L_aux
    if osnovna rešitev za L_aux je x_0 = 0 then
        return ohlapna oblika z originalno krit. f. in odstranjeno x_0
    else
        return “nerešljiv”
```

rešitev pomožnega LP

Implementacija simpleks

```
INITIALIZE-SIMPLEX(A,b,c)
   k ← indeks minimalnega b_i
   if b_k ≥ 0 then
      return ({1,2,…,n},{n+1,…,n+m},A,b,c,0)
   tvori pomožni linearni program L_aux
   (N,B,A,b,c,v) ← ohlapna oblika za L_aux
   l ← n+k
   (N,B,A,b,c,v) ← PIVOT(N,B,A,b,c,v,l,0)
   iteriraj zanko while v SIMPLEX do optimalne rešitve za L_aux
   if osnovna rešitev za L_aux je x_0 = 0 then
      return ohlapna oblika z originalno krit. f. in odstranjeno x_0
   else
      return “nerešljiv”
```

originalni LP je rešljiv

Implementacija simpleks

```
PIVOT(N,B,A,b,c,v,l,e)
    b'_e ← b_l / a_le
    for vsak j ∈ N−{e} do
        a'_ej ← a_lj / a_le
    a'_el ← 1 / a_le
    for vsak i ∈ B−{l} do
        b'_i ← b_i − a_ie b'_e
        for vsak j ∈ N−{e} do
            a'_ij ← a_ij − a_ie a'_ej
        a'_il ← −a_ie a'_el
    v' ← v + c_e b'_e
    for vsak j ∈ N−{e} do
        c'_j ← c_j − c_e a'_ej
    c'_l ← −c_e a'_el
    N' = N−{e} ∪ {l}
    B' = B−{l} ∪ {e}
    return (N',B',A',b',c',v')
```

Implementacija simpleks

```
PIVOT(N,B,A,b,c,v,l,e)
    b'_e ← b_l / a_le
    for vsak j ∈ N−{e} do
        a'_ej ← a_lj / a_le
    a'_el ← 1 / a_le
    for vsak i ∈ B−{l} do
        b'_i ← b_i − a_ie b'_e
        for vsak j ∈ N−{e} do
            a'_ij ← a_ij − a_ie a'_ej
        a'_il ← −a_ie a'_el
    v' ← v + c_e b'_e
    for vsak j ∈ N−{e} do
        c'_j ← c_j − c_e a'_ej
    c'_l ← −c_e a'_el
    N' = N−{e} ∪ {l}
    B' = B−{l} ∪ {e}
    return (N',B',A',b',c',v')
```

izrazitev vstopne spremenljivke iz enačbe za izstopno spremenljivko

Implementacija simpleks

```
PIVOT(N,B,A,b,c,v,l,e)
    b'_e ← b_l / a_le
    for vsak j ∈ N−{e} do
        a'_ej ← a_lj / a_le
    a'_el ← 1 / a_le
    for vsak i ∈ B−{l} do
        b'_i ← b_i − a_ie b'_e
        for vsak j ∈ N−{e} do
            a'_ij ← a_ij − a_ie a'_ej
        a'_il ← −a_ie a'_el
    v' ← v + c_e b'_e
    for vsak j ∈ N−{e} do
        c'_j ← c_j − c_e a'_ej
    c'_l ← −c_e a'_el
    N' = N−{e} ∪ {l}
    B' = B−{l} ∪ {e}
    return (N',B',A',b',c',v')
```

vstavitev enačbe za vstopno spremenljivko v enačbe ostalih osnovnih spremenljivk

Implementacija simpleks

```
PIVOT(N,B,A,b,c,v,l,e)
    b'_e ← b_l / a_le
    for vsak j ∈ N−{e} do
        a'_ej ← a_lj / a_le
    a'_el ← 1 / a_le
    for vsak i ∈ B−{l} do
        b'_i ← b_i − a_ie b'_e
        for vsak j ∈ N−{e} do
            a'_ij ← a_ij − a_ie a'_ej
        a'_il ← −a_ie a'_el
    v' ← v + c_e b'_e
    for vsak j ∈ N−{e} do
        c'_j ← c_j − c_e a'_ej
    c'_l ← −c_e a'_el
    N' = N−{e} ∪ {l}
    B' = B−{l} ∪ {e}
    return (N',B',A',b',c',v')
```

vstavitev enačbe za vstopno spremenljivko v enačbo kriterijske funkcije

Implementacija simpleks

```
PIVOT(N,B,A,b,c,v,l,e)
    b'_e ← b_l / a_le
    for vsak j ∈ N−{e} do
        a'_ej ← a_lj / a_le
    a'_el ← 1 / a_le
    for vsak i ∈ B−{l} do
        b'_i ← b_i − a_ie b'_e
        for vsak j ∈ N−{e} do
            a'_ij ← a_ij − a_ie a'_ej
        a'_il ← −a_ie a'_el
    v' ← v + c_e b'_e
    for vsak j ∈ N−{e} do
        c'_j ← c_j − c_e a'_ej
    c'_l ← −c_e a'_el
    N' = N−{e} ∪ {l}
    B' = B−{l} ∪ {e}
    return (N',B',A',b',c',v')
```

posodobitev indeksov osnovnih in neosnovnih spremenljivk

Metoda simpleks – zgled 3

- Kmet ima na voljo 18 hektarjev zemlje, na kateri želi posejati pšenico in ječmen. Na voljo ima 200 kg gnojiva in 120 litrov insekticida. Za vsak hektar posejane pšenice porabi 15 kg gnojiva in 8 litrov insekticida, za vsak hektar posejanega ječmena pa 10 kg gnojiva in 6 litrov insekticida. Pričakovani dobiček od hektarja pšenice je 120 evrov, od hektarja ječmena pa 90 evrov. Koliko pšenice in koliko ječmena naj kmet poseje, da bo maksimiziral pričakovani dobiček?

Metoda simpleks – zgled 3

- Kmet ima na voljo 18 hektarjev zemlje, na kateri želi posejati pšenico in ječmen. Na voljo ima 200 kg gnojiva in 120 litrov insekticida. Za vsak hektar posejane pšenice porabi 15 kg gnojiva in 8 litrov insekticida, za vsak hektar posejanega ječmena pa 10 kg gnojiva in 6 litrov insekticida. Pričakovani dobiček od hektarja pšenice je 120 evrov, od hektarja ječmena pa 90 evrov. Koliko pšenice in koliko ječmena naj kmet poseje, da bo maksimiziral pričakovani dobiček?

Maksimiziraj

$$120x_1 + 90x_2$$

ob pogoju

$$x_1 + x_2 \leq 18$$

$$15x_1 + 10x_2 \leq 200$$

$$8x_1 + 6x_2 \leq 120$$

$$x_1, x_2 \geq 0$$

Metoda simpleks – zgled 3

- Kmet ima na voljo 18 hektarjev zemlje, na kateri želi posejati pšenico in ječmen. Na voljo ima 200 kg gnojiva in 120 litrov insekticida. Za vsak hektar posejane pšenice porabi 15 kg gnojiva in 8 litrov insekticida, za vsak hektar posejanega ječmena pa 10 kg gnojiva in 6 litrov insekticida. Pričakovani dobiček od hektarja pšenice je 120 evrov, od hektarja ječmena pa 90 evrov. Koliko pšenice in koliko ječmena naj kmet poseje, da bo maksimiziral pričakovani dobiček?

Maksimiziraj

$$120x_1+90x_2$$

ob pogoju

$$x_1+x_2\leq 18$$
$$15x_1+10x_2\leq 200$$
$$8x_1+6x_2\leq 120$$
$$x_1, x_2\geq 0$$

$$z=120x_1+90x_2$$
$$x_3=18-x_1-x_2$$
$$x_4=200-15x_1-10x_2$$
$$x_5=120-8x_1-6x_2$$

Metoda simpleks – zgled 3

- prva vhodna spremenljivka naj bo x_1
- Δ_3=18, Δ_4=200/15=13.33, Δ_5=120/8=15
- izstopna spremenljivka postane x_4
- enačbo x_1=13.33 – 0.67x_2 – 0.067x_4 vstavimo v ostale*
- druga vhodna spremenljivka je x_2
- Δ_3=4.67/0.33=14, Δ_1=13.33/0.67=20, Δ_5=13.33/0.67=20
- izstopna spremenljivka postane x_3
- enačbo x_2=14 – 3x_3 + 0.2x_4 vstavimo v ostale*
- rešitev: kmet poseje 4 hektarje pšenice in 14 hektarjev ječmena

$$z = 120\,x_1 + 90\,x_2$$
$$x_3 = 18 - x_1 - x_2$$
$$x_4 = 200 - 15\,x_1 - 10\,x_2$$
$$x_5 = 120 - 8\,x_1 - 6\,x_2$$

Metoda simpleks – zgled 3

- prva vhodna spremenljivka naj bo x_1
- Δ_3=18, Δ_4=200/15=13.33, Δ_5=120/8=15
- izstopna spremenljivka postane x_4
- enačbo $x_1=13.33 - 0.67x_2 - 0.067x_4$ vstavimo v ostale*
- druga vhodna spremenljivka je x_2
- Δ_3=4.67/0.33=14, Δ_1=13.33/0.67=20, Δ_5=13.33/0.67=20
- izstopna spremenljivka postane x_3
- enačbo $x_2=14 - 3x_3 + 0.2x_4$ vstavimo v ostale*
- rešitev: kmet poseje 4 hektarje pšenice in 14 hektarjev ječmena

$$z=1600+10\,x_2-8\,x_4$$
$$x_3=4.67-0.33\,x_2+0.067\,x_4$$
$$x_1=13.33-0.67\,x_2-0.067\,x_4$$
$$x_5=13.33-0.67\,x_2+0.533\,x_4$$

Metoda simpleks – zgled 3

- prva vhodna spremenljivka naj bo x_1
- Δ_3=18, Δ_4=200/15=13.33, Δ_5=120/8=15
- izstopna spremenljivka postane x_4
- enačbo $x_1=13.33 - 0.67x_2 - 0.067x_4$ vstavimo v ostale*
- druga vhodna spremenljivka je x_2
- Δ_3=4.67/0.33=14, Δ_1=13.33/0.67=20, Δ_5=13.33/0.67=20
- izstopna spremenljivka postane x_3
- enačbo $x_2=14 - 3x_3 + 0.2x_4$ vstavimo v ostale*
- rešitev: kmet poseje 4 hektarje pšenice in 14 hektarjev ječmena

$$z=1740-30x_3-6x_4$$
$$x_2=14-3x_3+0.2x_4$$
$$x_1=4+2x_3-0.2x_4$$
$$x_5=4+2x_3+0.4x_4$$