Izbrana poglavja iz optimizacije – celoštevilsko linearno programiranje

Janez Povh

Univerza v Ljubljani, Fakulteta za matematiko in fiziko

Ljubljana, januar 2023

Primer: celoštevilski transportni problem

Podjetje ima v dveh različnih krajih obrata za proizvodnjo izdelka I, ki ga prodaja štirim trgovskim verigam. Podatki o mesečnih kapacitetah obratov in mesečnih potrebah trgovskih verig so podani v tonah, transportni stroški pa v Eur/t. Če podjetje ne zadovolji povpraševanja, nastane zanj škoda, ki znaša 35 Eur za tono nezadovoljene potrebe T_1, 30 Eur za tono nezadovoljene potrebe T_2, 45 Eur za tono nezadovoljene potrebe T_3 in 40 Eur za tono nezadovoljene potrebe T_4. Podjetje razmišlja o izgradnji obrata O_3, kar bi povzročilo povečanje mesečnih stalnih stroškov za 2500 Eur.

	T_1	T_2	T_3	T_4	kapacitete
O_1	6	8	4	8	1000
O_2	24	3	12	5	450
O_3	7	10	12	6	630
Potrebe	600	550	350	150	

(a) Sestavite matematični model za minimiziranje vsote mesečnih transportnih stroškov, škode zaradi nezadovoljenega povpraševanja in možnega povečanja mesečnih stalnih stroškov, ki bi jih povzročila možna odločitev o izgradnji obrata O_3.

(b) Rešite problem iz točke a) z uporabo Pythona.

(c) Poiščite in zapišite optimalni način transporta za primer, če bi podjetje investiralo v izgradnjo obrata O_3. Koliko znašajo minimalni transportni stroški?

Primer-mat. model

$$\min \quad 2500y + 6x_{11} + 8x_{12} + 4x_{13} + \cdots + 12x_{33} + 6x_{34} + 35(600 - x_{11} - x_{21} - x_{31}) + 30(550 - x_{12} - x_{22} - x_{32}) + 45(350 - x_{13} - x_{23} - x_{33}) + 40(150 - x_{14} - x_{24} - x_{34})$$

pri pogojih:

$$\begin{aligned}
x_{11} + x_{12} + x_{13} + x_{14} &\leq 1000 \\
x_{21} + x_{22} + x_{23} + x_{24} &\leq 450 \\
x_{31} + x_{32} + x_{33} + x_{34} &\leq 630y \\
x_{11} + x_{21} + x_{31} &\leq 600 \\
x_{12} + x_{22} + x_{32} &\leq 550 \\
x_{13} + x_{23} + x_{33} &\leq 350 \\
x_{14} + x_{24} + x_{34} &\leq 150 \\
y \in \{0, 1\}, x_{ij} &>= 0;
\end{aligned}$$

Metoda reševanje 1: rešimo ločeno za $y = 1$ in $y = 0$.

Reševanje s Pythonom - način 1

Način 1: rešimo ločeno za $y = 1$.

```
Aub=[[1, 1, 1,1, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,]]
Aeq=[[1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0],
[0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0],
[0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0],
[0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1]]
bub=[1000, 450, 630]
beq=[600, 550, 350,150]
f1=[6, 8, 4, 8, 24, 3, 12, 5, 7, 10, 12, 6]
f2=[35,30,45,40,35,30,45,40,35,30,45,40]
ff=np.add(f1,np.dot(-1,f2))
opt_TP3a=linprog(c=ff,A_ub=Aub,b_ub=bub,A_eq=Aeq,b_eq=beq,method="revised simplex")
min_str1=opt_TP3a.fun+35*600+30*550+45*350+40*150+2500
```

Optimum:

	T_1	T_2	T_3	T_4	odpeljano
O_1	550	100	350	0	1000
O_2	0	450	0	0	450
O_3	50	0	0	150	200
Dostavljeno	600	550	350	150	

Minimalni stroški: 10.660

Reševanje s Pythonom - način 2

Način 1: rešimo ločeno za $y = 0$.

```
Aub1=[[1, 1, 1,1, 0, 0, 0, 0 ],
[0, 0, 0, 0, 1, 1, 1, 1],
[1, 0, 0, 0, 1, 0, 0, 0],
[0, 1, 0, 0, 0, 1, 0, 0],
[0, 0, 1, 0, 0, 0, 1, 0],
[0, 0, 0, 1, 0, 0, 0, 1]]
bub1=[1000, 450,  600, 550, 350,150]
f1a=[6, 8, 4, 8, 24, 3, 12, 5 ]
f2a=[35,30,45,40,35,30,45,40]
ff2a=np.add(f1a,np.dot(-1,f2a))
opt_TP3b=linprog(c=ff2a,A_ub=Aub1,b_ub=bub1,method="revised simplex")
min_str2=opt_TP3b.fun+35*600+30*550+45*350+40*150
```

Optimum:

	T_1	T_2	T_3	T_4	odpeljano
O_1	600	0	350	50	1000
O_2	0	350	0	100	450
Dostavljeno	600	350	350	150	

Minimalni stroški: 12.950

Reševanje s Pythonom kot MCLP

Uporabimo knjižnico za modeliranje MCLP: **import pulp as pl**

```
import pulp as pl
# definiramo reševalnik
solver = pl.getSolver('MOSEK')  #default: CBC

# definiramo celoštev. spremenljivke, >=0
x11 = pl.LpVariable("x11", 0, None)
x12 = pl.LpVariable("x12", 0, None)
x13 = pl.LpVariable("x13", 0, None)
x14 = pl.LpVariable("x14", 0, None)
x21 = pl.LpVariable("x21", 0, None)
x22 = pl.LpVariable("x22", 0, None)
x23 = pl.LpVariable("x23", 0, None)
x24 = pl.LpVariable("x24", 0, None)
x31 = pl.LpVariable("x31", 0, None)
x32 = pl.LpVariable("x32", 0, None)
x33 = pl.LpVariable("x33", 0, None)
x34 = pl.LpVariable("x34", 0, None)

y = pl.LpVariable("y", 0, 1, pl.LpInteger)

# definiramo MIP TP  problem
prob = pl.LpProblem("MIP_TP", pl.LpMinimize)

# kriterijska funkcija
prob += 2500*y + 6*x11 + 8*x12 + 4*x13 + 8*x14 + 24*x21 + 3*x22 + 12*x23 + 5*x24 + 7*x31 + 10*x32
+ 12*x33+6*x34 +35*(600- x11 -x21-x31)+ 30*(550- x12 -x22-x32)+ 45*(350- x13 -x23-x33)
+40*(150- x14 -x24-x34)

# omejitve
prob += x11 +x12+x13+x14 <= 1000
prob += x21 +x22+x23+x24 <= 450
```

Reševanje s Pythonom kot MCLP

```
print(pl.LpStatus[status])  # izpišemo status
# izpišemo vrednosti
print("x11", pl.value(x11))
print("x12", pl.value(x12))
print("x13", pl.value(x13))
print("x14", pl.value(x14))
print("x21", pl.value(x21))
print("x22", pl.value(x22))
print("x23", pl.value(x23))
print("x24", pl.value(x24))
print("x31", pl.value(x31))
print("x32", pl.value(x32))
print("x33", pl.value(x33))
print("x34", pl.value(x34))
print("y", pl.value(y))
print("objective=", pl.value(prob.objective))
```

```
Optimal
x11 550.0
x12 100.0
x13 350.0
x14 0.0
x21 0.0
x22 450.0
x23 0.0
x24 0.0
x31 50.0
x32 0.0
x33 0.0
x34 150.0
y 1.0
objective= 10600.0
```

Celoštevilsko linearno programiranje

Definicija

Mešani celoštevilski linearni program (MCLP) v osnovni obliki (izraz "primaren" tukaj izpustimo, saj nas redko zanima dual) definiramo kot

$$
\begin{array}{ll}
\max & c^T x \\
 & \text{pri pogojih} \\
 & \begin{array}{rcl} A\mathbf{x} & \leq & \mathbf{b}, \\ x_i & \in & \mathbb{Z}, \quad \text{za vsak } i \in \mathcal{I} \subseteq \{1, 2, \ldots, n\}. \end{array}
\end{array}
\qquad \text{(MCLP)}
$$

Celoštevilsko linearno programiranje

Definicija

Mešani celoštevilski linearni program (MCLP) v osnovni obliki (izraz "primaren" tukaj izpustimo, saj nas redko zanima dual) definiramo kot

$$\begin{array}{lllll} \max & c^T x & & & \\ & \text{pri pogojih} & & & \\ & \quad A\mathbf{x} & \leq & \mathbf{b}, & \\ & \quad x_i & \in & \mathbb{Z}, & \text{za vsak } i \in \mathcal{I} \subseteq \{1, 2, \ldots, n\}. \end{array} \tag{MCLP}$$

Opomba:

- Pogoj nenegativnosti $x \geq 0$ navadno vključimo v linearne neenačbe $Ax \leq b$.
- Množica $\mathcal{I}$ je množica indeksov spremenljivk, ki morajo biti celoštevilske. Če je $\mathcal{I} = \{1, \ldots, n\}$, potem imamo problem celoštevilskega programiranja.
- Predpostavka: vsi podatki so celoštevilski, torej $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$. Če ni izpolnjena, jo lahko izpolnimo z množenjem obeh strani neenačb z najmanjšimi skupnimi imenovalci, saj imamo v praksi za podatke vedno racionalna števila.

Primer

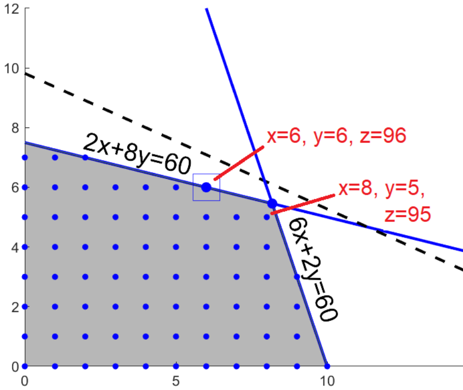
$$\max z = 5x + 11y$$

$$\begin{aligned} \text{pri pogojih } x &\geq 0 \\ y &\geq 0 \\ 2x + 8y &\leq 60 \\ 6x + 2y &\leq 60 \\ x,\ y &\in \mathbb{Z} \end{aligned}$$

Primer

$$\max z = 5x + 11y$$

$$\begin{array}{rcl} \text{pri pogojih } x & \geq & 0 \\ y & \geq & 0 \\ 2x + 8y & \leq & 60 \\ 6x + 2y & \leq & 60 \\ x,\ y & \in & \mathbb{Z} \end{array}$$



Slika: Množica dopustnih točk

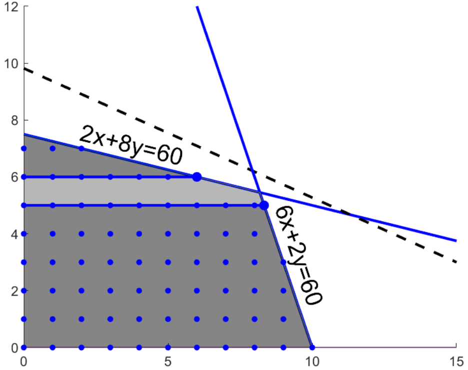
Nauk

- Zaokrožanje k najbližnji celoštevilski rešitvi ne prinese nujno optimalne rešitve.
- Optimalna rešitev je lahko precej daleč od rešitve LP ali najbližje celoštevilske rešitve.
- Če se hočemo izogniti preverjanu vseh možnih rešitev (vsake celoštevilske točke) in uporabiti metode LP, moramo najti način, da bo v okolici optimalne rešitve dopustno območje opisano z linearnimi neenačbami.

Reševanje s Pythonom

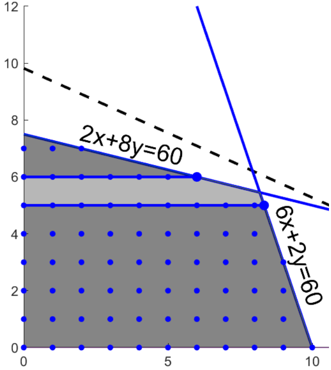
```
#%% MCLP 1
import pulp as pl
# definiramo reševalnik
solver = pl.getSolver('MOSEK')  #default: CBC

# definiramo celoštev. spremenljivke, >=0
x = pl.LpVariable("x", 0, None, pl.LpInteger)
y = pl.LpVariable("y", 0, None, pl.LpInteger)

# definiramo MCLP problem
prob = pl.LpProblem("MIP1", pl.LpMaximize)

# kriterijska funkcija
prob += 5 * x + 11 * y

# omejitve
prob += 2 * x + 8 * y  <= 60
prob += 6 * x + 2 * y  <= 60

status = prob.solve()  # Rešimo MCLP
print(pl.LpStatus[status])  # izpišemo status

# izpišemo vrednosti
print("x", pl.value(x))
print("y", pl.value(y))
print("objective=", pl.value(prob.objective))
```

Metoda razveji in omeji

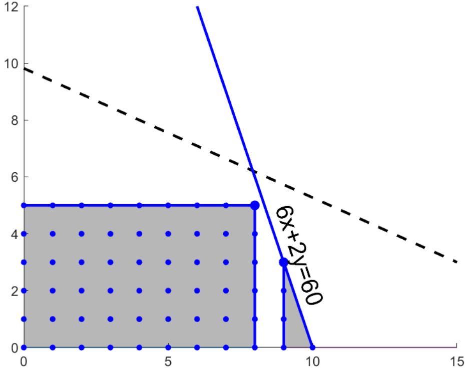


Slika: Množica dopustnih točk, če razvejimo v smeri y

Metoda razveji in omeji



Slika: Množica dopustnih točk, če razvejimo v smeri y



Slika: Množica dopustnih točk, ko razvejimo še v smeri x

Metoda Razveji in omeji (Branch and Bound)

VHOD: podatki za (MCLP): $c, A, b, \mathcal{I}$
INICIALIZACIJA:

1. Naj bo $x^{(0)}$ rešitev linearnega programa, ki ga dobimo iz (MCLP) z izpustitvijo pogojev za celoštevilskost:
$$\max c^T x \text{ pri pogojih } Ax \leq b.$$
2. **ČE** $x_i^{(0)} \in \mathbb{Z}$ za vsak $i \in \mathcal{I}$, **POTEM** je $x^{(0)}$ opt. rešitev in optimum je $OPT = c^T x^{(0)}$. **KONEC**
3. Poišči dopustno rešitev x^S za (MCLP) in izračunaj spodnjo mejo za optimalno vrednost (MCLP) $S = c^T x^S$.
4. Kreiraj razveji in omeji drevo z začetno točko, ki vsebuje trenutno rešitev linearnega programa $x^{(0)}$ in trenutne linearne omejitve $Ax \leq b$.

Algoritem Razveji in omeji (Branch and Bound) - nadaljevanje

DOKLER razveji in omeji drevo ni v celoti pregledano

5. Najdi in označi kot pregledano naslednjo točko iz drevesa s podatki $x^{(1)}$, $\tilde{A}x \leq \tilde{b}$.
6. **ČE** $c^T x^{(1)} \leq S$, **POTEM** nadaljuj s korakom 5.
7. Izberi $i \in \mathcal{I}$, da $x_i^{(1)} \notin \mathbb{Z}$.
8. Izračunaj $x^{(2)}$ kot rešitev linearnega programa
$$\max c^T x \ \text{ pri pogojih } \ \tilde{A}x \leq \tilde{b},\ x_i \leq \lfloor x_i^{(1)} \rfloor.$$
9. **ČE** $c^T x^{(2)} \geq S$ in $x_j^{(2)} \in \mathbb{Z}$ za vsak $j \in \mathcal{I}$, **POTEM** $S = c^T x^{(2)}$, $x^S = x^{(2)}$.
10. **ČE** $c^T x^{(2)} \geq S$ in obstaja $j \in \mathcal{I}$ da $x_j^{(2)} \notin \mathbb{Z}$, **POTEM** dodaj v drevo točko s podatki $x^{(2)}, \tilde{A}x \leq \tilde{b}, x_i \leq \lfloor x_i^{(1)} \rfloor$.
11. Izračunaj $x^{(3)}$ kot rešitev linearnega programa
$$\max c^T x \ \text{ pri pogojih } \ \tilde{A}x \leq \tilde{b},\ x_i \geq \lceil x_i^{(1)} \rceil + 1.$$
12. **ČE** $c^T x^{(3)} \geq S$ in $x_j^{(3)} \in \mathbb{Z}$ za vsak $j \in \mathcal{I}$, **POTEM** $S = c^T x^{(3)}$, $x^S = x^{(3)}$.
13. **ČE** $c^T x^{(3)} \geq S$ in obstaja $j \in \mathcal{I}$ da $x_j^{(3)} \notin \mathbb{Z}$, **POTEM** dodaj v drevo točko s podatki $x^{(3)}, \tilde{A}x \leq \tilde{b}, x_i \geq \lceil x_i^{(1)} \rceil + 1$.

IZHOD: $x_{opt} = x^S,\ OPT = S.$

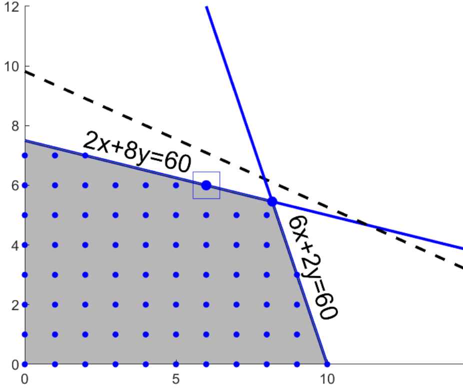
Primer uporabe algoritma Razveji in omeji

$$
\begin{array}{rcl}
\multicolumn{3}{c}{\max z = 5x + 11y} \\
\hline
\text{pri pogojih } x & \geq & 0 \\
y & \geq & 0 \\
2x + 8y & \leq & 60 \\
6x + 2y & \leq & 60 \\
x,\ y & \in & \mathbb{Z}
\end{array}
$$

Primer uporabe algoritma Razveji in omeji

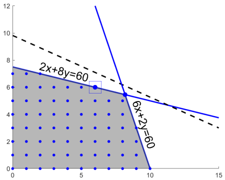
$$\max z = 5x + 11y$$

$$\begin{array}{rcl}
\text{pri pogojih } x & \geq & 0 \\
y & \geq & 0 \\
2x + 8y & \leq & 60 \\
6x + 2y & \leq & 60 \\
x,\ y & \in & \mathbb{Z}
\end{array}$$



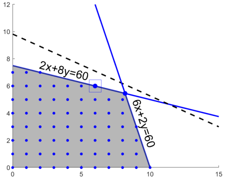
Slika: Množica dopustnih točk

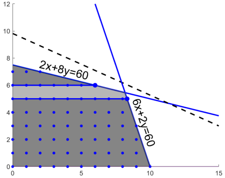
Primer



- Opt. rešitev: $x = 8.18$, $y = 5.45$, optimum: $z = 100.91$. To je zgornja meja za optimalno vrednost celoštevilskega problema. Ker optimalna rešitev ni celoštevilska, v koraku 2 ne končamo.
- Korak 3: z zaokrožanjem navzdol najdemo dopustno rešitev $x = 8$, $y = 5$, kar nam da spodnjo mejo $S = 5 \cdot 8 + 11 \cdot 5 = 95$ za optimum.
- Korak 4: S temi podatki kreiramo korensko vozlišče razveji in omeji drevesa.

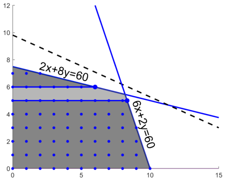
Primer

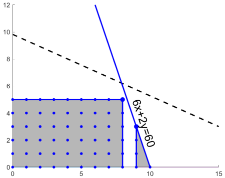




- Korak 5: iz drevesa vzamemo korensko vozlišče, torej imamo: $x^{(1)} = (8.18,\ 5.45)$.
- Korak 6: ker $c^T x^{(1)} = 100.91$ ni manjša od spodnje meje $S = 95$, nadaljujemo s korakom 7.
- Korak 7: za spremenljivko, na kateri bomo razvejili drevo, vzamemo y, saj je njen decimalni del bliže 0.5.
- V koraku 8 rešimo osnovni linearni program z dodano neenačbo $y \leq 5$, v koraku 11 pa osnovni linearni program z dodano neenačbo $y \geq 6$.
- V koraku 8 dobimo rešitev $x^{(2)} = (8.33,\ 5.00)$ in optimalno vrednost $c^T x^{(2)} = 96.67$. V koraku 10 to rešitev in pripadajoči linearni program vstavimo v drevo.
- V koraku 11 izračunamo rešitev osnovnega linearnega programa z dodatno omejitvijo $y \geq 6$. Optimalna rešitev je $x^{(3)} = (6.00,\ 6.00)$ z optimalno vrednostjo $z = 96.00$. Ker je ta rešitev celoštevilska, posodobimo $x_S = (6.00, 6.00)$ in $S = 96.00$.

Primer



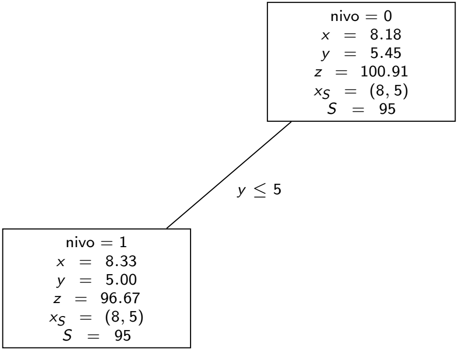


- V drugi ponovitvi korakov 5–13 vzamemo v obravnavo levo točko $x = (8.33,\ 5)$. Optimalna vrednost $z = 96.67$ je večja od trenutne spodnje meje $S = 96$ zato nadaljujemo s koraki 7–13.
- Drevo razvejimo na $x = 8.33$. Najprej rešimo linearni program, kjer osnovnim omejitvam in že dodani omejitvi $y \leq 5$ dodamo še $x \leq 8$. Rešitev $x = 8,\ y = 5$ je celoštevilska, katere vrednost je nižja $S = 96$, zato je ne dodamo v drevo.
- V koraku 11 rešimo osnovni linearni program, z dodanimi omejitvami $y \leq 5$ in $x \geq 9$.
- Optimalna rešitev $x = 9,\ y = 3$ je celoštevilska, a opt. vrednost 78 je od $S = 96$, zato v koraku 12 optimalne rešitve ne posodobimo.
- Ker ni več nobene točke za obravnavo, se algoritem konča in zadnja rešitev $x_S = (6,\ 6)$ je tudi optimalna rešitev, ki da optimalno vrednost 96.

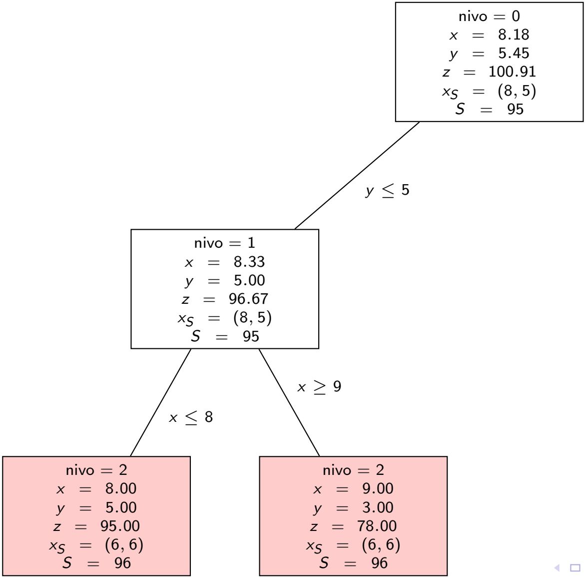
Drevo

$$\begin{array}{c} \text{nivo} = 0 \\ x \;=\; 8.18 \\ y \;=\; 5.45 \\ z \;=\; 100.91 \\ x_S \;=\; (8, 5) \\ S \;=\; 95 \end{array}$$

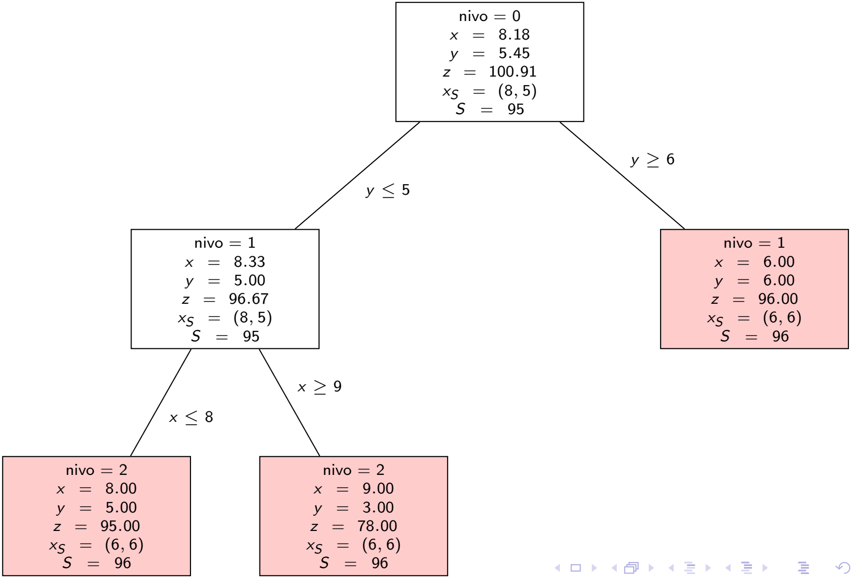
Drevo


nivo = 0
$x = 8.18$
$y = 5.45$
$z = 100.91$
$x_S = (8, 5)$
$S = 95$
$y \leq 5$
nivo = 1
$x = 8.33$
$y = 5.00$
$z = 96.67$
$x_S = (8, 5)$
$S = 95$

Drevo


nivo = 0
$x = 8.18$
$y = 5.45$
$z = 100.91$
$x_S = (8, 5)$
$S = 95$
$y \leq 5$
nivo = 1
$x = 8.33$
$y = 5.00$
$z = 96.67$
$x_S = (8, 5)$
$S = 95$
$x \leq 8$
$x \geq 9$
nivo = 2
$x = 8.00$
$y = 5.00$
$z = 95.00$
$x_S = (6, 6)$
$S = 96$
nivo = 2
$x = 9.00$
$y = 3.00$
$z = 78.00$
$x_S = (6, 6)$
$S = 96$

Drevo



nivo = 0
$x = 8.18$
$y = 5.45$
$z = 100.91$
$x_S = (8, 5)$
$S = 95$
$y \leq 5$
$y \geq 6$
nivo = 1
$x = 8.33$
$y = 5.00$
$z = 96.67$
$x_S = (8, 5)$
$S = 95$
nivo = 1
$x = 6.00$
$y = 6.00$
$z = 96.00$
$x_S = (6, 6)$
$S = 96$
$x \leq 8$
$x \geq 9$
nivo = 2
$x = 8.00$
$y = 5.00$
$z = 95.00$
$x_S = (6, 6)$
$S = 96$
nivo = 2
$x = 9.00$
$y = 3.00$
$z = 78.00$
$x_S = (6, 6)$
$S = 96$