DEVOIR 2

ROMAN NAKHUDA 20207082
LUCKY KHOUNVONGSA 20172476
ANGE LILIAN TCHOMTCHOUA TOKAM 20210129

OPTIMISATION LINEAIRE - IFT2505
PROFESSEUR FABIEN BASTIN

Université de Montréal
À remettre le 16 Octobre 2023 à 23:59

Question 1

Soit le problème:

$$\begin{array}{lrcrcrcl} \min\ x = & -2x & - & y & - & 3z & & \\ \text{tel que} & 2x & + & 3y & + & 4z & \leq & 120 \\ & x & + & 2y & & & \leq & 50 \\ & x & & + & & 2z & \leq & 50 \\ & x, & & y, & & z & \geq & 0 \end{array}$$

- **(a) Resolvons le problème à la main avec la méthode du simplexe.**

On transforme le problème sous la forme standard à l'introduction des variables d'écarts $s_1,\ s_2$ et s_2.

On obtient:

$$
\begin{array}{llcccccccccccc}
\min\ x = & -2x & - & y & - & 3z & & & & & & & & \\
\text{tel que} & 2x & + & 3y & + & 4z & + & s_1 & & & & & = & 120 \\
 & x & + & 2y & & & & & + & s_2 & & & = & 50 \\
 & x & & & + & 2z & & & & & + & s_3 & = & 50 \\
 & x, & & y, & & z, & & s_1, & & s_2, & & s_3 & \geq & 0
\end{array}
$$

Avec les variables dans la base s_1, s_2 et s_2, On obtient le tableau du simplexe suivant;

	x	y	z	s_1	s_2	s_3	T.D.
s_1	2	3	4	1	0	0	120
s_2	1	2	0	0	1	0	50
s_3	1	0	2	0	0	1	50
$-z$	-2	-1	-3	0	0	0	0

On prend z comme variable d'entrée et s_3 comme variable de sortie.

	x	y	z	s_1	s_2	s_3	T.D.
s_1	0	3	0	1	0	0	20
s_2	1	2	0	0	1	0	50
z	$1/2$	0	1	0	0	$1/2$	25
$-z$	$-1/2$	-1	0	0	0	$3/2$	75

On prend y comme variable d'entrée et s_1 comme variable de sortie.

	x	y	z	s_1	s_2	s_3	T.D.
y	0	1	0	$1/3$	0	0	$20/3$
s_2	1	0	0	$-2/3$	1	0	$110/3$
z	$1/2$	0	1	0	0	$1/2$	25
$-z$	$-1/2$	0	0	$1/3$	0	$3/2$	$245/3$

On prend x comme variable d'entrée et s_2 comme variable de sortie.

	x	y	z	s_1	s_2	s_3	T.D.
y	0	1	0	$1/3$	0	0	$20/3$
x	1	0	0	$-2/3$	1	0	$110/3$
z	0	0	1	0	$1/3$	$1/2$	$20/3$
$-z$	0	0	0	0	$1/2$	$3/2$	100

Donc,

$$x = 110/3, \; y = 20/3, \; z = 20/3$$
$$s_1 = s_2 = s_3 = 0$$

et la valeur optimale du problème est $z^{\star} = -100$.

- **(b) Resolvons le problème avec la méthode du simplexe revisée.**

Considérons le problème sous forme canonique:

$$\begin{array}{lccccccccccccc}
\min \; x = & -2x & - & y & - & 3z & & & & & & & & \\
\text{tel que} & 2x & + & 3y & + & 4z & + & s_1 & & & & & = & 120 \\
 & x & + & 2y & & & & & + & s_2 & & & = & 50 \\
 & x & & & + & 2z & & & & & + & s_3 & = & 50 \\
 & x, & & y, & & z, & & s_1, & & s_2, & & s_3 & \geq & 0
\end{array}$$

On se limite à:

Var				x_B
s_1	1	0	0	120
s_2	0	1	0	50
s_3	0	0	1	50

On a, $\lambda^T = c_B^T B^{-1} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} B^{-1} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$

et $r_D^T = C_D^T - \lambda^T D = \begin{pmatrix} -2 & -1 & -3 \end{pmatrix}$

On obtient le dictionnaire;

$$\begin{array}{ccccccc}
s_1 & = & 120 & -2x & -3y & -4z \\
s_2 & = & 50 & -x & -2y & \\
s_3 & = & 50 & -x & & -2z \\
z & = & 0 & -2x & -y & -3z
\end{array}$$

La base B est constituée des colonnes associées aux variables $s_1, \; s_2$ et s_3.

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Le variable entrante retenue est z, et on construit le tableau

Var				x_B	x
s_1	1	0	0	120	4
s_2	0	1	0	50	0
s_3	0	0	1	50	2

Après le pivot,

Var				x_B
s_1	1	0	-2	20
s_2	0	1	0	50
z	0	0	$1/2$	25

Nous avons également $C_B^T = \begin{pmatrix} 0 & 0 & -3 \end{pmatrix}$ et

$$\lambda^T = c_B^T B^{-1} = \begin{pmatrix} 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} 1 & 0 & -2 \\ 0 & 1 & 0 \\ 0 & 0 & 1/2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -3/2 \end{pmatrix}$$

Les coûts reduits se calculent de manière similaire $$

$$\begin{aligned} r_D^T &= C_D^T - \lambda^T D \\ &= \begin{pmatrix} -2 & -1 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 & -3/2 \end{pmatrix} \begin{pmatrix} 2 & 3 & 0 \\ 1 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} -1/2 & -1 & 3/2 \end{pmatrix} \\ &= \begin{pmatrix} r_1 & r_2 & r_3 \end{pmatrix} \end{aligned}$$

Par la suite, On prend y comme variable entrante, on obtient;

$$y = \begin{pmatrix} 1 & 0 & -2 \\ 0 & 1 & 0 \\ 0 & 0 & 1/2 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 0 \end{pmatrix}$$

\

Var				x_B	y
s_1	1	0	-2	20	3
s_2	0	1	0	50	2
z	0	0	$1/2$	25	0

Après le pivot:

Var				x_B
y	$1/3$	0	$-2/3$	$20/3$
s_2	$-2/3$	1	$4/3$	$110/3$
z	0	0	$1/2$	25

Nous avons également $C_B^T = \begin{pmatrix} -1 & 0 & -3 \end{pmatrix}$ et

$$\lambda^T = c_B^T B^{-1} = \begin{pmatrix} -1 & 0 & -3 \end{pmatrix} \begin{pmatrix} 1/3 & 0 & -2/3 \\ -2/3 & 1 & 4/3 \\ 0 & 0 & 1/2 \end{pmatrix} = \begin{pmatrix} -1/3 & 0 & -5/6 \end{pmatrix}$$

$$\begin{aligned} r_D^T &= \begin{pmatrix} -2 & 0 & 0 \end{pmatrix} - \begin{pmatrix} -1/3 & 0 & -5/6 \end{pmatrix} \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} -1/2 & 1/3 & 5/6 \end{pmatrix} \\ &= \begin{pmatrix} r_1 & r_2 & r_3 \end{pmatrix} \end{aligned}$$

Le variable entrante retenue est x, et on construit le tableau

$$x = \begin{pmatrix} 1/3 & 0 & -2/3 \\ -2/3 & 1 & 4/3 \\ 0 & 0 & 1/2 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1/2 \end{pmatrix}$$

\

Var				x_B	x
y	$1/3$	0	$-2/3$	$20/3$	0
s_2	$-2/3$	1	$4/3$	$110/3$	1
z	0	0	$1/2$	25	$1/2$

Après le pivot et la variable sortante s_2,

Var				x_B
y	$1/3$	0	$-2/3$	$20/3$
x	$-2/3$	1	$4/3$	$110/3$
z	$1/3$	$-1/2$	$-1/6$	$20/3$

On a, $\lambda^T = \begin{pmatrix} -1 & -2 & -3 \end{pmatrix} \begin{pmatrix} 1/3 & 0 & -2/3 \\ -2/3 & 1 & 4/3 \\ 1/3 & -1/2 & 1/6 \end{pmatrix} = \begin{pmatrix} 0 & -1/2 & -3/2 \end{pmatrix}$

$$r_D^T = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & -1/2 & -3/2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & 1/2 & 3/2 \end{pmatrix}$$
$$\implies r_1 = 0, r_2 = 1/2, r_3 = 3/2$$

D'où la solution optimale $x = \begin{pmatrix} 110/3, & 20/3, & 20/3, & 0, & 0, & 0 \end{pmatrix}$

- **(c) Utilisons JuMP en Julia pour vérifier la solution du problème.**

In [1]:
```julia
using JuMP, HiGHS
```

In [2]:
```julia
m = Model(HiGHS.Optimizer)

@variable(m, 0 <= x)
@variable(m, 0 <= y)
@variable(m, 0 <= z)

@constraint(m, 2x + 3y + 4z <= 120.0)
@constraint(m, x + 2y <= 50.0)
@constraint(m, x + 2z <= 50.0)

@objective(m, Min, -2x-y-3z)

print(m)
```

```
Min -2 x - y - 3 z
Subject to
 2 x + 3 y + 4 z <= 120
 x + 2 y <= 50
 x + 2 z <= 50
 x >= 0
 y >= 0
 z >= 0
```

In [3]:
```julia
status = optimize!(m)
```

```
Running HiGHS 1.5.3 [date: 1970-01-01, git hash: 45a127b78]
Copyright (c) 2023 HiGHS under MIT licence terms
Presolving model
3 rows, 3 cols, 7 nonzeros
3 rows, 3 cols, 7 nonzeros
Presolve : Reductions: rows 3(-0); columns 3(-0); elements 7(-0) - Not reduced
Problem not reduced by presolve: solving the LP
Using EKK dual simplex solver - serial
  Iteration        Objective     Infeasibilities num(sum)
          0    -5.9999935790e+00 Ph1: 3(15); Du: 3(5.99999) 0s
          3    -1.0000000000e+02 Pr: 0(0) 0s
Model   status      : Optimal
Simplex   iterations: 3
Objective value     : -1.0000000000e+02
HiGHS run time      :          0.01
```

In [4]:
```
println([value(x), value(y), value(z)])
```

```
[50.0, 0.0, -0.0]
```

In [5]:
```
function pivot!(M::Matrix, i::Int, j::Int)
    m, n = size(M)
    @assert M[i, j] != 0
    M[i, :] = M[i, :]/M[i, j]
    for k in setdiff(1:m, i)
        M[k, :] -= M[k, j] * M[i, :]
    end
    return M
end

function getReducedCosts(M::Matrix)
    m, n = size(M)
    return M[end, 1:n-1]
end
function getxB(M::Matrix)
    m, n = size(M)
    return M[1:m-1, end]
end
function enteringVar(M::Matrix)
    rc = getReducedCosts(M)
    index = argmin(rc)
    return rc[index] >= 0 ? -1 : index
end
function exitingVarIndex(M::Matrix{T}, enteringVar::Int) where T
    col = M[1:end-1, enteringVar]
    xB = getxB(M)
    m, n = size(M)
    index = -1
    val = T(Inf)
    for i in 1:m-1
        if (col[i] > 0) && (xB[i]/col[i] < val)
            val = xB[i]/col[i]
            index = i
        end
    end
    return index
```

```julia
end
function isOneHot(v::Vector)
    n = length(v)
    return (sum(iszero, v) == n-1) && (sum(isone, v) == 1)
end
function isoptimal(M)
    return enteringVar(M) == -1
end
function findInitialBasis!(M::Matrix)
    m, n = size(M)
    m-=1
    n-=1
    basis = [-1 for _ in 1:m]
    for i in 1:n
        if isOneHot(M[1:end-1, i])
            index = findfirst(isone, M[:, i])
            basis[index] = i
        end
    end
    @assert !any(t-> t == -1, basis) "problem not caconical"
    for i in 1:m
        j = basis[i]
        pivot!(M, i, j)
    end
    return basis
end
function simplexSolver(A::Matrix{T}, b::Vector, c::Vector; verbose::Bool = false) w
    M = [A b; c' 0]
    basis = findInitialBasis!(M)
    k = 1
    nmax = 1000
    while !isoptimal(M) && k < nmax
        k+=1
        verbose && display(M)
        entering = enteringVar(M)
        entering == -1 && (println("------------"); return T[-1, -1])
        leaving = exitingVarIndex(M, entering)
        leaving == -1 && (println("------------"); return T[-1, -1])
        verbose && @show (entering, leaving)
        basis[leaving] = entering
        pivot!(M, leaving, entering)
    end
    verbose && display(M)
    m, n = size(M)
    xstar = zeros(T, n - 1)
    xstar[basis] = getxB(M)
    return xstar
end
```

```
simplexSolver (generic function with 1 method)
```

In [6]:

```julia
A=[2//1 3 4 1 0 0 ;
    1 2 0 0 1 0;
    1 0 2 0 0 1;
]
```

```
b=[120, 50, 50]
c = [-2,-1,-3,0,0,0]
```

```
6-element Vector{Int64}:
 -2
 -1
 -3
  0
  0
  0
```

In [7]:
```
simplexSolver(A, b, c, verbose=true)
```

```
4×7 Matrix{Rational{Int64}}:
  2//1   3//1   4//1  1//1  0//1  0//1  120//1
  1//1   2//1   0//1  0//1  1//1  0//1   50//1
  1//1   0//1   2//1  0//1  0//1  1//1   50//1
 -2//1  -1//1  -3//1  0//1  0//1  0//1    0//1
4×7 Matrix{Rational{Int64}}:
  0//1   3//1  0//1  1//1  0//1  -2//1  20//1
  1//1   2//1  0//1  0//1  1//1   0//1  50//1
  1//2   0//1  1//1  0//1  0//1   1//2  25//1
 -1//2  -1//1  0//1  0//1  0//1   3//2  75//1
4×7 Matrix{Rational{Int64}}:
  0//1  1//1  0//1   1//3  0//1  -2//3   20//3
  1//1  0//1  0//1  -2//3  1//1   4//3  110//3
  1//2  0//1  1//1   0//1  0//1   1//2   25//1
 -1//2  0//1  0//1   1//3  0//1   5//6  245//3
4×7 Matrix{Rational{Int64}}:
 0//1  1//1  0//1   1//3   0//1  -2//3   20//3
 1//1  0//1  0//1  -2//3   1//1   4//3  110//3
 0//1  0//1  1//1   1//3  -1//2  -1//6   20//3
 0//1  0//1  0//1   0//1   1//2   3//2  100//1
(entering, leaving) = (3, 3)
(entering, leaving) = (2, 1)
(entering, leaving) = (1, 2)
6-element Vector{Rational{Int64}}:
 110//3
  20//3
  20//3
   0//1
   0//1
   0//1
```

Question 2

On va résoudre la démonstration par contradiction

On suppose $\urcorner p$ ainsi: On suppose que toutes solutions de base réalisable dégénérée peuvent être optimales en satisfaisant $r_j \geq 0$ pour tout j.

On considère ainsi le problème linéaire suivant:

Considérons le programme linéaire suivant:

$$\begin{array}{lcccccc} \max & x_1 & + & x_2 & & \\ \text{tel que} & x_1 & + & x_2 & \leq & 0 \\ & & & x_2 & \leq & 0 \\ & x_1, & & x_2 & \geq & 0 \end{array}$$

$$\Longrightarrow \quad \begin{array}{lcccccccc} -\ \min & -x_1 & - & x_2 & & & & & \\ \text{tel que} & x_1 & + & x_2 & + & x_3 & & = & 0 \\ & & & x_2 & & + & x_4 & = & 0 \\ & x_1, & & x_2, & & x_3, & x_4 & \geq & 0 \end{array}$$

En utilisant la méthode du simplexe on a:

$$\begin{array}{cccccc} & x_1 & x_2 & x_3 & x_4 & b \\ x_3 & 1 & 1 & 1 & 0 & 0 \\ x_4 & 0 & 1 & 0 & 1 & 0 \\ c^T & -1 & -1 & 0 & 0 & 0 \end{array}$$

Il y a des coéficientes reduits négatifs, par la suite on a:

$$\begin{array}{cccccc} & x_1 & x_2 & x_3 & x_4 & b \\ & 1 & 1 & 1 & 0 & 0 \\ & 0 & 1 & 0 & 1 & 0 \\ c^T & 0 & 0 & 1 & 0 & 0 \end{array}$$

On obtient donc la solution optimale (0, 0, 0, 0) et $z = 0$ avec des coûts réduits négatifs,
ce qui mène à une contradiction à $\neg p$:

Par conséquent on a prouvé par contradiction qu'une solution de base peut être optimale en satisfaisant $r_j \geq 0$ }$

Question 3

- **(a) Identifions les variables de base Initiales et Determinons B^{-1} :**

x_1	x_2	x_3	$x4$	x_5	b
a	b	c	1	0	6
-1	2	e	0	1	1
d	-1	3	0	0	0

Les variables de base initiales sont x_4 et x_5 car elles leurs coeficients forment la matrice indentité.

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} B^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- **(b) Identifions les variables de base Initiales et Determinons B^{-1} sur la deuxième itération:**

x_1	x_2	x_3	$x4$	x_5	b
f	$2/3$	$2/3$	$1/3$	0	i
g	h	$-1/3$	$1/3$	1	j
0	$1/3$	k	l	m	n

Les variables de base initiales sont x_1 et x_5

On obtient B^{-1} directement depuis le nouveau tableau du simplexe:

$$B^{-1} = \begin{pmatrix} 1/3 & 0 \\ 1/3 & 1 \end{pmatrix}$$

- **(c) Trouvons les valeurs de a, b, c, d, e, f, g, h, i, j, k, l, m, n**

On refait les étapes du simplexe sur le tableau initiale, on a;

x_1	x_2	x_3	$x4$	x_5	b
a	b	c	1	0	6
-1	2	e	0	1	1
d	-1	3	0	0	0

x_1 entre dans la base donc on divise toute la première ligne par a et on rend x_1
variable isolée en additionant la ligne 1 à la ligne 2 et en soustrayant d fois la ligne 1 à la ligne 3, on obtie

x_1	x_2	x_3	$x4$	x_5	b
1	b/a	c/a	$1/a$	0	$6/a$
0	$2+(b/a)$	$e+(c/a)$	$0+(1/a)$	1	$1+(6/a)$
0	$-1-d(b/a)$	$3-d(c/a)$	$0-d(1/a)$	0	$0-d(6/a)$

En comaparant le tableau obtenu avec le premier et le deuxième tableau du simplexe donnés par l'énoncé, $$

$$
\begin{aligned}
& f=1, g=0, a=3, m=0, \\
b/a = 2/3 \quad & \Longrightarrow \; b=2, \\
c/a = 2/3 \quad & \Longrightarrow \; c=2, \\
i = 6/a \quad & \Longrightarrow \; i=2, \\
h = 2+b/a \quad & \Longrightarrow \; h=8/3, \\
e+(c/a) = -1/3 \quad & \Longrightarrow \; e=-1, \\
j = 1+(6/a) \quad & \Longrightarrow \; j=3, \\
-1-d(b/a) = 1/3 \quad & \Longrightarrow \; d=-2, \\
k = 3-d(c/a) \quad & \Longrightarrow \; k=13/3, \\
l = 0-d(1/a) \quad & \Longrightarrow \; l=2/3, \\
n = 0-d(6/a) \quad & \Longrightarrow \; n=4
\end{aligned}
$$

- **(d) Tableau de la deuxième itération** En remplaçant les valeurs obtenues en c) on obtient le tableau suivant;

	x_1	x_2	x_3	$x4$	x_5	b
x_1	1	$2/3$	$2/3$	$1/3$	0	2
x_5	0	$8/3$	$-1/3$	$1/3$	1	3
r^T	0	$1/3$	$13/3$	$2/3$	0	4

On a la solution réalisable non dégénérée qui est $(x_1, x_2, x_3, x_4, x_5) = (2, 0, 0, 0, 3)$ avec $z^* = -4$ la solution optimale car tous les coûts réduits sont non négatifs. $$

- **(e) Écrivons le problème initial** En remplaçant les valeurs obtenues en c) dans le tableau initial du simplexe, on obtient;

$$
\begin{array}{lccccccccccc}
\min \; z = & -2x_1 & - & x_2 & + & 3x_3 & & & & & & \\
\text{tel que} & 3x_1 & + & 2x_2 & + & 2x_3 & + & x_4 & & & = & 6 \\
 & -x_1 & + & 2x_2 & + & -x_3 & & & + & x_5 & = & 1 \\
 & x_1, & & x_2, & & x_3, & & x_4, & & x_5 & \geq & 0
\end{array}
$$

Question 4

Soit le système

$$x_1 + x_2 \leq 1$$
$$x_1 + x_2 \geq 2$$
$$x_1, x_2 \geq 0$$

Utilisons la phase I du simplexe pour montrer que ce système ne possède pas de solution

On converti le problème sous forme standard à l'introduction de deux variables artificiels x_3, x_4, on a;

$$\begin{array}{lccccc}
\min & & & x_3 & +x_4 & \\
\text{tel que} & x_1 & +x_2 & +x_3 & & = 1 \\
 & x_1 & +x_2 & & -x_4 & = 2 \\
 & x_1, & x_2, & x_3, & x_4 & \geq 0
\end{array}$$

On obtient le problème sous forme de tableau

$$\begin{array}{cccccc}
 & x_1 & x_2 & x_3 & x_4 & b \\
 & 1 & 1 & 1 & 0 & 1 \\
 & 1 & 1 & 0 & -1 & 2 \\
c^T & 0 & 0 & 1 & 1 & 0
\end{array}$$

Pour avoir les colonnes de la matrice identité, on multipli la deuxième ligne par -1 on a;

$$\begin{array}{cccccc}
 & x_1 & x_2 & x_3 & x_4 & b \\
 & 1 & 1 & 1 & 0 & 1 \\
 & -1 & -1 & 0 & 1 & -2 \\
c^T & 0 & 0 & 1 & 1 & 0
\end{array}$$

Pour obtenir une solution de base réalisable, il faut annuler les coefficients de x_3 et x_4.
Nous obtenons le nouveau tableau

$$\begin{array}{cccccc}
 & x_1 & x_2 & x_3 & x_4 & b \\
 & 1 & 1 & 1 & 0 & 1 \\
 & -1 & -1 & 0 & 1 & -2 \\
c^T & 0 & 0 & 0 & 0 & 1
\end{array}$$

On ne peut donc pas continuer la résolution du système en utilisant le simplexe comme d’ordinaire car on a un système linéaire incompatible qui n'a aucune solution.

QED

La partie en julia;

In [8]:
```
function pivot!(M::Matrix, i::Int, j::Int)
    m, n = size(M)
    @assert M[i, j] != 0
    M[i, :] = M[i, :]/M[i, j]
    for k in setdiff(1:m, i)
        M[k, :] -= M[k, j] * M[i, :]
    end
    return M
end

function getReducedCosts(M::Matrix)
    m, n = size(M)
    return M[end, 1:n-1]
end
function getxB(M::Matrix)
    m, n = size(M)
    return M[1:m-1, end]
end
function enteringVar(M::Matrix)
    rc = getReducedCosts(M)
    index = argmin(rc)
    return rc[index] >= 0 ? -1 : index
end
function exitingVarIndex(M::Matrix{T}, enteringVar::Int) where T
    col = M[1:end-1, enteringVar]
    xB = getxB(M)
    m, n = size(M)
    index = -1
    val = T(Inf)
    for i in 1:m-1
        if (col[i] > 0) && (xB[i]/col[i] < val)
            val = xB[i]/col[i]
            index = i
        end
    end
    return index
end
function isOneHot(v::Vector)
    n = length(v)
    return (sum(iszero, v) == n-1) && (sum(isone, v) == 1)
end
function isoptimal(M)
    return enteringVar(M) == -1
end
function findInitialBasis!(M::Matrix)
    m, n = size(M)
    m-=1
    n-=1
    basis = [-1 for _ in 1:m]
    for i in 1:n
        if isOneHot(M[1:end-1, i])
            index = findfirst(isone, M[:, i])
            basis[index] = i
```

```julia
        end
    end
    @assert !any(t-> t == -1, basis) "problem not caconical"
    for i in 1:m
        j = basis[i]
        pivot!(M, i, j)
    end
    return basis
end
function simplexSolver(A::Matrix{T}, b::Vector, c::Vector; verbose::Bool = false) w
    M = [A b; c' 0]
    basis = findInitialBasis!(M)
    k = 1
    nmax = 1000
    while !isoptimal(M) && k < nmax
        k+=1
        verbose && display(M)
        entering = enteringVar(M)
        entering == -1 && (println("------------"); return T[-1, -1])
        leaving = exitingVarIndex(M, entering)
        leaving == -1 && (println("------------"); return T[-1, -1])
        verbose && @show (entering, leaving)
        basis[leaving] = entering
        pivot!(M, leaving, entering)
    end
    verbose && display(M)
    m, n = size(M)
    xstar = zeros(T, n - 1)
    xstar[basis] = getxB(M)
    return xstar
end
```

```
simplexSolver (generic function with 1 method)
```

In [9]:

```julia
function  devoir3_4()

    A=[1//1 1 1 0 ;
          -1 -1 0 1;
    ]

    b=[1, -2]

    c = [0, 0, 1, 1]
    simplexSolver(A, b, c, verbose=true)
end

devoir3_4()
```

```
3×5 Matrix{Rational{Int64}}:
  1//1   1//1  1//1  0//1   1//1
 -1//1  -1//1  0//1  1//1  -2//1
  0//1   0//1  0//1  0//1   1//1
```

```
4-element Vector{Rational{Int64}}:
  0//1
  0//1
  1//1
 -2//1
```