Université de Nantes
Master informatique parcours “Optimisation en Recherche Opérationnelle”

Installer JULIA, GLPK, JuMP sur MacOSX pour la résolution de LP/MIP

Xavier Gandibleux

November 13, 2016

Contents

Chapter 1

Installation

Marche à suivre validée sur MacOSX :

1.1 Installer JULIA

Récupérer JULIA :
`http://julialang.org/downloads/`

1.2 Installer GLPK

Récupérer GLPK :
`http://www.gnu.org/software/glpk/#downloading`

Installer sur MacOSX :
`http://hichenwang.blogspot.fr/2011/08/fw-installing-glpk-on-mac.html`

1.3 Installer JuMP pour GLPK, ainsi que le wrapper GLPKMathProgInterface

Mise à jour si nécessaire des packages de votre distribution
```
Pkg.update()
```

Ajouter le package JuMP (à faire une et une seule fois)
```
Pkg.add("JuMP")
```

Ajouter le package GLPKMathProgInterface (à faire une et une seule fois)
```
Pkg.add("GLPKMathProgInterface")
```

Ajouter le package GLPK (à faire une et une seule fois)
```
Pkg.add("GLPK")
```

Chapter 2

Utiliser JuMP avec GLPK et GNUmathprog

2.1 Packages à invoquer à chaque nouvelle session

Invoquer l'utilisation de JuMP
```
using JuMP
```

Invoquer l'utilisation de GLPKMathProgInterface
```
using GLPKMathProgInterface
```

Invoquer l'utilisation de GLPK
```
using GLPK
```

2.2 Sélectionner le solveur pour le modèle à résoudre

```
# LP
solver=GLPKSolverLP()

# MIP
solver=GLPKSolverMIP()
```

2.3 Exemple d'un LP : problème basique à 2 variables

```
mBasique = Model(solver=GLPKSolverLP())

# --- Indices, donnees, variables ---
  @variable(mBasique, xA >= 0)
  @variable(mBasique, xB >= 0)

# --- Modèle à résoudre, résolution ---
  @setObjective(mBasique, Max, xA + xB)
  @addConstraint(mBasique, 2*xA - xB <= 8)
```

```
  @addConstraint(mBasique, -xA + 5xB <= 5)
  println("Probleme a resoudre :")
  print(mBasique)
  status = solve(mBasique)

# --- Extraction des résultats ---
  println("z  = ", getObjectiveValue(mBasique))
  println("xA = ", getvalue(xA))
  println("xB = ", getvalue(xB))
```

2.4 Exemple d'un IP : problème de sac-à-dos (01UKP)

```
mKnapsack = Model(solver=GLPKSolverMIP())

# --- Indices, donnees, variables ---
  @variable(mKnapsack, x[1:7], Bin)
  profit = [ 112, 90, 15, 12, 12, 9, 26]
  poids = [ 16, 15, 3, 3, 4, 3, 13 ]
  capacite = 35

# --- Modèle à résoudre, résolution ---
  @objective(mKnapsack, Max, dot(profit, x))
  @constraint(mKnapsack, dot(poids, x) <= capacite)
  status = solve(mKnapsack)

# Extraction des résultats ---
  println("z = : ", getobjectivevalue(mKnapsack))
  for i = 1:5
    println("x[$i] = ", getvalue(x[i]))
  end
```

Chapter 3

Visualisation des solutions

3.1 Script de visualisation des valeurs des solutions $zInf,\ zSup,\ zOpt,\ z(x)$

```
using PyPlot

# Valeur inf de la fonction
  vInfFct =0
# Valeur sup de la fonction
  vSupFct =10

 # Parametres de l'ordonnee
   ylim(-1, 1)
   ycentre = 0

# Trace l'axe central
   xlim(vInfFct, vSupFct)
   plot([vInfFct,vSupFct], [ycentre,ycentre], "k", linewidth=0.5)
   xlabel(L"$f(x), >$", fontsize=15.0)

 # Trace la borne primale
  zInf = 2
  annotate(L"$z_{Inf}$",  xy = (zInf, 0.1), xytext = (zInf, 0.1), fontsize=15.0,
           color = "blue")
  scatter(zInf, ycentre, c = "blue", s=100)

 # Trace la borne duale
  zSup = 9.1
  annotate(L"$z_{Sup}$",  xy = (zSup, 0.1), xytext = (zSup, 0.1), fontsize=15.0,
           color = "red")
  scatter(zSup, ycentre, c = "red", s=100)

 # Trace la valeur optimale
  zOpt = 8.3
  annotate(L"$z_{Opt}$",  xy = (zOpt, 0.1), xytext = (zOpt, 0.1), fontsize=15.0,
```

```
          color = "green")
  scatter(zOpt, ycentre, c = "green", s=100)

# Valeurs de z
  zx = [] ; zy = []
  for i = 1:10
    push!(zx, zInf + rand()*(zOpt-zInf))
    push!(zy, ycentre)
  end
  scatter(zx, zy, c = "yellow")
```

La figure 3.1 illustre le résultat produit par ce script :

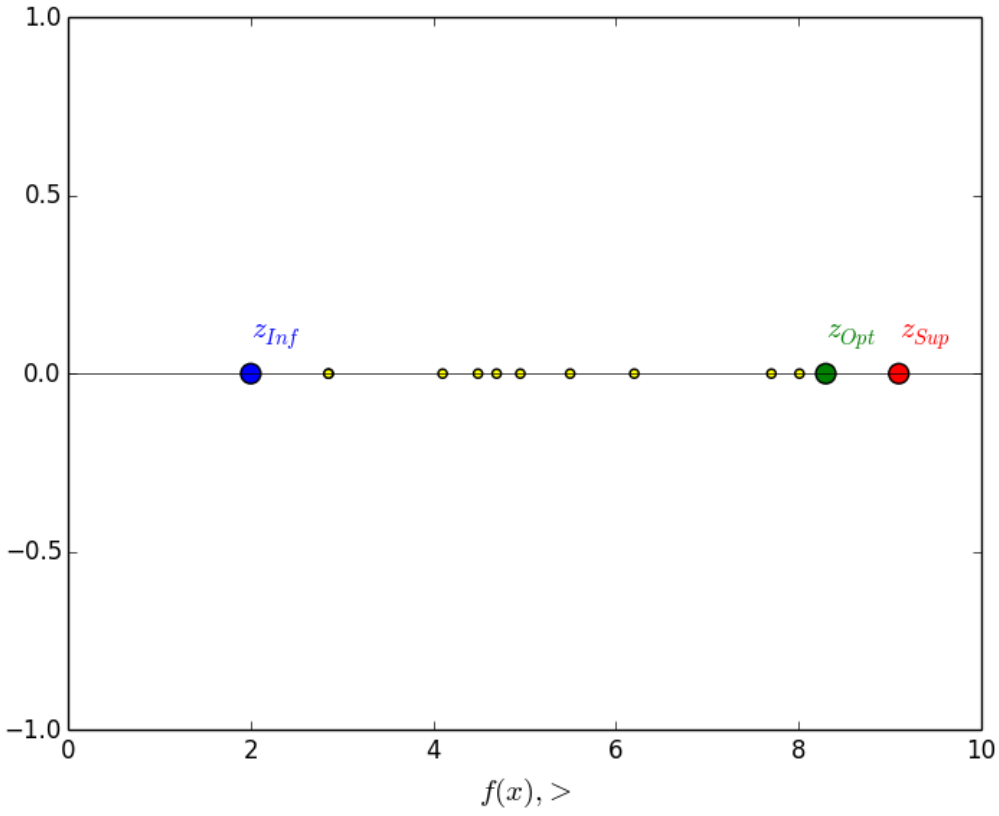

Figure 3.1: Valeurs des solutions remarquables et visitées lors de la résolution

Chapter 4

Liens utiles

https://jump.readthedocs.io/en/latest/installation.html#getting-solvers

http://www.juliaopt.org/JuMP.jl/0.14/index.html

https://github.com/JuliaOpt/GLPK.jl

http://www.juliaopt.org/notebooks/Shuvomoy%20-%20Getting%20started%20with%20JuMP.html