

Recherche Opérationnelle :
TP 1 : modélisation et résolution de PLNE avec un solveur
(GLPK)

Sandra U. Ngueveu, Corentin Boennec, Arthur Clavière, Aloïs Duguet
ngueveu@laas.fr

2021

Table des matières

1	Prise en main de GNU MathProg et de GLPK	2
2	Rapport de TP attendu : Instructions	2
3	BONUS : modélisation et résolution de PL(NE) avec Julia/JuMP	3

Le but est d'implémenter et résoudre différents problèmes vus et modélisés en TD, à l'aide du solveur opensource glpsol de GLPK.

Rendus attendus (à soumettre sur moodle à l'emplacement dédié) : archive .zip contenant :

- les fichiers .mod et .dat ou les fichiers .lp permettant de définir les modèles
- les fichiers solutions obtenus grâce à l'option -o de glpsol pour chaque modèle
- un rapport synthétique (3 pages max) au format pdf ou un notebook en julia

1 Prise en main de GNU MathProg et de GLPK

Télécharger de Moodle les tutoriels d'installation et d'utilisation du solveur glpsol de GLPK, ainsi que les fichiers d'exemples illustratifs.

Exécuter le script "Instructions_GLPK.txt" à partir du terminal, pour télécharger puis compiler glpsol.

Tester les instructions du tutoriel sur les fichiers d'exemples pour prendre en main le solveur.

Il est aussi possible d'exécuter des commandes comme dans le terminal depuis un notebook julia en entourant la commande de backticks et en utilisant la fonction run : run('./glpsol -lp model.txt -o sol.txt'). Julia implémente sa propre commande cd : cd("Documents")

Eléments de compréhension des logs de GLPK :

Résoudre le problème de ciment modélisé au format lp dans le fichier "PbCiment.lp.txt". Repérer dans l'affichage obtenu les différentes étapes de résolution du problème :

- lecture des données
- manipulation de la matrice A
- Résolution de la relaxation PL initiale : quelle est la valeur de la solution du PL trouvée ?
- optimisation en variable entière : quel est l'algorithme de PL utilisé pour cette partie ? Pourquoi ? Quelle est la valeur de la solution du PLNE trouvée ?

2 Rapport de TP attendu : Instructions

Ce rapport de 3 pages maximum pourra être inclus dans un notebook. Il participera à la note globale de TP du cours de Recherche Opérationnelle. Il est à rendre en binôme du même groupe de TP.

Le fichier sera à rendre au plus tard le **mercredi 24 novembre 23h55** sur moodle.

Pour chacun des exercices 1.1 à 1.4 (exercice 2 en bonus) du sujet de TD, modéliser le problème au format lp ou mod/dat suivant ce qui est le plus adapté, et donner, par exemple :

- Ce qui a motivé votre choix de format (lp ou mod/dat) de la modélisation
- Quelques détails sur les points clés et non triviaux de votre modélisation (ce que représentent vos variables de décisions, une contrainte élaborée ...)
- Une courte argumentation de l'adéquation du résultat avec l'instance résolue (la solution obtenue fait-elle sens dans le contexte défini par l'énoncé ?)
- Quelques éléments d'analyse, par exemple :
 - Pour PL : Les matrices de ces exemples sont-elles creuses ? (Dans la pratique, il est fréquent qu'une contrainte ne rassemble que de 5 à 10 variables.)

- Pour PLNE : En combien d'itérations est trouvée la solution optimale continue ? Combien de fois GLPK a amélioré la meilleure solution entière ? Combien d'itérations du simplexe ont été nécessaires ? Combien de nœuds de l'arbre ont été explorés ?

3 BONUS : modélisation et résolution de PL(NE) avec Julia/JuMP

L'intérêt de JuMP est de proposer une interface unique à partir de laquelle les solveurs utilisés sont interchangeables à quelques options près. Pour le bonus de ce TP vous pourrez utiliser CBC ou GLPK qui sont opensource. La liste complète des solveurs disponibles est accessible à l'adresse <https://jump.dev/JuMP.jl/stable/installation/> (avec des instructions pour les installer et les utiliser avec JuMP).

Modéliser et résoudre avec Julia/JuMP les problèmes de ce TP et comparer les résultats et performances des solveurs CBC et GLPK. A titre illustratif, ne pas hésiter à récupérer sur Moodle un notebook montrant un exemple de modélisation et résolution avec Julia/JuMP+solveur PLNE du problème de fabrication de ciment vu en cours.

Quelques instructions utiles :

- ouvrir jupyter-notebook à partir du terminal

```
>> jupyter-notebook
```

- installer le kernel julia dans jupyter-notebook (s'il n'y est pas déjà), à partir du terminal

```
>> julia
```

```
>> using Pkg
```

```
>> Pkg.add("IJulia")
```

attendre la fin de la compilation et relancer jupyter-notebook. Il devrait y être possible de choisir julia comme kernel