

Documentation du bouzin Solveur de problèmes LCP

EH

6 décembre 2010

Table des matières

1	Introduction	1
2	Utilisation du programme	2
2.1	Lancement	2
2.2	Format du fichier	2
2.2.1	flcp	2
2.2.2	jeu bimatriciel	2
2.3	Liste des méthodes implémentées	3
3	Modèle Mathématique	3
3.1	Modèle de LCP	3
3.2	Modèle de jeu bimatriciel sous forme de LCP	3
3.3	Descriptif des méthodes de résolutions	3
3.3.1	Lemke I augmentee	3
3.3.2	Lemke Scheme II unaugmented problem	4
4	FAQ	4
5	Appendix : Exemple de fichiers	4
6	Appendix : Notes sur la programmation	6

1 Introduction

Ci joint la documentation du petit programme qui calcule soit les solutions d'un problème de complémentarité linéaire soit d'un jeu à deux joueurs bimatriciel.

Le descriptif a deux parties. L'une qui décrit le fonctionnement du programme et l'autre qui décrit les bases mathématiques.

2 Utilisation du programme

Le programme se présente sous la forme d'un fichier exécutable (Ap.bouzin pour Apple, Lin.bouzin pour linux et Win.bouzin.exe pour windows) et d'un fichier de paramétrage qui doit obligatoirement s'appeler inputs.dat et qui décrit les valeurs pour lesquelles on veut résoudre le problème.

2.1 Lancement

Il y a deux manières de lancer le programme soit en mode console soit par double clic sur une icône. **Attention** dans les deux cas, le fichier d'entrée doit être dans le même répertoire que le fichier exécutable .

- **double clic.** Cette solution est adaptée au cas *Windows*. Il suffit de double cliquer sur l'icône dont le nom est l'exécutable (pour *Windows* le nom est Win.bouzin.exe), pour lancer le programme. Une fenêtre doit s'ouvrir dans laquelle s'afficheront alors les résultats ¹.
- **en mode console.** Cette méthode marche quelque soit le système. Ouvrir un terminal ou une console de commande selon le système. Il faut aller dans le répertoire contenant l'exécutable (au moyen de la commande `cd`). Puis taper le nom de l'exécutable pour le lancer. Les résultats (*si il existe une solution*) s'affichent dans la console.

2.2 Format du fichier

Le format du fichier est à respecter strictement pour qu'il soit lu (si les sauts de lignes et les espaces ne sont pas respectés des erreurs apparaîtront). Attention le fichier doit être écrit dans un éditeur de texte (style bloc note de windows) dans le même système que celui dans lequel on va lancer le programme (un fichier écrit sous linux et lancé en windows ne marche pas). Des exemples de fichiers sont donnés en section 5.

La première ligne contient le type de problème (*flcp* ou *jeu bimatriciel*) ainsi que la méthode de résolution. La liste des méthodes et leur mots clef est donnée en section 2.3. Ainsi, par exemple `#type: flcp` indique que le problème est un lcp.

Ensuite,

2.2.1 flcp

- la dimension m sur la ligne suivante
- une ligne vide
- le nom de la matrice
- une matrice carrée de dimension m
- puis une ligne vide
- le nom du vecteur
- puis un vecteur de dimension m

2.2.2 jeu bimatriciel

- une dimension m sur la ligne suivante
- une dimension n sur la ligne suivante
- une ligne vide

1. Pour lancer en double cliquant sous apple et linux il faut écrire un script.

- un nom de matrice
- puis une matrice de dimension $m \times n$
- puis une ligne vide
- un nom de matrice
- puis une matrice de dimension $m \times n$

On partira de la convention qu'une matrice de m lignes n colonnes sera donnée dans le fichier sous la forme de m lignes de n entrées, chacune des entrées étant séparée par un espace. On supposera également que le nom de la matrice précède les entrées.

2.3 Liste des méthodes implémentées

La liste des méthodes est la suivante :

- Résolution d'un jeu bimatriciel. Ligne d'appel `#type: bimatrx`
- Résolution d'un problème de complémentarité linéaire avec la méthode classique de Lemke augmentée. Ligne d'appel `#type: flcp`.

Le description de ces méthodes provient de [1] et [2] et le rappel de leur justification mathématique est rappelée section suivante.

3 Modèle Mathématique

Dans cette partie une description un peu plus formelle est effectuée.

3.1 Modèle de LCP

We consider the model

$$w = Mz + q$$

avec $w \geq 0$, $z \geq 0$ et $w_i z_i = 0 \forall i$. It is under these assumptions that the model should be given. Nevertheless from a resolution point of view, we solve it under the form :

$$q = w - Mz$$

and we built a matrix $I - M$.

3.2 Modèle de jeu bimatriciel sous forme de LCP

3.3 Descriptif des méthodes de résolutions

Nous présentons ici une brèves description du fonctionnement des méthodes de résolution. Celles-ci sont décrites pages 267 à 276 de [1].

3.3.1 Lemke I augmentee

La méthode appelée *Lemke scheme I - augmented problem* est décrite page 267 de [1] et décrite page 77 dans [2].

Algorithme de résolution et convergence We add the vector **1** to the system.

Methode consiste en creer une solution presque realisable. Puis de parcourir toutes les solutions presque realisables (pour un indice on a pas le produit nul) avant de trouver celle qui est realisable

Convergence en temps fini Si $q \geq 0$ alors il y a une solution evidente.

If the problem is non degenerate then the algorithm will finish in a finite number of step. Theoreme 4.4.8 of Cottle. It finishes but can not find the solution. Hence two finishing manners occurs either a complementary feasible solution or a ray termination (meaning there is no feasible solution).

theoreme 4.4.10 page 276 donne une condition necessaire
matrice co positive et matrice semi monotones

3.3.2 Lemke Scheme II unaugmented problem

Algorithme de résolution et convergence

Convergence en temps fini

4 FAQ

Pas de questions pour l'instant.

Références

- [1] R. W. Cottle, J.-S. Pang, and R.E. Stone. *The linear complementarity problem*. Computer Science and Scientific Computing. Academic Press, 1992.
- [2] K.G. Murty. *Linear complementarity, linear and nonlinear programming*. Sigma Series in Applied Mathematics. Heldermann Verlag, http://ioe.engin.umich.edu/people/fac/books/murty/linear_complementarity_webbook, 1988.

5 Appendix : Exemple de fichiers

Un exemple de fichier pour un jeu.

```
#type: bimatrice
3
2

A
1 2
1 2
3 4

B
1 2
1 2
3 4
```

Exemple pour un LCP

```
#type: flcp  
4
```

```
M  
1.0 -1.0 -1.0 -1.0  
-1 1 -1 -1  
1 1 2 0  
1 1 0 2
```

```
q  
3 5 -9 -5
```

6 Appendix : Notes sur la programmation