Chapitre 4

Problèmes du flot maximal

Dans le chapitre précédent, nous avons montré avec des illustrations guidées comment utiliser Matlab pour résoudre les problèmes du plus court chemin. Ce chapitre présente des illustrations sur les problèmes du flot maximal. Les algorithmes adoptés par le logiciel Matlab sont : l'algorithme d'Edmonds et de Goldberg.

.

4.1. La fonction graphmaxflow

Calculer le débit maximal dans un graphe orienté.

Syntaxe

```
[MaxFlow,FlowMatrix,Cut] = graphmaxflow(G,SNode,DNode,'Capacity',CapacityValue, ..
.
                                         'Method',MethodValue)
```

`[MaxFlow,FlowMatrix,Cut] = graphmaxflow(G,SNode,DNode)` calcule le flux maximal d'un graphe orienté G du nœud SNode au nœud DNode. L'entrée une matrice creuse $\boldsymbol{N} \times \boldsymbol{N}$. Les entrées non nulles dans la matrice représentent les capacités des arcs.

Argument d'entrée	
G	Graphe orienté représenté par son formalisme mathématique sous-djacent qui est une matrice d'adjacence, spécifiée comme une matrice creuse $\boldsymbol{N} \times \boldsymbol{N}$. Les entrées non nulles dans la matrice représentent les poids des arcs. Types de données: double
SNode	Nœud dans G.
DNode	Nœud dans G.
CapacityValue	Vecteur colonne spécifie les capacités personnalisées pour les arcs de la matrice G. Il doit avoir une entrée pour chaque valeur non nulle (arc) dans la matrice G. L'ordre des capacités personnalisées dans le vecteur doit correspondre à l'ordre des valeurs non nulles dans la matrice G lorsqu'il est parcouru colonne par colonne. Par défaut, `graphmaxflow` obtient les informations de capacité des entrées non nulles dans la matrice G.
MethodValue	Vecteur de chaîne de caractères qui spécifie l'algorithme utilisé pour trouver l'arbre couvrant minimal (ACM). Deux choix sont possibles: o 'Edmonds' - Utilise l'algorithme de Edmonds et Karp, dont l'implantation est basée sur une variante appelée algorithme d'étiquetage ***(labeling algorithm)***. La complexité temporelle est $O(N \times E^2)$, où N et E sont respectivement le nombre de nœuds et d'arcs. o 'Goldberg' - Algorithme par défaut. Utilise l'algorithme de Goldberg, qui utilise la méthode générique appelée preflow-push. La complexité temporelle est $O(2^N \times \sqrt{E})$, où N et E sont respectivement le nombre de nœuds et d'arcs.

Tabeau 4.1. Arguments d'entrée – graphmaxflow –

Argument de sortie	
MaxFlow	Débit maximal.
FlowMatrix	Matrice creuse avec toutes les valeurs de débit pour chaque arc. *FlowMatrix (X,Y)* est le flux du nœud X au nœud Y.
Cut	Vecteur ligne logique indiquant les nœuds connectés à SNode après avoir calculé la coupe minimale entre SNode et DNode. S'il existe plusieurs solutions au problème de coupure minimale, alors *Cut* est une matrice.

Tabeau 4.2. Arguments de sortie – graphmaxflow –

N.B : L'algorithme qui détermine la coupe, toutes les coupes minimales, a une complexité temporelle de $O(2^N)$, où N est le nombre de nœuds. Si ces informations ne sont pas nécessaires, utilisez la fonction graphmaxflow sans la troisième sortie.

Illustration 4.1

Cette illustration montre comment calculer le débit maximal dans un graphe orienté.

Créez un graphe orienté composé de six nœuds et huit arcs.

```
M = sparse([1 1 2 2 3 3 4 5],[2 3 4 5 4 5 6 6],[2 3 3 1 1 1 2 3],6,6)
M =

   (1,2)        2
   (1,3)        3
   (2,4)        3
   (3,4)        1
   (2,5)        1
   (3,5)        1
   (4,6)        2
   (5,6)        3
```

Calculez le débit maximal dans le graphe entre le nœud 1 et le nœud 6.

```
[MF,FMx,Cut] = graphmaxflow(M,1,6)

MF =

     4

FMx =

   (1,2)        2
   (1,3)        2
   (2,4)        1
   (3,4)        1
   (2,5)        1
   (3,5)        1
   (4,6)        2
   (5,6)        2

Cut =

     1     1     1     1     0     0
     1     0     1     0     0     0
```

Notons que `Cut` est une matrice à deux rangées car il existe deux solutions possibles au problème de coupe minimale.

Voir le graphe avec les capacités originales.

```
h = view(biograph(M,[],'ShowWeights','on'))
```

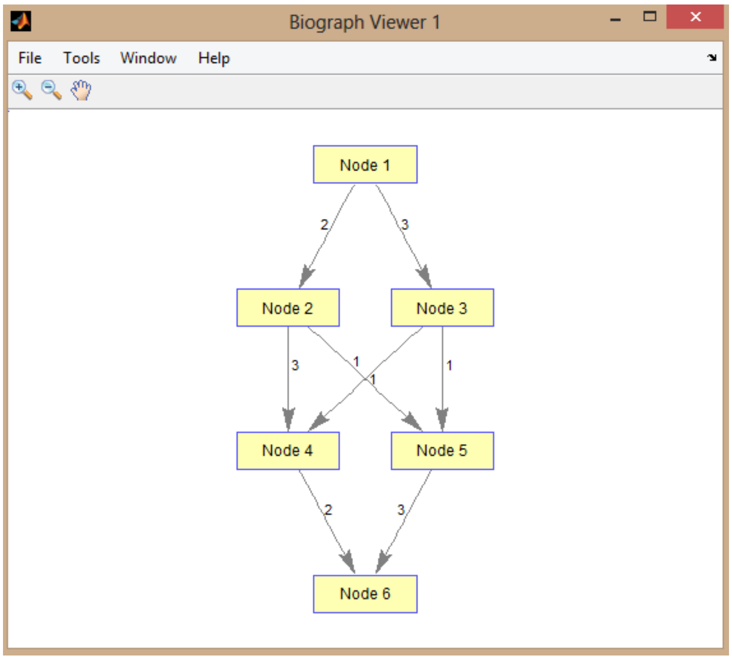

Figure 4.1. Graphe – graphmaxflow –

Un graphe composé de 6 nœuds et 8 arcs.

Affichez le graphe avec les débits maximums calculés.

```
view(biograph(FMx,[],'ShowWeights','on'))
```

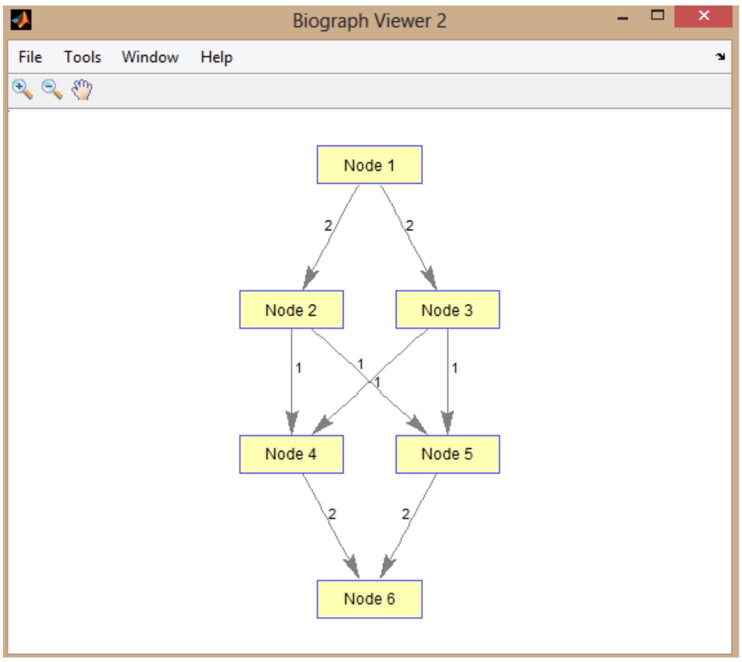

Figure 4.2. Graphe avec débit maximum – graphmaxflow –

Montrez une solution au problème de coupe minimale dans le graphe original.

```
set(h.Nodes(Cut(1,:)),'Color',[0 1 0])
```

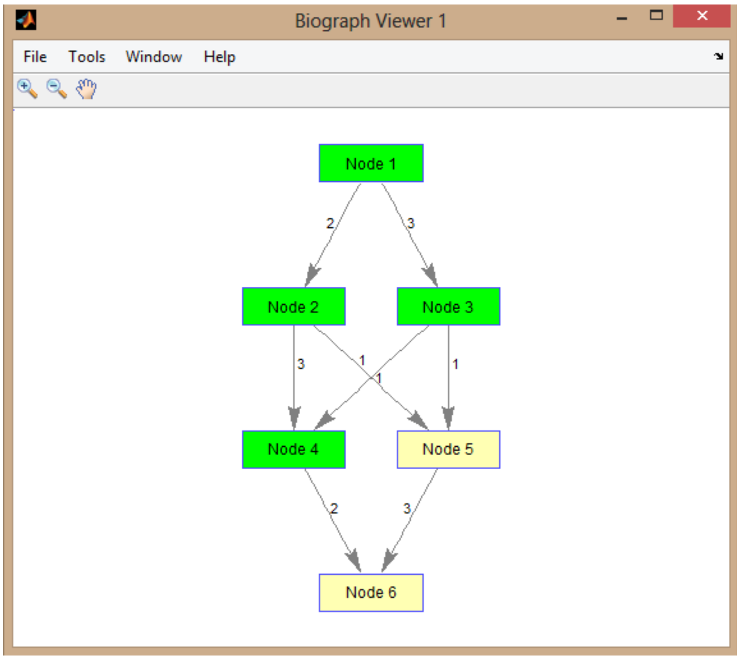

Figure 4.3. Graphe avec coupe minimale – graphmaxflow –