Modélisation du trafic et de la pollution atmosphérique à l'échelle d'une ville
Éléments de réponse

Mathieu Pellerin

October 10, 2017

Contents

1 Introduction

2 Description des outils

2.1 Les graphes et Metanet

2.2 Structure des graphes

2.3 Création d'un graphe avec Metanet

1. Ouvrir, sous Scilab, l'application Metanet par la commande

   ```
   metanet();
   ```

2. Ouvrir un nouveau graphe (files/new), le nommer avec une extension *.graph* et le choisir orienté ;

3. Créer deux nœuds en utilisant le bouton droit de la souris, et un arc orienté en cliquant, avec le bouton droit de la souris sur le premier nœud puis le second. Modifier les caractéristiques par la rubrique **Modify** et les afficher par la rubrique **Graph**.

2.4 Création d'un graphe directement sous scilab

1. On utilise la fonction *make_graph* dont les arguments sont décrits dans l'aide

   ```
   g=make_graph('essai',1,2,[1],[2])
   ```

2. On crée ensuite le fichier *essai.graph* avec la fonction *save_graph*

   ```
   save_graph(g,'essai')
   ```

3. On visualise maintenant le graphe

   ```
   metanet('essai')
   ```

4. Sans créer de fichier, on peut directement visualiser le graphe

   ```
   show_graph(g)
   ```

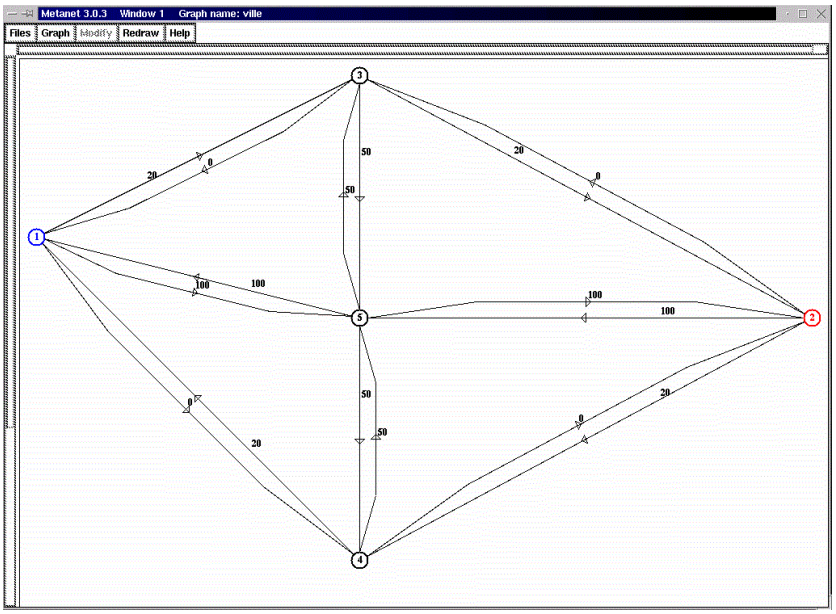

Figure 1: Graphe Metanet de la ville

3 Modélisation d'une ville sous forme de graphe

3.1 Création du graphe

1. Créer par Metanet le graphe correspondant au réseau de la ville. Un tel fichier peut aussi être obtenu par les commandes suivantes sous Scilab :

```
g5=[1,5,1,4,5,2,5,4,4,2,5,3,3,1,3,2];
g6=[5,1,4,1,2,5,4,5,2,4,3,5,1,3,2,3];
g=make_graph('ville',1,5,g5,g6);
g(9)=[130,630,360,360,370];
g(10)=[250,250,55,460,245];
g(24)=[15000,15000,4000,0,15000,15000,6000,
6000,4000,0,6000,6000,4000,0,0,4000];
save_graph(g,'ville')
```

3.2 Génération du trafic

1. Créer le vecteur E donnant les émissions pour toutes les zones.

```
p=[30000,35000,15000,10000,5000];
a=[10000,12000,6000,6000,2000];
alpha1=[0.4,0.4,0.4,0.4,0.5];
```

```
alpha2=[1,1,1,1,1];
E=alpha1.*p+alpha2.*a;
```

2. Créer le vecteur A donnant les attractions pour toutes les zones.

```
e=[4000,4000,2000,2000,24000];
beta=[2.1,2.1,2.1,2.1,2.1];
A=beta.*e;
```

3.3 Distribution du trafic

1. Créer la matrice $5 * 5$, donnant les T_{ij}

```
d=[0,15,10,7,8;15,0,8,6,7;10,8,0,9,5;7,6,9,0,4;8,7,5,4,0];
k=0.00004;
z=E'*A;
gam=0.25;
T=k*z.*exp(-d*gam)
```

3.4 Affectation du trafic

1. Charger le graphe *ville.graph* sous Scilab.

```
g=load_graph('ville');
```

2. Affecter le trafic T_{12} et déterminer les flux générés sur les arcs. En déduire les capacitées maximales restantes à réintroduire dans le graphe. Initialiser les arguments de la fonction $min\_lcost\_cflow()$:

```
c=0;
v=0;
flag=0;
f=zeros(1,16);
```

Garder les valeurs des capacités maximales initiales sous $capa0$:

```
capa0=zeros(1,16);
capa0=g(24);
```

Utiliser la fonction, les flux sont donnés par f :

```
[c,f,v,flag]=min_lcost_cflow(1,2,T(1,2),g);
```

Réintégrer les capacités maximales :

```
g(24)=capa0-f;
```

Visualiser les résultats :

```
show_graph(g);
```

3. En déduire, par une itération permettant d'affecter tous les T_{ij}, les flux totaux sur les tronçons. Les visualiser sous Metanet. Les flux finaux sont visualisés par exemple sur le vecteur 22 du graphe correspondant au coût de l'arc.

```
g=load_graph('ville');
c=0;
v=0;
flag=0;
f=zeros(1,16);
capa0=zeros(1,16);
capa0=g(24);
capa=zeros(1,16);
capa=capa0;
for i=1:5,
for j=1:5,
[c,f,v,flag]=min_lcost_cflow(i,j,T(i,j),g);
capa=capa-f;
g(24)=capa;
end;
end;
g(22)=capa0-g(24);
g(24)=capa0;
show_graph(g);
```

1. Programmer la fonction $Em(V)$, donnant les émissions moyennes par véhicule et kilomètre pour les vitesses V.

   Cette fonction est à enregistrer dans un fichier, nommé pour l'exemple $fonction\_emission$. On remarque que V peut être un vecteur.

```
function [X]=Em(V)
pc=[0.35,0.35,0.2,0.1];
coef_a=[0.6089,0.4767,0.9037,0.9037;-0.0118,-0.0107,-0.0168,-0.0168;
0.0001,0.0001,0.0001, 0.0001];
X=0;
z=size(V);
X=sum((pc'*ones(1,z(1)))'.*([ones(z(1),1),V,V^2]*coef_a),'c');
X
```

Charger la fonction sous SCILAB

```
getf fonction_emission;
```

2. Tracer $Em(V)$ pour V de 0 km/h à 130 km/h.

```
V=[0:130];
Em=Em(V');
plot2d(V,Em')
```

3.5 Émissions sur chaque tronçon

1. Calculer les vitesses sur les tronçons.

```
f=g(22);
c=g(24);
c(4)=1;c(10)=1;c(14)=1;c(15)=1;
V0=[130,130,50,0,130,130,80,80,50,0,80,80,50,0,0,50];
V1=[0.8,0.8,0.5,0,0.8,0.8,0.6,0.6,0.5,0,0.6,0.6,0.5,0,0,0.5];
V=V0.*((1.1*ones(1,16)-f./c)./(1.1*ones(1,16)-V1.*(f./c)))
```

2. Les longueurs des tronçons sont donnés dans l'énoncé. Calculer les émissions de chaque arc, pour les flux calculés précédemment et les visualiser sur le graphe. On décide que le poids de l'arc sous METANET représente les émissions.

```
d=[8,8,7,7,7,7,4,4,6,6,5,5,10,10,8,8]
Em=Em(V');
Em(4)=0;Em(10)=0;Em(14)=0;Em(15)=0;
g(27)=d.*Em'.*f;
show_graph(g);
```

3. Modifier l'épaisseur des arcs proportionnellement aux émissions.

```
g(18)=0.001*g(27);
show_graph(g);
```

4 Résumé du code informatique

4.1 Les fonctions

```
function [X]=Em(V)
pc=[0.35,0.35,0.2,0.1];
coef_a=[0.6089,0.4767,0.9037,0.9037;-0.0118,-0.0107,-0.0168,-0.0168;
```

```
0.0001,0.0001,0.0001, 0.0001];
X=0;
z=size(V);
X=sum((pc'*ones(1,z(1)))'.*([ones(z(1),1),V,V^2]*coef_a),'c');
X
```

4.2 Les commandes

```
g5=[1,5,1,4,5,2,5,4,4,2,5,3,3,1,3,2];
g6=[5,1,4,1,2,5,4,5,2,4,3,5,1,3,2,3];
g=make_graph('ville',1,5,g5,g6);
g(9)=[130,630,360,360,370];
g(10)=[250,250,55,460,245];
g(24)=[15000,15000,4000,0,15000,15000,6000,6000,4000,0,
6000,6000,4000,0,0,4000];
save_graph(g,'ville')
p=[30000,35000,15000,10000,5000];
a=[10000,12000,6000,6000,2000];
alpha1=[0.4,0.4,0.4,0.4,0.5];
alpha2=[1,1,1,1,1];
E=alpha1.*p+alpha2.*a;
e=[4000,4000,2000,2000,24000];
beta=[2.1,2.1,2.1,2.1,2.1];
A=beta.*e;
d=[0,15,10,7,8;15,0,8,6,7;10,8,0,9,5;7,6,9,0,4;8,7,5,4,0];
k=0.00004;
z=E'*A;
gam=0.25;
T=k*z.*exp(-d*gam)
g=load_graph('ville');
c=0;
v=0;
flag=0;
f=zeros(1,16);
capa0=zeros(1,16);
capa0=g(24);

capa=zeros(1,16);
capa=capa0;

for i=1:5,
for j=1:5,
[c,f,v,flag]=min_lcost_cflow(i,j,T(i,j),g);
```

```
capa=capa-f;
g(24)=capa;
end;
end;

g(22)=capa0-g(24);
g(24)=capa0;
show_graph(g);
Em=Em(V');
plot2d(V,Em')
f=g(22);
c=g(24);
c(4)=1;c(10)=1;c(14)=1;c(15)=1;
V0=[130,130,50,0,130,130,80,80,50,0,80,80,50,0,0,50];
V1=[0.8,0.8,0.5,0,0.8,0.8,0.6,0.6,0.5,0,0.6,0.6,0.5,0,0,0.5];
V=V0.*((1.1*ones(1,16)-f./c)./(1.1*ones(1,16)-V1.*(f./c)))
d=[8,8,7,7,7,7,4,4,6,6,5,5,10,10,8,8]
Em=Em(V');
Em(4)=0;Em(10)=0;Em(14)=0;Em(15)=0;
g(27)=d.*Em'.*f;
show_graph(g);
g(18)=0.001*g(27);
show_graph(g);
```