Flots dans des graphes

1) Exemples d'utilisation de la théorie des flots

2) Formalisation et résolution :
 A) Flot maximum
 B) Flot de coût minimum

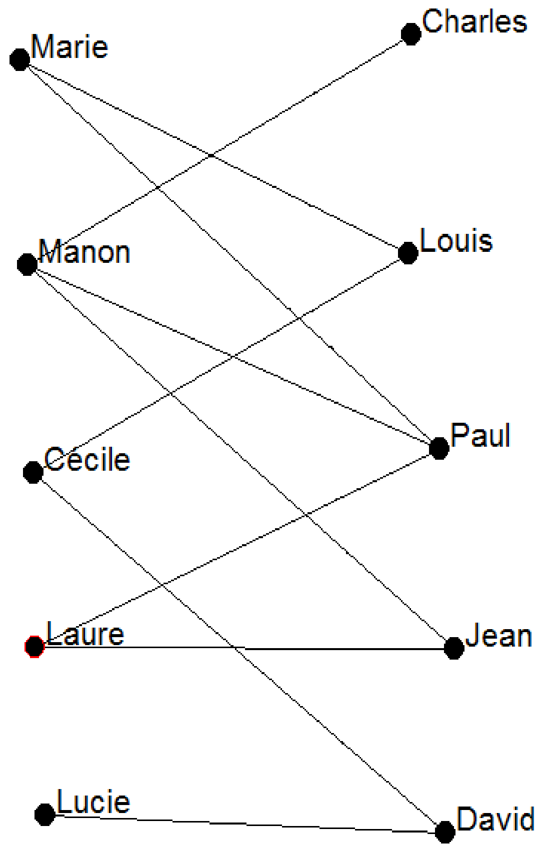
3) Programmation en langage Caml

4) Idées d'approfondissements

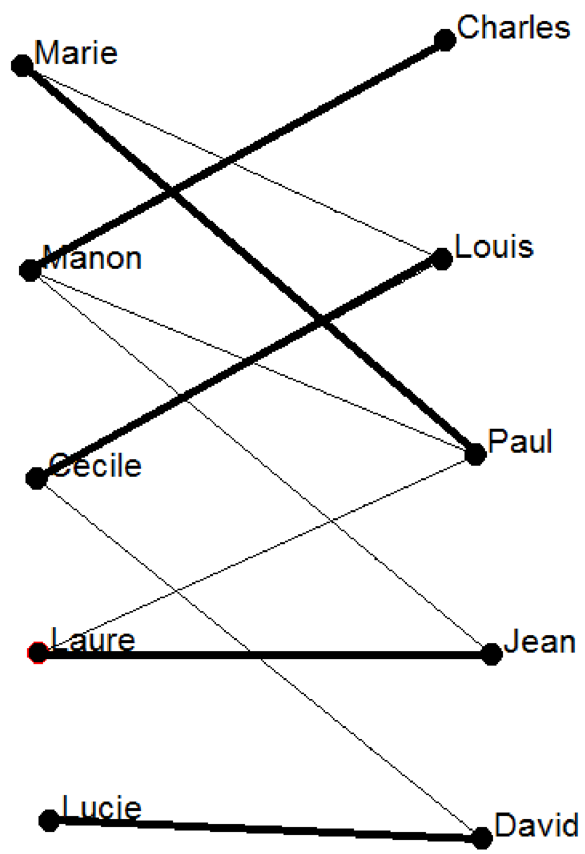
1) Exemples

a) Le problème du couplage maximum dans un graphe biparti (pb des mariages)

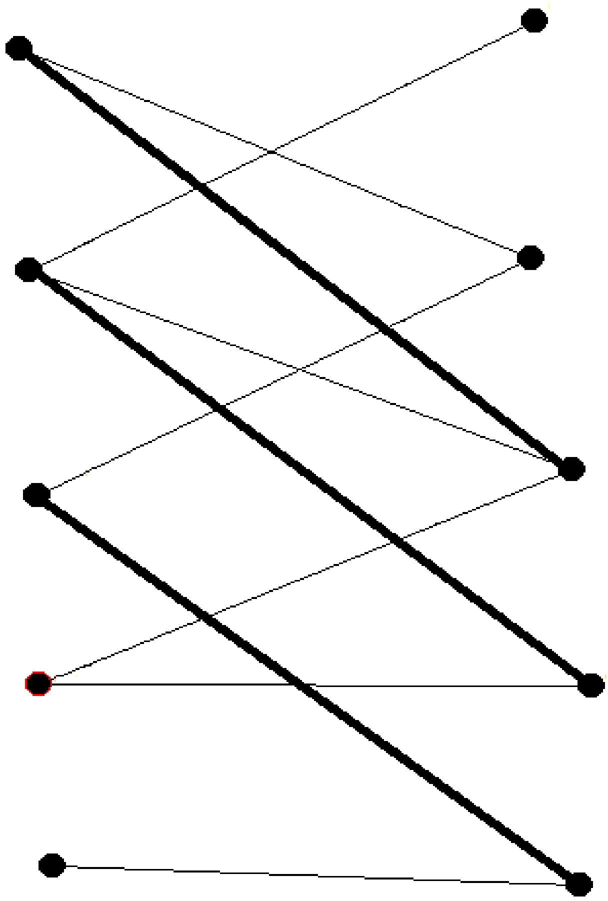
Problème



Une solution



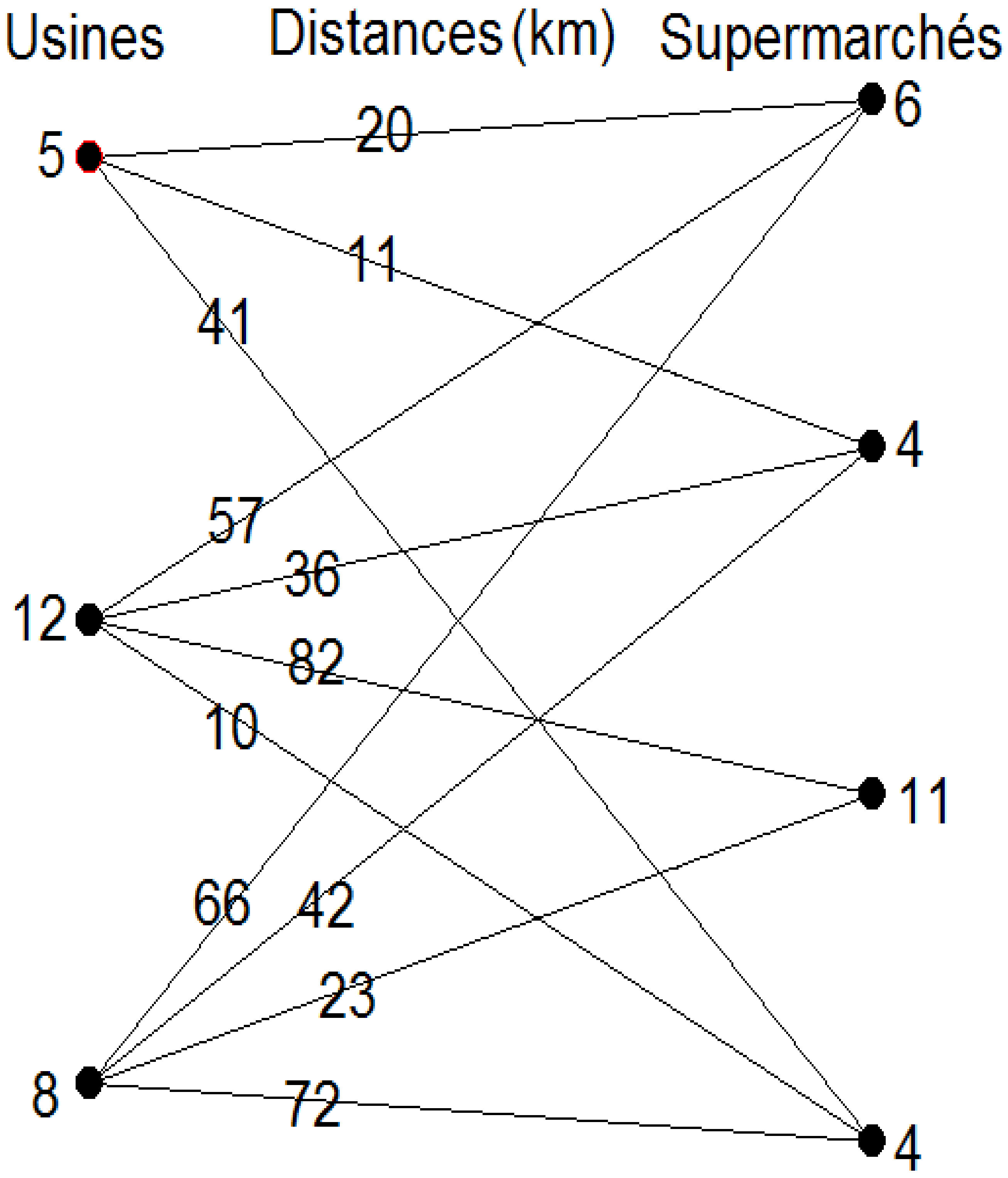
Attention : un couplage maximal n'est pas toujours maximum



La méthode gloutonne naïve ne fonctionne pas

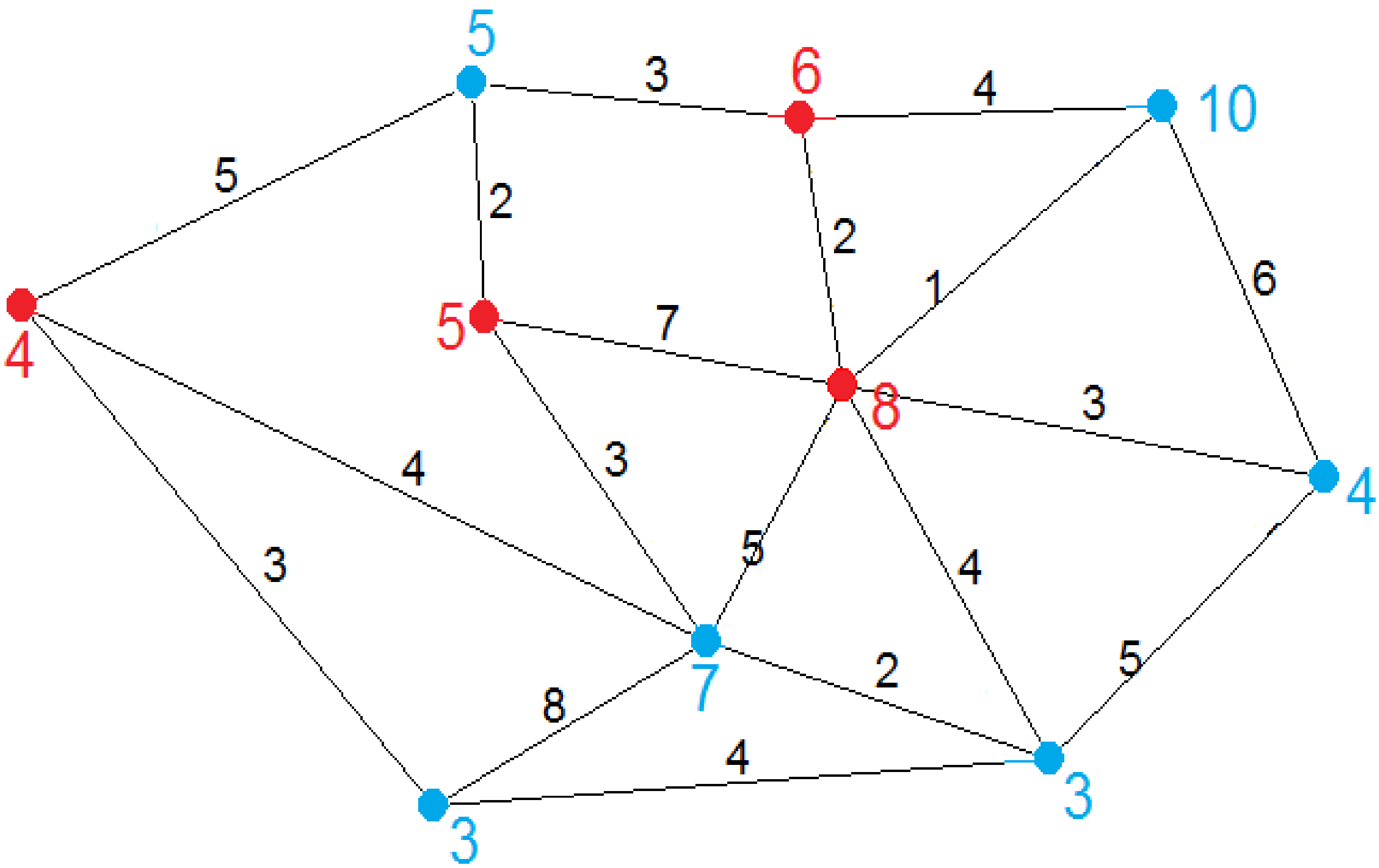
Une autre méthode est nécessaire

b) En lien avec le mouvement : problème du transport optimal (optimisation logistique)



c) Flots sur des graphes non bipartis

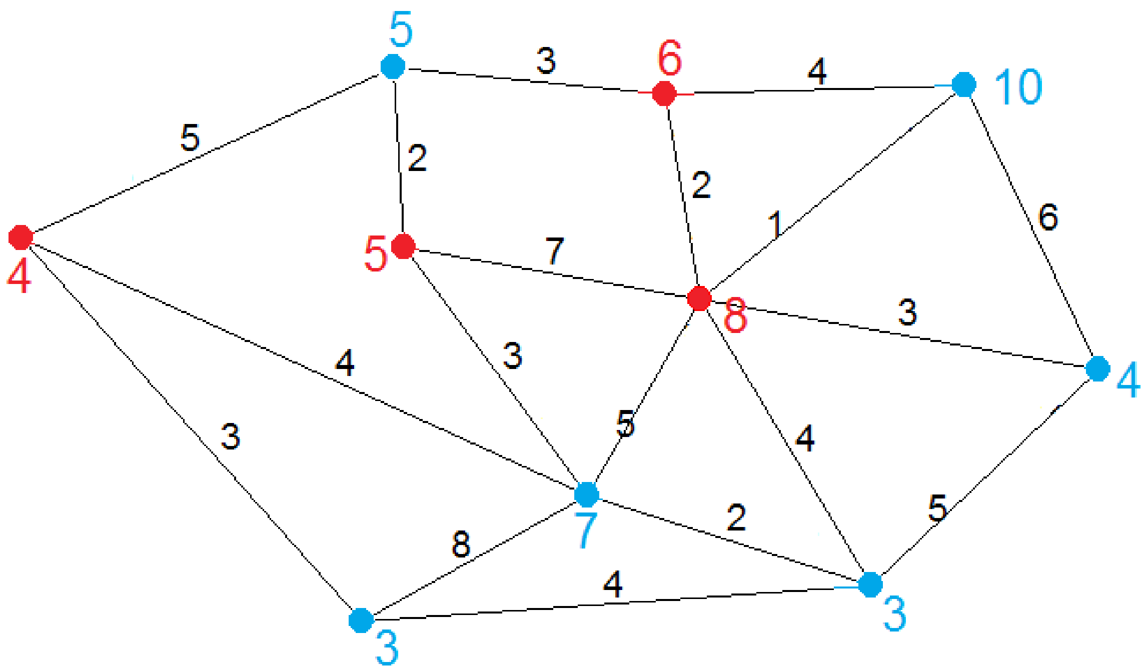
Réseau électrique

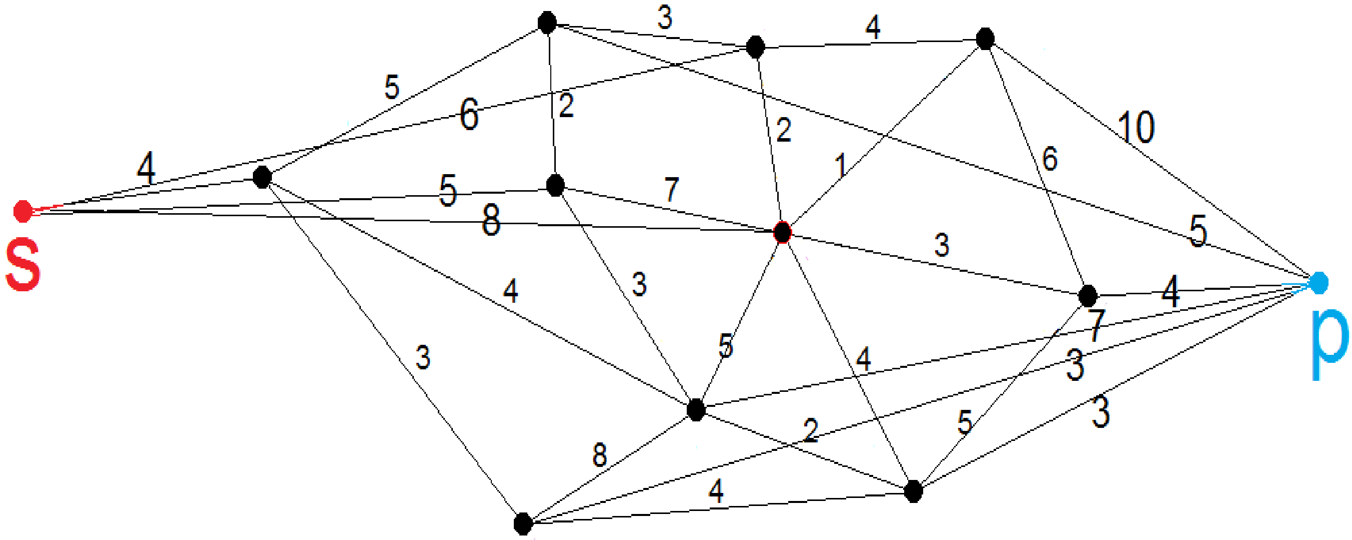


Un point représente une ville
Un trait représente une ligne à haute tension
Un nombre en noir représente la capacité de transfert de puissance maximale d'une ligne
Un nombre en rouge représente un excès de production dans une ville.
Un nombre en bleu représente un déficit.

Objectif : redistribuer un maximum d'énergie en excès pour approvisionner le maximum de gens possible en électricité.

Super-source et super-puit fictifs





2) Formalisation et résolution

A) Flot maximum

A1) Définition

(X,A,u,s,p) un problème de flot maximum :
- (X,A) un graphe orienté
- u une valuation des arcs à valeur positive (représente la capacité des arcs)
- s et p deux sommets distinct de X, appelés respectivement la source et le puit.

Objectif : trouver un flot f, une valuation des arc vérifiant :
- la contrainte des capacités :

$$\forall (x,y) \in A, 0 \leq f(x,y) \leq c(x,y)$$

- la condition de conservation du flot :

$$\forall x_0 \in X \setminus \{s,p\}, e(x_0) = f(X, x_0) - f(x_{0,} X) = 0$$

- qui maximise la valeur du flot e(p) (qui vaut aussi -e(p) car :

$$e(s) + e(p) = \Sigma_{(x \in X)} e(x) = \Sigma_{((x,y) \in A)} (f(x,y) - f(x,y)) = 0$$

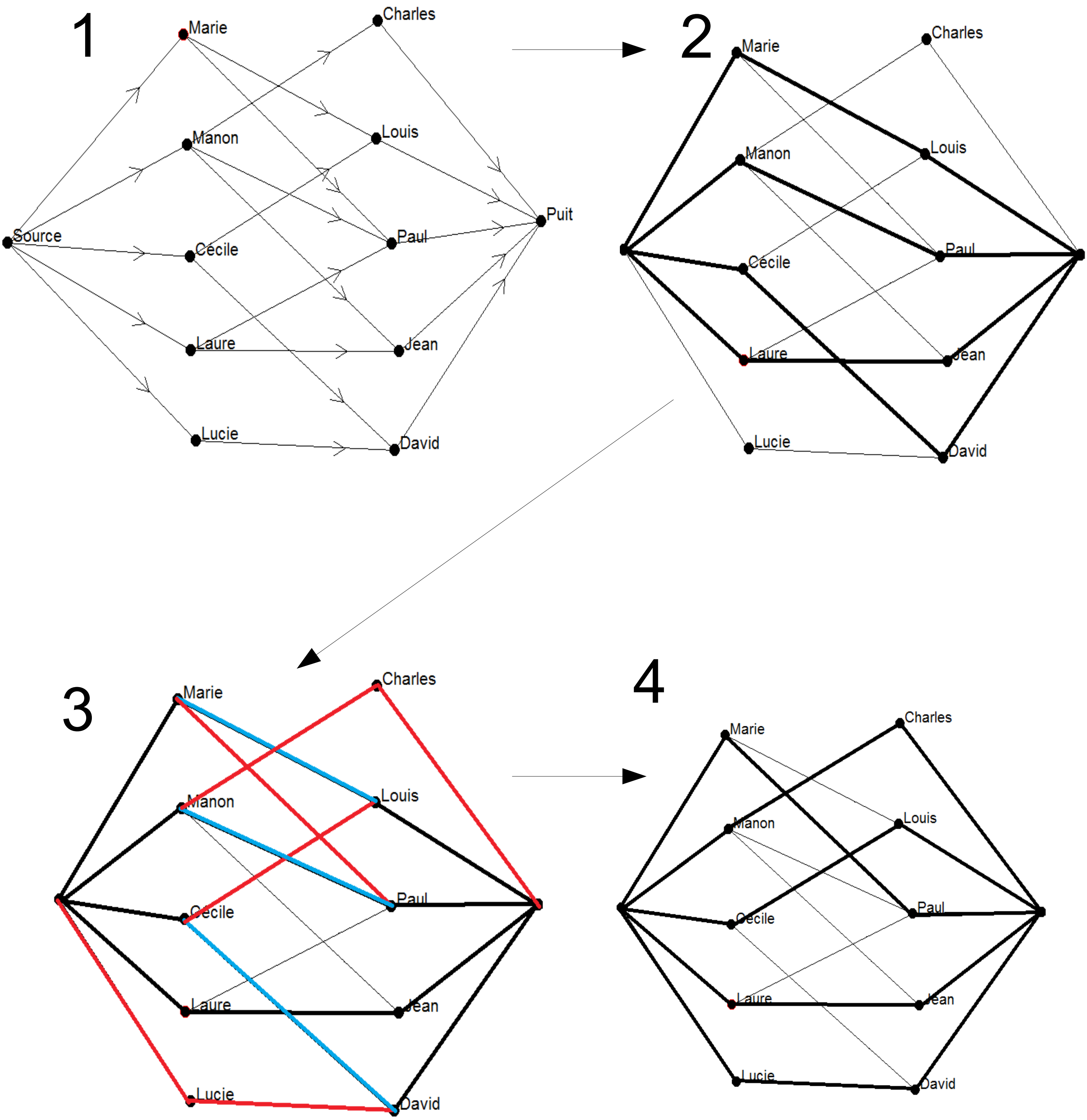
A2) Algorithme de Ford et Fulkerson (F&F)

• Chemin augmentant : chemin de s à p utilisant deux sortes d'arcs :
 - des arcs (x,y) tq $f(x,y) < c(x,y)$
 - des arcs (y,x) tq $f(x,y) > 0$.
(on se donne le droit
- d'augmenter le flot suivant des arcs non saturés
- de le diminuer dans des arcs où un flot passe déjà).

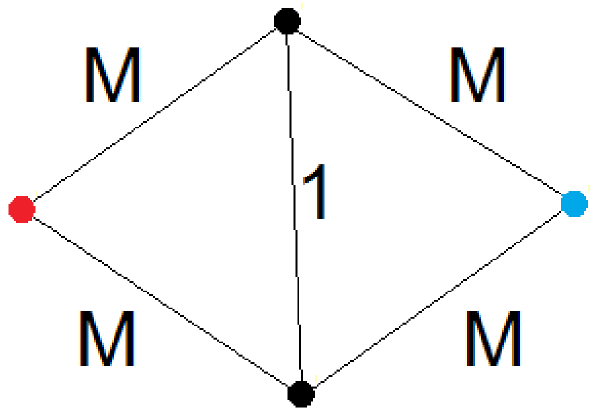
• F&F :
{Tant que (il existe un chemin augmentant)
 { Augmenter le flot autant que possible suivant le chemin augmentant trouvé }
Fin tant que}

Justification de l'algorithme : cf dossier

Exemple


1
Marie
Charles
Manon
Louis
Source
Puit
Cécile
Paul
Laure
Jean
Lucie
David
2
Marie
Charles
Manon
Louis
Cécile
Paul
Laure
Jean
Lucie
David
3
Marie
Charles
Manon
Louis
Cécile
Paul
Laure
Jean
Lucie
David
4
Marie
Charles
Manon
Louis
Cécile
Paul
Laure
Jean
Lucie
David

A3) Amélioration de Edmonds et Karp



Ce graphe peut être résolu en 2M étapes si on s'y prend mal.

FF a une complexité non polynomiale en $|X|$ et $|A|$

Solution de Edmonds et Karp : chercher les chemins de s à p ayant le moins d'arêtes possibles (parcours en largeur)

Complexité en $O(|X|.|A|^2)$ (cf dossier).

B) Flot de coût minimum

Cette fois ci les arcs sont doublement valués : en plus de la capacité u(x,y), on ajoute le coût unitaire de passage c(x,y).

Objectif : trouver, parmi les flots de valeur maximum, un flot minimisant le coût total du flot,

$$c(f) = \Sigma_{(a \in A)} c(a) f(a)$$

B2) Deux variations du problème

- flot de coût minimum pour une valeur de flot fixée
- flot de valeur maximum pour un coût fixé

B3) Résolution du problème de flot maximum de coût minimum

- Comme FF, mais recherche de chemin de longueur minimale en considérant les coûts comme des longueurs

Justification de l'algorithme : cf dossier

3) Programmation en langage caml

Objectif fixé

Créer une interface graphique permettant de rentrer les données d'un problème de flot, calculer sa solution, et l'afficher.

3 parties distinctes

- Structures de données
- Algorithmes
- Partie graphique

A) Structures de données pour coder les graphes doublement valués :

3 possibles :

- (nombre vect) vect vect
- (int * nombre vect) list vect
- (int * nombre vect) avl vect

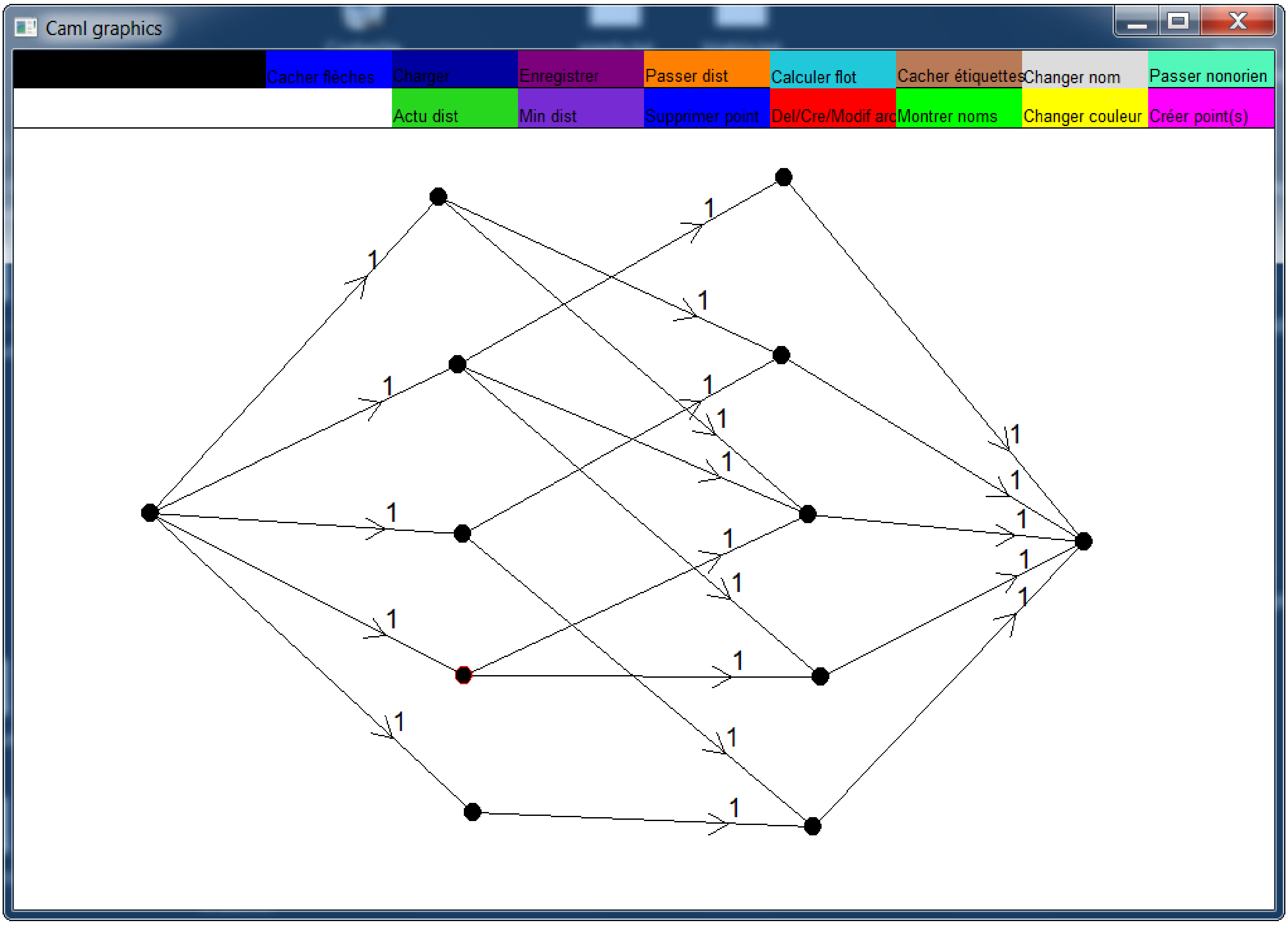
B) Algorithmes codés

Recherche de chemin :
PP, PL, Djikstra, Bellman

Flots maximum : F&F, E&K.

Flots de coût minimum :
Les 3 versions du problème.

C) Interface graphique



Code source et description du logiciel : cf dossier

4) Idées d'approfondissement

- Flots de plusieurs matières
- Flots dynamiques en temps continu ou discrétisé
- Codage d'algorithmes spécifiques aux problèmes de graphes bipartis : algorithme hongrois, algorithme de Kuhn
- Une autre méthode pour maxflow : push-relabel
- 2 autres méthodes pour maxflow-mincost : méthode de la suppression de cycles, méthode du simplexe.