

UV Recherche Opérationnelle  
Compte-Rendu de projet sur l'algorithme de  
Ford-Fulkerson

M. CHAMBREUIL, N. FOURNEL, V. GREMMEL, L. TRAORE et M ZEHNI

16 janvier 2003

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation du sujet</b>	<b>3</b>
2.1	But . . . . .	3
2.2	L'algorithme de Ford-Fulkerson . . . . .	3
2.2.1	Fonction Marquage . . . . .	3
2.2.2	Fonction Augmentation Du Flot . . . . .	4
<b>3</b>	<b>Réalisations/Déroulement du projet</b>	<b>5</b>
3.1	Comprehension de l'Algorithme . . . . .	5
3.2	Définition d'un modèle . . . . .	5
3.3	Codage du modèle et l'algorithme . . . . .	6
3.4	Implémentation de l'interface graphique . . . . .	6
<b>4</b>	<b>Conclusion</b>	<b>7</b>
<b>A</b>	<b>How-To Ford-Fulkerson</b>	<b>8</b>
<b>B</b>	<b>Diagrammes des classes</b>	<b>9</b>

# Chapitre 1

## Introduction

Dans le cadre de l'unité de valeur (U.V.) Recherche Opérationnelle, nous avons été amené à étudier l'agorithme de Ford-Fulkerson. Ce même algorithme est devenu le sujet de notre projet dans cette UV.

Nous présenterons dans un premier temps, l'algorithme de Ford-Fulkerson plus en détails. Dans un deuxième temps nous passerons en revue les différentes phases de la réalisation de ce projet.

## Chapitre 2

# Présentation du sujet

### 2.1 But

Le but est de créer un logiciel appliquant un algorithme de flot maximum. L'algorithme devant être appliqué est l'algorithme de Ford-Fulkerson. Les principales contraintes de ce projet sont que le logiciel doit être graphique, et que celui-ci est à but pédagogique.

On comprendra donc facilement que le langage qui a été choisi pour coder est le Java. Les performances du logiciel ne sont pas le but premier du projet, mais plutôt son ergonomie et son aspect pédagogique.

A ces contraintes, nous avons rajouté de notre propre chef la nécessité d'un système de chargement de Graphe, ainsi que si le temps nous en était donné un éditeur graphique de Graphe. Le système de sauvegarde de Graphe envisagé était le XML.

### 2.2 L'algorithme de Ford-Fulkerson

L'algorithme de Ford-Fulkerson est un algorithme permettant le calcul du flot maximum datant de 1956. Cet algorithme est relativement simple, il consiste en deux principales fonctions.

- Marquage ;
- Augmentation du flot.

On répète ces deux fonctions successivement tant que la première trouve une chaîne améliorante.

#### 2.2.1 Fonction Marquage

Cette fonction est une fonction qui parcourt le graphe en partant de la source vers le puits. Elle a pour but de trouver une chaîne améliorante. Elle effectue cela en marquant les sommets qu'elle peut atteindre. Elle fixe pour chaque sommet marqué un delta et une provenance. Ce delta correspond au

flot maximum que l'on peut faire parvenir au sommet en empreintant la chaîne qui a servi à le marquer. Cette fonction s'arrête donc lorsque le puit est marqué, on alors une chaîne améliorante. Dans le cas où elle s'arrête parce qu'elle ne peut plus trouver de chemin jusqu'au puit, on peut dire que le flot est maximum.

Voici l'algorithmie de cette fonction :

Algorithme de Marquage.

```

 $G \leftarrow (X, U, C), w \leftarrow (p, s), \phi \text{ un flot};$ 
 $\delta \leftarrow \delta(s) \leftarrow C_w - \phi_w; Y \leftarrow \{s\};$ 
while  $p \notin Y \ \& \ \delta > 0$  do
  if  $\exists u = (xy) \setminus x \in Y, y \notin Y \ \& \ \phi_u < C_u$  then
     $Y \leftarrow Y \cup \{y\}; \mathcal{A}(y) \leftarrow u; \delta(u) \leftarrow \min[\delta(x), C_u - \phi_u];$ 
  else
    if  $\exists u = (yx) \setminus x \in Y, y \notin Y \ \& \ \phi > 0$  then
       $Y \leftarrow Y \cup \{y\}; \mathcal{A}(y) \leftarrow u; \delta(u) \leftarrow \min[\delta(x), \phi_u];$ 
    else
       $\delta \leftarrow 0$ 
    end if
  end if
end while
if  $p \in Y$  then
   $\exists$  un chemin de  $s$  à  $p$ 
end if

```

### 2.2.2 Fonction Augmentation Du Flot

Cette fonction a un rôle beaucoup moins complexe. Celui-ci se résume en l'augmentation du flot sur la chaîne améliorante trouvée par la fonction précédente. Pour cela, elle parcourt le graphe du puit vers la source en utilisant sur chaque sommet l'indication de provenance fixée par la fonction marquage.

Voici l'algorithmie de cette fonction :

Algorithme d'augmentation du flot

```

 $x \leftarrow p; \delta \leftarrow \delta_p$ 
while  $x \neq s$  do
   $u = (xy) \leftarrow \mathcal{A}(x); \phi_u = \phi_u + \delta; x \leftarrow y;$ 
end while

```

## Chapitre 3

# Réalisations/Déroulement du projet

Nos réalisations se sont déroulées en différentes phases, qui sont les suivantes :

### 3.1 Compréhension de l'Algorithme

La première étape du projet fut la compréhension de l'algorithme. Cette étape est sans doute la plus importante du projet car elle permet de partir dans la bonne direction dès le début pour les étapes suivantes. L'explication de l'algorithme ayant été faite précédemment, nous ne la répèterons pas ici.

### 3.2 Définition d'un modèle

La deuxième étape du projet a été consacré aux choix techniques. Ces choix techniques ont principalement été centrés sur la définition d'un modèle adéquat pour la programmation de l'algorithme. Pour cela, nous nous sommes inspirés de la documentation disponible sur le sujet. De plus, compte tenu du langage de programmation choisi (JAVA), un certain nombre de caractéristiques nous était offert.

Nous avons donc choisi de modéliser un Graphe sous la forme suivante (Voir Annexe B) :

La représentation repose donc sur trois classes (Graphe, Sommet, Arc).

- Un Graphe est donc une liste de Sommets, une source et un puit.
- Un Arc est un Sommet initial et un Sommet terminal.
- Un Sommet est une liste de Arcs entrants et une liste de Arcs sortants.

Un Arc sortant est un arc ayant pour Sommet initial le Sommet en question, et un Arc entrant a pour Sommet terminal ce même Sommet.

Ceci ne représente que la base du modèle d'un Graphe, il y a été ajouté d'autres attributs pour le codage de l'algorithme en lui-même.

### **3.3 Codage du modèle et l’algorithme**

Une première phase de codage a tout d’abord été initié, celle du codage du modèle et de l’algorithme. Cette phase a consisté en le codage sans interface graphique de l’algorithme, tout en ayant en ligne de mire l’implémentation du système de déroulement pas à pas de celui-ci. Nous avons, lors de cette phase, pu confirmer que nos choix en matière de modélisation était bon. Ce qui explique que celle-ci a été relativement rapide et sans problème.

### **3.4 Implémentation de l’interface graphique**

Aux vus de l’avancement rapide des premières étapes, nous avons pris la décisions d’implémenter la partie déroulement de l’algorithme (le Viewer) mais aussi la partie ditteur de graphe et le système de sauvegarde au format XML.

Une fois l’interface prise en compte, le diagramme des classes devient assez chargé (Voir Annexe B).

## Chapitre 4

# Conclusion

Le bilan est que nous avons atteint les objectifs fixés en début de projet. Pour conclure, on peut dire que ce projet à été particulièrement intéressant.

D'un point de vue pédagogique, le fait de devoir créer un logiciel pédagogique nous a permis de comprendre encore mieux le fonctionnement de l'algorithme de flot maximum.

En ce qui concerne l'aspect programmation, même s'il ne s'agit pas du but principal, il s'avère que ce projet nous a permis de mettre en application un certain nombre de connaissances apprises dans des UV des semestres précédents.

## Annexe A

# How-To Ford-Fulkerson

Pour utiliser notre projet il suffit d'insérer le CDRom fourni dans le lecteur CD de votre PC. Un script s'exécutera et lancera automatiquement le programme.

Dans le cas où cela ne viendrait pas à se passer correctement, vous pouvez lancer le programme en lançant le script à la main. Il s'appelle "Lancer".

Les Sources sont disponibles dans le repertoire source à la racine du CD.

**Annexe B**

**Diagrammes des classes**