RAIRO. RECHERCHE OPÉRATIONNELLE

W. BIENIA
V. LETROUIT

Sur les α-flots

RAIRO. Recherche opérationnelle, tome 31, n° 1 (1997), p. 67-71
<http://www.numdam.org/item?id=RO_1997__31_1_67_0>

Recherche opérationnelle/Operations Research
(vol. 31, n° 1, 1997, pp. 67-71)

SUR LES α-FLOTS (*)

par W. BIENIA ([1]) et V. LETROUIT ([2])

Communiqué par Pierre TOLLA

Résumé. – *La présente note contient quelques remarques sur les résultats de Minoux exposés dans* [2]. *Nous signalons que les méthodes proposées pour résoudre le problème des α-flots maximaux et des flots à taux de sécurité maximal peuvent dégénérer et ne convergent pas toujours vers les solutions optimales. Nous montrons que les solutions de ces problèmes peuvent être obtenues très simplement par des méthodes classiques. Nous proposons un algorithme glouton fortement polynomial, basé sur la méthode de Newton couplée avec l'algorithme de flot maximum. L'efficacité des méthodes dichotomiques classiques pour ces problèmes est signalée.*

Mots clés : Flots avec sécurité dans les réseaux, algorithmes pour les flots, complexité, optimisation combinatoire.

Abstract. – *We take up Minoux ideas which are explained in* [2]. *We point out that the algorithms he has proposed to solve the problem of maximum α-flow and flow with a maximum rate of safety fail. We propose a simple and efficient, greedy, strongly polynomial algorithm based on Newton's method and max-flow algorithm.*

Keywords: Network flows with safety considerations, algorithms for flows, complexity, combinatorial optimization.

La théorie classique des flots est souvent généralisée pour tenir compte des problèmes liés à la sécurité et la fiabilité des réseaux. En 1976 Minoux [2] introduit la notion du α-flot. Nous commençons par un bref rappel de définition dans **1**. Nous citons quelques articles dont les auteurs étudient les différentes généralisations de ce problème. Nous présentons dans **2** les dysfonctionnements de l'algorithme proposé dans [2]. Dans **3** nous proposons un algorithme fortement polynomial, basé sur la méthode de Newton couplée avec l'algorithme classique de flot maximum

(*) Reçu en décembre 1993.

([1]) ARTEMIS. IMAG, BP 53X, 38041 Grenoble Cedex, France.

([2]) ARTEMIS. IMAG, BP 53X, 38041 Grenoble Cedex, France et LICIT, INRETS-ENTPE, 109, avenue Salvador-Allende, 69675 Bron Cedex, France.

Recherche opérationnelle/Operations Research, 0399-0559/97/01/$ 7.00

et nous signalons aussi l'efficacité des méthodes dichotomiques pour ce problème.

1. DÉFINITIONS

Soit $G = (X, U)$ un graphe orienté avec $U = \{u_1, u_2, \ldots, u_M\}$ où $u_1 = (t, s)$ est l'arc « de retour » entre deux sommets particuliers s (source) et t (puits). Chaque arc u_i est muni d'une capacité $c_i \geq 0$. Soit $\mathbf{f} = (f_1, f_2, \ldots, f_M)$ – un flot (vecteur à composantes non négatives, qui vérifie les contraintes de capacité pour chaque arc et de conservation du flux aux sommets de G) et soit un réel $\alpha \in]0\,;\,1[$. On appelle un α-**flot** de s à t un flot $\mathbf{f}$ avec la propriété : $f_i \leq \alpha\, f_1$ pour $i = 2, 3, \ldots, M$. Par la destruction d'un arc quelconque du réseau ($\neq u_1$) on ne peut pas perdre plus qu'une fraction α de la quantité totale du flot f_1 ; 1-α s'interprète donc comme taux de sécurité du flot $\mathbf{f}$. Un α-**flot maximum** c'est un α-flot avec la valeur maximale de f_1.

Minoux [2] pose deux problèmes :

1.1. Comment déterminer un α-flot maximum pour un α donné ?

1.2. Comment répartir le flot $f = (f_1, f_2, \ldots, f_M)$ avec f_1 donné pour minimiser α ?

De nombreuses généralisations de ces problèmes ont été étudiées – citons-en quelques unes :

– flots submodulaires, entiers, rationnels, à valeur dans un module totalement ordonné [3, 4] ;

– flots canalisés avec les bornes linéaires de type :

$$\min\{b_i\,;\,\alpha'_i\, f_1 - \beta'_i\} \leq f_i \leq \max\{c_i\,;\,\alpha_i\, f_1 + \beta_i\}$$

pour $i = 2, 3, \ldots, M$ [3, 4].

Le problème 1.1 se modélise facilement (*voir* [2]) par un programme linéaire paramétré par α. Ce n'est plus le problème classique de flot maximum, car la variable à maximiser intervient dans les contraintes et, en plus, on perd ainsi l'unimodularité de la matrice d'où l'intérêt de la recherche d'une nouvelle méthode efficace.

Minoux [2] propose une méthode originale et séduisante basée sur la propriété de connexité des réseaux. Nous avons montré dans [1] que ces algorithmes dégénèrent et qu'ils ne convergent pas toujours vers les solutions optimales. Ces dysfonctionnements n'avaient jamais été mis en évidence auparavant car leurs raisons sont profondes.

2. LES DYSFONCTIONNEMENTS DE L'ALGORITHME PROPOSÉ DANS [2]

Voici un exemple où l'algorithme proposé dans [2] ne donne pas une solution optimale au problème de l'α-flot maximal. Appliqué au réseau de la figure 1 pour $\alpha = 3/5$:

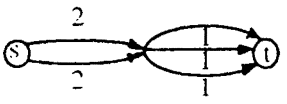

Figure 1. – Le réseau avec des capacités

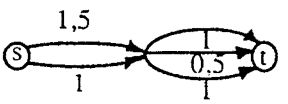

Figure 2.

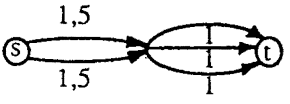

Figure 3. – 3/5-flot maximal

il donne en deux itérations un 3/5-flot « maximum » dont la valeur totale est égale à 2,5 (*fig.* 2), ce qui n'est pas la valeur maximale (*fig.* 3). Nous renvoyons le lecteur intéressé à [1] pour une étude détaillée des dysfonctionnements suivants:

– l'algorithme ne construit pas la courbe qu'il se propose de construire;

– la courbe construite n'est pas toujours concave alors qu'elle devrait l'être;

– sur un même exemple l'algorithme peut construire deux courbes distinctes;

– l'algorithme peut cycler;

– l'algorithme peut donner sur un même exemple des α-flots présumés maximaux ayant des valeurs totales de flot différentes.

3. RÉSOLUTION DES PROBLÈMES 1.1 ET 1.2

Les publications postérieures à [2] préconisent différentes méthodes (y compris fortement polynomiales) destinées *a priori* à résoudre les généralisations des 1.1 et 1.2 [3, 4]. Nous montrons ici que les problèmes initiaux pouvaient être résolus efficacement. Nous proposons un algorithme glouton, fortement polynomial, facile à implémenter et rapide en exécution. Il est basé sur la méthode de Newton légèrement modifiée couplée avec la recherche de flot maximum classique. La preuve de la forte polynomialité utilise les propriétés élémentaires des programmes linéaires paramétrés ainsi que les propriétés élémentaires des graphes.

Les valeurs $f = \max f_1(c)$ des solutions optimales d'un programme linéaire avec le second membre paramétré forment, en fonction du paramètre c, une courbe concave, linéaire par morceaux. Dans le cas particulier du problème du flot maximum avec les capacités paramétrées, les segments linéaires ont des pentes entières, égales au nombre minimal d'arcs avec les capacités réellement bornées par le paramètre, dans une coupe minimale. Ces valeurs commencent par $p_{\max}$ égale au nombre maximum des chemins arcs-disjoints entre s et t dans G pour le flot nul et finissent par zéro au niveau du flot maximum non paramétré (certaines valeurs peuvent être absentes). Résoudre le problème 1.1 équivaut à trouver le flot maximum dans le réseau où les capacités existantes sont limitées par l'abscisse $\underline{c}$ du point d'intersection de la droite $f = 1/\alpha \cdot c$ avec cette courbe. Une solution non triviale existe seulement pour $\alpha \in [1/p_{\max}\,;\, 1[$. Pour l'obtenir rapidement, il suffit d'appliquer la méthode de Newton où les vraies pentes sont remplacées par des nombres naturels consécutifs. En considérant ces pentes comme le compteur (décalé de 1) du nombre de recherches du flot maximum, on voit facilement que l'algorithme ci-dessous s'arrête sur une solution optimale pour une pente $p \leq \lceil 1/\alpha \rceil - 1 \leq p_{\max} - 1$, où $\lceil \cdot \rceil$ désigne la partie entière par excès. Par conséquent sa complexité est $p_{\max} \cdot K$ où K est la complexité de la recherche d'un flot maximum.

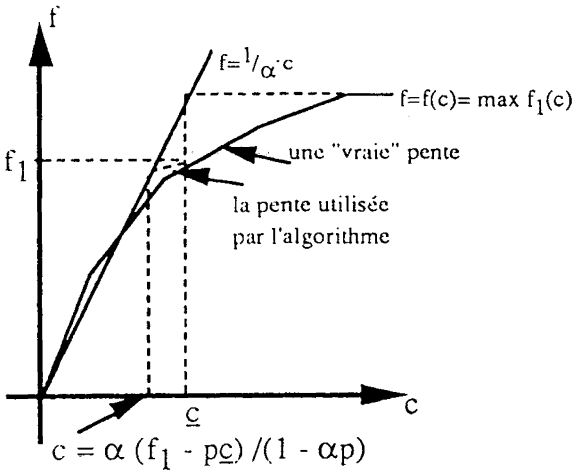

Figure 4. – Illustration du déroulement de l'algorithme proposé.

ALGORITHME

Données : réseau $\{G = (X ; U) ; s ; t)$ avec ses capacités $c_i \geq 0\,(i = 2, 3, \ldots, M)$; $p_{\max}$ (égale au nombre maximum des chemins arcs-disjoints entre s et t dans G) ; un réel $\alpha \in [1/p_{\max} ; 1[$.

Résultat : la solution du problème 1.1 – un α-flot maximum.

$p := 0$ et $c := 0$.

FLOT MAX : Chercher un flot maximum $\mathbf{f} = (f_1, f_2, \ldots, M)$ entre s et t dans G avec c_i.

si $f_i \leq \alpha f_1$ pour $i = 2, 3, \ldots, M$; **alors** $\mathbf{f}$ est un α-flot maximal, aller à **FIN ALGORITHME** **sinon** $c := \alpha(f_1 - pc)/(1 - \alpha p)$; $c_i := \min\{c_i ; c\}$; $p := p + 1$; **retourner à FLOT MAX**

FIN ALGORITHME

On peut très facilement adapter cette méthode pour résoudre le problème 1.2 [1]. En règle générale, cette idée, qui permet d'éviter la recherche des vraies pentes, simplifie beaucoup l'implémentation pratique des algorithmes.

Les méthodes dichotomiques classiques sont aussi très efficaces. Nous signalons dans [1] qu'il existe une condition d'arrêt indépendamment de α, qui garantit la finitude de ces méthodes, néanmoins elles ne sont pas fortement polynomiales.

RÉFÉRENCES

1. W. Bienia, V. Letrouit, Les α-flots maximaux et les flots à taux de sécurité maximal, *Rapport de recherche, Laboratoire Artemis, Imag.,* 1995.
2. M. Minoux, Flots équilibrés et flots avec sécurité, *EDG-Bulletin de la direction des études et recherches,* Série C-Mathématiques, Informatiques, 1976, *1*, p. 5-16.
3. U. Zimmermann, Duality for balanced submolar flows, *Discrete Applied Mathematics,* 1986, *15*, p. 365-376.
4. U. Zimmermann, On the complexity of the dual method for maximum balanced flows, *Discrete Applied Mathematics*, 1994, *50*, p. 77-88.