

# Gestion en temps réel d'un réseau d'assainissement : vérification de l'optimalité et de l'applicabilité de la théorie des graphes par rapport à la programmation linéaire mixte

Real-time management of a sewage system: Verification of the optimality and applicability of graphical linear programming compared to mixed linear programming

J. VAZQUEZ\*<sup>1</sup>, M. FRANÇOIS<sup>2</sup>, D. GILBERT<sup>3</sup>

Reçu le 19 juillet 2001, accepté le 1<sup>er</sup> septembre 2003\*\*.

## SUMMARY

The first stage of real-time management of wastewater systems could, for example, consist of making sure that the use of controls such as valves and pumps can indeed minimise the discharge into the natural environment. This management step, referred to as reference management, is used to determine the control strategies over the entire duration of a rainfall event known in advance. The calculation is therefore performed at the end of the rainfall event and is used to determine what could have been done with the regulation components (e.g. in terms of minimising the volumes discharged). The calculation can also show whether or not it is necessary to control the valves and pumps during the rainfall occurrence (dynamic management) rather than fixing the flow rates in advance (static management) if the receiving body of water is to be protected from discharges.

In the area of operational research, management controls can be determined with the help of linear programming. Here the aim is to minimise the linear function (f), generally called the cost function or objective function, under different linear constraints (g). There are several variants of linear programming. The first one is mixed linear programming, where some variables are required to be integers or even binary. Conventional calculation techniques such as branch-and-bound method provide a global minimum solution, but the calculation is very lengthy.

1 Département Eau et Environnement, École Nationale du Génie de l'Eau et de l'Environnement de Strasbourg, 1, quai Koch, B.P.1039 F, 67070 Strasbourg, France. Tél. : (33) 3 88 24 82 79 - Fax : (33) 3 88 24 82 24.

2 Direction départementale de l'Agriculture de Meaux, France.

3 Direction départementale de l'Agriculture de Strasbourg, France.

\* Correspondance. E-mail : jvazquez@engees.u-strasbg.fr

\*\* Les commentaires seront reçus jusqu'au 30 juin 2004.

Another variant of linear programming is graph programming. This optimisation technique consists of modelling the hydraulic behaviour of most of the constructions that can be found in wastewater systems (main drains, storm overflows, storm water basins, etc.). It has been applied to the wastewater system of Saverne in order to minimise the volumes discharged into the natural environment. In order to ensure that the constructions are modelled correctly and to optimise the functioning of the controls, saturation constraints had to be added to the choice of the arcs of the graph. The primal-dual algorithm no longer provides a global minimum solution when these constraints are added. In contrast, the calculation time is much shorter than that for mixed linear programming.

This article is aimed at comparing the results in terms of the minimisation of the discharged volumes by means of linear programming with graphs and mixed linear programming, as part of the reference management applied to the wastewater system of Saverne. The final goal was to be able to select a compromise between the relevance or the accuracy of the results and the means to achieve them.

We have shown that wastewater constructions such as main drains, storm basins and overflows can be modelled simply with the two techniques above. However, it is necessary to add binary variables in the case of mixed linear programming and a degree of arc saturation if the graph approach is used. The branch-and-bound algorithm used for mixed linear programming can be used to obtain a global minimum solution, with a very long calculation time. In contrast, even though the convergence time is very short for linear programming with graphs, the global minimum cannot be ensured because the algorithm used imposes independence with respect to the choice of the arcs to be saturated.

Given the benefits and drawbacks of each approach, we have attempted to use the example of the wastewater system of Saverne to quantify the calculation time and the differences in terms of the discharged volume. The results have shown that mixed linear programming requires calculation times that can last over 24 h. In contrast, with the graph approach, the calculation takes approximately five minutes on average, with a global minimum in terms of volume that does not exceed 5% as compared to the solution obtained by mixed linear programming. We have shown that a solution requiring a much shorter calculation time is available and offers a compromise between exact determination and an optimised associated calculation time.

**Key-words:** *combined sewer network, real time controls, optimisation, mixed linear programming, graph linear programming.*

## RÉSUMÉ

Dans le cas de la gestion en temps réel des réseaux d'assainissement, la première étape peut, par exemple, consister à vérifier qu'une manipulation des organes de contrôle tels que les vannes et pompes est capable de minimiser les déversements vers le milieu naturel. Cette gestion, que l'on appellera « gestion de référence », permet de déterminer les stratégies de commande sur toute la durée de l'événement pluvieux connu à l'avance. Ce calcul se fait donc à la fin de l'événement pluvieux et permet de dire ce qui aurait pu être fait avec les organes de régulation en terme de minimisation des volumes déversés. La programmation linéaire par les graphes et la programmation linéaire mixte permettent de déterminer une solution optimale. Cet article s'intéresse à la vérification de l'optimalité et à l'applicabilité de la programmation linéaire par les graphes comparée à la programmation linéaire mixte dans le cas de la « gestion de référence » sur le réseau d'assainissement de Saverne (France). En comparant les volumes déversés par ces deux

techniques d'optimisation sur 34 événements pluvieux, nous pouvons confirmer que l'approche par les graphes ne donne pas toujours le minimum global. Les résultats ont montré que la programmation linéaire mixte fournit des temps de calcul qui peuvent atteindre plus de 24 heures. Par contre, l'approche par les graphes permet un temps de calcul de l'ordre de 5 minutes en moyenne avec un minimum global en terme de volume déversé atteint qui n'excède pas 5 % par rapport à la solution fournie par la programmation linéaire mixte.

**Mots clés :** *réseau d'assainissement, contrôle en temps réel, optimisation, programmation linéaire mixte, programmation linéaire par les graphes.*

## 1 – INTRODUCTION

La majorité des systèmes d'assainissement des grandes villes européennes sont du type unitaire, c'est-à-dire conçus pour transporter un mélange des eaux résiduaires et de l'eau de temps de pluie. Ces réseaux sont reliés à des installations de traitement de capacité limitée qui ne peuvent souvent pas faire face aux événements pluvieux. Le fait que les surfaces des villes deviennent de plus en plus imperméables a amené la construction de nombreuses structures et infrastructures pour limiter l'inondation ou le déversement intempestif vers le milieu naturel. La directive européenne n° 91/271 du 21 mai 1991 sur les eaux résiduaires urbaines oblige maintenant les autorités locales à considérer le traitement des eaux polluées transportées par le réseau d'égout par temps pluvieux afin de limiter les débordements de pollution dans le milieu naturel (à l'exclusion des précipitations dites « exceptionnelles »).

Les équipements existants sont souvent trop petits par rapport à cet objectif de traitement et rendent nécessaire la construction de nouvelles infrastructures. Si de nouveaux investissements sont nécessaires, ils doivent être faits dans des conditions qui semblent économiquement acceptables pour les autorités locales. Pour limiter le nombre de nouvelles infrastructures, on cherche à gérer de façon optimale les capacités de traitement de pollution par une meilleure utilisation des infrastructures existantes (collecteur, bassin de rétention, déversoir). Un « pilotage optimal » des ouvrages au cours d'un événement pluvieux suppose la mise en place de tout un système d'acquisition d'informations en temps réel des variables d'état du système (pluie, niveaux, débits, paramètres de pollution...) et des organes de contrôle. Il s'agit alors d'une gestion où les règles de contrôle peuvent être déterminées de façon automatique par un centre de calculs qui peut également procéder à leur application par le biais de télécommandes et d'automates programmables (FREROT, 1987). La gestion en temps réel permet donc non seulement d'optimiser l'efficacité d'un réseau d'assainissement (en termes de volumes et de pollution déversés), mais elle évite aussi de nombreux surcoûts en matière de construction d'ouvrages (RICARD, 1994).

La première étape d'une gestion en temps réel de réseaux d'assainissement peut, par exemple, consister à vérifier qu'une manipulation des organes de contrôle tels que les vannes et pompes est capable de minimiser les

déversements vers le milieu naturel. Cette gestion que l'on appellera « gestion de référence » permet de déterminer les stratégies de commande sur toute la durée de l'événement pluvieux connu à l'avance. Ce calcul se fait donc à la fin de l'événement pluvieux et permet de dire ce qui aurait pu être fait avec les organes de régulation par exemple en terme de minimisation des volumes déversés. Ce calcul peut aussi montrer ou non la nécessité de piloter les vannes et les pompes pendant l'événement pluvieux (gestion dynamique) plutôt que de fixer les débits à l'avance (gestion statique) dans le cas où l'on cherche à protéger le milieu récepteur d'un déversement.

Dans le domaine de la recherche opérationnelle, la détermination des commandes de gestion peut se faire grâce à la programmation linéaire. Il s'agit alors de minimiser une fonction linéaire  $f$ , appelée généralement fonction coût ou fonction objectif, sous différentes contraintes linéaires  $g$ . La programmation linéaire est la première technique d'optimisation à avoir été mise en place par G.B. Dantzig en mettant au point l'algorithme du simplexe. On démontre, dans le cadre général de la programmation linéaire, que l'ensemble des solutions réalisables est un ensemble convexe (DROESBEKE *et al.*, 1986). Si cet ensemble est convexe et borné, la programmation linéaire converge systématiquement vers une solution minimale. Or, les conditions de convergence sont très importantes pour l'exploitant du réseau d'assainissement. En effet, au cours d'un événement pluvieux, la méthode d'optimisation doit être capable de fournir une solution, ce qui n'est pas toujours le cas en programmation non linéaire (FRETROT, 1987). Les modifications dans l'algorithme du simplexe conduisent aujourd'hui à un temps de calcul compatible avec la gestion en temps réel des réseaux d'eau potable (GUHL, 1999) et d'assainissement (NELEN, 1993 ; SCHILLING et PETERSEN, 1987 ; WINDSOR, 1973).

Il existe plusieurs variantes en programmation linéaire. Parmi celles-ci, citons d'abord la programmation linéaire mixte pour laquelle certaines variables sont astreintes à être entières ou même binaires. Cette technique permet de transcrire en programmation linéaire des contraintes non linéaires ou des implications du type « si, alors » (GULH, 1999). Ce n'est que par l'introduction de la méthode du simplexe par Dantzig en 1951, que des algorithmes efficaces ont pu être mis au point pour la résolution de tels problèmes (MINOUX, 1983, DROESBEKE *et al.*, 1986). Quand des variables entières sont utilisées, l'optimisation peut se faire soit par la méthode de coupe de Gomory, soit par un algorithme basé sur la « branch-and-bound method » (KHELIL *et al.*, 1993). Les logiciels résolvant les problèmes linéaires par ces algorithmes sont courants ; un exemple de ceux-ci est CPLEX 6.0 produit par ILOG [ILOG, 1998]. Ces techniques de calcul garantissent une solution minimale globale mais les temps de calcul deviennent très importants.

Une autre variante de la programmation linéaire est la programmation par les graphes (VAZQUEZ, 1997). Cette technique d'optimisation peut modéliser le comportement hydraulique de la plupart des ouvrages rencontrés en réseau d'assainissement (collecteur, déversoir d'orage, bassin d'orage...). Nous l'avons ainsi appliquée au réseau d'assainissement de Saverne dans le but de minimiser les volumes déversés dans le milieu naturel. Afin de garantir une modélisation correcte des ouvrages et dans le but d'optimiser le fonctionnement des organes de commandes, il a fallu ajouter des contraintes de saturation sur le choix des arcs du graphe qui conduisent à des variables binaires que nous détaillerons par la suite. Bien que les conditions de stabilité du

système autour d'un point de fonctionnement et de robustesse en stabilité (maintient de la stabilité en présence d'erreurs) soient vérifiées (VAZQUEZ *et al.*, 1998), l'algorithme utilisé Primal-Dual de Sakarovitch (SAKAROVITCH, 1984) ne garantit plus, dans le cas de l'ajout de ces contraintes, une solution minimale globale. Par contre, le temps de calcul est beaucoup plus rapide que pour celui de la programmation linéaire mixte, mais l'algorithme peut se faire « piéger » par un minimum local.

L'objectif de cet article est, dans un premier temps, de vérifier l'optimalité de la programmation linéaire par les graphes dans laquelle on impose au graphe un ordre de saturation des arcs pour prendre en compte l'hydraulique du réseau que l'on détaillera par la suite. On utilise l'algorithme Primal-Dual de Sakarovitch pour résoudre le problème de flot maximal à coût minimal. On a modifié cet algorithme afin de garantir l'ordre de saturation des arcs. Dans ce contexte, on ne garantit plus un minimum global. Cette vérification se fait par rapport à la programmation linéaire mixte utilisant un algorithme basé sur la « branch-and-bound method » capable de prendre en compte les variables binaires traduisant les contraintes de saturation précédentes. Pour les deux approches, on utilisera la même formulation mathématique. La comparaison se fera dans le cas de la « gestion de référence » sur le réseau d'assainissement de Saverne (France) et porte sur les volumes déversés vers le milieu naturel. Dans un deuxième temps, on compare les temps de calcul de chaque méthode. On pourra ainsi vérifier l'applicabilité des techniques de calcul en vue de la gestion en temps réel d'un réseau d'assainissement. Le but final est de pouvoir choisir un compromis entre la pertinence ou la précision des résultats et le moyen de les obtenir.

Nous allons, dans un premier temps, décrire le principe de modélisation hydraulique des principaux ouvrages (collecteur, bassin d'orage et déversoir d'orage) par la programmation linéaire par les graphes et par la programmation linéaire mixte. Dans un deuxième temps, après une présentation succincte du réseau d'assainissement de Saverne, nous détaillerons les résultats des optimisations.

## 2 – PRINCIPE DE MODÉLISATION

### 2.1 Système d'équations à résoudre

Selon la programmation linéaire par les graphes ou mixte, on peut écrire le système d'équations à résoudre sous la forme :

$$\text{Min} \left( \sum_{i=1}^n \gamma_i \cdot \varphi_i \right) \text{ sous les contraintes : } M \cdot \varphi = B \text{ et } 0 \leq \varphi \leq \text{cm.}$$

$n$  : le nombre d'arcs (Arc : transfert d'un volume entre deux lieux du réseau d'assainissement),  $\gamma_i$  : coût unitaire de transit,  $\varphi_i$  : flot (volume) de transit,  $M$  : matrice d'incidence,  $B$  vecteur dépendant du site d'étude et  $\text{cm}$  capacité de transit maximale.

Le calcul consiste en la minimisation d'une fonction linéaire sous un certain nombre de contraintes linéaires avec variables réelles et binaires. La fonction

linéaire représente le volume déversé minimisé sur l'ensemble de la durée de la pluie. Un coût unitaire par volume n'est affecté qu'aux arcs déversant vers le milieu naturel. Cette valeur est prise constante et identique pour tous les arcs déversants. Les variables réelles sont les volumes d'eau écoulés dans chaque collecteur pendant un pas de temps donné. Les contraintes et les variables binaires représentent les lois hydrauliques du réseaux d'assainissement que nous détaillerons par la suite. On en déduit, par exemple, les temps de démarrage/d'arrêt de pompes ou les temps d'ouverture/fermeture de vannes.

Le pas de temps doit être assez court pour que la gestion soit réalisée en temps réel et soit suffisamment précise pour faire face aux pluies violentes de courte durée. Il doit être aussi assez long pour obtenir toutes les informations des différents capteurs de débit, par exemple (mesure de l'état du réseau), pour calculer les nouvelles consignes et pour les envoyer à chaque ouvrage de contrôle (pompes, vannes, ...). Pour déterminer ce pas de temps, il faut aussi connaître le temps de transfert des débits dans le réseau. Pour ces raisons, le pas de temps choisi sur le réseau de Saverne est de 4 minutes (VAZQUEZ, 1997).

## 2.2 Propriété d'implication

Nous aurons besoin de traduire en programmation linéaire mixte les implications SI ALORS des différents algorithmes « collecteur », « bassin » et « déversoir » que nous verrons par la suite. Il faut écrire sous forme de contraintes linéaires des conditions de saturation du type « si  $f(s) > 0$  alors  $g(s) = g_{\max}$  ». L'idée est ici de trouver des fonctions  $f$  et  $g$  qui respectent les conditions d'écoulement.

Soit l'implication suivante :  $f(s) > 0 \Rightarrow g(s) = g_{\max}$

avec pour contrainte  $0 \leq f(s) \leq f_{\max}$  et  $0 \leq g(s) \leq g_{\max}$

Pour transformer cette implication en contraintes qui pourront alors s'ajouter à celles du programme linéaire, nous introduisons une variable binaire supplémentaire  $x$ . On écrit alors :

$$f(s) \leq f_{\max} (1 - x) \text{ et } g(s) \geq g_{\max} (1 - x).$$

Nous obtenons bien l'implication recherchée puisque si  $f(s) > 0$ ,  $x = 0$  (la valeur 1 étant impossible puisqu'elle donne  $f(s) \leq 0$ ) et on a  $g(s) \geq g_{\max}$  et compte tenu de la contrainte  $g(s) \leq g_{\max}$  c'est-à-dire  $g(s) = g_{\max}$ . Il faut aussi remarquer que, quand  $f(s) \leq 0$ ,  $x$  peut prendre la valeur 0 ou 1. Par contre si  $g(s) < g_{\max}$  alors  $x = 1$  et dans ce cas  $f(s) = 0$ . Nous avons utilisé cette technique pour modéliser l'écoulement à travers les ouvrages.

## 2.3 Modélisation des ouvrages

Dans cette partie, nous allons décrire la modélisation du comportement hydraulique des collecteurs, déversoirs d'orage et bassins d'orage. Nous donnerons également les limites de chaque modèle.

### 2.3.1 Modélisation de l'écoulement dans un collecteur

L'écoulement dans un collecteur est représenté mathématiquement par les équations différentielles non-linéaires de Barré de Saint Venant en unidimen-

sionnel. Compte tenu de la difficulté de résolution de ces équations, nous avons préféré utiliser le modèle de Muskingum (CUNGE, 1969) qui en est une simplification. Certains phénomènes hydrauliques tels que les mises en charge ou les influences aval provoquées par exemple par la présence d'un seuil à l'aval ne seront pas pris en compte. Toutefois lorsque ces phénomènes hydrauliques ne sont pas importants, ils jouent un rôle qui peut être négligeable. On se place donc dans ces conditions hydrauliques.

Le modèle de Muskingum utilisant les équations de stockage et de conservation propose une relation du type :

$$Qs(t + \Delta t) = \beta_1 \cdot Qe(t) + \beta_2 \cdot Qe(t + \Delta t) + \beta_3 \cdot Qs(t)$$

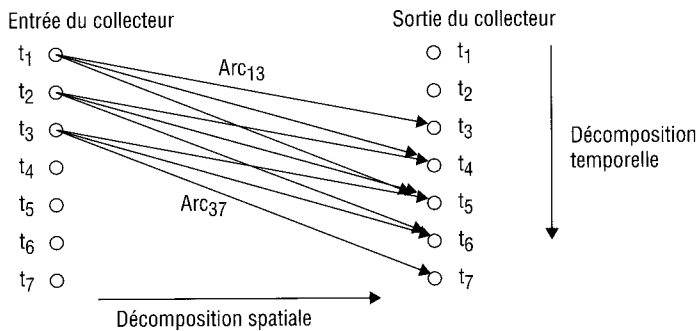
où,  $Qs(t)$ ,  $Qe(t)$  représentent respectivement le débit de sortie et d'entrée à l'instant  $t$ , et  $\beta_1$ ,  $\beta_2$ ,  $\beta_3$  des paramètres définis comme suit :

$$\beta_1 = \frac{2.K.\alpha + \Delta t}{2.K.(1 - \alpha) + \Delta t} ; \beta_2 = \frac{-2.K.\alpha + \Delta t}{2.K.(1 - \alpha) + \Delta t} ; \beta_3 = \frac{2.K.(1 - \alpha) - \Delta t}{2.K.(1 - \alpha) + \Delta t}$$

avec :  $\beta_1 + \beta_2 + \beta_3 = 1$

$K$  et  $\alpha$  sont des paramètres de calage du modèle de Muskingum que nous avons modifié (VASQUEZ, 1999).

Afin de représenter le fonctionnement hydraulique d'un collecteur par la programmation linéaire par les graphes, on fait une décomposition spatiale (entrée/sortie du collecteur) et temporelle (décomposition en pas de temps) comme indiquées sur la figure 1.

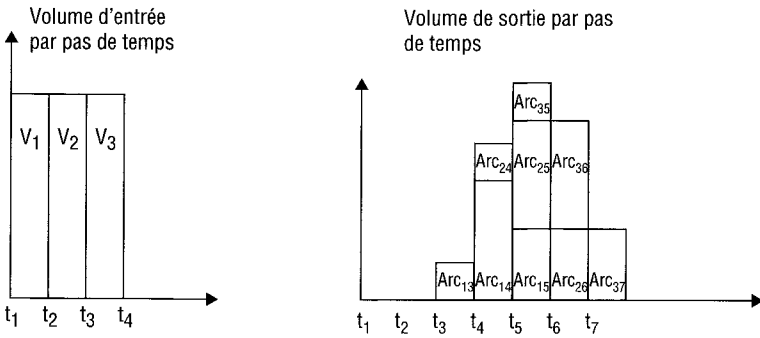


**Figure 1** Représentation par les graphes d'un transfert et d'une diffusion de débit.

*Graph representation of flow transfer and diffusion.*

L'arc  $1 \rightarrow 3$  ( $Arc_{13}$ ) montre qu'un volume arrivant à l'instant  $t_1$  se propage avec un temps de transfert de  $t_1-t_3$ . On voit également sur la figure 1 qu'un volume arrivant à l'instant  $t_1$  va se diffuser à travers les arcs  $Arc_{13}$ ,  $Arc_{14}$  et  $Arc_{15}$ .

La représentation sous forme d'hydrogramme de la figure 1 peut être représentée par la figure 2. Le volume  $V_1$  arrivant à l'instant  $t_1$  va être transféré par les volumes dans les  $Arc_{13}$ ,  $Arc_{14}$  et  $Arc_{15}$ . On fait de même pour les volumes  $V_2$  et  $V_3$ .

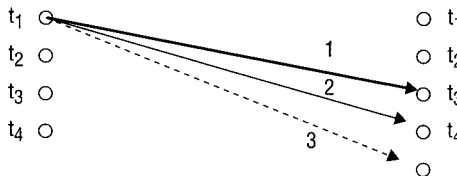


**Figure 2** Décomposition d'un hydrogramme.  
Decomposition of a hydrograph.

Pour déterminer le volume des arcs, on utilise la technique de l'hydrogramme unitaire (SHERMAN, 1932).

Par cette technique de décomposition spatiale et temporelle, on prend en compte la diffusion et le temps de transfert des débits en réseaux d'assainissement.

On constate ainsi que dans un collecteur plus le débit est faible, plus le temps de transfert est long. Afin de prendre en compte ce principe, les arcs les plus éloignés sont d'abord saturés (c'est-à-dire qu'ils atteignent la capacité maximale en terme de volume) dans le sens 3-2-1 de la figure 3.



**Figure 3** Principe de saturation des arcs.  
Principle of arc saturation.

Pour fixer la capacité maximale de chaque arc, on utilise le principe décrit par la figure 2. On simule par le modèle de Muskingum N hydrogrammes rectangulaires. On détermine la capacité maximale (volume d'eau maximal transit) pour chaque arc  $Arc_{i,i+2}$ ,  $Arc_{i,i+3}$  et  $Arc_{i,i+4}$  partant de l'instant  $t_i$  de telle sorte qu'elle soit constante quel que soit l'instant  $t_i$ .

*L'algorithme COLLECTEUR :*

*Formulation pour la théorie des graphes :*

Les arcs doivent être saturés dans l'ordre décroissant des temps de transfert, d'abord  $i = n$  puis  $n-1$  jusqu'à 1. Donc, saturer les  $n$  tronçons dans l'ordre (d'abord  $i = n$ ,  $i = n - 1$ , ...,  $i = 1$ ) revient à dire d'un point de vue mathématique que :



**Si  $\exists i \in \{1, \dots, n\}$  tel qu'un volume transitant dans l'arc<sub>i</sub> > 0 alors le volume transitant dans l'arc<sub>i+1</sub> = capacité maximale de l'arc<sub>i+1</sub>.**

C'est-à-dire :

**Si  $\exists i \in \{1, \dots, n\}$  tel que  $d_i > 0$  alors  $d_{i+1} = cm_{i+1}$ .**

Avec  $d_{i+1}$  : le Volume transitant dans l'arc  $i + 1$ , et  $cm_{i+1}$  : la capacité de transit maximale dans l'arc  $i + 1$ .

*Formulation pour la programmation linéaire mixte :*

$$\forall i, 0 \leq d_i \leq cm_i$$

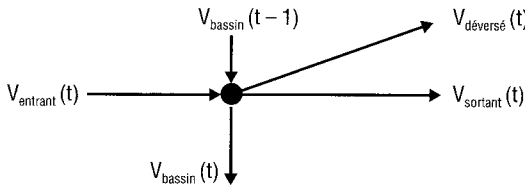
$$d_i \leq cm_i (1 - x_{i+1})$$

$$d_{i+1} \geq cm_{i+1} (1 - x_{i+1})$$

$x$  : Une variable binaire.

### 2.3.2 Modélisation d'un bassin d'orage

Selon la représentation adoptée dans cette étude, la modélisation du volume d'eau dans un bassin d'orage se base sur la conservation de la matière : la variation entre le volume dans le bassin  $V_{bassin}(t-1)$  à l'instant  $t-1$  et  $V_{bassin}(t)$  à l'instant  $t$  est égale à la variation entre le volume entrant dans le bassin  $V_{entrant}(t)$  et la somme des volumes sortant  $V_{sortant}(t)$  et déversés  $V_{déversé}(t)$  comme le montre la figure 4.



**Figure 4** Représentation élémentaire du bassin d'orage.  
*Elementary representation of the storm water basin.*

Ce schéma est reproduit à chaque instant, ce qui permet de fournir la décomposition temporelle du volume d'eau dans le bassin (figure 5).

Il n'existe qu'une seule implication à vérifier sur un bassin d'orage : le bassin ne peut déverser qu'une fois plein.

*L'algorithme BASSIN :*

*Formulation pour la théorie des graphes :*

**Si  $\exists i \in \{bassin\}$  tel qu'un volume transitant dans l'arc<sub>i</sub> de déversement > 0 alors le volume transitant dans l'arc<sub>i+1</sub> = capacité maximale du bassin.**

C'est-à-dire :

**Si  $\exists i \in \{bassin\}$  tel que  $db_i > 0$  alors  $db_i = cmb$ .**

$cmb$  : la capacité maximale des arcs correspond au volume total du bassin,

$db$  : le volume transitant dans l'arc  $i$  du bassin,

$db_i$  : le volume qui est rejeté vers le milieu naturel.

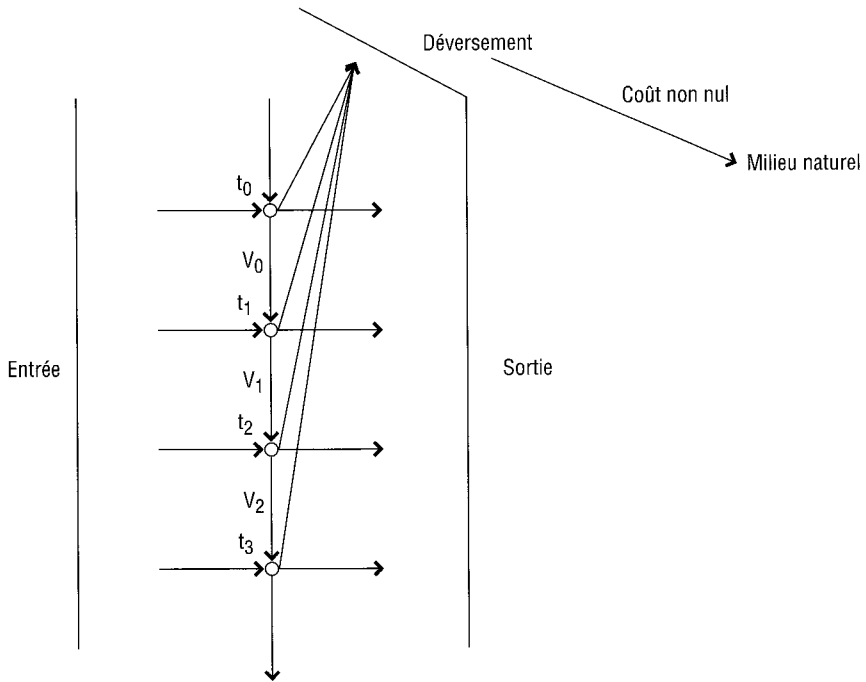
Formulation pour la programmation linéaire mixte :

$$\forall i \in \{\text{bassin}\} \quad 0 \leq db_i \leq cmb ; \quad db_i \geq 0$$

$$db_i \leq CP (1 - x_i)$$

$$db_i \geq cmb (1 - x_i)$$

CP : représente la capacité maximale de déversement pour un pas de temps.

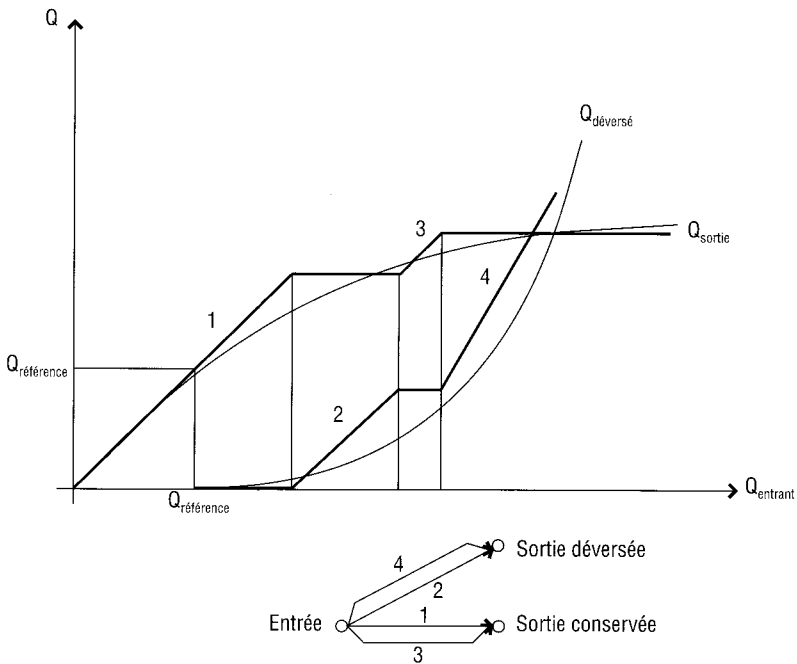


**Figure 5** Représentation du bassin d'orage par les graphes.  
Graphical representation of the storm water basin.

### 2.3.3 Modélisation d'un déversoir d'orage

La courbe de fonctionnement d'un déversoir peut être prise en compte par la programmation linéaire par les graphes en décomposant le processus de la façon suivante : en appelant  $Q_{\text{référence}}$  le débit à partir duquel le déversoir commence à déverser, la figure 6 montre le principe de fonctionnement du déversoir. Les arcs sont saturés dans l'ordre suivant : 1, 2, 3 et 4.

En numérotant les arcs impairs se dirigeant vers la sortie conservée et les arcs pairs vers la sortie déversée, on peut reprendre le même algorithme que celui du collecteur.



**Figure 6** Représentation de la courbe de fonctionnement du déversoir d'orage par les graphes.  
*Graphical representation of the storm overflow characteristics curve.*

*L'algorithme DÉVERSOIR :*

*Formulation pour la théorie des graphes :*

Les arcs doivent être saturés dans l'ordre croissant de la numérotation des arcs du déversoir. Donc, saturer les  $n$  tronçons dans l'ordre (d'abord  $i = 1$ ,  $i = 2, \dots, i = n$ ) revient à dire d'un point de vue mathématique que :

**Si  $\exists i \in \{1, \dots, n\}$  tel qu'un volume transitant dans l'arc $_i > 0$  alors le volume transitant dans l'arc $_{i-1} =$  capacité maximale de l'arc $_{i-1}$  du déversoir.**

C'est-à-dire :

**Si  $\exists i \in \{1, \dots, n\}$  tel que  $d_i > 0$  alors  $d_{i-1} = \text{cmdv}_{i-1}$ .**

$\text{Cmdv}_{i-1}$  : la capacité maximale de l'arc  $i-1$  correspond au volume de déversement.

*Formulation pour la programmation linéaire mixte :*

$$\forall i, 0 \leq d_i \leq \text{cmdv}_i$$

$$d_i \leq \text{cmdv}_i (1 - x_{i-1})$$

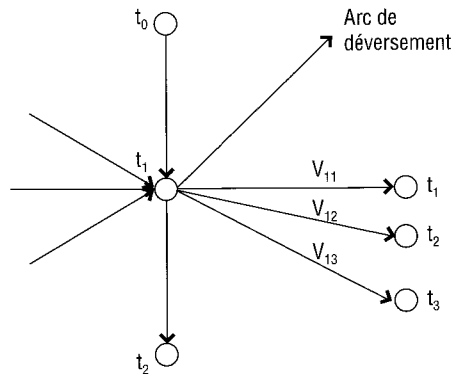
$$d_{i-1} \geq \text{cmdv}_{i-1} (1 - x_{i-1})$$

### 2.3.4 Contraintes de continuité

À chaque confluence et défluence du réseau d'assainissement, le respect des contraintes de continuité hydraulique est assuré par l'écriture des lois aux nœuds.

### 2.3.5 Détermination des lois de commande

Nous avons vu que l'ensemble du réseau est décomposé suivant un pas de temps fixé au départ. La détermination de la loi de commande en un point du réseau à un instant donné va consister à faire la somme des volumes des arcs sortant de ce point à l'instant désiré. Dans l'exemple suivant, on représente à la figure 7 le graphe d'un bassin à l'instant  $t_1$ .



**Figure 7** Représentation d'un pas de temps d'un bassin d'orage par les graphes.  
*Graphical representation of a time-step of the storm water basin.*

Afin de connaître le débit de vidange à imposer à l'ouvrage de contrôle disponible dans le bassin, il suffit de faire la somme des volumes  $V_{11}$ ,  $V_{12}$  et  $V_{13}$ , et de diviser ce volume par le pas de temps.

## 2.4 Algorithmes de résolution utilisés

### 2.4.1 Programmation linéaire mixte

L'ensemble des équations précédemment définies pour chaque ouvrage du réseau d'assainissement de Saverne a été transcrit dans le logiciel CPLEX 6.0 produit par ILOG (ILOG, 1998). On dispose de 94500 contraintes, 26500 variables réelles et 19000 variables binaires. Nous avons utilisé l'algorithme « Branch and Bound » (séparation – évaluation). L'intérêt de celui-ci est de s'affranchir d'un examen de toutes les solutions (qui représenteraient  $2^n$  possibilités où  $n$  est le nombre de variables binaires) tout en s'assurant d'obtenir la solution optimale (MINOUX, 1983). Afin de fournir une solution plus rapide sans pour autant aller jusqu'à la convergence exacte, le logiciel CPLEX encadre la fonction objectif par des valeurs minimale et maximale pouvant être atteintes. La valeur minimale que l'on appellera « PLM min », consiste à fixer une partie des variables binaires comme étant des variables réelles comprises entre 0 et 1, alors que la valeur maximale que l'on appellera « PLM max », garantit les

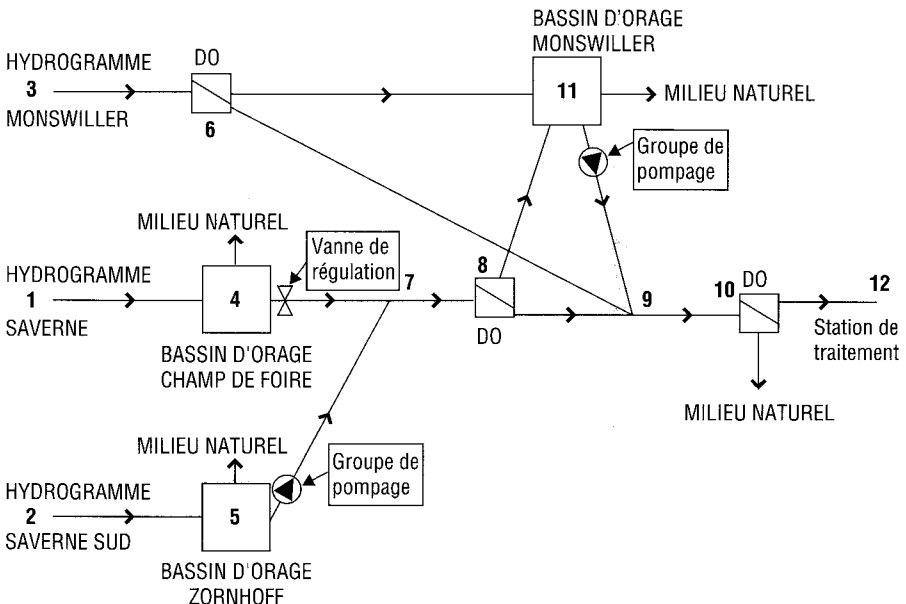
variables binaires comme étant égales à 0 ou 1 mais la solution trouvée est un minimum local. Cette démarche permet de stopper le calcul alors que le minimum global n'est pas atteint. En effet, dans certaines situations, nous avons atteint 24 heures de temps de calcul. L'encadrement du minimum global est donc assuré dans le cas où les temps de calcul deviennent très importants.

#### 2.4.2 Programmation linéaire par les graphes

Nous avons développé notre propre logiciel de génération du graphe ainsi que de calcul du flot maximal à coût minimal (VAZQUEZ, 1997). Le graphe du réseau de Saverne représente 27000 arcs. Nous avons utilisé l'algorithme primal-dual de Sakarovitch (SAKAROVITCH, 1984) pour résoudre le problème d'optimisation. Nous appellerons « PLG » la valeur de la fonction objectif après convergence. Les modèles que nous avons développés précédemment imposent un ordre de saturation des arcs du graphe. Or, une des hypothèses à respecter pour garantir un minimum global en utilisant l'algorithme primal-dual est l'indépendance du choix des arcs. On ne garantit donc plus un minimum global. Les temps de calcul sont de l'ordre de 5 minutes.

La partie suivante s'intéresse à la comparaison de la valeur « PLG » avec les valeurs de « PLM min » et « PLM max » pour les 34 pluies définies précédemment.

### 3 – APPLICATION AU RÉSEAU DE SAVERNE



**Figure 8** Schéma du réseau de Saverne.  
*Diagram of the Saverne sewer system.*

### 3.1 Description du réseau et choix des événements

Le Syndicat d'Assainissement de la Région de Saverne-Zorn-Mossel regroupe 10 communes situées à une distance de 35 km au Nord-Ouest de Strasbourg et compte environ 16 800 habitants. Le réseau est principalement unitaire. Il comporte 92 déversoirs d'orage, 12 bassins d'orage et 3 stations de refoulement. Après simplification du réseau intercommunal, la figure 8 présente le réseau d'assainissement étudié. Le bassin d'orage 4 (champ de foire) est régulé par l'intermédiaire d'une vanne motorisée alors que les bassins 11 (Monswiller) et 5 (Zornhoff) disposent chacun d'un groupe de pompage. L'objectif de la gestion en temps réel étant la minimisation des volumes déversés vers le milieu naturel, nous imposerons donc un coût sur les volumes déversés. Le pas de temps choisi sur Saverne est de 4 minutes.

Les événements pluvieux proviennent d'une base de données regroupant environ 2000 hyétogrammes. Après simulation du réseau, on a constaté que les pluies ayant une lame d'eau totale inférieure à 2 mm provoquent un volume total inférieur à 4 % du volume du bassin. Elles ne sont pas prises en compte. Il reste cependant 535 événements pluvieux.

Plusieurs paramètres ont été choisis pour extraire de cette base de données les événements pour l'optimisation de référence :

- Événements provoquant un déversement dans le cas d'une gestion statique,
- Durée de l'événement,
- Apparition du débit de pointe en début, milieu et fin d'événement pluvieux,
- Nombre de pics en débit par événement pluvieux,
- Décalage temporel d'apparition des pointes de débit entre les 3 bassins versants.

Compte tenu de ces critères, nous avons retenu 34 événements pluvieux.

### 3.2 Résultats

La comparaison entre la fonction objectif fournie par la programmation linéaire par les graphes « PLG » après convergence et le maximum de la fonction objectif fourni par la programmation linéaire mixte, « PLM max », nous permet de confirmer que l'approche par les graphes ne donne pas toujours le minimum global. Les résultats du tableau 1 montrent que pour 6 pluies où la programmation linéaire mixte donne une meilleure solution, la différence des déversements peut atteindre 40 m<sup>3</sup>. Mais ceci ne représente que 0,30 % du volume total de la pluie, 0,96 % du volume total des 3 bassins et 6,28 % du volume déversé.

Pluie	Pluie totale (m <sup>3</sup> )	Différence des déversements (m <sup>3</sup> ) Volume <sub>PLG</sub> - Volume <sub>PLM max</sub>	Différence des déversements par rapport au volume de la pluie	Différence des déversements par rapport au volume des bassins	Différence des déversements par rapport au volume total déversé
1-50	13 031,1	40,4	0,30 %	0,96 %	6,28 %
1-63	10 206,0	0,5	< 0,01 %	0,01 %	< 0,01 %
10-63	13 295,5	35,3	0,13 %	0,84 %	0,24 %
10-85	16 589,6	5,7	0,02 %	0,14 %	0,03 %
14-39	14 368,1	34,3	0,23 %	0,82 %	5,26 %
15-25	9 099,3	13,1	0,12 %	0,31 %	0,79 %

**Tableau 1** Écarts absolus et relatifs entre les déversements pour chaque bassin par événement pluvieux.

**Table 1** Absolute and relative differences between the overflows per rain event.

La comparaison entre la fonction objectif fournie par la programmation linéaire par les graphes « PLG » et la valeur « PLM min » nous apparaît intéressante car elle nous montre de combien au maximum pourrait être amélioré le minimum trouvé par les graphes. Ces différences sont inférieures à 5 %. On constate que pour 6 pluies sur 34, CPLEX n'a pas convergé par manque de capacité de la mémoire de l'ordinateur. En effet, le calcul de « PLM min », consistant à fixer une partie des variables binaires comme étant des variables réelles comprise entre 0 et 1, nécessite la création d'un arbre mémorisant l'ensemble des résultats. Quand Cplex trouve une solution entière, celle-ci devient une valeur de coupe et l'arbre est diminué fortement. Par contre, quand une solution entière n'est pas atteinte, l'ensemble des résultats est conservé, c'est ainsi que dans certains cas on sature la mémoire du PC.

Nous nous intéressons maintenant à la répartition des déversements selon les trois bassins. Pour le bassin 4, les écarts ( $\text{Volume}_{\text{graphe}} - \text{Volume}_{\text{linéaire mixte}}$ ) sont positifs ou nuls alors que pour les 2 autres bassins, ils sont négatifs. Nous avons donc calculé le taux de déversement par bassin et pour chacune des pluies (tableau 2), c'est-à-dire pour la pluie p et le bassin n :

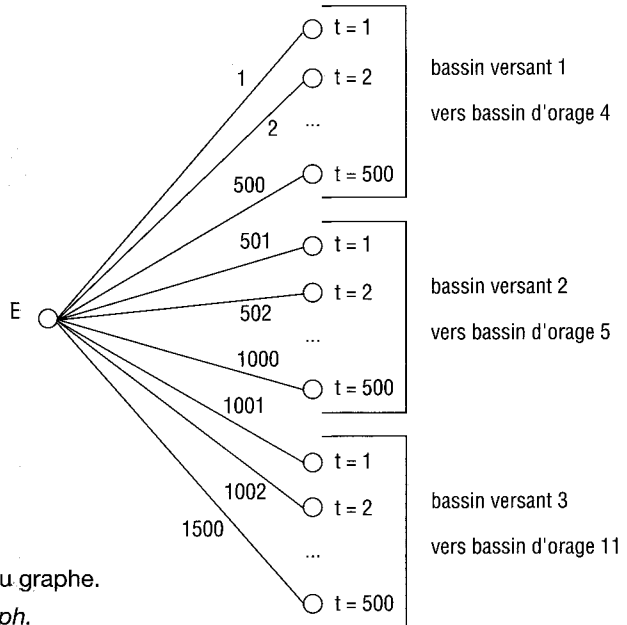
$$\frac{\text{Volume déversé au bassin n pour la pluie p}}{\sum \text{déversements pour la pluie p}}$$

pluie	% de déversement par la programmation linéaire mixte			% de déversement par la programmation linéaire par les graphes		
	Bassin 4	Bassin 5	Bassin 11	Bassin 4	Bassin 5	Bassin 11
1-50	100 %			100 %		
1-63	85 %	7 %	8 %	88 %	5 %	7 %
10-63	91 %	0 %	9 %	92 %		8 %
10-85	84 %	9 %	7 %	84 %	9 %	7 %
14-39	100 %			100 %		
15-25	100 %			100 %		

**Tableau 2** Pourcentages de déversements par bassin et par événement pluvieux.

**Table 2** Percentage overflow per basin and per rain event.

On constate, comme le montre le tableau 2, que les déversements sont légèrement privilégiés à travers le bassin 4 pour la programmation linéaire par les graphes par rapport aux résultats obtenus par la programmation linéaire mixte. Ce constat s'est également révélé pour l'ensemble des autres pluies. Le fait que le bassin 4 soit toujours plus rempli et qu'il déverse toujours plus dans le cas de l'approche par les graphes peut venir de la numérotation. En effet, intéressons-nous au marquage des sommets et à la numérotation des arcs du graphe représentée à la figure 9.



**Figure 9**

Numérotation des arcs du graphe.

*Arc numbering in the graph.*

Les 1500 premiers arcs ont pour origine le sommet 1 qui est l'entrée du système. Les 500 premiers arcs numérotés 1-500 sont les arcs d'entrée du bassin versant 1 (bassin d'orage 4), les arcs numérotés 501-1000 sont les arcs d'entrée du bassin versant 2 (bassin d'orage 5) et finalement les arcs numérotés 1001-1501 sont les arcs d'entrée du bassin versant 3 (bassin d'orage 11).

L'approche par les graphes cherche d'abord à saturer le maximum d'arcs sans déversement. En effet, nous n'avons affecté qu'un coût aux arcs déversant vers le milieu naturel. Donc, l'algorithme va d'abord faire transiter l'eau vers la station de traitement (voir figure 8). Etant donnée la numérotation, elle traitera en premier l'eau du bassin versant 4.

De même, les arcs de stockage du bassin 4 portent les numéros pairs de 23444 à 24440, ceux du bassin 5 les numéros pairs de 24444 à 25440 et ceux du bassin 11 les numéros pairs de 25444 à 26440. Une fois que, pour chaque bassin versant, le flot maximal à coût nul est calculé, c'est-à-dire qu'on ne peut pas envoyer plus d'eau à la station de traitement alors le flot supplémentaire est stocké. Les premiers arcs trouvés par l'algorithme sont fonction de la numérotation et correspondent au bassin 4. Il en est exactement de même pour les déversements. Ceci permet d'expliquer les constatations faites sur les déversements et les volumes stockés.



## 4 – CONCLUSIONS

La « gestion de référence » d'un réseau d'assainissement permet de renseigner l'exploitant sur l'opportunité de gérer un réseau d'assainissement en temps réel dans l'objectif, par exemple, de minimiser les volumes déversés. Parmi les techniques de recherche opérationnelle, la programmation linéaire par les graphes et la programmation linéaire mixte permettent de fournir les stratégies de commande des organes de contrôle (pompes, vannes motorisées). Nous avons montré qu'il est possible de modéliser de façon simplifiée par les deux techniques précédentes les ouvrages disponibles en assainissement tels que les collecteurs, bassins et déversoirs d'orage. Toutefois, il est nécessaire d'introduire des variables binaires dans le cas de la programmation linéaire mixte et un ordre de saturation des arcs par l'approche par les graphes. L'algorithme utilisé par la programmation linéaire mixte « Branch and Bound » peut fournir une solution minimale globale moyennant un temps de calcul qui peut être très important. Par contre, bien que pour la programmation linéaire par les graphes le temps de convergence soit très rapide, on ne garantit plus un minimum global. En effet, l'algorithme utilisé impose une indépendance sur le choix des arcs à saturer, ce qui n'est pas vérifié.

Compte tenu des avantages et inconvénients de chaque approche, nous avons voulu quantifier sur l'exemple du réseau d'assainissement de Saverne le temps de calcul et les différences en terme de volume déversé. Les résultats ont montré que la programmation linéaire mixte fournit des temps de calcul qui peuvent atteindre plus de 24 heures. Par contre, l'approche par les graphes permet un temps de calcul de l'ordre de 5 minutes en moyenne avec un minimum global en terme de volume déversé atteint qui n'excède pas 5 % par rapport à la solution fournie par la programmation linéaire mixte.

On vient ainsi de montrer qu'on peut choisir un compromis entre la détermination exacte d'une optimisation et le temps de calcul qui lui est associé avec une solution approchée acceptable présentant un temps de calcul beaucoup plus faible.

## ABRÉVIATIONS

A	: section mouillée ;	h	: tirant d'eau ;
B	: vecteur dépendant du site d'étude ;	M	: matrice d'incidence ;
cm	: capacité de transit maximale ;	n	: le nombre d'arcs (transfert d'un volume entre deux lieux du réseau d'assainissement) ;
cmb	: la capacité maximale des arcs correspond au volume total du bassin ;	Q	: débit ;
cmdv	: la capacité maximale des arcs correspond au volume de déversement ;	$S_f$	: perte de charge ;
db	: le volume qui est rejeté vers le milieu naturel ;	$S_o$	: pente du canal ;
$d_i$	: le volume transitant dans l'arc $i$ ;	u	: vitesse ;
		x	: une variable binaire ;
		$\gamma_i$	: coût unitaire de transit ;
		$\phi_i$	: flot (volume) de transit.

## RÉFÉRENCES BIBLIOGRAPHIQUES

- CUNGE J.A., 1969. On the subject of a flood propagation computation method (Muskingum method). *J. Hydr. Res., International Association for Hydraulic research*, vol. 7 n° 2, pp. 205-230.
- DROESBEKE F., HALLIN M., LEFEVRE Cl., 1986. Programmation linéaire par l'exemple, Éditions Ellipses.
- FREROT A., 1987. Procédures d'optimisation des consignes de gestion d'un réseau d'assainissement automatisé. Th. Doct. École Nationale des Ponts et Chaussées, 296 p.
- GUHL F., 1999. Gestion optimale des réseaux d'eau potable. Th. Doct. Univ. Louis Pasteur de Strasbourg, 148 p.
- ILOG, 1998, CPLEX 6.0 Technical Manual, ILOG Software.
- KHELIL A., KNEMEYER B., DEHNHARDT J., 1993. Comparison of optimisation algorithms to determine control strategies in UDS, 6<sup>th</sup> *International Conference on Urban Storm Drainage* (Proceedings pp 1395-1400).
- MINOUX M., 1983. Programmation mathématique : théorie et algorithmes, Tomes 1 et 2 Éditions Dunod, Paris.
- NELEN F., 1993. On the use of optimization techniques for urban drainage operation, 6<sup>th</sup> *International Conference on Urban Storm Drainage* (Proceedings pp 1387-1394).
- RICARD, 1994. Comment éviter de polluer et d'inonder à tort ? Apports de l'expérimentation numérique et intégration dans un processus d'apprentissage sur un site réel. Th. Doct. Univ. Lyon, 320 p.
- SAKAROVITCH M., 1984. Optimisation combinatoire, Méthode mathématiques et algorithmes : Programmation discrète. Éditions Hermann, Paris.
- SCHERMAN L.K., 1932. Stream flow from rainfall by the unit graph method, *Engineering new records*, vol. 108, pp. 501-505.
- SCHILLING W., PETERSEN S.O., 1987. Real time operation of urban drainage systems - validity and sensitivity of optimization techniques, *Proceedings of the symposium : Systems analysis in water quality management*, University of London.
- VAZQUEZ J., 1997. Gestion en temps réel d'un réseau d'assainissement : minimisation des rejets urbains par temps de pluie, Th. Doct. Univ. Louis Pasteur de Strasbourg, 203 p.
- VAZQUEZ J., GILBERT D., BELLEFLEUR D., GRANDJEAN B., 1998. Sensibilité de la gestion en temps réel des réseaux d'assainissement utilisant la théorie des graphes. *Conférence Novatech 98*, 4, 5 et 6 mai 1998 à Lyon.
- VAZQUEZ J., ZUG M., BELLEFLEUR D., GRANDJEAN B., SCRIVENER O., 1999. Utilisation d'un réseau de neurones pour appliquer le modèle de Muskingum aux réseaux d'assainissement, *Revue des Sciences de l'Eau*, vol. 12, n° 3, pp. 577-595.
- WINDSOR J. S., 1973. Optimization model for the operation of flood control systems, *Water Resources Research*, vol. 9, october 1973, n° 5, pp. 1219-1226.