

# Un algorithme efficace pour une généralisation du *pooling problem*

Paulin Jacquot<sup>1</sup>, Maxime Fender<sup>2</sup>

<sup>1</sup> EDF R&D - CMAP, École Polytechnique, Inria Saclay, France  
{paulin.jacquot}@polytechnique.edu

<sup>2</sup> Artelys, France  
{maxime.fender}@artelys.com

**Mots-clés** : *pooling, network design, traitement des eaux, MINLP*

## 1 Introduction

Nous nous intéressons ici à une version très générale du *pooling problem* qui a été proposée lors du challenge Mixed Integer Non Linear Optimization (programme MINO, cf [5]). Ce problème consiste à la gestion d'un réseau de traitement de l'eau, où l'eau provient d'un ensemble de différentes sources (*WS*) "contaminées" (chaque contaminant  $c \in C$  est présent avec une concentration  $\rho_c$ ), passe à travers le réseau et est acheminée aux unités de demandes (*WD*).

L'intérieur du réseau est constitué de différentes unités : des unités de traitement (*TU*) qui permettent d'abaisser la concentration des contaminants de l'eau les traversant, et des unités d'opération (*PU*) qui ont besoin d'eau pour fonctionner (usines, centrales...) et augmentent les concentrations des contaminants de l'eau les traversant et enfin des unités d'attente (*BU*) dans lesquelles une quantité d'eau peut être stockée pendant une période de temps. Des tuyaux de différents diamètres relient les différentes sorties d'unités aux différentes entrées (les cycles sont autorisés).

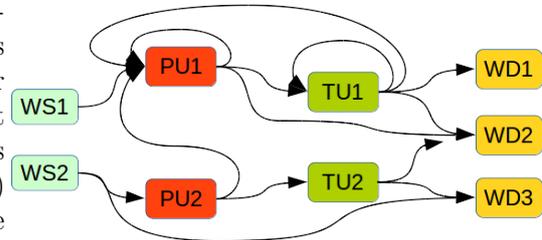


FIG. 1 – Exemple de réseau de traitement

Nous souhaitons résoudre à la fois les problèmes couplés du design du réseau et de son fonctionnement optimal sur plusieurs périodes  $t \in T$ . Nous devons ainsi décider de l'existence des tuyaux et de leurs diamètres, entre chacune des unités, mais également de l'installation d'unités de traitements (*TU*) et d'attente (*BU*) plus ou moins onéreuses et efficaces. Les unités d'opération, les sources et les demandes ainsi que leurs caractéristiques sont, elles, fixées.

Le fonctionnement optimal du réseau est déjà un problème NP difficile : nous devons trouver les valeurs des flux pour chaque période  $t \in T$ , qui satisfassent à la fois les contraintes de flux, les contraintes non-linéaires de conservation et les seuils maximaux des contaminants.

## 2 Formulation et contraintes

Le problème général peut se formuler comme un programme d'optimisation en variables mixtes, non linéaire et non convexe, c'est pourquoi il est même difficile d'en trouver une solution admissible.

Les différentes variables sont :

- $\eta_{s,m,d} \in \{0, 1\}$  pour l'existence d'un tuyau de diamètre  $d$  entre les unités  $s$  et  $m$ ,
- $\eta_{tu} \in \{0, 1\}$  pour l'existence de l'unité de traitement  $tu$ ,
- $\varphi_{s,m}(t)$  pour le flux d'eau dans le tuyau allant de l'unité  $s$  à l'unité  $m$ ,
- $\rho_{s,m,c}(t)$  pour la concentration du contaminant  $c \in C$  dans le tuyau  $(s, m)$ .

Les contraintes sont de différentes formes : les bornes min et max sur les flux d'entrée et les concentrations d'entrée et de sorties, les équations régissant l'impact des unités d'opération et les unités de traitement (non linéaires), et enfin les équations de conservation apportant la non-convexité :

$$\sum_s \varphi_{s,m} = \varphi_m^{in} \qquad \sum_s \varphi_{s,m} \rho_{s,m,c} = \varphi_m^{in} \rho_{m,c}^{in}$$

### 3 Méthode de résolution et résultats

L'algorithme développé repose sur une résolution séquentielle de problèmes non linéaires continus (NLP) et linéaires mixtes (MILP). Une relaxation linéaire du problème est obtenue en utilisant principalement des inégalités de type McCormick (voir [4] ) et le problème est résolu en utilisant le solveur *Xpress*. Nous obtenons ainsi un optimum global de la relaxation, qui fournit un point de départ pour lancer la résolution locale du NLP. Pour cela, nous utilisons l'algorithme *active set* (SLEQP) fournit par le solveur Knitro (voir [1],[2],[3] ). En affinant à chaque pas notre relaxation MILP en ajoutant des coupes, la résolution locale du NLP finit par converger vers un point réalisable. Les points réalisables trouvés fournissent des bornes supérieures au problème.

Notre algorithme a remporté le challenge MINO 2016 (*cf* [5]) et a résolu toutes les instances réalisables en un temps très court par rapport aux méthodes proposées par les autres concurrents. La plus importante des instances proposées était de taille considérable avec  $|C| = 11$  contaminants,  $|TU| = 16$  unités de traitements potentielles, 41 noeuds, 1254 tuyaux potentiels et  $|T| = 4$  périodes de temps, résultant en un grand nombre de variables pour le problème initial.

### Références

- [1] Richard H Byrd, Nicholas IM Gould, Jorge Nocedal, and Richard A Waltz. An algorithm for nonlinear optimization using linear programming and equality constrained subproblems. *Mathematical Programming*, 100(1) :27–48, 2003.
- [2] Richard H Byrd, Nicholas IM Gould, Jorge Nocedal, and Richard A Waltz. On the convergence of successive linear-quadratic programming algorithms. *SIAM Journal on Optimization*, 16(2) :471–489, 2005.
- [3] Richard H Byrd, Jorge Nocedal, and Richard A Waltz. Knitro : An integrated package for nonlinear optimization. In *Large-scale nonlinear optimization*, pages 35–59. Springer, 2006.
- [4] Garth P McCormick. Computability of global solutions to factorable nonconvex programs : Part i convex underestimating problems. *Mathematical programming*, 10(1) :147–175, 1976.
- [5] MINO. Initial training network, mixed integer non linear optimization, 2016.