EasyChair Preprint

№ 2558

Modeler and solver integration inside GRTgaz optimization tools

Nicolas Derhy

February 5, 2020

Intégration des modeleurs et solveurs dans les outils d'aide à la décision de GRTgaz

Nicolas Derhy

RICE, GRTgaz, Villeneuve la Garenne
nicolas.derhy@grtgaz.com

Mots-clés *: transport de gaz, modeleur, solveur, outils d'aide à la décision, intégration.*

GRTgaz et RICE

GRTgaz construit, entretient et développe le réseau de transport de gaz naturel sur la majeure partie du territoire Français. Il contribue à l'équilibre du système gazier, dont dépend l'alimentation des consommateurs de gaz naturel :

- Les sites industriels directement raccordés au réseau de transport
- Les particuliers, collectivités et entreprises desservis par les réseaux de distribution publique, eux-mêmes alimentés par le réseau de transport

Avec plus de 32 000 km de canalisations enterrées, GRTgaz transporte le gaz de ses clients dans des conditions optimales de sécurité, de coût et de fiabilité tout en préservant l'activité des territoires, les paysages et la biodiversité.

RICE (Research and Innovation Center for Energy) accompagne ainsi les entités opérationnelles du groupe notamment en leur fournissant des outils d'aide à la décision pour leur permettre d'accomplir au mieux leurs missions. La plupart de ces outils reposent aujourd'hui sur l'utilisation de modeleurs et de solveurs dont l'intégration est devenue un élément déterminant des développements informatiques.

Un historique ancien et de nombreux changements

Le développement d'outils d'aide à la décision pour appuyer les entités opérationnelles est ancien et existait déjà du temps de GazDeFrance. Au milieu des années 2000, deux solveurs, Cplex [2] et Clp [1], sont intégrés lors du développement de nouveaux logiciels. A cette époque, ils sont utilisés via leur API C++ et permettent la résolution de programmes linéaires de grande taille.

Cependant, l'utilisation du C++ et des API bas niveau des solveurs rendent la maintenance et les évolutions de plus en plus complexes. Les codes informatiques sont de très grande taille et souvent peu accessibles : ils requièrent un coût d'entrée important notamment lors de l'arrivée d'un nouveau développeur. Dans l'objectif de rationaliser les développements sur le long terme, un premier virage majeur est entrepris en 2009/2010 avec l'adoption de modeleurs permettant de séparer le modèle mathématique du code informatique. Le modèle devient un fichier texte facilement exploitable en dehors de l'application.

L'utilisation des modeleurs se généralisent et un grand nombre d'entre eux (Opl [2], Mosel [6], Glpk [3], Gams [5], …) sont utilisés en fonction des besoins, des spécificités informatiques, des préférences des développeurs ou du coût des licences. L'éparpillement trop important de ces modeleurs, au fonctionnement parfois très différent, aboutit à une réflexion pour se recentrer sur un nombre limité de modeleurs. La décision est prise en 2013 de concentrer au maximum les travaux autour de Glpk et d'Ampl [4].

En parallèle du choix de ces modeleurs, l'utilisation des API des modeleurs et des solveurs est petit à petit abandonnée. Pour permettre une intégration plus simple aux outils désormais développés en C#, l'utilisation directe des exécutables des modeleurs et solveurs est préconisée. La manière de développer s'uniformise permettant notamment de passer facilement de Glpk à Ampl. Enfin, des connecteurs entre Glpk et des solveurs comme Cbc et Cplex sont développés : ils permettent de rendre Glpk multi-solveurs de manière transparente pour les développeurs.

A la recherche du compromis idéal

Les différentes évolutions au cours des dix dernières années ont été motivées par trois objectifs principaux :

- La flexibilité : pouvoir basculer d'un modeleur à un autre ou changer de solveur avec un faible impact sur le code informatique est primordial. Il permet d'être plus flexible et évite une dépendance trop forte avec un modeleur/solveur donné.
- La performance : il est indispensable de pouvoir fournir aux entités opérationnelles des outils d'aide à la décision fiables, performants et répondant à leurs besoins. Passer à la dernière version d'un solveur plus rapide ou switcher sur un modèle non linéaire en cas de nouveau besoin doit être possible sans nécessiter de tout redévelopper.
- Le coût maitrisé : les coûts de licences des modeleurs et des solveurs peuvent être élevés et un frein à certains développements. Il est donc intéressant d'être en mesure de développer des logiciels ne nécessitant aucun achat de licences.

Glpk, combiné notamment à Cbc, permet ainsi de fournir des outils d'aide à la décision performants sans nécessité de payer des licences additionnelles. Si les performances sont insuffisantes, il est possible de basculer sur Cplex en ne modifiant que quelques lignes de code.

En plus d'être « parent » de Glpk, Ampl permet de couvrir un plus large spectre de solveurs. Il offre également la possibilité de traiter des problèmes non linéaires que l'on rencontre fréquemment dans le monde gazier.

Ces choix répondent en grande partie aux exigences et aux besoins décrits ci-dessus. On pourra regretter l'absence d'un modeleur gratuit prenant en charge des modèles non linéaires.

Enfin, si l'abandon de l'utilisation des API permet une meilleure maintenabilité et une intégration plus aisée, certaines possibilités de customisation avancée des algorithmes sont perdues.

Références

[1] Coin-OR. https://www.coin-or.org/

[2] Cplex, Opl. IBM-Ilog. https://www.ibm.com/analytics/cplex-optimizer

[3] Glpk. Andrew Makhorin. https://www.gnu.org/software/glpk/

[4] Ampl. https://ampl.com/

[5] Gams. https://www.gams.com/

[6] Xpress-Mosel. Fico. https://www.fico.com/en/latest-thinking/brochures/mosel-an-overview