



HAL
open science

” Solution Engineering ” : génération et classification de solutions enrichies par des vecteurs de qualité

Thierry Petit, Andrew C Trapp

► To cite this version:

Thierry Petit, Andrew C Trapp. ” Solution Engineering ” : génération et classification de solutions enrichies par des vecteurs de qualité. ROADEF 2018, 19e congrès annuel de la société Française de Recherche Opérationnelle et d’Aide à la Décision, Feb 2018, Lorient, France. hal-01686802

HAL Id: hal-01686802

<https://hal.science/hal-01686802v1>

Submitted on 17 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

“Solution Engineering” : génération et classification de solutions enrichies par des vecteurs de qualité

Thierry Petit¹, Andrew C. Trapp²

¹ IMT Atlantique (DAPI), TASC, LS2N, CNRS, Nantes, France,

Thierry.Petit@imt-atlantique.fr

² Worcester Polytechnic Institute, USA,

atrapp@wpi.edu

Mots-clés : *optimisation, solutions multiples et diverses, programmation par contraintes, programmation mathématique.*

1 Introduction

Les outils permettant de générer des solutions multiples aux problèmes d’optimisation sont en pratique au mieux limités par leur aptitude à incorporer à la fois une notion de diversité entre les solutions et des contraintes métier spécifiques. Ces contraintes sont généralement imparfaitement définies, car trop difficiles à formaliser par un utilisateur avant qu’il n’ait accès à un ensemble de solutions qu’il peut évaluer. Pour pallier ce problème, nous proposons un paradigme prenant un modèle d’optimisation en donnée, permettant de générer un ensemble de solutions proches de l’optimal, diverses et classifiées à l’aide de contraintes fonctionnelles supplémentaires. Chaque solution est enrichie par un vecteur de valeurs correspondant à des métriques de qualité, qui expriment des notions génériques, e.g., une répartition homogène de coûts, une notion de persistance, une limite sur les variation d’utilisation d’une ressource, etc.

2 Principe, implémentation et résultats

Le principe de cette approche est de raffiner les critères d’exploration de l’espace des solutions : deux solutions distinctes selon des notions de qualité de nature sémantique peuvent tout à fait avoir de nombreuses valeurs en commun, et ainsi ne pas être identifiées comme distinctes par les techniques de génération de solutions multiples classiques. La Figure 1 illustre ce cas.

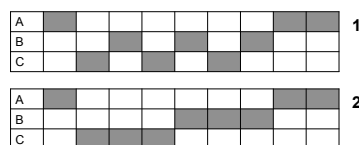


FIG. 1 – Deux solutions à un problème d’affectation. A, B et C sont des machines et les cellules grises sont des tâches. Bien que les deux solutions aient 7 variables sur 9 prenant la même valeur, elles diffèrent selon des critères tels que l’alternance des machines utilisées, le nombre de fois où l’on utilise une machine, le temps d’utilisation d’une même machine en continu. Ces critères sémantiques correspondent pour la plupart à des contraintes globales connues. Ils peuvent être retenus pour exprimer la diversité entre deux solutions, au même titre que les distances mathématiques mesurant si chaque variable est affectée avec une valeur (significativement) distincte dans les deux solutions.

Nous définissons un paradigme retournant un ensemble de solutions enrichies qui, selon le problème considéré, peut être très vaste. Un utilisateur peut alors poser des requêtes à l’aide

d'outils de traitement de données (e.g., des requêtes SQL) pour obtenir la solution qui pourra être appliquée en pratique. Un point crucial est que ces requêtes n'ont pas à être intégrées directement dans le modèle sous la forme de contraintes ou de critères d'optimisation.

Après avoir généré une première solution S_0 à un problème d'optimisation, idéalement optimale, l'idée consiste à infuser dans le problème, successivement, chaque notion de qualité "générique" retenue via une contrainte fonctionnelle; si l'utilisateur souhaite définir une métrique agrégeant plusieurs contraintes de qualité, il peut le faire sans restriction. À chaque étape, l'objectif initial du problème est remplacé par un nouvel objectif incluant un critère de diversité classique Δ entre les solutions générées (e.g., distance de Hamming), une distance d à la valeur d'objectif de la solution S_0 et une variable q exprimant la notion de qualité. Par exemple, pour un problème de minimisation, le nouvel objectif pourra être la maximisation d'un ratio pondéré par trois constantes a , b et c : $r = \frac{a \times \Delta + b \times q}{c \times d}$.

Soit **OnlineGen**(\mathcal{L}) l'algorithme générant à chaque étape un ensemble de solutions diverses pour une notion de qualité donnée, où (\mathcal{L}) est une liste de paramètres, contenant entre autres S_0 et la contrainte globale exprimant cette notion. L'algorithme 1 ci-dessous génère un ensemble classifié des solutions enrichies par des vecteurs de valeurs de qualité.

input : A model $M = (X, \mathcal{C}, f)$ for problem p , a set of quality notions \mathcal{Q} , and an integer $k > 0$.
output: A set \mathcal{S} of solution sets and a $|\mathcal{Q}| \times (k \times |\mathcal{Q}|)$ matrix of score vectors, i.e., one set of $|\mathcal{Q}|$ -length score vectors per quality notion.

```

1  $\mathcal{S} := \emptyset$ ;
2 for  $\ell \in \{1, 2, \dots, |\mathcal{Q}|\}$  do
3   | Generate the parameter list  $\mathcal{L}$  according to  $Q_\ell$ ;
4   |  $\mathcal{S}_\ell = \mathbf{OnlineGen}(\mathcal{L})$ ;  $\mathcal{S} := \mathcal{S} \cup \{\mathcal{S}_\ell\}$ ;
5  $\Psi = \emptyset$ ;
6 for  $\ell \in \{1, 2, \dots, |\mathcal{Q}|\}$  do
7   |  $\Psi_\ell := |\mathcal{S}_\ell| \times |\mathcal{Q}|$  integer matrix;
8   | for  $S_j \in \mathcal{S}_\ell$  do
9     | for  $m \in \{1, 2, \dots, |\mathcal{Q}|\}$  do
10    | |  $\Psi_\ell[j][m] := \text{Score of } q_m \text{ when evaluating } Q_m \text{ on solution } S_j$ ;
11   |  $\Psi := \Psi \cup \{\Psi_\ell\}$ ;
12 Return ( $\mathcal{S}, \Psi$ );

```

Algorithm 1: Quality Enhanced Solution Generation.

Nous avons implémenté ce paradigme en programmation par contraintes (PPC) et en programmation mathématique (MP). Outre l'algorithme **OnlineGen**(\mathcal{L}), cette implémentation requiert plusieurs contributions scientifiques : en PPC, un propagateur GAC pour une contrainte de diversité générique (inspiré d'approches existantes, e.g., [2]) et un principe d'adaptation automatique de la stratégie de recherche du modèle fourni en entrée; en MP, des décompositions linéaires des contraintes globales employées pour exprimer des notions de qualité. L'analyse statistique de nos résultats expérimentaux [3] sur des instances du problème GAP [1] confirme sans ambiguïté la possibilité concrète de générer des ensembles constitués de solutions que les approches de l'état de l'art ne peuvent pas identifier et/ou classier.

Références

- [1] P. C. Chu and J. E. Beasley. A genetic algorithm for the generalised assignment problem. *Computers & Operations Research*, 24(1) :17–23, 1997.
- [2] E. Hebrard, B. Hnich, B. O'Sullivan, and T. Walsh. Finding diverse and similar solutions in constraint programming. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference*, pages 372–377. American Association for Artificial Intelligence, 2005.
- [3] Thierry Petit and Andrew C. Trapp. Enriching solutions to combinatorial problems via solution engineering. *Optimization Online, Tech. report, WPI*, Combinatorial optimization (6252), 2017.