MFRDC

TR 02-071

ABITA+

Manuel théorique

Michel Ferry
Juillet 2002

MFRDC
6 rue de la Perche, 44700 Orvault, France
http://www.mfrdc.com

Table des matières

Manuel théorique

Introduction

Après avoir évalué les informations disponibles, les contraintes d'exploitation supportables et les besoins à satisfaire aux premières heures d'un projet architectural, il est très vite apparut que le problème posé par l'optimisation économique du découpage spatial était d'une complexité fatale si un certaines d'hypothèses simplificatrices n'étaient retenues. Les problèmes voisins traités par l'industrie tel que la répartition de ressources, le partitionnement de maillage, la répartition de charge des réseaux ou le positionnement de composants électroniques sont tous NP-Complet, c'est à dire qu'ils n'ont vraisemblablement pas de solution accessible en un temps polynomial vis-à-vis de la taille des données. Pourtant tous se posent en termes plus simples que ceux de l'optimisation spatiale envisagée ici dans la mesure ou les variables sont bien définies.
Aussi notre première analyse aura-t-elle consisté à poser le problème en termes simples et suffisamment souples pour assurer un usage confortable du logiciel final tout en lui conservant un intérêt pratique. Pour ce faire, l'hypothèse la moins contraignante et la plus versatile nous est apparue être la définition d'un découpage préliminaire de l'espace en éléments. Les éléments, définis par l'utilisateur, maillent la totalité du domaine et constituent les entités discrètes de l'optimisation combinatoire menée pour résoudre le problème.

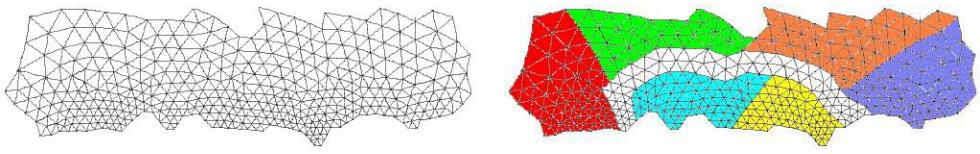

figure 1 : maillage d'un terrain et optimisation en six lots

Certains de ces éléments sont définis comme possiblement associable à des passages ou voie de circulation, d'autres sont imposés comme des entrées ou

sorties du domaine. Enfin chacun peut être affublé d'une plus-value témoignant par exemple de la vue, de l'ensoleillement, de l'étage, etc.
Ainsi posé, le problème d'optimisation revient à trier la totalité des éléments en un certain nombre de lots de manière à optimiser la valeur économique de l'ensemble tout en respectant des contraintes d'ordre géométrique (assurer un accès à chaque lot, imposer la connexité des lots, respecter les aires minimum et maximum des lots) et cardinal (nombre de lots par tranche d'aire).

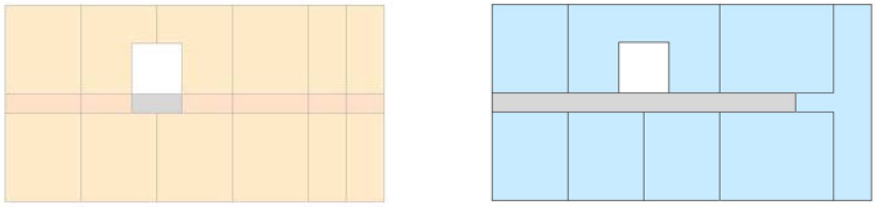

figure 2 : maillage d'un étage d'immeuble et optimisation en huit lots

La suite de ce document est consacrée à l'exposé succinct des méthodes numériques complètes et incomplètes explorées lors de l'élaboration de notre algorithme puis, dans une seconde partie, à la description de la méthode originale crée pour l'occasion.

Méthodes de résolution complètes

On appelle méthode complète ou exacte une méthode garantissant l'obtention d'une solution optimale. Nous nous limiterons dans le cadre de cette étude aux méthodes complètes les plus largement utilisées : les méthodes basées sur une recherche arborescente. Ces méthodes procèdent par une énumération systématique et exhaustive de toutes les affectations possibles des variables du problème d'optimisation combinatoire valuée avec contraintes (Valued Contraint Satisfaction Problem).

Schéma général

La racine, point de départ de la recherche, correspond à l'affectation vide (aucune variable n'est affectée). Les feuilles de l'arbre de recherche entièrement développé correspondent aux affectations complètes des variables du problème Un noeud de profondeur *i* correspond à une affectation des *i* premières variables et les fils de ce noeud aux différentes manières d'étendre cette affectation en affectant une prochaine variable x_i avec toutes les valeurs de son domaine d_i. Notons que le parcours de l'arbre de recherche s'effectue généralement en profondeur d'abord et que l'ordre d'affectation des variables et des valeurs peut être aussi bien statique que dynamique. La taille de ces arbres de recherche est

potentiellement énorme puisque pour une instance de n variables possédant chacune un domaine de *d* valeurs, le nombre d'affectations complètes est d^n.

L'efficacité de ces méthodes repose donc essentiellement sur leur pouvoir de coupe, c'est-à-dire leur capacité à élaguer au maximum l'espace de recherche. L'objectif est de pouvoir détecter, le plus tôt possible, qu'une affectation partielle ne peut pas être étendue en une affectation complète optimale. Pour cela, on se dote :

- d'une borne supérieure *ub* de l'optimum que nous appellerons par abus de langage « optimum courant ». En pratique, *ub* est réactualisée à chaque fois que l'on trouve une meilleure affectation complète. A la fin de la recherche, *ub* correspond à la meilleure évaluation trouvée, c'est-à-dire, l'évaluation optimale du problème ;

- d'une fonction de calcul de borne inférieure *lb* des affectations partielles. Cette fonction calcule une borne inférieure des évaluations de toutes les extensions complètes d'une affectation partielle Ap.

Ainsi, dès que *lb(Ap)<ub*, l'affectation partielle courante *Ap* peut être abandonnée puisqu'elle ne peut pas mener à une affectation complète d'évaluation inférieure à l'optimum courant (monotonie de l'opérateur). On passe alors à la prochaine valeur de la variable courante s'il en reste, dans le cas contraire, on remonte à la variable précédente et on réitère le processus (on parle alors de retour arrière ou Backtrack). C'est le mécanisme de séparation et évaluation (Branch and Bound) bien connu dans le domaine de la recherche opérationnelle.

Notons que la fonction de calcul de bornes inférieures des affectations partielles est probablement le point clé de l'efficacité de ces méthodes. Cette fonction se doit pour être efficace d'être une borne inférieure la plus élevée possible tout en étant rapidement calculable. Les principaux algorithmes complets de résolution des problèmes d'optimisation combinatoire valuée avec contraintes ne se différencient que par cette fonction.

Backward Checking

Il s'agit de l'algorithme le plus simple dans lequel la fonction de calcul de borne inférieure est triviale et consiste simplement à ne prendre en compte que les contraintes complètement affectées[1]. Cette borne inférieure est bien sûr assez

[1] *On dit qu'une contrainte est complètement affectée lorsque toutes les variables qu'elle relie le sont.*

grossière et par conséquent peu efficace. Le Backward Checking constitue l'algorithme de base et n'est utilisé que pour les cas d'école.

Forward Checking

Cette fois, la qualité de la borne inférieure est bien meilleure puisque le calcul prend en compte les contraintes affectées ainsi que les contraintes quasi-affectées[2]. Bien que le calcul de cette borne inférieure soit plus coûteux que celle du Backward Checking, le gain sur la qualité de la borne inférieure permet de réduire de plusieurs ordres de grandeur le nombre de nœud développés ainsi que le temps de calcul.
Le Forward Checking constitue ainsi un bon compromis entre temps de calcul et qualité de borne inférieure. Associé à de bonnes heuristiques sur l'ordonnancement des variables et des valeurs, il est probablement l'algorithme le plus efficace dans le cas général et certainement le plus utilisé.

Méthodes de résolution incomplètes

On appelle méthode incomplète, inexacte ou approchée toute autre méthode. Les méthodes incomplètes sont des méthodes non exhaustives qui explorent de manière opportuniste l'espace des affectations complètes. On distingue grossièrement trois grands types de méthodes incomplètes.

Affaiblissement des méthodes complètes

La méthode souche est une méthode complète que l'on affaiblit en effectuant des « coupes sauvages » dans l'espace de recherche. On décide alors de ne pas explorer certains sous espaces jugés a priori trop peu intéressants. On peut ainsi espérer accélérer notablement la recherche au prix de la perte de la preuve d'optimalité de la solution trouvée. La méthode la plus simple basée sur ce schéma est la méthode Greedy Search ; partant d'une affectation vide, elle affecte chaque nouvelle variable par la valeur de son domaine qui minimise la borne inférieure *lb* jusqu'à l'obtention d'une affectation complète. Il n'y a donc plus de retours arrière puisque cette méthode ne revient jamais sur les choix effectués.

[2] *On dit qu'une contrainte est quasi-affectée si toutes les variables qu'elle relie, sauf une et une seule, le sont.*

Méthodes de recherche locales

Les méthodes de recherche locales (local search) sont probablement les méthodes incomplètes les plus largement utilisées. Leur principe est conceptuellement des plus simples : partir d'une solution potentielle et essayer de l'améliorer en effectuant des changements locaux de manière itérative. On entend par changement local le fait de modifier l'affectation courante en modifiant l'affectation d'une ou plusieurs variables (généralement, à chaque itération, on change de manière stochastique la valeur d'une seule variable). Lorsque la solution courante ne peut plus être améliorée (on parle d'optimisation local), la recherche est alors stoppée et on recommence éventuellement avec une autre configuration initiale (on parle alors d'essais). Naturellement, dans le cas général, un optimum local à peu de chances d'être un optimum global.

L'efficacité de ces méthodes repose sur leur caractère opportuniste qui leur permet, généralement, de pouvoir donner très rapidement une « bonne solution », sans qu'il soit néanmoins possible d'avoir une quelconque idée de la distance à l'optimum de cette solution. Notons que contrairement aux méthodes de recherches arborescentes, ces méthodes manipulent uniquement des affectations complètes.

On peut caractériser une méthode de recherche locale par :

- le choix des affectations initiales pour chaque essai;
- une fonction de voisinage vk, qui associe à toute affectation A l'ensemble des affectations vk(A) qui ne se différencient de A que par le changement de valeur de k variables (appelé « flip »). Généralement, le voisinage utilisé est un voisinage avec k=1, mais certaines applications peuvent nécessiter des voisinages étendus;
- une fonction de choix dans le voisinage f. Cette fonction appliquée au voisinage de l'affectation courante A, retourne la prochaine affectation A'=f(vk(A)) à tester;
- un critère d'acceptation de la prochaine affectation. Généralement, le critère porte directement sur la différence d'évaluation entre l'affectation courante et l'affectation testée. Si l'affectation à tester est acceptée elle remplacera l'affectation courante;
- un critère d'arrêt permettant d'abandonner l'essai en cours ; par exemple, lorsque l'on ne peut plus améliorer la solution courante;
- un critère d'arrêt permettant d'abandonner la recherche ; typiquement, un nombre d'essais maximum ou une limite dans le temps.

Généralement, le voisinage choisi est un voisinage où k=1 et l'on effectue plusieurs essais afin de permettre un recouvrement plus important de l'espace

de recherche. Aussi, les principales méthodes de recherche locale ne se différencient que par leur fonction de choix dans le voisinage et leurs critères d'acceptation.

Descente en gradient

Il s'agit de la méthode de base dans laquelle le critère d'acceptation est tel que l'on ne permet pas de dégrader la qualité de la solution courante. On parle alors de descente en gradient (Hill-Climbing ou Descent Search). Elle est généralement utilisée avec une fonction de choix dans le voisinage qui effectue un choix aléatoire uniforme.

Descente en gradient gloutonne

Dans la descente en gradient gloutonne (Greedy Decsent Search), le critère d'acceptation est toujours vérifié et la fonction de choix dans le voisinage retourne une affectation de l'évaluation minimale. Cette méthode garantit une descente rapide et une exploration exhaustive du voisinage au prix d'une complexité et d'un risque de cycles importants.

Descente en gradient avec minimisation des conflits

Cette méthode, appelée Min-Conflict Hill-Climbing est une descente en gradient dans laquelle le critère d'acceptation est toujours vérifié et la fonction de choix dans le voisinage est la suivante : choisir aléatoirement une variable parmi les variables en conflit[3] et l'affecter d'une valeur de son domaine qui minimise l'évaluation de l'affectation courante. Cette méthode semble être un bon compromis entre l'avidité coûteuse du Gredy Search et de l'opportunisme d'un Hill-Climbing avec un choix aléatoire uniforme dans le voisinage.

Recherche Tabou

Pour pallier au problème de cycle de la méthode gloutonne, une idée originale consiste à mémoriser les dernières affectations visitées[4] et à essayer de limiter (via le critère d'acceptation) le nombre de fois où l'on revient sur une affectation déjà visitée. On parle alors de Taboo Search.

Ajout de bruit

Dans le souci constant d'échapper aux optima locaux, une idée consiste à injecter une part de bruit (Random-Walk) durant une descente en gradient.

---

[3] Variables sur les quelles porte une contrainte non satisfaite.

[4] Une variante de plus en plus utilisée consiste à mémoriser la liste des mouvements effectués.

L'idée consiste à coupler une part de mouvement aléatoire à une des méthodes précédentes :

- avec un probabilité *p* on effectue un mouvement aléatoire (typiquement on effectue un flip quelconque) ;
- avec une probabilité *1-p* on suit la méthode générale.

L'ajout de bruit aux méthodes traditionnelles permet parfois d'augmenter très nettement la qualité des solutions obtenues. On peut cependant regretter le fait que la probabilité *p* de bruit soit difficile à ajuster pour chaque instance.

Méthode de Monte Carlo

Cette méthode est un raffinement de la méthode précédente :

- le choix de la prochaine affectation *A'* dans le voisinage de l'affectation courante *A* est un choix aléatoire uniforme ;
- le critère d'acceptation dépend de la différence d'évaluation δ :
  - si la différence est positive alors *A'* devient la nouvelle affectation courante ;
  - sinon, *A'* ne devient la nouvelle affectation courante qu'avec une probabilité *p* qui est une fonction décroissante de δ.

Ainsi, plus un mouvement dégrade la qualité de la solution courante, moins il a de chances d'être accepté.

Recuit simulé

L'algorithme de recuit simulé (Simulated Annealing) est un algorithme basé sur une analogie entre les phénomènes naturels étudiés par la physique statistique et les problèmes d'optimisation combinatoire. Grossièrement, il s'agit de la méthode de Monte Carlo dans laquelle la probabilité[5] d'acceptation *p* est exponentielle $p=e^{-\delta/T}$. Ainsi, *p* ne dépend plus seulement de la différence d'évaluation mais aussi d'un autre paramètre *T*, appelé température, que l'on fera progressivement décroître. Initialement, la température *T* est suffisamment élevée pour autoriser tous les mouvements dans le voisinage. Progressivement, la température est abaissée pour finalement interdire tout mouvement dégradant la qualité de la solution courante. Notons que cette méthode a l'avantage de disposer de certaines bases théoriques.

[5] *Cette probabilité est issue de la loi de distribution de Gibbs-Boltzmann.*

Algorithmes génétiques

Les algorithmes génétiques sont des algorithmes d'optimisation qui empruntent leurs mécanismes à la théorie de la sélection naturelle Darwinienne. A la différence des recherches locales classiques, ces algorithmes manipulent non plus une seule mais une population d'affectations (individus). L'évolution de la population d'affectation est réalisée à l'aide de trois opérateurs :

- un opérateur de sélection : chaque affectation de la population courante participe à la génération suivante avec une probabilité proportionnelle à son adaptation (inverse de l'évaluation) ;
- un opérateur de mutation : cette opération consiste à modifier aléatoirement un ou plusieurs individus de la population ;
- un opérateur de croisement : permet de créer deux nouveaux individus à partir de deux individus préalablement sélectionnés.

Notons que les algorithmes génétiques sont en fait une sorte de recherche locale manipulant un ensemble d'affectations.
Les algorithmes génétiques connaissent un succès grandissant dans la communauté scientifique, mais n'ont pas encore vraiment fait leur preuves dans le cadre général des problèmes d'optimisation combinatoire valuée avec contraintes, et encore moins dans celui, plus particulier, des problèmes de regroupement (Grouping Algorithms). En effet, si le sens des opérateurs de sélection et de mutation paraît clair (focalisation et diversification de la recherche), celui de l'opérateur de croisement l'est beaucoup moins. Or l'efficacité de ces algorithmes repose essentiellement sur cet opérateur qui, pour être opérationnel, doit être proprement adapté au problème.

Autres méthodes incomplètes

Les méthodes incomplètes présentées ci-dessus travaillent sur des affectations complètes (pour les recherches locales) ou partielles (pour les affaiblissements de méthodes complètes). Il existe une troisième catégorie de méthodes incomplètes ne travaillant pas directement sur des affectations mais sur l'estimation (généralement des probabilités) des valeurs des paramètres du problème. On peut notamment citer la méthode du recuit en champ moyen qui recherche, pour chaque valeur de chaque variable, la probabilité que la variable soit affectée de cette valeur dans une configuration optimale.

La méthode retenue

La grande particularité de l'optimisation spatiale en lots telle qu'envisagée ici repose sur le caractère global de la validité des affectations possibles. En effet, en plus du fait que chaque lot doive rester continu, il convient également de lui conserver un accès aux zones de circulation qui elle-même doive rester connectée à une entrée-sortie. Or cette contrainte forte de circulation s'est avérée rendre l'ensemble des méthodes classiques exposées ci-dessus soit très inefficaces soit extrêmement difficiles à régler. Aussi avons-nous dû, à partir de notre propre expérience et d'algorithmes éprouvés, créer une méthode originale hybride se décomposant en deux phases :

- Génération progressive d'une population de solutions viables
- Optimisation exhaustive locale de voisinage

Génération de solutions viables

A l'instar des algorithmes génétiques notre méthode travaille sur une population de solutions possibles. Ceci présente deux avantages : le premier est de proposer en final plusieurs solutions classées à l'utilisateur qui pourra choisir celle la mieux adapter à ses propres critères ; le second est de permettre un meilleur recouvrement de l'espace des solutions et d'éviter les minima locaux à l'image des méthodes de Monte Carlo ou de recuit simulé. Toutefois à la différence des algorithmes génétiques les solutions ne sont pas croisées entre elle mais générées à partir d'un ensemencement aléatoire. En effet, le croisement de deux solutions viables, par exemple par échange de lots, ne donne qu'exceptionnellement une solution viable. Il nous est apparu plus important de construire un l'algorithme permettant la construction indéfectible de solutions viables. Cet algorithme ce décompose ainsi :

- Ensemencement aléatoire de *n* éléments non susceptibles d'être des éléments de circulation, mais en contact direct avec au moins un de ces éléments.
- Croissance par diffusion des *n* lots ainsi définis jusqu'à affectation complète des éléments.

L'étape de croissance par diffusion consiste à tenter d'agrandir à tour de rôle chaque lot d'un élément contigu non encore affecté. Lorsque l'élément libre est un élément de circulation, celui-ci n'est affecté au lot que dans la mesure ou cette action permet d'une part de conserver la continuité des espaces de circulations vers une entrée-sortie, et d'autre part, ne prive pas un autre lot de son dernier accès à un espace de circulation.
Cette technique permet de construire au moins une solution viable par ensemencement et plus encore lorsqu'un bruit aléatoire est introduit au niveau

de la sélection des éléments libres voisins et de l'ordre d'inspection des lots. De cette manière plusieurs milliers de tirages permettent de construire une population conséquente de solutions viables. Lorsque l'on souhaite travailler sur une population de taille raisonnable les tirages sont triés par ordre décroissant de valeur et seulement les meilleures solutions sont conservées.

Optimisation locale de voisinage

Lorsque l'on possède une population de solutions viables suffisamment riche, une seconde étape consiste à optimiser celle-ci en inspectant systématiquement son voisinage. Pour ce faire, chaque élément frontière de chaque lot est à tour de rôle cédé au lot voisin. Si la solution ainsi créée est viable, elle est évaluée et introduite dans la population lorsque sa valeur dépasse celle de la plus mauvaise solution. Pour conserver une taille constante à la population la plus mauvaise solution est alors retirée.