Questions théoriques liées à l'algorithme du simplexe :

Analyse de complexité la règle de pivot "greatest improvement" appliquée à des cuboïdes

Victor Kani[1], Quentin Louveaux[2]

[1] MPRO, Paris, France
victor.kani@eleves.enpc.fr
[2] Université de Liège, Liège, France
q.louveaux@uliege.be

Mots-clés : *recherche opérationnelle, optimisation, algorithme du simplexe, règle de pivot, greatest improvement, cuboïdes, cubes combinatoires, complexité.*

L'algorithme du simplexe est un algorithme central en optimisation continue et combinatoire. Il a été inventé par George Dantzig en 1947 pour permettre de résoudre une classe de problèmes appelés programmes linéaires et est encore intensément utilisé aujourd'hui. Pourtant, une question non élucidée à ce jour est : l'algorithme du simplexe est-il de complexité polynomiale en le nombre de variables et de contraintes, et éventuellement en la valeur numérique des données ? Nous savons que les programmes linéaires peuvent être résolus polynomialement par des algorithmes tels que les points intérieurs ou les ellipsoïdes, mais ces méthodes ont une approche continue du problème. L'algorithme du simplexe lui a une approche beaucoup plus combinatoire. En effet les contraintes linéaires du problème forment un polyèdre convexe et le simplexe travaille sur le graphe défini par les sommets de ce polyèdre, se déplaçant de voisin en voisin sur ce graphe jusqu'à atteindre une solution optimale. Plus précisément, il est paramétré par une règle dite de pivot qui définit comment nous choisissons à un sommet courant le voisin sur lequel se déplacer. Généralement la règle de pivot n'accepte que les voisins améliorants. La vraie question est alors : existe-t-il une version polynomiale de l'algorithme du simplexe ?

Nous connaissons cependant certaines familles de polyèdres ou de programmes linéaires pour lesquels l'algorithme du simplexe est polynomial avec une certaine règle de pivot, par exemple les problèmes de flot dans un réseau. Nous nous proposons alors d'étudier une famille de polytopes particulière, les cubes combinatoires que nous nommerons cuboïdes, pour lesquels nous avons l'espoir de trouver une version du simplexe polynomiale. En effet ces polytopes vérifient une propriété intéressante : Lorsqu'on oriente les arêtes d'un cuboïde selon une fonction objectif linéaire, alors il existe toujours un chemin orienté de longueur au plus n entre tout sommet du cuboïde et le sommet optimal. Cette propriété en fait donc de très bons candidats, car nous avons la garantie qu'un chemin de longueur linéaire en le nombre de variables existe et peut être potentiellement atteint par l'algorithme du simplexe.

Nous souhaitons étudier la règle "greatest improvement" qui sélectionne toujours le sommet voisin le plus améliorant. Comme la plupart des règles de pivot proposées pour le simplexe, elle a été prouvée non polynomiale dans le cas général mais nous souhaitons savoir si elle est polynomiale ou non sur les cuboïdes. Pour répondre à cette question nous aurons une approche à la fois théorique et pratique. Nous présenterons alors les considérations théoriques que nous avons trouvées et démontrées et nous présenterons aussi les outils que nous avons développés pour mener des expériences et développer des conjectures. Ces outils utilisent astucieusement l'optimisation pour répondre efficacement à nos besoins.

Notre étude se fait en trois parties principales avec trois résultats majeurs. La première partie porte sur la règle "greatest improvement" et son comportement. Nous la définissons et nous définissons des chemins compatibles avec la règle dans un polytope. Nous pouvons ensuite donner une propriété de ces chemins dans le graphe du polytope puis plus généralement dans le polytope lui-même avec une caractérisation géométrique de ces chemins. Cette caractérisation nous permet alors d'identifier facilement, en résolvant un programme linéaire, si un chemin est compatible avec la règle de pivot étudiée.
Elle nous a également permis de développer notre résultat principal pour cette partie : un algorithme efficace de recherche de plus longs chemins compatibles avec la règle "greatest improvement" dans un polytope donné. L'algorithme fait usage d'optimisation linéaire pour identifier la compatibilité des chemins trouvés mais il fait surtout usage de réoptimisation durant la recherche en se servant de la base optimale calculée pour un chemin à une itération donnée pour évaluer plus rapidement les nouveaux chemins à tester aux itérations suivantes.

La seconde partie porte sur les cuboïdes. Nous les définissons et nous donnons une caractérisation algébrique de ceux-ci qui peut se découper en deux conditions principales à vérifier pour qu'un polytope soit un cuboïde.
Nous nous attaquons ensuite à un problème assez difficile : la génération de cuboïdes aléatoires. Ce problème nous paraissait difficilement attaquable de front, car en générant aléatoirement des sommets ou même des facettes, les chances d'obtenir un cuboïde sont très faibles. Même en rajoutant de l'intelligence dans la génération, le problème ne nous paraissait pas calculable efficacement. Nous nous y prenons alors de manière détournée en déformant aléatoirement des cuboïdes existants. Cette manière de faire est également difficile mais plus pratique et a mené au développement d'une procédure de déformation très efficace. Elle exploite beaucoup les deux conditions caractérisant un cuboïde et nous montrons notamment que maintenir la première condition après déformation peut se faire en temps polynomial, en résolvant n programmes linéaires. La seconde condition est cependant beaucoup plus lente à vérifier, mais nous montrons comment une troisième condition plus pratique peut être équivalente à la seconde condition sous une certaine hypothèse. Cela nous permet alors dans la majorité des cas de maintenir la seconde condition en temps polynomial également.
Cela permet alors de déformer facilement des cuboïdes et d'en générer aléatoirement. De plus la famille de déformation que nous utilisons est très générale et est donc puissante. Enfin les résultats que nous avons obtenus pour garantir cette procédure ont également des retombées théoriques intéressantes sur les cuboïdes plus généralement.

La troisième partie porte principalement sur la complexité de la règle de pivot "greatest improvement" appliquée aux cuboïdes. Elle se découpe en une approche théorique et une approche expérimentale. L'approche expérimentale s'appuie intensivement sur l'algorithme de recherche de pires chemins présenté en première partie et sur la procédure de déformation des cuboïdes présentée en seconde partie. Nous montrons qu'il est possible d'effectuer une recherche locale directement sur les cuboïdes afin de générer des cuboïdes intéressants pour nos recherches. La première application de cela a été la recherche de cuboïdes "adversariaux" pour la règle "greatest improvement", ce qui a mené à la construction de conjectures intéressantes dont certaines que nous avons prouvées. La seconde application a été la possibilité de tester facilement des hypothèse.
Dans l'approche théorique, nous montrons divers résultats sur le comportement de la règle de pivot appliquée aux cuboïdes, notamment des bornes supérieures sur le nombre d'itération que peut faire la règle, et une famille de chemins candidate à un contre-exemple exponentiel à la règle. Nous avons ensuite étudié les cuboïdes en basse dimension et comment se comportait la règle dans ce cas-là. Nous montrons notamment que dans un cuboïde en 4 dimensions, un chemin issu de la règle "greatest improvement" ne peut pas être de longueur supérieure à 6, ce qui est un gain par rapport aux bornes théoriques que nous présentons.