Aspects numériques de l'algorithme LLL

Damien Stehlé et Gilles Villard
Proposition de stage, L3 informatique de l'ÉNS Paris

Titre du stage : Aspects numériques de l'algorithme LLL.
Mots clés : Réduction de réseaux, algorithme LLL, algèbre linéaire numérique.
Durée : 2 à 3 mois entre le 1er juin et le 15 septembre 2012.
Encadrants : Damien Stehlé (CR CNRS) et Gilles Villard (DR CNRS).
Lieu du stage : Équipe-Projet Aric, LIP, École Normale Supérieure de Lyon, 46 allée d'Italie, F69364 Lyon Cedex 07. `http://www.ens-lyon.fr/LIP/Arenaire/`
Téléphone : +33 (0)4 72 72 87 95.
Mél : `{damien.stehle,gilles.villard}@ens-lyon.fr`
Web : `http://perso.ens-lyon.fr/{damien.stehle,gilles.villard}`

Un réseau euclidien est un sous-groupe additif discret de R^n pour un certain n, ou encore l'ensemble des combinaisons linéaires à coefficients entiers d'une famille (appelée base) de vecteurs linéairement indépendants. Le caractère discret implique l'existence d'un vecteur non-nul de longueur minimale. Calculer un tel vecteur à partir d'une base quelconque est NP-difficile lorsque la dimension augmente (sous des réductions probabilistes), aussi se contente-t-on souvent d'un vecteur qui n'est pas nettement plus long. L'algorithme LLL, créé par Arjen et Hendrik Lenstra et László Lovász en 1982 [1], permet d'obtenir un vecteur de longueur inférieure à $2^{n/2}$ fois la longueur minimale. Cela s'avère suffisant pour de très nombreuses applications : en cryptographie (cryptanalyse de variantes de RSA), en calcul formel (factorisation de polynômes), en arithmétique des ordinateurs (calcul de polynômes approximateurs à coefficients flottants), en optimisation (programmation linéaire entière), etc (voir [2]).

Dans la pratique, le code le plus performant pour effectuer une réduction LLL est `fplll` [3], disponible sous licence LGPL, et intégré dans plusieurs bibliothèques de calculs mathématiques (Pari GP, Magma, SAGE). Le code effectue une série d'appels à des variantes heuristiques et rapides, et une variante plus lente mais rigoureuse implémentant l'algorithme L^2 de [4]. Cet enchainement de variantes permet d'obtenir une code à la fois rigoureux et efficace. L'efficacité pratique de `fplll` provient de l'utilisation de l'arithmétique flottante en faible précision pour les calculs d'orthogonalisation de Gram-Schmidt sous-jacents. Récemment, nous avons proposé une variante de L^2, appelée H-LLL, dans laquelle ces calculs d'orthogonalisation de Gram-Schmidt sont remplacés par l'algorithme de Householder [5]. Ce dernier a un comportement numérique très nettement meilleur, ce qui permet en théorie d'effectuer des réductions LLL en utilisant des précisions plus faibles pour les calculs flottants.

Après une familiarisation avec les réseaux euclidiens et l'algorithme LLL, l'étudiant devra implanter l'algorithme H-LLL, en langage C/C++. Il effectuera alors une comparaison avec le code actuel, à la fois du point de vue de la qualité numérique des calculs sous-jacents, et du point de vue de la performance. Il essaiera ensuite de proposer des optimisations de H-LLL, et d'intégrer H-LLL et ses éventuelles variantes dans la succession automatique des variantes de LLL implémentées dans `fplll`. Le code obtenu sera distribué librement (sous licence LGPL).

L'algorithmique des réseaux euclidiens est complexe à la fois du point de vue mathématique et informatique. Ainsi, ce stage est destiné à un étudiant avec à la fois une bonne capacité d'abstraction, et un goût prononcé pour la programmation.

[1] A. Lenstra, H. Lenstra and L. Lovász. Factoring polynomials with rational coefficients. Math. Ann. 261(4):515-534, 1982.
[2] P. Nguyen and B. Vallée (Eds). The LLL Algorithm, Survey and Applications. Springer, 2010.
[3] `http://perso.ens-lyon.fr/xavier.pujol/fplll`
[4] P. Nguyen and D. Stehlé. An LLL Algorithm with Quadratic Complexity. SIAM Journal on Computing, 2009.
[5] I. Morel, D. Stehlé and G. Villard. H-LLL: Using Householder inside LLL. In the proceedings of ISSAC 2009.