3J－4

ナップザック問題における解法拡張可能性の分析

伊藤禎敏，乾伸雄，野瀬隆，小谷善行，西村恕彦
東京農工大学工学部電子情報工学科コンピュータサイエンスコース

1. はじめに

ナップザック問題は代表的なＮＰ完全問題であり、一般には多項式時間で解く方法は見つかっていない。しかし、ナップザックベクトル $A=\{a_1, a_2, \cdots, a_n\}$ の大きさ n と、その要素の最大ビット数 $x = \log_2(\max\{a_1, a_2, \cdots, a_n\})$ から、$d = n / x$ で定義される密度 d が低い場合に限り、多項式時間で解く方法[1]が知られている。

本研究では、L^3 アルゴリズム[2]を用いた低密度ナップザック問題解決法、アルゴリズム SV[1]を実現する方法について研究した。

本稿ではその概要を報告する。

2. アルゴリズム SV の実現

アルゴリズム SV は、ナップザック問題を元にラティスの基底を作り、その基底から L^3 アルゴリズムを用いて reduced basis を計算するアルゴリズムである。すなわち基本的なアルゴリズムは、L^3 アルゴリズムである。よってアルゴリズム SV を実現するということは、L^3 アルゴリズムを実現することである。

L^3 アルゴリズムは、正確な数字の実数計算とベクトル計算によって実現される。

正確な数字の実数計算が要求されるので、浮動小数演算では問題がある。そこで、分数による実数表現を実現した。

3. 分数による実数表現

分数によって実数を正確に表現するには、分子分母ともに、桁数制限のない整数演算を行う必要がある。桁数制限のない整数を、次のように実現した。

```
struct {
        int len ;
        unsigned char *data ;
}
```

len には整数の byte 長を、data には整数を下位 byte から格納した。また、負数は len の値を負にすることで表現した。

a. 0 の表現

0 の byte 長は 0 byte だが、len の値は 1 にすることにした。また、正負どちらの場合も考えられるが、変換をして正に統一した。

各種演算を、基本的に byte ごとに演算することによって実現しているためである。また、同じ数値の表記を複数許すのは、混乱を招くおそれがあるので、正に表現を統一した。

b. 乗算

乗算は筆算の方式で実現した。

加算の繰り返しにより実現可能だが、その方法では加算の回数が膨大になる可能性がある。

それを筆算と 4bit ごとの表引きを利用して実現すると、基本的には表引きで結果を出すことができる。また加算回数も繰り返しに比べ、減少させることができる。

Analyzing Possibilities of Extending Solution in Knapsack Problem

Sadaharu ITO, Nobuo INUI, Takashi NOSE, Yoshiyuki KOTANI, Hirohiko NISHIMURA
Dept. of Computer Science, Tokyo University of Agriculture and Technology

4. 拡張の可能性

一般のナップザック問題に対して、アルゴリズムSVを適用した場合、その過程で得られるreduced basisについて次のことが重要だと考える。

なおその過程で得られるreduced basisのうち、

$$\mathbb{b}_i^* = (b_{i,1}^*, \cdots, b_{i,n+1}^*)$$

$$(b_{i,j}^* \in \{0, \lambda\}, 1 \leqq j \leqq n, {}^\forall\lambda)$$

となる$\mathbb{b}_i^*$について、

$$\mathbb{c}_i = (c_{i,1}, \cdots, c_{i,n})$$

$$c_{i,j} = b_{i,j}^* / \lambda$$

とした$\mathbb{c}_i$を以後候補ベクトルと呼ぶ。

a. 解とのハミング距離

候補ベクトルと解とのハミング距離の調査。

b. 候補ベクトル中のi番目の要素

各候補ベクトル$\mathbb{c}_j$のi番目の要素$c_{j,i}$

$$c_{j,i} = c_{k,i} \quad j \neq k$$

となるiがあるかどうかの調査。また、あった場合、解を$\mathbf{X} = (x_1, x_2, \cdots, x_n)$とすると、

$$x_i = c_{j,i}$$

であるかどうかを調べること。

c. 各候補ベクトル間の関係

各候補ベクトル間のハミング距離の調査。

5. 今後の課題

一つのナップザック問題にアルゴリズムSVを適用したとき、得られる候補ベクトルの個数はあまり多くはない。reduced basisのほとんどは、候補ベクトルにはならない、普通のベクトルである。候補ベクトルだけに注目するということは、reduced basisに含まれる情報の大半を落としてしまっていることにならないだろうか。

この普通のベクトルにも、解へつながる情報が含まれている可能性がないとはいえない。このベクトルもreduced basisであるから、ユークリッド距離が短いという傾向を持つ。このことが解を求める手掛かりになることはないだろうか。候補ベクトル以外のreduced basisの特徴を調査することが今後の課題である。

参考文献

[1] J. C. Lagrias and A. M. Odlyzko :
"Solving low density subset sum problems"
J. Assoc. Comp. Mach., Vol. 32, pp. 229-246, 1985

[2] Lenstra, A. K., Lenstra, H. W., Jr., and Lovasz, L. :
"Factoring polynomials with rational coefficients"
Math. Annalen 261, pp. 515-534, 1982

[3] 岡本栄司：
『暗号理論入門』
共立出版株式会社, 初版, 1993

[4] Andrew Clark, Ed Dawson, and Helen Bargen :
"Combinatorial Optimization and the Knapsack Cipher"
CRYPTOLOGIA, Vol. XX, No. 1, pp. 85-93, 1996