岡山理科大学紀要 第35号 A pp. 107-112(1999)

多次元0-1ナップザック問題に対する遺伝的アルゴリズムの適用

上山 圭一・岩崎 彰典*・太田垣 博一**
岡山理科大学大学院工学研究科修士課程電子工学専攻
*岡山理科大学情報処理センター
**岡山理科大学工学部電子工学科

(1999年11月4日 受理)

1. まえがき

多次元0-1ナップザック問題は，信頼性工学における最適化問題をはじめとして工学応用上頻繁に現れる重要な問題である。単一制約を持つ1次元0-1ナップザック問題は，従来分枝限定法や動的計画法などの有効な解法が研究されている[1)]。しかし，複数の制約条件を持つ多次元0-1ナップザック問題にそれらを適用することは困難である。その理由は，例えば分枝限定法では目的関数値と制約関数値の比により選択されるべき項目集合の順序の並び換えができ，それにより上限値と下限値を求めることができることを前提としているが複数制約を持つ問題では制約式が複数あるために，そのような並び換えを行うことができないからである。また動的計画法は整数優越性を使用するがこれも複数の制約条件を持つ問題では極端にその効率が悪くなるからである。

遺伝的アルゴリズム(Genetic Algorithm :GAs)は，生物の進化の過程を模倣した比較的単純な基本原理に基づいているが，従来の手法では解決が困難であったさまざまな最適化問題や探索問題に対して，実用的な最適解を速やかに得ることができる有効な手法として知られている[2)]。そこで，我々は多次元0-1ナップザック問題にGAsを適用し，その有効性を確かめる[3)]。

2. 多次元0-1ナップザック問題

ナップザック問題は目的関数値と制約関数値を持つ複数個の品物が与えられたときに制約関数が，ある制約許容量以内で幾つかの品物を選択し，そのときの目的関数を最大にする組合せをみつける問題である。

多次元0-1ナップザック問題は次のように定式化される。

[K]

$$\text{maximize} \qquad f(x) = \sum_{n=1}^{N} c_n x_n \tag{1}$$

$$\text{subject to} \qquad g_m(x) = \sum_{n=1}^{N} a_{mn} x_n \leq b_m \tag{2}$$

$$x_n \in \{0,1\}, \quad n = 1,2,\cdots,N, \quad m = 1,2,\cdots,M \tag{3}$$

ここで係数c_n，a_{mn}は，n番目の品物の価値や重さなどを表す正の数である。$b_m(>0)$は制約許容量，M，Nはそれぞれ制約条件と変数との個数を表している。

3. 遺伝的アルゴリズム

遺伝的アルゴリズム(Genetic Algorithm :GAs)は生物進化の原理に着想を得たアルゴリズムで最適化の一手法である。GAsでは遺伝子を持つ仮想的な生物の集団を計算機内に設定し，あらかじめ定めた環境に適応している個体が子孫を残す確率が高くなるよう世代交代シミレーションを実行し，遺伝子および生物集団を進化させる。このためこれらの仮想生物の進化によって，与えられた工学的課題の解が得られるようにGAsのプログラムミングを行う。GAsの特徴の一つは交叉，突然変異，選択という生存競争に基づく確率的原理により，組合せ問題のような計算量の多い問題において良好な解を見つけることができることにある[2)4)]。

遺伝的アルゴリズムでは図1に示す手順に従って解を改善させる。図1の各項目について以下に述べる。

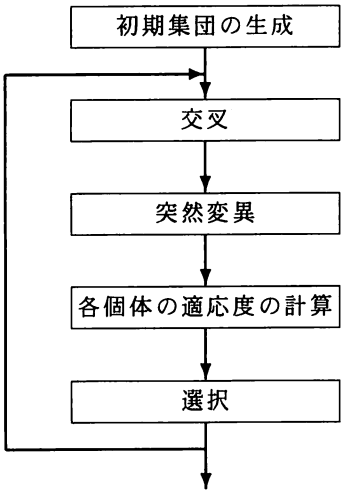

図1 GAsの処理手順

(1)初期集団の生成

GAsで解を探索するためには，まず問題の解候補を文字列にマッピングする。0-1ナップザック問題では各変数に遺伝子となる0か1をランダムに与え，それを一つの個体とする。同じようにいくつかの個体を生成し，それらを初期集団とする。ここで変数の個数をN，生成する個体の総数をLとし，各個体の遺伝子型を$I_n^l(l=1,2,\cdots,L,\ n=1,2,\cdots,N)$と表すことにする。図2に20変数の問題で$L$個の個体からなる初期集団の例を示す。

I_n^1 00100011000000000000
I_n^2 00000100010001000001
I_n^3 01001000100100100100
$\vdots$
I_n^L 00100000010010010000

図2 初期集団の例

(2)交叉

生成されたL個の初期集団の中から二つの個体のペアを何組かランダムに選択し，それぞれに対して交叉と呼ばれる操作を実行する。交叉は二つの個体の間で文字列の部分的な交換を確率的に行う操作である。本研究では，文字列をランダムに選んだ二ヶ所の交叉位置で切断し，中央部を入れ換える2点交叉を使用した。図3に2点交叉の例を示す。この図ではI_n^A，I_n^Bがある一つのペアとして選ばれた個体で，左から7ビット目と8ビット目の間，14ビット目と15ビット目の間を交叉位置とし，中央部を入れ換えることによって新たな固体I_n^C，I_n^Dを作っている[2)]。

親	I_n^A	0010000	0010010	000001
	I_n^B	0000100	0110000	001000
			↓	
子	I_n^C	0010000	0110000	000001
	I_n^D	0000100	0010010	001000

図3 2点交叉の例

交叉によって生成された子の個体I_n^C，I_n^Dは，親の個体I_n^A，I_n^Bのそれぞれから形質を継承した個体である。これは探索空間において，現在の探索点とは少し異なる位置に新しい探索点を生成させることに相当している。初期集団では，さまざまな遺伝子型をもつ個体があるため，交叉によっても同様にいろいろな個体が生じる。一方，進化が進み遺伝子型がある傾向に定まりつつある過程では，どの個体の遺伝子型にも大きな差がないため，交叉によって作り出される固体の遺伝子型も似通ったものになる。すなわち，探索空間を

はじめは大局的に調べ，傾向をつかんでからはより詳細に調べるということになる[2)]。

(3)突然変異

交叉に続いて，突然変異と呼ばれる操作を実行する。突然変異は，個体中の特定の遺伝子を別な遺伝子に確率的に交換する操作である。図4に突然変異の例を示す。この図では左から12ビット目の遺伝子に突然変異が生じている。

突然変異前　00010010000 | 0 | 01100001

↓

突然変異後　00010010000 | 1 | 01100001

図4　突然変異の例

この突然変異の役割は二つあり，一つは交叉で得られた解の近傍を探索すること，もう一つは交叉だけでは得られない遺伝子のパターンを生成することである。これにより局所解からの脱出が可能となる。

(4)各個体の適応度の計算

適応度は，個々の個体の競争力を表す。本研究では(4)式のような適応度関数を設計し，これによりそれぞれの個体の適応度を計算する。多次元0-1ナップザック問題では，全ての制約許容量を満たす解を得るためにペナルティ乗数λ_mをうまく設定する必要がある。

$$\begin{cases} f(x) & (g_m(x) \leq b_m) \\ f(x) - \sum_{m=1}^{M} \lambda_m (g_m(x) - b_m) & (g_m(x) > b_m) \end{cases} \tag{4}$$

(5)選択

選択は，適応度に応じてより環境に適した個体を選ぶ操作である。本研究では適応度の大きいものから順に選択するエリート戦略を使用した。

以上の処理によって，次世代のL個の個体を決定する。そして再び交叉，突然変異，適応度の計算，選択の過程を何世代か繰り返すことにより最終的にすべての制約条件を満たし，より目的関数の大きい解を得ることができる。

4. 計算機実験

個体数$L=20$とし乱数で生成した価値c_nおよび重さa_{1n}と容積a_{2n}をもつ2次元の0-1ナップザック問題について計算機実験を行った。またGAsにより解が改善される様子を視覚的に把握し、遺伝的アルゴリズムの各種パラメータを決定するため、図5のようなツールを作成し，このツールを使用して計算機実験を行った。この図5に50変数の2次元0-1ナップザック問題の解の初期状態を示す。

表1　2次元0-1ナップザック問題例

n	1	2	3	4	$\cdots$	$N-1$	N
c_n	24	83	49	36	$\cdots$	54	68
a_{1n}	38	67	92	78	$\cdots$	38	41
a_{2n}	53	28	87	19	$\cdots$	29	56
I_n^1	0	1	0	0	$\cdots$	1	1
I_n^2	1	1	0	0	$\cdots$	0	1
I_n^3	1	1	0	1	$\cdots$	1	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
I_n^{20}	1	0	0	1	$\cdots$	0	1

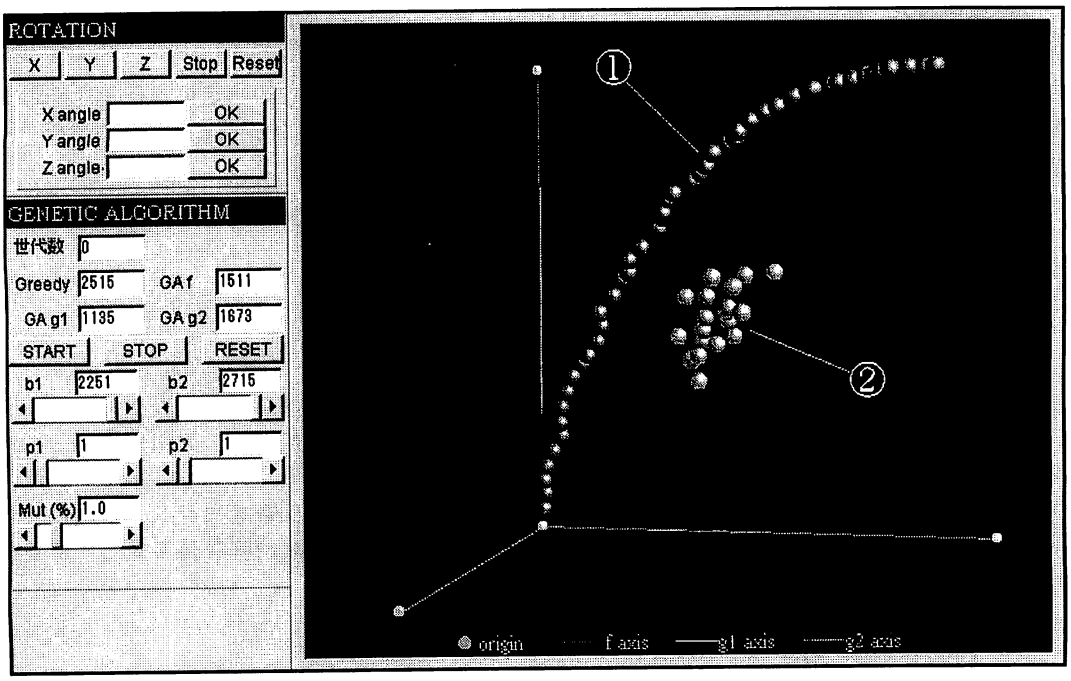

図5 50変数2次元0-1ナップザック問題に対する
GA適用のアニメーションツール

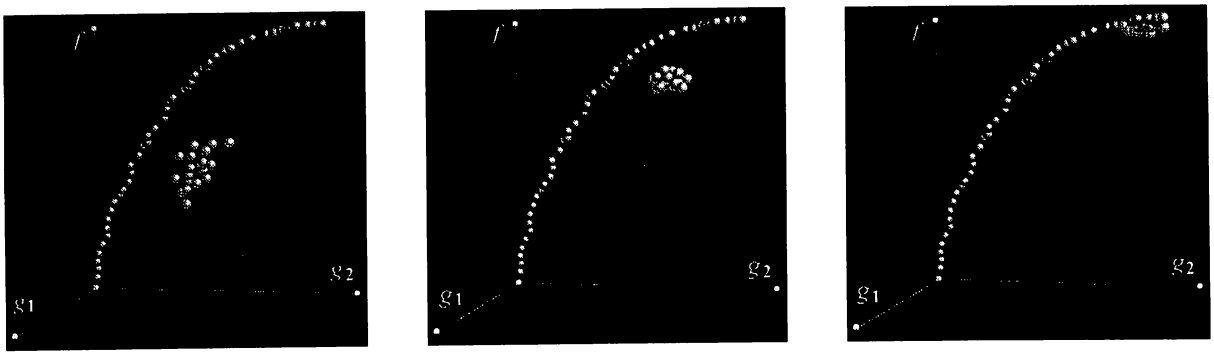

図6 解が改善される様子

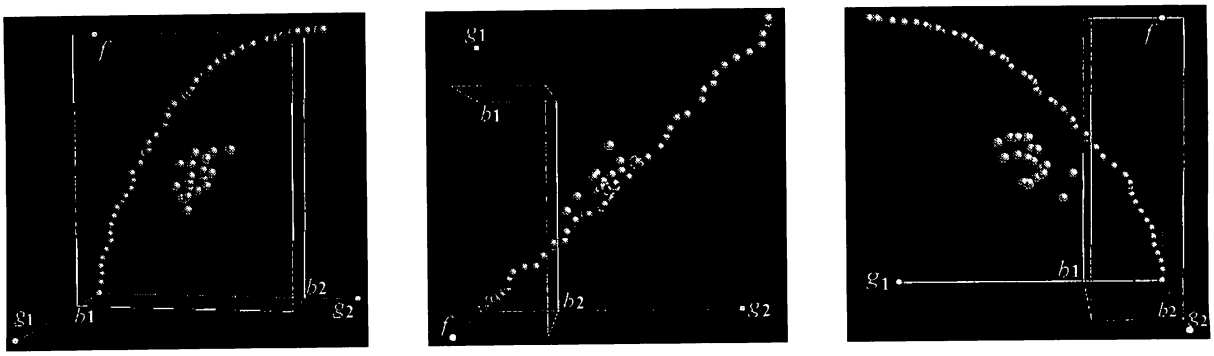

図7 視点の変更

図5において① は$c_n/\sqrt{a_{1n}^2+a_{2n}^2}$の大きい順に足したもので解が改善される様子を調べるための目安である。② はランダムに生成された初期集団で，これに遺伝的操作を行い解が改善される様子を調べる。このツールを用いて実験を行うことには，次のような特徴がある。

1. 解が改善される様子が視覚的にわかる(図6)
2. 解の挙動を見ながら制約許容量，突然変異率，ペナルティ乗数を容易に変更することができる
3. 全ての制約を満たしているかどうか角度を変えて見ることができる(図7)

5. 実験結果

制約条件の個数$M=2$，変数の個数$N=30,50,100$の問題についてそれぞれ制約許容量を変化させたときの結果を表2に示す。制約許容量$b1_{(\%)}, b2_{(\%)}$は、それぞれの制約関数値の合計に対する割合を表している。$f(x)$はGAsによって求めた目的関数，$\tilde{f}(x)$は制約許容量以内で$c_n/\sqrt{a_{1n}^2+a_{2n}^2}$の大きい順に再帰的に足すことによって求めた目的関数である。この計算機実験では毎回GAsのほうが良い解を得ることができた。

表2 計算機実験結果

N	$b1_{(\%)}$	$b2_{(\%)}$	$f(x)$	$\tilde{f}(x)$	$f(x)-\tilde{f}(x)$
30	50	50	1288	1267	21
	100	50	1319	1314	5
	50	100	1344	1312	32
	80	10	633	605	28
	10	80	677	533	144
50	50	50	2076	2067	9
	100	50	2120	2067	53
	50	100	2164	2147	17
	80	10	1003	890	113
	10	80	1023	858	165
100	50	50	4205	4170	35
	100	50	4294	4170	124
	50	100	4453	4425	28
	80	10	1994	1744	250
	10	80	3016	2794	222

6. むすび

2次元0-1ナップザック問題にGAsを適用する方法を提案し，3Dのアニメーションによって視覚化した。GAsを適用して解が改善される様子を視覚的に把握しすることによりGAsで良好な解を得るために必要なさまざまなパラメータの設定が容易にできた。

参考文献

1) 今野浩，鈴木久敏：整数計画法と組合せ最適化，日科技連出版社(1982)
2) 安居院猛，長尾智晴：ジェネティックアルゴリズム，昭晃堂(1993)
3) 上山圭一，岩崎彰典，太田垣博一：多次元0-1ナップザック問題に対する遺伝的アルゴリズムの適用 JAMS Annual Meeting (1999)
4) 北野弘明：遺伝的アルゴリズム，産業図書(1993)

An Application of a Genetic Algorithm to Multi-Dimensional 0-1 Knapsack Problems

Keiichi UEYAMA, Akinori IWASAKI * and Hirokazu OHTAGAKI **

Graduate School of Engineering,
**Information Processing Center,*
***Department of Electronic Engineering, Faculty of Engineering,*
Okayama University of Science
Ridai-cho 1-1, Okayama 700-0005, Japan

(Received November 4, 1999)

0-1 knapsack problems are very important, and appear frequently in a wide variety of engineering applications. Some effective methods are presented for solving 0-1 knapsack problems with a single constraint. However, 0-1 knapsack problems with multiple constraints are very difficult to solve. We propose a method based on a genetic algorithm to solve multi-dimensional 0-1 knapsack problems, and visualize the solution process to enable the various parameters of the genetic algorithm to be decided in order to obtain high quality solutions.