ナップザック問題に対する遺伝的アルゴリズムと動的計画法の比較

日大生産工 (学部)　○藤田 真広　　日大生産工　吉田典正

1.　はじめに

遺伝的アルゴリズムの応用的な利用において、基本となる考え方にナップザック問題の解法がある。本研究では遺伝的アルゴリズムの基礎研究として、ナップザック問題に対して、代表的な手法である動的計画法と、遺伝的アルゴリズムの2つの解法を用いて、計算時間と解の精度を比較、評価する。

2.　ナップザック問題

2.1 問題の定義

ナップザック問題とは、複数の品物とそれぞれの品物の体積と価値が与えられたときに、決められた容積の袋に入る価値が、最大になるような品物の組み合わせを求める問題である。

本研究で扱うナップザック問題では、容積Wのナップザックと、価値p_iと体積w_iを持つn個の品物$i(1 \le i \le n)$が与えられており、容積の範囲内でかつ価値が最大になる要素の選択を探すものである。

目的関数を最大とする要素数を求める問題を最大化問題といい、ここで、容積の範囲内というのは制約条件と呼ぶ。また、制約条件が満たされる品物組み合わせを解と呼ぶ。解の中で目的関数の価値が最大となるものを最適解という。ナップザック問題の応用は、地下に置けるパイプの配置や、輸送車両などによる荷物の積み込みなどがあげられる。

2.2 例題

表 2.1 のように各品物の価値･体積とナップザックの容積を与えられたものとする。

最適解は、要素 1 と 2 の組み合わせであり、価値の合計は 22 となる。しかし、他にも解が生成される。例えば要素 0 だけが選択され、価値の合計 12･体積の合計 16 の組み合わせなどがある。

表 2.1 ナップサック問題の例題

要素番号	体積	価値
0	4	3
1	5	7
2	11	15
3	13	19
ナップザックの容積		17

3.　動的計画法 (Dynamic Programming)

3.1 動的計画法の概要 1)

与えられた問題を解く際に、小規模な問題の解を用いて大規模な問題の解を求めると、簡易に問題を解くことが出来る事が多い。このような考え方に基づいたアルゴリズムの設計法が動的計画法である。

3.2 動的計画法による解法

2.1 節のナップザック問題が定義されているとき、小規模問題の解を記憶させて、より大規模な問題、ナップザックに入る要素を得ることが出来る。

ナップザックの容積が 0 の場合は要素がいくつであろうとも、ナップザックの中には入らないので総価値は 0 となる。

最初に、i=0 の段階では品物 0 だけを考えて大きさが 0 から 17 までのナップザックにはいる価値を最大にすることを考える。次にi=1 以降の段階では、以前の結果を利用して、それぞれの荷物を加えた場合に価値が増加するかどうかを調べる。以上の動作を繰り返し、i=3 まで終了した段階で最高の価値から逆算し、解を求める。

4.　遺伝的アルゴリズム (Genetic Algorithm)

4.1 遺伝的アルゴリズムの概要 2)

データ(解の候補)を遺伝子で表現し、選択･交叉･突然変異などの操作を繰り返しながら解を探索する。評価関数の可微分性や単峰性などの知識がない場合であっても、

Comparison of the Genetic Algorithm and the Dynamic Programming in a Knapsack Problem

Masahiro FUJITA and Norimasa YOSHIDA

適用可能な最適化手法である。 必要とされる条件は評価関数の全順序性と、探索空間が位相（トポロジー）を持っていることである。

4.2 遺伝的アルゴリズムの構成と流れ

初期染色体集団の生成、適応度の計算、両親の選択･交叉、突然変異、終了条件の設定を行う。

4.3 遺伝的アルゴリズムによる解法

4.3.1 初期染色体集団の生成

初期染色体集団は、染色体の個数 n と要素数である染色体長 l のパラメータで行列を生成する。l は各品物の最大で持ちうる大きさの合計に設定し、染色体集団の遺伝子はランダムに 0 か 1 に設定する。

4.3.2 適応度の計算

適応度とは、染色体が持っている遺伝子情報から得られた環境に適している度合いを示した値である。本研究の遺伝的アルゴリズムでは適応度は価値の合計とし、品物の合計がナップザックの容積を超えた場合は悪い適応度を与えた。

4.3.3 両親の選択

両親の選択とは、交叉を行う対象である 1 対の親を確率的に選択するために行うもので、適応度が高い固体が選択される可能性が高いようにしてある。本研究では、ルーレット選択を用いた。

4.3.4 交叉

交叉とは、両親の選択で選ばれた 1 対の染色体から新たな染色体を生成するもので、子は親の特徴を受け継ぎ、新しい染色体集団の一つとなる。本研究の交叉方法は 2 点交叉を用いた。

4.3.5 突然変異

突然変異とは、特定の遺伝子を別な遺伝子に変更する操作である。主に、交叉で得られた解の近傍を探索することと、交叉では得られない染色体を生成することを目的に行われる。本研究では 1 点逆位を用いた。

4.3.6 終了条件

本研究では一定処理回数 T において近似解に変化が無かった場合、それを終了条件とした。

5. 実行結果と比較

プログラムは VisualC++6.0 を用いて作成した。アルゴリズムとして単純な動的計画法は予測通りの時間で動いた。それに対し遺伝的アルゴリズムは、表 5.1 が示すように処理時間が長く正確性に欠いた。しかしながら、遺伝的アルゴリズムは、動的計画法で解くことのできない複雑な問題にも適用可能である。

表 5.1 遺伝的アルゴリズムと動的計画法の比較

容積		100	1000	10000
DP での最適解		140	1400	14000
GA での近似解	T =1000	105	109	Stack Overflow
	T =10000	138	Stack Overflow	Stack Overflow
DP での実行時間(sec)		1.130E-05	8.878E-05	6.118E-04
GA での実行時間(sec)	T =1000	2.241E-03	8.327E-03	Stack Overflow
	T =10000	3.303E+00	Stack Overflow	Stack Overflow

6. まとめ

本研究では遺伝的アルゴリズムの基礎としてナップザック問題を取り扱ったが、初期染色体集団の生成において、染色体長を配列として多くとりすぎたために、処理時間が多くかかり、最適解の正確性に欠いた。また、要素数が大きい場合においては Stack Overflow を起こしてしまった。

今後の研究としては、初期染色体集団の生成に改良を加え、交叉･突然変異などのパラメータを調節し、正確かつ短時間で終了する遺伝的アルゴリズムを作成すると共に、遺伝的アルゴリズムの応用的利用について考える。また、遺伝的アルゴリズムを利用し、社会現象のシミュレーションに取り組む予定である。

参考文献

1) 吉田典正,アルゴリズム入門,日大生産工 管理工学科,(2004), pp. 122-125.

2) 長尾智晴, 最適化アルゴリズム, 昭晃堂, (2000).