ナップサック問題のための探索範囲調節型 GA の提案

正員 井田 憲一* 学生員 菅 良平**
正員 玄 光男**

Proposal of Adjustment Type Genetic Algorithm for Knapsack Problem

Kenichi Ida*, Member, Ryouhei Suga**, Student Member, Mitsuo Gen**, Member

The greedy algorithm is one of solution methods for knapsack problem. Although this algorithm does not necessarily obtain the optimal solution, it can obtain a good solution in short time. We consider this algorithm is very effective at judging the importance of each item.

In this paper, we propose a new genetic algorithm for solving a knapsack problem. The algorithm can adjust a search area in consideration of the stability of each item which can obtain from the greedy algorithm. Moreover, we apply the proposed method to a multi-objective problem and a large-scale problem, and test the effectiveness.

キーワード：遺伝的アルゴリズム，ナップサック問題，貪欲解法，探索範囲の自動調節

Keywords: genetic algorithm, knapsack problem, greedy algorithm, automatic adjustment of search range

1. はじめに

一般に，ナップサック問題⁽¹⁾は複数の荷物のセットで成り立っており，各荷物には重量と価値が付随している。また，ナップサックには許容量が規定されており，中に入れた荷物の価値が最大になるような荷物の組合せを見つけるもので，一種の組合せ最適化問題である。この問題は NP 困難な問題とされ，どのようなサイズの問題に対しても実用時間内に確実に最適解が求まるアルゴリズムはまだ報告されていない。

したがって，この種の組合せ最適化問題では，厳密解法で実用時間内に解くことが困難とされる大規模な問題において，いかに効率よく精度の高い近似解を求めるかが重要な課題の一つになっている。この課題を解決するための手段としては，解空間の分割や縮小が効果的ともされており，これまでに多くの提案がなされている⁽¹⁾⁽²⁾。

このナップサック問題の解法の一つに貪欲解法 (greedy algorithm)⁽¹⁾がある。貪欲解法は単純な手法のため短時間で結果が求まるが，必ず優れた解が求まるとは限らない。

しかし，この手法は各荷物の重要度を判断する上で非常に有効であると考えられる。

本論文では，この種の組合せ最適化問題の有効な解法とされる遺伝的アルゴリズム (genetic algorithm: GA)⁽³⁾⁽⁴⁾に対し，貪欲解法を初期解生成アルゴリズムとして用いることで不安定な荷物に対応する遺伝子への重点的な探索を可能にするとともに，各荷物の安定性を考慮して探索範囲を徐々に拡大させる範囲調節型 GA (adjustment type GA: ATGA)⁽⁵⁾を提案する。更に，提案手法を多目的 0-1 ナップサック問題に適用することでその有効性を検証する。

2. ナップサック問題

ナップサック問題は，複数の荷物と一定容量まで荷物を入れられる袋があり，各荷物の重量とその価値 (重要度) は既知とするとき，袋の容量を越えない範囲でその価値の和が最大となるように入れる荷物を決定する問題である。

一般に， n 個の荷物をもつ 0-1 ナップサック問題は，次のように定式化される⁽¹⁾。

$$\max \quad v = \sum_{i=1}^n c_i x_i \cdots \cdots \cdots (1)$$

$$\text{subj. to} \quad \sum_{i=1}^n a_i x_i \leq b \cdots \cdots \cdots (2)$$

$$x_i = \{0, 1\} \quad (i = 1, \dots, n) \cdots \cdots (3)$$

ここで， c_i は荷物 i の価値， a_i は荷物 i の重量， b は荷物

* 前橋工科大学
〒 371-0816 前橋市上佐鳥町 460-1
Faculty of Engineering, Maebashi Institute of Technology
460-1, Kamisadori-cho, Maebashi 371-0816
** 早稲田大学大学院
〒 808-0135 北九州市若松区ひびきの 2-7
Graduate School of Information, Production and Systems,
Waseda University
2-7, Hibikino, Wakamatsu-ku, Kitakyushu 808-0135

を格納するナップサックの許容重量であり, x_i は荷物 i をナップサックに格納するとき 1, そうでないとき 0 をとる 0-1 変数である。また, 荷物には順に $1, \dots, n$ の番号がつけられているものとする。

3. 提案手法

ここでは, 本論文で提案する手法 ATGA を構成する上で重要な要素となる, 起因レベルの設定と改良点について示す。

3.1 起因レベル 貪欲解法より得られた解の安定度を本論文では起因レベルと呼ぶ。ここで, 貪欲解法とは, n 個の荷物を荷物の価値と重量の比率 (本論文では, これを価値密度と呼ぶ) c_i/a_i が高い順に並べ, 許容重量 b まで荷物を詰め込んでいくナップサック問題の近似解法の一つである⁽¹⁾。この解法は単純な手法のため短時間で結果が求まるが, 必ず最適解が求まるとは限らない。しかし, この手法で用いられている荷物の価値密度に従って並べ替える方法は, 各荷物の重要度を判断する上で非常に有効であると考えられる。

このため, 提案手法 ATGA の染色体表現 $x_i(1, \dots, n)$ においては, 全ての荷物を価値密度 c_i/a_i の高い順に並べるものとする。また, 本論文では, 貪欲解法で得られた解 (貪欲解) を染色体表現した個体のビット列において, 格納した荷物と格納されない荷物を表す 1 と 0 の境目を境界と呼び, その右隣の遺伝子座を $s = \min \{h \mid \sum_{i=1}^h a_i > b\}$ として定義する。また, 染色体表現において境界 s を基準に荷物が格納されている側を境界左側 ($1 \leq i < s$), 荷物が格納されていない側を境界右側 ($s \leq i \leq n$) と呼ぶことにし, この境界 s からの荷物の距離を $i - s (i = 1, \dots, n)$ で表す。

貪欲解法では, 得られた解に対し,

$$\sum_{i=1}^n a_i x_i = b \dots \dots \dots (4)$$

が成り立つとき, 最適解であることが保証される。

しかし, 多くの場合

$$\sum_{i=1}^n a_i x_i < b \dots \dots \dots (5)$$

となる。このことは, 許容重量 b であるナップサックに

$$d = b - \sum_{i=1}^n a_i x_i \dots \dots \dots (6)$$

で表すような容量の空き d が存在していることを意味し, この空きをベースに, 貪欲解において格納されている荷物と格納されていない荷物を交換することにより, 貪欲解より良い解となる (以下, エラーと呼ぶ) ことも十分考えられる。ここで, 貪欲解の染色体と比較して, 貪欲解より良い解を形成する要因となっている遺伝子 (荷物) をエラーを引き起こした遺伝子 (荷物) と呼ぶ。このような現象が起こる具体的な例を図 1 に示す。尚, このデータはあらかじめ全

Capacity(b) 16						
	Border					
Item number	1	2	3	4	5	6
Value(c_i)	8	6	7	5	2	1
Weight(a_i)	2	4	6	5	3	7
Distance from Border	-3	-2	-1	0	1	2
Greedy solution	1	1	1	0	0	0
Better solution	1	0	1	1	1	0
	Error items					
	Fitness Total Weight					
	21 12					
	22 16					

図 1 エラーを引き起こした遺伝子
Fig. 1. Gene which caused the error.

図 2 境界からの荷物の距離とエラー度数の関係
Fig. 2. Relationship between frequency and distance from "border".

ての荷物を価値密度の降順に並べたものを使用している。

一般に, 対象となる問題の規模の大きさに比例して探索コスト (または時間) が増大する。したがって大規模な問題においては, いかにこれを削減して効率よく優れた解を探索できるかが重要なポイントになっている。そのため, 本論文では各荷物のエラーを引き起こす割合に注目した。貪欲解に対してエラーを引き起こしやすい不安定な荷物をあらかじめ判定できれば, 不安定な荷物を重点的に探索することで効率的な探索が可能となるためである。

本論文では, このエラーを引き起こす確率の高い荷物の特徴を調べるための予備実験を行った。この実験は, 1) まず荷物数が 100, 500 の規模の問題で各 5 種類ずつ問題をランダムに生成し, 各問題で貪欲解を求める。2) 次にその染色体と貪欲解より良い解の染色体を比較することでエラーを引き起こす要因となった荷物を見つけ, 各問題の境界を基準としてエラーを引き起こした荷物の出現頻度を測定する, という手順で行われた。

実験結果を図 2 に示す。尚, この図の範囲外の距離 (荷物) ではエラーが全く起こらなかったため表示を省略している。

この結果より、明らかに境界付近でエラーを引き起こす頻度が高いことが確認できる。また、境界より少し離れた場所でエラーを引き起こした荷物について解析した結果、境界左側では比較的重い荷物、境界右側では軽い荷物がそれぞれエラーを引き起こしていることが確認できた。

この実験から、エラーを引き起こしやすい荷物の特徴として次の結論を得る。

- 境界左側 ($1 \leq i < s$) の荷物で、その平均値より重い荷物
- 境界右側 ($s \leq i \leq n$) の荷物で、その平均値より軽い荷物
- 境界 s 付近の荷物

本論文では、このエラーを引き起こしやすい荷物のことを不安定遺伝子、それ以外の荷物を安定遺伝子と呼ぶことにする。

3・2 起因レベルの設定 この結果から、本論文では各荷物（遺伝子）のエラーを引き起こす度合を起因レベル $R_i (i = 1, \dots, n)$ として定義し、以下のように設定する。

$$R_i = \begin{cases} r_i + 1 & (\frac{1}{2}s < i \leq \frac{3}{4}s \text{ または } \frac{5}{4}s < i \leq \frac{3}{2}s) \\ r_i + 2 & (\frac{3}{4}s < i \leq \frac{5}{4}s) \\ r_i & (\text{上記以外}) \end{cases} \dots\dots\dots (7)$$

ここで、

$$r_i = \begin{cases} \frac{a_i}{\bar{a}_l} & (1 \leq i < s) \\ \frac{\bar{a}_r}{a_i} & (s \leq i \leq n) \end{cases} \dots\dots\dots (8)$$

$$\bar{a}_l = \frac{\sum_{j=1}^{s-1} a_j}{s-1}, \quad \bar{a}_r = \frac{\sum_{j=s}^n a_j}{n-(s-1)} \dots\dots\dots (9)$$

である。

これらの式は、境界左側で重い荷物、境界右側で軽い荷物の起因レベルを高めることを目的とした式 (8)、(9) と、境界付近の荷物の起因レベルを高めることを目的とした式 (7) で構成されている。式 (9) の $\bar{a}_l$ 、 $\bar{a}_r$ はそれぞれ境界左側、境界右側にある荷物の重量の平均値を表しており、式 (8) の上の式は境界左側で平均値より重い荷物の起因レベルを上げ、下の式は境界右側で平均値より軽い荷物の起因レベルを上げる働きをする。また式 (7) は、式 (8) から得られた r_i の値に境界からの荷物の距離に応じた値を付加することで、境界付近の荷物の起因レベルを上げる働きをする。ここで、境界からの荷物の距離に応じて付加する値は、式 (8) から得られる平均的な値 $r_i = 1$ を考慮して設定されており、 r_i で求めた値に対して偏りが生じないようにしている。またその値を付加する範囲については、境界右側左側それぞれの範囲で、単純に半分より境界に近い荷物に値を付加し、さらにその半分より境界に近い荷物に対しては、より高い値を付加する。このように、式 (7) については比較的広い範囲の荷物に値を付加することで、極端な探索コ

ストの削減を避け、特性の異なる問題への汎用性を考慮した設計とした。

3・3 改良点と提案アルゴリズム ここでは、上記の安定遺伝子、不安定遺伝子を考慮することで、更なる探索効率の向上を目指した提案手法 ATGA の具体的な改良点とそのアルゴリズムを示す。

• 突然変異

起因レベルを突然変異確率として用いることで、不安定な荷物に対応する遺伝子を重点的に突然変異させ、探索を活性化させることを目的とする突然変異法を導入する。

• 局所探索の導入

探索空間の制約条件であるナップサックの許容量を大きく超える量の荷物が格納されている個体、もしくはナップサックの容量にまだ十分空きがある個体に対して、あらかじめ定めた有効解集団の範囲内に引き戻すための補正を行う。この局所探索法は、ナップサックの許容量を大きく超える量の荷物が格納されている個体に対しては不安定遺伝子を比較的高い確率で補正し、容量にまだ十分空きがある個体に対しては安定遺伝子を比較的高い確率で補正することで、質の良い解を得ることを目的とする。

• 起因レベルによる探索範囲の調節

図 2 から分かるように、ほとんどの問題において境界付近に不安定な遺伝子が多く存在する。一般に探索範囲が狭いほどその範囲で得られる解の質も高くなりやすい。しかし探索範囲を縮小する際、範囲の設定を誤ると最適解を得ることが不可能になる場合もある。そのため提案手法では、初期の段階では探索する重要性が最も高いと思われる境界付近の荷物を中心に範囲を縮小し、数世代ごとに範囲を拡大するような探索方式を採用している。具体的には以下の手順を GA に組み込むことで、段階的かつ効率的な探索を実現する。

- (1) 貪欲解法で得られた解に対し、 $[s-1, s]$ の範囲、つまり境界両隣の荷物まで探索範囲を縮小し、これを初期探索範囲として設定する。
- (2) 起因レベルによって定めた世代ごとに探索範囲を徐々に拡大させる。
- (3) 探索範囲が全ての不安定遺伝子に及ぶまで拡大し、あらかじめ定めた終了条件を満たすまで探索を続ける。このとき、最良解が更新された場合は (1) に戻る。

提案手法では、探索範囲を徐々に拡大させることで段階的な探索を可能にしているが、段階的な探索を効率的に行うには同じ探索範囲をどのくらいの期間（世代数的に）探索するか決定が重要となる。本論文では、この決定に各荷物の安定度を数値化した起因レベルに基づく尺度を導入することで、新たに探索範囲に含まれた遺伝子が不安定遺伝子ならば深く（長い期間）探索し、安定遺伝子ならば探索に時間をかけず探索コ

ストを削減するような探索範囲を自動調節する機能を実現した。

以下に、提案手法 ATGA のアルゴリズムを示す。

Step1: データの入力

対象とする問題の各要素を入力する。

Step2: 個体の遺伝子型の決定

荷物を表す変数を価値密度 c_i/a_i の高い順に整列し、各変数 x_i の値を並べた n ビットの 0-1 記号列を個体の遺伝子型とする。

Step3: 貪欲解法による初期個体群の生成

貪欲解法を適用して初期解を生成する。この解を集団サイズだけ複製し、これを初期個体群とする。したがって、これらの初期個体は全て価値密度の高い順に許容量まで荷物が格納されている状態にある。

Step4: 起因レベルの計算

式 (7) を用いて各荷物の起因レベル R_i を計算し、その結果を遺伝子情報の一つとして格納する。

Step5: 初期探索範囲の設定

初期探索範囲は貪欲解法で得られる解の境界 s を中心に左右一定のビット列分の範囲とする。このとき縮小させる範囲は境界に最も近い不安定遺伝子を含む範囲とする。本論文では、範囲 $[s - 1, s]$ を初期探索範囲としている。

Step6: 解の探索

終了条件を満足するまで以下の手順を繰り返す。

(1): 個体の評価

各個体の評価値を求める。このとき、最良解が更新された場合は、Step5 に戻る。

(2): 探索範囲の調節

起因レベルを用いて探索範囲を拡大する。同一範囲での探索は、新たに探索範囲に含まれた遺伝子、もしくは現在の探索範囲で最も外側にある遺伝子の起因レベルに基づく値 $10 * R_i^{10}$ に等しい世代数分 (期間) 続けられ、その後、外隣の遺伝子まで探索範囲を拡大する。

(3): 遺伝的操作

選択を行い、定められた探索範囲内で交叉と突然変異を行うことでそれぞれ子個体を生成する。尚、突然変異については起因レベルに基づく値 $R_i/10$ を各荷物の突然変異確率として用いる。

(4): 局所探索

個体群の中で $\sum_{i=1}^n a_i x_i \leq b - W_{max}$ を満たす個体に対して補正を行う。ただし、 W_{max} は荷物の中で最も重い荷物の重量とする。ここで補正の対象となる荷物は、不安定遺伝子に比べ、安定遺伝子を高い確率で選択させるため起因レベルの逆数 $1/R_i$ の確率で選択する。また、 $b + W_{max} \leq \sum_{i=1}^n a_i x_i$ を満たすような個体に対しては、安定遺伝子に比べ、不安定遺伝子を高い確率で選択させるため起因レベル R_i の確率で選択するようにしている。

個体が補正を適用する条件を満たさなくなるまで、またはあらかじめ設定しておいた回数 (N_r) に達するまで

この操作を繰り返す。

尚、アルゴリズム中の起因レベルに基づく各値については、実験による試行錯誤的な経験から得られたものを用いている。

4. 数値実験

4.1 改良点の性能分析と解の信頼度 ここでは、提案手法の各改良点の有効性を検証するため、以下の実験を行う。

実験での遺伝的操作は、選択法にルーレット戦略とエリート戦略を用い、エリート保存数は 1 とする。また、交叉法は 2 点交叉を用いる。更に、集団サイズを 500 とする。パラメータは、あらかじめ予備実験により求めた、各手法にそれぞれ適した値を設定している。試行は各手法について 10 回行い、十分な収束が得られるまでとした。問題のサイズは、荷物数 500 である。実装は AthlonXP1900+、メモリ 256MB の PC 上の VisualC++6.0 で行っている。

最初の実験は、提案する突然変異の効果を確認するためのものである。実験結果を表 1 に示す。表中の “None” は、提案突然変異法を使用しない一般の突然変異を意味する。表 1 から、最良値 (Best) については両手法とも同じ値を得ることができたが、最悪値 (Worst)、平均値 (Average) において提案突然変異法を用いた場合が良い結果となることが確認できる。また、計算時間 (Time) もほとんど変わらないことから、提案突然変異法は十分効果があると判断できる。

次の実験は、局所探索の効果を確認するためのものである。実験結果を表 2 に示す。表中の “Random”、“None” は、それぞれ、局所探索を行う際、対象となる遺伝子をランダムに選択する方法、局所探索を行わない方法を意味する。表 2 から、やはり最悪値 (Worst)、平均値 (Average) において提案手法を用いた場合が良い結果となっており、十分効果があることが確認できる。

3 番目の実験は、提案する解の探索範囲調節の効果を確認するためのものである。実験結果を表 3 に示す。表中の “None” は、解の探索範囲調節を行わないことを意味する。

表 1 突然変異の有効性

Table 1. The efficiency of mutation.

Mutation	Solution			
	Best	Worst	Average	Time(s)
ATGA	183946	183936	183942.3	279.2
None	183946	183931	183941.7	276.1

表 2 局所探索の有効性

Table 2. The efficiency of local search.

	Solution			
	Best	Worst	Average	Time(s)
ATGA	183946	183936	183942.3	279.2
Random	183946	183931	183939.8	291.3
None	183931	183910	183922.3	268.9

表 3 解の探索範囲調整の有効性

Table 3. The efficiency of the adjustment of solution space.

	Solution			
	Best	Worst	Average	Time(s)
ATGA	183946	183936	183942.3	279.2
None	183875	183536	183686.6	8212.9

表 4 総当り法との比較

Table 4. The comparison with the enumeration method.

Algorithm	Problem size				
	10	20	25	30	35
Enumeration method	7110	8793	9683	10519	12258
Time(s)	0.015	15.31	489.9	7712	268870
ATGA	7110	8793	9683	10519	12258
Time(s)	0.0001	0.078	0.15	0.19	0.62

表 3 から、最良値 (Best), 最悪値 (Worst), 平均値 (Average) の全てにおいて提案手法が良い結果となっていることが確認できる。更に、計算時間 (Time) が約 1/30 と大幅に減少していることから、解の探索範囲調節は最も効果があると判断できる。

最後の実験は、提案手法により得られる解の信頼度を検証するため、確実に最適解が求まる総当り法との比較を行う。実験には、荷物の価値と重量をランダムに生成した 5 つのテスト問題を使用する。ただし、荷物数に関しては総当り法で解が求まる範囲のものを使用した。この実験結果を表 4 に示す。この結果から、提案手法は、荷物数 (遺伝子数) が 35 までの全ての問題で最適解を得ることができる信頼性に加えて十分な収束速度をもつことが確認できる。今回の実験環境では、総当り法は、荷物が 35 を越えると実行時間で解くことが困難と判断した。

4.2 既存手法との比較 ここでは、総当り法では実行時間内に解くことが困難なサイズの大規模問題を取り上げ、すでに提案されている各手法との比較実験を行う。

既存の比較対象手法としては、動的計画法 (DP)⁽⁷⁾、貪欲解法 (Greedy)⁽¹⁾、単純 GA (SGA)⁽³⁾、中谷らの手法 (VEA)⁽⁸⁾、シミュレーテッド・アニーリング (SA)⁽⁷⁾ の 5 手法を取り上げる。ただし、ここでの SA⁽⁷⁾ は初期解の獲得に貪欲解法を用いている。

各手法のパラメータ設定等の条件は前項の実験と同様とし、荷物の数を 100 個とするモデル 1、500 個とするモデル 2 について実験を行った。ただし、VEA では子個体の生成を行わないため集団サイズを 1000 とし、その他の手法は集団サイズを 500 としている。各手法の具体的なパラメータを表 5 に示す。

また、モデル 1、2 の単一目的問題で使った問題の全荷物に対する起因レベルを求める過程で、各問題の安定遺伝子と不安定遺伝子の含有量について調査したものをそれぞれ表 6、表 7 に示す。これらの結果からも明らかなように、不安定遺伝子が含まれる範囲を重点的に探索する提案

表 5 各手法のパラメータ

Table 5. Parameters of each method.

	SGA	VEA	ATGA	SA
Population size	500	1000	500	
Crossover rate	0.70	0.75	0.75	
Mutation rate	0.05	0.05	0.10	
Initial temperature				100
Cooling parameter				0.90

表 6 最良解に含まれる安定遺伝子と不安定遺伝子 (モデル 1)

Table 6. The stable and unstable genes content of the best solution (model 1).

	The stable gene	The unstable gene	
Causal level R_i	$0 \leq R_i \leq 1$	$1 < R_i \leq 2$	$2 < R_i$
Content (%)	59.0	12.0	29.0
Total content (%)	59.0	41.0	

表 7 最良解に含まれる安定遺伝子と不安定遺伝子 (モデル 2)

Table 7. The stable and unstable genes content of the best solution (model 2).

	The stable gene	The unstable gene	
Causal level R_i	$0 \leq R_i \leq 1$	$1 < R_i \leq 2$	$2 < R_i$
Content (%)	50.6	15.6	33.8
Total content (%)	50.6	49.4	

手法による解の探索範囲は通常の半分以下に抑えられ、計算時間の短縮に十分貢献すると考えられる。

実験は、各モデルごとに、それぞれ単一目的問題と多目的問題を取り上げ、比較検討を行う。なお、多目的問題の実験は、目的関数を次式を用いて単一目的に変換することにより、実現している⁽⁴⁾。

$$v = \sum_{k=1}^p w_k v_k \dots \dots \dots (10)$$

ただし、 w_k は次式で与えられる重みとする。

$$\sum_{k=1}^p w_k = 1, \quad w_k \geq 0 \dots \dots \dots (11)$$

多目的問題の実験では、単一目的問題に目的関数を 1 つ追加した 2 目的モデルを使用し、目的関数 1 の重み w_1 を 1 から 0.11 刻みに 0 まで減らすことで、10 種類のパレート最適解を求めることを試みている。

モデル 1 での単一目的問題、多目的問題の各手法による実行結果を表 8、図 3 に示す。また、モデル 2 の実行結果をそれぞれ表 9、図 4 に示す。

表 8 より、モデル 1 では SGA、貪欲解法以外の手法で動的計画法により得られた値 (最適解) と同様の結果が得られることを確認できる。特に、提案 ATGA ですべて動的計画法と同じ解が得られたことは特筆すべきである。また、図 3 の多目的問題の実行結果においてもそれぞれ類似したパレート最適解を確認することができたが、やはり提

表 8 単一目的問題の実験結果 (モデル 1)

Table 8. The result of the single-objective problem (model 1).

Algorithm	Solution			
	Best	Worst	Average	Time(s)
DP	36198	-	-	0.5
Greedy	36156	-	-	0.00074
SGA	36131	35417	35881.5	102.5
VEA	36198	36090	36147.5	80.6
SA	36198	36156	36178.0	0.9
ATGA	36198	36198	36198.0	12.7

表 9 単一目的問題の実験結果 (モデル 2)

Table 9. The result of the single-objective problem (model 2).

Algorithm	Solution			
	Best	Worst	Average	Time(s)
DP	183946	-	-	32791.8
Greedy	183865	-	-	0.0013
SGA	177327	156789	161527.5	10535.5
VEA	166430	153508	158262.1	5514.7
SA	183931	183865	183900.0	1.1
ATGA	183946	183936	183942.3	279.2

図 3 多目的問題の実験結果 (モデル 1)

Fig.3. The result of the multi-objective problem (model 1).

図 4 多目的問題の実験結果 (モデル 2)

Fig.4. The result of the multi-objective problem (model 2).

案 ATGA の優秀さを窺うことができる。

次に、より大規模な問題を扱ったモデル 2 の実験結果では、SGA や、モデル 1 において性能の良い結果を確認できた VEA で良い結果を得ることができなかったのに対し、提案 ATGA はより良い結果を得ることができた。特に、提案 ATGA は動的計画法と比較して収束速度が速く、解の精度も動的計画法とほとんど変わらない優れた結果を高速に得ている。

計算時間だけを見ると、SA が格段に速く、解の精度もかなり高いが、SA の場合パラメータ設定の困難性という問題が常につきまとうことになる。今回の実験でも最適と思われるパラメータ値を見つけるまでにかなりの時間を要した。これに対して、提案 ATGA はパラメータ設定も容易で、解の精度も SA と比較して高く、しかも計算時間も実用レベルにあることから、大規模な問題に対して有効な手法であると言えるだろう。

また図 4 より、SGA、VEA は一部パレート最適解を得ることができなかったのに対し、提案 ATGA では 10 個のパレート最適解を求めることができた。しかもそのほとんどが動的計画法で得られた解と一致している。このことから、提案 ATGA の有効性は明らかであろう。

また、同一条件でランダムに生成した別の 10 問題に対す

る数値実験においても同様な傾向が見られたことから、この種の問題に対する提案手法のロバスト性も十分あると思われる。

5. まとめ

本論文では、組合せ最適化問題の解法として有効とされる GA に対し、貪欲解法の考え方を適用することで問題の解空間を大幅に縮小させるとともに、各荷物の安定性を考慮して探索範囲を徐々に拡大させる範囲調節型 GA (ATGA) を提案した。更に、提案手法を 0-1 ナップサック問題に適用することでその有効性を検証した。

今回の実験から、VEA は大規模問題に対する比較対象としては適当でない可能性がある。VEA に限らず、大規模問題を不得意とする手法は多々あり、動的計画法もその一つと考えてよいだろう。動的計画法は、通常最適解が得られると言われているが、大規模問題や複雑問題に対して解を得ることが困難な場合があり、しかも問題のタイプによってはかなりの時間を要することもある手法と言える。

計算時間という観点からは、提案手法と同じ貪欲解法を導入した SA の高速性が目につくが、パラメータ設定が非常に難しく、今回の実験でも最適と思われるパラメータ値を見つけるまでにかなりの時間を要した。それでも動的計

画法で得られた解を得ることができなかった点に注目すべきである。

これに対して、提案手法はパラメータ設定も容易で、解の精度も動的計画法と同等であり、しかも計算時間が実用レベルにあることから、動的計画法では実用時間内で解くことが困難なより大規模サイズの問題に対しても有効な手法であると言えるだろう。このことは、多目的問題に対する実験でも証明されている。

しかし、今回実験的に設計した起因レベルは、試行錯誤的に設定した部分も多く、問題の特性によっては正常に作用しないことも考えられるため、一般性に欠ける部分もある。今後、どのようなタイプの問題に対しても正常に動作できるような高い汎用性をもつように改良する必要があると思われる。

(平成 15 年 10 月 27 日受付, 平成 16 年 4 月 5 日再受付)

文 献

- (1) 林 芳男:「0-1 ナップザック問題の数理とアルゴリズム」, 近畿大学商経学会, pp.5-8 (2000)
- (2) G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi: Complexity and Approximation, Springer-Verlag, pp.40-43 (1999)
- (3) 北野宏明: 遺伝的アルゴリズム 1, 産業図書, pp.5-16 (1993)
- (4) 北野宏明: 遺伝的アルゴリズム 2, 産業図書, pp.71-86 (1995)
- (5) 井田憲一・菅 良平・玄 光男:「遺伝的アルゴリズムによる多目的ナップザック問題の一解法」, 平 15 電気学会電子情報システム部門大会講演論文集, pp.370-372 (2003)
- (6) R.E. Steuer: Multiple Criteria Optimization: Theory, Computation, and Application, John Wiley & Sons, pp.138-183 (1986)
- (7) 加瀬正喜:「ナップザック問題のメタ戦略に関する実験的性能分析」, <http://tdweb.am.chs.nihon-u.ac.jp/semi4/2001/kase.htm>, 日本大学文理学部平 13 卒業研究 (2001)
- (8) N. Nakaya, A. Kanasugi, and K. Kondo: "An Evolutionary Algorithm based on the Virus Theory of Evolution", *J. IPSJ*, Vol.40, No.5, pp.2346-2355 (1999-5) (in Japanese)
中谷直司・金杉昭徳・近藤邦雄:「ウイルス進化論に基づく進化型アルゴリズム」, 情報処理学論誌, 40, 5, pp.2346-2355 (1999-5)

井 田 憲 一 (正員) 1952 年生。1976 年工学院大学工学部電気工学科卒業。1978 年同大学大学院工学研究科電気工学専攻修士課程修了。同年足利工業大学工学部経営工学科 (現, 経営情報工学科) 助手, 1985 年同大学講師, 1990 年同大学助教授, 1996 年同大学教授。1999 年前橋工科大学工学部情報工学科教授, 現在に至る。工学博士。1988 ~ 1899 年米国 Univ. of Nebraska-Lincoln 客員研究員。システム最適化法, 数値計算法, ファジィ数理計画法, ソフトコンピューティング技法等の研究・開発に従事。「Mathematica ニューラルネットワークと最適化」, 共立出版 (1998) 共著他。電子情報通信学会, 日本知能情報ファジィ学会, 日本オペレーションズ・リサーチ学会, 日本経営工学会, 情報処理学会, IIE 等各会員

菅 良 平 (学生員) 1981 年生。2004 年前橋工科大学工学部情報工学科卒業。同年早稲田大学大学院情報生産システム研究科修士課程入学, 現在に至る。組合せ最適化, ソフトコンピューティング技法等の研究・開発に従事。

玄 光 男 (正員) 1944 年生。1969 年工学院大学工学部電子工学科卒業。1974 年同大学大学院工学研究科電気工学専攻博士課程修了。1974 年足利工業大学工学部経営工学科 (現, 経営情報工学科) 講師, 1982 年同大学助教授, 1987 年同大学教授。2003 年早稲田大学大学院教授, 現在に至る。工学博士。1982 ~ 1983 年米国 Univ. of Nebraska-Lincoln 客員研究員。遺伝的アルゴリズム, ニューラルネットワーク, ファジィ理論とそれらのネットワーク設計, スケジューリング, システム信頼性設計等への応用に関する研究に従事。「Genetic Algorithm & Engineering Optimization」, John Wiley & Sons (1999) 共著他。「Comp. & Industrial Engineering」誌 Area Editor。電子情報通信学会, 日本知能情報ファジィ学会, 日本経営工学会, 日本信頼性学会, IIE, IEEE 等各会員