ナップサック問題における Discrete Particle Swarm Optimizationの有効性

Empirical Study of Discrete Particle Swarm Optimization for the knapsack problem

江本久雄	中村秀明	河村圭	宮本文穂
Hisao Emoto	Hideaki Nakamura	Kei Kawamura	Ayaho Miyamoto
山口大学大学院	山口大学	山口大学	山口大学
Yamaguchi Univ.	Yamaguchi Univ.	Yamaguchi Univ.	Yamaguchi Univ.

Abstract Particle swarm optimization (PSO) is a novel multi agent optimization system inspired by social behavior of bird flocking or fish schooling. In PSO, instead of using more traditional genetic operators, each particle (individual) adjusts its "flying" according to its own flying experience and its companions flying experience. This paper will compare the performance of the discrete PSO and Genetic Algorithms on sets of combinatorial problems. Moreover, the experimental results indicate that the PSO is a promising optimization method for combinatorial problems and the PSO always converges very quickly towards the optimal positions.

1. はじめに

PSO(Particle Swarm Optimization)[1] は，1995 年に Kenedy と Eberhart によって，鳥や魚の個々の行動と群れの社会的行動の研究を工学的に応用した確率的な最適化技術である．初期の PSO は，連続関数の最適化問題のために開発され，連続値の設計変数に対して非線形性の強い関数でも最適化が可能であり，GAs(Genetic Algorithms) と比べて，そのアルゴリズムの簡単さや収束性の良さのため注目を集めてきた．現実的な工学問題の多くは，例えば，構造最適設計などの分野では，部材材料，部材形状，部材本数など，その設計変数は離散値を取り扱うため，組合せ最適化問題となる．さらに，近い将来，構造設計において，デザイン，景観，環境への影響度など連続値では表すことのできない設計変数を取り入れて最適化設計を行う可能性がある．このように，最適化計算は，効率性や論理性などを考慮するためにも必ず行われるため，離散値の組合せ最適化問題が重要となる．PSO において，Kenedy と Eberhart らは，離散値を取り扱うことのできるように PSO の拡張を行い，DPSO(Discrete Particle SwarmOptimization) を提案している [2, 3] ．しかし，解探索能力が不明であるので，本研究では，組合せ最適化問題としてナップサック問題を利用して，荷物数つまり次元数を増やすことで探索能力を調査し，繰り返し計算回数についても検討を行う．また，GA と比較することで DPSO の有効性を示し，DPSO のアルゴリズムの特徴を示す．

2. DPSO の概要

PSO では，解候補を粒子と呼び，すべての粒子は，設計変数となる位置と，探索方向を決定するための速度をもっている．また，それぞれの粒子は，解探索の最良点の履歴を，群れ全体としては，すべての粒子群の中での最良点を保持している．通常の連続値を扱う PSO との大きな違いは，粒子位置の更新式が大きく異なる点である．以下に DPSO の処理手順を述べる．

手順 1：[初期群の生成]

ランダムに粒子の位置と速度を初期化する．

手順 2：[目的関数の計算]

各粒子に対して目的関数を計算する．

手順 3：[個々の粒子の最良位置の保存]

個々の粒子について，最初から現在までの繰返し計算回数の中での最良値を記憶する．これを $pBest_i$ と呼ぶ．i は，i 番目の粒子を表す．

手順 4：[個体群全体での最良位置の保存]

すべての粒子群の中で，最初から現在までの繰返し計算回数の最良値を記憶する．これを $gBest$ と呼ぶ．

手順 5：[粒子速度の計算]

次式によって，粒子速度を計算する．

$$v_i^{k+1} = v_i^k + r1 \times c1 \times (pBest_i - x_i^k) + r2 \times c2 \times (gBest - x_i^k) \quad (1)$$

ここに，v_i^{k+1} は次世代の i 番目の粒子の速度，$r1, r2$ は 0 から 1 の一様乱数，$c1, c2$ は 0 から 2 の範囲の学習係数である．また，粒子の速度は，最大速度 V_{max} によって制限される．

手順 6：[粒子位置の更新]

すべての粒子に対して次式によって位置を更新する．

$$\rho_i^{k+1} < sig(v_i^{k+1}) \text{ then } x_i^{k+1} = 1; \text{ else } x_i^{k+1} = 0 \quad (2)$$

$$sig(v_i^{k+1}) = \frac{1}{1 + exp(-v_i^{k+1})} \quad (3)$$

ここで，ρ_i^{k+1} は，0 から 1 までの乱数値である．

手順 7：[終了判定]

最大繰返し回数に達したか，または，十分に収束したかを判定する．条件を満たさない場合は，手順 2 から手順 6 を繰り返す．

式 (1) のように，それぞれの粒子は，これまでの経験の中での最良点とすべての粒子群中の最良点と現在の位置とのベクトル和によって粒子速度を求める．さらに，離散値問題の場合は，次の位置を決めるために，式 (2) のように確率的な閾値によって決められる．例えば，もし v_i^{k+1} が，閾値より高い値ならば，粒子は 1 になり，そうでなければ，0 になる．そのため，閾値が 0 から 1 の範囲で必要となるため，式 (3) に示すようなシグモイド関数を利用する．

3. シミュレーション

3.1. ナップサック問題の定式化

一般に，荷物 i (i=1,$\cdots$,N) の重量および価値をそれぞれ $a_i > 0$ および $c_i > 0$，袋の許容重量を $b > 0$ とすると，「袋の許容重量内で価値を最大にする荷物を選ぶ」というナップサック問題は以下のように定式化される．

$$\begin{aligned} &maximize \sum_{i=1}^{N} c_i x_i \\ &subject\ to \sum_{i=1}^{N} a_i x_i \leqq b \\ &x_i \in \{0, 1\} \end{aligned} \quad (4)$$

ただし、決定変数 x_i については、荷物 i を袋に入れることを $x_i = 1$，入れないことを 0 で表すものとする．

3.2. シミュレーション条件

本研究では，DPSO の解探索能力を明らかにするために，荷物数（次元数）と粒子数，また，繰り返し回数について着目してシミュレーションを行う．また，進化計算手法の GA と比較することで DPSO の特徴を明らかにする．以下にシミュレーション条件をまとめる．

3.2.1. シミュレーション 1

荷物数と粒子数が解探索能力に与える影響について検討する．荷物数と粒子数の検討を行うため，30 荷物，50 荷物，100 荷物に対してそれぞれ，粒子数が 30 個，50 個，100 個，150 個，200 個とする．さらに，荷物数が増加することで，粒子数の探索空間における割合が異なるので，荷物数が 50 の時は，粒子数を 350，500 と 800，荷物数が 100 の時は，粒子数を 700 と 800 についても検討した．また，GA についても同じ荷物数で計算を行う．ここで，DPSO の粒子数は，GA では個体数に対応する．

3.2.2. シミュレーション 2

繰り返し回数が解探索能力に与える影響について検討する．荷物数を 50，粒子数を 200 と設定し，繰り返し回数に関しては，最大 500 回まで検証を行う．GA についても同様に計算を行う．

GA のパラメータは，通常よく用いられると思われるルーレット選択，2 点交叉，エリート保存戦略とする．また，突然変異率は 1/遺伝子長とする．GA は問題により種々のパラメータの影響が大きいため，GA パラメータの比較検討が必要であるが，ここでは，DPSO と比較することが目的なので，一般的と思われるパラメータを採用する．また，DPSO,GA ともに確率的な手法によって最適解を探索し，集団を進化させているので，それぞれのシミュレーションを 1,000 回繰り返した．

3.3. 結果と考察

3.3.1. 粒子数と荷物数の影響

シミュレーション 1 から，粒子数と荷物数について考察を行う．1,000 試行回数中に最適解の求まった割合について，それぞれ荷物数が 30，50，100 について図 1，図 2，図 3 に示す．また，100 荷物の時に得られた最適解の度数分布を，それぞれ粒子数が 200 と 800 について図 5 と図 6 に示す．まず，DPSO について考察を行う．図 1，図 2，図 3 から荷物数と粒子の数が適切でないと最適解が得られにくいことが分かる．つまり，ほぼ確実（9 割以上）に最適解を求めるためには，次元数と粒子数を適切に設定する必要がある．30 荷物の場合，粒子数が，100 程度は必要であり，50 荷物の場合，800 程度は必要であることが分かる．100 荷物の場合，粒子数が 800 以上必要であるが，プログラム上メモリが確保できず計算ができないため，評価できなかった．ここで，解空間中，つまり荷物の組合せ数の中で探索点数（粒子数，個体数）の占める割合を図 4 に示す．図 4 から，荷物数が増加すると探索点数の占める割合が少なくなることが分かり，100 荷物の場合の適切な粒子数 (100) は，30 荷物の場合の適切と思われる粒子数の解空間中に占める割合と等しくすると，粒子数が 100×10^{20} となり計算不可能となる．そこで，図 5，図 6 に示すように荷物数 100 の場合に得られる最適解のバラツキについて検討する．

図 5，図 6 は，それぞれ粒子数が 200，800 の場合について示している．図 5 から粒子数が 200 では，粒子数と最適解の関係で明らかに，十分な粒子数ではないが，全く最適解が得られないわけでない．しかし，図 5 から求まる解の山が最適解よりも低い解を多く得ていること，また，最適解に近いものの解が広範囲にばらついていることが分かる．図 6 から粒子数が 800 の場合は，最適解が粒子数 200 のときよりも，多く得られたことが分かり，解のバラツキが少なくなることが分かる．つまり，計算コストや計算可能性を考慮すると，100 荷物の場合，粒子数が 800 程度あれば，比較的精度良く最適化を得ることは可能である．

次に，GA との比較を行う．図 1 から GA によっても最適解が求まるが，その精度は，DPSO に比べかなり低く，図 2，図 3 のように荷物数が多くなると，GA では，最適解が得られにくいことが分かる．ここで，荷物数 50，個体数 200 の時の度数分布表を図 7 に示す．図 7 から GA では準最適解は求まっていることが分かる．

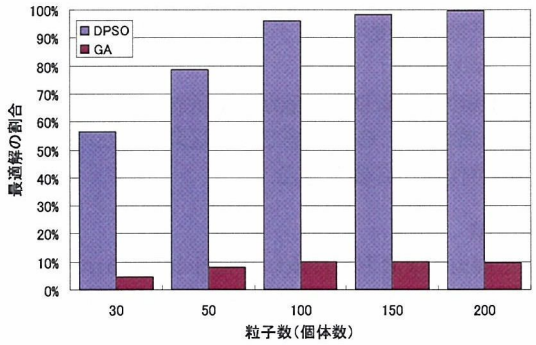

図 1: 30 荷物のときの最適解の割合

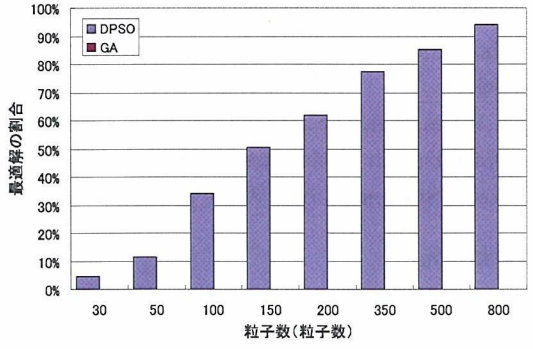

図 2: 50 荷物のときの最適解の割合

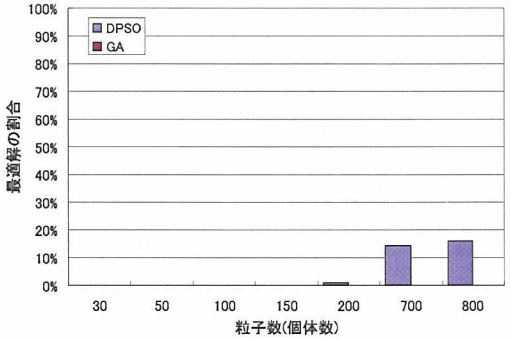

図 3: 100 荷物のときの最適解の割合

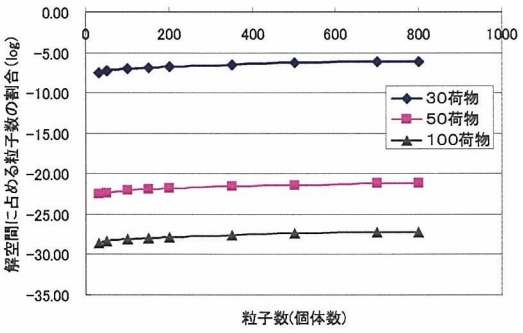

図 4: 探索点数の解空間に占める割合

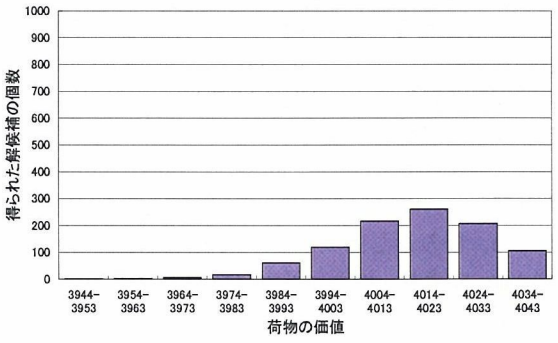

図 5: 荷物数 100，粒子数 200 のときの度数分布

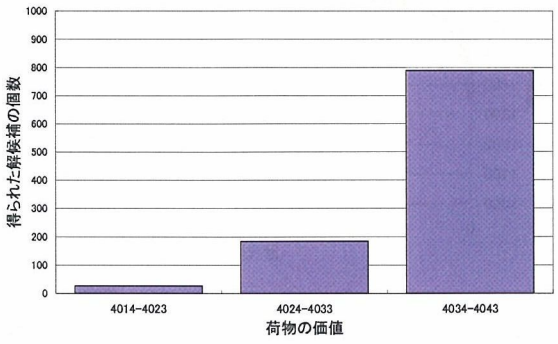

図 6: 荷物数 100，粒子数 800 のときの度数分布

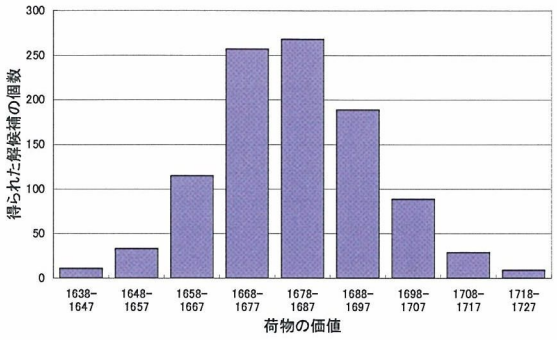

図 7: GA による 100 荷物のときの最適解の割合

3.3.2. 繰り返し回数の影響

シミュレーション2から，繰り返し回数が探索能力に与える影響について考察する．図8に50荷物，200粒子 (個体) の時の適応度の推移について示す．これは，1000試行回数の平均を示している．まず，DPSOに関して，図8から繰り返し回数が100以降では，適応度の推移がフラットになり変化があまりない．また，図9に荷物数50，粒子数200の時の得られた最適解の割合を示す．図9から，繰り返し回数が50の時は，明らかに進化の途中であることが分かり，繰り返し回数が100程度から最適解の求まる割合が高くなるので，計算回数を考慮すると，繰り返し回数が100から，より確実に求まる200程度までが適当であることが分かる．次に，GAに関して，図8からGAでは世代数を重ねるごとに最良な解が更新されていることが分かる．これは，GAの進化方法が集団で交叉や突然変異を繰り返すことによって進化するため，十分な世代数を必要とするためと思われる．一方，DPSOでは，個々の粒子の過去の最良点と粒子群全体の最良点と現在の方向に向かっていく，単純な進化方法であるため，繰り返し回数が100前後で最適解を発見している．つまり，単純なアルゴリズムのため比較的早い段階で収束することが分かる．

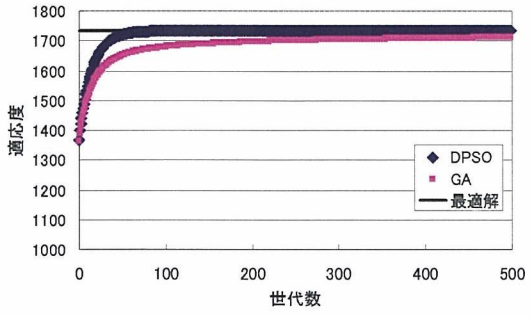

図 8: 50 荷物，200 粒子 (個体) の時の適応度の推移

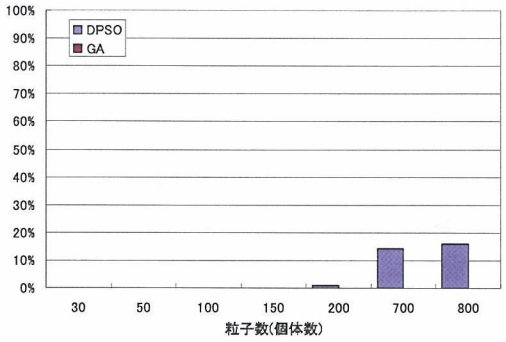

図 9: DPSO による最適解の割合

4. まとめ

- 次元数により粒子数を適切に設定する必要がある．
- 50 次元程度までならば，粒子数を適切に設定することで最適化に利用できる．
- 繰り返し回数は，100 回から 200 回程度が適当である．
- 繰り返し回数を 200 回より多くすると，より確実に最適解を得やすくなるがその効果と計算コストを考えると無駄な計算となる．
- DPSO は，GA よりも，荷物数が多くなっても，粒子数を適度に設定することで最適解を得ることができる．
- DPSO は，繰り返し数が 100 前後で収束する．それに対して，GA 手法では，収束するために十分な世代数が必要となる．

参考文献

[1] J.kennedy,R.Eberhart, Particle Swarm Optimization, Proc. of IEEE International Conference on Neural Networks(ICNN'95), Vol.IV, pp.1942-1948, Perth, Australia, 1995.

[2] J.kennedy,R.Eberhart, A discrete binary version of the particle swarm optimization algorithm, Proc. of the 1997 conference on System, Man, and Cybernetics (SMC'97),pp.4104-4109, 1997.

[3] K.Y.Lee,M.A.El-Sharkawi, Mordern Heuristic Optimization Techniques with Application to Power Systems, John Wiley & Sons,July 2004.

問い合わせ先

〒 755-8611
宇部市常盤台２丁目１６－１
山口大学工学部知能情報システム工学科
システム設計工学研究室
TEL: 0836-85-9530 FAX: 0836-85-9530
E-mail: emoto@design.csse.yamaguchi-u.ac.jp