ナップサック問題におけるDiscrete PSO のパラメータの検討

Empirical Study of Discrete Particle Swarm Optimization for the knapsack problem

江本久雄*・中村秀明**・別府万寿博***・河村圭****　・宮本文穂*****
Hisao EMOTO, Hideaki NAKAMURA, Masuhiro BEPPU, Kei KAWAMURA and Ayaho MIYAMOTO

*修(工)　山口大学大学院博士後期課程　理工学研究科システム工学専攻（〒755-8611 宇部市常盤台 2-16-1）
**博(工)　山口大学助教授　工学部知能情報システム工学科（〒755-8611 宇部市常盤台 2-16-1）
***博(工)　防衛大学校助手　建設環境工学科（〒239-8686 神奈川県横須賀市走水 1-10-20）
****博(工)　山口大学助手　工学部知能情報システム工学科
*****工博　山口大学教授　工学部知能情報システム工学科

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.

Key Words: *Discrete Particle Swarm Optimization*, *Knapsack Problem*, *Parameter selection*

1　はじめに

PSO(Particle Swarm Optimization)[1)]は，1995 年に Kennedy 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．PSO において，Kennedy 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 Swarm Optimization)[2)]を提案している．

DPSO を構造分野に適応するにあたり，基本的なアルゴリズムとその特性を把握する必要がある．土木構造物の最適設計は組合せ問題となることが多いため，ナップサック問題を用いて検証を行う．そのため，一般的な DPSO のアルゴリズムの概要を説明するとともに，NP(Non-deterministic Polynomial)困難な代表的例題であるナップサック問題のシミュレーションを通して，DPSO のパラメータの特徴を明らかにする．とくに，DPSO の解探索能力を明らかにするために，荷物数(次元数)と粒子数が与える影響について，SGA(Simple GA)と比較しながら検討する．

2　DPSO のアルゴリズムの概要

DPSO では，探索点を粒子と呼び，すべての粒子は位置情報(座標値)と，探索方向を決定するための速度をもっている．位置情報(座標値)により設計変数を表す．また，個々の粒子は探索過程における過去の最良位置の情報 ($pBest_i$) をもち，また，群全体として，すべての粒子の中での最良の位置情報 ($gBest$) を保持している．

PSO による最適化では，現在の位置情報や $pBest_i$, $gBest$ を用いて，解空間を探索する．探索点(粒子)の更新方法は，**図-1** に示すような簡単なベクトルの合成和となる．ここで，x_1^k，x_2^k は探索点(粒子)の位置情報，v_1^k，v_2^k は速度ベクトル，添え字 k はステップ，w は粒子の慣性，c_1, c_2 は学習係数，さらに，r_1, r_2 は [0, 1] の一様乱数を示す．なお，計算の 1 ステップあたりの移動時間 Δt は単位時間(Δt = 1)を考える．

例えば，x_1^{k+1} は粒子 1 の前回の速度ベクトル (v_1^k)，群全体の最良解への速度ベクトル $(gBest - x_1^k)$ とその粒子自身の最良解への速度ベクトル $(pBest_1 - x_1^k)$ の線形結合によって求まる．ここで，群全体への速度ベクトル $(gBest - x_1^k)$ は大域探索を，粒子自身の速度ベクトル

$(pBest_1 - x_1^k)$は局所探索を意味する．一般的に，i 番目の粒子の k+1 回目の速度ベクトルは，次式のように求められる．

$$v_i^{k+1} = wv_i^k + \frac{r_1 c_1 (pBest_i - x_i^k)}{\Delta t} + \frac{r_2 c_2 (gBest - x_i^k)}{\Delta t} \qquad (1)$$

また，探索点の位置は，次式で表される．

$$x_i^{k+1} = x_i^k + v_i^{k+1} \Delta t \qquad (2)$$

ここで，式(1)，(2)中には計算の 1 ステップあたりの移動時間が含まれているが，Δt は単位時間を考えているので，$\Delta t = 1$ とする．

DPSO の処理手順を**図−2** に示す．離散値を取り扱う DPSO と PSO との主な違いを以下に記述するが，探索点である粒子位置の更新を行う処理が式(2)に示すようなベクトル和でなくシグモイド関数を用いて 0 と 1 に区別し，離散化する点である．以下，計算手順について説明する．

Step 1：[初期群の生成]
初期の粒子位置と速度をランダムに決める．
Step 2：[評価値の計算]
各粒子の座標からそれぞれの粒子の評価値を計算する．
Step 3：[個々の粒子の最良位置の保存]
個々の粒子について，これまでに移動してきた軌跡の中での最良の評価値を記憶する．これを $pBest_i$ と呼ぶ．ここで i は，i 番目の粒子を表す．
Step 4：[個体群全体での最良位置の保存]
すべての粒子の中での，最良の評価値($pBest_i$の中で一番良いもの)を記憶する．これを $gBest$ と呼ぶ．
Step 5：[粒子速度の計算]
式(1)によって，各々の粒子速度を計算する．
Step 6：[粒子位置の更新]
すべての粒子に対して式(3)および式(4)によって位置を更新する．

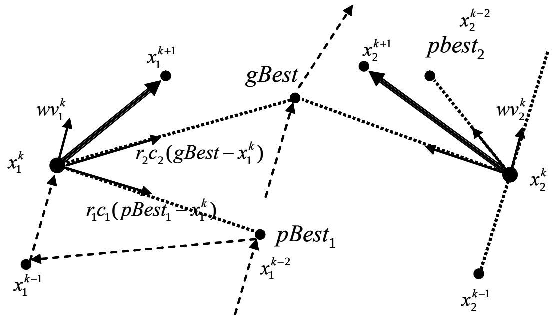
$$\text{if} \quad \rho < sig(v_i^{k+1}) \quad \text{then} \quad x_i^{k+1} = 1;$$

$$\text{else} \quad x_i^{k+1} = 0 \qquad (3)$$

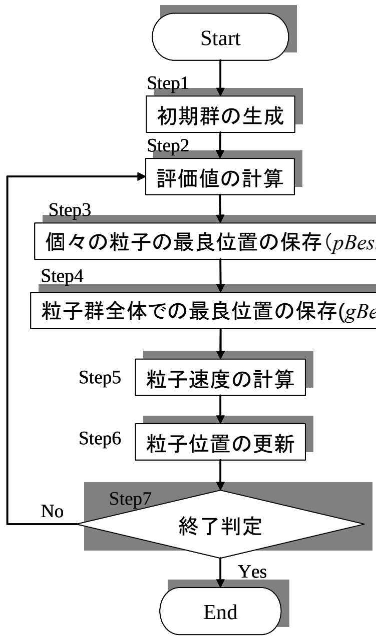
$$sig\left(v_i^{k+1}\right) = \frac{1}{1 + \exp(-v_i^{k+1})} \qquad (4)$$

ここで，ρ は 0 から 1 までの一様乱数，$sig(v_i^{k+1})$ はシグモイド関数である．

この操作によって連続量が離散値となり PSO と大きく異なっている点である．すなわち，DPSO の位置更新は，



図−1 移動する粒子の模式図



図−2 DPSO の処理手順

式(3)のように確率的な閾値 ρ によって決められる．例えば，もし $sig(v_i^{k+1})$ が，閾値 ρ 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．閾値 ρ は 0 から 1 の範囲であるため，本研究では式(4)に示すようなシグモイド関数を利用している．
Step 7：[終了判定]
繰り返し回数に達しない場合は，Step2 から Step6 を繰り返す．

以上が計算手順であるが，実際の設計問題に適用する際には，設計変数を 2 進数にコーディングし，粒子の位置情報と対応させる必要がある．

3　ナップサック問題のシミュレーション

DPSO のパラメータの特徴，探索能力を把握するため，ナップサック問題でのシミュレーションを行った．ナップサック問題とは，荷物 i (i = 1,…, N)の重量および価値をそれぞれ a_i と c_i，さらに，袋の許容重量を b とすると，「袋の許容重量 g 内で価値の総量 f を最大にする荷物を選ぶ」と定義され，以下のように定式化される．

$$\text{maximize} \quad f = \sum_{i=1}^{N} c_i x_i \tag{5a}$$

$$\text{subject to} \quad g = \sum_{i=1}^{N} a_i x_i \le b \tag{5b}$$

$$x_i \in \{0,1\}$$

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 $x_i = 0$ で表すものとする．

荷物の重量 a_i と価値 c_i は, [1,100]の一様乱数で作成し, 許容重量 b は重量の総和の半分とした．以下，価値の総量 f を適応度と呼ぶ.

3.1 パラメータの比較，検討

一般に，DPSO のパラメータは，c_1，c_2 ともに 2.0, w は 0.9 が良いとされている [3)]が，その根拠は明確ではない．そこで, これらのパラメータが解探索能力に与える影響を調べる.

図-3 には，荷物数を 50，c_1，c_2 ともに 2.0 と設定し，w を変化させた場合の適応度を示す．これより，w の値を大きくするにしたがい適応度は向上し，w = 1.0 のときに最適解を得ていることがわかる．しかし，w が 1.0 を超えると適応度は急激に低下している．これは，w は粒子の慣性を表しているため，w が 1.0 よりも過小あるいは過大な場合は，粒子が群から取り残されたり，あるいは先行しすぎる結果となり，集団としての行動が行えず，結果的に群としても良い位置を持続できないものと考えられる．また，荷物数を 30，100 に変更した場合も同様な傾向を示した．

図-4 には，c_1 を 2.0，荷物数を 50，繰り返し回数を 100 回と設定した場合に，c_2 が適応度に与える影響を示す．なお, 同図には適応度の最大値付近の拡大図も示している．学習係数 c_2 は，大域探索へ影響を与える係数であり，群全体の情報を反映する程度を表す．したがって，c_2 が 0 の場合は粒子が個別に探索するだけで, 群全体の情報が不足しているので解探索の効率が悪く, 最も緩やかな曲線となっている．一方，c_2 が 4.0 の場合，*gBest* 付近の局所探索となってしまうために，**図-4** に示すように群全体が局所解（1,722）へと急激に収束してしまい最適解が得られなかった．c_2 が 0.50～2.0 のときは，上記の中間的な挙動を示した．すなわち，DPSO では，3 つの速度ベクトルの適切な協調により, 最適解が得られやすいものと考えられる．ちなみに，この問題では，c_2 が 0.75 のときに最適解が得られた.

図-5 には，c_2 を 0.75，荷物数を 50，繰り返し回数を 100 回と設定し，c_1 を 0～4 まで 1.00 刻みで変化させた場合の適応度の推移を示す．c_1 は，各々の粒子の最良値が影響を与える係数であり, 自己の経験を重視するかどうかの影響度を表す．c_1 が 0 の場合，探索に影響を与える要因は，群全体の最良の値を保持している *gBest* のみであるため，最適解が得られなかったものと思われる．一方，c_1 が 1～

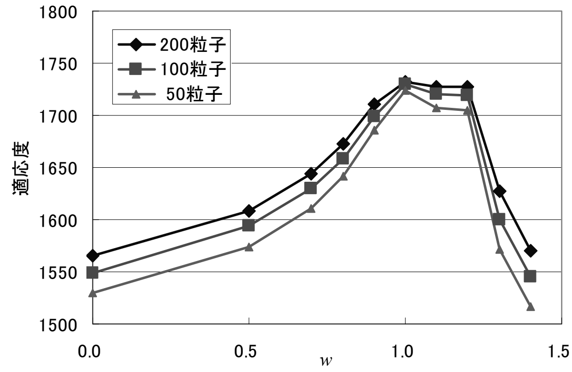


図-3 慣性の重み w が最適解に与える影響

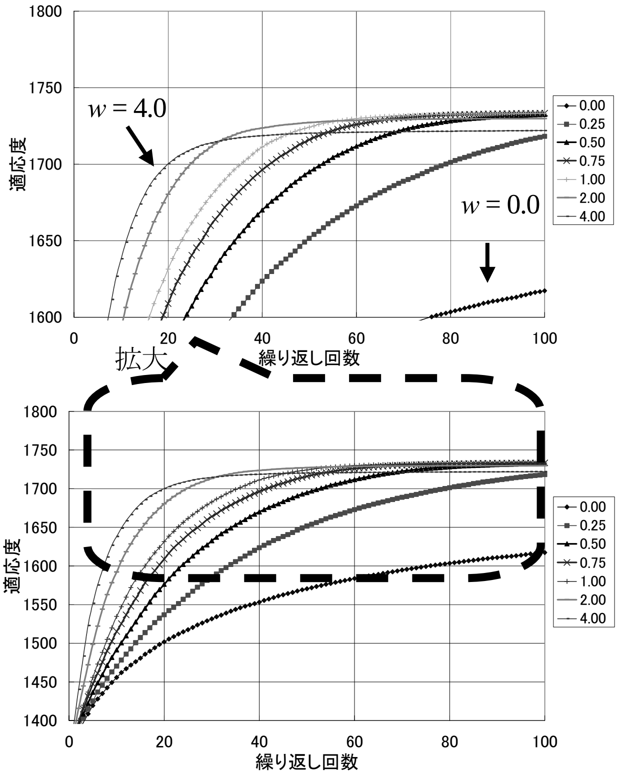


図-4 学習係数 c_2 が適応度に与える影響

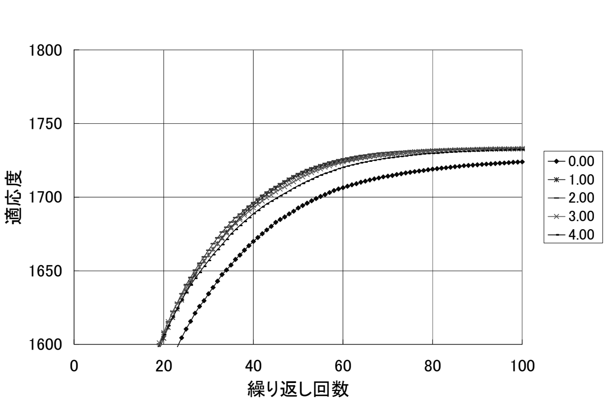


図-5 学習係数 c_1 が適応度に与える影響

4 の場合，最適解探索のプロセスはあまり変わらないが，c_1 が 2.0 のときの適応度が最もよい結果となり，学習係数 c_2 の検証結果と整合している．

以上の検討で得られたパラメータ値（w = 1.0，c_1 = 2.0，c_2 = 0.75，以下，最適パラメータ群と呼ぶ）と既往の推奨値（w = 0.9，c_1 = 2.0，c_2 = 2.0）による解を比較してみる．

まず，50 荷物のナップサック問題において，1,000 回の試行で得られた最適解の割合と粒子数の関係を示す．**図-6** より全体的に，最適パラメータ群による結果は，従来の推奨パラメータ群による結果に比べて，最適解の得られる割合が著しく増加した．とくに，粒子数が少ない場合ではより顕著である．

次に，100 荷物の場合に，後述する近似最適化手法の評価指標のひとつである相対比を用いて，パラメータによる近似最適化手法の影響を評価する．一般に，近似最適化手法の特徴は，厳密な最適解が得られない場合でも最適解に近い解(近似最適解)を求められることである．そこで，以下のような相対比 [4] を用いて近似解の精度を評価した．

$$r = \frac{f(x)}{f(x^*)} \tag{6}$$

ここで，相対比を r，最適解を $f(x^*)$，近似最適解を $f(x)$ とする．式(6)は，相対比が大きいほど最適解に近いことを示している．

図-7 には，100 荷物のナップサック問題で得られた相対比を示す．この図より，最適パラメータ群を用いることによって全てのケースにおいて近似最適解を評価する相対比でも改善されたことがわかる．

3.2 荷物数と粒子数の影響

ここでは，荷物数と粒子数の影響を検討するため，荷物数が 30, 50 および 100 の問題を DPSO および SGA によって検討する．それぞれの問題において，粒子数および個体数を 30，50，100，150，200 に変化させて，合計 15 ケースのシミュレーションを行った．ここで，DPSO のパラメータは，最適パラメータ群とした．SGA のパラメータは，通常よく用いら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/遺伝子長とした．世代数および繰り返し回数は 100，DPSO，SGA ともにシミュレーションを 1,000 回繰り返した．

図-8 には，荷物数 30 の問題において 1,000 回の試行中で最適解が得られた割合を示す．まず，SGA の結果をみると，個体数が増えるにつれて最適解が得られる割合は徐々に増えている．一方，DPSO では，粒子数が 30 と少ない場合でも最適解の割合は 80%を超えている．さらに，粒子数が 150～200 にまで増加すると最適解の割合は 100%となり，常に最適解を求めている．これより，DPSO は SGA に比べて極めて高い探索能力があることがわかる．

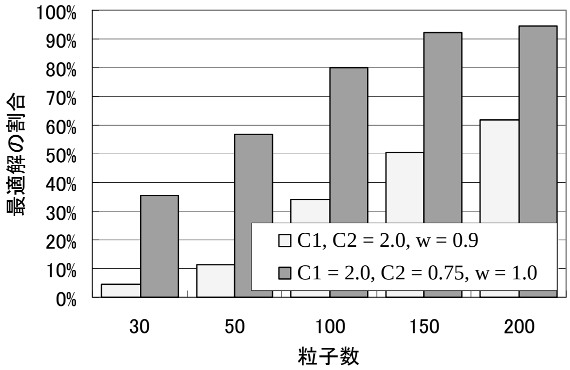


図-6 パラメータが最適解の割合に及ぼす影響（50 荷物）

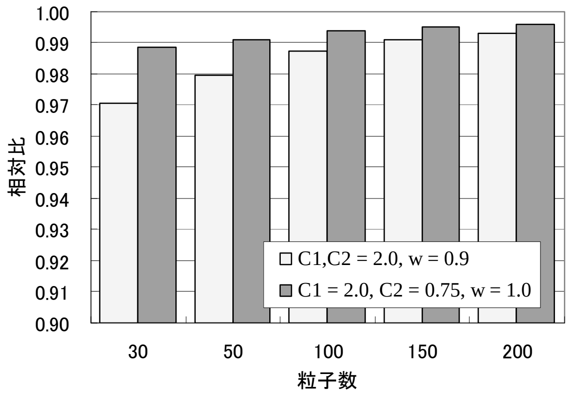


図-7 パラメータが相対比へ及ぼす影響（100 荷物）

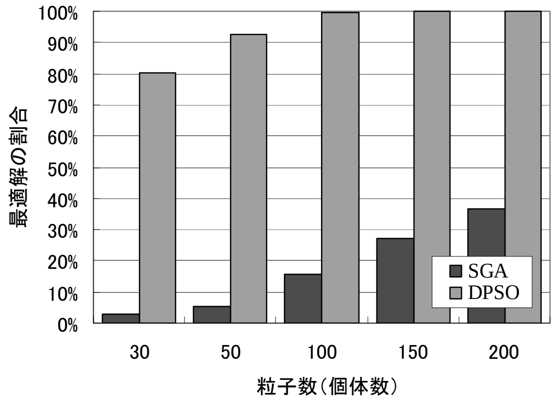


図-8 30 荷物のときの最適解の割合

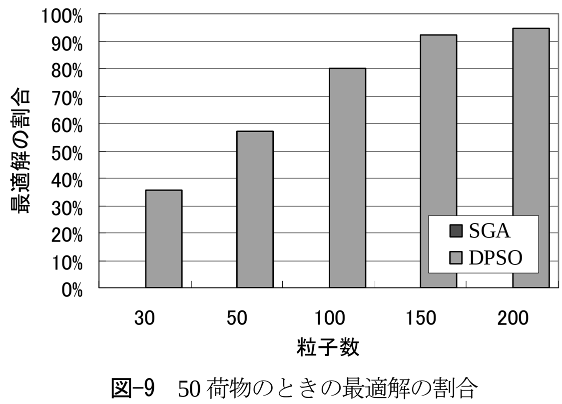


図-9 50 荷物のときの最適解の割合

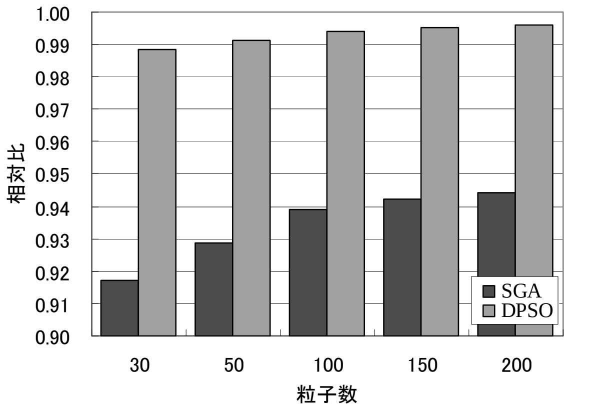


図-10　DPSO と SGA の相対比(100 荷物)

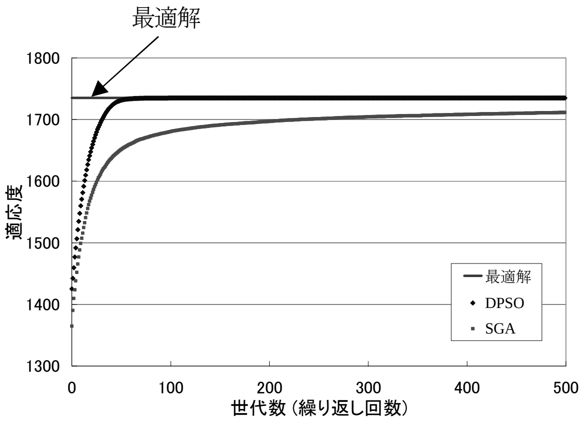


図-11　50 荷物 200 粒子（個体）の時の適応度の推移

図-9 には，荷物数 50 の問題の結果を示す．まず，SGA では全てのケースにおいて最適解が得られなかった. この理由は，SGA における世代数が，問題に対して不十分であったためと考えられる．ちなみに，荷物数 100 のときも同様の結果となった. 一方, **図-9** に示す荷物数 50 の DPSO の結果では, 粒子数の増加にしたがって着実に最適解の割合が増加し, 粒子数 200 では 95%の割合で最適解を得た．

図-10 には，100 荷物のときの SGA と DPSO の相対比を示す．ここでは，DPSO と SGA ともに相対比が 0.9 以上となり，また，粒子数および個体数が多いほど相対比が高くなっている．すなわち，厳密な最適解の探索はできなかったものの, どちらの手法も比較的高い近似最適解を求めていることがわかる．とくに，DPSO の相対比は全てのケースで 0.98 以上となり非常に精度が良いことがわかる．

図-11 には，荷物数 50，粒子数および個体数 200 のケースにおける適応度の推移を示す．この結果は 1,000 試行回数の平均を示している．まず，SGA の場合は，30 世代までに適応度が急激に増加して進化している様子がわかる．しかし，50 世代から緩やかな増加傾向を示すようになり 500 世代に至ってもまだ収束していない. 一方, DPSO の場合は繰り返し回数 50 未満で, 適応度は収束しており, SGA に比べて収束性も良いことがわかる．

4　まとめ

本研究では，組合せ最適化問題の代表的例題であるナップサック問題のシミュレーションによって DPSO のパラメータの検討を行った．また，解探索能力を比較するために SGA との比較を行った．なお，その他の問題事例によってもパラメータの検討が必要と考えている．以下に，本研究によって得られた結果を要約する．

(1)　組合せ最適化問題としてナップサック問題を用いて，DPSO と SGA の比較を行った. その結果, DPSO は, SGA に比べ極めて高い探索能力と早い収束性を有していることがわかった．また，近似最適解を評価する相対比においても DPSO の方が，SGA よりも良いことが分かった．

(2)　DPSO のパラメータが解の探索能力に与える影響について検討を行った．その結果，適切なパラメータの範囲が存在することがわかった．本研究に用いたモデルでは，w = 1.0，c_1 = 2.0，c_2 = 0.75 のとき最も効率が良いことが分かった．

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)　Shi Yuhui, R.Eberhart : Empirical Study of Particle Swarm Optimization , *Proc. of IEEE Congress on Evolutionary Computation* , pp1945-1950, 1999.

4)　茨木俊秀：離散最適化法とアルゴリズム，株式会社岩波書店，1993.