DPSO による 15 パズルの解探索に関する研究

A Study on Finding Solutions of 15 Puzzle Using DPSO

印南 信男（近畿職業能力開発大学校）

Michio Innami

15 パズルは世界中でよく知られているゲームである。本研究では DPSO によって 15 パズルの解の探索を試みた。DPSO はメタヒューリスティクスの手法のひとつであり、解空間が連続値をとる PSO を改良し離散値を扱えるようにしたものである。パズルのコマを 10～50 回シャッフルして生成された問題に対し、探索性能を調べるため試行を行った。計算の結果、30 以下のシャッフル回数で生成された問題に対しては、50%以上の試行で解を得ることができた。遺伝的アルゴリズムによる探索をあわせて行い比較したところ、DPSO による探索の方でより良好な結果が得られた。

キーワード：メタヒューリスティクス、組合せ最適化、DPSO、遺伝的アルゴリズム、15 パズル

1.　はじめに

15 パズルゲームは 1870 年頃にアメリカ合衆国で考案され[1]、その後世界中に広まりよく知られるゲームのひとつとなった。図 1 に示す 16 セルからなる盤上にランダムに配置された 1～15 のラベルを有する 15 個のコマに対し、1 手につき 1 コマを 1 セル分たてまたは横のいずれかに移動させて、正解となる配置に並べ替えるゲームである。

出題されるコマの配置が正解の偶置換であれば解くことができるが、奇置換の場合は解が存在しない[2, 3]。図 2 は有名な 14-15 パズルであるが、図 1 の奇置換となっているため、図 1 のように並べ替えることは不可能である。

図 1　15 パズル

図 2　14-15 パズル

出題されたパズルに対する解答は、4 方向いずれかのコマの移動を順に組合せることによって示される。したがって本パズルは組合せ最適化問題として考えることができる。

本研究ではメタヒューリスティクスの 1 手法として知られている DPSO (Discrete Particle Swarm Optimization)を用いた解の探索を試みる。メタヒューリスティクスは多くの問題に対して汎用的な取り扱いができることが特徴である。本問題に対する DPSO の解の探索性能を調べるとともに、メタヒューリスティクスの手法として有名な遺伝的アルゴリズムによる探索をあわせて行う。両者の探索性能の比較を行うことによって、DPSO の優位性について検証を行う。

2.　PSO と DPSO

2.1　PSO

粒子群最適化(Particle Swarm Optimization : PSO)は、鳥や魚などの群れの行動をモデルとして 1995 年に J. Kennedy と R. C. Eberhart によって提案されたメタヒューリスティクスの手法である[4]。連続変数をとる関数の最適化を目的とし、アルゴリズムが単純であり収束性に優れるという特徴を有する。解空間に存在する集団内の粒子(個体)が、自己のそれまでの移動履歴と全粒子の移動履歴の内から最優良探索点の情報を得て、その値に基づいて解空間内での移動を繰り返す。

各粒子は、初期値として解空間内にランダムな位置と速度を与えられる。粒子 i において k イテレーション目から $k+1$ イテレーション目への位置の更新は式(1)、(2)にしたがって行われる。

$$\boldsymbol{v}_i^{k+1} = w\boldsymbol{v}_i^{k} + c_1 r_1 (\boldsymbol{pbest}_i - \boldsymbol{x}_i^{k}) + c_2 r_2 (\boldsymbol{gbest} - \boldsymbol{x}_i^{k}) \quad (1)$$

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k + \boldsymbol{v}_i^{k+1} \tag{2}$$

ここで、$\boldsymbol{x}_i^k$、$\boldsymbol{v}_i^k$はそれぞれkイテレーション目の粒子iにおけるn次元の位置ベクトル、速度ベクトルであり、w、c_1、c_2は定数、r_1、r_2は0から1までの範囲の一様乱数である。$\boldsymbol{pbest}_i$は粒子iの移動履歴における最良探索点、$\boldsymbol{gbest}$は集団内の全粒子の移動履歴における最良探索点である。なお最良探索点には、あらかじめ設定された目的関数(適応度関数)の値が最小あるいは最大となる粒子が選定される。

イテレーションが所定の値に達するまで、式(1)、(2)の計算を繰り返す。

2.2 DPSO

DPSOは、PSOで離散値の探索を行うために提案された手法であり[5]、$\boldsymbol{x}_i^k$の各成分は0、1のいずれかの値をとる。$\boldsymbol{x}_i^{k+1}$を得るためには式(2)の代わりに以下に示す式(3)が用いられる。

$$x_{i,j}^{k+1} = \begin{cases} 1 & \text{if } \rho < S\left(v_{i,j}^{k+1}\right) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

ここで、$x_{i,j}^{k+1}$、$v_{i,j}^{k+1}$はそれぞれ$\boldsymbol{x}_i^{k+1}$、$\boldsymbol{v}_i^{k+1}$のj番目の座標成分($1 \le j \le n$)、ρは0から1までの範囲の一様乱数である。また$S(v_{i,j}^{k+1})$は式(4)によって与えられるシグモイド関数である。

$$S(v_{i,j}^{k+1}) = \frac{1}{1+\exp(v_{i,j}^{k+1})} \tag{4}$$

3. 15パズル問題へのDPSOの適用

目的関数を最小化する組合せ最適化問題は式(5)、(6)のように定式化することができる。

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \tag{5}$$

$$\text{subject to } \boldsymbol{x} \in \mathsf{F} \tag{6}$$

ここで、$f(\boldsymbol{x})$:目的関数、F :実行可能領域である。

15パズルにおいて目的関数などを適切に設定することにより、効率的な解の探索を行うことが可能になると考えられる。

図1の配置からランダムにコマをシャッフルして生成された問題を用意し、そのパズルに対してもとの配置に並べ替えるためのコマの移動順序を探索する。ここで1手分の操作は、空白セルに隣接するコマを空白の位置に移動することである。これは空白セルを移動していると考えることができる。本稿では空白セルの移動方向を←、→、↑、↓で表すこととする。たとえば←は空白セルの左隣に位置するコマを右に移動することを意味する。

移動方向をランダムに決定して並べ替えを続けていくと、空白セルがいずれ盤から外れる可能性が高くなる。そこで本研究では、これを避けるため空白セルの移動方向を以下のように設定した。

図3に示すように実線と破線で示す2種類の閉路を考える。いずれも盤内の全てのセルを通る。それぞれについて時計回りと反時計回りの回転方向を考える。いずれのセルにおいても、閉路の種類と方向を指定することで移動先のセルを指定できる。すなわち1手につき2ビットの情報で、全てのセルにおいてその位置で可能なあらゆる移動方向を表現することができるとともに、盤からコマが外れる状態が発生することはない。これにより遺伝的アルゴリズムにおける致死遺伝子に相当するような粒子の発生を抑えることができる。

j番目の手で行う空白位置の移動については、閉路の種類、方向をそれぞれ位置ベクトルの成分x_{2j-1}、x_{2j}で指定し、いずれも0か1の値をとるものとする。

偶置換の場合、全ての問題は80手以内で解けることが明らかとなっている[6]。そこで、探索では150手までのコマの移動を考える。適応度は解答に要した手数とする。ただし、150手までに目的の配置に達することができなかった場合は、150手目における各コマについてそれぞれ図1の配置とのマンハッタン距離をとり、その全コマについての和に150を加えたものを適応度と定義した。

なお得られた解において、途中のある1手がその直前の操作を打ち消すように作用する場合(たとえば←の直後が→)については、その両者を位置ベクトル成分から除去し手数に含めない。

4. 問題の生成

問題は図1の配置から定められた回数だけ空白位置をランダムに移動(シャッフル)することによって生成した。

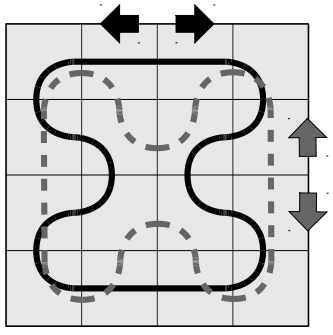


図3　空白セルの移動方向の定義

表 1　各粒子数において解が得られた試行数

粒子数	最大イテレーション	解が得られた試行数	最適解が得られた試行数
20	15000	13	8
60	5000	14	11
100	3000	16	9
150	2000	16	7
200	1500	17	10
250	1200	19	10
300	1000	20	8
500	600	16	9
1000	300	18	12

表 2　w の各値に対して解が得られた試行数

w	解が得られた試行数	最適解が得られた試行数
0.4	15	14
0.6	17	13
0.8	18	13
0.9	17	9
1.0	19	10
1.1	15	14
1.2	15	12
1.4	6	5
1.6	8	7

表 3　c_1 の各値に対して解が得られた試行数

c_1	解が得られた試行数	最適解が得られた試行数
1.5	18	8
2.0	19	10
2.1	19	10
2.2	17	10
2.3	20	16
2.4	18	10
2.5	20	15
2.7	17	13
3.0	16	11

このとき、コマが盤から外れることがないようにした。またシャッフル途中のコマの配置がその状態に至るまでの履歴のいずれとも同一となることがないようにした。

表 4　c_2 の各値に対して解が得られた試行数

c_2	解が得られた試行数	最適解が得られた試行数
1.0	17	9
1.5	17	13
1.8	19	16
1.9	18	8
2.0	20	16
2.1	18	14
2.2	17	13
2.5	16	9
3.0	13	9

ただし上記による問題生成方法では、シャッフル回数未満の手数で解が得られる場合が起こり得る。

5.　探索の手順と結果

5.1　パラメータの決定

式(1)などに現れるパラメータの最適値を決定するために、予備計算を行った。図 1 の配置からランダムに 15 回空白位置をシャッフルして生成された問題に対して、以下のそれぞれのパラメータについてさまざまな値で 20 回ずつの試行を行い、各値と解が得られた試行数の関係を調べた。

5.1.1　粒子数の決定

式(1)中のパラメータを PSO において最適値とされる $w=1.0$、$c_1=2.0$、$c_2=2.0$ とし[7]、表 1 に示すように粒子数と最大イテレーションを、両者の積がほぼ一定値となるさまざまな組合せで試行を行った。その結果解が得られた試行の数も表 1 に示した。なお、ここではシャッフル回数すなわち 15 以下の手数で解くことができた解を最適解と定義している。得られた結果より、粒子数を 250 に決定した。以後の試行にはこの値を用いる。

5.1.2　定数 w の決定

w は粒子の慣性項の重みを表す。w を表 2 に示す各値に設定し試行を行った。
結果も表 2 に示した。得られた結果より、$w=1.0$ とし、以後の試行にこの値を用いる。

5.1.3　定数 c_1 の決定

c_1 は自己の最良探索点に関する重みを表す。c_1 にさまざまな値を設定し、試行を行った結果を表 3 に示す。この結果より $c_1=2.3$ とし、以後の試行にこの値を用いる。

表 5　GA のパラメータ

個体数	30
交叉率	0.8
突然変異率	0.01
エリート個体数	3

5.1.4　定数c_2の決定

c_2は集団内の全粒子中の最良探索点に関する重みを表す。c_2にさまざまな値を設定し、試行を行った結果を表 4 に示す。この結果より$c_2 = 2.0$とした。

5.2　GA のパラメータ

本研究では DPSO の探索結果と比較するために、同一問題に対して遺伝的アルゴリズム(Genitic Algorithm : GA)による探索をあわせて行った。

DPSO の位置ベクトルの代わりに、GA では空白位置の移動の情報を染色体に格納する。1 手につき 2 個の遺伝子座が用いられ、図 3 の閉路とその方向がいずれも 0 か 1 の値で指定される。探索アルゴリズムにはルーレット選択、エリート選択、2 点交叉を採用した。パラメータは表 5 に示すとおり、GA で一般によく用いられる値を採用した。適応度の定義は DPSO の場合と同じである。

6.　計算結果

前章で決定したパラメータを用いて DPSO による計算を行った。図 1 の状態からランダムに 10、20、30、40、50 回空白位置のシャッフルを行い問題を生成した。それぞれについて 10000 イテレーション目までの試行を 30 回繰り返した。なお試行ごとに新たに問題を生成した。

図 4 に各シャッフル回数で生成された問題について、150 手以内での解答が得られた試行の数の推移を示す。またシャッフル回数以下の手数で問題が解けた解に対して、便宜上最適解とみなすと、その推移は図 5 のようになった。

シャッフル回数 10 の問題に対しては、全ての試行で 100 イテレーション目までに解が得られた。また 50 イテレーション目までに 25 の試行で最適解が得られ、その数は 10000 イテレーション目まで変わらなかった。シャッフル回数 30 の問題では、約 50%の試行で解が得られた。

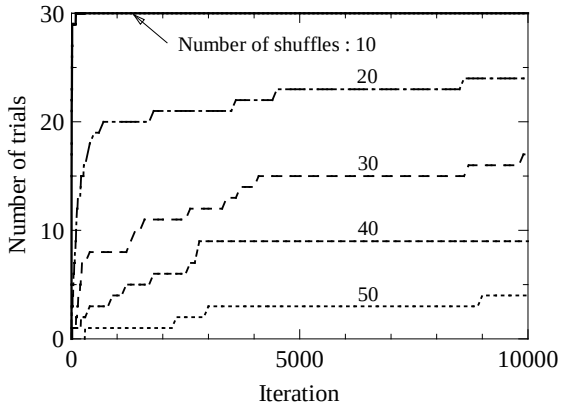


図 4　解が得られた試行数の推移(DPSO)

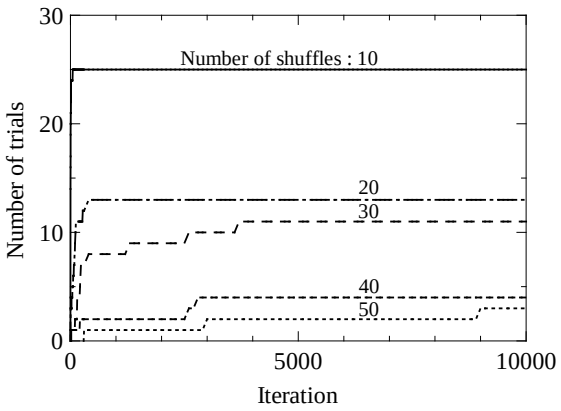


図 5　最適解が得られた試行数の推移(DPSO)

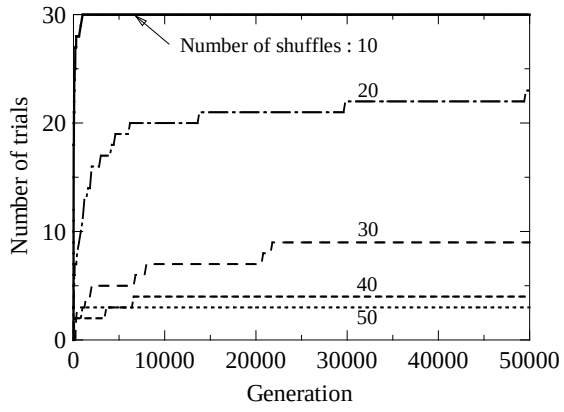


図 6　解が得られた試行数の推移(GA)

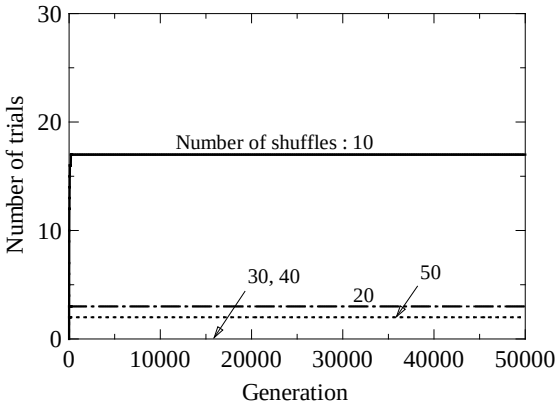
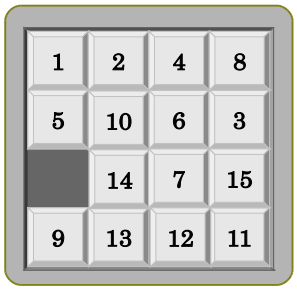
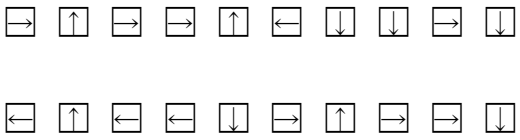


図 7　最適解が得られた試行数の推移 (GA)



(a)　生成された問題(シャッフル回数 20)



(b)　探索で得られた解

図 8　生成された問題と得られた解の一例

ただし最適解が得られた試行は全体の 40%未満であった。一方、シャッフル回数 50 の問題では 300 イテレーション目に 1 試行で解が求められた。これは最適解でもあった。

いずれの場合も、5000 イテレーション目までに解の探索はほぼ収束していることが認められる。

比較のために行った、GA による探索の結果を図 6、7 に示す。GA による探索は 50000 世代目まで行った。個体数(粒子数)と世代数(イテレーション数)の積が同じ値となるように比較すると、DPSO で 5000 イテレーション目は GA ではほぼ 42000 世代目に相当する。両者を比較すると DPSO の探索性能が優れていることがわかる。また GA の 50000 世代目までの試行に要する時間よりも DPSO の 10000 イテレーション目までに要する時間のほうがおおよそ 2〜3 割短い結果となった。

生成された問題と DPSO による探索で得られた解(空白位置の移動順序)の一例を図 8 に示す。

7.　まとめ

本研究では DPSO によって 15 パズルの解の探索を試みた。予備計算により最適なパラメータが明らかとなった。これらの値を用いて試行を行ったところ、10〜50 回のシャッフルによって生成された問題に対しては、いずれも 5000 イテレーション目までに解の探索がほぼ収束していることが認められた。シャッフル回数 10 の場合、100 イテレーション目までに全ての試行で解が得られ、約 8 割の試行で 50 イテレーション目までに最適解が得られた。一方、シャッフル回数 50 の問題に対しては、10000 世代目までの探索で解が求められた試行は約 10%であった。GA による探索と比較すると、より少ない計算コストでより優れた結果が得られることが確認された。

解は一連の操作手順列として表現されている。本問題の場合には部分解同士の依存関係が強いため、GA におけるビルディングブロック仮説 [8]の観点からは GA のみならず DPSO に対しても難しい問題であると考えられる。解の表現方法の変更や他のアルゴリズムとの組合せなどにより、探索性能の向上をはかることが今後の課題である。

参考文献

1. W. W. Johnson: Notes on the ‘15’ puzzle. I, American Journal of Mathematics, **2**, pp.397-399(1879).
2. A. F. Archer: A Modern treatment of the 15 puzzle, American Mathematical Monthly, **106**, pp.793-799(1999).
3. R. G. Campos, C. H. S. Tapia: ¿Realmente imposible?, Miscelánea Matemática, **40**, pp.69-75(2004).
4. J. Kennedy, R. C. Eberhart: Particle Swarm Optimization, Proc. of IEEE Int’l Conf. on Neural Networks, Piscataway, NJ, pp.1942-1948(1995).
5. J. Kennedy, R. C. Eberhart: A discrete binary version of the particle swarm optimization, Proc. of the 1997 Conf. on System, Man, and Cybernetics, pp.4104-4109(1997).
6. A. Brungger, A. Marzetta, K. Fukuda, and J. Nievergelt: The Parallel Search Bench ZRAM and its applications, Annals of Operations Research, **90**, pp.45-63(1999).
7. Y. Shi, R. C. Eberhart: A Modified Particle Swarm Optimizer, Proc. of IEEE Int’l Conf. on Evolutionary Computation, Anchorage, Alaska, Piscataway, NJ, pp.69-73(1998).
8. S. Forrest, M. Mitchell: Relative Building-Block Fitness and the Building-Block Hypothesis, in D. Whitley, ed. FOGA 2, San Mateo, CA, 1993, Morgan Kaufmann.

（原稿受付 2014/01/15、受理 2014/03/25）

*印南信男,
近畿職業能力開発大学校，〒596-0103　大阪府岸和田市稲葉町 1778　email:Innami.Michio@jeed.or.jp
Michio Innami, Kinki Polytechnic College, 1778 Inabacho, Kishiwada, Osaka 596-010