

大規模問題に対するパラメータフリー GA と 実験計画法を用いた GA の効率に関する考察

Sangheon Han*

1 はじめに

遺伝的アルゴリズム (Genetic Algorithm : GA) は、生物の進化の過程をモデル化した最適化手法の一つである。多くの問題に対して優れた性能を示す GA であるが、初期集団サイズや交叉率、突然変異率などのパラメータにより性能が大きく変化してしまう。GA においては、交叉率 (Crossover rate)、突然変異率 (Mutation rate) など各種パラメータのチューニングはもちろん、それらのアルゴリズムそのものも重要な研究対象となっている。そのため、GA のパラメータの設定の必要がない GA (Parameter free Genetic Algorithm : PfGA) が提案された [8]。PfGA の設計思想としては少ない集団サイズでより良い個体を追求する探索を行う点が挙げられる。本研究では、GA の効率的なパラメータの決め方に対して、パラメータフリーな方法と実験計画法によるパラメータの設定方法 (Design of experiment method GA : DeGA) [2] を挙げて、これらの GA の性能比較と探索効率の向上を目的とする。

2 遺伝的アルゴリズム (Genetic Algorithm)

遺伝的アルゴリズム (Genetic Algorithms : GA) は、生物が環境に適応して進化していく過程を科学的に模倣した学習型アルゴリズムの一つである。GA の研究は 1960 年代後半から 1970 年のはじめに Michigan 大学の J.H.Holland [4] らによって始められ、その研究成果は 1975 年に「Adaptation in Natural and Artificial System」という題で出版されている。また、1989 年に出版された Goldberg の「Genetic Algorithms in Search, Optimization, and Machine Learning」以降、日本でも数多くの研究実績があり、現在では生物学を含めて、工学、経営学、経済学など様々な分野に応用されている。自然界における生物の進化過程においては、ある世代を形成している個体の集合、すなわち母集団の中で、環境により適応した個体がより高い確率で生き残り、次の世代に優秀な因子を残す。このメカニズムをモデル化し、環境に対して最もよく適応した個体 (Individual)、すなわち目的関数に対して最適値を与えるような解を計算機上で求めようというのが GA の仕組みである。GA では、個体は設計数の値がコーディングされた染色体 (Chromosome) と呼ばれる文字列として表現され、この染色体をデコーディングすることにより設計変数を読み出し、目的関数の値を計算する。このとき、染色体の構造のことを遺伝子型 (Genotype)、これによって定まる個体の形質を表現型 (Phenotype) と呼ぶ。また、個体の集団のことを母集団 (Population) と呼ぶ。GA はこの母集団に対して選択 (Selection)、交叉 (Crossover)、突然変異 (Mutation) などの遺伝的操作を繰り返し行うことによって解を求めて行く。

* Faculty of Management Information Science, Nagoya University of Commerce & Business, Komenoki, Nishin, Aichi 560-0043, Japan

- Step1: ランダムに木構造 $GTYPE\{g_t(i)\}$ を生成
- Step2: 各 $GTYPE\{g_t(i)\}$ の表現型 $PTYPE\{p_t(i)\}$ に対して、適合度 $\{f_t(i)\}$ を求める
- Step3: 適合度の大きな $GTYPE$ に対して一定数のペアを取り出す
- Step4: 取り出したペアに対して crossover を行い、適合度の小さな $GTYPE$ と置き換える
- Step5: 各 $GTYPE$ に関して、ランダムに inversion, mutation を適用する
- Step6: 求められた新しい $GTYPE$ を、次の世代の $\{g_{t+1}(i)\}$ として、Step2 へ戻る

- **Initialization (母集団の初期化)**

この操作は、あらかじめ設定された数だけランダムに個体を生成するものである。生成した個体の数のことを母集団サイズ (Population Size) や単に個体数と呼び、ここで生成した個体の集団を初期母集団とする。

- **Evaluation (評価)**

この操作は、各個体の持つ染色体を問題空間にデコードして解を求めるものである。目的関数の評価値でもある。ここで求めた解のことを評価値 (Evaluation Value) という。

- **Selection (選択)**

この操作は、生物の適者生存を模倣したものである。この操作では、まず各個体の評価値から次世代への生き残りやすさを求め、これに基づいて次世代の母集団を形成する。この次世代への生き残りやすさのことを適合度 (Fitness) と呼ぶ。

- **Crossover (交叉)**

この操作は、生物の有性生殖を模倣したものである。この操作により、個体間で染色体情報が交換される。最適解を表す個体の一部分を持った個体同士が交叉すれば、より最適解に近い個体が得られる可能性が高くなる。個体集団のうち何割の個体が交叉するかを交叉率 (Crossover Rate) と呼ばれるパラメータによって定める。

- **Mutation (突然変異)**

染色体は遺伝子 (Gene) を格納する複数の遺伝子座 (locus) から構成され、遺伝子座に入りうる遺伝子のことを対立遺伝子という。突然変異とは、染色体上の遺伝子座の遺伝子を別の対立遺伝子に置き換える操作のことであり、自然界における DNA 複写の際に起こるコピーミスにあたる。染色体のうち何割の遺伝子座が変異するかを突然変異率 (Mutation Rate) と呼ばれるパラメータによって定める。

- **Terminate Condition (終了判定)**

この操作は、あらかじめ定められた終了条件に基づいて GA を終了させるためのものである。

3 パラメータ設定におけるアプローチ

3-1. PfGA の概要

PfGA では、探索が進むと集団サイズが増加し、停滞すると減少する。アルゴリズムは以下の通りである。

- (1) 1 個体を作成し、初期集団 P とする
- (2) 1 個体を作成し、P に追加する

- (3) P から 2 個体 (P1,P2) をランダムに選択
- (4) 多点交叉を行い, 子個体 C1,C2 を作成
- (5) C1,C2 のうち, 1 個体に突然変異
- (6) 以下の条件式により, 1 個体ないし 3 個体を P に戻す
- (7) 集団数が 2 以上なら (3) に戻り, 1 なら (2) に戻って繰り返す

次世代に残す個体の選択は次のように行う

- ・ $C1 > P1, P2$ かつ $C2 > P1, P2$ の時
C1,C2 と親のいい方を集団 P へ (集団サイズ増加)
- ・ $P1 > C1, C2$ かつ $P2 > C1, C2$ の時
親のいい方を集団 P へ (集団サイズ減少)
- ・ $P1 > C1, C2$ もしくは $P2 > C1, C2$ の時
子の大きい方と親の大きい方を集団 P へ
- ・ $C1 \geq P1, P2$ もしくは $C2 \geq P1, P2$ の時
子のいい方を集団 P へ
さらに, 1 個体作成し, 集団 P へ追加定される

3-2. GAAP の概要

GAAP では, PfGA とは逆に探索が進むと集団サイズは減少し, 停滞すると増加させ探索範囲を広げて進化が促される. アルゴリズムは下記の通りである.

- (1) 2 個体を生成し, 初期集団 P とする
- (2) ルーレット選択を行い, 親個体を選択
- (3) 交叉を行い, P に追加. 2 つの子が, どちらも親より良かった回数 (good2), どちらかが良かった回数 (good1), どちらも悪かった回数 (bad2) を記録する
- (4) 次の世代の集団サイズを計算する
- (5) 計算された集団サイズ分, 適合度の高い方から次の世代に残す
- (6) エリート保存による突然変異
- (7) (2) に戻り, 繰り返す

次世代の集団サイズ (Psize) は下記の式により決定される

- ・ (good2 $\geq$ 1, bad $\geq$ 1) の時
$$Psize(t+1) = Psize(t) + 2 - good2 - \sqrt{good1}$$
- ・ (good2 $\geq$ 1, bad=0) の時
$$Psize(t+1) = Psize(t) + 1 - good2 - \sqrt{good1}$$
- ・ (good2=1, bad $\geq$ 1) の時
$$Psize(t+1) = Psize(t) + 2 - good1/2$$
- ・ (good2=0, bad=0) の時
$$Psize(t+1) = Psize(t) + 1 - good1/2$$

3-3. Design of Experiment Method GA (DeGA) の概要

DeGA では、PfGA と GAAP とは違って、問題ごとに実験計画法を用いて適切なパラメータを設定する方法である。GA の集団サイズは重要なパラメータの一つであるが、今回の研究では、遺伝的プロセッサである交叉確率 (crossover probability) と突然変異確率 (mutation probability) に興味深い。集団サイズは固定して、交叉確率と突然変異確率に対して、統計的な手法である実験計画方法を行い二つのパラメータの適切な組み合わせを求める。パラメータの設定のみ実験計画方法を使うため、さまざまな GA を利用することができる。アルゴリズムは下記の通りである。詳しい内容については、参考文献の [2] Han SH (2001) を参照されたい。

4 数値実験

4-1. テストモデル (Vehicle Routing Problem Model) ¹

パラメータの設定に関する実験には、組み合わせが、以下のような組み合わせ最適化問題の典型的な問題である TSP (Traveling Salesman Problem) に、様々な現実的な制約を考慮した VRP (Vehicle Routing Problem) を採用した。VRP は、配送先の需要や配送手段であるトラックのキャパシティなどを考慮しながら最適な巡回のルートを求める問題で、巡回先の数が増えると非常に難解となり、NP 完全問題に属するものである。

$$\begin{aligned}
 &\text{Minimize} && \sum_{(i,j) \in A} c_{ij} x_{ij} \\
 &\text{subject to} && \sum_{j \in V'} x_{ij} = 1 && \forall_i \in N \\
 &&& \sum_{j \in V'} x_{ji} = 1 && \forall_i \in N \\
 &&& x_{ij} = 1 \Rightarrow y_i - q_i = y_j && \forall (i,j) \in I \\
 &&& q_i \leq y_i \leq Q && \forall_i \in V \\
 &&& x_{ij} \in \{0,1\} && \forall (i,j) \in A
 \end{aligned}$$

4-2. GA パラメータの効果

パラメータの変動による効果を観測するため、crossover rate と mutation rate をそれぞれ 0.0 から 1.0 まで 0.1 の間隔で模擬実験を行った。二つのパラメータの相関性は無視できないため、単独ではなく組み合わせで実験を行った。

- Population size : 101
- Selection : エリート戦略
- Crossover method : one-point crossover
- Mutation method : bit inverse
- Generation : 2000

¹ VRP モデルについて、詳しくは Han (2002) と Golden (1998) を参照されたい。またこのデータは、参考文献 [3] Han SH (2007) のモデルとデータを使用した。

- Crossover & Mutation rate : 0.0 ~ 1.0

実験は、Windows XP, Pentium 4 2GHz, 1G-RAM の PC で行い、開発ツールとしては、Borland C++ Builder を使用した。

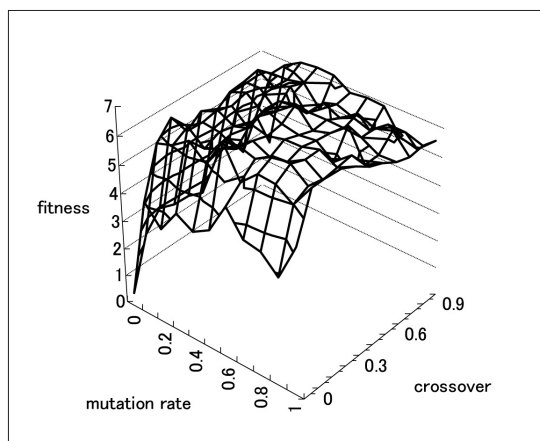


図 4-1 交叉率と突然変異率の組み合わせによる GA の効果

図 4-1 は、交叉と突然変異のパラメータの変化による Fitness Value（評価値）を表している。その二つのパラメータの変動による評価値の変動が激しく、解の精度にも影響があることが観測できた。さらに、図 6-1 と 6-2 のように、GA の重要パラメータである、交叉と突然変異の影響力も確かめることができた。

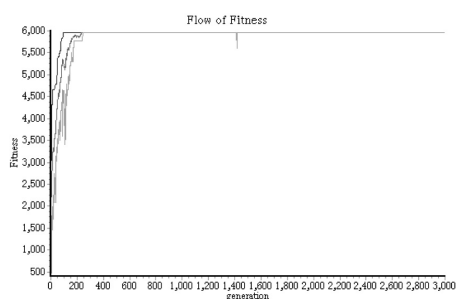


図 5-1 悪いパラメータの設定例

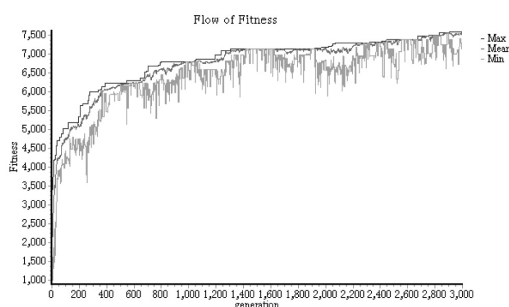


図 5-2 良いパラメータの設定例

6 数値実験と考察

テスト問題に対する数値実験の結果をまとめる。まず、小規模問題に対する実験を行いその効率を図る。解の精度と計算時間を効率として評価する。また、初期状態の値と計算後の値を比較し改善率を求め全体的な効率を表す。小規模の問題としては、訪問先の数が 50 箇所、大規模の問題としては 200 箇所と設定し、計算世代数は 3000 回として模擬実験を行った。また、それぞれの条件に対して、同じ実験を 10 回を行い平均値で評価値を表した。

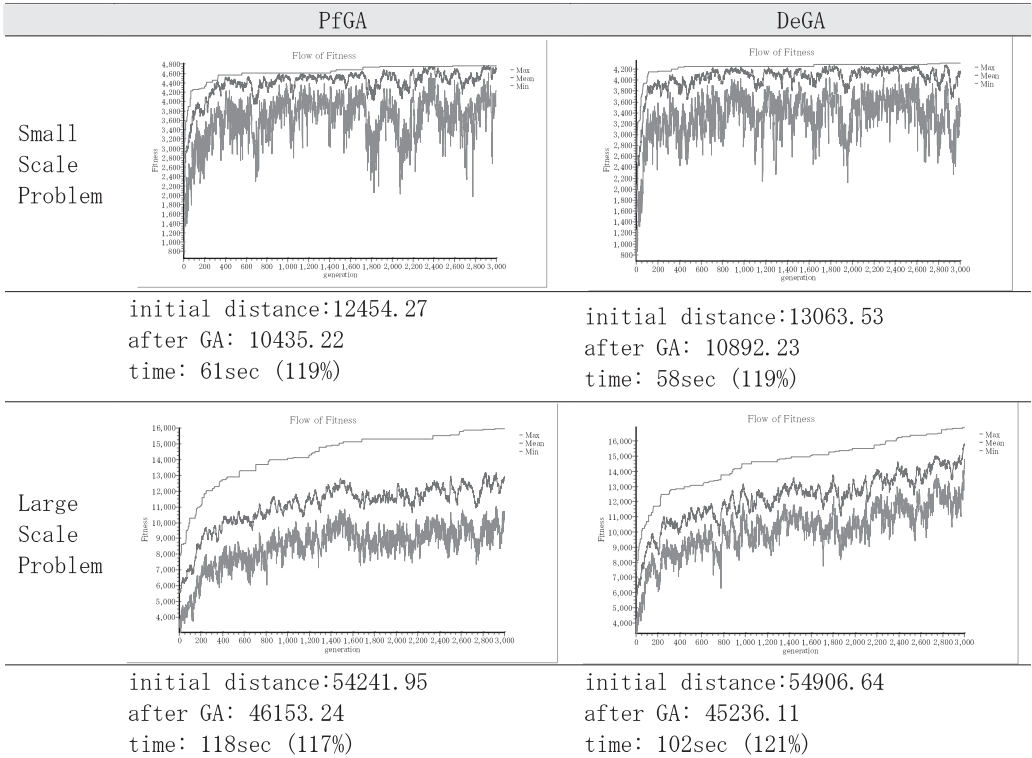


図 6-1 模擬実験結果

図 6-1 は、10 回の実験の中で代表的なケースの結果を表している表である。グラフから見ると、小規模の問題に対しては、二つの手法の違いがあまり見られない。しかし、大規模の問題に対しては、DeGA の方が PfGA より 4% 高い改善率を表している。さらに、計算時間に対しても、僅かな差でもあるが DeGA の方が早い結果を出している。

実験回数	PfGA	DeGA
1	116	120
2	117	119
3	115	123
4	117	121
5	118	120
6	121	117
7	114	119
8	118	119
9	119	122
10	109	120
平均	116.4	120

表 6-2 解の改善率

実験回数	PfGA	DeGA
1	110	104
2	120	120
3	118	116
4	118	108
5	119	103
6	124	113
7	132	113
8	117	120
9	132	121
10	118	113
平均	120.8	113.1

表 6-3 計算所要時間

7 結論および今後の課題

今回の模擬実験結果から、実験計画法を用いたパラメータの決め方が PfGA より良いと判断するのは難しい。表 6-4 と 6-5 の平均的なスコアからは、DeGA の方が明らかに良い結果を出しているが、実験計画法の実施などの実験のためのセットアップ時間を考慮すると、10 秒 20 秒という時間はそれほど大きい意味があると考えにくい。ただし、解に対する精度に対しては、DeGA は十分良い性能を表している。まだ残っている課題は多い。まず、世代数（繰り返し）が 5000 回、10000 回、50000 回のように増えていく場合、解の精度は変わる可能性がある。

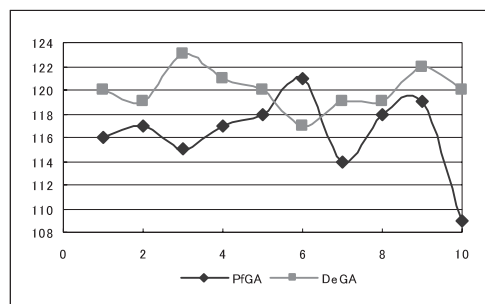


表 6-4 解の改善率

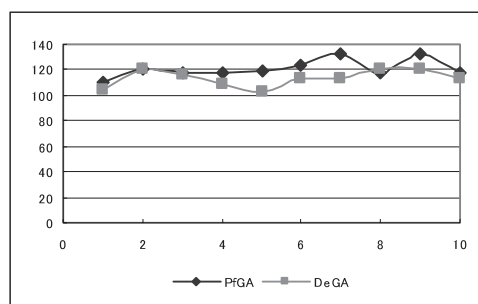


表 6-5 計算所要時間

また、大規模の問題に対しては、計算時間と解に対する精度の解析が変わる可能性もある。実用的なレベルから考えると、決められた時間内での納得する結果を出せば十分な意味を持つケースがしばしばある。GA だけではなく様々な分野において、パラメータの設定に関する研究が行われている現状、今回の研究の成果として挙げられるのは、自動化するパラメータの設定には十分意味があるが、問題の特性を判断するのは人間であり、その判断により適切な手法を選ぶのが望ましい。最後に、優れた性能の PfGA の利点を生かして、他のパラメータの設定方法との融合性を考えてみたい。今後の GA のパラメータの設定における残された課題は、今回の研究ではあまり言及してないが、PfGA の優れた部分でもある母集団サイズの変動的なアルゴリズムに関する部分であり、今後、実用的なレベルで広く活用できることを期待している。

参考文献

- [1] Goldberg DE (1989) Genetic Algorithm in Search, Optimization, and Machine Learning, Addison-Wesley.
- [2] Han SH (2001) Design of experiments to identify optimal parameters of genetic algorithm, Journal of Scientiae Mathematicae Japonica, Vol. 55 (3), pp. 539-545
- [3] Han SH (2007) A Study of Parameter Optimization in Genetic Algorithm using Auto tuning System, NUCB journal of economics and information science, Vol.52, No.1 (20070800) pp. 71-78
- [4] Heinz M (1997) Genetic algorithms, In Local Search in Combinatorial Optimization, John Wiley & Sons, Chichester, 137-172.
- [5] Holland JH (1975) Adaption in Natural and Artificial Systems. University of Michigan Press, Ann Arbor.
- [6] Ph. Vincke, Multicriteria Decision-Aid, John Wiley and Sons, Chichester, New York, 1989.
- [7] Thangiah Sam (1995) Vehicle routing with time windows using genetic algorithms. In Application Handbook of Genetic Algorithms, New Frontiers, CRC Press, 2, 253-277.
- [8] B. L. Golden, A. A. Assad (Eds.) : "Vehicle Routing: Methods and Studies, Studies in Management Science and Systems", Vol. 16, Elsevier Science Publ., Amsterdam, 1988, 7-45.
- [9] 澤井秀文, 木津左千夫, 遠藤哲郎:「パラメータフリー遺伝的アルゴリズムと定常状態遺伝的アルゴリズムとの性能比較」電子情報通信学会論文 Vol.J81-D-「 No.6 pp.1455-1459 (1998 年 6 月)