

Tabu Search を導入した ECGA についての検討

佐竹 佑太[†] 棟 朝 雅 晴^{††} 赤 間 清^{††}

Extended Compact Genetic Algorithm (ECGA)³⁾ は集団の分布を表した確率モデルを構築し、構築したモデルを基に新たな個体を生成するアルゴリズムである。構築したモデルによって、互いに依存する複数の遺伝子を検出することができるため、ECGA は広範囲の最適化問題を解くことができる。ECGA の探索能力をさらに高めるために ECGA に近傍探索法を組み込んだ手法が存在する²⁾。しかしながら、もっとも探索能力の高い近傍探索法のうちの 1 つである Tabu Search⁴⁾ は近傍探索法として用いられてこなかった。そこで、本論文では Tabu Search を ECGA に組み込んだ手法を提案し、その手法の有効性について検討する。

Considering ECGA which Introduces Tabu Search

YUTA SATAKE,[†] MASAHARU MUNETOMO^{††} and KIYOSHI AKAMA^{††}

Extended Compact Genetic Algorithm (ECGA)³⁾ builds its probabilistic model which represents distribution of promising solutions and generates new solutions based on the model. Because ECGA detects interdependent loci by using the model, it can solve a wide spectrum of optimization problems effectively. ECGA which introduces neighbourhood search in order to enhance its performance is already proposed²⁾. However, they did not use tabu search⁴⁾, which is one of the most effective neighbourhood searches, as neighbourhood search. In this paper, we propose ECGA which introduces tabu search and consider its effectiveness.

1. はじめに

広範囲の最適化問題を解くことが可能なアルゴリズムとして、分布推定アルゴリズム (Estimation of Distribution Algorithms, EDAs)¹⁾ が注目されている。EDAs は集団の分布を表した確率モデルを構築し、構築されたモデルを基に新たな個体集団を生成する。EDAs は一般に大域的な探索を得意とするが、突然変異などのオペレータが存在しないため局所的な探索を苦手とする。そこで、EDAs に近傍探索法を組み込んだハイブリッド手法が提案されている²⁾。近傍探索法を組み込むことで EDAs は局所的な探索能力をもつことができ、より強力なアルゴリズムとなる。近傍探索法にはさまざまなものが存在するが、単純な近傍探索法はしばしば局所解におちいり、そこから探索が進まなくなるという欠点がある。いっぽう、近傍探

索法のうちの 1 つである Tabu Search は、メモリをもった探索法であるため局所解から脱出することが可能である。にもかかわらず、EDAs に近傍を組み込んだ手法において、近傍探索法として Tabu Search は用いられていない。そこで、本論文では EDAs に Tabu Search を組み込んだ手法を提案し、その有効性について検討する。

まず最初に、本論文で扱う EDAs である Extended Compact Genetic Algorithm (ECGA)³⁾ について簡単に説明した後に、本論文で扱う近傍探索法である Tabu Search について簡単に説明する。次いで、ECGA に Tabu Search を導入した手法とその目的について述べる。最後に、実験結果を示し考察を加える。

2. Extended Compact Genetic Algorithm

Extended Compact GA (ECGA)³⁾ は Harik によって提案された分布推定アルゴリズムである。ECGA は複数の変数間の依存関係を検出できるため、適用範囲が広い。ECGA の手順を図 1 に示す。

ECGA では 2 つ以上の遺伝子の同時確率分布を表すことのできる MPM (Marginal Product Model)

[†] 北海道大学大学院 情報科学研究科

Division of Systems and Information Engineering,
Graduate School of Information Science and Technol-
ogy, Hokkaido University.

^{††} 北海道大学 情報基盤センター

Division of Large-Scale Computational Systems, Infor-
mation Initiative Center. Hokkaido University.

- (1) 個体集団 (個体数 N) をランダムに初期化する
- (2) トーナメント選択法により選択をおこなう
- (3) 貪欲 MPM 探索 により集団の確率モデルを構築する
- (4) モデルに基づいて新たな集団を生成する
- (5) 終了条件が満たされた場合は終了. それ以外の場合は 2. へ戻る

図 1 Extended Compact Genetic Algorithm
Fig. 1 Extended Compact Genetic Algorithm

という確率モデルを構築する. この MPM を用いて MDL (Minimum Description Length) 原理に基づいたリンケージ学習をおこなう. つまり, 集団に存在する個体の分布をできる限り短い記述長で記述できるようなリンケージを探索する. この記述長を表すために, 次に示す 2 つの複雑度が定義される:

$$C_m = \log N \sum_i 2^{S[i]} \quad (1)$$

$$C_p = N \sum_i E(M[i]) \quad (2)$$

ここで, N は集団サイズ, $S[i]$ は i 番目のリンケージの長さ, $M[i]$ は i 番目のリンケージに関する周辺分布, $E(M[i])$ は $M[i]$ のエントロピーである.

貪欲 MPM 探索は, 全ての遺伝子が独立である状態 (個体長が l の場合, $[0][1] \dots [l-1]$) から始める. 2 つのリンケージを選んで 1 つのリンケージにした場合の複雑度 $C = C_m + C_p$ をすべてのリンケージの組み合わせについて計算する. その中でもっとも小さな複雑度が現在の複雑度よりも小さくなる場合に, そのリンケージの組み合わせを採用する. 以上のリンケージ学習を組み合わせるリンケージのペアがなくなるまで繰り返す. このようにして得られたリンケージを基に, 新たな個体集団を生成する.

3. Tabu Search

Tabu Search⁴⁾ は Glover によって提案された近傍探索法である. たいていの近傍探索法は局所解におちいるとそこから探索が進まなくなるが, Tabu Search は過去の探索の履歴を Tabu List と呼ばれるリストに保存しておき, それを利用することで局所解からの脱出をねらう. ここでは, 近傍は 1 反転近傍としているので, 反転した遺伝子の ID を Tabu List に保存する. Tabu List の長さは Tabu 期間と呼ばれ, この Tabu 期間が Tabu Search の探索能力に影響する.

4. Tabu Search を導入した ECGA

GA と近傍探索法とを組み合わせる手法は古くから研究されてきた手法である⁵⁾. このハイブリッドアルゴリズムは GA の大域的な探索と近傍探索法による局所的な探索によって強力なアルゴリズムとなる.

EDAs に近傍探索法を組み合わせる手法についてもすでに研究がおこなわれており, EDA/L²⁾ などが提案されている. EDAs に近傍探索法を組み合わせる目的の 1 つに, クラシカルな GA と同様, GA の大域的な探索と近傍探索法による局所的な探索によって強力なアルゴリズムとするという目的がある. 本論文では, 近傍探索法として局所解から脱出可能な Tabu Search を用いる. これにより, 近傍探索法の探索能力の向上をねらう. ここでは EDAs のうちの 1 つである ECGA を用いる. Tabu Search は選択したすべての個体に適用する. すなわち, ECGA の (2) のステップ後に Tabu Search を適用する.

以後, Local Search を導入した ECGA を ECGA+LS, Tabu Search を導入した ECGA を ECGA+TS と呼ぶものとする. Local Search には最急勾配山登り法を用いる.

5. 実験

5.1 テスト関数

テスト関数として, 5 ビットの 2 進数からなる 5 ビットトラップ関数 trap_5 , および 6 ビットの 2 進数からなる W-6 ビットトラップ関数 wtrap_6 の 2 つ関数を用いた. 対象問題はそれぞれの関数の和で定義する. trap_5 の和は以下ようになる:

$$F_{\text{trap}_5}(x) = \sum_{i=1}^n \text{trap}_5(u_i) \quad (3)$$

$$\text{trap}_5(u) = \begin{cases} 4 - u, & 0 \leq u \leq 4 \\ 5, & u = 5 \end{cases} \quad (4)$$

ここで, u は 0, 1 からなる長さ 5 の個体中の 1 の数, n は部分関数の個数である. trap_5 の概形を図 2 に示す. 個体長 l は $l = 5n$ であり, 実験では $l = 50, 75, 100, 125, 150$ ($n = 10, 15, 20, 25, 30$) とした.

wtrap_6 の和は以下ようになる:

$$F_{\text{wtrap}_6}(x) = \sum_{i=1}^n \text{wtrap}_6(u_i) \quad (5)$$

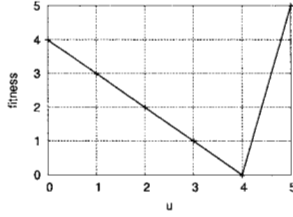


図 2 5 ビットトラップ関数
Fig. 2 the 5-bit trap test function

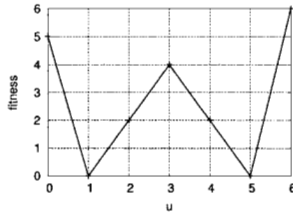


図 3 W-6 ビットトラップ関数
Fig. 3 the W-6-bit trap test function

$$\text{wtrap}_6(u) = \begin{cases} 5 - 5u, & u = 0, 1 \\ -2 + 2u, & u = 2, 3 \\ 10 - 2u, & u = 4, 5 \\ 6, & u = 6 \end{cases} \quad (6)$$

ここで、 u は 0, 1 からなる長さ 6 の個体中の 1 の数、 n は部分関数の個数である。wtrap₆ の概形を図 3 に示す。個体長 l は $l = 6n$ であり、実験では $l = 60, 90, 120$ ($n = 10, 15, 20$) とした。

trap₅ および wtrap₆ はだまし性をもつため、局所的な探索はしばしば局所最適解を与える。したがって、リンケージ学習によって各変数間の依存関係を検出するなどして、ビルディングブロックを破壊することなしに組み合わせる必要がある。

5.2 実験結果

ECGA, ECGA+LS, ECGA+TS の比較をおこなう。実験は SGI Onyx 300 (MIPS R14000 600MHz × 32 CPU, 16GB 共有メモリ) 上でおこなった。比較は、適応度評価、モデル構築などのすべてのコストを考慮に入れた全実行時間でおこなう。ここで示す実験結果は、異なる乱数シードを用いて 10 回実行した結果の平均である。なお、パラメータは最適なものを実験的に選んだ。局所探索をおこなう回数は各世代において 1 個体あたり $2l$ 回、Tabu 期間は trap₅ にお

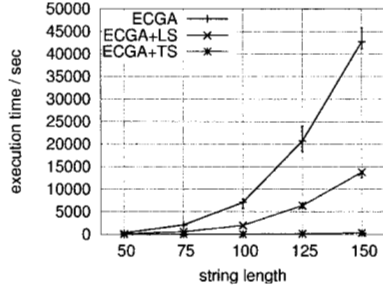


図 4 5 ビットトラップ関数における全実行時間
Fig. 4 Overall execution time for the 5-bit trap test function

いては $0.8l$ 、wtrap₆ においては $0.5l$ とした。

trap₅ の場合の実験結果を図 4 に示す。この図からわかるように、ECGA よりも ECGA+LS および ECGA+TS の方が高速に最適解を得ている。特に ECGA+TS は非常に高速に最適解を得ている。ECGA に比べ ECGA+LS が高速に最適解を得ているのは、ECGA において全実行時間中のほとんどの時間を占めるモデル構築にかかる時間を大幅に短縮できたためである (表 1)。ECGA+LS では新たに Local Search の時間を要するが、短縮できるモデル構築時間に比べればかなり短い。ECGA+TS が非常に高速に最適解を得ているのは、Tabu Search のみで trap₅ を解くことができるためである。次に、wtrap₆ の場合の実験結果を図 5 に示す。ECGA では最適解を得られず、ECGA+LS と ECGA+TS はほぼ変わらない結果となった。wtrap₆ に対しては Tabu Search 単体では有効でないことがわかる。しかしながら、ECGA 単体では wtrap₆ を解くことができていないことから、ECGA に近傍探索法を組み込む手法は wtrap₆ に対して有効であるといえる。

5.3 考察

まず最初に、trap₅ において ECGA+LS および ECGA+TS が高速に最適解を得る理由について考察する。近傍探索法を適用した場合、各部分関数を構成する 5 ビットの部分列は局所最適解 (00000) と大域最適解 (11111) に到達する。Local Search ではこれ以上探索は進まず、各部分列が 2 つの最適解に到達した解からモデルを構築する。各部分列のエントロピーが減少し初期の世代から正しいリンケージ集合が得られるために、高速に最適解を得ることができる。いっぽう、Tabu Search は局所解に到達した時点からさらに探索を続ける。00000 と 11111 を破壊する場合

表 1 5 ビットトラップ関数 ($l = 150$) における各実行時間
 Table 1 Execution time for the 5-bit trap function ($l = 150$)

	ECGA	ECGA+LS	ECGA+TS
集団サイズ	7800	3200	3200
選択時間 (秒)	0.515	0.105	0.002
局所探索時間 (秒)	—	3722.149	338.449
モデル構築時間 (秒)	42758.592	10056.177	0.0
交叉時間 (秒)	27.093	5.765	0.0
適応度評価回数	167700	784838080	72003200
リンケージ同定正解率 (%)	100	100	—
平均世代数	21.5	10.9	1.0
全実行時間 (秒)	42786.2	13797.0	339.0

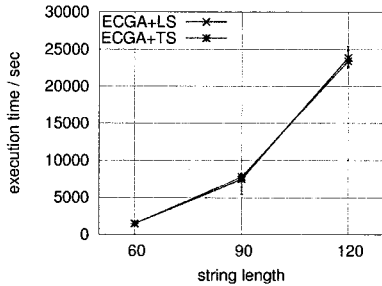


図 5 W-6 ビットトラップ関数における全実行時間
 Fig.5 Overall execution time for the W-6-bit trap test function

に、00000 を破壊するほうが適応度の減少が小さいので、00000 を優先して破壊していく。そうしていくうちに、部分列中に $u = 4$ となる部分列が現れ、最適解 11111 に到達していく。このように、 trap_5 は Tabu Search のみで解けるので、高速に最適解を得ることができる。

次に、 wtrap_6 において近傍探索が有効な理由について考察する。局所解は 000000 となる場合と $u = 3$ となる場合であり、最適解は 111111 となる場合である。これらの適応度はほとんど差がない。0 と 1 で構成される 6 ビットを考えた場合、 $u = 3$ となる部分列は 20 通り存在する (すべての場合は $2^6 = 64$ 通り)。したがって、ECGA では多数の局所解を含む解が選択され、各部分列のエントロピーが大きいためリンケージ同定がされにくい。いっぽう、ECGA+LS および ECGA+TS では、近傍探索法により $u = 1$ となる 6 通りの部分列が 000000 に、 $u = 5$ となる 6 通りの部分列が 111111 に到達する。これにより、各部分列のエントロピーが減少し、リンケージ同定が可能になったと考えることができる。最後に、ECGA+LS と ECGA+TS の結果がほとんど変わらない理由につ

いて考察する。Tabu Search は $u = 3$ となる局所解から抜け出すことができない。なぜならば、 $u = 3$ となる局所解が Tabu Search により $u = 2$ か $u = 4$ となったとすると、次は近傍の中からもっとも良い解を選ぶためかならず $u = 3$ となる局所解に戻るからである。

6. おわりに

ECGA に Tabu Search を組み込んだ手法 ECGA+TS を提案した。 trap_5 および wtrap_6 の 2 つのテスト関数で実験をおこない、 trap_5 に対して ECGA+TS は有効で、 wtrap_6 に対して ECGA+TS は ECGA+LS と変わらない性能を示すことがわかった。また、ECGA+TS が有効である場合と有効でない場合について考察をした。

参考文献

- 1) Martin Pelikan, David E. Goldberg and Fernando Lobo: A Survey of Optimization by Building and Using Probabilistic Models, IlliGAL Report No.99018, *Computational Optimization and Applications*, Vol.21, No.1, pp.5-20 (2002).
- 2) Qingfu Zhang, Jianyong Sun, Edward Tsang and John Ford: Hybrid Estimation of Distribution Algorithm for Global Optimization, *Engineering Computations*, Vol.21, No.1, pp.91-107 (2004).
- 3) Georges Harik: Linkage Learning via Probabilistic Modeling in the ECGA, IlliGAL Report No.99010 (1999).
- 4) Fred Glover: Tabu Search: A Tutorial, *Interfaces*, Vol.20, No.4, pp.74-94 (1990).
- 5) Patrick Mills, Edward Tsang, Qingfu Zhang and John Ford: A Survey of AI-based Metaheuristics for Dealing with Local Optima in Local Search, *Department of Computer Science, University of Essex*, No.CSM-416 (2004).