Vol.2010-MPS-77 No.23
2010/3/5

BOA におけるベイジアンネットワーク構築の効率化に関する検討

堀伸哉[1]　棟朝雅晴[2]　赤間清[2]

[1] 北海道大学工学部情報工学コース、[2] 情報基盤センター

本論文は BOA の改良型アルゴリズム、EBOA（Effective BOA）を提案する。EBOA は BOA のベイジアンネットワーク構築フェイズにおいて、探索する遺伝子ノード数をエントロピーの値によってクラスタリングし、それぞれのクラスターでベイジアンネットワークを構築することで遺伝子の探索時間を減少させる。また、これと同時にエントロピーの値による探索遺伝子数の絞込みも導入する。これら二つの手法を取り入れた EBOA は BOA において問題となる多大な実行時間を短縮することで複雑で巨大な問題の最適化を行うことが可能となる。

An effective Bayesian Network Construction Method for BOA

Shinya Hori[1] , Masaharu Munetomo[2] , Kiyoshi Akama[2]

[1]Faculty of Engnering, and [2]Information Initiative Center Hokkaido University

This paper proposes an Effective Bayesian Optimization Algorithm (EBOA) which improves network construction process of BOA. EBOA performs clustering of locus nodes based on their entropy measures and at the same time, removes loci that are not necessary to be modeled based on their entropy. After the clustering, it constructs a Bayesian Network in each cluster to reduce the running time for searching networks for BOA to solve large and complex optimization problems.

1 はじめに

EDA（Estimation of Distribution Algorithm）[1]は確率分布を用いることで従来のGAでは解くことが困難な問題の効率的解決を目指すアルゴリズムである。数あるEDAの中で複雑な問題を正確に解くことができると言われているのがベイジアンネットワークを確率モデルとして使用する、BOA(Bayesian Optimization Algorithm)[2]である。BOAは遺伝子間の依存関係をベイジアンネットワークの条件付確率によって記述するアルゴリズムで、複雑な遺伝子の関係を記述することが可能な手法である。BOAの長所は各遺伝子が複雑に影響しあうような問題を解けることだが、ベイジアンネットワーク構築の時間が遺伝子長に対応して大きく増大することが短所として挙げられている。BOAは遺伝子間のベイジアンネットワークを構築する際に全遺伝子の組合せを貪欲に探索するので、計算量が$O(L^3)$ほどかかり、問題の大きさに対応して計算時間が大きく増大していく。

本研究では BOA の個体選択後の各遺伝子座のエントロピーの値を計測し、その値によって遺伝子座のクラスタリングを行い、各遺伝子座クラスターにおいてベイジアンネットワークの部分グラフを作ることで実行時間の削減を目指す。それと同時に、エントロピーの値による探索遺伝子数の絞込みも行う。これら二つの手法を導入して作られた EBOA（Effective BOA）によって BOA の実行時間の減少を目指す。

2　BOA(Bayesian Optimization Algorithm)

BOA は確率分布と個体群の生成・評価・選択を利用して最適解を発見する分布推定アルゴリズムである。確率モデルとしてベイジアンネットワークを使用し、遺伝子間の依存関係を条件付確率で記述する。ベイジアンネットワークは遺伝子間の依存関係を詳細に記述するので、複雑な問題の解決が可能となっている。BOA のプロセスは Figure.1 によって記述されている。

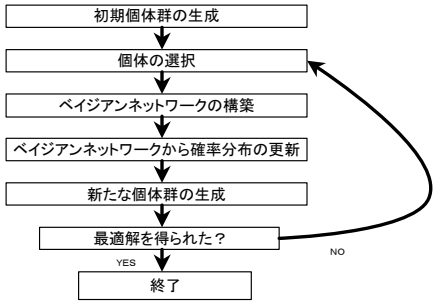

Figure.1　BOA のフローチャート

Vol.2010-MPS-77 No.23
2010/3/5

BOA はまず与えられた初期確率分布を基に解空間のビット列個体の値を標本化し、個体を生成する。次に生成された個体群の各個体を評価関数に従って評価し、個体群の中で評価値が高い個体を選択して生存させる。そして選択された個体の遺伝子について K2 法で遺伝子間の依存関係を探索し、現在の遺伝子列にとって最適であるベイジアンネットワークを構築することで遺伝子間の依存関係を条件付確率で記述する。Figure.2 はベイジアンネットワークの例を描いており、各ノード Xi は BOA における各遺伝子を表現している。ベイジアンネットワーク構築の終了後、構築されたベイジアンネットワークに従って確率分布を更新する。この時、他の遺伝子と依存関係が存在しないと思われる遺伝子に対しては独立な確率分布が生成され、他の遺伝子と依存関係のある遺伝子に対してはその遺伝子との条件付確率が生成される。この時点で最適解が得られていない場合、更新した確率分布から新たに子個体群を生成し、個体の評価に戻る。以上の動作を繰り返す中でベイジアンネットワークと確率分布を更新し、最適解を見つけていく。

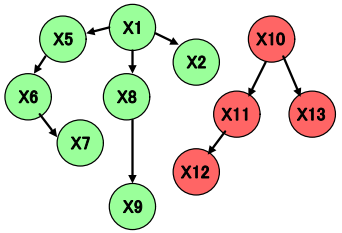

Figure.2　ベイジアンネットワーク

3　ベイジアンネットワークの構築

BOA は複雑な問題に対応できるアルゴリズムだが、大規模問題における実行時間が欠点として挙げられている。問題が大きいほど実行時間も対応して大きく上昇していくので現状では複雑で大規模な問題を BOA によって解決することは難しく、このような問題を解くためには BOA の実行時間を減少させることが必要となる。BOA ではベイジアンネットワーク構築フェイズが実行時間の大半を占めている。よって、BOA の実行時間を減少させるためにはベイジアンネットワーク構築フェイズの実行時間を減らすことが有効であると言える。

BOA の改良法を示す前に BOA のベイジアンネットワーク構築の流れである K2 アルゴリズムについて説明する。BOA は各世代で正確なベイジアンネットワークを構築するために全ての遺伝子が他の全ての遺伝子と依存関係があるのかどうかを照合していき、最適なベイジアンネットワークを構築していく。K2 アルゴリズムの流れは Figure.3 の通りである。

（１）ある遺伝子Xiがある遺伝子Xjと依存関係をもっていると
　　仮定した時のBICスコアを全ての遺伝子の組について計算する。

（２）全ての遺伝子組の中でBICスコアの最も高い遺伝子組にエッジを張る

（３）エッジを保存したままで全ての遺伝子組についてBICスコアを計算する。

（４）もし、全てのスコアが負の数だったら終了する。
　　そうでなければ（２）へ行く。

Figure.3　Ｋ２アルゴリズム

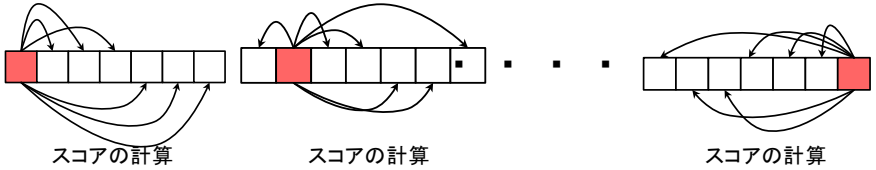

Figure.4　ベイジアンネットワーク構築のためのスコア計算

（１）で遺伝子Xiと他の遺伝子Xjとの間のスコアScore(Xi,Xj)を計算する。このスコアは現在の遺伝子Xiが遺伝子Xjと依存関係があった場合にどの程度の利益があるのかを示すスコアであり、これを全ての遺伝子において計算する。このスコアには以下のBICメトリック[3]が使用される。

$$BIC(B)=\sum_{i=1}^{n}\left(-H(Xi\,|\,\Pi i)N-2^{|\Pi i|}\frac{\log_2(N)}{2}\right) \qquad (1)$$

ここでの Xi は現在見るノード、Πi は Xi の親ノードの集合、N は個体群の個数を示している。ある遺伝子 Xj を Xi の親ノード集合Πi の要素として追加した時の BIC（B）を求めることで Xi と Xj の間に依存関係があると考えた時のベイジアンネットワークのスコアとしている。（２）で、（１）で求めた各遺伝子ペアのスコア Score(Xi,Xj)の中で最も高いスコアをもつ遺伝子ペア Xmax1,Xmax2 を抽出し、その遺伝子間にエッジを結ぶ。そして、エッジを結んだ二つの遺伝子 Xmax1,Xmax2 間の依存関係を保存した状態で BIC スコアをそれぞれの遺伝子の組み合わせでもう一度計算する。これらの動作を BIC スコアの値が負になるまで繰り返していき、最適なベイジアンネットワー

クを探索していく。

以上からベイジアンネットワーク構築フェイズで各遺伝子同士の BIC スコアを計算する時に遺伝指数に対応して $O(L^3)$の計算量がかかっているのがわかる。よってベイジアンネットワーク構築に必要な実行時間を減らすためには探索する遺伝子数を減らす必要がある。

4　提案手法　EBOA(Effective BOA)

BOA は各世代でベイジアンネットワークを構築しており、その各構築フェイズで全ての遺伝子を探索して最適なベイジアンネットワークを構築していく。このベイジアンネットワーク構築フェイズにおいて $O(L^3)$で実行時間は上昇していくので、このフェイズの実行時間を減らすためにはベイジアンネットワーク構築のための探索遺伝子数を減らすことが必要である。ここでベイジアンネットワーク構築フェイズでは全ての遺伝子を探索する必要が必ずしもあるわけではない、ということに注目する。BOA のベイジアンネットワークは Figure.2 を見ればわかるように全てのノードをエッジで結ぶわけではなく、いくつかの部分ベイジアンネットワークを作ることで問題のビルディングブロックを表現している。つまり、ある遺伝子 Xi にとって全ての遺伝子とのスコアを計算する必要はなく、同一のベイジアンネットワークを構築しないと思われる遺伝子 Xj との間のスコアの計算を削減することができれば、探索する遺伝子数を減少し、ベイジアンネットワークの構築に必要な時間を減少することができる、本研究では同一のベイジアンネットワークを構築すると思われる遺伝子をクラスタリングによって分割し、それぞれのクラスターでベイジアンネットワークの部分グラフを構築する。この時のクラスタリングの尺度にはエントロピーを利用する。また、遺伝子座のクラスタリングとは別に、エントロピーの値による探索遺伝子絶対数の絞込みも同時に行う。これら二つの手法を取り入れた EBOA で、最適解へ到達するまでに必要な実行時間の削減を目指した。以下でその二つの手法と EBOA についての説明を行う。

4-1　エントロピーを利用した遺伝子座のクラスタリング

本研究ではk-means法[4]を使用して遺伝子座のクラスタリングを行う。クラスタリングの尺度としてエントロピーを使用する。主なプロセスの流れはFigure.5 の通りとなる。

（1）各遺伝子座のエントロピーの値を計算する
（2）全ての遺伝子座に1からkまでのクラスター番号をいずれか一つランダムに与える
（3）同じクラスター番号をもつ遺伝子座を集める
（4）各クラスターの遺伝子座のもっているエントロピーの値の平均をとる
（5）各遺伝子座のエントロピーと各クラスターのエントロピーの平均を比較する
（6）各遺伝子座は自身のエントロピーの値と最も近いクラスターを選択する。
（7）選択したクラスターにその遺伝子を所属させる
（8）更新されたクラスターのエントロピーの平均をとる
（9）クラスターの移動がなくなるまで（5）～（8）を繰り返す
（10）各クラスター内でBICスコアを独立に計算する
（11）各クラスターでBICスコアに基づいて部分ベイジアンネットワークを構築する

Figure.5　遺伝子座のクラスタリング

まず、各遺伝子座のエントロピーの値を計算する。次にこのエントロピーの値を利用して遺伝子座のクラスタリングを行う。以下でそのクラスタリングの詳細を説明する。まず全ての遺伝子座にランダムにクラスター番号 1 から k のいずれかを割り振る。そして各クラスター内の遺伝子のエントロピーの平均値を計算し、その平均値に最も近いエントロピーの値をもつ遺伝子がそのクラスターに所属するように再度遺伝子の割り振りを行う。その後、もう一度各クラスターのエントロピーの平均値を求めて遺伝子座を再配置する、という動作を各クラスターに変動がなくなるまで繰り返す。これによって遺伝子座は k 個のクラスターへと分割され、各クラスター内でベイジアンネットワークの部分グラフを構築することができる。クラスタリングの終了後、各クラスターで遺伝子の依存関係のスコア Score(Xi,Xj)を計算し、それぞれのクラスターで最適なベイジアンネットワーク部分グラフを構築し、最後に全てのベイジアンネットワーク部分グラフを組み合わせて確率分布を更新する。これによって各クラスターの遺伝子の探索数は減少し、実行時間は削減される。Figure.6 でクラスタリングの様子が描かれている。遺伝子座がクラスタリングされた後で、各クラスターがベイジアンネットワークを構築しているのがわかる。

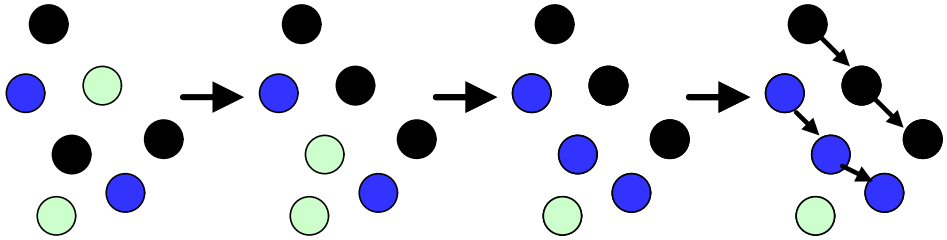

Figure.6　遺伝子座のクラスタリング及びベイジアンネットワークの構築

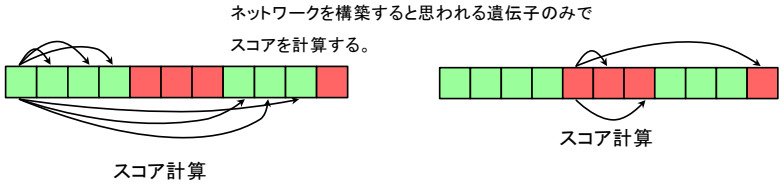

Figure.7　各クラスタリングで BN を構築する。

EBOA では Figure.3 のように全ての遺伝子間のスコアを計算するわけではなく、Figure.7 のように各遺伝子にとって探索が必要な遺伝子のみを探索することで無駄な遺伝子に対するスコア計算のコストを削減し、従来の BOA よりも計算量と時間を削減することができる。

4-2　エントロピーの値による探索遺伝子の絞込み

クラスタリングとは別に、本研究ではエントロピーの値がある値 α ～ β の範囲に収まらないような遺伝子をベイジアンネットワーク構築のための探索フェイズから取り除く。例えばエントロピーの値が 0 であるような遺伝子は値が収束しているのでネットワークを構築することはない。従来の BOA はこのような遺伝子も探索フェイズでスコアを計算し、ネットワークを構築するかどうかを探索している。このような無駄な探索を防ぐために、あるエントロピーの値に満たない遺伝子はクラスタリングとは別に探索フェイズそのものから外すことにした。また、エントロピーの値が 1.0 に近い遺伝子は、個体群におけるその遺伝子の値は 0 と 1 がほぼ半々となり、このような遺伝子の値は選択であまり考慮されないと考えられる。よってこれらの遺伝子も正確なベイジアンネットワークを構築しないので、遺伝子の探索から排除することができる。以上のことを踏まえて、本研究ではエントロピーの値が極端に大きい、もしくは極端に小さいような遺伝子は値が収束している、もしくは正確なネットワークを構築しない、と考えることができ、これらの遺伝子を探索から排除する。これはエントロピーの値がある範囲にある場合のみだけ遺伝子の探索を行うと言い換えることができるので、本研究では遺伝子のエントロピーの値がある上限 α と下限 β の間に存在している時だけ、遺伝子探索を行った。これによって無駄な遺伝子の探索のための計算をアルゴリズムから排除し実行時間のさらなる減少を目指す。

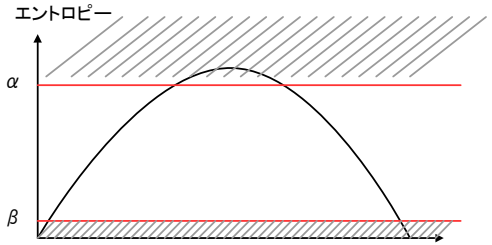

Figure.8 エントロピーの値による遺伝子の絞込み

これら 2 つの手法（エントロピーによる遺伝子座のクラスタリング、エントロピーによる探索遺伝子の絞込み）を同時に取り入れて作成したアルゴリズムを EBOA(Effective BOA)と名づけた。EBOA の概略図を以下に記載する。

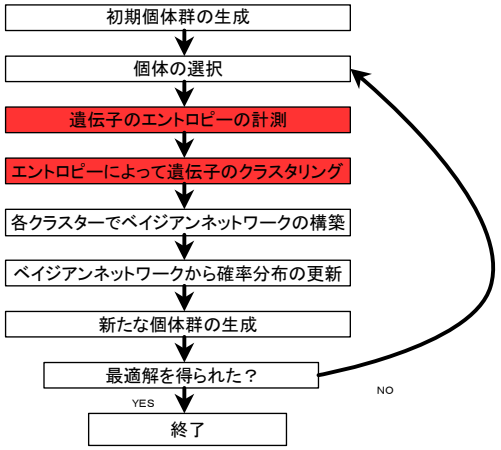

Figure.9　EBOA のフローチャート

EBOA は初期個体群をランダムに生成し、評価関数に従って選択を行う。ここまでは通常の BOA と同じ流れを踏むが、選択が終わるとそれぞれの遺伝子のエントロピーを計測する。そして k-means 法を利用して遺伝子座のクラスタリングを行う（遺伝子座のクラスタリング）。この時、エントロピーの値が α ～ β の範囲から外れている遺伝子はどのクラスターの中にも含めず、探索から排除する（探索遺伝子の絞込み）。そしてそれぞれのクラスターでベイジアンネットワークの部分モデルを構築し、各部分モ

デルを統合することで確率分布を更新し、これを基に個体を再生成する。EBOA はこれらの作業を繰り返して少ない実行時間で最適解を導く最適化アルゴリズムである。次章で EBOA の効果の実験を行うための評価関数を説明する。

5 評価関数

EBOA の実験のために使用する関数として巨大で複雑で、評価関数の重みの差が大きい問題を選択する。今回は Trap5 関数を n 個つなげ、それぞれの Trap5 関数に 5 ブロックごとに指数的な重み scale を乗算した関数を評価関数として使用した。

$$f(s)=\sum_{i=0}^{n} scale(i)\times trap_5(s_i) \qquad (2)$$

$$scale(i)=\begin{cases}1 & ,(0<i<6\\ 2 & ,(5<i<11\\ 4 & ,(10<i<16\\ 8 & ,(15<i<21\\ 16 & ,(20<i<26\cdots\end{cases}$$

$$trap_5(u_i)=\begin{cases}4-u_i & \text{if}\quad 0\le u_i\le 4\\ 5 & \text{if}\quad u_i=5\end{cases}$$

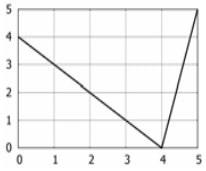

Figure.10 Trap5 関数

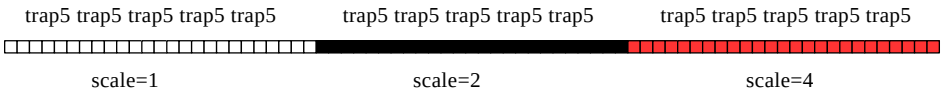

Figure.11 遺伝子列 s の評価関数 f(s)

Figure.10 のように Trap5 関数は 5 つの遺伝子を見て、5 つの遺伝子の値が全て 1 である時は評価値 5 を返し、それ以外の時は 5 つの遺伝子中に含まれる 0 の値が評価値となる。この Trap5 関数に 5 ブロックごとに変数 scale をかける。この変数は Trap5 関数をさらに 5 ブロック毎に分割する役割をもっており、5 ブロックの Trap5 関数毎にこの scale の値は二倍される。Figure.11 はこの関数の評価値のグループ分けを示しており、Trap5 が 5 つずつグループになっていることがわかる。この変数によって評価値が指数的に増大していくのでビルディングブロックに格差が存在する問題となり難易度が向上する。ここではこの関数を指数 Trap5 関数と名づけた。次章ではこの指数 Trap5 関数を使用して EBOA と BOA の性能を実際に比較する。

6 実験

指数 Trap5 関数を評価関数として BOA と EBOA の性能比較を行う。個体群の大きさは 10000 を起点として最適解を出せるまで 1000 ずつ上昇させる。収束条件は最適解を見つけること、もしくは遺伝子が全て収束することのどちらかとする。実験には Intel Core(TL2)DUO CPU の 2.80GHz を使用した。また、パラメータとして α =0.7, β =0.001,k=2 と設定した。これは、この評価関数においてはこのパラメータの値が最適解までの実行時間が最も短いからである。以上の条件下で指数 Trap5 関数の遺伝子長を 100 から 500 まで増大したさせて BOA と EBOA を動かし、その実行時間の比較を行った。また、提案した二つの手法単体での効果も見るために遺伝子座のクラスタリングのみを実装した EBOA、エントロピーによる値の絞込みのみを実装した EBOA も同様の条件化で実験を行い、計 4 つのアルゴリズムの比較を行った。

Table.1 指数 Trap5 関数での BOA の実験結果

BOA	遺伝子	個体群	実行時間(秒)	世代	評価数
	100	13000	53.561	42	286000
	200	42000	1077.225	71	1533000
	300	70000	5624.757	99	3535000
	400	97000	17625.428	127	6256500
	500	120000	41633.944	155	9420000

Table.2 指数 Trap5 関数での EBOA の実験結果

EBO	遺伝子	個体群	実行時間(秒)	世代	評価数
	100	13000	46.289	43	295000
	200	43000	614.655	71	1569500
	300	70000	3069.476	99	3535000
	400	97000	9715.808	129	6353500
	500	120000	21501.625	154	9360000

Table.3 指数 Trap5 での BOA と EBOA の実行時間（秒）の比較

遺伝子長	BOA	EBOA	スピードアップ率
100	53.561	46.289	13%
200	1077.225	614.655	43%
300	5624.757	3069.476	45%
400	17625.428	9715.808	45%
500	41633.944	21501.625	48%

スピードアップ率＝(BOA の実行時間-EBOA の実行時間)／(BOA の実行時間)

Vol.2010-MPS-77 No.23
2010/3/5

Table.4　指数 Trap5 関数でのクラスタリングのみと遺伝子絞込みのみの実行時間（秒）の比較

遺伝子長	BOA	EBOA	クラスタリング	遺伝子絞込み
100	53.561	46.289	40.073	45.527
200	1077.225	614.655	849.675	796.048
300	5624.757	3069.476	3720.162	3698.663
400	17625.428	9715.808	11610.373	10861.231
500	41633.944	21501.625	26812.613	24759.124

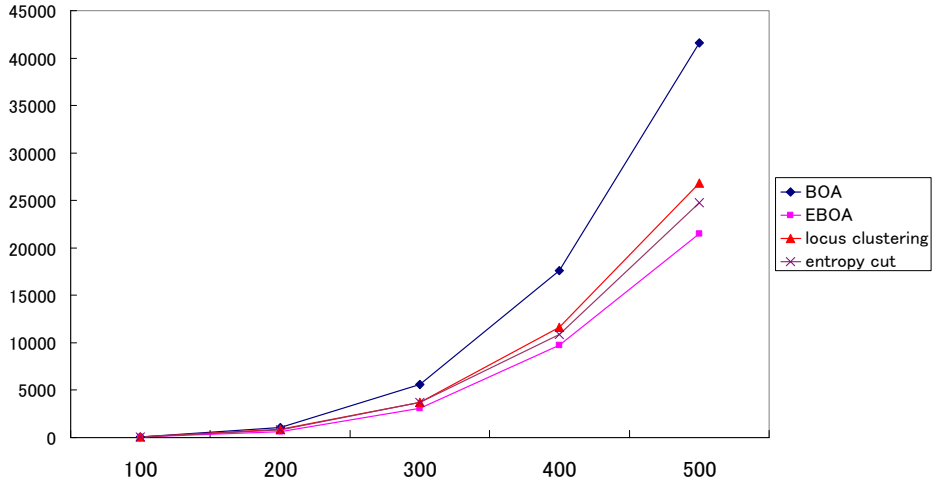

Figure.12　BOA と EBOA、各操作のみのアルゴリズムの比較グラフ
横軸は遺伝子長、縦軸は実行時間（秒）を示している。

Table.1 は遺伝子長が 100 から 500 まで増加した時の指数 Trap5 関数を解く時の BOA の性能を示した表である。遺伝子長が増大するほどに実行時間が大きく上昇している事がわかる。Table.2 は同条件下で EBOA を使用して最適解を導いた時の性能を示している。EBOA も実行時間は大きく上昇しているが、EBOA の実行時間は BOA の実行時間から計算して 50%程度減少していることがわかる。世代数と計算量をほとんど削ることなく実行時間を減少させることができた事から、EBOA は解の精度を落とすことなく少ない実行時間で最適解を導き出すことができたと言える。二つのアルゴリズムの実行時間は Table.3 で比較されている。スピードアップ率は BOA を基準として EBOA の実行時間は何%削減されたのかを示している。この表ではスピードアップ率は遺伝子長が 500 である時に最大 48%まで上昇している。Talbe.4 で「遺伝子座のクラスタリング」と「エントロピーによる遺伝子の絞込み」の二つの動作のうち、片方だけを実装した時のアルゴリズムそれぞれと BOA、EBOA の実行時間の比較を示している。これら二つの動作は単体だけの実装でも実行時間の削減に成功しているが、両方を組み合わせた EBOA よりも実行時間がかかっている。BOA と EBOA そして遺伝子座のクラスタリングのみを利用した EBOA、エントロピーによる遺伝子の絞込みだけを利用した EBOA の実行時間を比較するグラフを Figure.12 で示す。これから遺伝子長が増加すればするほど BOA と EBOA の間の差が広まっていくことがわかる。

EBOA の BOA からのスピードアップ率は遺伝子長が伸びるほどに上昇するわけではなく、最高で 48%程度に留まった。これは遺伝子長が大きい場合、遺伝子探索を 2 つのクラスターに分割したとしてもそれぞれのクラスターで探索する遺伝子数は大きいので、最適解の発見までに必要な実行時間は結果的に大きくなってしまうからだと考えられる。クラスター数をさらに増加させると実行時間を削減できると思われたが、クラスター数を増加させて実験をするとベイジアンネットワークの精度が落ち、収束までの世代数が増加するので、指数 Trap5 関数では実行時間はクラスター数 k=2 である時が最も実行時間が少なかった。これはクラスター数を増加することで遺伝子を細かくクラスタリングすることになり、クラスタリングのミスを引き起こすからだと考えられる。これからの課題はどのようにベイジアンネットワークの精度を落とさずにクラスター数を増加させるかということが挙げられる。

7　まとめ

本研究は BOA の実行時間の削減のための二つの操作を新たに実装した EBOA を提案した。一つ目の操作はエントロピーの値によって遺伝子座をクラスタリングして、ベイジアンネットワーク構築のための遺伝子探索数を各クラスターに分割する操作、もう一つはエントロピーの値がある範囲に含まれないような遺伝子を探索から除外するという操作である。これら二つの操作を同時に取り入れた EBOA は、最大で BOA の半分の速度で最適解を見つけることに成功した。しかし、大規模問題に対応できるほど実行時間を減らすことができたとは言えず、EBOA も BOA と同様に実行時間は大きく上昇しているので、現状では大規模問題を解くことは難しい。

今回のアルゴリズムではエントロピーの値によるクラスター数 k=2 と設定して実験を行った。これは数度の実験結果より、この関数においては正しいベイジアンネットワークを構築するクラスター数は 2 であると確認したからであるが、クラスタリングの精度を維持しつつクラスター数を上昇させることができれば、さらに高い効率を目指すことが可能であると考えられる。また、もう一つのアプローチとしてデータクラスタリングを同時に導入することで、エントロピーの値をデータのクラスター毎に偏

らせ、各ネットワークの信頼度を上昇させることが考えられる。今後はこのようなアプローチでアルゴリズムを改良していき、さらに大規模な問題に BOA を適用させることができるように、さらなる実行時間の減少を目指していきたい。

参考文献

1) Pedro Larranage and Jose A.Lozano: Estimation of Distribution Algorithms a New Tool for Evolutionary Computation, Kluwer Academic Publishers (2002)

2) 棟朝雅晴 ：遺伝的アルゴリズム -その理論と先端的手法- ,森北出版(2008)

3) Martin Perikan, Kumara Sastry, Erick Cantu-Paz: Scalable optimization via Probabilistic Modeling, Hierarchical Bayesian Optimization Algorithm, pp63-90, Springer (2006)

4) 上田尚一：クラスター分析,朝倉書店(2003)