

並列リンケージ同定と並列 BOA に関する性能比較

村尾 直哉[†] 棟 朝 雅 晴^{††} 赤 間 清^{††}

通常の遺伝的アルゴリズム (Genetic Algorithms, GA) で解くことが困難な問題に対する解法として、リンケージ同定手法や BOA (Bayesian Optimization Algorithm) などが提案されている。しかし、これらのアルゴリズムは有用ではあるが、計算コストが高いという欠点がある。このコストを小さくするための並列化の研究がいくつか行われているが、解くべき問題に対してどの手法が有用であるかが明らかではない。本稿では、近年提案されている並列リンケージ同定と並列 BOA を扱い、これらの性能を比較することで、問題の性質から適用すべき手法を選択する基準について検討する。

Performance comparisons between parallel linkage identification and parallel BOA

NAOYA MURAO,[†] MASA HARU MUNETOMO^{††} and KIYOSHI AKAMA^{††}

The linkage identification and the Bayesian Optimization Algorithm are proposed to solve GA-difficult problems. These algorithms are competent, while their computational cost is high. Recently, parallelized version of linkage identification and BOA are studied to decrease time to obtain solutions. However, it is no clear which algorithms should we use to solve the problem. In this paper, we compare the performance of parallel linkage identification to that of parallel BOA, and then we consider a relation between the nature of problem and their performances to obtain a guideline to select parallel GAs.

1. はじめに

進化的計算の一種である遺伝的アルゴリズム (Genetic Algorithm, GA) が最適化問題を解くアルゴリズムとして着目されているが、通常の GA では解く事が困難な問題 (符号化が適切に行われていない, など) が存在することが知られており、そのような問題を解くためのアルゴリズムとして、関連する遺伝子座を同定し、まとめて扱う手法がいくつか提案されている。そのアルゴリズムとして、GA の前処理として関連する遺伝子座をリンケージとしてまとめて扱うリンケージ同定手法 (LINC [4], LIMD [6] など) や、集団中の個体の分布から遺伝子座ネットワークを構築する分布推定アルゴリズム (Estimation of Distribution Algorithm, EDA) などがある。これらのアルゴリズムは、通常の GA では解くことが困難な問題において、最適解を求めることが可能になるという点で有用であるが、リンケージ同定やネットワーク構築などの計算コストが高く、実行時間に大きな影響を及ぼす場合がある。そのため、これらのアルゴリズムに関する並列化の研究が近年すすめられており、リンケージ同定の並列化 (以後、並列リンケージ同定) により実行時間を大きく減少で

きることを示されている [5]。また、EDA の一種である BOA (Bayesian Optimization Algorithm) [2] の並列化が Ocenasek らにより行われており、ネットワーク構築のコストを減少させることができている [1]。

これらの結果を受けて、本稿では並列リンケージ同定と並列 BOA に関していくつかの問題を取り上げ、その性能について考察し、どちらのアルゴリズムがどのような問題に適しているかを調査する。実験結果により、GA を適用するユーザが問題を解く際に、その問題の性質から採用すべき並列アルゴリズムを知るための手がかりを得ることが期待される。

2. 並列リンケージ同定

通常の GA において、符号化が適切に行われていなければ組換えがうまくいかず、適切に解を探索することができないという欠点がある。それを解決するために、通常の GA の処理を行う前に、関連する遺伝子座をまとめてリンケージとして扱うリンケージ同定手法が提案されている。リンケージ同定手法の一種である LINC (Linkage Identification by Nonlinearity Check) [4] では、各遺伝子座のペアについて、それぞれの遺伝子座を摂動させた個体との適応度の差分を考え、それが非線形であるかどうかを調べる手法である。具体的には、ある個体 s の遺伝子座のペア (i, j) について考える。ここで、個体 s の遺伝子座 i を摂動させた (個体がバイナリ表現される場合は、遺伝子座 i のビットを反転させた) 個体 s_i との適応度の差分 Δf_i

[†] 北海道大学工学研究科システム情報工学専攻
Division of Systems and Information Engineering,
Graduate School of Engineering, Hokkaido University
^{††} 北海道大学情報基盤センター 大規模計算システム研究部門
Division of Large-Scale Computational Systems, Information Initiative Center, Hokkaido University

を考える。同様に遺伝子座 j について Δf_j を得る。また、遺伝子座 (i, j) をともに摂動させた個体 s_{ij} と個体 s との適応度の差分 Δf_{ij} を考え、これらの値が、 $\Delta f_{ij} \neq \Delta f_i + \Delta f_j$ である遺伝子座のペア (i, j) をリンケージとして扱うことにする。

この処理をすべての遺伝子座の組について計算してリンケージ集合を生成し、これらの遺伝子をまとめて扱うことで適切に解を探索できるようになる。しかし、このリンケージ同定はすべての遺伝子座の組について調べているので、そのコストは $O(nl^2)$ となることが知られている (n は集団サイズであり、ビルディングブロックの大きさ k に対して $n \sim O(2^k)$ である。 l は個体長)。これは問題サイズが大きくなればそのコストが指数的に増加し、実行時間が非実用的になってしまう可能性があることを示している。しかし、各遺伝子座のペアについて非線形であるかどうかを調べる計算は (i, j) 以外の他の遺伝子座から独立しているため並列化が容易であり、このリンケージ同定を並列化した手法が提案されている [5]。並列リンケージ同定 (以後、pLINC と呼ぶ) のアルゴリズムは以下のようになる。

- リンケージ集合の初期化
- 各プロセッサへ遺伝子座の組の分配
- 各プロセッサにおいて非線形性検出
- 結果集約・リンケージ集合生成

また、リンケージ同定により得られるリンケージ集合内部に対して通常の GA オペレータを適用する処理も並列化可能であり、実行時間を減少させることができる。この pLINC はその計算が独立しており、並列化が容易であることから、その台数効果は線形に近くなることが分かっている [5]。本稿では、この pLINC と次節で説明する並列 BOA との性能比較を行っていく。

3. 並列 BOA

通常の GA では困難な問題を解くための別のアルゴリズムは、集団中の優れた個体の分布をモデル化し、それをを用いて次世代の個体を生成する EDA と呼ばれる手法である。本稿では EDA の中でも特にベイジアンネットワークを用いたアルゴリズムである BOA (Bayesian Optimization Algorithm) [2] を扱う。

BOA は分布推定アルゴリズム (EDA) の一種で、ベイジアンネットワークを用いて次世代の解の候補を生成していく手法である。一般的な EDA のアルゴリズムは以下のとおりである。

- (1) 初期集団の生成
- (2) 集団の分布をもとに確率モデルを生成
- (3) 確率モデルから次世代の個体群を生成
- (4) 終了条件が満たされるまで 2, 3 を繰り返す

BOA では、長さ l の個体において各遺伝子は依存グラフの 1 つのノードを表現する。各変数 X_i に対して、変数の集合 Π_{X_i} が定義され、個体の分布は、

$$p(X) = \prod_{i=0}^{n-1} p(X_i | \Pi_{X_i})$$

で表現される。一般に、ネットワーク内の X_j から X_i へのエッジが存在することは、変数 X_j が集合 Π_{X_i} に含まれることを意味している。また、ネットワークの空間を減少させるために、各ノードへの入力エッジ数が k 個までに限定される。

この BOA において、ネットワーク構築に要する時間は全体の実行時間の大部分をしめており、近年、このネットワーク構築の並列化に関する研究が行われている。しかし、このネットワーク構築において、各遺伝子間に依存関係が存在するため並列化が容易ではない。Ocenasek らの論文 [1] では、順列を用いて追加可能なエッジの方向を固定することで、ネットワーク構築の並列化を行っている。

まず、各世代の初めにおいて $0, 1, \dots, l-1$ のランダムな順列が生成され、 $perm$ 配列に保存される。各プロセッサは同じ順列を持っており、ネットワークのすべてのエッジの方向はこの順序に従う。すなわち、 X_j から X_i へのエッジの追加は、 $perm[j] < perm[i]$ の場合に行われる。このようにすることで探索空間が減少してしまうが、各世代で新たな順列を用いることにより、これを補正する仕組みをとる。

このようにしてネットワーク構築の方向付けをすることで、各ノードから他のノードへエッジが引けるかどうかを調べる処理は、他のノードにあまり依存しなくなる。

また、Ocenasek らにより次世代個体群の生成方法の違いから PBOA (Parallel BOA)、DBOA (Distributed BOA) が提案されている。PBOA では、子個体の生成は l 個に並べたプロセッサに各遺伝子座を対応させ、各プロセッサが各遺伝子をそれぞれ決定する線形パイプライン処理を行う。例えば、 l 個のプロセッサを用いているなら、 i 番目のプロセッサは $(i-1)$ 番目のプロセッサから i 個の遺伝子が決定された個体を受け取り、変数 X_k の値を決定する ($perm[k] = i$)。一方、DBOA では各プロセッサがそれぞれいくつかの個体を分散して生成する方法をとる。この個体の生成方法からわかるように、PBOA では各遺伝子座にプロセッサを割り当てるために、各遺伝子座にプロセッサを割り当てられるほどの多くのプロセッサを持つ共有メモリ型の計算機に適した手法であり、DBOA は個体生成を分散して行うために分散環境においても性能が高い手法であるといえる。

本稿では、計算機環境から DBOA を扱うことにし、この DBOA と前節の pLINC についての性能比較を行うことにする。

4. 実 験

この節では、これまでに説明した pLINC と DBOA に関する性能比較を行う。対象関数として、Ocenasek らの論文 [1] で使用された 5 ビットトラップ関数と OneMax 関数、またその他に 3 ビットのトラップ関数を用いることにする。

k ビットトラップ関数は、 n 個の k ビット部分列 s_i を持つ個体 $s = \{s_1, s_2, \dots, s_n\}$ に対して、以下の式で表現される。

$$F(s) = \sum_{i=1}^n F_{trap}(u_i) \quad (1)$$

$$F_{trap}(u_i) = \begin{cases} k & \text{if } u_i = k \\ k - 1 - u_i & \text{else} \end{cases} \quad (2)$$

ここで、 u_i は k ビット部分列 s_i に含まれる '1' の数である。また、OneMax 関数は文字列内の '1' の数が適応度となる関数である。本稿では、この 3 つのテスト関数を用いて、pLINC と DBOA の性能を、その台数効果と実行時間の観点から比較する。すべての実験は共有メモリ型並列計算機の SGI Onyx300(MIPS R14000/500MHz×32CPU/16GB 共有メモリ) 上で行われ、それぞれ試行回数 20 回の平均値を記録している。

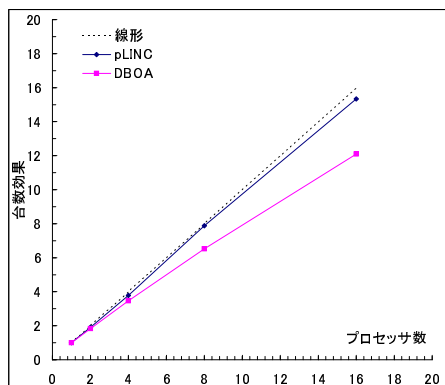


図 1 5 ビットトラップ関数における台数効果

まず、5 ビットトラップ関数について、その台数効果を求める実験を行う。個体長を 105 ビットとし、実験を行った結果が図 1 である。この結果からわかるように、pLINC の台数効果が非常に高いことがわかる。これは、pLINC のほとんどすべての処理が高い独立性を持っているためである。前の節でも説明したように、遺伝子座の非線形性を検出する処理は、それぞれの遺伝子座のペアについてのみ関係するものであり、その他の遺伝子座には何の影響も及ぼすことは無い。そのため、リンケージ同定は線形に近い台数効果を生み出すことができている。また、通信頻度が低いことも台数効果を高めている要因である。一方で、DBOA におけるネットワーク構築では遺伝子座のペアだけで

はなく、いくつかの遺伝子座が関係している。このことから、DBOA の独立性は pLINC より低いと言える。

次に、OneMax 関数と 3 ビットトラップ関数、5 ビットトラップ関数について、DBOA のネットワーク構築時間や pLINC における非線形性の検出に要する時間とその他の計算時間を比較したものが図 2, 3, 4 である。それぞれの手法を比較するために個体長を 105 ビットとして評価する。

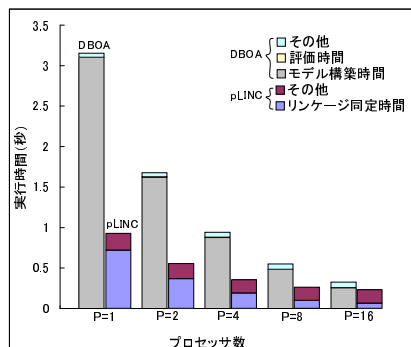


図 2 OneMax 関数におけるリンケージ同定時間とその他の実行時間の比較 (左) とネットワーク構築とその他の実行時間の比較 (右)

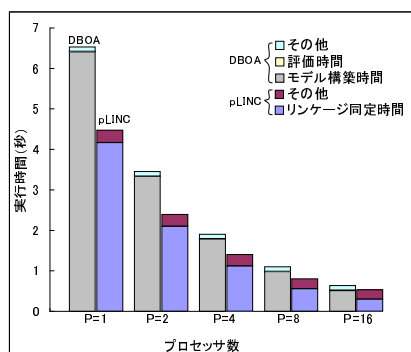


図 3 3 ビットトラップ関数におけるリンケージ同定時間とその他の実行時間の比較 (左) とネットワーク構築とその他の実行時間の比較 (右)

この結果からわかるように、2 つの手法における実行時間を比較すると pLINC がより高速に解を求めていることがわかる。この要因として考えられることは、各対象関数における 1 個体の評価時間は非常に小さなものであり、pLINC の非線形性検出の時間が非常に小さいことがあげられる。ここで簡単化のために、1 プロセッサ時の実行における pLINC の非線形性検出にかかるコストを考えると、 $\frac{l(l-1)}{2}nT_f$ と書ける (個体長 l , 集団サイズ n , 1 個体の評価時間 T_f)。これより、非線形性検出のコストは評価時間 T_f と個体長 l , 集団サイズ n に依存していることがわかる。一方で、DBOA のネットワーク構築時間は個体の評価時間 T_f

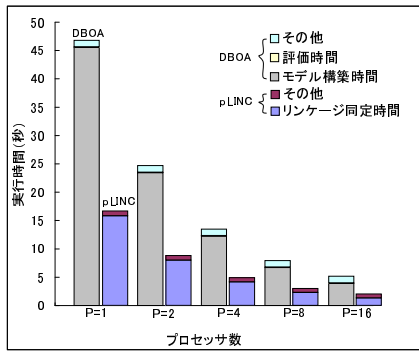


図 4 5 ビットトラップ関数におけるリンケージ同定時間とその他の実行時間の比較 (左) とネットワーク構築とその他の実行時間の比較 (右)

には関係せず、集団サイズ n と個体長 l に依存していることが分かっている。それゆえ、本稿で扱ったテスト関数では、評価時間が非常に短く、ネットワーク構築のコストのほうが高いために DBOA の実行に時間がかかる結果となっている。また、pLINC におけるリンケージ同定以外に要する計算時間はほとんど少ないことも、実行時間が DBOA よりも短い要因である。

この結果を受けて、以下では対象関数の中にウェイトをいれて、1 個体の評価時間を長くした関数を擬似的に作成する。対象関数としては 5 ビットトラップ関数を用いて個体長 105 ビットとし、1 個体の評価にウェイトを 0.001, 0.01, 0.1, 1 ミリ秒と入れた時の実行時間を測定する。また、通信に関する時間を含みせず単純に評価時間とネットワーク構築時間を比較するために、用いるプロセッサ数は 1 個とした (図 5)。

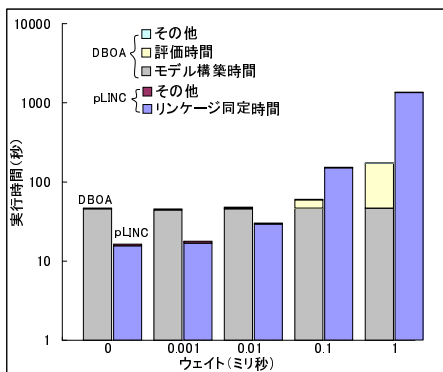


図 5 1 個体の評価にウェイトを入れたときの実行時間の比較

結果を見ると、評価時間が長くなるほどリンケージ同定のコストが大きくなっていることがわかる。一方で、BOA ではネットワーク構築に要するコストはほぼ一定であるが、毎世代の個体の評価にかかる時間が長くなっているため、コストの増加が大きいと言える。

以上のことから、1 個体の評価時間が比較的にかかる問題では、DBOA が有効であると言えるだろう。一

方で、評価時間と比較してネットワーク構築に時間の要する問題では pLINC を用いるべきであると言える。しかし、図 5 の 1 ミリ秒のウェイトを入れたときのように、1 個体の評価に比較的時間のかかる問題であっても、図 1 で示したように pLINC の台数効果は DBOA よりも大きいので、より多くのプロセッサを用いて並列化することで DBOA との実行時間の差を小さくすることができる。

5. ま と め

本稿では、近年提案されている pLINC と DBOA の 2 つの手法を取り上げ、その性能について台数効果と実行時間の観点から比較を行った。本稿の結果から、個体の評価時間とモデル構築の時間にしたがって適用すべき手法を選択でき、ユーザにおける並列 GA 手法の適用基準となることが期待できる。今後は、Erickらによって研究された並列 GA [3] の理論的実験との比較も参考に、ユーザがより効率よく解を求めることのできる手法とは何かについて調査することを検討していきたい。

参 考 文 献

- 1) Ocenasek Jiri : Parallel Estimation of Distribution Algorithms. PhD. Thesis, Faculty of Information Technology, Brno University of Technology, Brno, Czech Rep., 2002, pp. 1-154.
- 2) Martin Pelikan, David E. Goldberg and Erick Cantú-Paz : BOA: The Bayesian optimization algorithm. Technical Report IlliGAL Report No.99003, University of Illinois at Urbana-Champaign, Urbana, IL, 1999.
- 3) Erick Cantú-Paz : Designing Efficient and Accurate Parallel Genetic Algorithms. PhD thesis. University of Illinois at Urbana-Champaign, 1999.
- 4) Masaharu Munetomo and David E. Goldberg. Designing a genetic algorithm using the linkage identification by nonlinearity check. Technical Report IlliGAL Report No.98014, University of Illinois at Urbana-Champaign, Urbana, IL, 1998.
- 5) 棟朝雅晴, 村尾直哉, 赤間清 : 並列遺伝的アルゴリズムの適用に関する考察 - どの問題にどの手法を用いるべきか -, MPS シンポジウム論文集 (進化的計算シンポジウム 2 0 0 2), pp.53-60 (2003)
- 6) Masaharu Munetomo and David E. Goldberg. Linkage identification by non-monotonicity detection for overlapping functions. Technical Report IlliGAL Report No.99005, University of Illinois at Urbana-Champaign, Urbana, IL, 1999.