社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2004－ARC－157　(29)
2004－HPC－ 97　(29)
2004／3／3

Bayesian Optimization Algorithmの並列化に関する実験的検証

村尾直哉[†]　棟朝雅晴[††]　赤間　清[††]

BOA(Bayesian Optimization Algorithm) は，集団における優良個体群の分布推定に基づいて次世代の個体群を生成する手法であり，通常の遺伝的アルゴリズム (Genetic Algorithms, GA) では解くことが困難な問題を効率よく解くことのできる手法として提案されている．しかし，分布推定にかかる計算コストが大きいため，BOA の並列化のための研究が行われている．既存の研究では，この並列 BOA に対していくつかの実験を行っているが，並列プロセッサ台数などが少ないなど不十分であると考えられる．本稿では，この並列 BOA の性能に関する実験的検証をより詳細に行っていく．また，並列 BOA の計算コストを実験的に調査していく．

Empirical Investigations on the parallelized Bayesian Optimization Algorithm

Naoya Murao ,[†] Masaharu Munetomo [††] and Kiyoshi Akama[††]

The Bayesian optimization algorithm is an advanced search technique generating population of string by estimating distribution of promising solutions, which can solve difficult problems efficiently that simple Genetic algorithm cannot solve. However, since computational cost for estimating distribution is large, research for parallelizing of BOA is necessary. Although Parallel BOA is experimented in the previous research, it is performed on a small number of processors. In this paper, empirical investigation on the performance of parallel BOA is performed in detail.

1. はじめに

遺伝的アルゴリズム (Genetic Algorithms, GAs) は進化的計算の一種として提案され，回路設計などの設計問題や，スケジューリングなどの組み合わせ最適化問題など，広範囲の問題に対して適用されている．GA を含む進化的計算は生物の進化の過程を模倣して作られた工学的手法であり，GA では解の候補である個体をビット列などの文字列で表現し，複数の個体から形成される集団に対して適応度評価・選択・交叉・突然変異などの遺伝オペレータを適用することで次世代の集団を生成する．これを繰り返すことで，より良い解を生成していくことで，最終的に解を得る手法である．GA では部分解を構成すると期待される部分列の組み換えを行なうことで最適解を探索するために，部分解を構成する部分列が密に符号化されている必要がある．しかし，このような適切な符号化が保証されないような問題においては，単純な GA では解くことが困難となる場合がある (このような問題を GA-difficult problem という)．このような GA-difficult な問題を解くための手法として，個体集団の分布推定に基づく手法 (Estimation Distribution of Algorithms, EDAs) が提案されている．Pelikan らによって提案された BOA(Bayesian Optimization Algorithm) [2] は，この EDA の一種であり，近年，注目されている優れた手法の一つである．BOA は，ベイジアンネットワークを用いて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手法であり，GA-difficult な問題においても解を適切に求めることができる．また，個体の適応度評価回数に対する計算コストが $O(l^{1.55})$ とスケーラビリティの高い手法としても知られている (ここで，l は個体長である)．しかし，BOA における遺伝子座ネットワーク構築にかかるコストも無視できず，問題によっては実行時間に大きな影響を及ぼす場合がある．近年，BOA の並列化に関する研究が Ocenasek らによって行われており，この遺伝子座ネットワーク構築のコストを減少できることが示されている [1].

本稿では，Ocenasek らによって提案された並列 BOA に対して性能評価実験を行い，計算コストの実験的な測定を行う．また，適用する問題の特徴と並列 BOA の性能から並列 BOA が有効である問題クラスについても推定する．

2. BOA

GA-difficult な問題を解くための手法として，集団中の優れた部分個体集団の分布を推定し，それに基づいて次世代の集団を生成する EDA と呼ばれる手法が

† 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

提案されている．本稿ではこの EDA の中でも特に優れているといわれている BOA [2] を並列化した手法の実験的な性能検証を行っていくが，ここではまず，BOA を含む EDA 一般のアルゴリズムについて説明する．EDA の基本的なアルゴリズムは以下のようになる．

(1) 個体集団の初期化を行う
(2) 個体集団の適応度評価を行う
(3) 集団内における適応度の高い部分個体群を選択する
(4) 選択された部分個体群から，その分布を推定する
(5) 推定された分布から次世代の集団を生成する
(6) 終了条件が満たされれば終了，そうでなければ2へ

BOA はベイジアンネットワークに基づく確率モデルを構築し，分布推定を行うアルゴリズムである．ベイジアンネットワークは変数間の依存関係を表現し，ネットワークを構築する際に用いた集団と同様の性質を持つ新たな集団の生成に用いることができる．このベイジアンネットワークにおいてノードが個体中のビット (文字変数) の位置を表し，向きを持つ辺が文字の分布に関する条件付き確率 $p(X)$ を表す．このときの条件付き確率 $p(X)$ は，

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

と表される．ここで，$X = (X_0, X_1, \ldots, X_{n-1})$ は文字変数 X_i から構成される個体を表現し，Π_{X_i} は BOA のネットワーク中で X_i の方向への辺を接続しているノードの集合をあらわす (図 1)． また，BOA では条

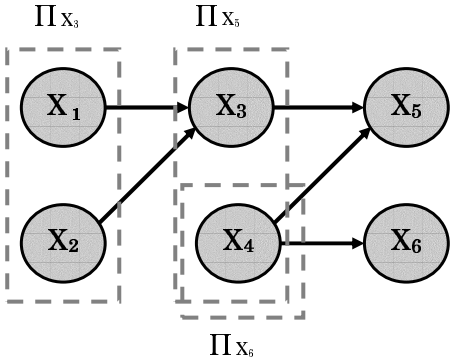

図 1 bayesian network の例

件付き確率を用いるため，非循環ネットワークを構築する．

この確率モデル構築に際して，集団の分布を正しく反映したネットワークを構築するために，ネットワークの品質の指標として，いわゆる Bayesian Dirichlet(BD) メトリックが用いられる．BOA では，このメトリックを最大化するようにネットワークを構築していく．また，計算コストの制限のためにノードに入る辺の数を k 本に限定して探索を行う．BOA における確率モデルの構築のアルゴリズムは以下のようになる．

(1) 辺のつながれていないノードのみからなる初期ネットワーク B を生成する
(2) ランダムに選んだノードのペアをつなぐ辺をネットワーク B に追加したものを B' とする
(3) ネットワーク B' が循環ネットワークになっている場合は，追加した辺を取り除き 2 へ戻る
(4) ネットワーク B と B' のメトリックを比較して，メトリックが増加する場合には $B \leftarrow B'$ とする
(5) 終了条件が満たされれば終了，そうでなければ 2 へ戻る

上記を行うことで，現在の集団を反映したネットワークを構築し，また，その確率分布から次世代の個体を生成していくことで探索が行われる．

BOA ではネットワークの構築中に個体の適応度評価を行わないため，個体長 l の増加によって生じる計算コストの増加が，線形もしくはそれに近い結果が示されており，$O(l^{1.55})$ であることがわかっている．しかし，上記のコストは適応度評価回数にもとづく計算コストであり，確率モデルの構築に関するコストは含まれていない．BOA における確率モデル構築のコストは $O(kl^3) + O(kl^2 2^k N)$ であることもわかっており (ここで，k はネットワーク内のノードに入る辺の数，l は個体長，N は集団サイズ)，実際にはこの計算コストを無視できないことのほうが多いと考えられる．そのため，この確率モデル構築の並列化に関する研究が Ocenasek らによって行われており，次節でその並列化方法について説明することにする．

3. 並列 BOA

これまでに述べてきたように，BOA におけるネットワーク構築の計算コストが実行時間に大きく影響を及ぼすような場合が存在する．しかし，ネットワーク構築の方法からわかるように，各ノードから他のノードへ辺を追加可能かどうかを調査するので，ノード間での依存関係が存在し，単純に並列化してしまうと不整合が生じてしまうことがある．そこで，Ocenasek ら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，ネットワーク構築を並列化している [1].

まず，各世代の初めにおいて $0, 1, \ldots, l-1$ のランダムな順列が生成され，配列 $O = (O_0, O_1, \ldots, O_{l-1})$ に保存される (l は個体長)．各ノードに対応付けられた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辺の方向はこの順序に従う．すなわち，X_j から X_i への辺の追加は，$O_j < O_i$ の場合に行われる．3 ノード $X = (X_0, X_1, X_2)$ に対し，順列 $O = (1, 0, 2)$

を持つ図 2 を例に考えると，X_1 は X_0 と X_2 に辺を追加可能かどうかを調べる必要があるが，X_0 は X_2 にのみ辺が追加可能かを調べるだけでよく，X_1 に対して調べる必要がなくなる (図 2 において，太い矢印が追加可能な方向を示し，細い矢印は追加不可能な方向を示す)．この順列を用いることで，構築されるネットワークが循環であるかどうかを調べる必要がなく，また，ノード間の依存関係を減らすことができる．しかし，このようにすることで探索空間が減少してしまう問題もあるが，各世代で新たな順列を生成しなおすことにより，これを補正する仕組みをとる．

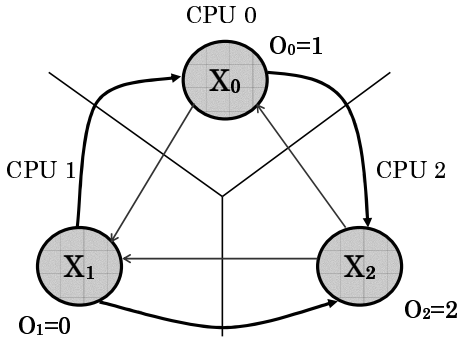

図 2 順列を用いたネットワーク構築の例

上記の並列ネットワーク構築は以下のようなアルゴリズムになる．

(1) 従来の BOA と同様に，辺のつながれていないノードのみからなる初期ネットワーク B を生成する
(2) ランダムな順列 $O = (O_0, O_1, \ldots, O_{l-1})$ を生成する
(3) 各プロセッサに各ノードを割り当てる
(4) 順列 O を各プロセッサに送信する
(5) 以下の操作を，それぞれの割り当てられた順列の要素に従って並列に処理する
  (a) 順列に従って選ばれたノードのペアをつなぐ辺をネットワーク B に追加したものを B' とする
  (b) ネットワーク B' が循環ネットワークになっている場合は，追加した辺を取り除き 5a へ戻る
  (c) ネットワーク B と B' のメトリックを比較して，メトリックが増加する場合には $B \leftarrow B'$ とする
  (d) 終了条件が満たされれば終了，そうでなければ 5a へ戻る

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辺が追加可能どうかを調べる処理は，他のノードにあまり依存せずに，並列にネットワークを構築することが可能となる．また，論文内ではネットワーク構築にかかるコストも示されており，$O(kl^2 2^k N/p)$ であることがわかっている (ここで，p は並列化に用いるプロセッサ台数，その他のパラメータは上述したものと同じ)．本稿では，この結果に関する実験的検証を次節で行っていく．

また，Ocenasek らにより 2 つの次世代個体群の生成方法が提案されており，それぞれ PBOA(Parallel BOA)，DBOA(Distibuted BOA) と呼ばれている．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させる．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 $(i-1)$ 個の遺伝子が決定された個体を受け取り，変数 X_i の値を決定する．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 (図 3)．

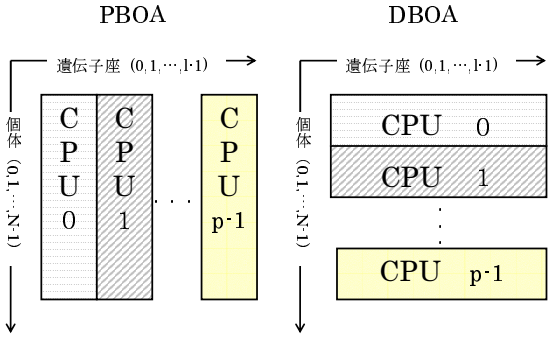

図 3 PBOA と DBOA における個体生成の違い

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，個体生成の容易さと実験環境から DBOA を用いて，その性能を検証していく．

4. 実　　験

この節では，これまでに説明した DBOA に関する性能調査を行う．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 関数を用いることにする．

5 ビットトラップ関数は，n 個の 5 ビット部分列 s_i を持つ個体 $s = \{s_1, s_2, \ldots, s_n\}$ に対して，以下の式

で表現される．

$$F(s) = \Sigma_{i=1}^{n} F_{trap}(u_i) \tag{2}$$

$$F_{trap}(u_i) = \begin{cases} 5 & \text{if } u_i = 5 \\ 4 - u_i & \text{else} \end{cases} \tag{3}$$

ここで，u_i は 5 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，これらのテスト関数を用いて DBOA の性能を，その台数効果と実行時間 (単位は秒) 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) 上で行われた．また，すべての実験において，ネットワーク構築において各ノードに入る辺の数 $k = 2$ として実験を行うことにする．

まず，DBOA における台数効果を測定する．対象関数として 5 ビットトラップ関数を用いて個体長を 105 ビットとし，20 回の平均結果を記録した．図 4 は解が得られるまでの実行時間，ネットワーク構築時間のそれぞれの台数効果について示している．

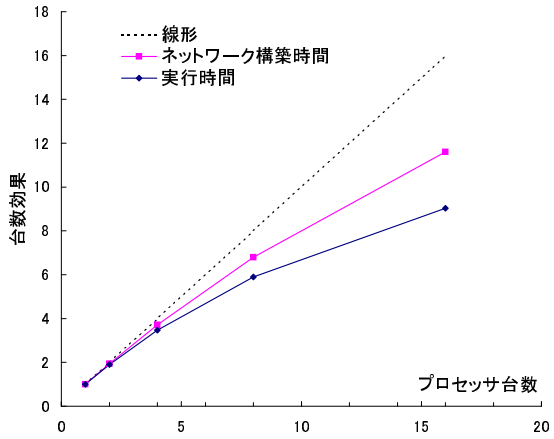

図 4 DBOA における台数効果

実験結果が示すように，ネットワーク構築の並列化などにより実行時間の減少が可能となっていることがわかる．また，ネットワーク構築の台数効果は実行時間の台数効果よりも大きくなっているが，この原因として考えられる理由は，用いている集団サイズ (ここでは集団サイズ $N = 1400$) と比較してプロセッサ台数が小さいので，個体生成に時間を要しているためであると考えられる．

次に，BOA と DBOA(用いるプロセッサは 8 台とする) の性能を比較する実験を行う．対象関数として OneMax 関数と 5 ビットトラップ関数を用いる．また個体長は 105 ビットとし，最適解が得られまでの実行時間，ネットワーク構築時間，評価時間，平均世代数について 20 回の平均結果を測定する (表 1)．

この結果が示すように，DBOA は BOA と同程度の世代数で最適解を求めることが可能であり，また，そのネットワーク構築時間を大きく減少させることが可能となっている．DBOA において，順列を用いてネットワーク構築を並列化することで探索空間が減少することが懸念されたが，従来の BOA よりも高速に最適解を見つけることができている．

では，この並列 BOA が有効な問題クラスとはどのようなものだろうか？　本稿では，適応度評価時間とネットワーク構築時間の 2 つの観点から検証する．対象関数としては 5 ビットトラップ関数を用い，試行回数 20 回の平均値を記録する．また，用いたプロセッサ台数は 8 台とした．

まず，適応度評価時間を擬似的に長くした関数を設定する．個体長は 105 ビットとする．先ほどの 5 ビットトラップ関数における 1 個体の適応度評価時間 T_f に対し，

$$T_f' = T_f + w \tag{4}$$

とウェイト w を追加した関数を擬似的に作成する．このときの実行時間，モデル構築時間，ネットワーク構築時間，適応度評価時間を測定する．結果は表 2 のとおりである．比較のために，棟朝らによって提案されたリンケージ同定に基づく並列 GA [5] についての同様の実験結果も示す．2 つの手法を比較してもわかるように，DBOA は 1 個体の評価に対しウェイトを挿入しても大きく実行時間には影響していないことがわかる．これは，前述したようにネットワーク構築に関して適応度評価は行われないことが要因である．このように適応度評価時間が長くなるような問題に対して，DBOA は実行時間が大きく変化しないため有効であると考えることができる．一方でリンケージ同定に基づく並列 GA は適応度評価時間に大きく影響していることがわかる．

次に，個体長を長くした場合の実行時間，ネットワーク構築時間，適応度評価時間を測定する．結果は表 3 のとおりである．このように個体長が長くなるにつれ，ネットワーク構築のコストが増加していることがわかる．また，評価時間も増加しているが，ネットワーク構築のコストの増加に比べると小さいことがわかる．このように，ネットワーク構築時間が全実行時間の大部分を占めるような問題では，問題サイズが大きくなるにつれ，その実行時間が実用的でなくなるという問題がある．また，先の実験と同様にリンケージ同定に基づく並列 GA と比較して考えると，ともに実行時間の増加はあるものの，伸び率は DBOA の方が高いことがわかる．これは，リンケージ同定に基づく並列 GA はおおまかに近似して $O(l^2/p)$ の計算コストがかかるのに対し，DBOA のネットワーク構築だけで $O(kl^2 2^k N/p)$ のコストがかかることが原因であると考えられる．このように，ネットワーク構築の時間が

表 1 BOA と DBOA の性能比較

onemax関数	実行時間	ネットワーク構築時間	評価時間	平均世代数
BOA	3.152	3.110	0.001	21.3
DBOA	0.550	0.481	0.001	22.6
5ビットトラップ関数	実行時間	ネットワーク構築時間	評価時間	平均世代数
BOA	46.811	45.563	0.0735	36.6
DBOA	7.932	6.701	0.0735	38.2

表 2 適応度評価時間を長くした場合の DBOA の実行時間の変化 (一番下はリンケージ同定に基づく並列 GA の実行結果)

ウェイト(ミリ秒)	0	0.001	0.01	0.1	1
実行時間	8.084	8.111	9.472	20.995	134.516
ネットワーク構築時間	6.841	6.620	6.835	6.983	6.974
評価時間	0.0735	0.239	1.335	12.67	126.09
リンケージ同定に基づく並列GA	2.162	2.346	4.008	20.619	186.712

表 3 個体長を長くした場合の実行時間の変化 (一番下はリンケージ同定に基づく並列 GA の実行結果)

ビット長	50	100	200	300	400	500
実行時間	0.745	7.230	82.337	340.9064	1072.985	2424.975
評価時間	0.320	1.158	4.290	8.640	15.230	23.240
ネットワーク構築時間	0.425	5.540	75.857	332.254	1057.743	2328.325
リンケージ同定に基づく並列GA	0.301	2.028	18.866	61.663	144.433	281.020

大きいような問題では，DBOA の実行時間が大きくなる可能性があることがわかる．

最後に，表 3 で示した結果から並列 BOA における計算コストを実験的に算出する．適用する問題が大きく，すなわち，個体長が長くなるにつれ，その計算コストがどのように変化するかを測定し，それに対する近似式を用いて計算する．算出結果は図 5 である．

既存の結果によると，BOA の適応度評価に関する計算コストは $O(l^{1.55})$ 程度であったが，本稿の結果はそれよりも多い計算コスト $O(l^{1.856})$ を要していることがわかった．しかし，本稿で比較のために用いた，リンケージ同定に基づく並列 GA は計算コストが $O(l^2/p)$ であり，DBOA の方が小さな計算コストで最適解を探索できることがわかる．また，ネットワーク構築のコストが全実行時間の大部分を占める問題も存在し，ネットワーク構築のコストも考慮したスケーラビリティを求める必要があることがわかる．図 5 の結果からわかるように，本稿の測定ではネットワーク構築のコストは $O(l^{3.747})$ であることも測定された．このことから，DBOA は問題の大きさに対して，適応度評価の観点で言うとスケーラビリティのある手法であると言えるが，ネットワーク構築の観点ではそうとは限らないことがわかる．本稿で用いた DBOA は Ocenasek らの論文内で示された構築方法に基づいており，ネットワーク構築や次世代集団の生成を並列化した手法であるため，適応度評価に関しては並列化されていない．問題サイズが大きくなるにつれ，必要となる集団サイズが大きくなることから，従来の並列 GA の研究 [3] におけるマスタースレーブ方式のような手法で適応度評価を並列化するなど，さらなる計算コストの削減をすることは可能であるが，これは適応度評価に要する時間が長い場合 (表 2 のような場合) に有効であるが，表 3 のようにネットワーク構築に時間がかかるような問題では効果的ではないだろう．

以上のように，適応度評価に要する時間が長い問題の場合に適した手法であると言える．一方で，ネットワーク構築に時間がかかる問題では BOA の実行時間は他の手法よりも劣る結果となることが明らかとなった．

5. ま と め

本稿では，Ocenasek らによって提案された並列 BOA に関して実験的検証を行ってきた．本稿の結果から，並列 BOA は適応度評価時間が大きくなっても実行時間に大きな影響を及ぼさない手法であると言える．一方で，個体長が長くなるなど，ネットワーク構築のコストが大きくなるにつれ，実行時間を大きく増加させる結果を示していた．このように，各手法には適した問題クラスというのが存在し，解くべき問題に対して適切な手法を選択する必要がある．一般に，問題に関する前提知識をもっていることは少ないが，小さなコストで調べることは可能である．このようにして，解くべき問題に関する情報を調べ，その問題に適した手法を選択することにより，誤った手法選択によ

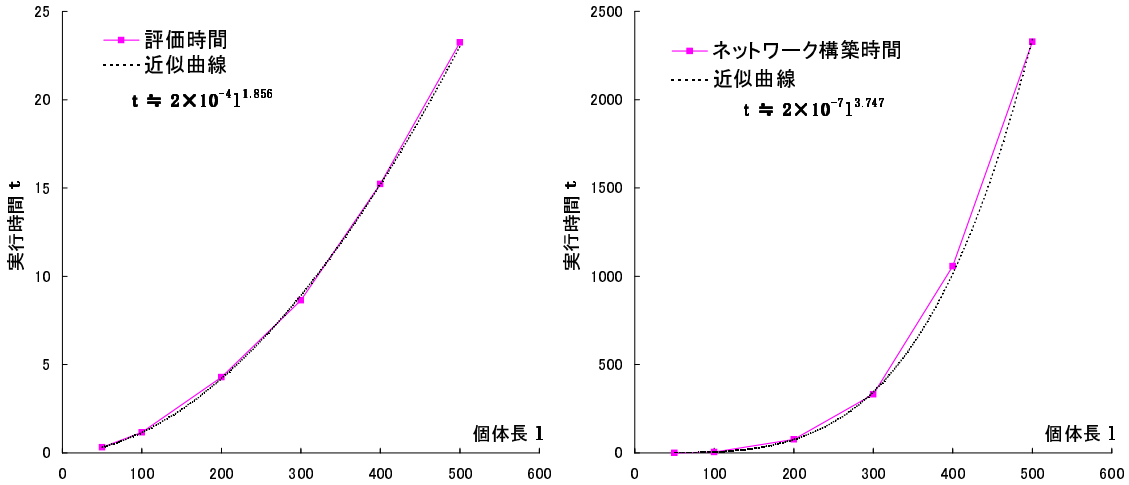

図 5 DBOA の計算コストの測定 (左は評価時間に関する計算コスト，右はネットワーク構築に関する計算コスト)

る計算コストの増加を防ぐことが可能となるだろう．そして，本稿の結果が，この手法選択のための一部となることが期待される．今後は，既存の研究結果や他の手法との比較を行い，問題クラスと適用手法との関係について調査していくつもりである．

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) 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　ーどの問題にどの手法を用いるべきかー, MPS シンポジウム論文集（進化的計算シンポジウム 2002）, pp.53-60 (2003)
5) Masaharu Munetomo, Naoya Murao and Kiyoshi Akama: A Parallel Genetic Algorithm Based on Linkage Identification, Proceedings of the 2003 Genetic and Evolutionary Computation Conference (GECCO-2003) Part 1, LNCS2723, pp.1222-1233 (2003)