2008－MPS－68 (28)
2008／3／5

散布探索法を導入したBOAについての検討

佐 竹 佑 太† 棟 朝 雅 晴†† 赤 間 清††

Bayesian Optimization Algorithm (BOA)[4] 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，BOA は広範囲の最適化問題を解くことができる．BOA の探索能力をさらに高めるために BOA に局所探索法を組み込んだ手法が提案されている[6]．しかしながら，新たな探索点を効果的に生成可能な散布探索法[2] は局所探索法として用いられてこなかった．そこで，本論文では散布探索法を BOA 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．

An Empirical Study on BOA which Introduces Scatter Search

YUTA SATAKE,† MASAHARU MUNETOMO†† and KIYOSHI AKAMA††

Bayesian Optimization Algorithm (BOA)[4] builds its probabilistic model which represents distribution of promising solutions and generates new solutions based on the model. Because BOA detects interdependent loci by using the model, it can solve a wide spectrum of optimization problems effectively. BOA which introduces local search in order to enhance its performance is already proposed[6]. However, they did not use Scatter Search[2], which can create new search points effectively, as local search. In this paper, we propose BOA which introduces scatter search and discuss its effectiveness.

1. はじめに

遺伝的アルゴリズム (Genetic Algorithms, GAs) の中心的なオペレータである交叉が有効にはたらくためには，ビルディングブロックを破壊しないようにする必要がある．しかしながら，設計変数間に依存関係がある問題や，ビルディングブロックが密になるように符号化されていない問題などに対しては，単純な交叉を用いると容易にビルディングブロックを破壊してしまう．

このような従来の GA の問題を解決する手法として，確率モデル構築型遺伝的アルゴリズム (Probabilistic Model-Building Genetic Algorithms, PMBGAs)[5] がある．PMBGAs 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．これにより，問題構造を同定しながら探索することが可能になるため，広範囲の最適化問題を解くことが可能である．

PMBGAs は一般に大域的な探索を得意とするが，局所的な探索を苦手とする．そこで，PMBGAs に局所探索法 (Local Search, LS) を組み込んだハイブリッド手法が提案されている．LS を組み込むことで PMBGAs 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．ここでは，LS として集団ベースの LS である散布探索法 (Scatter Search, SS)[2] を用いる．SS は複数の個体がそれぞれ LS をおこない，一定期間が経つと新たな解を生成しそこから再び探索を始める．解の生成は集団の情報が使われるため，探索の集中化と多様化をおこなうことが可能である．そのため，SS はそれ単体でも高い性能を示す．にもかかわらず，PMBGAs に LS を組み込んだ手法において，LS として SS は用いられてこなかった．そこで，本論文では PMBGAs に SS 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，本論文であつかう PMBGAs である Bayesian Optimization Algorithm (BOA)[4] について簡単に説明したあとに，本論文であつかう LS である SS について簡単に説明する．次いで，BOA に SS

† 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 Technology, 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.

(1) $t \leftarrow 0$ とし，初期個体集団 $P(0)$ をランダムに生成する．
(2) $P(t)$ から親個体集団 $S(t)$ を選択する．
(3) $S(t)$ を基に Bayesian ネットワーク B を構築する．
(4) B によって示される結合確率分布に基づいて新たな個体集団 $O(t)$ を生成する．
(5) $P(t)$ の一部を $O(t)$ で置き換えることで次世代の個体集団 $P(t+1)$ を生成し，$t \leftarrow t+1$ とする．
(6) 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．

図 1 Bayesian Optimization Algorithm
Fig. 1 Bayesian Optimization Algorithm.

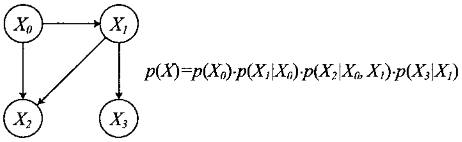

図 2 Bayesian ネットワークとこのネットワークで表される結合分布の簡単な例
Fig. 2 A simple example of a Bayesian network and the joint distribution encoded by this network.

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. 散布探索法を導入した BOA

2.1 Bayesian Optimization Algorithm

Bayesian Optimization Algorithm (BOA) は，Bayesian ネットワークによる確率モデルを構築することで探索をおこなうアルゴリズムである．BOA の手順を**図 1** に示す．

Bayesian ネットワークでは，条件付き確率を用いて複数の遺伝子座間の依存関係を表現する．新たな個体は，以下に示す $p(X)$ に基づいて生成される．

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

ここで，l は個体長，$X=(X_0,X_1,\ldots,X_{l-1})$ は文字変数 X_i から構成される個体，Π_{X_i} はネットワーク中で親となっているノードの集合を表す．Bayesian ネットワークと結合分布の例を**図 2** に示す．

構築されるネットワークは集団の分布を正しく反映したネットワークである必要がある．ネットワーク

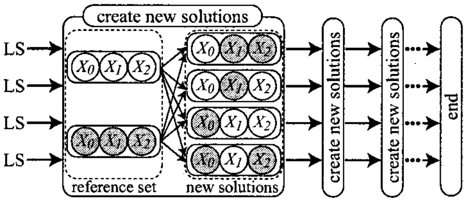

図 3 散布探索法
Fig. 3 Scatter Search.

の評価尺度として，Bayesian-Dirichlet (BD) metric $p(D,B|\xi)$ が用いられる．

$$\begin{aligned} &p(D,B|\xi) \\ &= p(B|\xi) \prod_{i=0}^{l-1} \prod_{\pi_{X_i}} \frac{m'(\pi_{X_i})\pi_{X_i}!}{(m'(\pi_{X_i})+m(\pi_{X_i}))!} \\ &\times \prod_{x_i} \frac{(m'(x_i,\pi_{X_i})+m(x_i,\pi_{X_i}))!}{m'(x_i,\pi_{X_i})!} \end{aligned} \tag{2}$$

ここで，B はネットワーク，D はデータ，ξ は背景情報，$p(B|\xi)$ はネットワーク B の事前確率，$m(\pi_{X_i})$ は D において変数 Π_{X_i} が π_{X_i} であるようなデータの数，$m(x_i,\pi_{X_i})$ は D において変数 X_i が x_i，Π_{X_i} が π_{X_i} であるようなデータの数，$m'(x_i,\pi_{X_i})$ は事前情報としての $m(x_i,\pi_{X_i})$ の値を表している．この BD 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．最初はノードのみから構成されるネットワークからスタートし，ランダムに辺を追加していくことで探索を進める．探索には貪欲法が用いられる．

2.2 散布探索法

散布探索法 (Scatter Search, SS) は集団ベースの局所探索法 (Local Search, LS) である．SS のポイントは，新たな探索点を生成しそこから再び探索を始めるところにある．SS の手順を**図 3** に示す．

LS はどのようなアルゴリズムを使ってもよい．本論文では LS として，局所解から脱出可能なタブー探索法 (Tabu Search, TS)[1] を用いる．reference set は適応度基準，多様性基準などで選ばれる．

2.3 散布探索法を導入した BOA

GA と LS 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[3]．このハイブリッドアルゴリズムは GA の大域的な探索と LS 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．

PMBGAs に LS を組み合わせた手法についてもすでに研究がおこなわれており，EDA/L などが提案されている[6]．PMBGAs に LS 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 LS 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，LS として集団ベースの SS を，PMBGAs として BOA を用いる．SS は TS により局所解からの脱出可能である．また，集団の情報を用いた個体生成により探索の集中と多様化が可能である．これらにより，LS の探索能力の向上をねらう．SS は新たに生成された個体の一部（10 個体～集団サイズの 10% 程度）に適用する．すなわち，BOA の (5) のステップ後に SS を適用する．以後，SS を導入した BOA を BOA+SS と呼ぶものとする．

3. 数値実験

3.1 実験条件

SS, BOA, BOA+SS の比較をおこなう．対象問題は**表.1** に示す 10 のテスト関数である．個体はバイナリ表現であるため，実数値にデコードしてから評価値を計算している．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示す．なお，パラメータは最適なものを実験的に選んだ．SS において，タブー期間は $0.8l$ (l は個体長) とし，TS の適用回数は 1 個体あたり $1.5l$ 回とした．実験マシンは SGI Onyx 300 (MIPS R14000 600MHz × 32 CPU, 16GB 共有メモリ) を使用した．

3.2 実験結果

実験結果を**表.2** に示す．表には，各アルゴリズムが最適解を得るまでの適応度評価回数，実行時間（秒）を示している．Griewank 関数，Rosenbrock 関数については SS で最適解が得られなかった．

まず，適応度評価回数をみてみる．Ackley 関数以外の 9 つの関数で BOA がもっとも小さくなった．Ackley 関数については SS がもっとも小さくなった．SS，BOA+SS の適応度評価回数が多くなっているのは，SS での TS が多くの適応度評価を必要とするためである．

次に，実行時間についてみていく．SS で解ける関数については SS が最速となっており，SS で解けない関数については BOA+SS が高速に解を得ていることがわかる．SS で解ける関数において，SS よりも BOA+SS のほうが時間がかかっているのは，BOA+SS は SS にかかるコストに加えて BOA にかかるコストを要するためである．BOA と BOA+SS を比べると，すべての関数において BOA よりも BOA+SS のほうが高速に解を得ている．

3.3 考　　察

ここでは，(1) SS が有効でない問題においても BOA より BOA+SS のほうが高速に解を得ている理由，(2) SS が有効な問題，BOA+SS が有効な問題について考える．

表 3　集団サイズ
Table 3　Population size.

	BOA	BOA+SS
F_{Ackley}	7000	100
F_{Alpine}	9000	100
$F_{Rastrigin}$	5000	100
$F_{Schwefel}$	6000	100
F_{Sphere}	5000	100
F_{Step}	1000	100
$F_{Bohachevsky}$	6000	100
$F_{Griewank}$	6000	1500
F_{Levy}	5000	100
$F_{Rosenbrock}$	100000	50000

まず，SS が有効でない問題においても BOA より BOA+SS のほうが高速に解を得ている理由について考える．これは集団サイズに関係している．各関数に対する BOA と BOA+SS の集団サイズを**表.3** に示す．SS が有効でない問題，つまり Griewank 関数と Rosenbrock 関数についてみてみると，BOA に比べ BOA+SS は集団サイズが小さいことがわかる．この理由は，SS による解の質の向上により，少ないサンプル数で良好な確率モデルを得られるためである．BOA の処理では Bayesian ネットワークの構築に要するコストが大きい．BOA におけるネットワーク構造の探索コストは $O(k2^kl^2N + kl^3)$ (N は集団サイズ，l は個体長，k はそれぞれの流入する辺の最大値) と見積もられていることから，集団サイズは小さいほうがよい．

次に，SS が有効な問題，BOA+SS が有効な問題について考察する．SS はほとんどの問題に対して有効であることがわかったが，設計変数間に依存関係のある問題の一部について最適解が得られない．設計変数間に依存関係のない問題に対しては SS は有効といえるだろう．いっぽう，BOA+SS はすべての問題で最適解が得られている．さらに，BOA よりも高速に解を得ることが可能である．このことから，もっとも BOA+SS はもっともロバストな性能を示したといっていいだろう．ただし，BOA+SS は適応度評価回数が多くなるので，評価に時間がかかる問題に対しては BOA に劣る結果を示すことも考えられる．PMBGAs と LS のハイブリッドが有効な条件については，各処理にかかるコストから見積もることが可能である[7]．

4. おわりに

BOA に SS を組み込んだ手法 BOA+SS を提

表 1　テスト関数
Table 1　Test functions.

関数	探索区域	変数間の依存関係
$F_{Ackley}(x) = -20\exp\left(-0.2\sqrt{1/n\sum_{i=1}^{n} x_i^2}\right) - \exp\left(1/n\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^n$	なし
$F_{Alpine}(x) = \sum_{i=1}^{n} \lvert x_i \sin x_i + 0.1x_i \rvert$	$[-10, 10]^n$	なし
$F_{Rastrigin}(x) = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i))$	$[-5, 5]^n$	なし
$F_{Schwefel}(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$[-512, 512]^n$	なし
$F_{Sphere}(x) = \sum_{i=1}^{n} x_i^2$	$[-5, 5]^n$	なし
$F_{Step}(x) = 6n + \sum_{i=1}^{n} \lfloor x_i \rfloor$	$[-5, 5]^n$	なし
$F_{Bohachevsky}(x) = \sum_{i=1}^{n}(x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7)$	$[-5, 5]^n$	あり
$F_{Griewank}(x) = 1 + \sum_{i=1}^{n} x_i^2/4000 - \prod_{i=1}^{n}\cos(x_i/\sqrt{i})$	$[-512, 512]^n$	あり
$F_{Levy}(x) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2(1 + 10\sin^2(\pi y_{i+1}))$ $+(y_n - 1)^2(1 + 10\sin^2(\pi y_1)),\ y_i = 1 + (x_i - 1)/4$	$[-10, 10]^n$	あり
$F_{Rosenbrock}(x) = \sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$	$[-2, 2]^n$	あり

表 2　実験結果 (l は 個体長，$\#ofeval$ は適応度評価回数，$time$ は実行時間 (秒))
Table 2　Experimental result (l: string length, $\#ofeval$: # of evaluation, $time$: excution time(sec.)).

対象問題			SS		BOA		BOA+SS	
テスト関数	n	l	$\#ofeval$	$time$	$\#ofeval$	$time$	$\#ofeval$	$time$
F_{Ackley}	5	80	**96010**	**0.9**	115850	118.3	1650250	8.1
F_{Alpine}	10	110	1452080	**16.8**	**161100**	353.7	3993350	19.8
$F_{Rastrigin}$	10	100	1710114	**20.6**	**84750**	169.7	3300350	25.1
$F_{Schwefel}$	10	100	810054	**10.2**	**100500**	153.1	1650250	14.4
F_{Sphere}	10	100	150010	**1.5**	**88250**	150.6	1650250	6.4
F_{Step}	10	100	150010	**1.8**	**9150**	20.3	1650250	8.8
$F_{Bohachevsky}$	10	100	150010	**1.9**	**95400**	162.4	1650250	9.3
$F_{Griewank}$	10	100	×	×	**116700**	200.4	22509000	**143.4**
F_{Levy}	10	100	2220148	**28.7**	**94500**	154.2	3300350	35.7
$F_{Rosenbrock}$	5	60	×	×	**1435000**	3264.6	127067500	**1112.2**

案した．10 のテスト関数で実験をおこなった結果，BOA+SS がもっともロバストな性能を示すことがわかった．SS が有効でない関数においても BOA+SS が有効な理由について考察した．さらに，SS が有効な問題，および BOA+SS が有効な問題について考察した．今後は，BOA+SS をリガンドドッキング問題に適用し，実問題における BOA+SS の有効性を探る予定である．

参 考 文 献

1) Fred Glover: Tabu Search: A Tutorial, *Interfaces*, Vol.20, No.4, pp.74–94 (1990).
2) Fred Glover, Manuel Laguna and Rafael Marti: Scatter Search, *Advances in Evolutionary Computation: Theory and Applications*, A. Ghosh and S. Tsutsui (Eds.), Springer-Verlag, New York, pp.519–537 (2003).
3) 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).
4) Martin Pelikan, David E. Goldberg and Erick Cantú-Paz: Linkage Problem, Distribution Estimation, and Bayesian Networks, IlliGAL Report No.98013, University of Illinois at Urbana-Champaign, Urbana, IL (1998).
5) 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).
6) 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).
7) 佐竹佑太, 棟朝雅晴, 赤間清: 局所探索を導入した確率モデル型遺伝的アルゴリズムの計算コストについての検討, 情報処理学会研究報告, Vol.2006, No.56, pp.25–28 (2006).