

複数の群れを有する並列 Cuckoo Search による連続値最適化

河面 知定[†] アランニャ・クラウド[‡] 狩野 均[‡]

筑波大学 システム情報工学研究科[†] 筑波大学 システム情報系[‡]

1. はじめに

Cuckoo Search(CS)は、連続値最適化のために考案されたメタヒューリスティクスである。従来、島モデルのように CS の群れを複数に分割し、並列化した手法が提案されている。しかし、これらの手法は各群れが同一のパラメータを持つため、並列化を十分に活かしてきれていない。

本研究では、Slave(群れ)毎に異なる Lévy 安定分布形状決定パラメータを割り当てることで、大域探索と局所探索の役割分担を探索過程で適応的に調整する並列化手法を提案する。

本手法と従来手法を関数最適化問題に適用し、本手法の有効性を示す。

2. 基礎事項

2.1 Cuckoo Search(CS)

CS は、カッコウの托卵と、鳥類の飛行挙動である Lévy Flight をモデル化したメタヒューリスティクスである[1]。エリート保存をしながら解の改善を許すことで大域的収束性が証明されていることや[2]、事前に設定すべきパラメータ数が少ないことが特徴として挙げられる。

2.2 Lévy Flight(LF)

LF は、Lévy 安定分布に従うランダムウォークである。Lévy 安定分布は式(1)のように裾がベキ関数に比例する heavy tail な分布として知られる。ここで α は Lévy 安定分布の分布の形状を決定するパラメータである。

$$p(x) \propto x^{-(1+\alpha)}, 0 < \alpha \leq 2 \quad (1)$$

$\alpha > m$ のときにのみ Lévy 安定分布の m 次モーメントは存在するため、 $0 < \alpha \leq 1$ のとき平均と分散を持たず、 $1 < \alpha < 2$ のとき分散を持たない($\alpha = 2$ のとき標準正規分布と一致)。これにより、 α が小さいほど大きな乱数値を生成しやすくなる。

CS では、Mantegna's Algorithm(MA)を用いて $0.3 \leq \alpha \leq 1.99$ の範囲で LF を近似している[3]。

3. 提案手法

3.1 アルゴリズム

文献[4]を参考に、並列化 CS を提案する。疑似コードを図 1 に示す。各 Slave は世代毎に式(2)の α の設定則を適用しつつ、CS のアルゴリズムをそれぞれ独立に実行する。Master に $Sbest$ (Slave の最良個体)と α を送信し、2 回目の CS を実行する。Master 内の CS によって改善した $Sbest_i$ は送信元の $Slave_i$ に送り返される。

```

1  初期化
2  while ( 終了条件 )
3    for ( each Slavei )
4      αiを更新 → 式(2)
5      for ( 小集団サイズ )
6        CS のアルゴリズムを実行
7      Master が全ての Sbesti と αiを受信
8      for ( each Sbesti )
9        CS のアルゴリズムを実行
10     全ての Sbestiを Slaveiに送信

```

図 1 提案手法のアルゴリズム

本手法の特徴を以下に示す。

- α の違いによって各 Slave が大域探索・局所探索を分担
- Master 内で比較し合った $Sbest$ が各 Slave に送り返されることで Slave 間の個体の移住を実現

3.2 α の設定方法

α は式(2)を用いて世代ごとに更新する。ここで n は Slave の数、 $U[A,B]$ は $A \leq \alpha \leq B$ の一様乱数、 $\alpha_{min} = 0.3$ 、 $\alpha_{max} = 1.99$ は MA のパラメータの下限と上限を表す。 $r = \frac{\alpha_{max} - \alpha_{min}}{n-2}$ である。

Parallel Cuckoo Search Algorithm for Continuous Optimization
[†]Kazusada Kawamo, Graduate School of Systems and Information Engineering, University of Tsukuba
[‡]Claus Aranha, Hitoshi Kanoh, Faculty of Engineering, Information and Systems, University of Tsukuba

$$\alpha_i = \begin{cases} \alpha_{min} & (i = 1) \\ \alpha_{max} & (i = n) \\ U[\alpha_{min} + (i - 2) \times r, \alpha_{min} + (i - 1) \times r] & \text{Othewise} \end{cases} \quad (2)$$

4. 実験

4.1 実験条件

50次元の Sphere, Rastrigin, Ackley, Griewank, Step, Levy, Rosenbrock-Chain, Rosenbrock-Star, K-tablet の9種の関数に対して提案手法と既存の並列化CS(MLS-CS)[4], Basic CS (CS) [1]を用いた比較実験を行った. 表1に実験条件を示す. パラメータの値は予備実験によって最適化した. ただし, 2種のRosenbrock関数の世代数のみ100万回とした.

表1 実験条件

	CS	MLS-CS	proposed
世代数	10万	10万	10万
個体数×Slave数	40×1	10×4	10×4
α	0.3	0.3	式(2)
stepsize($\times 10^{-9}$)	1	1	1
P_a	0.25	N/A	0.2

4.2 実験結果と考察

比較実験の結果, 全ての関数に対し提案手法が最も良い目的関数値を得た. 表2に比較実験の結果を抜粋して示す. 提案手法が最も優れた目的関数値とCPU時間を獲得している. 図2と図3にSphere関数とRastrigin関数最適化の目的関数値の推移を示す. 提案手法は小さな α を持つSlaveが序盤に探索を行い, 大きな α を持つSlaveが中盤~終盤にかけて探索を行うことで, 効率の良い探索が可能になったと考えられる.

5. おわりに

異なるパラメータを有する並列化CSを提案した. 実験結果より, 本手法の有効性を確認した. 今後の予定として, パラメータの分割方法の検討, ならびに実問題に対する適用を実施する.

参考文献

[1] Xin-She Yang, Suash Deb, Cuckoo Search via Lévy Flights, Nature & Biologically Inspired Computing, no.37, pp. 210-214 (2009).
 [2] Wang F., He X.S., Wang Y., Yang S.M., Markov model and convergence analysis of cuckoo search algorithm, Computer Engineering, 38(11), pp.180-185, (2012).
 [3] Rosario Nunzio Mantegna, Fast, accurate algorithm

for numerical simulation of Levy stable stochastic processes, Physical Review E, Vol.49, no.5, pp.4677-4683(1994).

[4] Jaddi NS, Abdullah S, Abdul Malek M, Master-Leader-Slave Cuckoo Search with Parameter Control for ANN Optimization and Its Real-World Application to Water Quality Prediction, PLoSONE, 12(1):e0170372, https://doi.org/10.1371/journal.pone.0170372, (2017).

表2 実験結果の例

function	result	CS	MLS-CS	proposed
Sphere	best	1.03E-12	7.44E-13	6.53E-19
	mean	1.20E-12	9.18E-13	8.27E-19
	std	1.08E-13	1.04E-13	8.51E-20
	time(s)	118	80	71
Rastrigin	best	1.73E-10	1.42E-10	0.00
	mean	2.38E-10	2.11E-10	3.79E-14
	std	2.69E-11	3.35E-11	2.08E-14
	time(s)	114	94	79
Rosenbrock-Chain	best	4.08E-07	1.21E-06	5.01E-14
	mean	5.41E-07	2.77E-06	2.67E-12
	std	1.18E-07	6.02E-07	1.21E-11
	time(s)	1365	876	764

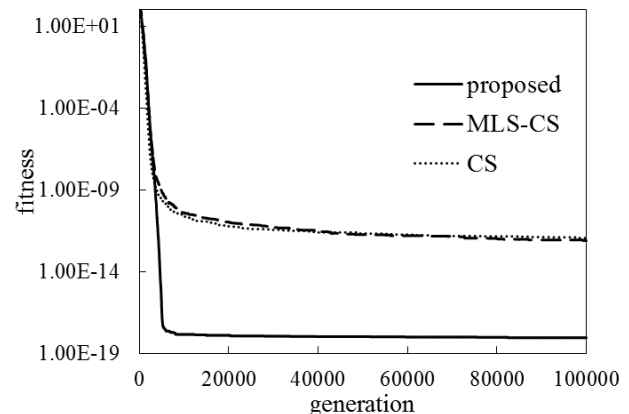


図2 Sphere関数の進化のグラフ

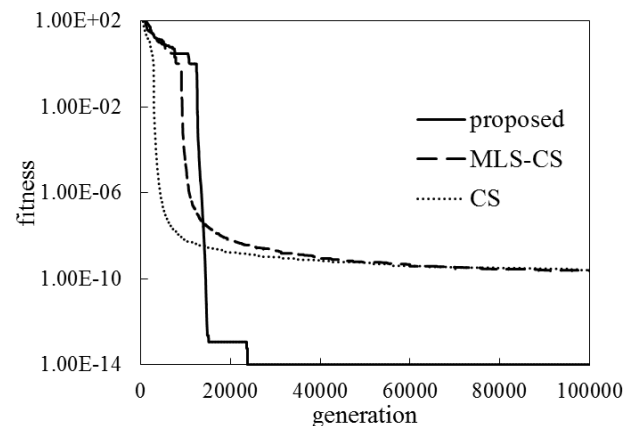


図3 Rastrigin関数の進化のグラフ