情報処理学会研究報告
IPSJ SIG Technical Report

Vol.2014-MPS-97 No.6
2014/3/3

島モデルに基づく適応型差分進化の性能評価

中島健一[†1]　　田川聖治[†2]

概要：島モデルに基づく差分進化（IbDE）では一定間隔の任意交叉により島の間で情報を交換する．本稿では，この任意交叉の間隔を変化させるための適応ルールを提案する．提案手法が IbDE の性能に与える影響をマルチコア CPU において評価した．その結果，提案手法により得られる解が従来の IbDE による解より優れていることを確認した．

1. はじめに

差分進化（DE : Differential Evolution）[1]は，決定変数が実数値をとる単目的の関数最適化問題を対象とした進化計算の一種である．近年，複数のプログラムを同時に実行できるマルチスレッド・プロセッサや，1 つのチップ上に複数のコア（プロセッサ）を集積したマルチコア CPU が増えている．著者らは，マルチコア CPU を利用した並行型差分進化（CDE ： Concurrent DE）を提案している[2]．マルチコア CPU を利用する DE の並行プログラムは，並列化の規模は小さいが，特別なハードウェアや専用の開発ツールを必要とせず，既存の DE の実行時間を着実に短縮できる．また，汎用性や移植性の面でも優れている．

著者らは島モデルと呼ばれるサブ集団（島）に基づく差分進化（IbDE : Island-based DE）も提案している[3]．IbDE では集団を複数の島に分け，島ごとに独立に DE を実行する．また，島の間で個体を移動させる従来の島モデルの移民と異なり，優れた個体の形質のみを交換する任意交叉と呼ぶ手法を採用した. IbDE についても並行プログラムとして実装し，マルチコア CPU を搭載した PC で実行させる．

本稿では, IbDE で行われている島の間の情報を交換するための任意交叉の間隔を変化させるための適応ルールを提案する. 提案手法により任意交叉の間隔を変化させる IbDE を AIbDE（Adaptive Island based DE）と呼ぶ．マルチコア CPU による数値実験と統計的検定により，従来の CDE 及び IbDE と比較して, AIbDE は解の精度で勝ることを示す．

2. Concurrent DE （CDE）

DE は主集団 P と副集団 Q を持ち，主集団 P から副集団 Q を生成した後，主集団 P を副集団 Q で上書きする．著者らは, DE の世代交代モデルを改良した DER を提案した[3]. DER では，主集団 P から副集団 Q を生成した後，副集団 Q から主集団 P を生成する．このため，集団の上書きに要する計算時間を節約できる．本稿では，マルチコア CPU を対象とした DER の並行プログラムを CDE と呼ぶ．CDE は，1 つのメイン・スレッドと N_T 個（N_T≧1）のワーカー・スレッドから構成される．主集団 P と副集団 Q は，それぞれ式（1）のように N_T 個のサブ集団 P_n と Q_n（n=1，…，N_T）に分割されて，全ワーカー・スレッドの共有メモリに保存される．

$$\begin{cases} P = P_1 \cup \cdots \cup P_n \cup \cdots \cup P_{N_T} \\ Q = Q_1 \cup \cdots \cup Q_n \cup \cdots \cup Q_{N_T} \end{cases} \quad (1)$$

CDE のメイン・スレッドは初期集団 P をランダムに生成した後，N_T 個のワーカー・スレッドを同時に起動し，それらの処理の終了を待つ．n 番目のワーカー・スレッドは他ワーカー･スレッドと同期して DER を実行する. すなわち，集団 P からサブ集団 Q_n を生成した後，他のワーカー・スレッドと待ち合わせ集団 Q からサブ集団 P_n を生成する．

3. Island-based DE （IbDE）

IbDE は，1 つのメイン・スレッドと N_T 個（N_T≧1）のワーカー･スレッドから構成される. 主集団 P と副集団 Q は，それぞれ式（1）のように N_T 個のサブ集団に分割されて共有メモリに保存される．ここで，サブ集団 P_n と Q_n の組を「島」と呼ぶ．メイン・スレッドでは N_T 個のワーカー・スレッドを同時に起動し，それらの処理の終了を待つ．各ワーカー・スレッドは担当する島（P_n と Q_n）において独立に DER を実行し，一定の世代間隔 N_C で任意交叉を行う．

任意交叉では全てのワーカー・スレッドが任意の個体を読みだすことができ，n 番目のワーカー・スレッドは Q から P_n を生成する．IbDE の実装では，任意交叉の前後でワーカー・スレッドの同期が必要であるが，各ワーカー・スレッドは大半の世代では非同期に実行される．このため，スレッドの同期制御に要するオーバーヘッドは非常に小さい．また，任意交叉は島の間で個体を移動させるのではなく，優れた個体の形質のみをトライアル・ベクトルを介して伝播させる．このため，ある個体のコピーが複数の島に重複して存在することがなく，集団内の個体の多様性が保持される．さらに，IbDE の島モデルのトポロジーは完全結合であり，形質の伝播方向に制約がない．

4. Adaptive Island-based DE （AIbDE）

AIbDE は任意交叉の制御パラメータ C_P と新たな個体が作られる割合である進化率 p の比較により適応的に任意交叉の間隔を変化させる．n 番目のワーカー・スレッドは指定された世代間隔 N_C に達すると定数 C_P に対し進化率 p が

†1 近畿大学総合理工学研究科
Graduate School of Science and Engineering Research , Kinki University
†2 近畿大学理工学部
School of Science and Engineering, Kinki University

$p \leq C_p$ となる場合は IbDE と同様に任意交叉を行い，そうでない場合任意交叉を行わない．n 番目のワーカー・スレッドが N_C 世代中にターゲット・ベクトルをトライアル・ベクトルで更新した回数を sum とし，各島の個体数を N_S として，進化率 p は式（2）のように計算される．

$$p = \frac{sum}{N_S \times N_C} \quad (2)$$

AIbDE では N_C 世代毎に任意交叉を行うか否かを上記のアルゴリズムによりワーカー・スレッドごとに決定する．

5. 数値実験

5.1 テスト問題

以下の 10 種類のテスト問題 f_p を使用した．

- Sphere Function : f_1
- Shifted Sphere Function : f_2
- Schwefel Function : f_3
- Shifted Schwefel Function : f_4
- Rosenbrock Function : f_5
- Rastrigin Function : f_6
- Shifted Rastrigin Function : f_7
- Shifted Rotated Rastrigin Function : f_8
- Ackley Function : f_9
- Griewank Function : f_{10}

ただし，テスト問題の次元はすべて D=30 とする．

5.2 実験方法

先に報告した CDE および IbDE と，本稿で提案した AIbDE の性能を比較する．まずこれら全ての DE の並行プログラムは，Java 言語で実装した．また，オペレーティング・システムには Microsoft 社の Windows®7 Home Premium を使用した．さらに，マルチコア CPU には同時に 2 つのスレッドを実行できるマルチスレッド・プロセッサを 4 個集積し，最大 8 個のスレッドを並列処理できる Intel®Core™ i7-2670QM プロセッサー（@2．2[GHz]）を用いた．すべての DE において，個体数 N_P=192，スケール係数 F=0.5，交叉率 C_R=0.9，世代数 G_M=2000 として，ワーカー・スレッド数は N_T=4，8 とした．このとき $N_P=N_S*N_T$ よりの関係より IbDE と AIbDE の島のサイズはそれぞれ N_S=48，24 となる．ここで，全 DE を各テスト問題 f_p に 50 回ずつ適用した．

5.3 実験結果

各 DE の並行プログラムを 10 種類のテスト問題に適用して得られた解の質を目的関数値で比較した結果を表 1 に示す．表 1 において○は 50 回平均で最小の目的関数値の値が得られ，それが Wilcoxon 検定による比較の結果，他の 1 つ以上のプログラムと有意な差が認められたことを示す．また，表中に○がないものは各プログラムに有意な差がなかったものである．表 1 から，6 種類のテスト問題で AIbDE は他の DE に勝り，3 種類のテスト問題で CDE と IbDE，AIbDE に有意な差はない．すなわち，f_5 を除いて AIbDE が他の DE に劣るということはなかった．

表 1 Wilcoxon 検定による比較

f_p		NT4	NT8
f_1	CDE		
	IbDE		
	AIbDE		
f_2	CDE		
	IbDE		
	AIbDE		
f_3	CDE		
	IbDE		
	AIbDE		○
f_4	CDE		
	IbDE		
	AIbDE		○
f_5	CDE	○	○
	IbDE		
	AIbDE		
f_6	CDE		
	IbDE		
	AIbDE		○
f_7	CDE		
	IbDE		
	AIbDE		○
f_8	CDE		
	IbDE		
	AIbDE		○
f_9	CDE		
	IbDE		
	AIbDE		○
f_{10}	CDE		
	IbDE		
	AIbDE		

6. 結論

本稿では，IbDE における任意交叉の間隔を適応的に変化させる AIbDE を提案した．従来の CDE および IbDE と比較して AIbDE は得られる解の質で勝ることを示した．

参考文献

[1] R．Storn and K．Price：“Differential evolution – a simple and efficient heuristic for global optimization over continuous space”，Journal of Global Optimization，Vol．11，No．4，pp．341-359（1997）

[2] K．Tagawa：“Concurrent differential evolution base on generational model for multi-core CPUs”，Proc．of 9th International Conference，Simulated Evolution and Learning，LNCS7673，Springer，pp．12-21（2012）

[3] 中島健一，田川聖治：島モデルに基づく差分進化の性能評価，情報処理学会関西支部 支部大会 講演論文集，B-102（2013）