

1. はじめに

遺伝的アルゴリズム(GA)は、個体数と、世代数の増加によって解の向上を図れるが、同時に計算コストも増加する。そこで本研究では、GAの高速化を目的として、分散遺伝的アルゴリズム(DGA)のマルチコア CPU による並列化を提案し、並列化後の経過時間について調査する。

2. マルチコア上での実装

DGA は、最適化手法の一つである遺伝的アルゴリズムの分散処理モデルである[1]。DGA は、解候補(個体)からなる母集団を島と呼ばれる複数のサブ母集団に分割し、各島に対して GA を適用する。図 1 に DGA の処理手順を示す。

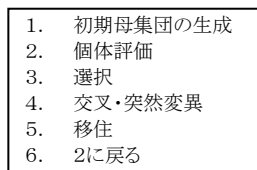


図 1 DGA の流れ

DGA では手順 1~4 に相当する遺伝的アルゴリズムの処理を島ごとに適用する。本研究では、マルチコア CPU 上の各コアに複数の島での処理を配分し、複数スレッドに DGA の処理を分散した時の並列化の効果について調べる。なお、実装にあたり OpenMP[2]を用いる。

3. 数値実験

DGA の評価関数としてベンチマーク関数の一つである Ridge 関数を用いる。このベンチマーク関数に対する最適化において、最終世代までの経過時間を記録する。実験では以下の 2 種の実行環境を用いる。また、DGA の各種パラメータとして表 1 の値を用いる。

表 1 DGA の各種パラメータ	
	DGA
個体数	512
島数	8
世代数	10000
スレッド数	1(非並列), 2

・実験機材(1) : Otter

Intel Core2Duo E8500, 2GB RAM,
Windows XP SP3, MS Visual C++ 2005 ,
OpenMP 2.0

・実験機材(2) : Octopus

AMD Opteron 2376, 8GB RAM,
openSUSE 11.1 , GNU Compiler 4.3.1 ,
OpenMP 2.5

DGA による最適化の経過時間を、(1)非並列(1スレッド)の場合、(2)Otter 上でスレッド化した場合、(3)Octopus 上でスレッド化した場合について各々測定し、(1)の経過時間を 1 とした時の相対的な経過時間を図 2 と図 3 に示す。図 2 は、図 1 の手順において「2.評価」のみを、図 3 は、図 1 の手順 1~手順 4 をスレッド化した場合の経過時間を比較している。

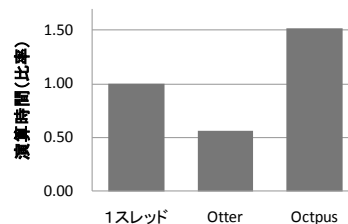


図 2 評価部分の並列化の効果

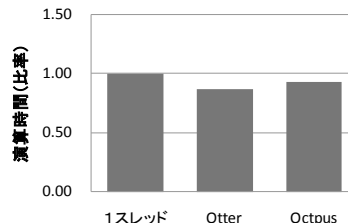


図 3 DGA 全体の並列化の効果

結果より、Otter では評価部分の並列化効果が大きいのにに対して DGA 全体の並列化効果は 20%弱の経過時間の軽減にとどまっている。一方 Octopus では、評価部分の並列化効果がないにも関わらず、DGA 全体の並列化効果は Otter と同等になっている。

4. おわりに

異なる 2 つのマルチコア環境で DGA を OpenMP により並列実装した。その結果、同一のプログラムにも関わらず、OpenMP による実装がマルチコア環境の相違に大きく依存することを確認した。

参考文献

- [1]相吉英太郎, 安田恵一朗 (編), メタヒューリクスと応用, 電気学会, 2007
- [2] 黒田久泰, “C 言語による OpenMP 入門”, プログラミング講習会資料, 東京大学情報基盤センター