

DNAS 上で動く GA

谷口 義樹

1 前回からの課題

- DNAS 上で動く GA の作成

2 進捗状況

2.1 DNAS 上で動く GA

Distributed Network Application System(以下 DNAS) は、分散したノードを階層的につなぎ、その上でアプリケーションを動作させるためのシステムである。またこれら分散したノード間の通信は、DNAS の提供する API を用いて行う。今回、最適化問題を解くための手法の一つである遺伝的アルゴリズム (以下 GA) を DNAS システム上に実装し、その動作を確認した。

2.1.1 各ノードで動く SGA

まず、知的システムデザイン研究室が開発した分散遺伝的アルゴリズムプログラム ga2k を鳥数 1 の設定にし、移住プロセスを省略することで単純 GA (Simple GA, 以下 SGA) とし、それを各ノード (4 ノード) で動かした。この段階では、それぞれのノード間で通信は行っていない。実行をした結果、正常に動作することが確認された。

2.1.2 ga2k との比較

各ノードで動作している GA (以下、DNAS/ga) とオリジナルの ga2k が同様の動作をしているかを確かめるため、同じパラメータ設定で、ノード間で個体情報交換を一切行わないように 1 ノードのみを使って動作させ、かつその中で鳥数を 1 に設定して、両者の比較を行った。5 つのテスト関数 (Rastrigin 関数, Schwefel 関数, Griewank 関数, Rosenbrock 関数, Ridge 関数) を解き、その関数評価値の推移を調べた。設定したパラメータを Table 1 に示す。

その結果、関数評価値の推移はほぼ同じであることが明らかとなり、オリジナルの ga2k を改良して DNAS システム上で動いている GA は、正常に動作していることが証明された。

2.1.3 各ノード間での個体情報の交換

次に各ノード間で、DNAS の提供する API を用いて個体情報の交換を行った。今回のモデルは、まず、DNAS にその世代で最も適合度の高い個体を送信する。受信の際には、DNAS からランダムに個体を受信し、自分のエリート (エリート保存が 2 個体以上の場合は、そのエリート個体内で最も適合度の低い個体) と置き換える。さらに DNAS では 3 秒に一度、通信が行われるため、

Table 1 DNAS/ga および ga2k のパラメータ設定

個体数	100
エリート数	5
設計変数	10
遺伝子長	100
初期個体の発生法	ランダム
鳥数	1
選択	トーナメント選択
交叉率	0.6
突然変異率	0.01
最大世代数	1000

最初の個体情報が到達する前に受信しようとする場合において、受信する個体がないため受信に失敗してしまう (以下でこれを受信エラーと呼ぶ)。この受信エラーを少なくするため、個体情報を送信してから受信するまでの間に、待ち状態を追加し実験した。待ち時間と受信エラー数の関係を Table 2 に示す。受信エラー数値として 4 ノード間での通信を行った中で最もエラー数が多かったノードの値をとった。

Table 2 送信から受信までの待ち時間と受信エラー数

待ち時間	受信エラー数
なし	304
0.03 秒	47
0.3 秒	5
3 秒	0

待ち時間を長くすると、それだけ受信エラーが少なくなり、十分に移住が行われることがわかった。また、各ノードでそれぞれ SGA を動かして得られた結果よりも、DNAS の提供する API を用いて各ノード間で個体情報の交換を行ったほうが良好な結果を示した。

3 翌月への課題

DNAS 上で動く GA が実装できたので、翌月はどのような個体を移住させるかについての検討を行う。また、DNAS 上の各ノードにそれまでに探索されたベストな解を保存しておくアーカイブを持たせて、解探索を進めた場合にどのような挙動を示すかについての検討を行う。