

3H-2

SIMD型遺伝アルゴリズムの提案
—巡回セールスマン問題への適用—浜田 喜生 高橋 義造
徳島大学工学部知能情報工学科

1. はじめに

最適化問題の解法として、遺伝アルゴリズム(GA)が注目を集めている。GAは、単一の局所解に陥ることが少ないことや、最適化するための評価関数は単に値や比較可能な情報を出力する計算手続きであればよいという利点をもつ。

しかし、逐次型計算機にGAを実装したとき、その処理能力の限界により、小規模の問題にしか適用できないという問題がでてくる。そこで、本稿では、GAにおける個々の個体の独立性と各遺伝操作における同期的な計算方法に注目して、価格性能比がよく設計が容易なSIMD型並列計算機に実装する方法を提案する。また、代表的な組み合わせ問題である巡回セールスマン問題(TSP)に適用して、ワークステーション(WS)上でシミュレーションを行い、逐次型計算機上での計算結果と比較したので、その結果を報告する。

2. TSPに対するGA

2.1 個体表現方法

TSPに対するGAの適用法については、いろいろな研究がなされている。今回の適用法では、マトリックス法を採用する[1]。この方法では、遺伝子を0と1として、それをマトリックス上に配置したものをパス表現として使用する。都市*i*から都市*j*に向うときマトリックスの(*i*, *j*)要素を1として巡回経路を表現する。この表現方法によると同じ巡回路を表現する個体が少なくなる。

2.2 適応度関数

個体数 *p* からなる個体群 $P(t)$ (*t*: 世代数) が存在し、個体 X_1 から X_p まで巡回路の総距離が $L_i(t)$ ($i: 1, \dots, p$) にあるとする。TSPでは、巡回路の短い方が優秀な個体であるので、その評価関数 $x_i(t)$ を以下のように計算する。

$$x_i(t) = \max_i L_i(t) - L_i(t)$$

この評価値を偏差値に変換して、0から100までの値にしたものが適応度関数 $z_i(t)$ である。[2]

$$z_i(t) = \frac{x_i(t) - \mu}{\sigma}$$

$$\mu = \frac{1}{N} \sum_{i=1}^p x_i \text{ (平均得点)} \quad \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^p (x_i - \mu)^2} \text{ (標準偏差)}$$

2.3 アルゴリズムの手順

以下の手順により、TSPを解析する。
Step1 世代数を $t=0$ とし、*p* 個の個体からなる初期個体群 $P(0)$ をランダムに生成する。
Step2 適応度を計算してセクションを行う。ここで選ばれなかった個体は削除する。
Step3 交叉を行う個体を選び交叉を行う。このときサブ巡回路が生成される可能性があるため、正しい巡回路が生成されるように修正する。
Step4 突然変異を行う個体を選び突然変異を起こす。
Step5 終了条件が満たされれば、そのときに得られる最良の個体を問題の解とする。終了条件が満たされなければ、 $t=t+1$ として、Step 2へ戻る。

2.4 遺伝的操作法

使用する遺伝的操作を以下に示す。

交叉：2-point MX operatorを採用する。これは2つのcut-pointを使用して、お互いの親の1部分を交換する方法である。マトリックス法では交叉のあと、いくつかの短い巡回路ができる可能性があるため1つの巡回路になるように結合が必要になる。

突然変異：任意に2つの都市を選びその間の巡回路を逆にする。

セクション：先の適応関数を用いたルーレットホイール法を採用する。

交叉を行う確率と突然変異を行う確率は事前に設定しているものとし、計算中は固定である。

3. SIMD型遺伝アルゴリズム

3.1 SIMD型並列計算機

SIMD型並列計算機として、多数の比較的簡単なプロセッサ要素(PE)を2次元アレイ状に配置したものを考える。ホストコンピュータに接続される制御ユニット(CU)にプログラムとデータが送られる。PEでの計算結果はホストコンピュータに返される。GAは個々の個体が独立しており、評価、交叉、突然変異は同期的に処理されるため、SIMD向きであると考えられる。

今回は、各 *x*, *y* 座標に単一方向のみ通信経路を持つトーラスを考える。各PE内部は32bitのALU、ランダムジェネレータ、メモリーインターフェイスとネットワークインターフェイスで構成される。マシンサイクルは1MIPSを予定している。(図1)

3.2 SIMD型遺伝アルゴリズム

遺伝アルゴリズムを並列処理でする場合、ルーレット生成や各PEに個体を再配置するセクションの操作が逐次的な処理になるため、処理速度を低下させる原因になる。また、セクションと交叉の操作に必要な通信を並列化し、時間を短縮する方法にする。

Proposal for SIMD Parallel Genetic Algorithm and its Application to the TSP.

Yoshio HAMADA and Yoshizo TAKAHASHI.

Department of Information Science and Intelligent Systems, University of Tokushima.

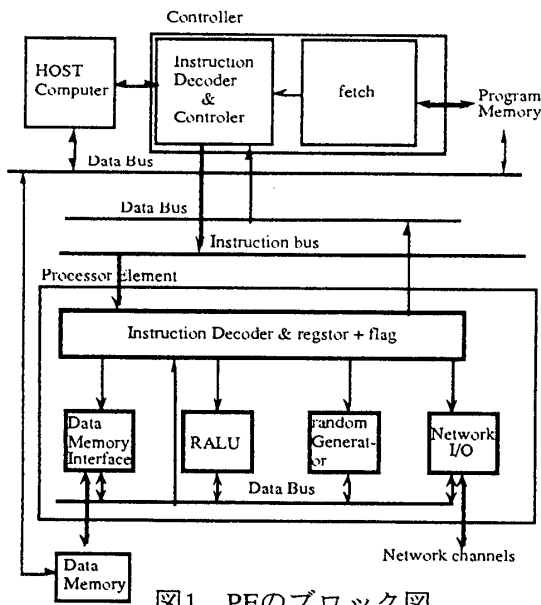


図1. PEのブロック図

各PEには2個の個体を割り当てるものとする。したがって、交叉は1回、評価と突然変異は2回繰り返すだけでよい。

以上のSIMD型遺伝アルゴリズムの手順は次のようになる。

Step1 初期個体群をホストコンピュータで生成して各PEに送る。

Step2 セレクションを各x座標、各y座標を1つのクラスタとして2回行う。

(1) 最初にx座標の同じPE間で評価値を通信してルーレットを生成する。

(2) 各PEでランダムに2つ個体を選択する。

(3) 選択された個体をその個体を持つPEに送る。PE間の距離が等しいものは同時に送る。(図2)

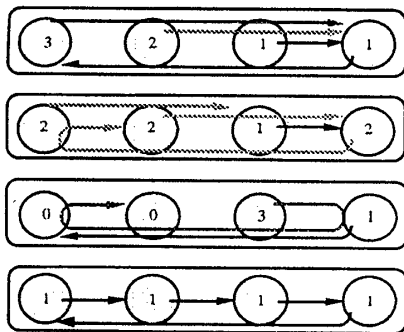


図2. 選択の伝達 (PE内の数字は移動距離)

(4) 各PEのなかで2個以上の個体を選択された場合、あふれた個体は最も近くで空きのあるPEに送る。

Step3 交叉を行う時は各PEが移動させる個体をランダムに選択して、すべての選ばれた個体がランダムにx,y座標を移動して交叉の相手を決定する。(図3)

Step4 各々のPEがランダムに選択した個体について突然変異を起こす。

Step5 得られた解をホストコンピュータに送る。

以上のように最適化に関する計算の部分はまったく逐次型と同じアルゴリズムであるため、簡単に他の問題に適用可能である。

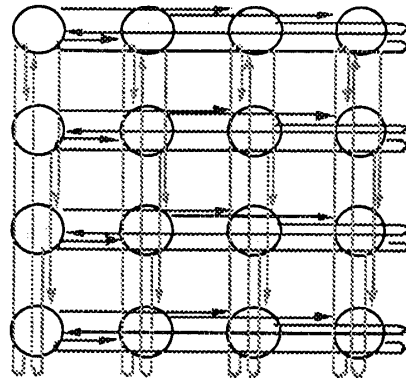


図3. 交叉 X-2 Y-2 移動

4. シミュレーション

実験の各パラメータは以下の表1で表されている。終了条件は1000世代の個体を生成するまでとする。この条件で逐次型とSIMD型についてGAをTSPに適用させたシミュレーションをSS2(27MHz)上で行った。SIMD型における個体の通信は実際の並列計算機と同様に行った。

表1. GAのパラメータ

都市数	30	最短経路	160
個体数	450	世代数	1000
突然変異率	0.005	交叉率	0.6
PE台数	225		

実験結果を表2に示す。試行回数は5回である。終了値の平均値は1回毎にランダムな生成の初期値を変化させて1000世代までで求めた最短経路の平均値、収束世代は始めて最短経路が求まった時の世代、計算時間は逐次型が実測値で、SIMD型は計算に必要な命令数を数えて求めたものである。ただし、計算時間に通信時間は含まれないこととする。

表2. 実験結果

	逐次型	SIMD型
終了値の平均値	161.4	165.0
収束世代	554	706
計算時間(sec)	4767	1229

実験結果より、収束世代は逐次型計算機のほうが早いですが、計算時間はSIMD型のほうが優れていることが示された。

5. おわりに

本稿ではTSP問題を適用の対象として、SIMD型遺伝アルゴリズムが有効かどうかについて検証を行った。通信の時間を考慮にいれていないが、逐次型計算機よりはるかに早い処理能力があることが示された。今後はさらにアルゴリズムを改良し、並列計算機を開発したいと思う。

参考文献

- (1) Zbigniew Michalewicz: "Genetic Algorithms + Data Structures = Evolution Programs", Springer-Verlag, 1992
- (2) 佐藤隆博: "教育情報工学入門", コロナ社, 1989