

# ストリーム計算におけるGAを用いた静的タスク割付

青柳 洋一, 上原 稔, 森 秀樹<sup>††</sup>  
東洋大学工学部情報工学科

## 1 はじめに

HPC の分野においてデータストリーミング [2] の技術の重要度が増している。データストリーミングはパイプラインによって実現され、この分野においてはパイプラインを対象としたタスク割付が重要である。タスク割付は NP 困難な問題とされ、従来ヒューリスティックなアルゴリズムを用いたタスク割付法として CP 法、CP/MISF 法 [3] 等が提案されている。これらの方法は、タスクのクリティカルパスを解析して割付に用いており、タスクに周期性がある場合には、必ずしも最適な割付を求めることができなかった。本アルゴリズムではタスク割付に遺伝的アルゴリズム (GA) を取り入れ、タスク割付の解の探索範囲を拡大して最適解の近似を効率よく発見する手法を提案する。また、本方式では、割付対象のタスクの実行コスト、タスク間の通信のコスト (配置プロセッサの関係によって変化) が解析によって得られることを前提としており、タスク解析の容易でタスク割付に適したストリーム計算モデル [1] を導入した結果、良質な割付が可能であることを確認した。

## 2 GA を用いたタスク割付

本割付法は、多様なタスク割付のパターンの集合を生成し、割付パターンに対してスループットの予測を行って最適割付を決定する手法であり、この操作の効率化に GA を取り入れている。また、周期性を有するタスクを対象とするため、タスク集合の 1 回あたりの実行時間を周期と呼び、スループットを周期の逆数で定義する。

**タスクの割付パターンの生成** GA の染色体はタスクのランダムな並びのリストで表し、染色体からタスクの割付パターンを生成することにする。タスクの割付パターンは、タスクの実行される順序のリスト (以下 実行順リスト) とタスクのプロセッサへの配置 (以下 タスク配置) の組合せで表現し、実行順リストは、染色体をタスクの

<sup>†</sup>Static Task Allocation Using GA on Stream-based Computing

<sup>††</sup>Yoichi Aoyagi, Minoru Uehara, Hideki Mori  
Department of Information and Computer Sciences, Toyo University

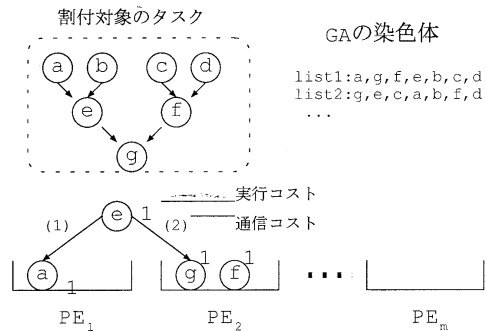


図 1: ナップザック法を応用したタスク配置の生成過程

依存順にソートして最初に得られるものを用いることにした。タスク配置は、高スループットの得られるタスク割付を実現するため、GA の応用例として知られているナップザック法を応用して生成する。図 1 は、タスク配置を生成する過程で、 $m$  台のプロセッサ (PE) を対象に  $n$  個のタスクを染色体 “list1:a,g,f,e,b,c,d” の順番に以下の操作で配置してゆく過程を表している。

- リモート通信コストの削減  
染色体の順番にタスクをプロセッサに配置してゆき、これから配置するタスクと配置済みタスクとの間の通信コストに応じて配置プロセッサを決め、隣接したタスク同士が同一プロセッサに含まれやすくする。
- プロセッサ負荷の均等化  
プロセッサ負荷の偏りを小さくするため、プロセッサ負荷の容量に上限を設ける。

**GA の適用手順** GA は以下の手順で実行される。

1. 染色体の初期集団をランダムに生成する。
2. 以下の (a) ~ (c) の手順を繰り返す。
  - (a) 染色体を用いて割付パターン (実行順リストとタスク配置の組) を生成する。
  - (b) 割付パターンの適応度を求め、GA の評価をする。適応度は割付パターンのスループットに対する理想のスループットで割った値で表現する。理想のスループットは、タスクが均等分割されたと仮定したときのスループットとする。

表 1: 最適タスク割付決定に要した解探索の試行回数

m	GA		BA	
	n <sub>trial</sub>	speedup	n <sub>trial</sub>	speedup
1	1	1	1	1
2	120	1.93	40320	1.93
4	120	3.06	40320	3.06

m: タスク割付に用いたプロセッサ数  
 n<sub>trial</sub>: 割付決定に要した解探索の試行回数  
 (世代数 × 染色体の個体数)  
 speedup: 速度向上率

(c) GA の基本操作である選択、交叉、突然変異を行い、次世代の集団の準備をする。

### 3 タスク割付の評価

**GA 適用による計算量削減** 表 1 に、GA を適用した場合 (GA) と、ナップザック法で用いるリスト (染色体) の組合せを総当たりで探索を行った基本処理 (BA) の比較を示す。評価時の条件として評価機の AP3000 を想定し、ランダムに生成した 8 タスクの割付決定に要した試行 (割付パターン生成・スループット予測の操作) の回数を示したものである。タスクの実行コストを平均 100 ms の指数乱数で与え、通信は 1 回当たり 1024 Bytes のデータを送信すると仮定して、リモート通信コストを 4.187 ms、ローカル通信コストを 0.0125 ms かかるものとした。表 1 において、速度向上率 *speedup* は以下の式で定義した。

$$speedup = T(1)/T(m) \quad (1)$$

$T(1)$  は 1 台のプロセッサへ割付を行ったときのタスクの実行周期、 $T(m)$  は  $m$  台のプロセッサを対象にタスク割付を行ったときのタスクの実行周期であり、実行周期の値は予測値を用いた。

表 1 より、BA における試行回数が 4 万回を超えているのに対し、GA では同等の *speedup* の割付が得られるまでの試行回数が 120 回であり、大幅な試行回数の改善が見られる。このことから GA 適用による、割付決定に要する計算量の削減効果が確認された。

**本割付法と CP/MISF 法の比較** 図 2 のグラフは、タスク数 30 のランダムタスク 10 種類について、分割数と *speedup* (平均) の関係を求めたものである。この実験結果では、分割数が 8 のときを見ると、提案法は CP/MISF

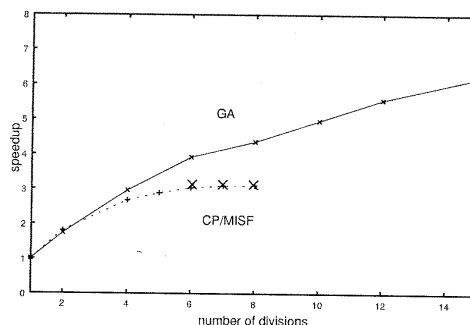


図 2: 分割数と速度向上率の関係 (平均)

法の約 1.4 倍の *speedup* が得られている。図 2 において、CP/MISF 法による割付では 6 ~ 8 分割のときの割付の周期がクリティカルパス長に一致し、並列化の効果の限界となっている [3] (図 2 の × 印)。これに対し、提案法では分割数をさらに増やしても並列度を上げることが可能であり、*speedup* の改善が確認された。

### 4 まとめ

本論文では、周期性を有するタスク割付を対象に、タスク割付に遺伝的アルゴリズムを導入して、広範なタスク割付の候補の中から最適なタスク割付の近似を効率よく求める方法を提案した。ランダムタスクの割付に対する速度向上率の比較を行った結果、提案法が従来法である CP/MISF 法に比べ、スループット向上に対して有効であることを示した。

### 参考文献

- [1] Yoichi Aoyagi, Minoru Uehara, and Hideki Mori. A Case Study on Predictive Method of Task Allocation in Stream-based Computing. *The 12th International Conference on Information Networking (ICOIN-12)*, IPSJ, IEEE Tokyo and Taipei, Jan 1998. pp.316-321.
- [2] Patrick T. Gaughan. Data Streaming: Very Low Overhead Communication for Fine-grained Multi-computing. *In Proceedings of 7th IEEE Symposium on Parallel and Distributed Processing*.
- [3] 笠原博徳, 成田誠之介. マルチプロセッサ・スケジューリング問題に対する実用的な最適及び近似アルゴリズム. 信学論 (D), July 1984. vol.J67-D, No.7, pp.792-799.