

## ニューラルネットワークによるプログラム分割アルゴリズムの改善

齊藤哲哉\* 高田雅美† 城和貴† 國枝義敏\* 福田晃‡

\* 和歌山大学 システム工学部

† 奈良女子大学 理学部

‡ 奈良先端科学技術大学院大学 情報科学研究科

### 概要

分枝限定法を用いて最適なプログラム分割を求める Girkar のプログラム分割アルゴリズムでは、限定操作に用いられる有効な下限値が提案されておらず、また、そのような下限値を求めることは困難であるため、大規模な問題に対して適用しづらかった。本稿では、限定操作を行う際に利用する下限値の予測にニューラルネットワークを導入することを提案し、最適ではない可能性があるが、実用的に問題のない程度の近似解を得ることを目指す。

## Improving a Program Partitioning Algorithm using a Neural Network

Tetsuya SAITO\* Masami TAKATA† Kazuki JOE† Yoshitoshi KUNIEDA\* Akira FUKUDA‡

\* Faculty of Systems Engineering, Wakayama University

† Faculty of Science, Nara Women's University

‡ Graduate School of Information Science, Nara Institute of Science and Technology

### Abstract

This paper proposes an artificial neural network based approach to estimate lower bounds for a program partitioning algorithm with branch and bound method. The program partitioning algorithm is implemented without lower bounds and enough numbers of randomly generated task graph sets are investigated with the algorithm. The internal states of the algorithm are analyzed to design an architecture of neural networks to estimate lower bounds for the algorithm. The designed neural network is confirmed to have the ability to learn (estimate) lower bounds corresponding to the characteristics parameters of given task graph sets. Finally the artificial neural network based approach with a branch and bound based partitioning algorithm is described.

## 1 はじめに

既存の逐次プログラミング言語で記述した、現在まで利用してきたアプリケーションプログラムを並列・分散処理環境で実行したいという要求は、ワークステーションクラスタなどの並列・分散処理環境を容易に、そして安価に構築できるようになった今

日、以前にも増して高まってきている。しかし、そのようなアプリケーションプログラムの多くは、非常に規模の大きなものであるため、プログラムの内容をすべてを理解して、対象となる並列・分散処理環境に適した形で並列プログラムに移植するのは困難な作業である。

自動並列化コンパイラは、このような移植作業に

よる本質的には意味のない負担からプログラムを解放するために、研究開発が進められている。自動並列化コンパイラの役割は、逐次プログラミング言語で記述されたアプリケーションプログラムを詳細に解析し、プログラムの意味を変えずに、対象となる並列・分散処理環境で効率よく実行できる形式に自動的に変換することである。

並列・分散処理環境でアプリケーションプログラムを実行できるようにするためには、複数の計算ノードで並列に実行できるようにアプリケーションプログラムを複数のタスクに分割する必要がある。このように、プログラムをいくつかのタスクに分割することをプログラム分割という。また、プログラム分割に応じて、アプリケーションプログラムが利用するデータについてもそれぞれの計算ノードに分割し、配置する必要がある。特にコンピュータをイーサネットに接続したワークステーションクラスタのように、通信コストが非常に高くつく環境の場合、タスクとデータをどのように分割し、配置するかによって、アプリケーションプログラムの実行性能が大きく左右される。しかし、最適なプログラムやデータの分割・配置を決定する問題は、一般に NP 完全であることが知られている。

Girkar が提案したプログラム分割アルゴリズム [1] は、並列化の対象となるプログラムをタスクグラフとして表現し、分枝限定法を利用して最適なプログラム分割を求めることができる。しかし、Girkar のプログラム分割アルゴリズムでは、ほとんど枝刈りが行われなため、問題の規模が大きくなると同アルゴリズムを適用するのが困難となる。さらに、Girkar のプログラム分割アルゴリズムには有効な下限値が提案されておらず、また、そのような下限値を求めることは極めて困難である。そのため、規模の大きい問題に対して適用するためには、何らかの方法で有効な下限値を得て、限定操作を行うようにする必要がある。

そこで本稿では、Girkar のプログラム分割アルゴリズムの実行効率を改善し、最適なプログラム分割は求められない可能性があるが、実用的に問題のな

いプログラム分割を求めるために、ニューラルネットワークを導入し、分枝限定法の限定操作に利用する下限値を予測することを提案する。

## 2 Girkar のプログラム分割アルゴリズム

プログラムは一般に、閉路を持たない重み付き有向グラフ  $G = (N, E)$  で表すことができる。節点  $n \in N$  はプログラムの一文に相当し、節点のコスト  $t(n)$  は節点を実行するのに必要な時間を表す。枝  $e = (n_i, n_j) \in E$  は節点  $n_i$  から節点  $n_j$  への通信を表し、枝のコスト  $c(e)$  は節点  $n_i$  と節点  $n_j$  が別々のプロセッサに割り当てられた場合に、互いに通信するのに必要なコストを示している。また、枝  $e$  は依存関係を表し、節点  $n_i$  の実行が終了するまで節点  $n_j$  の実行を開始できないことを示している。また、グラフ  $G$  のクリティカルパスのコストは、すべての節点を別々のプロセッサに割り当てることができた場合に、実行に必要な最小の並列実行時間を表している。

Girkar のプログラム分割アルゴリズムは、 $G$  の各枝について、始点と終点の節点を融合する (intra) か融合しない (inter) かで探索木を作成し、分枝操作を繰り返し行って部分解を生成していく。各部分解は、 $\langle G, A, a \rangle$  の形で保持されている。ただし、 $G$  はその時点での分割状態を示すグラフであり、 $A$  は unexamined な枝の集合、 $a$  は  $G$  のクリティカルパスのコストである。暫定解は部分解の中で最小の  $a$  を持つものであり、アルゴリズムの開始時には、初期暫定解として、あらかじめ元のグラフのすべての枝を unexamined とし、節点のコストの最大値を  $a$  とした部分解をリストに入れておく。部分解が持つ  $A$ 、すなわち unexamined な枝の集合が空になれば、すべての枝が intra か inter に分けられたことになり、アルゴリズムは終了する。そのときの  $G$  は最適なプログラム分割を示すグラフとなっている。

Girkar のプログラム分割アルゴリズムにおける分枝限定法の手順をまとめると以下ようになる。

1. 部分解のリストから  $a$  が最小の部分解  $\langle G, A, a \rangle$  を取り出す。
2.  $A$  に unexamined な枝が残っているか調べる。すべての枝が inter もしくは intra に分けられていれば  $A$  は空になっており、アルゴリズムは終了する。このときのグラフ  $G$  が最適なプログラム分割を表している。
3. unexamined な枝  $e = (n_i, n_j)$  を 1 つ選択する。
4. 選択した枝  $e$  を inter にした場合を求め、部分解のリストに加える。
5. 選択した枝  $e$  を intra にした場合を求め、部分解のリストに加える。ただし、3 で選択した枝の始点  $n_i$  から終点  $n_j$  へ至るすべてのパスに含まれる枝の中に inter になっている枝が存在する場合には、融合する際にループができてしまうため、部分解のリストに追加しない。
6. 1 に戻る

Girkar のアルゴリズムでは、最小の  $a$  という自明な下限値と元のグラフのクリティカルパスのコストを上限値とする限定操作と、節点を融合する際にループができてしまうような場合を除くことにより枝刈りを行っているが、それではまだ実用上不十分である。そこで、有効な下限値が必要となるわけだが、Girkar のプログラム分割アルゴリズムに対して有効な下限値は提案されておらず、また、そのような下限値を求めるのは非常に困難である。そこで我々は、ニューラルネットワークを導入して下限値の予測を行ってみることにした。

### 3 ニューラルネットワークの導入

分枝限定法を用いてプログラム分割を実現する場合、前述したように上限はタスクグラフのクリティカルパスのコストで与えることが可能であり、これは静的に計算可能である。しかしながら、下限に関しては他のスケジューリング問題等に対する分枝限

定法の適用と異なり、静的に求めるのは極めて困難である。実際、分枝操作の初期段階におけるノードの融合（プログラム分割では、タスクグラフのノードに対して、計算コストと通信コストを考慮して融合／非融合を決める）は、それらのノードに依存関係を持つ分枝操作終盤のノードの融合／非融合に直接影響を与えるため、正確な下限値を算出するには結局全探索が必要となる。

一方、我々はプログラム分割を行う際に、下限による限定操作を行わずにヒューリスティック値を利用した方法を提案している [2]。このヒューリスティック値は、プログラムの実行とデータ参照の局所性を利用して計算されるものであり、得られる分割は分枝限定法により求められる最適解に対し、平均 10% 以内の誤差であるのに対し、分枝限定法では探索空間の爆発的増大のために解が得られなかった分割問題も、解を得ることが可能となった。つまり、プログラム分割を行う際に、探索途中のローカルな情報のみでヒューリスティックを利用しても、最適解に近い解を得ることが可能であり、しかも解を得るために必要とするメモリ量は大幅に減少する。

これらの経緯を踏まえて、我々はプログラム分割を分枝限定法を利用して行う際に、正確な下限値を計算する代わりに、探索途中のローカル情報と与えられたタスクグラフ全体の性質を表すパラメータを与えることで、下限値の予測が可能ではないかと考えたのである。このような予測は、モデル化を行い定式化して求める他に、ニューラルネットワークの利用が考えられる。以下に、当該予測を行うためのニューラルネットワーク構築法について説明する。

Girkar のプログラム分割アルゴリズムの下限値を予測するために、誤差逆伝搬法によって学習を行うフィードフォワード型ニューラルネットワークを利用する。考えられる入力パラメータを以下に挙げる。

- 節点の数
- 枝の数
- 節点のコスト（平均、最大値、偏差）
- 枝のコスト（平均、最大値、偏差）

- クリティカルパスのコスト
- unexamined, intra, inter の割合
- 入次数, 出次数

下限値を予測するためには, 各部分解で生成されているグラフの形状から評価値  $a$  の予測値を学習させる必要がある. そこで, Girkar のプログラム分割アルゴリズムを適用して求められる範囲の小さな問題(枝数 30 程度)に対して, 最適なプログラム分割が得られるまでの各部分解を記録しておき, それらを教師データとしてニューラルネットワークの学習に利用する.

実際に Girkar のプログラム分割アルゴリズムに対して予測した下限値を利用する場合には, 各部分解から上記パラメータを抽出し, ニューラルネットワークに通して予測した下限値を得て, その値と部分解の持つ評価値とを比較して限定操作を行う. 分枝限定法を用いた Girkar のプログラム分割アルゴリズムでは, 最適なプログラム分割を求めることができたが, 予測した下限値を利用する場合には, 最適なプログラム分割ではないプログラム分割が求められる可能性がある.

しかし, もともと Girkar のプログラム分割アルゴリズムで利用している節点や枝のコストは静的に求めたものであり, 特に通信量をパラメータとして利用する Girkar のプログラム分割アルゴリズムでは, 最適なプログラム分割が最適な並列実行時間に直結するわけではない. 従って, 最適なプログラム分割を求めるために膨大な時間を消費せずに, プログラム分割を求めるのに必要となる計算時間と生成されるプログラム分割の精度とのトレードオフを考えながら, 実用的に問題のない近似解も積極的に採用する方針を取ることにした.

## 4 おわりに

本稿では, 分枝限定法を利用した Girkar のプログラム分割アルゴリズムに対して, 下限値を予測するためにニューラルネットワークを導入することを提

案した. 学習によって予測した下限値によって限定操作を行うことで, Girkar のプログラム分割アルゴリズムをより規模の大きい問題に対して適用することができるようになることが期待できる. 現在, 本手法を用いて, 下限値を予測するための学習は収束することは確認されているが, 実際の分枝限定法への利用までには至っていない.

今後は, ニューラルネットワークの学習を進めて, どのパラメータがどう影響しているのか, また, 実際にどれくらいの精度でプログラム分割を行えるかを詳細に調査し, 実際に自動並列化コンパイラに組み込んで利用できるように開発を進めていく予定である.

## 謝辞

本研究の一部は日本学術振興会未来開拓学術研究推進事業(知能情報・高度情報処理分野)JSPS-RFTF96P00505 の援助による. また, Girkar のプログラム分割アルゴリズムに関して, 詳細な情報と有意義なコメントを頂いた Polychronopoulos 教授に深く感謝する.

## 参考文献

- [1] Girkar, M. and Polychronopoulos, C.: Partitioning Programs for Parallel Execution, *Proceedings of International Conference on Supercomputing*, pp. 216-229 (1988).
- [2] 齊藤哲哉, 高田雅美, 城和貴, 國枝義敏, 福田晃: 実用的な近似解を与えるプログラム分割アルゴリズム, 情報処理学会研究報告 MPS-27-8, pp. 29-32 (1999).