1N-3

動的計画法の汎用並列化ツールの作成*

樫本　悠介†
東京農工大学大学院　工学教育部

品野　勇治‡
東京農工大学大学院　共生科学技術研究部

1．はじめに

本研究では，組合せ最適化問題に対する解法の１つである動的計画法を，PC クラスタなどの並列計算環境上で並列実行させるためのシステムを構築した．動的計画法には，並列実行可能な部分がある．しかし,対象とする問題に対する解法の違いに依存する部分もあるため，汎用的な並列化システムの構築は困難となる．

参考文献[1]にあるように，既存の汎用並列化システムの構築も試みられているが，単一の並列化方式のみしか適用できないので，特徴的な性質を持つ解法のみを対象としていると考えられる．本研究では，解法の特徴に応じて並列化方式を選択可能とすることで，動的計画法の汎用並列化ツールの構築を試みた．構築したシステムでは，対象とする問題とは独立に，実行時に並列化方式の指定が可能である．

システムの検証は，複数の組合せ最適化問題を取り上げ，その並列化を考えた場合に特徴的な違いが生じる動的計画法を構築した汎用ツールを用いて実装して行った．

2．動的計画法の並列性

動的計画法は，最適解が，その一部だけに注目しても，対応する部分問題の最適解になっているという構造（最適性の原理）で特徴付けられる問題に対して適用できる．最適解の値は再帰的に定義され，各状態の最適値は段階的に計算される．この各状態の最適値計算の際に，前の段階で計算済みの最適値をテーブルへ記録し，余分な計算を省いている．

M 状態の最適値をテーブルに管理する必要のある問題の動的計画法による計算過程を考える．k 段階目の計算における，ある状態 i の最適値を $G_{k,i}$ と表すと，再帰的に定義された関数方程式によって，計算に必要となる前の段階における最適値 $G_{k-1,m}(0 \le m \le M)$ の参照関係は各問題と解法に依存し，密な場合や疎な場合が生じる．

この参照関係により，各最適値計算には依存関係を生じるが，複数の最適値計算は同時に行えるので，動的計画法の解法には並列実行可能な構造がある．

例えば動的計画法を最短経路問題に適用した場合，始点から終点の枝の数を段階にとると，$G_{k,l}$ は，枝数 k 本以内での，始点から頂点 l への最短経路の距離となる．このとき，最適値の参照関係は図 1 となり，密な参照関係が存在する．

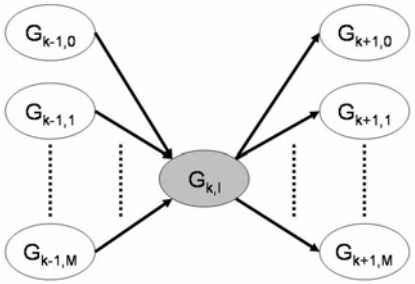

図１　最短経路問題における最適値の参照関係

3．システムの構成

システムは C++言語により開発し，図２に示すような３群のクラス群から構成される．システムの利用者は，利用者定義クラス群を解きたい問題専用に開発する．このクラス群を逐次実行環境構築用クラス群と組み合わせることで，逐次実行のためのプログラムが作成でき，並列実行環境構築用クラス群と組み合わせることで，並列実行のためのプログラムが作成できる．

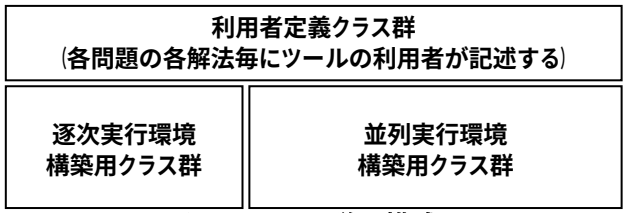

図２　クラス群の構成

利用者定義クラス群は，各環境構築用クラス群内のクラスから派生する次の３クラスである．まず，問題固有の入力データ形式を解釈し，ツール内部のデータ構造へ変換する，**データ入力クラス**，問題固有の関数方程式を定義する**関数方程式定義クラス**，そして，問題固有の計算結

*"An Implementation of a Generalized Parallelization Tool for Dynamic Programming Algorithms"

† Yusuke Kashimoto, Graduate School of Engineering, Tokyo University of Agriculture and Technology.
‡ Yuji Shinano, Institute of Symbiotic Science and Technology, Tokyo University of Agriculture and Technology.

果を表示するための**結果出力クラス**である．

4．並列実装

並列化は，値$G_{k,i}$を複数同時に実行することで実現する．この並列実行を可能にする実装に際しては，値$G_{k,i}$の計算プロセスを，どのようにプロセッサへ配置し，どのタイミングで同期を取るかの違いにより，図３に示す２つの並列化方式を実装した．これらの並列化方式は，プログラム実行時に指定できる．メッセージパッシングライブラリとしては，MPI を利用している．

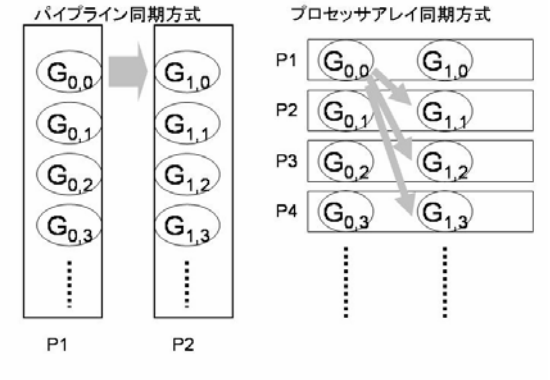

図３　計算プロセスのプロセッサへの割当

パイプライン同期方式では，プロセッサをリング状に配置された１次元プロセッサアレイとみなす．同じ段階の最適値計算のプロセスを１プロセッサへ割り当て，１状態１段階の計算が終わり次第，隣接するプロセスへ結果を送信し，次の計算へ移行する．一般に１状態１段階の計算の粒度は小さいので，通信のオーバーヘッドが相対的に大きくなる．そこで，１プロセスが複数状態複数段階の計算をまとめて実行することによる並列処理の粗粒度化や,データ送信時のデータのブロック化などを制御することで通信のオーバーヘッドを減らしている．

ブロードキャスト同期方式は，同じ状態の最適値計算プロセスを１プロセッサへ割り当て，１状態１段階の計算が終わると，全プロセッサへ結果をブロードキャストする．１プロセスが，いくつの状態の最適値計算を行うかは，実行時のプロセッサの数で決まり，各プロセスが計算する状態数を均等にしている．

5．評価実験

構築したシステムで複数の組合せ最適化問題（ナップザック問題，最短路問題，資源配分問題）の動的計画法を，逐次実行，パイプライン同期方式による並列実行，ブロードキャスト同期方式による並列実行したときの実行時間を測定し，スピードアップを求めた．評価実験には，PC が 10 台，100BaseT のネットワークで１つのスイッチに接続されたクラスタを利用した．クラスタを構成する各マシンは PentiumⅢ 1GHz CPU を２個，1GB メモリを持つ． MPI の実装としては，MPICH を利用した．

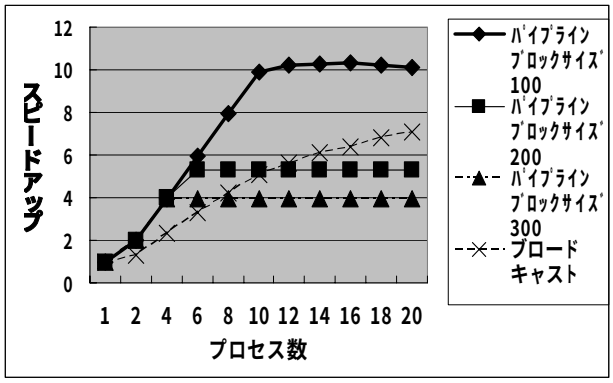

図４　資源配分問題における実行結果
(資源配分問題(配分場所 2000 資源数 4000)

例として，図 4 に資源配分問題に構築したシステムを適用した結果を示す．図 4 でパイプライン同期方式のブロックサイズは通信の際にブロック化する状態数を表す．資源配分問題では，ブロードキャスト同期方式では約７倍，パイプライン同期方式（ブロックサイズ 100）では約 10 倍のスピードアップを得た．他の問題でも，パラメータを調整することで同程度のスピードアップを得ている．

6．まとめ

本研究では，最適値計算の依存関係により，並列化方式を選択可能な動的計画法に対する汎用並列化ツールを開発した．開発したツールは，既存のツールと比較すると，広範囲の問題に対して適用可能であり，解法の特徴によらず，20 CPU による PC クラスタを利用して，10 倍程度のスピードアップを得た．また，汎用ツールによる実装を行ったため，同じ解法の同一プログラムを，異なった並列化方式で動作させたることが可能になり，解法の特徴と並列化方式の違いによる効果を明確に調べることが可能となった．

参考文献

[1]D.Morales,F.Almeida,F.Garcia J.Gonzalez J.Roda C.Roudriguez, “A Skeleton for Parallel Dynamic Programming,”Euro-Par’99, LNCS 1685,pp.877-887,1999.