

## 完全分散型分枝限定法並列化ツールの設計

02201800 東京理科大学 \*品野 勇治 SHINANO Yuji  
01505820 刈羽システムズ 檜垣 正浩 HIGAKI Masahiro  
東京理科大学 原田 賢一 HARADA Kenichi  
01504250 東京理科大学 平林 隆一 HIRABAYASHI Ryuichi

### 1.はじめに

分枝限定法はクラス NP に属する組合せ最適化問題を解くための代表的な解法である。特定の問題の特徴や性質を利用した下界値計算法や分枝変数の選択を工夫することで、各種問題毎に解ける問題規模が拡大してきている。しかし、分枝限定法を用いても問題規模がある規模以上になると、現実的な時間内に解けるインスタンスの数は極端に少なくなることが知られている。

一般的な枠組みとしての分枝限定法は、十分に研究され列挙法に基づく計算原理として確立している。並列分枝限定法は、多数のプロセッサを利用し並列に解空間を列挙することで、少しでも現実的な時間内に解けるインスタンスを増やすための試みである。

計算量が問題規模に対して指数的に増加する組合せ最適化問題に並列処理を適用し、数倍の計算時間の短縮が可能となっても、本質的に解ける問題規模を拡大するとは考え難い。しかし、一定規模の問題までは必ず解けるような解法は、クラス NP に属する問題では開発できないと考えられている。規模の大きな問題でも、少数のインスタンスに限れば分枝限定法によって、現実的な時間内に解かれている。並列処理の適用は、この現実的な時間内に解けるインスタンスの量を少しでも拡大するための手段である。

本研究では、筆者らが開発し実験をしてきたマスター・スレーブ型の並列分枝限定法で得た知見により、どのように完全分散型分枝限定法並列化ツールが設計されたかについて説明する。

### 2.マスター・スレーブ型分枝限定法並列化ツール

過去に開発されたマスター・スレーブ型の分枝限定法並列化ツール[1]は、ワークステーション群上に実装され、以下の特徴を持つ。

- 並列分枝限定法のための汎用ツールであり、特定の問題に依存しない並列分枝限定法のスケルトンを提供する。
- 割り込み機構を利用した迅速な限定操作を実現する。
- 深さ優先探索と下界値優先探索を組み合わせたハイブリッド探索を実装している。

マスター・スレーブ型分枝限定法並列化ツールにより、比較的簡単に各種問題に対する並列分枝限定法を記述できた。そこで、巡回セールスマン問題、整数計画問題、輸送制約付き枝巡回路問題を実装し、数値実験を実施した。それぞれの実験結果において、多数の超線形加速が観測され、顕著な並列化の効果が示された。

しかし、マスター・スレーブ型の実装では、子問題群を管理するための記憶容量は依然1台の計算機の限界を超えない。したがって、計算時間が短縮されたとしても、これまでで解けなかったインスタンスを解くことに対して貢献するものではなかった。

### 3.完全分散型分枝限定法並列化ツールの設計思想

以下の設計思想に基づいて開発された。

#### 3.1.分枝限定法の枠組みを提供する

並列分枝限定法の実装を、本質的に問題固有のルーチンのみを記述することで実現する。特に、各問題毎の実装を行うユーザの観点からは、逐次の分枝限定法としての動きが、明確に意識できるスケルトンを提供する。

#### 3.2.並列分枝限定法実行環境と下界値計算法など各問題固有のルーチンの開発環境を分離する

マスター・スレーブ型のツールでは、問題固有のアルゴリズムの実装に際して、複数プロセスが動作する環境でデバッグを行う必要があり、依然開発は困難なものであった。そこで、問題固有の下界値計算等のルーチンの開発は、並列動作環境とは独立に逐次処理の分枝限定法として開発できる環境を提供する。問題固有のルーチンのデバッグが完了した時点で、再コンパイル程度の手間で並列分枝限定法の動作環境への移行を可能にする。

#### 3.3.プロセッサの演算能力だけでなく、メモリの有効利用も図る

並列分枝限定法では、計算途中に保持すべき子問題の数が、逐次処理する分枝限定法と比較して増加する傾

向がある。現実的に解けるインスタンスを増やすためには、計算途中に保持すべき子問題群を管理できるだけのメモリ容量の確保が必要である。マスター・スレーブ型では、スレーブが割り当てられたプロセッサ側のメモリには子問題群を保持しないため、スレーブ側のメモリの有効利用が図られていない。完全分散型では、各プロセッサがローカルに子問題群を保持し、プロセッサの数を増やすことで、計算途中に保持できる子問題数を増やせるようにする。

### 3.4. 特定の計算機環境に依存せずに動作する

並列計算を行うための計算機環境を選ばないシステムであれば、実際に並列計算を行える環境の入手が容易となる。今日では、標準のメッセージ・パッシング・ライブラリを使用することで、特定の計算機環境に依存しない並列処理システムの構築が可能である。

### 3.5. 動作環境を並列分枝限定法の実行を停止することなく、動的に変更できる

近年、コンピュータ・ネットワークは、飛躍的に拡大してきている。日時を指定したり、誰も使用していない間という条件のもとで、計算機環境を入手できれば使用可能なプロセッサ数の増加が望める。これまでに解けなかったインスタンスを解くという観点からは、利用可能なプロセッサをできる限り入手し、並列分枝限定法を実行する必要がある。

また、効率の観点からも、計算に必要なだけの資源を動的に追加・削除できることのメリットは大きい。このことは、分枝限定法を数台のプロセッサで実行させ、必要に応じて、プロセッサを追加しながら解くことを可能にする。さらに、特定の計算機環境に依存しないシステムであれば、必要に応じて超並列計算機の追加をも可能となる。

### 3.6. 1000 個以上のプロセッサを使用した場合でも、負荷分散が適当に機能する

完全分散型のシステムでは、マスター・スレーブ型でボトルネックとなった、マスターへの処理の集中や、子問題を管理するためのメモリ容量不足は解消される。一方、マスター・スレーブ型での限定操作が有効に機能したのは、子問題群を集中管理することで、解の探索規則を完全に制御できたことによる効果が大い。

完全分散型のシステムでは、各計算機がローカルに子問題群を保持するため、ある計算機では子問題の枯渇や、あふれが生じることもあり、解の探索規則を完全に制御することはできない。また、枯渇やあふれに対応するように、子問題をプロセッサへの割り当てる負荷分散の仕組みも複雑になる。さらに、限定操作が機能するタイミングのずれも、マスター・スレーブ型と比較すると大きくなる。そのため、一般に完全分散型の並列分枝限定法の実装では、マスター・スレーブ型ほど顕著な計算時間の短縮は得られていない。

しかし、これまでに解けなかったインスタンスを解くという観点からは、完全分散型のシステムが不可欠である。完全分散型システムでの効果は、負荷分散の仕組みに依存する。しかも、効果の検証は実装と数値実験によってのみ検証される。よって、1000 個以上のプロセッサを使用しても、負荷分散が適当に機能するシステムを目標とし、現実的には負荷分散の仕組みを実装する部分の独立性を保ち、負荷分散の仕組みを変更可能なシステムを構築する。

## 4. 完全分散型分枝限定法並列化ツールの実装について

上述した設計思想に基づいたシステムの実装を、C++言語とPVM(Parallel Virtual Machine)を使用して行った。基底クラスを本ツールが提供し、各問題固有部分は解法毎に派生クラスとして記述する。よって、ツールにより提供される部分から、問題固有部分の記述が明確に分離された。現在の動作確認はワークステーション群上で行っているが、標準メッセージ・パッシング・ライブラリPVMの利用により、計算機環境からも独立したシステムとして実装されている。

## 5. おわりに

紙数の都合上、実装の詳細は記述できないが、発表当日は各設計思想がどのように実装されているかを詳細に説明したい。さらに、巡回セールスマン問題を例とした数値実験結果も報告する。

## 参考文献

- [1] Y. Shinano, M. Higaki, R. Hirabayashi, "A Generalized Utility for Parallel Branch and Bound," Proc. Of 7th IEEE Symposium on Parallel and Distributed Processing, Texas, 1995