

## 汎用並列分枝限定法ツール PUBB を用いた最大クリーク問題の厳密解法

01207140	東京理科大学	*品野 勇治	SHINANO Yuji
02501520	東京工業大学	藤江 哲也	FUJIE Tetsuya
01012660	東京理科大学	池辺 淑子	IKEBE Yoshiko
01504250	東京理科大学	平林 隆一	HIRABAYASHI Ryuichi

## 1 はじめに

無向グラフ  $G = (V, E)$  のクリーク  $C (\subseteq V)$  とは,  $C$  の任意の 2 頂点が枝で結ばれているものをいう. 要素数最大のクリークを求める最大クリーク問題は, NP 困難な問題として知られ, 厳密解法やヒューリスティック解法が多く提案されている. 特に厳密解法のほとんどは分枝限定法に基づくものである. 本稿では, 品野ら [5] によって開発された汎用並列分枝限定法ツール PUBB (Parallelization Utility for Branch-and-Bound algorithms) を用いた, 最大クリーク問題に対する厳密解法の実装及び実験結果について報告する.

本稿では, Balas and Xue [1], Manino and Sansano [4] による点彩色上界を用いた分枝限定法に, Ikebe and Tamura [3] によるヒューリスティックを組み合わせた解法を用いている. PUBB は, 問題に依存したサブルーチン (下界値・上界値の計算, 子問題の生成, など) の設計を行えば, 逐次・並列を問わず分枝限定法の実行が可能なシステムである. そこでまず, PUBB を用いて逐次分枝限定法を設計し, 数値実験を行って各サブルーチンにおけるパラメータを設定した. 次に, 並列分枝限定法を, 最高 51 台のワークステーション上で実行した. ここで, 上界値の計算時間がワークステーション間の伝達時間よりもはるかに短かったため, PUBB の子問題管理機構における負荷分散方式に改良を要した. 従来 PUBB は, 上界値計算に時間を要さない解法の並列化には向いていなかった. しかし, 適切な改良を加えることによって, 上界値が短時間で計算される解法でも, 十分なパフォーマンスが得られた. 特に, (著者らの知る範囲では) 最適性が保証されていなかった DIMACS ベンチマーク問題を 5 問解くことができた. したがって, PUBB は上界値の計算時間の長短 (粒度の大小) に依らない, 非常に適用範囲の広いシステムになったといえる.

## 2 解法の概略

本節では, 紙面の都合上, 分枝限定法の概略は既知であるとして, 分枝限定法の設計に必要なサブルーチンの説明を行う.

## 初期実行可能解

我々の分枝限定法は, 実行可能解を求めることから始まる. 近年多くのヒューリスティック解法が提案されており, その多くは, ほとんどの問題例に対して最適解もしくはそれに近い解を出力する. 本稿で適用した, Ikebe and Tamura [3] による STABJOIN も同様の傾向を示した.

## 子問題の探索

PUBB では, 子問題探索法を指定することが可能である. 本稿では, 1 回の分枝操作によって多くの子問題を生成するため, 深さ優先探索を適用している. ただし, 並列化の際改良が加えられている.

## 分枝方法

[4] に基づいている. 1 回の分枝操作で生成される子問題数は暫定値に依存する. 初期解の生成に用いた STABJOIN は, 最適値, あるいは極めて最適値に近い目的関数値をもつ解を生成するため, 総子問題数を減らすためにも有効であった.

## 上界値の計算

$G$  の安定集合  $S (\subseteq V)$  とは,  $S$  の任意の 2 頂点が枝で結ばれていないものをいう.  $S = \{S_1, \dots, S_K\}$  を安定集合による頂点集合  $V$  の分割とする.  $S$  は  $G$  の点彩色と呼ばれる. そして, よく知られているように, クリークの最大要素数は高々  $K$  であり, 点彩色の彩色数が上界値を与える. 一般のグラフに対して彩色数を求めることは NP 困難であるため, ヒューリスティック解法が多く提案されている. 本稿では, それらの解法の中で, greedy な解法と Brelaz [2] の方法を試みた ([1] にそれらの記述が見られる). また, [1, 4] によって提案されている, “fractional な点彩色” に基づく上界値の計算も実装した. この上界値は, 単なる点彩色に等しい, または良い上界を与えることが示されている. 実際, 生成される総子問題数を減らすことができたが, 逆に我々の実装では総計算時間は増加してしまった. したがって, 単なる点彩色を上界値として用いた. また同様の理由で, greedy な解法を採用した.

### 3 PUBB を用いた並列化

#### 3.1 PUBB

PUBB は、PVM(Parallel Virtual Machine) 上で動作し、主に、負荷分散を司る LB (Load Balancer), 子問題の評価計算を行う Solver, 暫定解の管理を行う PM (Problem Manager) の3種のタスクで構成される。通常は、ネットワークで接続されたワークステーション群を並列計算機として機能させ、各ワークステーションには LB と Solver が対をなして動作し、別の1台で PM を動作させる。PUBB は以下の3種類の実行モードを用意している(詳細は [5] を参照)。

##### MS (Master-Slave) モード

PM が未探査子問題群の一元管理を行い、子問題の授受は PM と Solver の間で行われる。

##### FD (Fully-Distributed) モード

各 Solver は逐次分枝限定法を実行し、同時に未探査子問題群の管理を行う。よって、子問題群は分散管理される。ただし、一つの子問題を解き終えるたびに、対となっている LB に情報(未探査子問題数や暫定値など)を送る。LB は情報を受け取り、未探査子問題数がパラメータ数よりも小さい場合は、他の Solver から子問題をもらうべく、パートナーとなる Solver に対応する LB を探し出す。

##### MStoFD (MS to FD) モード

このモードは、初め MS モードで実行し、PM で保管している、未探査子問題数がパラメータ数を上回った場合 FD モードに移行する。

#### 3.2 改良

すでに述べたように、子問題を解く時間、すなわち上界値の計算時間がワークステーション間の伝達時間よりもはるかに短かった(ワークステーション間の子問題伝達時間は平均 0.01 秒以上に対し、頂点数 200 の問題で、上界値平均計算時間は 0.002 秒以下)。そのため、本稿では2つの改良を施した。

一つは、Solver が LB に情報を送る部分である。これまでは、Solver が子問題を解くたびに情報を送っていたが、一定時間は送らないという制限を加えた。予備実験の結果、この時間を1秒に設定することによって、全体の計算時間は半分以上減少した。

もう一つは、Solver 間で子問題を授受する部分である。逐次分枝限定法を実行している各 Solver では深さ優先探索を採用しているが、Solver 間での子問題の授受は上界値優先で行うようにした(Best First Tree Search with Best Bound First Transfer)。これは、上界値の大きい子問題は解く

のに時間がかかる、すなわち多くの新しい子問題を生成すると考えられ、未探査子問題数が少ない Solver に送ることで、子問題数の総授受回数の減少が期待されるからである。実際、予備実験によって、Solver 間の子問題の総授受回数は10分の1から20分の1程度まで減少し、全体の計算時間もほぼ半分になった。

#### 3.3 DIMACS ベンチマーク問題

最後に、DIMACS のベンチマーク問題の中で、今までに最適性の保証がされていなかったとされる5問を解いた結果を表1に示す。動作環境は、IBM RS/6000 Model 25T [CPU:PowerPC601(66MHz), Memory:64Mbyte] 51台 (Solver 数は50) であり、イーサネットに結ばれている。表1で、“枝密度”は完全グラフの場合に対する枝数の割合を、“S.E.(Speedup Estimation)”は(上界値計算に要したCPU時間)/(実行時間)を示している。定義より、“S.E.”は実際の Speedup の下界値を与えている。

表1が示すように、良好な Speedup を実現することにより、規模が大きい問題を厳密に解くことができた。ただ暫定解の更新があったのは p\_hat1500-2 だけであり、しかも更新回数は1回であった。つまり、STABJOIN の出力する解は非常に精度が良かったことを示している。

表1: DIMACS ベンチマーク問題の計算結果

問題	頂点数	枝密度 (%)	計算時間 (秒)	生成された子問題数	S.E.
C250.9	250	89.9	25768.79	687551803	38.1
p_hat500-3	500	75.2	856.65	12398047	34.5
p_hat700-3	700	74.8	10637.92	131620573	43.1
p_hat1000-2	1000	49.0	1111.23	16098000	35.7
p_hat1500-2	1500	50.6	61694.40	628386474	45.2

#### 参考文献

- [1] E. Balas and J. Xue, “Weighted and Unweighted Maximum Clique Algorithms with Upper Bounds from Fractional Coloring,” *Algorithmica* 15 (1996) 397-412.
- [2] D. Brelaz, “New Methods to Color the Vertices of a Graph,” *CACM* 22 (1979) 251-256.
- [3] Y.T. Ikebe and A. Tamura, “Ideal Polytopes and face structures of some combinatorial optimization problems,” *Math. Prog.* 71 (1995) 1-15.
- [4] C. Manino and A. Sassano, “An Exact Algorithm for the Maximum Stable Set Problems,” *Computational Optimization and Applications* 3 (1994) 243-258.
- [5] Y. Shinano, K. Harada, and R. Hirabayashi. “Control Schemes in a Generalized Utility for Parallel Branch-and-Bound Algorithms,” In *Proc. of the 11th International Parallel Processing Symposium*, (1997) 621-627.