

PC クラスタ環境における並列分枝限定法

01207140 東京農工大学 *品野 勇治 SHINANO Yuji
02501520 神戸商科大学 藤江 哲也 FUJIE Tetsuya

1 はじめに

近年, PCの価格低下は著しく, 価格性能比はワークステーション(以下ではWSと略す), 並列計算機に比べ相対的に高くなっている. 本稿では, このような現状を踏まえ, PCクラスタ環境を用いた並列分枝限定法の最近の結果を報告する. まず, 著者らがこれまでにこなってきた, WS群を利用した動作環境での数値実験の結果と, 同じプログラムをPCクラスタ上で動作させた数値実験の結果を比較する. その後, 二次割当問題について, 1997年の時点で厳密解が得られている最大規模の問題[1]をPCクラスタによって解いた結果を報告する.

2 PC クラスタ環境

PCクラスタを構成するPCは, エプソンダイレクト社のEndeavor AT-700C(CPU:Pentium II 400MHz, メモリ:256M)である. このPC 21台を3Com社製Super Stack II Baseline 10/100 Switch 24 portで接続してPCクラスタを構成した. また, OSにはLinux(RedHat 6.0)を使用し, 並列分枝限定法を実現するソフトウェアは, 品野らが開発したPUBB(Parallelization Utility for Branch-and-Bound Algorithms)[3]を利用した.

3 最大クリーク問題

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. 最大クリーク問題とは要素数最大のクリークを求める問題である. [4]では, PUBBを用いた並列分枝限定法を実装し, WS群により数値実験を行った.

3.1 WS群の環境

WSは, IBM RS/6000 Model 25T(CPU:PowerPC601(66MHz), メモリ:64M)を用い, 各WSは10M bpsのイーサネットにより接続されている.

3.2 数値実験のデータとパラメタ設定

数値実験には, ランダムに生成した200頂点, 枝密度90%の問題10問を用いた. PCクラスタ環境での数値実験でも, [4]と同じプログラムを用いた.

ただし, 初期暫定解を得るときに利用している乱数が動作環境に依存したため, 初期暫定値が異なる場合がある. また, PUBBのパラメタであるNotification Interval Time(Solverにおける子問題群の情報の通知間隔)は, WS群を利用した場合1秒であったのに対して, PCクラスタ環境では0.5秒に設定している. この変更は, PCクラスタ環境の方が単体の計算時間も通信速度も速いことによる. また, 並列動作に伴う非決定的な挙動の影響を排除するため, 5回の試行による平均値で比較する.

3.3 数値実験結果の比較

まず, 今回利用したWSおよびPC単体の計算時間を比較するため, 10問の中で暫定解の更新がなかった(初期暫定解が最適であった)6問について計算時間の和をとった. すると,

$$\frac{\text{WS単体の計算時間の和}}{\text{PC単体の計算時間の和}} = 5.0$$

という結果を得た. すなわち, 計算機単体の計算速度はPCの方が5倍程度速いと考えられる.

次に, 両動作環境における並列分枝限定法の数値実験結果を表1に示す. 両環境とも21台(Solverは20個)のWS/PCを使用している. 表1において, “加速率”は次のように定義されている:

$$\text{加速率} = \frac{\text{逐次処理での計算時間}}{\text{20個のSolverによる並列計算時間}}$$

また, “更新”は, 暫定解の更新の有無を示している. 表1が示すように, Notification Interval Timeを適切に設定することによって, PCクラスタ環境においても十分な加速率が得られ, 並列の場合にも5倍程度速く解くことができた.

3.4 ベンチマーク問題での比較

[4]では, DIMACSチャレンジ問題集から, 1997年の時点において最適解が得られていなかった問題を5問解いた. その中で最も計算時間を要したp_hat1500-2(頂点数:1500, 枝密度:50.6%)をPCクラスタ環境で解いた. その結果を表2に示す. WS群の環境では, 51台(Solverは50個)のWSを使

用した(PCクラスタ環境では21台のPCを使用). この実験の試行回数は1回である. 表中の“子問題数”は, 評価計算(上界値の計算)を行った子問題の数である.

表 1: WS 群と PC クラスタの比較

Prob. No.	WS 群			PC クラスタ		
	Time(sec)	加速率	更新	Time(sec)	加速率	更新
1	1006.08	24.1	有	164.18	16.6	無
2	1082.22	20.2	有	211.12	17.4	有
3	1950.58	16.1	無	383.22	16.8	有
4	821.63	15.6	無	156.46	16.2	無
5	1468.25	15.9	無	265.33	16.9	無
6	769.58	12.9	無	144.41	16.3	無
7	1570.68	16.1	無	293.01	17.0	無
8	1462.07	16.1	無	267.71	17.0	無
9	1214.67	15.7	有	230.47	15.8	有
10	1073.00	16.0	無	199.96	16.7	無

表 2: p_hat1500-2 に対する計算結果

WS 群			PC クラスタ		
Time(sec)	子問題数	更新	Time(sec)	子問題数	更新
61694.40	628386468	有	34810.38	623538050	無

4 二次割当問題

二次割当問題 (Quadratic Assignment Problem: QAP) は, 2つの $n \times n$ 行列 $F = (F_{ij}), D = (D_{ij})$ が与えられたとき, 次のように定式化される問題である:

$$\min_{\pi \in S_n} \sum_{i=1}^n \sum_{j=1}^n F_{\pi(i)\pi(j)} D_{ij}.$$

ここで, S_n は $\{1, \dots, n\}$ 上の置換集合である. 二次割当問題については [2] に詳しい. 本稿では, 以下に述べる方法によって分枝限定法を構築した.

初期実行可能解の計算

Taillard によって提案された robust tabu search を用いた. また, QAPLIB のホームページで公開されているプログラムコードを参考にした.

評価計算 (下界値計算)

Gilmore-Lawler bound を用いた. すなわち, $i, j = 1, \dots, n$ に対して, L_{ij} を

$$L_{ij} \equiv \min_{\substack{\pi \in S_n \\ \pi(j)=i}} \sum_{k=1}^n F_{i\pi(k)} D_{jk}$$

と計算して求め, 行列 $L = (L_{ij})$ に関する (線形) 割当問題の最適値を求める. 良く知られているように, L_{ij} の計算において割当問題を解く必要はなく, Gilmore-Lawler bound の計算は高速に行なうことができる.

分枝方法

固定されていない添字 i を選び, 「 $\pi(i) = 1, \pi(i) = 2, \dots, \pi(i) = n$ 」または「 $\pi(1) = i, \pi(2) = i, \dots, \pi(n) = i$ 」と n 個の子問題を生成する. このとき, 行列 L に関する割当問題の最適解の情報や, 行列 F, D の情報を利用して無駄な分枝を見つけ, 削除することができる.

4.1 ベンチマーク問題に対する結果

QAPLIB ベンチマーク問題, nug22($n = 22$) および nug24($n = 24$) を PC クラスタ環境で解くことができた. 1997 年の論文 [1] では, nug22 を, NEC Cenju-3 のプロセッサを 48~96 台使用して (restart 機能があるため, 利用プロセッサ数を変更することができる), 766800(sec) で解いた. 本稿の PC クラスタ環境では

nug22 : 147378(sec)

nug24 : 1269218(sec)

という結果を得た. その他詳細については, 当日報告する.

5 おわりに

本稿で示したように, 特別に大規模でもない (PC 数 21 台の) PC クラスタ環境を用いても, 2 年前まで最新の並列計算機を用いなければ解けないとされていた規模の問題が解けるようになってきている. 今後は, より大規模な PC クラスタ環境の下で, より大規模な問題の解決に取り組んでいきたい.

参考文献

- [1] A. Brügger, A. Marzetta, J. Clausen and M. Perregaard, “Joining Forces in Solving Large-Scale Quadratic Assignment Problems in Parallel,” *Proc. of the 11th International Parallel Processing Symposium* (1997) 418-427.
- [2] E. Çela, *The Quadratic Assignment Problem*, Kluwer Academic Publishers (1998).
- [3] Y. Shinano, K. Harada and R. Hirabayashi, “Control Schemes in a Generalized Utility for Parallel Branch-and-Bound Algorithms,” *Proc. of the 11th International Parallel Processing Symposium* (1997) 621-627.
- [4] Y. Shinano, T. Fujie, Y. Ikebe and R. Hirabayashi, “Solving the Maximum Clique Problem Using PUBB,” *Proc. of the 12th International Parallel Processing Symposium* (1998) 326-332.