2003年日本オペレーションズ・リサーチ学会
春季研究発表会

半正定値計画問題を解くソフトウェアのPCクラスタ上における並列実装

02701900 東京工業大学 *山下 真 YAMASHITA Makoto
01507230 東京電機大学 藤沢 克樹 FUJISAWA Katsuki
01103520 東京工業大学 小島 政和 KOJIMA Masakazu

1 はじめに

半正定値計画問題(以下、SDP)は、線形計画問題の対称行列空間への拡張であり、システムの制御や組合せ最適化問題の緩和などさまざまな応用範囲を持つ数理計画問題のひとつである。また、線形計画問題に対しての強力な解法の主双対内点法は、半正定値計画問題にも適用することが可能であり、代表的なSDPA (SemiDefinite Programmming Algorithm) [1] を含め、多数のコンピュータソフトウェアが開発されている。

しかし、量子化学などから定式化されるSDPは、問題としての規模が大きく、計算時間、計算必要メモリの両面から見て1台の計算機で解くことが不可能である。その一方、近年のPCの価格の低下や、ネットワーク技術の発展にともない、並列計算を行う環境として、PCをネットワークで複数台つないで構成されたPCクラスタが注目を浴びはじめている。

本稿では、SDPAの並列実装であるSDPARA (SemiDefinite Programmming Algorithm PARAllel version) [2] の実装における概要を述べた後に、PCクラスタ上による数値実験によってSDPARAの高いスケーラビリティを示す。

2 SDPの標準形と主双対内点法

本稿で対象とするSDPは、次のような主問題$(\mathcal{P})$と双対問題$(\mathcal{D})$を合わせ持った標準形である。

$$\left\{\begin{array}{lll}(\mathcal{P}): & 最小化 & \sum_{i=1}^m c_i x_i \\ & 制約 & X = \sum_{i=1}^m F_i x_i - F_0, \\ & & X \succeq O,\ X \in \mathbb{S}^n. \\ (\mathcal{D}): & 最大化 & F_0 \circ Y \\ & 制約 & F_i \circ Y = c_i\ (i=1,2,\ldots,m) \\ & & Y \succeq O,\ Y \in \mathbb{S}^n.\end{array}\right.$$

ここで、$\mathbb{S}^n$はn次対称正方行列のなす空間であり、$X \succeq O (X \succ O)$は行列Xが半正定値(正定値)行列であることを示す。また、$U, V \in \mathbb{S}^n$に対して$U \circ V$は内積を表し、$U \circ V = \sum_{i=1}^n \sum_{j=1}^n U_{ij} V_{ij}$によって定義される。

SDPの最適解(x, X, Y)は、$(\mathcal{P})$,$(\mathcal{D})$の制約を満たし、双対ギャップ$X \circ Y$を0にすることができるため、次に示す主双対内点法によってSDPを解くことができる。

主双対内点法のアルゴリズム

Step 0 : $k := 0$, $X^0 \succ O$, $Y^0 \succ O$を満たす任意の(x^0, X^0, Y^0)を生成する。

Step 1 : (x^k, X^k, Y^k)が$(\mathcal{P})$,$(\mathcal{D})$の制約を満たし、双対ギャップ$X^k \circ Y^k$がある程度小さくなった時点で、アルゴリズムを終了し、(x^k, X^k, Y^k)を最適解として出力する。

Step 2 : 次の方程式系を線形近似して解くことにより、探索方向(dx, dX, dY)を計算する。

$$\left\{\begin{array}{l} X^k + dX = \sum_{i=1}^m F_i(x_i^k + dx_i) - F_0 \\ F_i \circ (Y^k + dY) = c_i\ (i=1,2,\ldots,m) \\ (X^k + dX)(Y^k + dY) = O. \end{array}\right.$$

Step 3 : $X^k + \alpha_p dX \succ O$, $Y^k + \alpha_d dY \succ O$となる$\alpha_p, \alpha_d$を求め、$(x^{k+1}, X^{k+1}, Y^{k+1}) = (x^k + \alpha_p dx, X^k + \alpha_p dX, Y^k + \alpha_d dY)$に更新する。$k := k+1$としてStep 1へ戻る。

Step 2における探索方向(dx, dX, dY)を求める計算は、Schur complement equationと呼ばれる連立方程式$Bdx = r$によってdxを求めることに帰着できる。ここで、係数行列Bは各要素が$B_{ij} = ((X^k)^{-1} F_i Y^k) \circ F_j$によって計算される正定値行列である。

表 1: PC の台数と SDPARA の実行時間 (秒) ただし、* はメモリ不足を示す

問題名 \ PC の台数	1	2	4	8	16	32	64
control11	685.3	363.1	195.0	112.1	66.6	42.9	31.8
theta6	600.6	341.7	168.2	112.8	68.4	51.3	38.3
quantum	*	*	*	3836.3	2076.5	1289.7	732.7

3 並列実装の概要

前節で示した主双対内点法を実装したSDPAにおいて、計算時間を考慮したときにボトルネックになるのは、係数行列 B の要素の計算と、連立方程式を解くための B の Cholesky 分解であり、この2つのボトルネックのために一般的に計算時間の 95 % 以上が使用されている。

SDPA の並列実装版である SDPARA では、この2つのボトルネックを解消するために、MPI と ScaLAPACK を用いて並列計算を行っている。係数行列 $\boldsymbol{B}$ の Cholesky 分解は ScaLAPACK のルーチンをほぼそのまま利用しているので、ここでは $\boldsymbol{B}$ の要素の計算をどのようにして並列に計算するかに注目をする。

複数の PC によって行われる並列計算の場合に重要なのは、それぞれの PC で行われる計算を独立させて PC 間のネットワークによる通信を減少させることと、1台の PC に計算が集中しないように計算を分散させることである。

$\boldsymbol{B}$ の各要素の計算 $B_{ij} = ((\boldsymbol{X}^k)^{-1}\boldsymbol{F}_i\boldsymbol{Y}^k) \bullet \boldsymbol{F}_j$ は互いに独立しているため、すべての PC が $\boldsymbol{X}^k, \boldsymbol{Y}^k, \boldsymbol{F}_i (i = 1, \ldots, m)$ の情報を持っていれば通信することなく各要素の計算が可能となる。しかし、$\boldsymbol{B}$ の i 行目はすべて同じ計算 $(\boldsymbol{X}^k)^{-1}\boldsymbol{F}_i\boldsymbol{Y}^k$ を行うので、複数の PC が同じ計算を行うよりも1台の PC のみの方が効率が良い。

そのため SDPARA では、 $\boldsymbol{B}$ を行ごとに分割し、一つの行に対して1台の PC を割り当てている。具体的にいえば、N 個の PC があったときに $\boldsymbol{B}$ の i 行目は、$i\%N$ 番目の PC が、$(\boldsymbol{X}^k)^{-1}\boldsymbol{F}_i\boldsymbol{Y}^k$ の計算を行い、その後の $\boldsymbol{F}_j (j = 1, \ldots, m)$ との内積計算までをまとめて行う。ここで、$a\%b$ は a を b で割った余りとする。この方法は1台の PC に計算が集中しないようにする効果も合わせ持つ。

SDPARA における並列計算の手法はシンプルであるが、次節の数値実験が示すように、その効果は非常に高いと言える。

4 PC クラスタ上における数値実験

表1は、PC クラスタ Presto III 上で PC の台数を変えながら、SDPARA を実行したときにかかる時間を表にまとめたものである。Presto III は、各 PC に CPU Athlon 1900+, Memory 768 MB を搭載し、Myrinet という高速ネットワークで PC 間をつないだ非常に高性能な PC クラスタであり、東京工業大学 数理・計算科学専攻 松岡研究室が設置、および管理をしている。また、control11, theta6, quantum は、それぞれ制御、組合せ最適化、量子化学において発生している SDP である。

control11 では、8台の PC を使ったときには1台の PC の 6.11 倍、64 台のときには 21.55 倍の速さで解くことができ、quantum では、64 台のときに 8 台の 5.23 倍の速度を達成することができている。このように、SDPARA では、前節でのべたようにシンプルな並列計算の手法を採用しているが、PC の台数が増えたときに得られるスケーラビリティはとても高い。

また、quauntum においては、SDP としての規模が大きいために1台の PC で解くことは使用メモリ量から見て不可能であり、PC クラスタを用いた並列計算をすることによって解を得ることに成功しているということができる。

参考文献

[1] SDPA 6.0 ホームページ
http://www.is.titech.ac.jp/~kojima/sdpa/

[2] SDPARA Research Report
http://www.is.titech.ac.jp/research/research-report/B/B-384.ps.gz