

TR-H-105

2次計画問題に対する主双対内点法とその数値実験

山川 栄樹
松原 康博 (奈良先端科技院大)

1994. 10. 24

ATR 人間情報通信研究所

〒619-02 京都府相楽郡精華町光台 2-2 ☎07749-5-1011

ATR Human Information Processing Research Laboratories

2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan

Telephone: +81-7749-5-1011

Facsimile: +81-7749-5-1008

2次計画問題に対する主双対内点法とその数値実験

山川 栄樹

株式会社 エイ・ティ・アール人間情報通信研究所 第六研究室

松原 康博

奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 システム基礎講座

1994年10月24日

1 Introduction

2次計画問題はポートフォリオ選択問題などの重要な応用分野を持つだけでなく、一般の非線形問題の反復解法における部分問題としてもしばしば用いられている。2次計画問題に対しては、Lemke法 [4] や Goldfarb - Idnani の方法 [3] など、これまでに数多くの解法が提案されている。ここでは、大規模な問題に対する応用を想定して、Path Following タイプの内点法的一种である Primal-Dual Interior Point Method [2] を適用することを考える。この方法は、元の2次計画問題とその双対問題の双方について、それらに含まれる不等式制約条件を古典的な Logarithmic Barrier ペナルティ関数を利用して目的関数に組み込み、等式制約条件のみを含む問題に変換した上で、それらの最適性の1次の条件の方程式系を Newton 法を用いて解くというものである。本稿では、その具体的な計算法について検討を加えると共に、並列計算機 Connection Machine Model CM-5 上への実現について述べ、数値実験の結果を報告する。

2 Logarithmic Barrier Method

次のような $x \in R^n$ に関する2次計画問題を考える。

$$\begin{aligned} & \text{minimize} && c^T x + \frac{1}{2} x^T G x, \\ & \text{subject to} && Ax = b, \\ & && x \geq 0. \end{aligned} \tag{1}$$

ここで、 b, c はそれぞれ R^m, R^n のベクトル、 G は $n \times n$ 半正定値対称行列、 A は $m \times n$ 行列で $\text{Rank}(A) = m$ とする。制約条件 $Ax = b, x \geq 0$ に対する Lagrange 乗数を、それぞれ $y \in R^m, 0 \leq z \in R^n$ と書くことにする。この時、問題 (1) に対する Lagrange の双対問題は、

$$\begin{aligned} & \text{maximize} && \inf_x L(x, y, z) \equiv c^T x + \frac{1}{2} x^T G x - y^T (Ax - b) - z^T x, \\ & \text{subject to} && z \geq 0 \end{aligned}$$

と書けるが、目的関数に含まれる \inf_x を実現する x は

$$\nabla_x L(x, y, z) = c + Gx - A^T y - z = 0$$

を満たすことに注意すると、次のように書直すことができる。

$$\begin{aligned} & \text{maximize} && b^T y - \frac{1}{2} x^T G x, \\ & \text{subject to} && A^T y + z - G x = c, \\ & && z \geq 0. \end{aligned} \tag{2}$$

原問題 (1) の非負制約を logarithmic barrier ペナルティ関数の考え方をを用いて目的関数に組込むと、次のような問題が得られる。

$$\begin{aligned} & \text{minimize} && c^T x + \frac{1}{2} x^T G x - \mu \sum_{i=1}^n \log x_i, \\ & \text{subject to} && A x = b. \end{aligned} \tag{3}$$

ここで、 μ は正のパラメータである。 μ の値を十分小さな正の数にとると、問題 (3) は原問題 (1) の良い近似を与えることに注意しよう。問題 (3) は、空間 $\{x \mid x > 0\}$ 上で定義可能な問題であるが、線形等式制約のみを含む凸計画問題となるため、不等式制約をも含む原問題 (1) に比べて幾分取扱いが容易になる。このような手法は、一般に Logarithmic Barrier Method と呼ばれている。同様にして問題 (2) の非負制約を log-barrier 化すると、

$$\begin{aligned} & \text{maximize} && b^T y - \frac{1}{2} x^T G x + \mu \sum_{j=1}^m \log z_j, \\ & \text{subject to} && A^T y + z - G x = c \end{aligned} \tag{4}$$

を得る。問題 (4) は、空間 $\{z \mid z > 0\}$ 上で定義可能な問題である。以下では、問題 (3) 及び問題 (4) を解くことによって原問題 (1) とその双対問題 (2) の解を得ることを考える。問題 (3) の Lagrange 関数は

$$L_p(x, y) = c^T x + \frac{1}{2} x^T G x - \mu \sum_{i=1}^n \log x_i - y^T (A x - b)$$

となり、問題 (4) の Lagrange 関数は

$$L_d(y, z, x) = b^T y - \frac{1}{2} x^T G x + \mu \sum_{j=1}^m \log z_j - x^T (A^T y + z - G x - c)$$

と書けることに注意すると、問題 (3) に対する K.K.T. (Karush-Kuhn-Tucker) 条件は

$$\begin{cases} \nabla_x L_p(x, y) = c + G x - \mu X^{-1} e_n - A^T y = 0, \\ \nabla_y L_p(x, y) = -A x + b = 0 \end{cases} \tag{5}$$

であり、問題 (4) に対する K.K.T. 条件は

$$\begin{cases} \nabla_y L_d(y, z, x) = b - A x = 0, \\ \nabla_z L_d(y, z, x) = \mu Z^{-1} e_m - x = 0, \\ \nabla_x L_d(y, z, x) = -G x - A^T y - z + 2G x + c = 0 \end{cases} \tag{6}$$

と記述される。ここで、 X, Y, Z はそれぞれベクトル x, y, z の各要素を対角成分とする対角行列であり、 $e_m \in R^m, e_n \in R^n$ はそれぞれすべての要素が 1 のベクトルである。結局 (5), (6) より、問題 (3), (4) の解を求めるには、 $x > 0, z > 0$ を満たす方程式系

$$\begin{cases} A x = b, \\ A^T y + z - G x = c, \\ Z x = \mu e_n \end{cases} \tag{7}$$

の解 (x, y, z) を求めればよいことがわかる。さらに $\mu \downarrow 0$ とすることにより、方程式系 (7) の解 (x, y, z) は原問題 (1) とその双対問題 (2) の解を表す点 (x^*, y^*, z^*) に近づいて行く。そこで先ず、方程式系 (7) を反復解法を用いて解くことを考えよう。

3 Basic Iteration of Primal-Dual Interior Point Method

方程式系 (7) に対して Newton 法に基づく反復解法を適用することを考える。以下では、 $x > 0, z > 0$ で、問題 (3), (4) の制約条件

$$\begin{aligned} Ax &= b, \\ A^T y + z - Gx &= c \end{aligned}$$

を満たす点 (x, y, z) を、実行可能内点と呼ぶことにする。ある実行可能内点 (x, y, z) が与えられている時、Newton 法の探索方向 $(\Delta x, \Delta y, \Delta z)$ は、方程式系 (7) を点 (x, y, z) で一次近似することによって得られる連立方程式

$$\begin{cases} A\Delta x = 0, \\ A^T\Delta y + \Delta z - G\Delta x = 0, \\ Z\Delta x + X\Delta z = \mu e_n - Zx \end{cases} \quad (8)$$

を解くことによって求められる。連立方程式 (8) を Δy について解いてみよう。先ず (8) の第 3 式より

$$\begin{aligned} \Delta z &= X^{-1}(\mu e_n - Zx - Z\Delta x) \\ &= X^{-1}(\mu e_n - Z\Delta x) - z \end{aligned} \quad (9)$$

を得る。これを (8) の第 2 式に代入すると、

$$A^T\Delta y + X^{-1}(\mu e_n - Z\Delta x) - z - G\Delta x = 0$$

となるが、整理すると

$$\begin{aligned} (G + X^{-1}Z)\Delta x &= A^T\Delta y + \mu X^{-1}e_n - z \\ \Delta x &= (G + X^{-1}Z)^{-1}(A^T\Delta y + \mu X^{-1}e_n - z) \end{aligned} \quad (10)$$

となる。これを (8) の第 1 式に代入して整理すると

$$\begin{aligned} A(G + X^{-1}Z)^{-1}(A^T\Delta y + \mu X^{-1}e_n - z) &= 0 \\ A(G + X^{-1}Z)^{-1}A^T\Delta y &= A(G + X^{-1}Z)^{-1}(z - \mu X^{-1}e_n) \end{aligned} \quad (11)$$

を得る。行列 G は半正定値対称で $x > 0, z > 0$, さらに行列 A は row full rank と仮定したことに注意すると、方程式 (11) の左辺の係数行列 $A(G + X^{-1}Z)^{-1}A^T$ は正定値対称となる。従って、方程式 (11) は唯一の解を持つ。結局、方程式 (11) を解いて Δy を求め、これを (10) に代入して Δx を、さらにその結果を (9) に代入して Δz を求めればよいことがわかる。

4 Line Search

大域的収束性を保証するために、何らかの方法でステップサイズ θ を求め、次の探索点を

$$\begin{aligned} x^+ &:= x + \theta\Delta x, \\ y^+ &:= y + \theta\Delta y, \\ z^+ &:= z + \theta\Delta z \end{aligned}$$

とすることを考える。但し、問題 (3), (4) の目的関数の定義域を考慮すると、ステップサイズ θ を

$$\eta = \max \{ \nu \mid x + \nu\Delta x \geq 0 \text{ and } z + \nu\Delta z \geq 0 \}$$

以上とすることはできない。実際 $\theta \in [0, \eta)$ ととれば、 (x^+, y^+, z^+) も実行可能内点となることがわかる。ここでは、問題 (3) の目的関数を

$$f_\mu(x) \equiv c^T x + \frac{1}{2} x^T G x - \mu \sum_{i=1}^n \log x_i$$

とおき、 f_μ を直線探索の際の評価関数としてステップサイズ θ を決定することにする。直線探索として Armijo の規則を用いることにすると、ステップサイズ θ は、

$$\theta = \rho(\gamma)^h, \quad \rho = \min\{\tau\eta, 1\}$$

により求められる。ここで、 $\gamma \in (0, 1)$, $\tau \in (0, 1)$ であり、 $\sigma \in (0, \frac{1}{2})$ とした時、 h は不等式

$$f_\mu(x + \rho(\gamma)^h \Delta x) \leq f_\mu(x) + \sigma \rho(\gamma)^h \nabla f_\mu(x)^T \Delta x$$

を満たす最小の非負整数である。なお、 f_μ の定義より

$$\nabla f_\mu(x) = c + Gx - \mu X^{-1} e_n$$

である。また、 τ, γ, σ の具体的な値としては、

$$\tau = 0.995, \quad \gamma = 0.5, \quad \sigma = 0.1$$

を用いることにする。

5 Termination Criterion

次に、停止条件について考える。相補条件の評価関数として

$$\phi_\mu(x, z) \equiv \frac{\|Zx - \mu e_n\|_1}{n}$$

を考える。ここで $\|\cdot\|_1$ は L_1 ノルムである。この時、十分小さな正の数 ε_μ に対して、

$$\phi_\mu(x, z) \leq \varepsilon_\mu \tag{12}$$

が成立すれば Newton 法の反復を終了し、得られた点 (x, y, z) を方程式 (7) 即ち問題 (3) 及び (4) の解と見なすことにする。Path following タイプの内点法では、本来ある μ に対して方程式 (7) を解いた後、その結果に基づいて μ をより小さな正の値に更新し、新たな μ の値に対して再び連立方程式 (7) を解くという手続きを繰返すことにより、問題 (1) 及び (2) の解を表す点 (x^*, y^*, z^*) に近づいて行く必要がある。しかしここでは、 $0 < \varepsilon_\mu \ll M_{tol} < M_\mu$ なるパラメータ M_{tol}, M_μ を用いて、

$$\phi_\mu(x, z) < M_{tol} \mu$$

が成立した段階で、方程式 (7) の近似解は得られたとして μ の値を

$$\mu := \frac{\phi_\mu(x, z)}{M_\mu}$$

により更新し、停止条件 (12) が成立つまで反復を継続することにする [6]。なお、 $\varepsilon_\mu, M_{tol}, M_\mu$ の具体的な値としては、

$$\varepsilon_\mu = 10^{-6}, \quad M_{tol} = 300, \quad M_\mu = 500$$

を用いる。

6 Algorithm

これまでの議論をまとめると、問題 (1) に対する主双対内点法のアルゴリズムは次のようになる。

アルゴリズム：主双対内点法

ステップ1 実行可能内点 (x^1, y^1, z^1) と十分大きな正の数 μ_1 を選ぶ。パラメータ $0 < \varepsilon_\mu \ll M_{tol} < M_\mu$, $\tau \in (0, 1)$, $\gamma \in (0, 1)$, $\sigma \in (0, \frac{1}{2})$ をセットして $k := 1$ とおく。

ステップ2 連立方程式

$$\begin{cases} A\Delta x = 0, \\ A^T\Delta y + \Delta z - G\Delta x = 0, \\ Z^k\Delta x + X^k\Delta z = \mu_k e_n - Z^k x^k \end{cases}$$

の解 $(\Delta x^k, \Delta y^k, \Delta z^k)$ を求める。

ステップ3 ステップサイズを

$$\begin{aligned} \eta_k &:= \max \{ \nu \mid x + \nu\Delta x \geq 0 \text{ and } z + \nu\Delta z \geq 0 \}, \\ \rho_k &:= \min \{ \tau\eta_k, 1 \}, \\ \theta_k &:= \rho_k (\gamma)^h, \end{aligned}$$

により求める。ここに、 h は

$$f_\mu(x^k + \rho_k (\gamma)^h \Delta x^k) \leq f_\mu(x^k) + \sigma \rho_k (\gamma)^h \nabla f_\mu(x^k)^T \Delta x^k$$

を満たす最小の非負整数であり、

$$f_\mu(x) = c^T x + \frac{1}{2} x^T G x - \mu \sum_{i=1}^n \log x_i$$

である。探索点を次のように更新する。

$$\begin{aligned} x^{k+1} &:= x^k + \theta_k \Delta x^k, \\ y^{k+1} &:= y^k + \theta_k \Delta y^k, \\ z^{k+1} &:= z^k + \theta_k \Delta z^k. \end{aligned}$$

ステップ4 停止基準

$$\phi_k(x^{k+1}, z^{k+1}) \equiv \frac{\|Z^{k+1}x^{k+1} - \mu_k e_n\|_1}{n} \leq \varepsilon_\mu$$

が成立てば、終了する。

ステップ5 $\phi_k(x^{k+1}, z^{k+1}) \leq M_{tol} \mu_k$ ならば

$$\mu_{k+1} := \frac{\phi_k(x^{k+1}, z^{k+1})}{M_\mu},$$

さもなければ $\mu_{k+1} := \mu_k$ とする。 $k := k + 1$ としてステップ2へ戻る。

なお、反復が進むにつれて、探索点は解に近付いて来るので、反復の最終段階においては直線探索の手続きを省略することができるものと考えられる。そこで、ある反復 k のステップ 4 において

$$\phi_k(x^{k+1}, z^{k+1}) \leq \varepsilon_{line}$$

が成立ったならば、以後の反復のステップ 3 では $\theta_k := \rho_k$ とおくことにする。但し、 ε_{line} の具体的な値としては、 $\varepsilon_{line} = 10^{-3}$ を用いることにする。

7 Initial Feasibility

前章のアルゴリズムは、初期実行可能内点 (x^1, y^1, z^1) が予め得られていることを前提としているが、一般にそのような点を求めることは容易でない。そこで、人為変数を導入することにより、初期実行可能内点が容易に得られるような問題を構成することを考える。原問題 (1) 及びその Lagrange の双対問題 (2) に対応して、それぞれ問題

$$\begin{aligned} & \text{minimize} && c^T x + \frac{1}{2} x^T G x + c_a x_a, \\ & \text{subject to} && Ax + d_p x_a = b, \\ & && d_d^T x + x_b = b_a, \\ & && x \geq 0, \quad x_a \geq 0, \quad x_b \geq 0 \end{aligned} \tag{13}$$

及び

$$\begin{aligned} & \text{maximize} && b^T y - \frac{1}{2} x^T G x - b_a y_a, \\ & \text{subject to} && A^T y - d_d y_a + z - G x = c, \\ & && d_p^T y + z_a = c_a, \\ & && -y_a + z_b = 0, \\ & && z \geq 0, \quad z_a \geq 0, \quad z_b \geq 0 \end{aligned} \tag{14}$$

を考える。ここで d_p, d_d はそれぞれ、初期解を (x^1, y^1, z^1) とした時の問題 (1), (2) の等式制約条件の残差

$$d_p = b - Ax^1, \quad d_d = A^T y^1 + z^1 - Gx^1 - c$$

である。問題 (13) 及び (14) は、次に示すような置換えにより、それぞれ問題 (1) 及び (2) と同じように取扱うことができる。即ち、

$$\hat{c} = \begin{pmatrix} c \\ c_a \\ 0 \end{pmatrix}, \quad \hat{G} = \begin{pmatrix} G & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \hat{x} = \begin{pmatrix} x \\ x_a \\ x_b \end{pmatrix}, \quad \hat{z} = \begin{pmatrix} z \\ z_a \\ z_b \end{pmatrix},$$

$$\hat{A} = \begin{pmatrix} A & d_p & 0 \\ -d_d^T & 0 & -1 \end{pmatrix}, \quad \hat{b} = \begin{pmatrix} b \\ -b_a \end{pmatrix}, \quad \hat{y} = \begin{pmatrix} y \\ y_a \end{pmatrix}$$

とする時、問題 (13), (14) は、それぞれ

$$\begin{aligned} & \text{minimize} && \hat{c}^T \hat{x} + \frac{1}{2} \hat{x}^T \hat{G} \hat{x}, \\ & \text{subject to} && \hat{A} \hat{x} = \hat{b}, \\ & && \hat{x} \geq 0 \end{aligned}$$

及び

$$\begin{aligned} & \text{maximize} && \hat{b}^T \hat{y} - \frac{1}{2} \hat{x}^T \hat{G} \hat{x}, \\ & \text{subject to} && \hat{A}^T \hat{y} + \hat{z} - \hat{G} \hat{x} = \hat{c}, \\ & && \hat{z} \geq 0 \end{aligned}$$

と書直すことができる。容易に確かめられるように、行列 G が半正定値対称ならば行列 \hat{G} も半正定値対称であり、行列 A が row full rank ならば行列 \hat{A} も row full rank である。しかも、 b_a , c_a をそれぞれ

$$\begin{aligned} x_b &= b_a - d_d^T x > 0, \\ z_a &= c_a - d_p^T y > 0 \end{aligned}$$

となるような十分大きな数にとれば、任意の $(x^1, y^1, z^1) > 0$ に対して

$$\begin{aligned} x_a^1 &:= 1, \\ x_b^1 &:= b_a - d_d^T x^1, \\ y_a^1 &:= 1, \\ z_a^1 &:= c_a - d_p^T y^1, \\ z_b^1 &:= 1 \end{aligned}$$

ととることによって、 $(\hat{x}^1, \hat{y}^1, \hat{z}^1)$ は問題 (13), (14) に関する実行可能内点となる。さらに、最適解において x_a, y_a はいずれも 0 となり、問題 (1), (2) の解を得ることができる。

8 Preconditioned Conjugate Gradient Method for Solving Linear Equations

主双対内点法のアルゴリズムの計算量の大半は、ステップ 2 において方程式 (11) を解くために費やされる。現実の大規模問題では、行列 G や A が疎である場合が多い。ここでは、共役勾配法を用いて方程式 (11) を解くことを考える。共役勾配法は、原問題の係数行列や定数ベクトルに対する計算のみでアルゴリズムを構成できるため、それらの値を保持するデータ構造を工夫することにより、原問題の疎な構造を維持したまま効率的に計算を実行することができる。今

$$\begin{aligned} M &= A(G + X^{-1}Z)^{-1}A^T, \\ q &= A(G + X^{-1}Z)^{-1}(z - \mu X^{-1}e_n), \\ u &= \Delta y \end{aligned}$$

とおくと、方程式 (11) は

$$Mu = q$$

と書くことができる。既に指摘したように、係数行列 M は正定値対称である。

共役勾配法の反復は、理論的には M の相異なる固有値の数と等しい回数で停止することが知られている。従って、正定値対称行列 M の逆行列の近似行列を何らかの方法で生成して方程式の係数行列の固有値を「固める」ことにより、共役勾配法の反復回数を削減することが考えられる。ここでは、 M の対角成分のみを抽出した行列

$$D \equiv \text{diag} \left\{ a_i^T (G + X^{-1}Z)^{-1} a_i \right\}_{i \in \{1, \dots, m\}}$$

の逆行列 D^{-1} により M の逆行列を近似する、いわゆる Diagonal Scaling の手法を用いることにする。この手法は、連立方程式の係数行列の対角要素をすべて 1 に揃えるような前処理である。Diagonal Scaling を組込んだ共役勾配法の手続きは、次のようになる。

手続き：前処理付共役勾配法

ステップ 1 初期探索点 $u^1 \in R^m$ を選んで、残差ベクトル

$$r^1 := q - Mu^1$$

を求める。許容誤差 $\varepsilon > 0$ をセットして、 $l := 1$ とおく。

ステップ 2 探索方向 p^l を前処理を施した最急降下方向にセットする。即ち、

$$p^l := D^{-1}r^l.$$

ステップ 3 正確な直線探索に基づき、ステップサイズ α_l を計算する。即ち、

$$\begin{aligned} t^l &:= Mp^l, \\ \zeta_l &:= (p^l)^T t^l, \\ \xi_l &:= (p^l)^T r^l, \\ \alpha_l &:= \xi_l / \zeta_l. \end{aligned}$$

ステップ 4 探索点および残差ベクトルの更新を行う。即ち、

$$\begin{aligned} u^{\ell+1} &:= u^\ell + \alpha_\ell p^\ell, \\ r^{\ell+1} &:= r^\ell - \alpha_\ell t^\ell. \end{aligned}$$

ステップ 5 停止基準 $\|r^{\ell+1}\|_\infty \leq \varepsilon \|q\|_\infty$ を満たせば終了する。

ステップ 5 $l = 0 \pmod{m}$ ならば、 $l := l + 1$ としてステップ 2 へ戻る（再出発）。さもなければ、

$$\beta_l := \frac{(r^{\ell+1})^T D^{-1} r^{\ell+1}}{\xi_l},$$

$$p^{\ell+1} := D^{-1} r^{\ell+1} + \beta_l p^\ell$$

により共役方向の生成を行い、 $l := l + 1$ としてステップ 3 へ戻る。

ところで、主双対内点法のアルゴリズムの初期反復において、探索点 (x^k, y^k, z^k) は解から遠く離れていると考えられる。このような場合には、方程式 (11) を必ずしも厳密に解く必要はない。よって実際の計算においては、前処理付共役勾配法の手続きの許容誤差 ε を、主双対内点法のアルゴリズムの反復回数 k を用いて

$$\varepsilon := \max \{ \varepsilon_0, 10^{-k-1} \}$$

のように緩和することにする [1]。なお、 ε_0 の具体的な値としては、 $\varepsilon_0 = 10^{-6}$ を用いることにする。

9 Numerical Results

ここでは、本稿で述べたアルゴリズムを並列計算機 Connection Machine Model CM-5 上へ実現する方法と、幾つかの数値実験結果について述べる。実験に用いた CM-5 は 32 個のプロセッシング・ノードから構成され、各プロセッシング・ノードは浮動小数点演算用のベクトル・ユニットを 4 個ずつ搭載している。各プロセッシング・ノードのピーク性能は 128 MFLOPS であり、超高速ワークステーション並の能力を持っている。アルゴリズムは、data parallel 型のプログラミング言語である CM Fortran によりコーディングを行った。CM Fortran は、大規模な配列に対する演算に適するように特化された FORTRAN 系の言語であり、対応する配列要素同士の演算を並列的に実行したり、配列要素の移動や合計値の計算を高速に実現する機能を備えている。なお、実験はすべて倍精度計算により行った。

数値実験は、ランダムに生成された疎な 2 次計画問題 (1) に対して、本稿で述べたアルゴリズムを適用する形で行った。ただし、方程式 (11) の係数に含まれる $G + X^{-1}Z$ の逆行列の計算が容易に行なえるようにするために、行列 G は対角行列とする。また、人為変数を導入することなく初期実行可能内点が得られるようにするために、次のような方法で各係数行列及びベクトルを生成した。まず行列 A は、各行が一次独立となり、各列に少なくとも一つの非零要素が存在するように n 個の非零要素の位置を決めた上で、予め指定した数になるまで非零要素の行番号と列番号とを重複が生じないようにランダムに生成する。行列 A の非零要素の値は、区間 $[-5,5]$ の一様乱数として生成する。一方、行列 G の対角要素は、区間 $[1,10]$ の一様乱数として発生させる。次に、初期実行可能内点 (x^1, y^1, z^1) を構成する各変数の要素を、何れも区間 $[1,100]$ の一様乱数で生成する。最後に、問題 (3), (4) の等式制約条件に基づいて、ベクトル b, c の値をそれぞれ

$$b = Ax^1, \quad c = A^T y^1 + z^1 - Gx^1$$

により決定する。

様々な m と n の組合せについてアルゴリズムの性能を確かめるために、4 種類のテスト問題を生成した。第一、第二のグループは、行列 A の非零要素数をそれぞれ 16384, 65536 に固定した問題であり、 m と n の値及びその比を様々に変化させながら実験を行なった。第三、第四のグループは、行列 A の非零要素の密度をそれぞれ 1.5625%, 0.390625% に固定し、 m と n の比はいずれも 1:4 に固定した上で様々なサイズの問題を生成している。各テスト問題の特徴を、表 1 に掲げる。

Connection Machine システム上にアルゴリズムを実現する上で問題となるのは、行列 A 及びその転置行列 A^T とベクトルとの積を求める計算である。ここでは、この計算を行うプログラムを 4 通りの方法で実現し、その比較実験を行った。行列とベクトルとの積を data parallel 型のプログラミング環境で実現する方法としては、segmented scan operations がよく知られている [7]。最初の 3 つのプログラムは、いずれも segmented scan operations に基づくものである。第一の方法では、CM Fortran のユーティリティ・ライブラリーの手続きである CMF_SCAN_ADD と CMF_SCAN_COPY とを用いて、segmented scan operations を実現している。一方、第二の方法は、CM Fortran 固有の機能である FORALL 文を用いて segmented scan operations を行っている。FORALL 文は配列要素への代入を可能な限り並列的に実行する処理であり、配列要素の合計計算や配列要素の部分的な抽出を行う操作と組合せることにより、ユーティリティ・ライブラリーの手続き CMF_SCAN_ADD 及び CMF_SCAN_COPY と同様な処理をこれらと同等またはそれ以上の性能で実現することが可能となっている。なお、これら 2 通りの方法においては、すべての変数の値を行列 A の非零要素数と同じ長さの 1 次元配列に格納している。メモリの利用という観点から見ると冗長なデータの持ち方であるが、対応する配列要素同士の演算は並列的に実行できるため、並列化効率の向上を図ることができる。また、その格納位置を行列 A の添字と対応づけ、

且つ、同じ添字を持つデータの格納位置を固めることにより、segmented scan operation を効率的に実行することが可能となる。但し、上で述べた 2 通りの掛け算を実行するためには、必要に応じて配列要素の順序を並べ換える必要がある。実際のプログラムにおいては、CM Fortran の Vector Valued Subscript と呼ばれる機能を用いて、この並べ換えを行っている。

第三の方法と第四の方法は、いずれも CMSSL (Connection Machine Scientific Software Library) の Sparse Matrix Operations ルーチンを用いて、行列とベクトルとの 2 通りの掛け算を行っている。CMSSL のルーチンを用いる場合には、各ベクトルの値を実際のサイズと同じ長さの 1 次元配列に格納することにより処理を進めることができ、配列要素の並べ換えを行なう必要はない。CMSSL のルーチンを使用する際には、内部処理の方式を選択することができる。第三の方法は segmented scan operations に基づく内部処理を指定した場合に対応し、第四の方法は gather-scatter operations に基づく内部処理を指定した場合に対応する。Gather-scatter operations に基づく CMSSL のルーチンは、setup と呼ばれる前処理により多くの時間を要する代わりに、segmented scan operations に基づくルーチンと比較してより短い時間で積の計算を実行できると言われている。従って、同じ構造を持つ行列とベクトルとの積を多数回行なう必要がある場合には、有利になるものと期待される。

4 通りの方法で実験した結果を、表 2 に掲げる。表からわかるように、アルゴリズム：主双対内点法の反復回数は、問題のサイズに関係なく十数回前後で一定している。一方、ステップ 2 において方程式 (11) を解くために用いている手続き：前処理付共役勾配法の反復回数は、原問題 (1) の制約条件の数 m が変数の数 n に近づくにつれて次第に増加するものの、方程式 (11) の次元数 m より十分小さな値にとどまっておき、前処理付共役勾配法が有効に動作しているものと考えられる。

一方、行列とベクトルとの掛け算に関する 4 通りの実現方法を比較してみると、utility library procedures による場合、FORALL statements を用いた場合、CMSSL の gather-scatter operations に基づくルーチンを用いた場合の順に計算時間は短くなり、よりよい結果を与えることがわかる。特に、手続き：前処理付共役勾配法の反復回数が多い場合において、第四の方法が他の方法に比べて非常に良い結果を示している。なお、segmented scan operations に基づく CMSSL のルーチンを用いた場合の性能は、第一の方法と第二の方法との中間的位置にあると考えられる。

10 Further Research

今回用いた主双対内点法および前処理付共役勾配法そのものは逐次的アルゴリズムであり、並列計算機を用いることによるメリットは、疎で大規模な行列やベクトルに対する演算の中に、並列的に実行できるものがあるという部分のみに依存している。しかし、多品種の輸送問題等、原問題 (1) の制約条件の係数行列が特徴的な構造を持つ場合には、方程式 (11) に対して並列アルゴリズムを適用することができるものと期待される。Yamakawa and Fukushima [5] は分離可能な 2 次計画問題に対する Block-Parallel Conjugate Gradient Method を提案しており、方程式 (11) に対する前処理付共役勾配法の手続きをこの考え方にに基づき並列化することが考えられる。

参考文献

- [1] M.G. Breitfeld and D.F. Shanno, "Computational Experience with Penalty-Barrier Methods for Nonlinear Programming," *Rutcor Research Report 17-93*, 1994.

- [2] T.J. Carpenter, I.J. Lustig, J.M. Mulvey and D.F. Shanno, "Separable Quadratic Programming via a Primal-Dual Interior Point Method and Its Use in a Sequential Procedure," *ORSA Journal on Computing*, Vol. 5, pp. 182-191, 1993.
- [3] D. Goldfarb and A. Idnani, "A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs," *Mathematical Programming*, Vol. 27, pp. 1-33, 1983.
- [4] C.E. Lemke, "Bimatrix Equilibrium Points and Mathematical Programming," *Management Science*, Vol. 11, pp. 681-689, 1965.
- [5] E. Yamakawa and M. Fukushima, "A Block-Parallel Conjugate Gradient Method for Quadratic Programming Problems," *Abstracts of the Third Conference of the Association of Asian-Pacific Operational Research Societies within IFORS*, W-3-B-2, p. 165, 1994.
- [6] H. Yamashita, "A Globally Convergent Primal-Dual Interior Point Method for Constrained Optimization," in *Optimization - Modeling and Algorithms - 3*, The Institute of Statistical Mathematics, Tokyo, pp. 272-297, 1993.
- [7] S.A. Zenios, "Data Parallel Computing for Network-Structured Optimization Problems," *Computational Optimization and Applications*, Vol. 3, pp. 199-242, 1994.

表 1: 分離可能な凸 2 次計画テスト問題とその特徴

Problem name	Problem size (n, m)	Density of matrix A (%)
QP11	(1024, 128)	12.5
QP12	(1024, 256)	6.25
QP13	(1024, 512)	3.125
QP14	(2048, 256)	3.125
QP15	(2048, 512)	1.5625
QP16	(2048, 1024)	0.78125
QP17	(4096, 512)	0.78125
QP18	(4096, 1024)	0.390625
QP19	(4096, 2048)	0.1953125
QP21	(2048, 256)	12.5
QP22	(2048, 512)	6.25
QP23	(2048, 1024)	3.125
QP24	(4096, 512)	3.125
QP25	(4096, 1024)	1.5625
QP26	(4096, 2048)	0.78125
QP27	(8192, 1024)	0.78125
QP28	(8192, 2048)	0.390625
QP29	(8192, 4096)	0.1953125
QP31 ¹	(2048, 512)	1.5625
QP32 ²	(4096, 1024)	1.5625
QP33	(8192, 2048)	1.5625
QP34	(16384, 4096)	1.5625
QP41 ³	(4096, 1024)	0.390625
QP42 ⁴	(8192, 2048)	0.390625
QP43	(16384, 4096)	0.390625
QP44	(32768, 8192)	0.390625

¹ QP31 is the same problem as QP15.

² QP32 is the same problem as QP25.

³ QP41 is the same problem as QP18.

⁴ QP42 is the same problem as QP28.

表 2: 分離可能な凸 2 次計画問題に対する主双対内点法の CM-5 上での 4 通りの実現方法についての比較

Problem	The number of Major Iterations	Total number of Inner Iterations	CPU(sec)			
			Utility library procedures	FORALL statements	CMSSL	
					Scan	Gather-scatter
QP11	12.6	182.8	3.85	3.55	3.69	2.68
QP12	12.4	275.0	5.47	4.92	5.16	3.52
QP13	12.6	577.0	10.85	9.65	10.29	6.35
QP14	12.6	186.0	3.92	3.50	3.72	2.65
QP15	12.2	280.8	5.58	4.90	5.30	3.48
QP16	13.0	614.0	11.59	10.11	10.88	6.74
QP17	13.8	206.0	4.34	3.78	4.09	2.84
QP18	13.6	324.6	6.45	5.61	6.02	3.97
QP19	13.4	717.4	13.46	11.69	12.67	8.07
QP21	12.4	186.2	10.11	9.49	9.84	7.17
QP22	12.6	294.8	15.05	13.95	14.60	9.20
QP23	13.2	633.8	31.06	28.07	29.12	16.33
QP24	13.0	196.8	10.72	9.85	10.26	6.91
QP25	13.2	311.2	16.06	14.55	15.07	9.33
QP26	13.6	661.8	32.41	29.15	30.21	17.42
QP27	13.0	197.4	10.73	9.75	10.11	6.76
QP28	13.4	327.6	16.82	15.15	15.82	9.87
QP29	14.8	741.4	36.62	32.45	34.11	20.24
QP31	12.2	280.8	5.58	4.90	5.30	3.48
QP32	13.2	311.2	16.06	14.55	15.07	9.33
QP33	13.6	320.2	58.19	52.54	55.39	33.44
QP34	14.0	334.2	274.40	253.02	249.72	137.86
QP41	13.6	324.6	6.45	5.61	6.02	3.97
QP42	13.4	327.6	16.82	15.15	15.82	9.87
QP43	13.6	331.2	60.18	53.77	56.56	33.99
QP44	14.2	341.4	281.56	258.61	248.13	147.48

¹ 表の各欄の数値は、各サイズのテスト問題の 5 つの問題例に対する結果の平均である。