

信頼領域を用いた大規模非線形計画問題 に対する並列部分空間法

京都大学・情報学研究科 高須啓介 (Keisuke Takasu)
福島雅夫 (Masao Fukushima)
Graduate School of Informatics
Kyoto University

1 はじめに

本稿では以下の制約なし最小化問題 (1) を考える.

$$\min_{x \in R^n} f(x) \quad (1)$$

ここで関数 $f: R^n \rightarrow R$ は 2 回連続的微分可能であり, 下に有界であるとする. 制約なし最小化問題を解く手法は, 数多く提案されているが, 通常, それらのアルゴリズムでは各反復で目的関数を線形もしくは二次のモデルに近似した部分問題を解き, 次の反復点を求める. 通常, 各反復における部分問題は元の問題と同じ次元の変数で定義される. そのため, 大規模な問題については各反復で大規模な近似問題を解く必要があり, 計算コストが大きくなる.

大規模な問題を解く方法として, 部分空間法が提案されている [2]. 部分空間法は次の反復点を探索する空間を低次元の部分空間に限定する方法である. 各反復の部分問題が低次元になるため, 一回の反復における計算量を減らすことができる. 部分空間法に関して, Fukushima [1] は, 並列変数変換法 (Parallel Variable Transformation algorithm, 以下 PVT 法) を提案している. PVT 法は各反復で複数の部分空間を生成し, それぞれの空間内で並列に部分問題を解き, それらの情報を用いて次の反復点を求めるアルゴリズムである.

また, 制約なし最小化問題を解くアルゴリズムの代表的な手法として信頼領域法が知られている. 信頼領域法は各反復において目的関数の 2 次モデルが妥当であると思われる領域 (信頼領域) 内でその 2 次モデルを最小化するステップを求める方法である. つまり, 各反復で制約つき最小化問題を解く必要がある. そのため, 元の問題が大規模な問題のときは, 各反復で大規模な制約つき最小化問題を解く必要

があり、計算コストがかかる。

本研究では大規模な問題 (1) を PVT 法を用いて解くアルゴリズムを提案する。提案アルゴリズムでは、低次元の部分問題に対して信頼領域法を用いる。また、いくつかのテスト問題に対して、並列プロセッサ上でアルゴリズムを実装して数値実験を行い、アルゴリズムの有効性を検証する。

なお、本稿の内容は主に [3] に基づく。

2 PVT 法

PVT 法は複数の低次元部分空間において並列に探索を行い、それらの情報を元に次の反復点を求める方法である。PVT 法は並列アルゴリズムを設計するための一般的な枠組みの一つになっている。

いま、 k 回目の反復点 $x^{(k)}$ と p 個の部分空間 $S_l^{(k)}$ ($l = 1, \dots, p$) が与えられているとする。 $S_l^{(k)}$ の次元は m_l で、その基底を $q_{l1}, q_{l2}, \dots, q_{lm_l}$ とする。このとき、部分空間 $S_l^{(k)}$ 上で目的関数 f を最小にする点を見つけるには次の最小化問題を解けばよい。

$$\min_{y \in R^{m_l}} \phi_l^{(k)}(y) \equiv f(x^{(k)} + Q_l^{(k)} y) \quad (2)$$

上式において、 $Q_l^{(k)}$ は $S_l^{(k)}$ の基底を並べた $n \times m_l$ 行列であり、 $Q_l^{(k)} = [q_{l1}, q_{l2}, \dots, q_{lm_l}]$ で定義される。PVT 法については、 p 個の部分空間によって n 次元空間が張られ、かつ、各反復で現れる低次元部分問題に対して一般的な降下法を適用すること、得られた p 個の解のうち最も良い解を次の反復点とすることで大域的収束性が保証されることが示されている [1]。

3 信頼領域法

信頼領域法は最急降下法や準ニュートン法等の直線探索法に比べ適用範囲が広く、数値的に頑健であることが知られている。信頼領域法は、各反復において目的関数の 2 次モデルが妥当であると思われる領域 (信頼領域) 内で 2 次モデルを最小化する。つまり各反復で次の部分問題 (3) を解く。

$$\begin{aligned} \min_d \quad & F^{(k)}(d) \equiv f(x^{(k)}) + \nabla f(x^{(k)})^T d + \frac{1}{2} d^T \nabla^2 f(x^{(k)}) d \\ \text{s.t.} \quad & \|d\| \leq \Delta^{(k)} \end{aligned} \quad (3)$$

ここで $\|d\| = \sqrt{d^T d}$ であり、 $\Delta^{(k)}$ は信頼半径と呼ばれる定数である。信頼領域は有界閉集合であるため、 $\nabla^2 f(x^{(k)})$ が正定値でなくとも、問題 (3) は最小解 $d^{(k)}$ をもつ。信頼領域法では、モデル関数の減少量

$$\Delta F^{(k)} \equiv F^{(k)}(0) - F^{(k)}(d^{(k)})$$

と目的関数の減少量

$$\Delta f^{(k)} \equiv f(x^{(k)}) - f(x^{(k)} + d^{(k)})$$

を比較し、反復点を更新するかどうかと信頼半径の増減を決める。信頼領域法のアルゴリズムは以下の通りである。

アルゴリズム (信頼領域法)

Step 1. 定数 $0 < \mu_1 < \mu_2 < 1$, $0 < \gamma_1 < 1 < \gamma_2$ を選ぶ。

$x^{(0)} \in R^n$, $\Delta^{(0)} > 0$ を与え, $k = 0$ とする。

Step 2. 終了条件を満たせば停止する。

部分問題 (3) を解き, $d^{(k)}$ を求める。

Step 3. $r^{(k)} = \frac{\Delta f^{(k)}}{\Delta F^{(k)}}$ を計算する。

$r^{(k)} \geq \mu_1$ ならば, $x^{(k+1)} = x^{(k)} + d^{(k)}$ とする。

$r^{(k)} < \mu_1$ ならば, $x^{(k+1)} = x^{(k)}$ とする。

Step 4. $r^{(k)} \geq \mu_2$ ならば, $\Delta^{(k+1)} = \gamma_2 \Delta^{(k)}$ とする。

$\mu_1 \leq r^{(k)} < \mu_2$ ならば, $\Delta^{(k+1)} = \Delta^{(k)}$ とする。

$r^{(k)} < \mu_1$ ならば, $\Delta^{(k+1)} = \gamma_1 \Delta^{(k)}$ とする。

$k = k + 1$ として Step2 へ戻る。

信頼領域法の考え方を説明する。部分問題 (3) の解 $d^{(k)}$ を採用したときに目的関数値が十分減少する ($r^{(k)} \geq \mu_1$) ならば, モデル関数が f の良い近似であると判断し, $x^{(k+1)} = x^{(k)} + d^{(k)}$ とする。目的関数値の減少量が少ないかもしくは逆に増加する ($r^{(k)} < \mu_1$) ならば, モデル関数が f の良い近似ではないと判断し, 反復点は $x^{(k)}$ から移動しない。また, そのようなときは Step 4 で信頼半径を小さくし, 次の反復では目的関数値が減少する可能性を増やす。一方, 目的関数とモデル関数のずれが小さいとき ($r^{(k)} \geq \mu_2$) は次の反復点でのステップが大きくなるように信頼半径を増加させる。

4 提案アルゴリズム

問題 (1) に対する PVT 法において, 各反復の部分問題を信頼領域法を用いて解くアルゴリズムを提案する。

Step 1. $x^{(0)} \in R^n$, $\Delta^{(0)} > 0$ を選び, $k = 0$ とする。

Step 2. 終了条件を満たせば停止する。

各 $l \in \{1, \dots, p\}$ について

$$\phi_l^{(k)}(y_l) \equiv f(x^{(k)} + Q_l^{(k)} y_l) \quad (l = 1, \dots, p)$$

として, p 個の部分問題 (4) を並列に解く。

$$\begin{aligned} \min_d \quad & \phi_l^{(k)}(0) + \nabla \phi_l^{(k)}(0)^T d + \frac{1}{2} d^T \nabla^2 \phi_l^{(k)}(0) d \\ \text{s.t.} \quad & \|d\| \leq \Delta^{(k)} \end{aligned} \tag{4}$$

Step 3. 得られた p 個の解 $d_l^{(k)}$ ($l = 1, \dots, p$) に対して,

$$d_{l'}^{(k)} \equiv \arg \min_{1 \leq l \leq p} \phi_l^{(k)}(d_l^{(k)})$$

を求める.

Step 4. 信頼領域法と同様に, $\phi_{l'}^{(k)}$ の減少量と $\phi_{l'}^{(k)}$ の 2 次モデル関数の減少量を比較し, 次の反復点 $x^{(k+1)}$ と信頼領域 $\Delta^{(k+1)}$ を定める.

$k = k + 1$ として Step2 へ戻る.

5 実験結果

以下の問題に対して, 提案アルゴリズムを実装し, 数値実験を行った. なお, 提案アルゴリズムにおいて, 探索する部分空間 $S_l^{(k)}$ は座標方向で張られる空間, つまり, $S_l^{(k)}$ の基底を並べた行列 $Q_l^{(k)}$ は次で定義する.

$$Q_l^{(k)} = [e_{(l-1)n/p+1}, \dots, e_{ln/p}]$$

$$\begin{aligned} \text{問題 1} & \begin{cases} \min f(x) = \sum_{i=1}^{n-1} \{(x_i^2 + x_n^2)^2 - 4x_i + 3\} \\ \text{初期点 } x^{(0)} = (3, \dots, 3) \end{cases} \\ \text{問題 2} & \begin{cases} \min_{x \in R^n} f(x) = 1 + \sum_{i=1}^{n-1} \{100(x_i - x_{i-1}^2)^2 + (1 - x_{i-1})^2\} \\ \text{初期点 } x^{(0)} = (0, \dots, 0)^T \end{cases} \end{aligned}$$

以下の実験結果において, T_p は Step 2 で p 個の部分問題を p 個のプロセッサで並列に計算したときの実行時間を表す. $p = 1$ のときは, 通常の信頼領域法アルゴリズムを用いたことを意味する.

$n = 400$	$p = 1$	$p = 4$	$p = 8$	$p = 16$
$T_p(\text{sec})$	0.298	0.028	0.032	0.034
反復回数	10	28	51	98
$n = 1200$	$p = 1$	$p = 4$	$p = 8$	$p = 16$
$T_p(\text{sec})$	3.524	0.340	0.140	0.111
反復回数	11	28	52	99

表 1: 問題 1 に対する実験結果

$n = 400$	$p = 1$	$p = 4$	$p = 8$	$p = 16$
$T_p(\text{sec})$	14.401	0.762	0.329	0.448
反復回数	937	1112	1457	2281
$n = 1200$	$p = 1$	$p = 4$	$p = 8$	$p = 16$
$T_p(\text{sec})$	714.23	20.349	6.042	3.330
反復回数	2752	2971	3222	3814

表 2：問題 2 に対する実験結果

提案アルゴリズムでは、通常の信頼領域法に比べ反復回数が増加した。それは次の反復点を低次元部分空間で探すため、反復点の動きが制限されたためと思われる。しかし、一回の反復における計算量が少ないため、全体での実行時間は短縮された。また、問題の次元 n が大きいほど並列プロセッサ数 p を増やしたときの効果が表れることが分かる。

6 まとめと今後の課題

本稿では大規模な問題に対し、PVT 法と信頼領域法組み合わせたアルゴリズムを提案した。また、提案アルゴリズムを実装し、数値実験を通してアルゴリズムの有効性を検証した。

今後の課題としては、探索する部分空間の提案やアルゴリズムの **Step 2** において得られる p 個の解を上手く利用するなど、アルゴリズムの効率化や既存のソルバーとの比較などが挙げられる。また、本稿の数値実験ではプログラムを作成する際に、大規模問題においてしばしば生じる行列の疎構造を考慮していない。そういったプログラムの実装上の改善も今後の重要な課題である。

参考文献

- [1] M. Fukushima, Parallel variable transformation in unconstrained optimization, SIAM J. Optim, Vol. 8, pp. 658–672 (1998)
- [2] Y. X. Yuan, Subspace method for large scale nonlinear equations and nonlinear least squares, Optim. Eng., Vol. 10, pp. 207–218 (2009)
- [3] 高須 啓介, 信頼領域を用いた大規模非線形計画問題に対する並列部分空間法, 京都大学工学部情報学科数理工学コース 特別研究報告書 (2010)