大規模問題に対するSQP法

02202793　静岡大学　＊齋賀　大賢　Hiroyoshi　SAIGA
01702180　静岡大学　八巻　直一　Naokazu　YAMAKI
01702330　東京理科大学　矢部　博　Hiroshi　YABE

1　はじめに

制約条件付き最小化問題に対する数値解法は、いろいろと開発されており、有力な解法がいくつか存在する。これらのいくつかの解法は、小規模ないしは中規模の非線形計画問題では、効率が良い解法であると言える。現在、最も性能の良い解法として「逐次２次計画法 (Sequential Quadratic Programming Method ; 以下 SQP 法)」があげられる。

一方、大規模な非線形計画問題に注目してみると、「いかに効率良く解くか」という大きな課題がある。中規模以下の非線形計画問題に用いられる数値解法そのままでは、必ずしも大規模問題に適しているとは言えない。大規模な非線形計画法の数値解法としては、例えば、逐次線形計画法 (SLP 法) があり、決して収束が速いとは言えないが、それなりに大規模問題に適用可能であることが大きな利点である。

そこで、中規模以下の非線形計画問題で最も効率の良い解法「SQP 法」の大規模問題へ適用を考える。本研究では、A Large Scale SQP 法の提案をする。

2　大規模問題

対象とする非線形計画問題は以下の通りである。

$$
\min f(x)
$$
$$
\text{s.t.} \begin{cases} g_i(x) \le 0, & i = 1, \cdots, m \\ h_j(x) = 0, & j = 1, \cdots, l \end{cases}
$$
$$
\left(\begin{array}{l} x \in R^n \\ f : R^n \to R \\ g : R^n \to R^m \\ h : R^n \to R^l \end{array} \right)
$$

本研究では、このような非線形計画問題のうち大規模で、特に制約条件が少ない場合について考える。例えば、変数 x の次元が "数万個" に対して、制約条件数が "数百個" のような問題である。

3　SQP 法

SQP 法 [3], [4] は、制約条件のない最小化問題における準 Newton 法の原理を、制約条件のある最小化問題に上手く組み入れることによって優れた性能の実現をねらった数値解法といえる。主な特徴として、以下の２点が挙げられる。

- 探索ベクトルを求める際に、毎回２次計画問題を解く。
- $\nabla_{xx}L$ を近似行列 B で置き換える。
  $\left(L(x,\lambda,\mu) = f(x) + \lambda^T g(x) + \mu^T h(x)\right)$

これらの特徴を持つ SQP 法のアルゴリズムは、以下のようになる。

STEP0)　初期設定
STEP1)　QP 部分問題を解いて探索方向を求める
STEP2)　停止判定
STEP3)　ステップ幅 α の決定
STEP4)　x の更新
　　$x_{k+1} = x_k + \alpha d_k$
STEP5)　近似行列 B の更新
　　$k = k+1$ として STEP1) へ

4　QP subproblem

SQP 法アルゴリズム　STEP1) の QP 部分問題は次のように表される。

$$\min \frac{1}{2}d^T B d + \nabla f(x)^T d$$
$$\text{s.t.} \begin{cases} g(x) + \nabla g(x)^T d \leq 0 \\ h(x) + \nabla h(x)^T d = 0 \end{cases}$$

これは、現在の近似点 x において、探索方向 d を求めるために解く問題である。大規模問題の場合、近似行列 $B \in R^{n \times n}$ が非常に大きくなるため、QP 部分問題を解くには、膨大な Memory が必要であるという問題が発生する。そこで、我々は Dual 側の QP 部分問題に注目した。

5 Dual QP subproblem

Dual 側の QP 部分問題は下記のようになる。

$$\min \frac{1}{2}w^T P w + q^T w$$
$$\text{s.t.} \quad \lambda \geq 0$$

$$w = \begin{bmatrix} \lambda \\ \mu \end{bmatrix} \in R^j$$
$$P = \begin{bmatrix} \nabla g(x_k)^T \\ \nabla h(x_k)^T \end{bmatrix} B^{-1} \begin{bmatrix} \nabla g(x_k)^T \\ \nabla h(x_k)^T \end{bmatrix}^T \in R^{j \times j}$$
$$q = \begin{bmatrix} \nabla g(x_k)^T \\ \nabla h(x_k)^T \end{bmatrix} B^{-1} \nabla f(x_k) - \begin{bmatrix} g(x_k) \\ h(x_k) \end{bmatrix} \in R^j$$
$$m + l = j$$

Dual 側の QP 部分問題をとることにより、元の問題の次元に関係なく、次元数が制約条件数になる。したがって、制約条件の少ない大規模問題に有効であると言える。この QP 部分問題を Gill-Murray 法 [1] で解く。

しかし、P 及び q を求める際に、近似行列 B^{-1} が存在するため、大規模な Memory が必要であるという問題に関しては解決したとは言えない。

この問題を解決するため、Limited Memory BFGS 法 [2], [3] を用いる。

6 Limited Memory BFGS 法

Limited Memory BFGS 法とは、近似行列を数本のベクトルで表すことにより、記憶領域の削減を行う手法である。

B^{-1} の公式（BFGS 公式）は次のようになる。

$$B_k^{-1} = H_k = V_{k-1}^T H_{k-1} V_{k-1} + G_{k-1}$$
$$\left(\begin{array}{l} V_{k-1} = I - \frac{y_{k-1}s_{k-1}^T}{s_{k-1}^T y_{k-1}} \\ G_{k-1} = \frac{s_{k-1}s_{k-1}^T}{s_{k-1}^T y_{k-1}} \\ s_k = x_{k+1} - x_k \\ y_k = \nabla L(x_{k+1}, \lambda_{k+1}, \mu_{k+1}) - \nabla L(x_k, \lambda_{k+1}, \mu_{k+1}) \end{array} \right)$$

更新式を書き下してみると以下のようになる。

$$\begin{aligned} B_k^{-1} &= U_0 U_0^T + U_1 \frac{s_0 s_0^T}{s_0^T y_0} U_1^T + \cdots + U_{k-2} \frac{s_{k-3}s_{k-3}^T}{s_{k-3}^T y_{k-3}} U_{k-2}^T \\ &\quad + U_{k-1} \frac{s_{k-2}s_{k-2}^T}{s_{k-2}^T y_{k-2}} U_{k-1}^T + \frac{s_{k-1}s_{k-1}^T}{s_{k-1}^T y_{k-1}} \\ &\quad (U_i = V_{k-1} \cdots V_i) \end{aligned}$$

したがって、$0 \sim (k-1)$ ステップまでの s、y を記憶しておけば、B を求めることが出来る。

Limited Memory では、さらに次のような近似を行う。

$$\begin{aligned} B_k^{-1} &= U_t U_t^T + U_{t+1} \frac{s_t s_t^T}{s_t^T y_t} U_{t+1}^T + \cdots + U_{k-2} \frac{s_{k-3}s_{k-3}^T}{s_{k-3}^T y_{k-3}} U_{k-2}^T \\ &\quad + U_{k-1} \frac{s_{k-2}s_{k-2}^T}{s_{k-2}^T y_{k-2}} U_{k-1}^T + \frac{s_{k-1}s_{k-1}^T}{s_{k-1}^T y_{k-1}} \end{aligned}$$

これにより、$t \sim k-1$ ステップ分の s、y を記憶しておけば、行列 B を十分に再構成することが出来る。実際には、過去 4 〜 5 ステップ分の s、y を記憶させておけばよく、計算量と Memory を大幅に削減することができる。

参考文献

[1] P.E.Gill and W.Murray : Numerically stable methods for quadratic programming, *Mathematical Programming* 14 (1978) pp.349-372.

[2] J.Nocedal : Updating quasi-Newton matrices with limited storage, *Mathematics of Computation* 35 (1980) pp.773-782.

[3] 矢部博, 八巻直一 : 非線形計画法, 朝倉書店, 1999.

[4] 八巻直一、宮田雅智、本郷茂、高橋悟、矢部博、内田智史 : パソコン FORTRAN 版非線形プログラミング, 日刊工業新聞社, 1991.