大規模非線形最適化パッケージ NUOPT 2.0 の開発 II. インプリメンテーション

01307380 数理システム　*田辺隆人 TANABE Takahito
01701240 数理システム　山下浩 YAMASHITA Hiroshi

1　概要

一般の大規模非線形計画問題に対して大域的収束性を保証する信頼領域法及び，Hesse 行列の非負定値性が仮定できる状況において有利な線形探索法を同一モジュールにて実現した．

実際の応用で現れる大規模問題を解くにあたってはアルゴリズムのみならず，数値的安定化のための前処理や，問題の sparsity を生かす計算の効率化が重要となるが，NUOPT においては線形関数の係数行列の scaling や，不要な変数・関数の除去 (cleaning) 等の前処理機能を充実させて数値的安定化と高速化を図るとともに，行列解法部分にブロックピボット付き multifrontal 法を採用するなど，計算の効率化を行っている．

NUOPT 本体は，関数値や微係数等，非線形関数についての情報を与えるインタフェースと連結して動作する．問題に特化したインタフェースとの連結も可能であるが，ここでは汎用インタフェースとしてモデリング言語 SIMPLE を用いたものを紹介する．その場合，高速自動微分法によって微係数の計算は自動化される．

以下では本体部，及びインタフェース部について解説する．最後の数値実験においては小規模及び大規模な非線形最適化問題の求解を行い，有効性を検証する．

2　本体部

2.1　問題の入力と前処理

非線形最適化問題は一般形として入力する．アルゴリズム実現部は変数に関する一般の上下限制約を直接扱うことができる．

数値的安定化のために，線形関数の係数行列には各行，各列をその非零要素の絶対値の最大，最小の相乗平均で割る scaling を施す．また，関数に現れない変数や固定された変数，値が決定する関数を除去し，アルゴリズム実現部で扱う変数や関数の数を削減する (cleaning)．これらの前処理は大規模な線形及び二次計画問題において特に有効である．

2.2　アルゴリズムの選択

直線探索アルゴリズムと信頼領域アルゴリズムのいずれを適用するかは最適化の実行時に指定する．信頼領域アルゴリズムが最も汎用的であるが，線形計画問題や凸二次計画問題，Hesse 行列の近似行列が非負定値となる準ニュートン法が適用可能な小規模非線形最適化問題等，(近似)Hesse 行列 G の非不定値性が保証可能な状況においては計算効率上の観点から直線探索アルゴリズムの適用が望ましい．

2.3　行列解法

反復ベクトルの計算の際に現れる一次方程式

$$\begin{bmatrix} G+X^{-1}Z & -A^t \\ -A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} b_x \\ b_y \end{bmatrix}$$

の解法部分は特に大規模問題の場合のパフォーマンスを大きく左右する．左辺行列は疎となることが多いが，対称不定値であるので pivoting には分解過程で現れる fill-in 減少のみならず，数値的安定性をも考慮しなければならない．

NUOPT は直接法によって一次方程式の求解を行うが，その際の行列の分解のアルゴリズムとして multifrontal 法 [1] を採用している．multifrontal 法において，大規模疎行列は小規模の密行列 (frontal matrix) の重ねあわせとして表現される．分解算法全体は fill-in 減少を考慮した minimum degree アルゴリズムが与える pivoting に基いて進行するが，必要ならば数値的安定のための pivoting が frontal matrix 中において行われる．(２×２) のブロックピボットを一部使用することによって算法の進行中にも行列の対称性を保持し，計算を効率化することができる．

3 インタフェース部

汎用のインタフェースとしてモデリング言語 SIMPLE を用いたものを紹介する．問題を Set(集合)，Element(要素)，Parameter(定数)，Variable(変数)，Objective(目的関数) 等のクラスを使用して，C++ のルーチンとして記述する．最適化の実行に必要な2階の微係数までは高速自動微分法 [2] によって自動計算されるため，ユーザは問題の記述を与えるのみで良い．

非線形最適化問題

minimize:

$$\sum_{j=1}^{5} e_j x_j + \sum_{i=1}^{5}\sum_{j=1}^{5} c_{ij} x_i x_j + \sum_{j=1}^{5} d_j x_j^3$$

s.t.:

$$\sum_{j=1}^{5} a_{ij} x_j - b_i \geq 0, \quad i = 1, \cdots 10$$

$$0 \leq x_i, \quad i = 1, \cdots 5$$

(Hock & Schittkowski No.86)

を SIMPLE によって記述した例を以下に示す．

```
void HS86()
{
    Set S,T;
    S = "1 ... 5";
    T = "6 ... 10";

    Parameter a(S|T,S),b(S|T)
              ,c(S,S),d(S),e(S);
    Element i(S),j(S),k(S|T);
    Variable x(S);
    Objective f;

    f = sum(e[j]*x[j],j)
         +sum(c[i,j]*x[i]*x[j],i,j)
          +sum(d[j]*x[j]*x[j]*x[j],j);

    minimize(f);
    sum(a[k,j]*x[j],j) >= b[k];
    0.0 <= x[i];
}
```

4 数値実験

以下の問題集を用いて数値実験を行った．適用したのはいずれも信頼領域アルゴリズムである．

4.1 Hock & Schttkowski テスト問題集

15 変数程度までの小規模非線形最適化問題が収められている [3]．同一のパラメータにより収蔵 115 問すべての最適解が求められた．KKT 条件を満たす点が存在しない問題 (No.13) を除いた反復回数の合計は 1408 回 (12.2 回/問)，関数値の評価回数の合計は 2330 回 (20.3 回/問) である．

4.2 CUTE テスト問題集

LANCELOT の開発グループがインタフェースツール群とともに提供している問題集で 大小 500 問程度が収められている [4]．変数・制約式の総数が 20 以上または可変で 2 階微分までが解析的に与えられている 148 問題を選んで実験を行ったところ 146 問の最適化が成功した．変数・制約式の数が可変のものについては最大規模としている．(ただし，146 問中 14 問は規模を縮小して初めて最適化が成功した．) 最適化が成功した問題についての平均反復回数は 27.3 回 である．

以下に計算結果の一例を示す．(n:変数総数 m:制約式総数．計算時間の単位は秒．)

問題名	n	m	反復回数	計算時間
LEAKNET	138	135	34	3.730
HYDROELL	1007	1006	16	20.87
HELSBY	1130	1121	26	85.5
SVANBERG	5000	5000	18	314.0
GRIDNETF	7564	3844	23	663.6
CORKSCRW	8997	7000	14	511.1
SREADIN3	10002	5001	34	1323
HAGER4	10000	5000	11	431.2
UBH5	19997	14000	4	828.0
TRAINF	20000	10002	24	4052

使用マシン：Sparc Station 10

参考文献

[1] I.S.Duff, and J.K.Reid, *The Multifrontal solution of indefinite sparse symmetric linear systems.*,ACM Transaction on Mathematical Software.Vol.9,No.3 ,302-325,1983.

[2] 伊理正夫，久保田光一，高速自動微分法 *(I)*，応用数理,Vol.1,17-35,1991.

[3] W.Hock, and K.Shittkowski, *Test examples for nonlinear programming codes*, Springer Verlag, 1981.

[4] I.Bongartz, A.Conn, N.Gould, and Ph.L.Toint, *CUTE:Constrained and Unconstrained Testing Environment* ,Research Report RC 18860, IBM T.J. Watson Research Center, Yorktown, USA. ,1993