Nonlinear Constrained Optimization of a fourth oder multivariable function

I need to find the minimum of a fourth order of an Objective Function of N variables with NC1

scalar constrains and M2 nonlinear constrains

So :

- Given an Objective Function of forth order of N variables
- The variables stay in a N-dimensional vector X[N] of doubles
- N (the number of variables) is known integer
- Given a positive integer M (see later, it needs to build Object Function)
- Given a positive integer M1 (see later, it needs to build Object Function)
- Given a positive integer M2 (see later, it needs to build Object Function)
- M = M1 + M2
- Given an matrix CONN of order Mx3 of positive integers (see later, it needs to build Object Function)
- Given an M1-dimensional double array W[]
- Given NC1 scalar constrains,an integer vector z[NC1] and a double RC[NC1] vector.It means that several variables must have assigned values.
- Given M2 Non Linear constrains ,two vector of integer ii[M2] and jj[M2] and a RK[M2] vector of positive doubles

All previous parameters are read from an ASCII FILE (see example1.dat or example2.dat)

Moreover see source.cpp for a C++ pseudo-code to read them from ascii files

ObjFunc (X) =

$$\sum_{\kappa=0} \quad w[k]*\{ (X[j] - X[i])^{2} +(X[j+1] - X[i+1])^{2} +(X[j+2] - X[i+2])^{2} \}^{2}$$

(Index k in the SUMM starts from zero to M1-1 (included))

where :

$i = 2 * CONN[k][2] +1$

$j = 2 * CONN[k][1] - 1$

The Objective function is built in this way (c++ pseudo code) :

```
void ObjFunc()
{
        int from,to;
        int I,j,k;
```

```
    obj =0.0 ;
    for(k = 0 ; i < M1; k ++)
    {
        from = CONN[k][2];
        to = CONN[k][1];
        J= 2*from+1; I = 2*to-1
        A1 = pow ( (X[j] - X[i] ), 2) ;
        A2 = pow ( (X[j+1] - X[i+1] ), 2) ;
        A3 = pow ( (X[j+2] - X[i+2] ), 2) ;
        Obj = obj + W[k] * pow( (A1+A2+A3),2)
    }
}
```

subjects to following constaints :

NC1 SCALAR CONSTRAINTS ON SEVERAL VARIABLES:

```
X[z[0]] = RC[0];
X[z[1]] = RC[1];
….. …..
X[z[NC1-1]] = RC[NC1-1];
```

M2 NON LINEAR CONSTRAINTS :

```
pow ((X[ii[0]] - X[jj[0]] ),2)+ pow ((X[ii[0] +1] - X[jj[0]+1] ),2)+pow ((X[ii[0]+2] -
X[jj[0]+2]),2) - RK[0] = 0.0;

pow ((X[ii[1]] - X[jj[1]] ),2)+ pow ((X[ii[1] +1] - X[jj[1]+1] ),2)+pow ((X[ii[1]+2] -
X[jj[1]+2]),2) - RK[1] =0.0;
…..
pow ((X[ii[M2-1]] - X[jj[M2-1]] ),2)+ pow ((X[ii[M2-1] +1] - X[jj[M2-1]+1] ),2)+pow
((X[ii[M2-1]+2] - X[jj[M2-1]+2]),2) - RK[M2-1] =0.0;
```

After solved (it means , the optimum solution have been found , write solution in a file , using WriteFile from source.cpp

example 1 (see example1.dat)

```
////////////////
N =12
M = 6
M1 = 4
M2 = 2
Nc1 = 6
CONN[0][0] = 0      CONN[0][1] = 1      CONN[0][2] = 2
CONN[1][0] = 1      CONN[1][1] = 2      CONN[1][2] = 3
CONN[2][0] = 2      CONN[2][1] = 3      CONN[2][2] = 4
CONN[3][0] = 3      CONN[3][1] = 4      CONN[3][2] = 1
CONN[4][0] = 4      CONN[4][1] = 1      CONN[4][2] = 3
CONN[5][0] = 5      CONN[5][1] = 2      CONN[5][2] = 4

W[0] = 1.0;
W[1] = 1.0;
W[2] = 1.0;
W[3] = 1.0;

Z[0] = 0
Z[1] = 1
Z[2] = 2
Z[3] = 3
```

Z[4] = 4
Z[5] = 5

RC[0] = 0.0
RC[1] = 0.0
RC[2] = 0.0
RC[3] = 353.53
RC[4] = 0.0
RC[5] = 0.0

ii[0] = 0
ii[1] = 1
Jj[0] = 2
jj[1] = 3

RK[0]=500.0;
RK[1]= 500.0;

Trial or start solution

X[0] = 0.0 //assigned value
X[1] = 0.0 //assigned value
X[2] = 0.0 //assigned value
X[3] = 353.0 //assigned value
X[4] = 0.0 //assigned value
X[5] = 0.0 //assigned value
X[6] = 433.83
X[7] = 248.56
X[8] = 0.0
X[9] = -131.03
X[10] = 123.1994
X[11] = 0.0

example 2 : see example2.dat
//////////////////
N=18 M = 5 M1 = 4 M2 = 1 NC1= 12
CONN[0][0] = 0   CONN[0][1] = 1   CONN[0][2] = 2
CONN[1][0] = 1   CONN[1][1] = 2   CONN[1][2] = 3
CONN[2][0] = 2   CONN[2][1] = 2   CONN[2][2] = 4
CONN[3][0] = 3   CONN[3][1] = 4   CONN[3][2] = 6
CONN[4][0] = 4   CONN[4][1] = 4   CONN[4][2] = 5

W[0] = 1.0;
W[1] = 1.0;
W[2] = 1.0;
W[3] = 1.0;

Z[0] = 0
Z[1] = 1
Z[2] = 2
Z[3] = 6
Z[4] = 7
Z[5] = 8
Z[6] = 9
Z[7] = 10
Z[8] = 11
Z[9] = 15
Z[10] = 16
Z[11] = 17

RC[0] = 0.0   RC[1] = 0.0   RC[2] = 0.0
RC[3] = 1500.00   RC[4] = 0.0   RC[5] = 0.0
RC[6] = 0.0   RC[7] = 2000.0   RC[8] = 0.0
RC[9] = 1500.00   RC[10] = 2000.0   RC[11] = 0.0

ii[0] = 1 Jj[0] = 3
RK[0]= 1120.8141;

Trial solution
X[0] = 0.0 //assigned value
X[1] = 0.0 //assigned value
X[2] = 0.0 //assigned value
X[3] = 500.0
X[4] = 0.0
X[5] = 0.0
X[6] = 1000.00 //assigned value
X[7] = 0.0 //assigned value
X[8] = 0.0 //assigned value
X[9] = 889.5091
X[10] = 1050.9552
X[11] = 0.0
X[12] = 0.0 //assigned value
X[13] = 2000.0 //assigned value
X[14] = 0.0 //assigned value
X[15] = 1500.0 //assigned value
X[16] = 2000.0 //assigned value
X[17] = 0.0 //assigned value