Optimización numérica

Proyecto final 2. MPI

Santiago Novoa Pérez. 8 de diciembre del 2015.

Introducción

Los métodos de puntos interiores (MPI) representan una alternativa a los métodos de conjunto para resolver problemas de optimización no lineal con restricciones de desigualdad. Mientras que la dificultad de los métodos como PCS está en encontrar el conjunto de restricciones de desigualdad que son activas en la solución (que alcanzan la igualdad), los métodos de puntos interiores abandonan tal búsqueda manteniéndose siempre dentro de las regiones de desigualdad en lugar de avanzar por su frontera; esto se logra con diversas penalizaciones o barreras que se le agregan a la función a minimizar (por esto es que los métodos de puntos interiores también se conocen como métodos de barrera).

Los métodos MPI también resuelven una secuencia de subproblemas de optimización (parecido a PCS en ese aspecto), sin embargo, lo único que cambia entre subproblema y subproblema es el parámetro de penalización (o el peso que se le da a la barrera sobre la función μ), es decir, qué tanto deja el método a los valores reales acercarse a la frontera de las restricciones de desigualdad. Para alcanzar la solución al problema original se intenta avanzar tanto hacia la frontera como sea necesario (el parametro de penalización μ debería tender a cero para que el método se aproxime a la solución).

MPI es apropiado para problemas grandes y pequeños con funciones no lineales; en estos casos la solución se alcanza en un número de iteraciones sustancialmente menor que n.

Nuestro enfoque para el método (aunque no es el único) será resolver el sistema de KKT de cada subproblema transformando al sistema de tal manera que tengamos las condiciones necesarias (simetría y curvatura positiva) para usar el algoritmo ldl (el cual nos deja usar la estructura de la matriz en la resolución de cada subproblema sin necesidad de calcular bases de espacios nulos ni matrices inversas paso a paso).

Una vez resuelto el sistema para una μ se actualizará el parámetro (reduciéndolo en .1) y se volverá a calcular el sistema de KKT correspondiente. Es importante que al actualizar el parámetro (ciclo externo) se elija comenzar el método (resolver el sistema de KKT resulta en un método iterativo que será nuestro ciclo interno) con ${x_0}^{k+1}$, la aproximación inicial, siendo el valor de la solución del subproblema anterior (${x_*}^k$). Al hacer ésto nuestro método empezará a converger de manera mucho más rápida cada que actualicemos el parámetro de barrera μ.

Historia

Los métodos de barrera para optimización se desarrollaron en los años sesenta, pero su popularidad cayó en las siguientes dos décadas. El éxito que tuvieron los métodos de puntos interiores para programación lineal en los años noventa renovó el interés en estos métodos para el caso no lineal. Para finales de 1990, había surgido una nueva generación de métodos y software para programación no lineal. Los métodos de puntos interiores a menudo son más rápidos que los métodos de programación cuadrática sucesiva en grandes problemas, sobre todo cuando el número de variables libres es grande. [1]

Construcción del método

.
El problema que buscamos resolver es:

$$\begin{array}{rc} \text{minimizar} & f(x) \\ \text{s.a.} & h(x) = 0 \\ & g(x) \leq 0 \end{array}$$

$$\begin{aligned} f : \mathbb{R}^n &\longrightarrow \mathbb{R} \\ h : \mathbb{R}^n &\longrightarrow \mathbb{R}^{m_h} \\ g : \mathbb{R}^n &\longrightarrow \mathbb{R}^{m_g} \end{aligned}$$

* f, h, g dos veces continuamente diferenciables

Transformamos el problema para evitar adivinar las restricciones activas:

$$\begin{array}{rc} \text{minimizar} & \psi(x) = f(x) - \mu * \sum \ln(s_i) \\ \text{s.a.} & h(x) = 0 \\ & g(x) + s = 0; \qquad s \geq 0 \end{array}$$

La barrera de este problema (barrera logarítmica) evita que los valores de s sean cero ya que penaliza demasiado a la función $\psi(x)$ para cualquier $s_i = 0$. De esta manera podemos resolver el sistema de KKT sin tener que preocuparnos por si estamos llegando a la frontera de las desigualdades.

[1] Nocedal, Jorge, and Stephen Wright. Numerical optimization. Springer Science & Business Media, 2006.

El siguiente paso a seguir sería encontrar los valores de (x,s) factibles que minimizan al problema. Como criterio se utilizan las condiciones de KKT.

- *C.N.P.O.*

Se buscan multiplicadores tales que en el minimizador local el gradiente de la función objetivo (en este caso la función de barrera $\psi(x)$) sea combinación lineal (con los multiplicadores λ como coeficientes) de los gradientes de las restricciones y se mantenga factibilidad. Para esto primero definiremos la función *Lagrangiana* del problema y cambiaremos ciertas variables para facilitar su uso:

$$\mathcal{L}(z, \lambda; \mu) = \phi_\mu(x) + \lambda_h^T h(x) + \lambda_g^T (g(x) + s) \tag{1}$$

Donde $s > 0$ es un vector de variables de holgura, $z = (x, s)^t$ y $\mu > 0$ es el parámetro de barrera.

A partir de esta función se pueden obtener las condiciones necesarias de primer orden (con las cuáles construiremos nuestro sistema de KKT):

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial x} &= \nabla f(x) + A_h(x)^t * \lambda_h + A_g(x)^t * \lambda_g \\
(\textit{donde } A_h,\ A_g \textit{ son} &\quad \textit{las Jacobianas de las restricciones } h(x),\ g(x)) \\
\frac{\partial \mathcal{L}}{\partial s} &= -\mu * S^{-1} e + \Lambda_g e \\
\frac{\partial \mathcal{L}}{\partial \lambda_h} &= h(x) \\
\frac{\partial \mathcal{L}}{\partial \lambda_g} &= g(x) + s;
\end{aligned}$$

Las C.N.P.O. nos indican que necesitaramos igualar las cuatro derivadas de arriba a cero, sin embargo, vamos a resolver el sistema de KKT con iteraciones del método de Newton por lo que para encontrar esa solución todava se necesitan las segundas derivadas de cada una de las ecuaciones anteriores con respecto a x, s, λ_h y λ_g.

- *C.N.S.O.*

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}(z,\lambda;\mu) & 0 & A_h^T(x) & A_g^T(x) \\ 0 & -\mu S^{-2}e & 0 & \mathbb{I} \\ A_h(x) & 0 & 0 & 0 \\ A_g(x) & \mathbb{I} & 0 & 0 \end{bmatrix} \begin{bmatrix} d_x \\ d_s \\ d_{\lambda_h} \\ d_{\lambda_g} \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + A_h(x)^t\lambda_h + A_g(x)^t\lambda_g \\ -\mu S^{-1}e + \Lambda_g e \\ h(x) \\ g(x) + s \end{bmatrix} \tag{2}$$

$\lambda = [\lambda_h, \lambda_g]^t$

$$\Longleftrightarrow$$

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}(z,\lambda;\mu) + A_h^t(x)(\lambda_h^+ - \lambda_h) + A_g^t(x)(\lambda_g^+ - \lambda_g) \\ -\mu S^{-2}d_s + (\lambda_g^+ - \lambda_g) \\ A_h(x)d_x \\ A_g(x)d_x + d_s \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + A_h(x)^t\lambda_h + A_g(x)^t\lambda_g \\ -\mu S^{-1}e + \lambda_g \\ h(x) \\ g(x) + s \end{bmatrix}$$

$$\Longleftrightarrow$$

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}(z,\lambda;\mu) + A_h^t(x)(\lambda_h^+) + A_g^t(x)(\lambda_g^+) \\ S*(-\mu S^{-2}d_s + (\lambda_g^+)) \\ A_h(x)d_x \\ A_g(x)d_x + d_s \end{bmatrix} = - \begin{bmatrix} \nabla f(x) \\ S*(-\mu S^{-1}e) \\ h(x) \\ g(x) + s \end{bmatrix}$$

$$\begin{aligned} sust.\ \hat{d}_s = &\quad S*d_s \\ \mu S^{-1} = &\quad \Lambda_g \end{aligned}$$

$$\Longrightarrow$$

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}(z,\lambda;\mu) & 0 & A_h^T(x) & A_g^T(x) \\ 0 & S\Lambda_g & 0 & S \\ A_h(x) & 0 & 0 & 0 \\ A_g(x) & S & 0 & 0 \end{bmatrix} \begin{bmatrix} d_x \\ \hat{d}_s \\ \lambda_h^+ \\ \lambda_g^+ \end{bmatrix} = - \begin{bmatrix} -\nabla f(x) \\ \mu e \\ -h(x) \\ -g(x) - s \end{bmatrix} \tag{3}$$

Finalmente llegamos al sistema que se va a resolver paso a paso (ciclo interno) actualizando (z,λ).

$$z^+ = z + \alpha_z d_s, \qquad \lambda^+ = \lambda + \alpha_\lambda d_\lambda$$

Para la actualización se cuida que ni $\lambda_g^+ \leq 0$ ni $s_{k+1} \leq 0$. A esto se le llama protección del método de Newton.

Para el ciclo externo sólo es necesario, una vez llegado al punto mínimo, actualizar $\mu_{k+1} = \frac{\mu_k}{10}$ y repetir el procedimiento con $x_0^{k+1} = x_*^k$ y calculando la λ por mínimos cuadrados:

$$A^t\lambda = -\nabla f(x)$$

$$\Longleftrightarrow$$

$$\lambda = -(AA^t)^{-1}A\nabla f(x)$$

Pseudo-código

I. Fase inicial

Dar aproximaciones iniciales x_0, s_0; $\mu_0 = 1$

Generar λ_0 por mínimos cuadrados.

$k_{int} = 0$ y $k_{ext} = 0; fact = 0{,}1$

II. Cuerpo

While not(convergencia) & $k_{ext} \leq 10$

While not(convergencia BL) & $k_{int} \leq 100$

Construir la matriz de coeficientes K

if inercia_correcta

Obtener $d_x, d_s, d_{\lambda_h}, d_{\lambda_g}$

Obtener α_z, α_λ y cortar pasos

Obtener z^+ y λ^+

$k_{int} = k_{int} + 1;$

else

print (inercia incorrecta)

end

$k_{ext} = k_{ext} + 1;$

$\mu_{k_{ext}} = fact * \mu_{k_{ext-1}};$

$x_0^{k_{ext}} = x_*^{k_{ext}-1};$

end

Ejemplos de funciones

```
An Interior Point Method solving problem: alsotame
Problem dimensions: n_x =    2;      m   =    5
                    n_s =    4;      m_e =    1

Parameters:         factor ................. 1.0e-01
                    mu0 .................... 1.0e+00
                    maxbar .................   10
                    Step lengths ........... Separate

Objective   ...............  1.00000000e+00
Barrier     ............... -1.19722458e+00
||c||_2     ...............  3.63429132e+00

k_ext  k_int        f               fbarrera         ||W||        ||c||         mu
------------------------------------------------------------------------------------
  1      1    1.92041590e-01  -1.23265157e+00    7.97e+00     5.29e-01     1.00e+00
  1      2    1.98987533e-01  -1.75162787e+00    3.46e+00     6.59e-02     1.00e+00
  1      3    2.32582914e-01  -1.79040666e+00    1.02e-01     9.57e-05     1.00e+00
  1      4    2.24999387e-01  -1.79178579e+00    7.96e-03     2.92e-13     1.00e+00
  1      5    2.24259023e-01  -1.79179792e+00    1.24e-04     4.44e-16     1.00e+00
  1      6    2.24229998e-01  -1.79179794e+00    4.92e-07     4.44e-16     1.00e+00
------------------------------------------------------------------------------------
  2      1    2.03424409e-01   4.90361773e-03    7.78e-02     0.00e+00     1.00e-01
  2      2    1.88469224e-01  -6.30731533e-03    6.12e-02     0.00e+00     1.00e-01
  2      3    1.42854222e-01  -2.60981340e-02    1.32e-02     4.44e-16     1.00e-01
  2      4    1.41306084e-01  -2.61310801e-02    2.23e-04     4.44e-16     1.00e-01
  2      5    1.41289418e-01  -2.61310840e-02    5.84e-07     4.44e-16     1.00e-01
------------------------------------------------------------------------------------
  3      1    1.12000790e-01   1.00022812e-01    2.09e-02     2.22e-16     1.00e-02
  3      2    1.00611372e-01   9.26073953e-02    9.48e-03     0.00e+00     1.00e-02
  3      3    9.19645378e-02   8.96284842e-02    7.89e-04     2.22e-16     1.00e-02
  3      4    9.14353183e-02   8.96121286e-02    1.66e-05     2.22e-16     1.00e-02
  3      5    9.14205226e-02   8.96121160e-02    4.04e-08     2.22e-16     1.00e-02
------------------------------------------------------------------------------------
  4      1    8.39135058e-02   8.53100707e-02    8.32e-04     4.44e-16     1.00e-03
  4      2    8.31643195e-02   8.50828451e-02    8.66e-05     0.00e+00     1.00e-03
  4      3    8.30781034e-02   8.50792820e-02    3.03e-07     0.00e+00     1.00e-03
------------------------------------------------------------------------------------
  5      1    8.21955101e-02   8.26145887e-02    1.06e-05     2.22e-16     1.00e-04
  5      2    8.21850519e-02   8.26140650e-02    1.22e-07     2.22e-16     1.00e-04
------------------------------------------------------------------------------------
  6      1    8.20951019e-02   8.21609254e-02    1.04e-07     4.44e-16     1.00e-05
------------------------------------------------------------------------------------
  7      1    8.20860103e-02   8.20948938e-02    1.17e-08     2.22e-16     1.00e-06
------------------------------------------------------------------------------------
  8      1    8.20850987e-02   8.20862184e-02    1.17e-10     2.22e-16     1.00e-07
------------------------------------------------------------------------------------
  9      1    8.20850086e-02   8.20851436e-02    1.16e-12     2.22e-16     1.00e-08

Tiempo de ejecucion                       0.08162208
```

- Podemos ver que conforme va avanzando el método el número de iteraciones dentro de cada ciclo interno va disminuyendo de manera muy importante.

```
An Interior Point Method solving problem: odfits
Problem dimensions: n_x =   10;      m   =   16
                    n_s =   10;      m_e =    6

Parameters:         factor ................. 1.0e-01
                    mu0 .................... 1.0e+00
                    maxbar .................    10
                    Step lengths ........... Separate

Objective   ............... -2.71686667e+03
Barrier     ............... -2.77001532e+03
||c||_2     ...............  1.85816681e+03

 k_ext  k_int       f              fbarrera         ||W||       ||c||         mu
-------------------------------------------------------------------------------------
-
   1      1  -2.23722147e+03  -2.28639483e+03    3.23e+00    7.17e-01    1.00e+00
   1      2  -2.25874546e+03  -2.31048485e+03    2.48e+00    2.05e-13    1.00e+00
   1      3  -2.31760580e+03  -2.37213550e+03    2.78e+00    1.48e-13    1.00e+00
   1      4  -2.36828653e+03  -2.42454643e+03    1.44e+00    3.64e-13    1.00e+00
   1      5  -2.37924035e+03  -2.43613394e+03    9.86e-01    9.10e-14    1.00e+00
   1      6  -2.37999962e+03  -2.43708513e+03    2.92e-01    1.36e-13    1.00e+00
   1      7  -2.37999019e+03  -2.43708978e+03    7.22e-03    1.37e-12    1.00e+00
   1      8  -2.37999026e+03  -2.43708978e+03    1.08e-05    2.27e-13    1.00e+00
   1      9  -2.37999026e+03  -2.43708978e+03    7.69e-10    1.14e-13    1.00e+00
-------------------------------------------------------------------------------------
-
   2      1  -2.38002637e+03  -2.38572976e+03    1.38e-02    2.27e-13    1.00e-01
   2      2  -2.38002640e+03  -2.38572976e+03    1.08e-04    9.10e-14    1.00e-01
   2      3  -2.38002640e+03  -2.38572976e+03    2.66e-08    2.27e-13    1.00e-01
-------------------------------------------------------------------------------------
-
   3      1  -2.38002677e+03  -2.38059704e+03    1.45e-04    2.27e-13    1.00e-02
   3      2  -2.38002677e+03  -2.38059704e+03    1.17e-07    2.27e-13    1.00e-02
-------------------------------------------------------------------------------------
-
   4      1  -2.38002677e+03  -2.38008380e+03    1.48e-06    9.10e-14    1.00e-03
-------------------------------------------------------------------------------------
-
   5      1  -2.38002677e+03  -2.38003248e+03    1.60e-07    9.10e-14    1.00e-04
-------------------------------------------------------------------------------------
-
   6      1  -2.38002677e+03  -2.38002734e+03    1.61e-09    2.27e-13    1.00e-05
-------------------------------------------------------------------------------------
-
   7      1  -2.38002677e+03  -2.38002683e+03    1.60e-11    2.27e-13    1.00e-06
-------------------------------------------------------------------------------------
-
   8      1  -2.38002677e+03  -2.38002678e+03    1.60e-13    9.10e-14    1.00e-07
-------------------------------------------------------------------------------------
-
   9      1  -2.38002677e+03  -2.38002677e+03    1.60e-15    2.27e-13    1.00e-08

 Tiempo de ejecucion                      0.09120340
```

- Además, el intercambio entre $\|W\|$ y $\|c\|$ es mucho menor en las primeras iteraciones que en las finales, esto también se debe al tipo de convergencia que hay al principio del algoritmo y cómo éste se acelera conforme va acercándose a la vecindad del mínimo (conforme se va acercando a la frontera).

```
An Interior Point Method solving problem: cb3
Problem dimensions: n_x =    3;      m   =    3
                    n_s =    3;      m_e =    0

Parameters:         factor ................. 1.0e-01
                    mu0 .................... 1.0e+00
                    maxbar .................   10
                    Step lengths ........... Separate

Objective   ...............  1.00000000e+00
Barrier     ...............  5.60517019e+00
||c||_2     ...............  1.90525589e+01

k_ext  k_int        f               fbarrera          ||W||        ||c||         mu
-------------------------------------------------------------------------------------
   1      1   1.76922135e+00   4.99515089e+00    6.20e+00     4.46e+01     1.00e+00
   1      2   4.38227077e+01   3.36856859e+01    1.14e+02     1.12e+03     1.00e+00
   1      3   4.65562751e+00  -4.58196552e+00    7.49e+02     8.91e+02     1.00e+00
   1      4  -3.58197678e+01  -3.91644561e+01    5.16e+02     5.99e+02     1.00e+00
   1      5   3.34893900e+01   2.81232023e+01    1.89e+02     1.86e+02     1.00e+00
   1      6   1.92565411e+01   1.91677650e+01    1.17e+02     1.19e+02     1.00e+00
   1      7   2.29525092e+01   1.69557704e+01    4.35e+01     3.78e+01     1.00e+00
   1      8  -1.36870507e+00  -1.80150164e+00    1.63e+01     3.10e+01     1.00e+00
   1      9   3.15511633e+00   3.80317850e+00    6.71e+00     8.13e+00     1.00e+00
   1     10   6.34678767e+00   1.04880245e+00    8.82e+00     1.41e+01     1.00e+00
   1     11   5.47479303e+00   3.81258318e+00    3.45e+00     2.14e+00     1.00e+00
   1     12   4.30997305e+00   3.55020743e-01    1.61e+00     2.14e+00     1.00e+00
   1     13   4.99607064e+00   1.61160198e+00    2.11e-01     1.08e-01     1.00e+00
   1     14   4.90005943e+00   1.62434664e+00    2.69e-02     2.63e-02     1.00e+00
   1     15   4.95576880e+00   1.63436949e+00    9.35e-04     1.86e-04     1.00e+00
   1     16   4.95642526e+00   1.63445439e+00    3.53e-06     1.81e-07     1.00e+00
-------------------------------------------------------------------------------------
   2      1   2.28193687e+00   2.71486660e+00    7.73e-02     7.24e-03     1.00e-01
   2      2   2.30921433e+00   2.63689831e+00    1.32e-02     6.15e-03     1.00e-01
   2      3   2.29449171e+00   2.64182811e+00    1.32e-03     9.53e-05     1.00e-01
   2      4   2.29469894e+00   2.64185267e+00    1.33e-05     3.46e-06     1.00e-01
   2      5   2.29468941e+00   2.64185552e+00    5.00e-08     3.00e-10     1.00e-01
-------------------------------------------------------------------------------------
   3      1   2.03052789e+00   2.13501045e+00    1.01e-02     1.90e-03     1.00e-02
   3      2   2.03146357e+00   2.13226904e+00    1.67e-03     5.16e-04     1.00e-02
   3      3   2.02995658e+00   2.13247763e+00    1.29e-04     1.27e-05     1.00e-02
   3      4   2.02985212e+00   2.13248504e+00    1.59e-06     1.02e-07     1.00e-02
-------------------------------------------------------------------------------------
   4      1   2.00309906e+00   2.02010850e+00    3.81e-04     1.31e-04     1.00e-03
   4      2   2.00301688e+00   2.02014088e+00    1.77e-05     1.31e-06     1.00e-03
   4      3   2.00299828e+00   2.02014156e+00    7.57e-08     2.35e-09     1.00e-03
-------------------------------------------------------------------------------------
   5      1   2.00030149e+00   2.00270348e+00    5.05e-06     1.93e-06     1.00e-04
-------------------------------------------------------------------------------------
   6      1   2.00003033e+00   2.00033955e+00    5.77e-07     2.44e-08     1.00e-05
-------------------------------------------------------------------------------------
   7      1   2.00000300e+00   2.00004086e+00    6.47e-09     2.56e-10     1.00e-06
-------------------------------------------------------------------------------------
   8      1   2.00000030e+00   2.00000478e+00    6.22e-11     2.08e-12     1.00e-07
-------------------------------------------------------------------------------------
   9      1   2.00000003e+00   2.00000055e+00    5.64e-13     2.00e-14     1.00e-08
```

- Por último, es importante hacer notar que la función de barrera va tendiendo a la función original f conforme el valor de μ va disminuyendo.

Conclusiones

- Aunque por algún tiempo los métodos de puntos interiores perdieron terreno contra otro tipo de algoritmos como lo es el PCS, actualmente compiten muy bien contra cualquier algoritmo aunque su proceso sea diferente a los de los demás (huir de la frontera en lugar de seguirla).

- De una base de datos grande varios problemas no pudieron correr completamente (por inercia incorrecta la mayoría), sin embargo este tipo de problemas pueden ser tratados como en PCS + RC para impulsar al algortimo a tener una convergencia global.

- Aún en problemas de gran escala, los métodos MPI pudieron dar resultados haciendo a veces menos de 23 iteraciones para problemas de tamaos en las decenas de miles. Eso es un gran poder de resolución para un algoritmo tan sencillo como lo es el de barrera logarítmica.

- El siguiente paso a seguir podría ser incluir algún tipo de separación en el programa que pudiera permitir usar MPI si el problema lo indica o PCS +GC con regiones de confianza si la curvatura no es correcta.

problema	n	m	f*	iter	iterf	||c||	CPU(s)
airport	294	210	4.7953e+04	19	33	1.7764e-15	4.4904e-01
aljazzaf	6	4	7.5015e+01	0	4	1.0001e+04	3.7974e-02
alsotame	6	5	8.2085e-02	15	29	2.2204e-16	8.1802e-02
batch	195	161	1.6692e+03	0	8	8.0494e+05	6.7065e-02
bigbank	3139	2180	-5.1944e+03	0	4	1.0000e-01	9.1614e-02
cantilvr	11	6	3.1200e-01	0	4	1.2410e+02	3.7127e-02
cb2	6	3	1.0000e+00	0	4	1.9100e+01	3.3821e-02
cb3	6	3	2.0000e+00	23	37	1.9991e-14	5.7460e-02
chaconn1	6	3	1.9522e+00	16	30	6.5919e-14	4.7731e-02
chaconn2	6	3	2.0000e+00	16	30	1.9991e-14	4.7291e-02
chenhark	10000	5000	2.1495e+03	0	4	0.0000e+00	4.9742e-01
clnlbeam	19998	9998	3.5000e+02	5	10	2.2204e-16	1.2046e+01
concon	20	16	-6.2308e+03	7	21	1.9645e-10	5.2878e-02
congigmz	8	5	2.0000e+00	0	4	1.4100e+01	3.6426e-02
csfi2	13	10	5.0000e-01	0	4	2.3424e+02	3.7567e-02
cvxbqp1	30000	20000	2.2502e+06	7	21	1.7764e-15	1.2777e+02
dipigri	11	4	7.1400e+02	0	4	0.0000e+00	3.5725e-02
dnieper	169	136	-5.1131e+02	0	6	1.2143e+01	4.8547e-02
eg3	500	400	0.0000e+00	0	4	1.1000e+00	5.0423e-02
eigmaxb	303	303	-9.6744e-04	6	20	2.2204e-16	7.3985e-02
eigminb	303	303	9.6744e-04	6	20	2.2204e-16	8.3841e-02
expfitb	106	101	1.4990e+02	0	4	1.0000e-01	3.6579e-02
explin	30000	20000	1.0000e+01	0	4	1.0000e-01	7.2791e+00
explin2	30000	20000	1.0000e+01	0	4	1.0000e-01	6.7137e+00
expquad	10020	20	1.0000e+01	0	4	1.0000e-01	2.1563e-01
gausselm	4470	4215	-1.0000e+00	0	4	1.0000e-01	2.2020e-01
haifas	16	9	-4.5000e-01	16	30	2.1211e-12	5.2294e-02
hanging	468	180	-6.2018e+02	25	39	6.4384e-12	1.4424e-01
himmelp6	10	8	1.7247e+00	0	4	3.5100e+01	3.6924e-02
hong	12	9	3.8960e+01	0	5	1.0000e+00	3.4733e-02
hubfit	4	2	1.6894e-02	17	31	0.0000e+00	4.9740e-02
jnlbrng1	30258	15129	4.5248e+01	0	4	1.1000e+00	4.4544e+00
jnlbrng2	30258	15129	4.0647e+01	0	4	1.1000e+00	4.5893e+00
jnlbrnga	30258	15129	-2.6845e-01	27	41	2.7756e-17	1.8034e+02
jnlbrngb	30258	15129	-6.2805e+00	25	39	1.3878e-17	1.5383e+02
loadbal	93	73	1.5467e+00	0	4	1.0000e-01	3.9760e-02
madsschj	239	158	0.0000e+00	0	4	9.8910e+02	4.3176e-02
makela2	6	3	0.0000e+00	0	4	5.6100e+01	2.9176e-02
makela3	41	20	2.0000e-07	15	29	8.4703e-22	6.0577e-02
mccormck	30000	20000	-9.1327e+03	13	27	8.8818e-16	1.8556e+02
mconcon	20	16	-6.2308e+03	7	21	1.9645e-10	5.8132e-02
mifflin1	5	2	-1.0000e+00	4	18	1.0527e-17	4.3237e-02
mifflin2	5	2	0.0000e+00	0	4	4.8500e+00	3.5763e-02
minmaxbd	25	20	0.0000e+00	0	4	8.2238e+02	3.5374e-02
minperm	2326	2246	0.0000e+00	0	5	1.0000e+00	3.2420e-01
ncvxbqp1	30000	20000	-4.9222e+07	0	4	2.7756e-17	6.9943e+00
ncvxbqp2	30000	20000	-2.8125e+07	0	4	2.7756e-17	6.9456e+00
ncvxbqp3	30000	20000	7.0341e+06	0	4	2.7756e-17	7.0218e+00
nobndtor	28800	14400	-4.4051e-01	23	37	1.1102e-16	1.1977e+02
nonscomp	30000	20000	1.8775e-04	26	40	1.4211e-14	2.4684e+02
obstclae	45387	30258	1.9010e+00	27	41	2.2737e-13	1.1038e+03
obstclbm	45387	30258	7.2802e+00	26	40	2.2204e-16	1.0161e+03
odfits	20	16	-2.3800e+03	10	24	2.2737e-13	9.1468e-02
optcntrl	59	50	5.5000e+02	693	707	1.7770e-15	5.4138e-01
optprloc	119	89	0.0000e+00	0	4	6.6335e+02	3.2590e-02