Developping a preprocessor for quadratic programs in the framework of SQP

Ph. Toint (with N. Gould)

FUNDP, Namur

The software context

– a successor to LANCELOT :

$\longrightarrow$ $\boxed{\text{GALAHAD}}$

– necessity to handle linear constraints better (than LANCELOT)

– availability of efficient large-scale QP solvers (see Nick's talk)

– good experience of colleagues (packages by Gill, Nocedal, . . .)

$\Rightarrow$ Sequential Quadratic Programming

GALAHAD general features

Our current intentions :

– general smooth (un)constrained nonconvex optimization
  + nonlinear equations
  and nonlinear least-squares
– SQP approach (Sℓ_1QP or filter)
– SIF (CUTE) and AMPL input format
– code aimed at both "small-dense" and "large-sparse" problems
– coordinate or sparse row-wise storage
– includes preprocessing of QPs for
  – analysis of linear constraints
  – a priori variable reordering for efficient QP solution (fast matrix-vector products, . . .)

QP vs LP preprocessing (1)

The problem :

$$\min_x f + g^T x + \tfrac{1}{2} x^T H x$$

subject to

$$x_l \leq x \leq x_u \quad \to z \text{ dual variables}$$

$$c_l \leq Ax \leq c_u \quad \to y \text{ multipliers}$$

but also (when available)

$$z_l \leq z \leq z_u$$

$$y_l \leq y \leq y_u$$

– bounds may be $-\infty$ or ∞

– lower and upper bounds may be equal

QP vs LP preprocessing (2)

(Local) optimality conditions :

$$g + Hx_* - A^T y_* - z_* = 0$$

$$x_l \leq x_* \leq x_u$$

$$c_l \leq Ax_* \leq c_u$$

$$[z_*]_j \begin{cases} \geq 0 & \text{when } [x_{l,*}]_j = [x_*]_j \\ = 0 & \text{when } [x_{l,*}]_j < [x_*]_j < [x_{u,*}]_j \\ \leq 0 & \text{when } [x_*]_j = [x_{u,*}]_j \end{cases}$$

$$[y_*]_i \begin{cases} \geq 0 & \text{when } [c_{l,*}]_i = [Ax_*]_i \\ = 0 & \text{when } [c_{l,*}]_i < [Ax_*]_i < [c_{u,*}]_i \\ \leq 0 & \text{when } [Ax_*]_i = [c_{u,*}]_i \end{cases}$$

$\Rightarrow$ preprocessing depends on H

QP vs LP preprocessing (3)

- much in common with LP case (Andersen and Andersen, Gondzio, . . .)
  - implications between variables and bounds
  - analysis of primal constraints
    - empty, singleton, doubleton and linearly dependent rows
    - constraint bound tightening/fixing
    - row sparsification
- but substantial differences too
  - linearly unconstrained variables (linear, quadratic)
  - analysis of dual constraints
    - empty and singleton columns
    - occurence of Hx in implied bounds
    - tightening and fixing of primal, dual variables and multipliers mixed up

An example : nonlinear singleton columns

Singleton column in A corresponding to a diagonal column in H

$$g_j + H_{jj}x_j - A_{ij}y_i = z_j$$

If $[x_u]_j = +\infty$, then $z_j \geq 0$

$$g_j + H_{jj}x_j - A_{ij}y_i \geq 0$$

for x_j	$A_{ij} > 0$	$A_{ij} < 0$
$H_{jj} > 0$	$x_j \geq -\dfrac{g_j - A_{ij}[y_l]_i}{H_{jj}}$	$x_j \geq -\dfrac{g_j - A_{ij}[y_u]_i}{H_{jj}}$
$H_{jj} < 0$	$x_j \leq -\dfrac{g_j - A_{ij}[y_l]_i}{H_{jj}}$	$x_j \leq -\dfrac{g_j - A_{ij}[y_u]_i}{H_{jj}}$

for y_i	$A_{ij} > 0$	$A_{ij} < 0$
$H_{jj} > 0$	$y_i \leq \dfrac{g_j + H_{jj}[x_u]_j}{A_{ij}}$	$y_i \geq \dfrac{g_j + H_{jj}[x_u]_j}{A_{ij}}$
$H_{jj} < 0$	$y_i \leq \dfrac{g_j + H_{jj}[x_l]_j}{A_{ij}}$	$y_i \geq \dfrac{g_j + H_{jj}[x_l]_j}{A_{ij}}$

+ symmetric bounds when $[x_l]_j = -\infty$ ...

QP preprocessing in the SQP context (1)

repeated application on successive QPs

- recognize constant and variable data
  - in objective function
  - in constraints (linear vs nonlinear)
- need to handle (as much as possible) structural data
- fast reapplication of previously determined transformations
- special tests for pure LP ($H = 0$) or pure linearly unconstrained ($A = 0$) problems

QP preprocessing in the SQP context (2)

Call sequence :

setup

↓

first analysis

↓

SQP iteration

nonlinear data modifications

↓

re-analysis + permutation

↓

QP solution

↓

solution recovery

↓

cleanup

Numerical experience

...not available yet, sorry ! !

Expectations :

– significant gain in efficiency
(size, ordering)

– useful mostly on OR applications

– strong influence of sparsity pattern of H

Thank you for your attention