Case Study: Optimization for a client with large-scale constrained problems

Jeremy Walton, Jan Fiala, Ken Kubat
The Numerical Algorithms Group (NAG), 801, Warrensville Road, Suite 185, Lisle, IL 60531
jeremy.walton@nag.co.uk, jan@nag.co.uk, kkubat@nag.com. www.nag.com

1. Introduction
Our client provides an analytical service to its customers in helping them with, for example:
• price promotion - fitting to a model used for the planning of retail price promotions throughout the year, and
• marketing mix - maximising the income from a product by combining product specification, distribution channel and promotional tactics. These problems (and others in business analytics) use optimization to find the best option from a set of alternatives. NAG has a wealth of experience in this area, and has produced several solvers which usually outperform user-written solutions (see figures below).

2. The problem
An all-purpose optimizer doesn’t currently exist, so it’s important to choose the most efficient method for the problem at hand. The client problem is the minimization of
$$c^T x + \frac{1}{2} x^T H x$$
where x and c are column vectors of length n, and H is a symmetric n x n matrix. This is a so-called quadratic programming (QP) problem. In this case, H is also diagonal: 
$$H_{ij} \neq 0 \text{ for } i = j; 0 \text{ otherwise}$$

The minimization is to be performed subject to the set of linear constraints:
$$l \leq \begin{bmatrix} x \\ Ax \end{bmatrix} \leq u$$
where the dimensions of the column vectors l and u and of the matrix A are determined by the number of constraints.

3. Initial solution
The client asked us to speed up the solution times for their problems (which ranged in size from n = 1,600 to 65,000). We first determined that they were using an optimizer that was more general than was necessary; replacing it with a different NAG optimizer which was specifically directed towards QP problems reduced the solution time from more than five hours to 148 minutes.

4. Choosing a new optimizer
We recognized that the diagonal structure of the matrix H (see figure below) made the problem well-suited for the use of a new optimizer (which is not yet part of the NAG Library). Further software engineering was required to create a tailored interface which was compatible with the client’s environment, thereby allowing the optimizer to be plugged into their existing application structure.

5. Constraint reduction
Our analysis revealed that some of the constraints in the original problem were linear combinations of each other. This made the constraint matrix A rank deficient, which in turn introduced numerical instabilities into the new optimizer. These increased the time to solution, and made it unreliable. Removing the redundant constraints made the use of the new optimizer feasible (see figures at right). We are continuing to support the optimizer, which is being heavily used by the client.