AN ALGORITHM FOR LARGE-SCALE LINEARLY CONSTRAINED
NONDIFFERENTIABLE CONVEX MINIMIZATION\(^1\)

Krzysztof C. Kiwiel
Systems Research Institute, Polish Academy of Sciences
Newelska 6, 01-447 Warsaw, Poland

Abstract. A partial proximal bundle method is given for solving a large convex program obtained by augmenting the objective of a linear program with a non-smooth convex function depending on relatively few variables. In contrast to other methods, the algorithm's direction finding subproblems can be solved efficiently by the existing software for large-scale smooth optimization.

1. INTRODUCTION

We present a method for solving the following problem

\[
\begin{align*}
\text{minimize } & \quad f(x) := c^T x_1 + f_n(x_n) \\
\text{over all } & \quad x = (x_1, x_n) \in \mathbb{R}^M \times \mathbb{R}^N \\
\text{in } & \quad S_h = \{ (x_1, x_n) : Ax_1 + Bx_n \leq b \},
\end{align*}
\]

where \( c \in \mathbb{R}^M, A \in \mathbb{R}^{P \times M}, B \in \mathbb{R}^{P \times N} \) and \( b \in \mathbb{R}^P \) are fixed, and \( f_n : \mathbb{R}^N \to \mathbb{R} \) is a (possibly nondifferentiable) convex function. We suppose that the feasible set \( S_h \neq \emptyset \) is bounded, and that at each \( y = (y_1, y_n) \in S_h \) we can compute \( f_n(y_n) \) and an arbitrary subgradient \( g(y_n) \in \partial f_n(y_n) \) that defines the linearization of \( f_n \) at \( y_n \)

\[
\tilde{f}_n(x_n; y_n) = f_n(y_n) + \langle g(y_n), x_n - y_n \rangle \quad \forall \ x_n \in \mathbb{R}^N.
\]

Problems of the form (1.1) are often encountered in practice, especially as deterministic equivalents of two-stage stochastic programming problems [EW]. In many applications the number \( M \) of "linear" variables (in \( x_1 \)) is much larger than the number \( N \) of "nonlinear" variables (in \( x_n \)), and the constraint matrices \( A \) and \( B \) are sparse (have relatively few nonzero entries). In such cases problem (1.1) can be solved by the existing algorithms for large-scale optimization (e.g. MINOS [MS]) if \( f_n \) is differentiable.

\(^1\) Research sponsored by Project CPBP.02.15.
The method presented in this paper modifies one given in [K2] to exploit the special structure of (1.1). It is a feasible point method of descent in the sense of generating a sequence \( \{x^k\}_{k=1}^{\infty} \subseteq S_h \) with \( f(x^{k+1}) < f(x^k) \) if \( x^{k+1} \neq x^k \), where \( x^1 \subseteq S_h \) is a given starting point. Additional trial points \( \{y^k\} \subseteq S_h \) are used for evaluating linearizations of \( f_n \). At the \( k \)-th iteration the method employs the following piecewise linear (polyhedral) approximations to \( f \) and \( f_n \)

\[
\hat{f}_n(x_n) = \max\{ \bar{f}_n(x_n; y_n^j) : j \in J^k \} \quad \text{for all } x_n
\]

\[
\hat{f}_n(x_n) = c^T x_n + \hat{f}_n(x_n) \quad \text{for } x = (x_1, x_n),
\]

with \( J^k \subseteq \{1, \ldots, k\}, |J^k| \leq N+2 \). The next trial point is

\[
y_{n+1} = \arg\min \{ c^T x_n + \hat{f}_n(x_n) + u_k^k |x_n - x_n^k|^2 / 2 : x \in S_h \},
\]

where \( u_k^k > 0 \) is chosen by safeguarded quadratic interpolation to estimate the curvature of \( f_n \) between \( y_n^k \) and \( x_n^{k-1} \). A serious step from \( x^k \) to \( x^{k+1} = y_{n+1} \) occurs if \( y_{n+1} \) is significantly better than \( x^k \) in the sense that

\[
f(y_{n+1}^{k+1}) \leq f(x^k) + m_L v^k,
\]

where \( m_L \in (0,0.5) \) is a parameter and

\[
v^k = \hat{f}_n(y_{n+1}^{k+1}) - f(x^k)
\]

is the predicted descent (if \( v^k \geq 0 \) the algorithm may stop with an optimal \( x^k \); see below). Otherwise, a null step \( x^{k+1} = x^k \) improves the next polyhedral approximation \( \hat{f}_n \) with \( \bar{f}_n \).

We show that the method is globally convergent under no additional assumptions. We may add that the method will find a solution in a finite number of iterations if \( f_n \) is polyhedral and either \( m_L = 1 \) in (1.4) or certain technical conditions are satisfied (see [K3] for details).

The method is implementable in the sense of requiring bounded storage and a finite number of simple operations per iteration. For problems with large sparse matrices \( A \) and \( B \) and relatively few nonlinear variables (\( N \ll M \)), the method can use MINOS [MS] for solving quadratic programming (QP) equivalents of (1.3) (see (2.1) below). In fact, an efficient implementation should exploit the fact that consecutive subproblems retain the original constraints of (1.1), differ only in a few auxiliary constraints on \( x_n \), have simple terms quadratic in \( x_n \) as the only nonlinearities in their objectives, etc. It would be interesting to perform the necessary numerical experimentation, but we have not had the means to do so.

In contrast to [K2] we do not add \( u^k |x_1 - x_1^{k+1}|^2 / 2 \) to the objective of (1.3); this would make all the variables "nonlinear" in (1.3) and, hence, its solu-