Optimization Methods and Software
Vol. 00, No. 00, July 2010, 1–8

RESEARCH ARTICLE

A strategy of finding an initial active set for inequality constrained quadratic programming problems

Jungho Lee†

Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN, USA

(*Received July 2010*)

It is well known that an active set method can often be slow due to a poor guess of the optimal active set. In this paper, we provide an efficient strategy of finding an initial active set and an initial guess for a quadratic programming problem with inequality constraints. We use a combination of this strategy with a primal-dual active set method as a nonsmooth Newton method [2] to solve a model problem.

Keywords: quadratic programming; inequality constraints; initial active set; primal-dual active set method; nonsmooth Newton method; domain decomposition methods

AMS Subject Classification: 49M29; 65N55; 90C33; 90C59

1. Introduction

The idea of an active set method, or a working set method, is to reduce an inequality constrained problem to a sequence of equality constrained problems; see [1, Chapter 5], [5]. It starts with an initial guess as to which *face* of the boundary of the feasible region the optimal solution lies on, or equivalently, the set of constraints that are satisfied as equalities at the solution. Such a guess is called an active set, or a working set. In the rest of this paper we are going to use the terminology an *active set* and an *active set method*.

An active set method has a finite termination property but also has a very pessimistic upper bound on the number of iterations needed to reach the correct solution; this is due to the fact that the number of possible active sets is $2^{|\mathcal{I}|}$, where $|\mathcal{I}|$ is the number of inequality constraints. This is a phenomenon known as the *combinatorial difficulty* [5, Chapter 15].

On the other hand, it is possible to cast an inequality constrained quadratic programming problem as a nonlinear equation so that a semismooth Newton method can be used; see [2] and the references therein. A semismooth Newton method has a superlinear convergence property, but such a convergence is guaranteed only when the initial guess is sufficiently close to the right solution. One can easily see that finding a good initial guess, again, becomes an issue of critical importance.

In this paper, we propose a strategy for finding an initial active set and an initial

†Computer Science and Mathematics Division, Oak Ridge National Laboratory, PO BOX 2008 MS6164, Oak Ridge, TN 37831-6164, USA. Telephone: 1-865-574-3653, Fax: 1-865-241-4811, Email: jungho@ornl.gov

ISSN: 1055-6788 print/ISSN 1029-4937 online

DOI: 10.1080/1055678xxxxxxxxxxxxx
http://www.informaworld.com

guess for a quadratic programming problem with inequality constraints. This work originated in the Ph.D. thesis of the author [3] concerning domain decomposition methods for contact problems. The rest of the paper is organised as follows. In Section 2, we provide the primal and the dual formulations of an inequality constrained quadratic programming problem that we consider in this paper. We also outline our strategy of finding an initial active set and an initial guess. In Section 3, we provide the details of the strategy. In Section 4, we describe a primal-dual active set method as a nonsmooth Newton method [2]. In Section 5, we provide the numerical results of a model problem using a combination of our strategy of finding an initial active set and an initial guess and the primal-dual active set method [4]. We provide a few concluding remarks in Section 6.

1.1. *Some notation*

We adopt the notation similar to the one established in [1, Chapter 1]. In the rest of the paper, the ith component of a vector $\boldsymbol{v} \in \mathbb{R}^n$ is denoted by $[\boldsymbol{v}]_i$. The relations between two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ are defined component-wise; that is, $\boldsymbol{u} = \boldsymbol{v}$ and $\boldsymbol{u} \leq \boldsymbol{v}$ are equivalent to $[\boldsymbol{u}]_i = [\boldsymbol{v}]_i, \forall i$ and $[\boldsymbol{u}]_i \leq [\boldsymbol{v}]_i, \forall i$, respectively. For given $\boldsymbol{u} \in \mathbb{R}^n$, the vector $\boldsymbol{u}^+ \in \mathbb{R}^n$ is defined by $[\boldsymbol{u}^+]_i = \max([\boldsymbol{u}]_i, 0), \forall i$. The zero vector in $\mathbb{R}^n$ is denoted by $\boldsymbol{o}_n$; when the dimension of the zero vector is clear from the context, we drop the subscript n.

Analogously, the (i,j)th component of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is denoted by $[\mathbf{A}]_{ij}$. The zero matrix in $\mathbb{R}^{m \times n}$ is denoted by $\mathbf{O}_{mn}$; when the dimension of the matrix is clear from the context, we drop the subscript mn.

2. Primal and Dual formulations of a quadratic programming problem with inequality constraints

We consider a quadratic programming problem of the following form, where $\mathbf{A} \in \mathbb{R}^{n \times n}, \boldsymbol{b} \in \mathbb{R}^n, \mathbf{B} \in \mathbb{R}^{m \times n}, m \leq n$, and $\mathbf{A}$ is symmetric and positive-definite:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \frac{1}{2} \boldsymbol{x}^T \mathbf{A} \boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x}, \quad \text{with} \quad \mathbf{B}\boldsymbol{x} \leq \boldsymbol{o}. \tag{1}$$

The dual form of (1) is as follows (see, for instance, [1]):

$$\min_{\boldsymbol{\lambda} \in \mathbb{R}^m} \frac{1}{2} \boldsymbol{\lambda}^T \mathbf{B} \mathbf{A}^{-1} \mathbf{B}^T \boldsymbol{\lambda} - \boldsymbol{d}^T \boldsymbol{\lambda}, \quad \text{with} \quad \boldsymbol{\lambda} \geq \boldsymbol{o}, \tag{2}$$

where $\boldsymbol{d} = \mathbf{B}\mathbf{A}^{-1}\boldsymbol{b}$. The dual form (2) has so-called *box constraints*, i.e., constraints of the form $-\infty \leq a \leq [\boldsymbol{\lambda}]_i \leq b \leq \infty$, whereas the primal form (1) does not.

We propose the following strategy to determine an initial active set for the problem (1):

Algorithm 1. Our strategy of finding an initial active set.

(1) Solve the *unconstrained* version of (2) with a preconditioned conjugate gradient (PCG) method, using a zero initial guess.
(2) Apply a *projection-like* operator, $\hat{P}$, to the resulting unconstrained solution, $\boldsymbol{\lambda}^*$.
(3) Define the initial active set as the set of constraints satisfied by $\hat{P}\boldsymbol{\lambda}^*$.

3. Details of the strategy of finding an initial active set/guess

We first solve the unconstrained version of (2):

$$\min_{\boldsymbol{\lambda}\in\mathbb{R}^m} \frac{1}{2}\boldsymbol{\lambda}^T\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T\boldsymbol{\lambda} - \boldsymbol{d}^T\boldsymbol{\lambda}, \tag{3}$$

and denote the solution of (3) by $\boldsymbol{\lambda}^*$. In Figure 1, we illustrate the projection of $\boldsymbol{\lambda}^*$ in the original coordinate system with the standard basis $\{\boldsymbol{e}^i\}_{i=1}^m$ and the transformed coordinate system defined by $\{\bar{\boldsymbol{e}}^i : \mathbf{M}^{-1/2}\bar{\boldsymbol{e}}^i = \boldsymbol{e}^i\}_{i=1}^m$, where $\mathbf{M} := \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$, in two dimensions. The concentric ellipses on the left in Figure 1 indicate the level sets of $f(\boldsymbol{\lambda}) := \frac{1}{2}\boldsymbol{\lambda}^T\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T\boldsymbol{\lambda} - \boldsymbol{d}^T\boldsymbol{\lambda}$, whereas the concentric circles on the right in Figure 1 indicate the level sets of the transformed function

$$\bar{f}(\bar{\boldsymbol{\lambda}}) := \frac{1}{2}\bar{\boldsymbol{\lambda}}^T\mathbf{M}^{-1/2}\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T\mathbf{M}^{-1/2}\bar{\boldsymbol{\lambda}} - \boldsymbol{d}^T\mathbf{M}^{-1/2}\bar{\boldsymbol{\lambda}} = \frac{1}{2}\bar{\boldsymbol{\lambda}}^T\bar{\boldsymbol{\lambda}} - \boldsymbol{d}^T\mathbf{M}^{-1/2}\bar{\boldsymbol{\lambda}}.$$

The feasible region $\Omega_B := \{\boldsymbol{\lambda} : \boldsymbol{\lambda} \geq \boldsymbol{o}\}$ has been transformed into $\{\bar{\boldsymbol{\lambda}} : \mathbf{M}^{-1/2}\bar{\boldsymbol{\lambda}} \geq \boldsymbol{o}\}$.

We make the following key observation: whereas the projection of $\boldsymbol{\lambda}^*$ onto Ω_B in the original coordinate system does not necessarily coincide with $\tilde{\boldsymbol{\lambda}}$, the minimiser of the inequality constrained problem (2), the projection of $\boldsymbol{\lambda}^*$ onto $\{\bar{\boldsymbol{\lambda}} : \mathbf{M}^{-1/2}\bar{\boldsymbol{\lambda}} \geq \boldsymbol{o}\}$ in the transformed coordinate system coincides with $\tilde{\boldsymbol{\lambda}}$. In practice the preconditioner will not be equal to the system matrix $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ and thus we cannot expect this to happen, but we can still expect the projection of $\boldsymbol{\lambda}^*$ in the transformed coordinate system to be a better approximation of $\tilde{\boldsymbol{\lambda}}$ than the projection in the original coordinate system.

We now explain what we mean by a *projection-like* operator $\hat{P}$. Note that $\Omega_B = \{\boldsymbol{\lambda} : \langle\boldsymbol{\lambda}, \boldsymbol{e}^i\rangle \geq 0, \forall i\}$ and the orthogonal projection of a vector $\boldsymbol{v} = \sum_i v_i\boldsymbol{e}^i$ onto Ω_B can be written as $\sum_i \max(\langle\boldsymbol{v}, \boldsymbol{e}^i\rangle, 0)\boldsymbol{e}^i$. Motivated by this, letting $\mathbf{M}^{-1/2}\boldsymbol{e}^i_{\text{new}} := \boldsymbol{e}^i$, we define $\tilde{P}\boldsymbol{w}$ where $\boldsymbol{w} = \sum_i w_i\boldsymbol{e}^i_{\text{new}}$ as

$$\tilde{P}\boldsymbol{w} := \sum_i \max\left(\langle\boldsymbol{w}, \frac{\boldsymbol{e}^i_{\text{new}}}{||\boldsymbol{e}^i_{\text{new}}||}\rangle, 0\right)\frac{\boldsymbol{e}^i_{\text{new}}}{||\boldsymbol{e}^i_{\text{new}}||}.$$

Let $\boldsymbol{u}_{\text{old}} = \boldsymbol{\lambda}^*$ and let $\boldsymbol{u}_{\text{new}}$ be defined by the relation

$$\mathbf{M}^{-1/2}\boldsymbol{u}_{\text{new}} = \boldsymbol{u}_{\text{old}}. \tag{4}$$

Recalling the assumption that $\boldsymbol{\lambda}^*$ was obtained by a PCG method with a zero initial guess (Algorithm 1), there exists a vector $\boldsymbol{u}$ such that

$$\boldsymbol{u}_{\text{old}} = \boldsymbol{\lambda}^* = \mathbf{M}^{-1}\boldsymbol{u}. \tag{5}$$

This $\boldsymbol{u}$ can be obtained by adding just a few lines to the original PCG algorithm and without any additional computational cost.

Our operator $\hat{P}$ is defined as follows: given $\boldsymbol{u}_{\text{old}}$, we apply $\tilde{P}$ to the corresponding $\boldsymbol{u}_{\text{new}}$. We then express the resulting vector in terms of the standard basis by left-multiplying it with $\mathbf{M}^{-1/2}$:

$$\mathbf{M}^{-1/2}\sum_{i=1}\frac{\max(\langle\boldsymbol{u}_{\text{new}}, \boldsymbol{e}^i_{\text{new}}\rangle, 0)}{\langle\boldsymbol{e}^i_{\text{new}}, \boldsymbol{e}^i_{\text{new}}\rangle}\boldsymbol{e}^i_{\text{new}}$$

$$= \mathbf{M}^{-1/2} \sum_{i=1} \frac{\max(\langle \mathbf{M}^{1/2}\boldsymbol{u}_{\text{old}}, \mathbf{M}^{1/2}\boldsymbol{e}^i_{\text{old}}\rangle, 0)}{\langle \mathbf{M}^{1/2}\boldsymbol{e}^i_{\text{old}}, \mathbf{M}^{1/2}\boldsymbol{e}^i_{\text{old}}\rangle} \mathbf{M}^{1/2}\boldsymbol{e}^i_{\text{old}}$$
$$= \sum_{i=1} \frac{\max(\langle \mathbf{M}\boldsymbol{u}_{\text{old}}, \boldsymbol{e}^i_{\text{old}}\rangle, 0)}{[\mathbf{M}]_{ii}} \boldsymbol{e}^i_{\text{old}}$$
$$= \sum_{i=1} \frac{\max(\langle \boldsymbol{u}, \boldsymbol{e}^i_{\text{old}}\rangle, 0)}{[\mathbf{M}]_{ii}} \boldsymbol{e}^i_{\text{old}}. \tag{6}$$

Noting that the entries of $\mathbf{M}^{-1}$ are easily available while those of $\mathbf{M}$ are not, we replace $1/[\mathbf{M}]_{ii}$ of (6) by $[\mathbf{M}^{-1}]_{ii}$:

$$\sum_{i=1} \frac{\max(\langle \boldsymbol{u}, \boldsymbol{e}^i_{\text{old}}\rangle, 0)}{[\mathbf{M}]_{ii}} \boldsymbol{e}^i_{\text{old}} \approx \langle [\mathbf{M}^{-1}]_{diag}, \boldsymbol{u}^+ \rangle =: \hat{\boldsymbol{\lambda}} =: \hat{P}\boldsymbol{\lambda}^*, \tag{7}$$

where $\mathbf{M}^{-1}_{diag} \in \mathbb{R}^m$ is a vector consisting of the diagonal elements of $\mathbf{M}^{-1}$.

We recall the KKT conditions for (1), which are satisfied by an optimal pair $(\boldsymbol{x}, \boldsymbol{\lambda})$:

$$\begin{aligned} \mathbf{B}\boldsymbol{x} &\le \boldsymbol{o}, \\ \boldsymbol{\lambda} &\ge \boldsymbol{o}, \\ \boldsymbol{\lambda}^T(\mathbf{B}\boldsymbol{x}) &= 0, \\ \mathbf{A}\boldsymbol{x} - \boldsymbol{b} + \mathbf{B}^T\boldsymbol{\lambda} &= \boldsymbol{o}. \end{aligned} \tag{8}$$

The second and the third equations of (8) indicate that $[\boldsymbol{\lambda}]_i > 0$ implies $[\mathbf{B}\boldsymbol{x}]_i = 0$. This motivates us to set

$$\mathcal{I}_0 = \{i : [\hat{\boldsymbol{\lambda}}]_i > 0\}, \quad \boldsymbol{\lambda}^0 = \hat{\boldsymbol{\lambda}}. \tag{9}$$

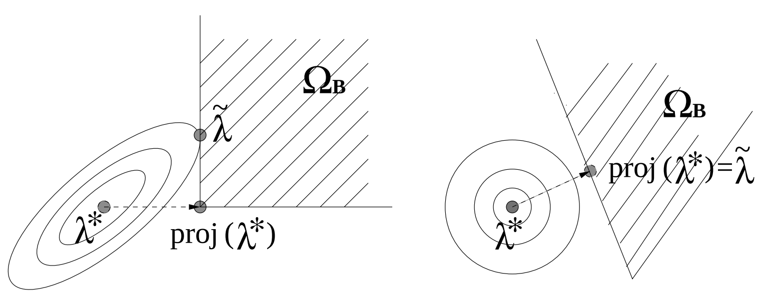

Figure 1. The projection of $\boldsymbol{\lambda}^*$ onto the feasible region in original and transformed coordinates, respectively. When the preconditioner is equal to the inverse of the system matrix (as shown in right), the projection of the solution of the unconstrained problem, $\boldsymbol{\lambda}^*$, onto the feasible region coincides with the solution of the constrained problem, $\tilde{\boldsymbol{\lambda}}$. Therefore we can expect proj$(\boldsymbol{\lambda}^*) \approx \tilde{\boldsymbol{\lambda}}$ with a good preconditioner.

4. A primal-dual active set method as a semismooth Newton method

In this section, we briefly describe a primal-dual active set method; for details, see [2] and the references therein.

Again, we consider the quadratic programming problem with inequality constraints (1), which is equivalent to the following problem

$$\begin{cases} \mathbf{A}\boldsymbol{x} + \mathbf{B}^T\boldsymbol{\lambda} = \boldsymbol{b}, \\ \mathbf{B}\boldsymbol{x} \leq \boldsymbol{o}, \quad \boldsymbol{\lambda} \geq \boldsymbol{o}, \quad \boldsymbol{\lambda}^T\mathbf{B}\boldsymbol{x} = 0. \end{cases} \tag{10}$$

The complementarity condition given in the second line is equivalent to

$$C(\boldsymbol{x}, \boldsymbol{\lambda}, c) := \boldsymbol{\lambda} - \max(\boldsymbol{o}, \boldsymbol{\lambda} + c\mathbf{B}\boldsymbol{x}) = \boldsymbol{o}, \tag{11}$$

for each $c > 0$. Here, the max function is to be understood component-wise. In the following, we will suppress the dependence on c and use the notation $C(\boldsymbol{x}, \boldsymbol{\lambda})$, for the sake of brevity; we will also use $c = 1$ and choose not to investigate different choices of c. The system (10) can thus be expressed as the following nonlinear system of equations:

$$\begin{cases} \mathbf{A}\boldsymbol{x} + \mathbf{B}^T\boldsymbol{\lambda} = \boldsymbol{b}, \\ \quad C(\boldsymbol{x}, \boldsymbol{\lambda}) \quad = \boldsymbol{o}. \end{cases} \tag{12}$$

It follows that a (semismooth) Newton step for the nonlinear system (12) is

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B}_{\mathcal{A}_k} & -\mathbf{I}_{\mathcal{I}_k} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}^k \\ \delta\boldsymbol{\lambda}^k \end{bmatrix} = \begin{bmatrix} f - (\mathbf{A}\boldsymbol{x}^k + \mathbf{B}^T\boldsymbol{\lambda}^k) \\ C(\boldsymbol{x}^k, \boldsymbol{\lambda}^k) \end{bmatrix} \tag{13}$$

and

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \delta\boldsymbol{x}^k, \quad \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \delta\boldsymbol{\lambda}^k, \tag{14}$$

where

$$\mathcal{I}_k = \{i : [\boldsymbol{\lambda}^k + \mathbf{B}\boldsymbol{x}^k]_i \leq 0\}, \quad \mathcal{A}_k = \{i : [\boldsymbol{\lambda}^k + \mathbf{B}\boldsymbol{x}^k]_i > 0\}, \tag{15}$$

and $\mathbf{B}_{\mathcal{A}_k}$ results from replacing the rows of $\mathbf{B}$ for which the index does not belong to $\mathcal{A}_k$ with zero row vectors. The matrix $\mathbf{I}_{\mathcal{I}_k}$ is defined similarly.
We can rewrite the second equation of (13) as follows:

$$[\mathbf{B}\boldsymbol{x}^k]_i = -[\mathbf{B}\boldsymbol{x}^k]_i, \quad \forall i \in \mathcal{A}_k, \quad \text{and} \quad -[\delta\boldsymbol{\lambda}^k]_i = [\boldsymbol{\lambda}^k]_i, \quad i \in \mathcal{I}_k. \tag{16}$$

We also rewrite the first equation:

$$\mathbf{A}\boldsymbol{x}^k + \mathbf{B}^T_{\mathcal{A}_k}(\delta\boldsymbol{\lambda}^k)_{\mathcal{A}_k} + \mathbf{B}^T_{\mathcal{I}_k}(\delta\boldsymbol{\lambda}^k)_{\mathcal{I}_k} = \boldsymbol{b} - (\mathbf{A}\boldsymbol{x}^k + \mathbf{B}^T_{\mathcal{A}_k}(\boldsymbol{\lambda}^k)_{\mathcal{A}_k} + \mathbf{B}^T_{\mathcal{I}_k}(\boldsymbol{\lambda}^k)_{\mathcal{I}_k}), \tag{17}$$

where $(\delta\boldsymbol{\lambda}^k)_{\mathcal{A}_k}$ results from replacing the components of $\delta\boldsymbol{\lambda}^k$ for which the index does not belong to $\mathcal{A}_k$ with zeros; $(\delta\boldsymbol{\lambda}^k)_{\mathcal{I}_k}$, etc., are defined similarly. This equation is equivalent to

$$\mathbf{A}\boldsymbol{x}^k + \mathbf{B}^T_{\mathcal{A}_k}(\delta\boldsymbol{\lambda}^k)_{\mathcal{A}_k} = \boldsymbol{b} - (\mathbf{A}\boldsymbol{x}^k + \mathbf{B}^T_{\mathcal{A}_k}(\boldsymbol{\lambda}^k)_{\mathcal{A}_k}), \tag{18}$$

due to (16).
Consequently, we can rewrite the Newton step defined by (13) and (14) as

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T_{\mathcal{A}_k} \\ \mathbf{B}_{\mathcal{A}_k} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}^k \\ \delta\boldsymbol{\lambda}^k \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} - (\mathbf{A}\boldsymbol{x}^k + \mathbf{B}^T_{\mathcal{A}_k}(\boldsymbol{\lambda}^k)_{\mathcal{A}_k}) \\ -\mathbf{B}_{\mathcal{A}_k}\boldsymbol{x}^k \end{bmatrix} \tag{19}$$

and

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \delta\boldsymbol{x}^k, \quad \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \delta\boldsymbol{\lambda}^k, \quad \text{where} \quad -[\delta\boldsymbol{\lambda}^k]_i = [\boldsymbol{\lambda}^k]_i, \quad i \in \mathcal{I}_k. \tag{20}$$

The following algorithm is simply the primal-dual active set strategy [2] combined with our strategy of finding an initial active set.

Algorithm 2. Primal-Dual active set method combined with our strategy of finding an initial active set.

(1) Choose $\boldsymbol{\lambda}^0$ as described in Section 3. Set $\boldsymbol{x}^0 = \mathbf{0}$. Set $k = 0$.
(2) Set $\mathcal{I}_k = \{i : [\boldsymbol{\lambda}^k + \mathbf{B}\boldsymbol{x}^k]_i \le 0\}$, $\quad \mathcal{A}_k = \{i : [\boldsymbol{\lambda}^k + \mathbf{B}\boldsymbol{x}^k]_i > 0\}$.
(3) Solve

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}_{\mathcal{A}_k}^T \\ \mathbf{B}_{\mathcal{A}_k} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{k+1} \\ \boldsymbol{\lambda}^{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{o} \end{bmatrix} \tag{21}$$

and set $\boldsymbol{\lambda}^{k+1} = \boldsymbol{o}$ on $\mathcal{I}_k$.
(4) Stop if $\mathcal{A}_{k+1} = \mathcal{A}_k$ and $\mathcal{I}_{k+1} = \mathcal{I}_k$. Otherwise return to 2.

5. Numerical examples

In this section, we provide the numerical results of Algorithm 2 applied to the following model problem, taken from [1, Chapter 8]:

$$\begin{aligned} &\min \sum_{i=1}^{2} \left(\frac{1}{2} \int_{\Omega^i} |\nabla u^i|^2 dx - \int_{\Omega^i} f u^i dx \right) \\ &\text{where } u^i \in H^1(\Omega^i), i = 1, 2, \quad \Omega^1 = (0,1) \times (0,1), \Omega^2 = (1,2) \times (0,1), \\ &\qquad u^1 = 0 \quad \text{on} \quad \Gamma_u^1 = \{0\} \times (0,1), \\ &\qquad u^2 - u^1 \ge 0 \quad \text{on} \quad \Gamma_c = \{1\} \times (0,1). \end{aligned} \tag{22}$$

The results we provide here are taken from [4]. As mentioned in [1, Chapter 8], we can view the solution of this problem as the displacement of two membranes, Ω_1 and Ω_2, under a body force. The left edge of the right membrane Ω_2 is not allowed to go below the right edge of the left membrane Ω_1, and the left edge of the left membrane is fixed. This problem is coercive and thus has a unique solution; see [1, Chapter 8] and the references therein.

We use a domain decomposition approach to solve this problem, in particular, a hybrid algorithm described in [4]. The membranes Ω_1 and Ω_2 are decomposed into $N \times N$ subdomains, which in turn are divided into $n \times n$ bilinear elements. The side lengths of a subdomain and of an element are $H := 1/N$ and $h := 1/(Nn)$, respectively.

The finite element discretisation of the problem (22) with a hybrid domain decomposition method [4] is a quadratic programming problem with inequality constraints of the form (1), and we solve it by combining our strategy of finding an initial active set and an initial guess and a primal-dual active set strategy. The results are summarised in Table 5. It is shown that for the combinations of H and h we tried, the number of outer iterations of the primal-dual active set method is at most 2 when combined with our strategy of finding an initial active set and an initial guess.

Table 1. Results: primal-dual active set method + hybrid method. *outer it.* denotes the number of outer iterations of the primal-dual active set method; *inner it.* denotes the number of iterations needed to solve the inner minimisation problems by the PCR (preconditioned conjugate residual) method, until the norm of the residual has been reduced by 10^{-5}, on the active faces identified in the outer iterations. *total it.* denotes the total number of inner iterations.

$N_{sub}(1/H)$	H/h	$N_{dof}(\boldsymbol{\lambda})$	$N_{dof}(total)$	outer it.	inner it.	total it.
16(4)	4	17	561	2	16 16	32
16(4)	8	33	2145	2	20 19	39
16(4)	12	49	4753	2	22 20	42
16(4)	16	65	8385	2	26 24	50
64(8)	4	33	2145	2	18 17	35
64(8)	8	65	8385	1	23	23
64(8)	12	97	18721	1	27	27
64(8)	16	129	33153	1	29	29
144(12)	4	49	4753	1	19	19
144(12)	8	97	18721	2	24 22	46
144(12)	12	145	41905	2	28 24	52
144(12)	16	193	74305	2	30 27	57
256(16)	4	65	8385	1	19	19
256(16)	8	129	33153	1	26	26
256(16)	12	193	74305	1	28	28
256(16)	16	257	131841	1	32	32

6. Concluding remarks

In this paper, we have considered an efficient strategy of finding an initial active set and an initial guess for a quadratic programming problem with inequality constraints. Numerical results from the application of this strategy to a simple model problem were presented, which show that our strategy finds the optimal active set quite accurately for the cases that were considered. The effectiveness of this strategy in more complicated problems, such as contact problems in linear elasticity, remains to be shown.

Acknowledgements

The author would like to thank Olof Widlund and Zdeněk Dostál for carefully reading the manuscript and for their encouragement. This author's work was supported in part by the U.S. Department of Energy under contracts DE-FG02-06ER25718 and DE-FC02-01ER25482 and in part by National Science Foundation grant DMS-0513251. This submission was sponsored by a contractor of the United States Government under contract DE-AC05-00OR22725 with the United States Department of Energy. The United States Government retains, and the publisher, by accepting this submission for publication, acknowledges that the United States Government retains, a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this submission, or allow others to do so, for United States Government purposes.

References

[1] Z. Dostál, *Optimal quadratic programming algorithms. With applications to variational inequalities*, Springer Optimization and Its Applications Vol. 23, Springer, New York, 2009.

[2] M. Hintermüller, K. Ito, and K. Kunisch, *The Primal-Dual Active Set Strategy as a Semismooth Newton Method*, SIAM J. on Optimization 13 (2002), pp. 865–888.

[3] J. Lee, *A Hybrid Domain Decomposition Method and its Applications to Contact Problems*, Ph.D. diss., Courant Institute of Mathematical Sciences, New York University, New York, NY, 2009.
[4] J. Lee, *A hybrid domain decomposition method based on one-level FETI and BDDC algorithms.*, preprint (2010), submitted to SIAM J. Sci. Comput.
[5] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, 2000.