A symmetric Primal-Dual Newton method for Minimizing a Sum of Norms

Knud D. Andersen[1]
Odense University
Campusvej 55, 5230 Odense M, Denmark
E-mail: kda@beta.dou.dk

Edmund Christiansen
Department of Mathematic and Computer Science
Odense University

rev 1.0

April 11, 1995

[1]This work was supported by a Danish SNF Research Studentship. Most of it was carried out while the author was a visiting scholar at the Computer Science Department of Stanford University and it is part of this authors ph.d. thesis

Abstract

In [3, 5, 7] was the dual problem for minimizing a sum of Euclidean Norms described. We will in this paper use this duality theory to develop a symmetric primal-dual Newton method for the minimizing a sum of norms problem (MSN). As in [3, 5] the method is actually develop for a smooth approximation problem to (MSN), using a HAP function. Numerical results are presented for large sparse problems arising in plastic collapse analysis. These results show that the new method is more efficient in measurement of iterations and time than the primal method described in [3, 5].

0.1 Introduction

We consider the following unconstrained convex minimization problem (MSN)

$$(P) \qquad \begin{cases} \min \sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 \\ s.t. \\ \qquad (\boldsymbol{x}, \boldsymbol{r}) \in X \end{cases}$$

where

$$X = \left\{(\boldsymbol{x}, \boldsymbol{r}) \in R^n \times R^{\kappa\nu} \mid \boldsymbol{A}^T\boldsymbol{x} - \boldsymbol{r} = \boldsymbol{b}\right\} \tag{0.1}$$

with $\boldsymbol{A} \in R^{n\times\nu\kappa}$, $\boldsymbol{b} \in R^{\nu\kappa}$ and $\boldsymbol{r} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_\kappa] \in R^{\kappa\nu}$. The function $\sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2$ is convex but is not differentiable at any point $(\boldsymbol{x}, \boldsymbol{r})$ where $\boldsymbol{r}_i = \boldsymbol{0}$ for some $i \in \{1, \ldots, \kappa\}$. The dual is (see [7] and [3])

$$(D) \qquad \begin{cases} \max -\boldsymbol{b}^T\boldsymbol{y} \\ s.t. \\ \qquad \boldsymbol{y} \in Y \end{cases}$$

with

$$Y = \{\boldsymbol{y} \in R^{\kappa\nu} \mid \boldsymbol{A}\boldsymbol{y} = \boldsymbol{0}, \quad \|\boldsymbol{y}_i\|_2 \leq 1, \quad i = 1, \ldots, \kappa\} \tag{0.2}$$

and $\boldsymbol{y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\kappa] \in R^{\kappa\nu}$.

This pair of dual problems has several applications, both in the primal form (P) and in the dual form (D). Examples are problems in multifacility-location, collapse load analysis ([10]), Steiner trees ([7]) and soap dual films (minization of area) ([8]).

As described in [3] the non-smoothness can be handled by applying a primal Newton barrier method to the smooth approximating problem:

$$(P_\epsilon) \qquad \begin{cases} \min \sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}} \\ s.t. \\ \qquad (\boldsymbol{x}, \boldsymbol{r}) \in X \end{cases}$$

This method is numerically stable and efficient, but the number of iterations is larger than for primal-dual interior-point methods in linear programming (see e.g. [11]). We therefore in this paper developed a new primal-dual method for (P) and (D), which in fact uses a number of iterations similar to primal-dual interior-point methods for linear programming.

In [5] it is demonstrated how the linearly constrained (MSN) problem can, preserving sparsity structure, be reduced to the unconstrained case by the use of an exact l_1 penalty function.

A primal-dual method works at both problems simultaneously, and therefore always has both a primal and a dual feasible point, as opposed to the primal method which, in every iteration, only produces a primal feasible point. In the case of a linearly constrained problem it is particularly important to have a dual feasible solution because the dual variables can be used to adjust the weight of the l_1 penalty function.

As for the primal Newton barrier method it is a difficulty that (P) is non-smooth and (D) has non-linear constraints. We replace the pair (P) and (D) with the smooth approximating problem (P_ϵ) and its dual (proved in [3]):

$$(D_\epsilon) \qquad \begin{cases} \max -\boldsymbol{b}^T\boldsymbol{y} + \epsilon \sum_{i=1}^{\kappa}(1 - \|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}} \\ s.t. \\ \qquad \boldsymbol{y} \in Y \end{cases}$$

This pair has the same feasible regions as (P) and (D) respectively. Furthermore, it is proved in [3] that an optimal solution to (P_ϵ) and (D_ϵ) is at least $\kappa\epsilon$-optimal for (P) and (D), measured in duality gap.

(P_ϵ) and (D_ϵ) can be shown to have the following complementarity conditions

$$\boldsymbol{r}_i - \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}} \boldsymbol{y}_i = \boldsymbol{0}$$

We propose a symmetric primal-dual method which applies Newton's method to these complementarity conditions. A related primal-dual Newton method has been proposed in [7].

An outline of the paper is: In Section 0.2 is the theoretical primal-dual method described. In Section 0.3 is the used Newton method described and different techniques for for improving the efficiency of the method is given. Some implementation details are given in 0.4. Numerical results are presented in Section 0.5.

Notation $\boldsymbol{I}_{p\times p}$ is used to present the identity matrix of dimension $p \times p$. $\boldsymbol{r}$ and $\boldsymbol{y}$ are respectively residuals and Lagrange multipliers for the linear system $\boldsymbol{A}^T\boldsymbol{x} - \boldsymbol{r} = \boldsymbol{b}$ where $\boldsymbol{r} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_\kappa] \in R^{\kappa\nu}$, $\boldsymbol{y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\kappa] \in R^{\kappa\nu}$, $\boldsymbol{A} = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_\kappa]$ and $\boldsymbol{b} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\kappa]$. Note that $\boldsymbol{y}_i, \boldsymbol{r}_i \in R^\nu$.

0.2 A symmetric primal-dual Newton method for the MSN problem

In this section we develop a symmetric primal-dual method for the (MSN)problem. Let (P_ϵ) be given with $\epsilon > 0$. The Lagrangian to (P_ϵ) is given by

$$L(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{y}) = \sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}} + \boldsymbol{y}^T(\boldsymbol{A}^T\boldsymbol{x} - \boldsymbol{r} - \boldsymbol{b}). \tag{0.3}$$

The necessary first order (KKT) conditions for $(\boldsymbol{x}, \boldsymbol{r}, \epsilon, \boldsymbol{y})$ to be optimal for (0.3) are

$$\frac{\partial L}{\partial \boldsymbol{r}_i} = [\boldsymbol{g}_r]_i - \boldsymbol{y}_i = \frac{1}{\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}}\boldsymbol{r}_i - \boldsymbol{y}_i = \boldsymbol{0}, \quad for\ i = 1, \ldots, \kappa. \tag{0.4}$$

$$\frac{\partial L}{\partial \boldsymbol{x}} = \boldsymbol{g}_x = \boldsymbol{A}\boldsymbol{y} = \boldsymbol{0}. \tag{0.5}$$

Let $\boldsymbol{r}^*$ be an optimal solution to (P_ϵ) it follows of (0.4) that an optimal solution $\boldsymbol{y}^*$ to (D_ϵ) satisfy

$$\boldsymbol{y}_i^* = \frac{1}{\left(\|\boldsymbol{r}_i^*\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}}\boldsymbol{r}_i^* \Rightarrow \|\boldsymbol{y}_i^*\|_2^2 = \frac{\|\boldsymbol{r}_i^*\|_2^2}{\left(\|\boldsymbol{r}_i^*\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}} < 1.$$

Note that an optimal solution to (D_ϵ) always is interior with respect to the quadratic constraints. This solution $\boldsymbol{y}^*$ will also be optimal to the following problem

$$\begin{cases} \max -\boldsymbol{b}^T\boldsymbol{y} + \epsilon \sum_{i=1}^{\kappa}(1 - \|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}} \\ s.t. \\ \qquad \boldsymbol{A}\boldsymbol{y} = \boldsymbol{0} \end{cases} \tag{0.6}$$

The quadratic non-linear constraints can then be dropped, and the solution method can be used at (0.6) instead of (D_ϵ). In every iteration of the method it only have to be ensured that the current point is interior with respect to the quadratic constraints, so that the object function $-\boldsymbol{b}^T\boldsymbol{y} + \epsilon \sum_{i=1}^{\kappa}(1 - \|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}}$ is well defined. The primal-dual method can be develop, using Newtons method at the first order necessary conditions:

$$\boldsymbol{F}_1(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{y}) = \begin{pmatrix} \boldsymbol{g}_r(\boldsymbol{r}) - \boldsymbol{y} \\ \boldsymbol{A}^T\boldsymbol{x} - \boldsymbol{r} - \boldsymbol{b} \\ \boldsymbol{A}\boldsymbol{y} \end{pmatrix} = \boldsymbol{0} \tag{0.7}$$

Using Newtons method at (0.7) gives

$$\boldsymbol{F}_1'(\boldsymbol{x}^0, \boldsymbol{r}^0, \boldsymbol{y}^0)(\Delta\boldsymbol{x}, \Delta\boldsymbol{r}, \Delta\boldsymbol{y}) = -\boldsymbol{F}_1(\boldsymbol{x}^0, \boldsymbol{r}^0, \boldsymbol{y}^0) \tag{0.8}$$

Note that $\boldsymbol{y}$ is linear in $\boldsymbol{F}_1$, which means that this is essential the primal method, but with a updated dual feasible solution. From the first order conditions we have that a solution is optimal if and only if $\boldsymbol{y}_i = \boldsymbol{r}_i / \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}}$, it follows that complementarity condition is given by

$$\boldsymbol{r}_i - \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}} \boldsymbol{y}_i = \boldsymbol{0}$$

These conditions is equivalent to the first order conditions but better conditioned. A primal-dual method is then to use Newtons method at this complementarity condition subject to primal and dual feasibility

$$\widehat{\boldsymbol{F}}_1(\boldsymbol{x}, \boldsymbol{r}, \boldsymbol{y}) = \begin{pmatrix} \boldsymbol{r}_i - \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}} \boldsymbol{y}_i, \quad i = 1, \ldots, \kappa \\ \boldsymbol{A}^T\boldsymbol{x} - \boldsymbol{r} - \boldsymbol{b} \\ \boldsymbol{A}\boldsymbol{y} \end{pmatrix} = \boldsymbol{0} \tag{0.9}$$

The first derivative of $\widehat{\boldsymbol{F}}_1$ is

$$\frac{\partial(\widehat{\boldsymbol{F}}_1)_i}{\partial \boldsymbol{r}_i} = \boldsymbol{I} - \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{-\frac{1}{2}} \boldsymbol{y}_i \boldsymbol{r}_i^T \tag{0.10}$$

We note that in contradiction to $\boldsymbol{F}_1$ is the first derivative of $\widehat{\boldsymbol{F}}_1$ non symmetric which means that a symmetric equation solver not can be used. If we assume that (0.9) is satisfied for the given point, i.e. the point is optimal we have that

$$\widehat{\boldsymbol{F}}_1' = (\widehat{\boldsymbol{F}}_1')^T$$

It follows that

$$\widehat{\boldsymbol{F}}_1' \approx \frac{1}{2}(\widehat{\boldsymbol{F}}_1' + (\widehat{\boldsymbol{F}}_1')^T) = -\overline{\boldsymbol{F}}_1' \tag{0.11}$$

Where $\overline{\boldsymbol{F}}_1'$ is a symmetric positive definite matrix. We then explicitly symmetries the Newton step by approximate $\widehat{\boldsymbol{F}}_1'$ with $\overline{\boldsymbol{F}}_1'$ i.e. (0.8) is replaced of

$$\overline{\boldsymbol{F}}_1'(\boldsymbol{x}^0, \boldsymbol{r}^0, \boldsymbol{y}^0)(\Delta\boldsymbol{x}, \Delta\boldsymbol{r}, \Delta\boldsymbol{y}) = -F_1(\boldsymbol{x}^0, \boldsymbol{r}^0, \boldsymbol{y}^0) \tag{0.12}$$

. We have that (0.11) is satisfied with equality if the point is optimal (or if $l = 1$). Furthermore because $\overline{\boldsymbol{F}}_1'$ is positive definite the "Newton" directions will reduce the object function, which means that $\overline{\boldsymbol{F}}_1$ will converge to $\widehat{\boldsymbol{F}}_1'$ as the method converges to the optimal point. This way of explicitly symmetries a primal-dual method is well-known from methods for Semidefinite Programming, see [1]. For a related primal-dual method for (MSN) problems independent suggested and motivated different, see [7]. They use the duality gap

$$\sum_{i=1}^{\kappa} \|\boldsymbol{r}_i\|_2 + \boldsymbol{b}^T\boldsymbol{y} = \sum_{i=1}^{\kappa}(\|\boldsymbol{r}_i\|_2 - \boldsymbol{y}_i^T\boldsymbol{r}_i) \geq 0$$

to show that complementarity conditions for the original problem is given by

$$\boldsymbol{r}_i - \|\boldsymbol{r}_i\|_2\, \boldsymbol{y}_i = \boldsymbol{0}, \quad i = 1, \ldots, \kappa \tag{0.13}$$

They then see ϵ as a centering parameter, and find a solution of the system (0.9). They further use a predictor-corrector high order method to set ϵ. However their primal-dual method is not symmetric and it will require twice as much memory as for the symmetric method, if for solution of the linear system, which arise from Newtons method, a linear equation solver which use only pivoting at the diagonal with out respect to numerical values is used.

0.3 A Newton method for (MSN)

In this section we describe two different Newton methods for the (MSN) problem. In the first method which is a barrier method, we see ϵ as a variable with a lower bound. As in [3] this lower bound is removed by a log term which keeps ϵ away from zero. The barrier problem is as follows

$$\begin{cases} \min \sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}} - \mu\log(\epsilon) \\ s.t. \\ \qquad (\boldsymbol{x}, \boldsymbol{r}) \in X \end{cases} \tag{0.14}$$

where μ is the barrier parameter which controls how fast ϵ are going to zero. The second method is to see ϵ as a constant and reduce it between every Newton iteration, this is a traditional (HAP) approach. This have up to day not been an efficient approach, because of lack of a theory for reducing ϵ. However with feasible dual solutions a true bound is known at the duality gap, and this can be used to reduce ϵ. Furthermore in [7] they suggested to see ϵ (see Section 0.2) as centering parameter, and then use a predictor-corrector method to reduce it.

The necessary conditions for a solution to be optimal for (0.14) is

$$\widehat{\boldsymbol{F}}(\boldsymbol{x},\boldsymbol{r},\epsilon,\boldsymbol{y}) = \begin{bmatrix} \widehat{\boldsymbol{F}}_r \\ \widehat{\boldsymbol{F}}_\epsilon \\ \widehat{\boldsymbol{F}}_x \\ \widehat{\boldsymbol{F}}_y \end{bmatrix} = \begin{pmatrix} \boldsymbol{r}_i - \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}} \boldsymbol{y}_i, \quad i = 1,\dots,\kappa \\ \epsilon^2 \sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{-\frac{1}{2}} - \mu \\ \boldsymbol{A}^T\boldsymbol{x} - \boldsymbol{r} - \boldsymbol{b} \\ \boldsymbol{A}\boldsymbol{y} \end{pmatrix} = \boldsymbol{0}$$

Note that we with-out loss of generality have multiple through with ϵ in $\widehat{\boldsymbol{F}}_\epsilon$, the benefit is that the first derivative is independent of μ. We have

$$\frac{\partial(\widehat{\boldsymbol{F}}_r)_j}{\partial \boldsymbol{r}_i} + \left(\frac{\partial(\widehat{\boldsymbol{F}}_r)_j}{\partial \boldsymbol{r}_i}\right)^T = [\boldsymbol{H}_r]_i = \begin{cases} \boldsymbol{I}_\nu - \frac{1}{2}\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{-\frac{1}{2}} (\boldsymbol{y}_i\boldsymbol{r}_i^T + \boldsymbol{r}_i\boldsymbol{y}_i^T) & for\ j = i \\ \boldsymbol{O}_\nu & for\ j \neq i \end{cases} \tag{0.15}$$

$$\frac{\partial \widehat{\boldsymbol{F}}_r}{\partial \boldsymbol{y}_i} = [\boldsymbol{H}_y]_i = -\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}} \boldsymbol{I}_\nu \qquad for\ i = 1,\dots,\kappa \tag{0.16}$$

$$\frac{\partial \widehat{\boldsymbol{F}}_r}{\partial \epsilon} = [\boldsymbol{H}_{r\epsilon}]_i = -\epsilon\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{-\frac{1}{2}} \boldsymbol{y}_i \qquad for\ i = 1,\dots,\kappa \tag{0.17}$$

$$\frac{\partial \widehat{\boldsymbol{F}}_\epsilon}{\partial \boldsymbol{r}_i} = [\boldsymbol{H}_{\epsilon r}]_i = -\epsilon\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{-\frac{3}{2}} \boldsymbol{r}_i \qquad for\ i = 1,\dots,\kappa \tag{0.18}$$

$$\frac{\partial \widehat{\boldsymbol{F}}_\epsilon}{\partial \epsilon} = \boldsymbol{H}_\epsilon = 2\epsilon\sum_{i=1}^{\kappa}\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{-\frac{1}{2}} - \epsilon^3\sum_{i=1}^{\kappa}\left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{-\frac{3}{2}}. \tag{0.19}$$

Where $\boldsymbol{H}_r$ is explicit symmeticezing as described in Section 0.2.

With $\widehat{\boldsymbol{F}}$' modified as described in Section 0.2 Newtons method used at $\widehat{\boldsymbol{F}}$ gives the following conditions for updating $(\boldsymbol{x},\boldsymbol{r},\epsilon,\boldsymbol{y})$ with directions $(\Delta\boldsymbol{x},\Delta\boldsymbol{r},\Delta\epsilon,\Delta\boldsymbol{y})$:

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{A}^T & -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{H}_y & \boldsymbol{0} & \boldsymbol{H}_r & \boldsymbol{H}_{r\epsilon} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{H}_{\epsilon r} & \boldsymbol{H}_\epsilon \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{y} \\ \Delta\boldsymbol{x} \\ \Delta\boldsymbol{r} \\ \Delta\epsilon \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{A}\boldsymbol{y}^0 \\ -\widehat{\boldsymbol{F}}_r \\ -\widehat{\boldsymbol{F}}_\epsilon \end{bmatrix} \tag{0.20}$$

This can be rewritten into the following system

$$\boldsymbol{A}\boldsymbol{H}\boldsymbol{A}^T\Delta\boldsymbol{x} = (-\boldsymbol{A}\boldsymbol{y}^0) + \boldsymbol{A}(-\boldsymbol{H}_y^{-1})(\widehat{\boldsymbol{F}}_r + \boldsymbol{H}_{r\epsilon}\boldsymbol{H}_\epsilon^{-1}\widehat{\boldsymbol{F}}_\epsilon) \tag{0.21}$$

$$\Delta\boldsymbol{r} = \boldsymbol{A}^T\Delta\boldsymbol{x} \tag{0.22}$$

$$\Delta\epsilon = -\boldsymbol{H}_\epsilon^{-1}(\widehat{\boldsymbol{F}}_\epsilon + \boldsymbol{H}_{\epsilon r}\Delta\boldsymbol{r}) \tag{0.23}$$

$$\Delta\boldsymbol{y} = -\boldsymbol{H}_y^{-1}(\widehat{\boldsymbol{F}}_r + \boldsymbol{H}_r\Delta\boldsymbol{r} + \boldsymbol{H}_{r\epsilon}\Delta\epsilon) \tag{0.24}$$

where $\boldsymbol{H} = -\boldsymbol{H}_y^{-1}(\boldsymbol{H}_r - \boldsymbol{H}_{r\epsilon}\boldsymbol{H}_\epsilon^{-1}\boldsymbol{H}_{\epsilon r})$.

An algorithm SPDMSN for the solution of the (MSN) problem is given in Figure 0.1.

Procedure $SPDMSN(\boldsymbol{A}, \boldsymbol{b}, (\boldsymbol{x}^0, \epsilon^0, \boldsymbol{y}^0), \textit{stopping criterion}, \mu, \lambda)$

$k \leftarrow 0$

repeat

$\boldsymbol{r}^k \leftarrow \boldsymbol{A}^T\boldsymbol{x}^k - \boldsymbol{b}$.

$\Delta\boldsymbol{x} \leftarrow (\boldsymbol{AHA}^T)^{-1}(-\boldsymbol{A}\boldsymbol{y}^k + \boldsymbol{A}(-\boldsymbol{H}_y^{-1})(\boldsymbol{g}_r + \boldsymbol{H}_{r\epsilon}\boldsymbol{H}_\epsilon^{-1}g_\epsilon))$.

$\Delta\boldsymbol{r} \leftarrow \boldsymbol{A}^T\Delta\boldsymbol{x}$.

$\Delta\epsilon \leftarrow -\boldsymbol{H}_\epsilon^{-1}(g_\epsilon + \boldsymbol{H}_{r\epsilon}^T\Delta\boldsymbol{r})$.

$\Delta\boldsymbol{y} \leftarrow \boldsymbol{H}_y^{-1}(\boldsymbol{g}_r + \boldsymbol{H}_r\Delta\boldsymbol{r} + \boldsymbol{H}_{r\epsilon}\Delta\epsilon)$.

if $\Delta\epsilon < 0$ then $\alpha_{max}^p \leftarrow -\lambda\frac{\epsilon^k}{\Delta\epsilon}$ *else* $\alpha_{max} \leftarrow \infty$.

Linesearch (see section 0.3.1), Find α_p, α_d such that:

$$F_\mu(\boldsymbol{r}^k + \alpha_p\Delta\boldsymbol{r}, \epsilon^k + \alpha_p\Delta\epsilon) \approx \min_{0<\theta\le\alpha_{max}^p} F_\mu(\boldsymbol{r}^k + \theta\Delta\boldsymbol{r}, \epsilon^k + \theta\Delta\epsilon).$$

$$-\boldsymbol{b}^T(\boldsymbol{y} + \alpha_d\Delta\boldsymbol{y}) + \epsilon^k\sum_{i=1}^{\kappa}(1 - \|\boldsymbol{y} + \alpha_d\Delta\boldsymbol{y}\|_2^2)^{\frac{1}{2}} \approx$$

$$\max_{0<\theta\le\alpha_{max}^d} -\boldsymbol{b}^T(\boldsymbol{y} + \theta\Delta\boldsymbol{y}) + \epsilon^k\sum_{i=1}^{\kappa}(1 - \|\boldsymbol{y} + \theta\Delta\boldsymbol{y}\|_2^2)^{\frac{1}{2}}$$

$\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k + \alpha_p\Delta\boldsymbol{x}$.

$\boldsymbol{y}^{k+1} \leftarrow \boldsymbol{y}^k + \alpha_d\Delta\boldsymbol{y}$.

$\epsilon^{k+1} \leftarrow \epsilon^k + \alpha_p\Delta\epsilon$.

$k \leftarrow k + 1$.

until stopping criterion satisfied

end SPDMSN.

where $0 < \lambda < 1$ is a scalar to ensure $\epsilon^{k+1} > 0$.

Figure 0.1: The SPDMSN algorithm

In Figure 0.1 is the barrier method shown, the only difference for the method with ϵ constant is that $\boldsymbol{H}_{r\epsilon}$, $\boldsymbol{H}_{\epsilon r}$, $\boldsymbol{H}_\epsilon$ and $\Delta\epsilon$ is zero in every iteration. ϵ is then reduced between every iteration of $SPDMSN$.

0.3.1 The primal and dual line search

In both the primal and dual space is a safe guarded Newton line search used. The original object function in the dual problem (D) is linear. The line search for the dual space therefore always take a step to the boundary if the given dual direction is an ascent direction $\boldsymbol{b}^T\Delta\boldsymbol{y} > 0$. This was found to work well in practice. The maximum feasible step in the dual space to the boundary of the quadratic constraints is found as follows. Let a direction $\Delta\boldsymbol{y}$ be given, we then find the step length α_{max}^d such that the new point $\boldsymbol{y} = \boldsymbol{y}^0 + \alpha_{max}^d\Delta\boldsymbol{y}$ satisfy

$$\|\boldsymbol{y}_i\|_2 = \left\|\boldsymbol{y}_i^0 + \alpha_{max}^d\Delta\boldsymbol{y}_i\right\|_2 < 1.0 \qquad i = 1, \ldots, \kappa$$

which is equivalent to:

$$\left\|\boldsymbol{y}_i^0 + \alpha_{max}\Delta\boldsymbol{y}_i\right\|_2^2 = \left\|\boldsymbol{y}_i^0\right\|_2^2 + 2\alpha_{max}^d(\boldsymbol{y}_i^0)^T\Delta\boldsymbol{y}_i + (\alpha_{max}^d)^2\left\|\Delta\boldsymbol{y}_i\right\|_2^2 < 1.0$$
$$(\alpha_{max}^d)^2\left\|\Delta\boldsymbol{y}_i\right\|_2^2 + 2\alpha_{max}^d(\boldsymbol{y}_i^0)^T\Delta\boldsymbol{y}_i + (\left\|\boldsymbol{y}_i^0\right\|_2^2 - 1) < 0$$

The left hand side is a second order polynomial which have the following two roots:

$$roots = \begin{cases} \frac{-\Delta\boldsymbol{y}_i^T\boldsymbol{y}_i^0 + \sqrt{\left\|\Delta\boldsymbol{y}_i\right\|_2^2(1.0 - \left\|\boldsymbol{y}_i^0\right\|_2^2) + (\Delta\boldsymbol{y}_i^T\boldsymbol{y}_i^0)^2}}{\left\|\Delta\boldsymbol{y}_i\right\|_2^2} = a_i \\ \frac{-\Delta\boldsymbol{y}_i^T\boldsymbol{y}_i^0 - \sqrt{\left\|\Delta\boldsymbol{y}_i\right\|_2^2(1.0 - \left\|\boldsymbol{y}_i^0\right\|_2^2) + (\Delta\boldsymbol{y}_i^T\boldsymbol{y}_i^0)^2}}{\left\|\Delta\boldsymbol{y}_i\right\|_2^2} = b_i \end{cases}$$

It follows that

$$\alpha_{max}^d = \lambda \min\left\{a_i \quad | \left\|\Delta\boldsymbol{y}_i\right\|_2^2 \neq 0, \quad i = 1, \ldots, \kappa\right\} \tag{0.25}$$

where $0 < \lambda < 1$ is a constant which ensures that the new point $\boldsymbol{y}$ is interior with respect to the quadratic constraints.

0.3.2 High order

In [7] they suggested to incorporate Mehrotra's predictor-corrector rule, which is known to be effective in the context of primal-dual methods for linear programming [11]. The idea is to approximate the non-linear system

$$\widehat{\boldsymbol{F}}(\boldsymbol{w}) = 0.$$

with a second order Taylor series

$$\widehat{\boldsymbol{F}}'(\boldsymbol{w}^0)\Delta\boldsymbol{w} + \frac{1}{2}\Delta\boldsymbol{w}^T\widehat{\boldsymbol{F}}''(\boldsymbol{w}^0)\Delta\boldsymbol{w} + \widehat{\boldsymbol{F}}(\boldsymbol{w}^0) = \boldsymbol{0}$$

It is not practical possible to find a solution of this system, so instead is an estimate for $\Delta\boldsymbol{w}$ predicted $\Delta\boldsymbol{w}^e$ and $\Delta\boldsymbol{w}$ is found as a solution to

$$\widehat{\boldsymbol{F}}'(\boldsymbol{w}^0)\Delta\boldsymbol{w} = -(\frac{1}{2}(\Delta\boldsymbol{w}^e)^T\widehat{\boldsymbol{F}}''(\boldsymbol{w}^0)\Delta\boldsymbol{w}^e + \widehat{\boldsymbol{F}}(\boldsymbol{w}^0))$$

Mehrotra suggested to calculate $\Delta\boldsymbol{w}^e$ as the solution to (0.20), with $\mu = 0$. This means that $\Delta\boldsymbol{w}^e$ only cost an extra solution of the linear system, because the coefficient matrix $\widehat{\boldsymbol{F}}'(\boldsymbol{w}^0)$ is unchanged. This method furthermore gives an effective rule for reducing $\mu(\epsilon)$. With duality gap given as

$$\begin{aligned} dgap &= \sum_{i=1}^{\kappa}\left\|\boldsymbol{r}_i^0\right\|_2 - \boldsymbol{r}_i^T\boldsymbol{y}_i^0 \\ ngap &= \sum_{i=1}^{\kappa}\left\|\boldsymbol{r}_i^0 + \Delta\boldsymbol{r}_i^e\right\|_2 - (\boldsymbol{r}_i + \Delta\boldsymbol{r}_i^e)^T(\boldsymbol{y}_i^0 + \Delta\boldsymbol{y}_i^e) \end{aligned}$$

we set

$$\mu = \left(\frac{ngap}{dgap}\right)^2 ngap \tag{0.26}$$

or if the non-barrier method is used (i.e. ϵ constant parameter)

$$\epsilon = \left(\frac{ngap}{dgap}\right)^2 \frac{ngap}{\kappa} \tag{0.27}$$

which is equivalent to the predictor-corrector method suggested in [7]. The reduction in the duality gap is used as measure for how close the given point is to the central path of optimal points for given μ. So a large reduction means that μ can be set low, i.e. the center direction is given a small weight.

0.3.3 The use of recentering in the primal-dual Newton method

We here look on how to improve the primal-dual Newton method, with $\epsilon = \mu$ seen as a penalty(centering) parameter as in [7]. Let us look again at the pair of dual smooth problems

$$(P_\epsilon) \qquad \begin{cases} \min \sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon^2\right)^{\frac{1}{2}} \\ s.t. \\ \qquad (\boldsymbol{x}, \boldsymbol{r}) \in X \end{cases}$$

$$(D_\epsilon) \qquad \begin{cases} \max -\boldsymbol{b}^T\boldsymbol{y} + \epsilon \sum_{i=1}^{\kappa} (1 - \|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}} \\ s.t. \\ \qquad \boldsymbol{y} \in Y \end{cases}$$

Clearly in a primal-dual Newton method, the risk is that only small step can be taken in the dual space if ϵ is reduced fast, because the same cost is assigned to every slack in the dual quadratic constraints. This is unsatisfactory because it introduce ill-conditioning and because it cause the method to converge slowly. Our idea is to introduce a different cost for every slack.

$$(P_\epsilon) \qquad \begin{cases} \min \sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon_i^2\right)^{\frac{1}{2}} \\ s.t. \\ \qquad (\boldsymbol{x}, \boldsymbol{r}) \in X \end{cases}$$

$$(D_\epsilon) \qquad \begin{cases} \max -\boldsymbol{b}^T\boldsymbol{y} + \sum_{i=1}^{\kappa} \epsilon_i (1 - \|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}} \\ s.t. \\ \qquad \boldsymbol{y} \in Y \end{cases}$$

The question is how shall ϵ_i be set in every iteration. We assume that an optimal solution $(\boldsymbol{x}^*, \boldsymbol{r}^*, \boldsymbol{y}^*)$ is known to the smooth problems, we then have

$$\sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i^*\|_2^2 + \epsilon_i^2\right)^{\frac{1}{2}} - (-\boldsymbol{b}^T\boldsymbol{y}^* + \sum_{i=1}^{\kappa} \epsilon_i (1 - \|\boldsymbol{y}_i^*\|_2^2)^{\frac{1}{2}}) = 0 \tag{0.28}$$

$$\Updownarrow$$

$$\sum_{i=1}^{\kappa} \|\boldsymbol{r}_i^*\|_2 + \boldsymbol{b}^T\boldsymbol{y}^* \leq \sum_{i=1}^{\kappa} \left(\|\boldsymbol{r}_i^*\|_2^2 + \epsilon_i^2\right)^{\frac{1}{2}} + \boldsymbol{b}^T\boldsymbol{y}^* = \sum_{i=1}^{\kappa} \epsilon_i (1 - \|\boldsymbol{y}_i^*\|_2^2)^{\frac{1}{2}} \tag{0.29}$$

With the same cost at every constraint we did set $\epsilon_i = \mu$, which means that every cost is centered with the same. If instead

$$\epsilon_i(1 - \|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}} = \mu \Leftrightarrow \epsilon_i = (1 - \|\boldsymbol{y}_i\|_2^2)^{-\frac{1}{2}}\mu \tag{0.30}$$

we still have that $\kappa\mu$ is a bound at the dual gap and every constraint is centered after how close it is to its bound. Using (0.30) the complementarity conditions for the smooth pair of problems is:

$$\boldsymbol{r}_i - \left(\|\boldsymbol{r}_i\|_2^2 + \epsilon_i^2\right)^{\frac{1}{2}} \boldsymbol{y}_i = \boldsymbol{0} \tag{0.31}$$

$$\Updownarrow$$

$$\boldsymbol{r}_i - \left(\|\boldsymbol{r}_i\|_2^2 + ((1 - \|\boldsymbol{y}_i\|_2^2)^{-\frac{1}{2}}\mu)^2\right)^{\frac{1}{2}} \boldsymbol{y}_i = \boldsymbol{0}$$

$$\Updownarrow \tag{0.32}$$

$$(1 - \|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}}\boldsymbol{r}_i - \left(\left\|((1 - \|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}}\boldsymbol{r}_i)\right\|_2^2 + \mu^2\right)^{\frac{1}{2}} \boldsymbol{y}_i = \boldsymbol{0} \tag{0.33}$$

It follows that the complementary conditions $(1 - \|\boldsymbol{y}_i\|_2^2)^{\frac{1}{2}}\boldsymbol{r}_i$, which is zero in the optimal solution, is center with μ instead of centering $\|\boldsymbol{r}_i\|_2$ with μ. Using Mehrotra's predictor-corrector rule (0.27) for setting μ we set in every iteration

$$\epsilon_i = (1 - \left\|(\boldsymbol{y}_i)^k\right\|_2^2)^{-\frac{1}{2}}\mu \tag{0.34}$$

Using the above, the method was observed in the dual space to take steps with α_d between 0.5 and 1.0 in most iterations. Due to the fact that a self correcting Newton is used (for a step size of 1.0 it theoretically correct the linear constraints exact), this method was also observed to give higher accuracy for the dual linear constraints.

0.4 Implementation details

The method was implemented in $ANSI - C$, and has been ported to several different computers. The name of the code is $GOPT$ which is short for global optimizer (i.e. a positive definite Hessian), and it can at current find solutions to MSN and LP problems (The GOPT code is joint between K. D. Andersen and E. D. Andersen). The MPS format is used as input format. Implementation details have already been given in [3]. A Cholesky decomposition is used for the positive symmetric system which is solved in every iteration. The code include the barrier as well as the non barrier method i.e. ϵ is seen as a constant. If ϵ is constant $\boldsymbol{H}_{\epsilon r}$ and $\boldsymbol{H}_{r\epsilon}$ is $\boldsymbol{0}$ so the linear system is symmetric. However if the Barrier method is used a solution have to be found of the following system:

$$\begin{bmatrix} \boldsymbol{A}(-\boldsymbol{H}_y^{-1})\boldsymbol{H}_r\boldsymbol{A}^T & \boldsymbol{A}(-\boldsymbol{H}_y^{-1})\boldsymbol{H}_{r\epsilon} \\ \boldsymbol{H}_{\epsilon r}\boldsymbol{A}^T & \boldsymbol{H}_\epsilon \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{x} \\ \Delta\epsilon \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix} \tag{0.35}$$

$\boldsymbol{A}(-\boldsymbol{H}_y^{-1})\boldsymbol{H}_r\boldsymbol{A}^T$ is symmetric and positive definite, and $(-\boldsymbol{H}_y^{-1})\boldsymbol{H}_r$ is a block diagonal matrix with every block of size $\nu \times \nu$. The Cholesky factor $\boldsymbol{L}$ is found of $\boldsymbol{A}(-\boldsymbol{H}_y^{-1})\boldsymbol{H}_r\boldsymbol{A}^T$.

As described in [3] the symbolic structure is found once for $\boldsymbol{L}$, using a minimum degree heuristic to reduce the fill-in in $\boldsymbol{L}$. However it is the unsymmetric system (0.35) we will find a solution of. Let $\boldsymbol{L}$ be the Cholesky factor of $\boldsymbol{A}(-\boldsymbol{H}_y^{-1})\boldsymbol{H}_r\boldsymbol{A}^T$ and set $\boldsymbol{L}\boldsymbol{p}_1 = \boldsymbol{A}(-\boldsymbol{H}_y^{-1})\boldsymbol{H}_{r\epsilon}$, $\boldsymbol{L}\boldsymbol{p}_2 = \boldsymbol{A}\boldsymbol{H}_{\epsilon r}$ and $\boldsymbol{L}\boldsymbol{q} = \boldsymbol{b}_1$. We have that (0.35) is equivalent to

$$\begin{aligned} \boldsymbol{L}\boldsymbol{L}^T\Delta\boldsymbol{x} + \boldsymbol{L}\boldsymbol{p}_1\Delta\epsilon &= \boldsymbol{L}\boldsymbol{q} && (0.36)\\ \boldsymbol{p}_2^T\boldsymbol{L}^T\Delta\boldsymbol{x} + \boldsymbol{H}_\epsilon\Delta\epsilon &= \boldsymbol{b}_2 && (0.37) \end{aligned}$$

and it follows that

$$\begin{aligned} (\boldsymbol{H}_\epsilon - \boldsymbol{p}_2^T\boldsymbol{p}_1)\Delta\epsilon &= \boldsymbol{b}_2 - \boldsymbol{p}_2^T\boldsymbol{q} && (0.38)\\ \boldsymbol{L}^T\Delta\boldsymbol{x} &= \boldsymbol{q} - \boldsymbol{p}_1\Delta\epsilon && (0.39) \end{aligned}$$

The Cholesky decomposition which is used is described in [2], and how to extend for handling dense columns is described in [9]

Linear constraints are handled as described in [5]. One of the advantage with the primal-dual method is that the method now have a dual feasible solution in every iteration. This can be used to set the weight at the l_1 penalty function. If we assume that a dual solution $(\boldsymbol{y}^k, \boldsymbol{y}_e^k) \in Y$ is known, the weight ρ^k at the linear penalty function is set to

$$\rho^{k+1} = \max(\rho^k, \beta 2\sqrt{2}\left\|\boldsymbol{y}_e^k\right\|_\infty). \tag{0.40}$$

motivated of theorem 3 in [5]. We simply always try to satisfy the conditions of the theorem. A large ρ can introduce ill-condition in the linear system which the method find a solution of in every iteration. We therefore reduce ρ if $\rho > 4\sqrt{2}\left\|\boldsymbol{y}_e^k\right\|_\infty$, it is then set to

$$\rho^{k+1} = \max(1, \frac{1}{2}\rho^k). \tag{0.41}$$

0.5 Computational results

All tests presented in this section are made on a CONVEX C3240 vector computer at Odense University. The code is compiled with the CONVEX-C compiler using option "-O2 -rl -peel " (full vectorization). 64-bit floating-point arithmetic (implying between 15 and 16 digits accuracy) is used in all numerical computations. Times are in CPU seconds and are measured with the ANSI-C procedure "clock". Computing time includes all processing except time for reading the data.

For all computation have we set $\beta = 1$ and $\lambda = 0.99$.

We use the same test problems as in [3, 5]. The problems is described in [4, 10, 6].

The test results are reported in Table $1-5$. $|\boldsymbol{A}|$ denotes the number of nonzero entries in $\boldsymbol{A}$, $|\boldsymbol{A}\boldsymbol{H}\boldsymbol{A}^T|/2$ is the number of nonzeros in $\boldsymbol{A}\boldsymbol{H}\boldsymbol{A}^T$ below and in the diagonal, $|\boldsymbol{L}|$ is the number of nonzeros in and below the diagonal in the Cholesky factor $\boldsymbol{L}$ of $\boldsymbol{A}\boldsymbol{H}\boldsymbol{A}^T$, ITE is the iteration count. The column heading "0-terms" indicates the number of $\boldsymbol{r}_i$-terms smaller than the tolerance 10^{-10} in the final solution. D_{gap} is the relative duality gap

$$D_{gap} = \frac{|F(\boldsymbol{r}) + \boldsymbol{b}\boldsymbol{y} + \boldsymbol{b}_e\boldsymbol{y}_e|}{F(\boldsymbol{r}) + 1}.$$

P_{fea} is the primal infeasibility given by the expression

$$P_{fea} = \frac{\left\| \boldsymbol{E}^T \boldsymbol{x} - \boldsymbol{b} \right\|_2}{\boldsymbol{b} + 1.0}$$

Finally L_{fea} denotes the infeasibility of the dual linear constraints given by $\|\boldsymbol{Ay} - \boldsymbol{Ey}_e\|_2$.

In Table $1-3$ is the results shown for the unconstrained problems from [3]:

Name	κ	n	$\|A\|$	$\|AHA^T\|/2$	$\|L\|$	ITE	CPU	$0-terms$	D_{gap}	L_{fea}
ssp4	25	15	224	97	105	7	0	0	2E-10	2E-14
ssp10	121	99	1760	1010	1785	8	1	0	2E-10	7E-15
ssp50	2601	2499	48800	31010	171083	9	65	0	2E-09	4E-13
ssp100	10201	9999	197600	127010	929515	10	354	0	1E-09	9E-14
ssp300	90601	89999	1792800	1161010	13203975	11	6709	0	1E-09	2E-13
ssp400	160801	159999	3190400	2068010	31011299	12	22139	0	2E-10	2E-13

Table 0.1: Results for "sspN" with $\nu = 3$

NAME	κ	n	$\|A\|$	$\|AHA^T\|/2$	$\|L\|$	ITE	CPU	$0-terms$	D_{gap}	L_{fea}
l3a13	49	52	1390	1142	1207	14	1	12	5e-10	2e-12
l12a13	625	640	21331	26406	57421	19	34	179	2e-09	4e-12
l21a13	1849	1876	64762	84384	278691	24	200	958	4e-10	4e-12
l30a13	3721	3760	131683	175086	726046	24	461	2315	7e-09	3e-11
l60a13	14641	14720	524403	713766	4337857	28	4710	11265	4e-09	1e-10
l99a13	39601	39732	1425134	1957631	14693151	30	17937	33503	7e-09	9e-10
l120a13	58081	58240	2092843	2881926	24202413	34	44144	50548	8e-09	4e-10

Table 0.2: Results for "lNa13" with $\nu = 2$

NAME	κ	n	$\|A\|$	$\|AHA^T\|/2$	$\|L\|$	ITE	CPU	$0-terms$	D_{gap}	L_{fea}
l20a20	1681	1718	59054	76945	246012	20	134	1224	9e-09	4e-11
l40a20	6561	6636	234145	315504	1511239	32	1416	6156	6e-09	3e-11
l60a20	14641	14754	525236	715653	4433304	32	4937	14181	7e-09	5e-11
l80a20	25921	26072	932327	1277402	8645485	32	12133	25359	8e-09	4e-11
l120a20	58081	58308	2094509	2885699	24240011	29	36516	57127	7e-09	3e-11

Table 0.3: Results for "lNa20" with $\nu = 2$

We see that the duality gap and $\|\boldsymbol{Ay}\|_2$ for all problems are satisfactory. Compared to the results in [3] is the number of iterations much lower for especial the two problems lNa13 and lNa20, with many active norms in the optimal solution. For example for l80a20 the primal method took 164 iterations against the 32 in the primal-dual method. We furthermore see the same reduction in CPU time. For the smooth problems sspN is the reduction in iterations practically 0, however the number of iterations is still lower than in [3]. Another benefit of the primal-dual method is that the number of iterations increase very slow with problem size and for lNa20, it even seems to be constant.

In Table $4-5$ is the results shown for the linear constrained problems, which also was used in [5]. As for the unconstrained problems the number of iterations and CPU time is significantly reduced. ρ is initial set to $2\sqrt{2}$, and for no problem did the method reset the weight more than two times, this was a clear improvement of the method used in [5]. This was observed to give a better conditioned convergence.

NAME	κ	n	l	$\lvert A\rvert$	$\lvert AAT\rvert$	$\lvert L\rvert$
cl3	9	26	10	163	168	198
cl12	144	320	145	2818	2790	7402
cl30	900	1880	901	17848	17511	78128
cl60	3600	7360	3601	71698	70126	413267
cl99	9801	19866	9802	195523	191000	1859559
cl120	14400	29120	14401	287398	280656	2047512
cl201	40401	81338	40402	807013	787580	9919534
cl399	159201	319466	159202	3182023	3103949	57885659

Table 0.4: Size for "clN" with $\nu = 2$

NAME	CPU	ITE	$\lVert \boldsymbol{r}_i\rVert_2 \leq \epsilon_l$	DGap	$\lvert Au\rvert$	E_{fea}
cl3	0	11	1	1.6e-08	4.4e-9	1.0e-15
cl12	4	13	95	6.7e-09	2.1e-13	1.1e-13
cl30	31	16	651	4.0e-09	2.3e-13	3.4e-12
cl60	180	20	2878	7.4e-09	1.3e-13	7.8e-11
cl99	890	24	8234	1.2e-08	1.0e-13	7.0e-13
cl120	1238	25	12311	1.2e-08	1.0e-11	2.0e-13
cl201	6179	24	35803	3.1e-08	1.0e-13	5.1e-13

Table 0.5: Results for "clN" with SPDMSN

In [5] was the results for the clNa13 problems compared to results achieved with an interior-point method for linear programming, at a linear discretisation of the problem. We have compared the new results to the LP results in Table 0.6. We see that as the problem size increase the CONVEX method is a magnitude faster than the LP method.

Clearly the number of iterations and the time required for the SPDMSN method is reduced a lot in comparison with the the LP method, however the time and memory requirement can even be reduced more. The cl problems contain one column which is relative dense. Using the technique described in [9] to handle this dense column, we receive the results which is shown in Table 0.7

As we can see the size of the $\boldsymbol{L}$ factor falls with up to over 25% without decreasing the numerical accuracy and with nearly the same number of iterations. For cl399 one column in $\boldsymbol{A}$ with only 400 non-zeros is treated with the dense column technique and for this problem the size of $\boldsymbol{L}$ falls with over 20 million nonzeros. This shows that the technique

N	CPU			ITE			$\|L\|$		
	MSN	LP	L/M	MSN	LP	L/M	MSN	LP	L/M
3	0	0	1.0	11	19	1.7	198	254	1.3
12	4	13	3.3	13	26	2.0	7402	16390	2.2
30	31	212	6.8	16	37	2.3	78128	218232	2.8
60	180	1655	9.2	20	51	2.6	413267	1150477	2.8
99	890	7848	8.8	22	64	2.9	1859559	3732759	2.0
120	1238	16479	13.3	25	76	3.0	2047512	5978370	2.9
201	6178	117099	19.0	24	98	4.1	9919534	20872252	2.1

Table 0.6: Comparison between MSN- and LP-approach for the test problems

NAME	$\|L\|$	CPU	ITE	D_{gap}	L_{fea}	P_{fea}
cl30	69131	28	16	3.0e-09	6.8e-12	2.7e-13
cl60	385862	189	21	3.0e-09	3.3e-10	2.6e-13
cl99	1232072	666	24	7.2e-09	1.4e-10	2.0e-13
cl120	1958961	1254	26	3.0e-09	6.0e-11	2.0e-13
cl201	6367553	4769	27	9.0e-09	1.0e-07	2.7e-13
cl399	30001436	53623	51	5.0e-07	2.7e-08	2.0e-13

Table 0.7: Results for “clN” with SPDMSN, using the dense column technique

described in [9] for handling of dense columns make it possible to find solutions of larger problems for this type problems.

0.6 Conclusion

We conclude that the described symmetric primal-dual pseudo (real for l=1) Newton method, is more efficient than the primal Newton method described in [3, 5]. The primal-dual method furthermore gives a better method for setting the weight at the l_1 penalty function.

Acknowledgment

The work described in this paper was first time presented of the first author in a talk at SCCM section of the Computer Science Department at Stanford University. The first author would like to thank Michael Saunders for correcting a mistake made in the talk and for pointing out that the method is a pseudo Newton method.

Bibliography

[1] F. Alizadeh, J.-P. A. Haeberly, and M.L. Overton. Primal-Dual Interior Point Methods for Semidefinite Programming. Technical report, 1994.

[2] E.D. Andersen and Y. Ye. Combining interior-point and pivoting algorithms for linear programming. Technical report, 1994. In preparation.

[3] K.D. Andersen. An efficient Newton barrier method for minimizing a sum of euclidean norms. preprint 42, Dept. Of Math. and Computer Sci., Odense University, 1993. To appear in SIAM J. on Optim. 1995.

[4] K.D. Andersen and E. Christiansen. Limit Analysis with the Dual Affine Scaling Algorithm. Preprint 26, Dept. Of Math. and Computer Sci., Odense University, 1993. To appear in the J. of Comput. and Applied Math.

[5] K.D. Andersen and E. Christiansen. A Newton Barrier method for Minimizing a Sum of Euclidean Norms subject to linear equality constraints. Preprint 7, Dept. of Mathematics and Comp. Sci., Odense University, 1995. submitted to SIAM Journal on Optimization.

[6] E. Christiansen and K.O. Kortanek. Computation of the collapse state in limit analysis using the lp primal affine scaling algorithm. *J. Comput. Appl. Math.*, 34:47–63, 1991.

[7] A. R. Conn and M. L. Overton. A primal-dual interior point method for minimizing a sum of Euclidean distances. In preparation.

[8] K.A. Brakke. Numerical solution of soap film dual problems. preprint, Susquehanna University, Mathematics Department, 1994.

[9] K.D. Andersen. A modified Schur complement method for handling dense columns in interior point methods for linear programming . Technical report, Odense University, 1994. Submitted to ACM Transaction on Mathematical Software.

[10] K.D. Andersen and E. Christiansen and M.L. Overton. Computing limit loads by Minimizing a sum of norms. Preprint 30, Dept. of Mathematics and Comp. Sci., Odense University, 1994. Submitted to SIAM Journal on Scientific Computing.

[11] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM J. on Optim.*, 2(4):575–601, 1992.

[12] M.L. Overton. A quadratically convergent method for minimizing a sum of Euclidian norms. *Math. Programming*, 27:34–63, 1983.