An Infeasible Active Set Method with Combinatorial Line Search for Convex Quadratic Problems with Bound Constraints*

Philipp Hungerländer† Franz Rendl‡

March 10, 2017

Abstract

The minimization of a convex quadratic function under bound constraints is a fundamental building block for solving more complicated optimization problems. The active-set method introduced by Bergounioux et al. [1, 2] has turned out to be a powerful, fast and competitive approach for this problem. Hintermüller et al. [15] provide a theoretical explanation of its efficiency by interpreting it as a semismooth Newton method. One major drawback of this method lies in the fact that it is not globally convergent for all classes of convex quadratic objectives.

Several modifications were introduced recently, that ensure global convergence. In this paper we introduce yet another modified version of this active set method, which aims at maintaining the combinatorial flavour of the original semismooth Newton method. We prove global convergence for our modified version and show it to be competitive on a variety of difficult classes of test problems.

Key words. primal-dual active set methods, semismooth Newton method, quadratic programming, box-constraints, convex programming

1 Introduction

We consider the convex programming problem

$$\min_{x} \quad J(x), \tag{1a}$$

$$\text{subject to} \quad x \leq b, \tag{1b}$$

where

$$J(x) = \frac{1}{2}x^\top Qx + q^\top x, \tag{2}$$

Q is a positive-definite $n \times n$ matrix, and $b, q \in \mathbb{R}^n$.

This is a prototype problem with a wide range of applications. Regularization with an ℓ_1-penalty term can be recast as (1). It is used as a basic building block in compressed sensing, with practical applications in imaging and face recognition, see e.g. Candès et al. [5] and Donoho [8]. Other areas of application are non-negative matrix factorization [Lee and Seung [23], Paatero and Tapper [27]], non-negative sparse coding [Hoyer [16]] and second-order conic-constrained quadratic programming [Goldberg and Leyffer [12]].

From a theoretical point of view this problem is well understood. Global optimality can be characterized by the Karush-Kuhn-Tucker conditions, see (3) below.

On the computational side there are several well established techniques to solve this problem. One of the oldest approaches is based on first order methods using projected gradients, see e.g. Figueiredo et al. [10]. Alternatively,

*This work was partially supported by the Austrian Science Fund (FWF) Project J3793-N95. Furthermore partial financial support from Marie-Curie ITN Project MINO, MC-ITN 316647, is gratefully acknowledged.

†Laboratory for Information & Decision Systems, MIT, USA, philipp.hungerlaender@aau.at

‡Department of Mathematics, Alpen-Adria Universität Klagenfurt, Austria, franz.rendl@aau.at

interior-point based algorithms have been introduced which can also incorporate general linear constraints, but specially taylored implementations for (1) are available. D'Apuzzo and Marino [7] propose a potential-reduction interior point method for (1) with box constraints and focus on parallel implementation issues. Kim et al. [21] introduce a specially taylored interior point method with a preconditioned conjugate gradient algorithm for the search direction and apply it to large sparse ℓ_1-regularized least squares problems, which can be formulated in the form (1).

Yet another group of algorithms follows the active-set strategy. There are variants which maintain primal feasibility throughout, but also infeasible active set methods have turned out to be competitive. The standard active set approach is described in detail in Nocedal and Wright [26, Section 16.5]. It is further investigated and extended by Moré and Toraldo [24] and Dostál and Schöberl [9], who combine it with projected gradients.

A special type of active-set method was introduced by Bergounioux et al. [1, 2] in connection with constrained optimal control problems and taylored to deal with discretisations of specially structured elliptic partial differential equations. Hintermüller et al. [15] show that this primal-dual active set approach can in fact be rephrased as a semi-smooth Newton method. Practical computational evidence shows that if this method converges at all, it typically takes very few iterations to reach an optimal solution, see for instance Kunisch and Rendl [22] and Júdice and Pires [20]. Unfortunately, global convergence of this active-set method, which we denote from now on as SN-method, can only be guaranteed for special classes of positive definite matrices. In [15] it is shown that the method converges globally if Q is a (sufficiently small perturbation of) an M-matrix, and it converges superlinearly if the starting point is sufficiently close to the optimum. In [22] global convergence of this method is shown under a strong form of diagonal dominance on Q. Convergence in the general case can not be ensured, as shown by Ben Gharbia and Gilbert [11] for the case of P-matrices of order $n \geq 3$ and Curtis et al. [6] in case of general symmetric positive definite matrices of order $n \geq 3$, see example 2 below.

There has recently been substantial scientific activity to figure out safeguards for the SN-method to ensure global convergence. Byrd et al. [4] insure global convergence by maintaining descent directions in combination with a safeguarding procedure proposed by Judice and Pires [20]. Curtis et al. [6] consider a more general setting, which includes also linear equations in the constraints. A special treatment is given to variables which change index sets very often, thereby forcing global convergence. In a previous paper [18] we investigate modifications of the SN-method to make it globally convergent. We focus on primal feasible iterates and use recursion to force decrease of the objective function.

It is the main purpose of the present paper to introduce yet another globalization strategy for the SN-method, which avoids recursion and which aims at maintaining the combinatorial flavour of the original SN-method. The new approach inherits the preferable features of the SN-method, like simplicity, finding the exact numerical solution, insensitivity with respect to initialisation and lack of strict complementary. Finally, we will argue that the computational overhead to safeguard the algorithm is in most cases only marginal.

The paper is organized as follows. At the end of this section we summarize notation used throughout the paper. In Section 2 we recall the SN-method. In Section 3 we introduce some theoretical features of the SN-method which are used in Section 4 to introduce safeguards to keep the SN-method from cycling. These essentially use a merit function to decide whether the SN-Update is acceptable or not. We also consider practical enhancements to avoid safeguard steps, which are explained in Section 5. Finally we close with computational experiments and comparisons on a variety of problem types in Section 6.

Notation: The following notation will be used throughout. For a subset $\mathcal{A} \subseteq \mathcal{N} := \{1, \ldots, n\}$ and $x \in \mathbb{R}^n$ we write $x_{\mathcal{A}}$ for the components of x indexed by $\mathcal{A}$, i.e. $x_{\mathcal{A}} := (x_i)_{i \in \mathcal{A}}$. If Q is a matrix and $\mathcal{A}, \mathcal{B} \subseteq \mathcal{N}$, then $Q_{\mathcal{A},\mathcal{B}}$ is the submatrix of Q, given by $Q_{\mathcal{A},\mathcal{B}} = (q_{ij})_{i \in \mathcal{A}, j \in \mathcal{B}}$. We write $Q_{\mathcal{A}}$ for $Q_{\mathcal{A},\mathcal{A}}$. For $a, b \in \mathbb{R}^n$ we write $a \circ b$ to denote the vector of element-wise products, i.e. $a \circ b := (a_i b_i)_{i \in \mathcal{N}}$.

2 The Semismooth-Newton-Method for (1)

It is well known that x together with a vector $\alpha \in \mathbb{R}^n$ of Lagrange multipliers for the simple bound constraints furnishes a (global) minimum of (1) if and only if (x, α) satisfies the KKT system

$$Qx + q + \alpha = 0, \tag{3a}$$
$$\alpha \circ (b - x) = 0, \tag{3b}$$
$$b - x \geq 0, \tag{3c}$$
$$\alpha \geq 0. \tag{3d}$$

Let us briefly recall the semismooth-Newton-Method, which we call SN-method for short. The crucial step in solving (1) is to identify those inequalities which are active, i.e. the set $\mathcal{A} \subseteq \mathcal{N}$, where the solution to (1) satisfies $x_{\mathcal{A}} = b_{\mathcal{A}}$. Then, with $\mathcal{I} := \mathcal{N} \setminus \mathcal{A}$, we require $\alpha_{\mathcal{I}} = 0$ for (3b) to hold.

The SN-method can be viewed as an infeasible active set method, generating a sequence of active sets $\mathcal{A} \subseteq \mathcal{N}$, until an optimal active set is reached. Specifically, given $\mathcal{A}$ and $\mathcal{I} := \mathcal{N} \setminus \mathcal{A}$, the $2n$ equations (3a), (3b) are solved under the condition $x_{\mathcal{A}} = b_{\mathcal{A}}, \alpha_{\mathcal{I}} = 0$. To simplify notation we introduce for given $\mathcal{A}$ the following set $KKT(\mathcal{A})$ of equations:

$$KKT(\mathcal{A}) \qquad\qquad Qx + q + \alpha = 0, \;\; x_{\mathcal{A}} = b_{\mathcal{A}}, \;\; \alpha_{\mathcal{I}} = 0.$$

The solution of $KKT(\mathcal{A})$ is given by

$$\begin{aligned} x_{\mathcal{A}} &= b_{\mathcal{A}}, & x_{\mathcal{I}} &= -Q_{\mathcal{I}}^{-1}(q_{\mathcal{I}} + Q_{\mathcal{I},\mathcal{A}} b_{\mathcal{A}}), \\ \alpha_{\mathcal{I}} &= 0, & \alpha_{\mathcal{A}} &= -q_{\mathcal{A}} - Q_{\mathcal{A}} b_{\mathcal{A}} - Q_{\mathcal{A},\mathcal{I}} x_{\mathcal{I}}. \end{aligned}$$

We also write $[x, \alpha] = KKT(\mathcal{A})$ to indicate that x and α satisfy $KKT(\mathcal{A})$. In some cases we also write $x = KKT(\mathcal{A})$ to emphasize that we only use x and therefore do not need to compute α. When we only determine α, we write $\alpha = KKT(\mathcal{A})$, and it is assumed that the corresponding x is available. The set $\mathcal{A}$ is called **optimal** if $[x, \alpha] = KKT(\mathcal{A})$ also satisfies $x \leq b, \alpha \geq 0$ since x is then the unique solution of problem (1).

Otherwise a new iteration is started with the following active set

$$\mathcal{A}^+ := \{i \in \mathcal{A} : \alpha_i > 0\} \cup \{i \notin \mathcal{A} : x_i > b_i\}. \tag{4}$$

If (x, α) satisfies $KKT(\mathcal{A})$, then the two equations (3a) and (3b) of the KKT system hold. The iterates of the algorithm are well defined, because $KKT(\mathcal{A})$ has a unique solution for every $\mathcal{A} \subseteq \mathcal{N}$, due to $Q \succ 0$. We summarize the workings of the SN-method in Table 1.

```
SN-method
Input: $\mathcal{A} \subseteq \mathcal{N}$
repeat
    $[x, \alpha] = KKT(\mathcal{A})$.
    $A \leftarrow \{i \in \mathcal{N} : x_i > b_i$ or $\alpha_i > 0\}$.
until $(x, \alpha)$ is optimal
```

Table 1: Semismooth-Newton-Method for (1).

We note that the solution x of $KKT(\mathcal{A})$ is the optimal solution of

$$\min\{J(x) : x_{\mathcal{A}} = b_{\mathcal{A}}\}. \tag{5}$$

We close this section with the following three small numerical examples to further clarify the workings of the SN-method and showcase instances on which it succeeds or fails.

Example 1. *For the following instance of dimension 3 the SN-method converges independent from the starting active set and yields the active-set-transition-graph depicted in Figure 1:*

$$Q = \begin{pmatrix} 1 & 1 & 1/2 \\ 1 & 2 & 1/3 \\ 1/2 & 1/3 & 3 \end{pmatrix}, \qquad q = \begin{pmatrix} -2 \\ -2 \\ -2 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 0.4 \\ 1 \end{pmatrix}.$$

$$\begin{tikzcd}
\mathcal{N} \arrow[dr, bend left] & & \{1,2\} \arrow[loop above] & & \{3\} \arrow[dl, bend right] \\
\{2,3\} \arrow[r] & \emptyset \arrow[r] & \{1\} \arrow[u] & \{2\} \arrow[l] & \{1,3\} \arrow[l]
\end{tikzcd}$$

Figure 1: Active-set-transition-graph of the SN-method for example 1.

Example 2. *Next let us recall the example from Curtis et al. [6] that causes the SN-method to cycle for 6 out of the 8 possible starting active sets. We are given the following problem data:*

$$Q = \begin{pmatrix} 4 & 5 & -5 \\ 5 & 9 & -5 \\ -5 & -5 & 7 \end{pmatrix}, \quad q = \begin{pmatrix} 2 \\ 1 \\ -3 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

The corresponding active-set-transition-graph is depicted in Figure 2.

$$\begin{tikzcd}
\{2,3\} \arrow[loop above] & \{1,3\} \arrow[r] & \{3\} \arrow[dr, bend right] & \emptyset \arrow[d] \\
\{1\} \arrow[u] & \{1,2\} \arrow[r] & \mathcal{N} \arrow[u] & \{2\} \arrow[l]
\end{tikzcd}$$

Figure 2: Active-set-transition-graph of the SN-method for example 2.

Example 3. *Finally let us give another larger example of dimension 12. The condition number cond*$(Q) \approx 314$*. This example will be used later on to compare the SN-method with its modified, globally convergent versions suggested in the following sections. Let*

$$Q = \frac{1}{100}\begin{pmatrix}
102 & 1 & -105 & -1 & 182 & -2 & -1 & -23 & 1 & 0 & 0 & 0 \\
1 & 92 & 1 & 16 & 56 & -27 & -16 & 3 & -13 & 1 & 10 & -2 \\
-105 & 1 & 114 & 0 & -196 & -4 & 1 & 25 & 0 & 1 & 1 & 0 \\
-1 & 16 & 0 & 36 & 1 & -2 & 5 & -5 & -26 & 1 & 1 & -3 \\
182 & 56 & -196 & 1 & 541 & 206 & -53 & -121 & 14 & -5 & 9 & -1 \\
-2 & -27 & -4 & -2 & 206 & 427 & -43 & -123 & 4 & 0 & 5 & -4 \\
-1 & -16 & 1 & 5 & -53 & -43 & 130 & 12 & -2 & 1 & 14 & 3 \\
-23 & 3 & 25 & -5 & -121 & -123 & 12 & 218 & -13 & -9 & 4 & 8 \\
1 & -13 & 0 & -26 & 14 & 4 & -2 & -13 & 339 & 11 & 15 & -6 \\
0 & 1 & 1 & 1 & -5 & 0 & 1 & -9 & 11 & 590 & 82 & -3 \\
0 & 10 & 1 & 1 & 9 & 5 & 14 & 4 & 15 & 82 & 685 & -13 \\
0 & -2 & 0 & -3 & -1 & -4 & 3 & 8 & -6 & -3 & -13 & 457
\end{pmatrix},$$

$$q = \begin{pmatrix} 1698 & 9728 & -8768 & 1601 & 26494 & 11490 & -3940 & -5555 & -527 & -18 & 968 & -83 \end{pmatrix}^\top$$

and $b = e$*, the all-ones vector. Starting with the active set* $\mathcal{A} = \{1, 2, 3, 6, 9, 11, 12\}$ *the algorithm generates the following active sets* $\mathcal{B} = \{1, 7, 9\}$, $\mathcal{C} = \{1, 3, 4, 5, 7, 8, 10, 11\}$, $\mathcal{D} = \{3, 4, 7, 8, 10, 11\}$ *and then the set* $\mathcal{A}$ *again. The SN-method cycles for about half of all possible starting active sets.*

3 Some basic properties for consecutive SN-iterates

The original SN-method avoids cycling only under some strong form of diagonal dominance of Q, but may cycle otherwise. First we propose a small change of the SN-Update rule from (4) to

$$\mathcal{B} := \{i \notin \mathcal{A} : x_i \geq b_i\} \cup \{i \in \mathcal{A} : \alpha_i \geq 0\}. \tag{6}$$

This change does not affect the convergence properties of the SN-method but will be helpful in the following convergence analysis.

We consider two consecutive iterations of the SN-method. Thus starting with $\mathcal{A} \subseteq N$ we get `[`x, α`]=`KKT`(`$\mathcal{A}$`)`, set $u = \min(x, b)$ and determine $\mathcal{B}$ given by (6). Then we compute `[`y, β`]=`KKT`(`$\mathcal{B}$`)` and set $v = \min(y, b)$. Thus one iteration of the SN-method moves from the configuration $(\mathcal{A}, (x, \alpha), u)$ to the new configuration $(\mathcal{B}, (y, \beta), v)$.

We declare the new configuration **successful** if

$$J(v) < J(u).$$

It is clear that if only successful iterates are generated by the SN-method, then it necessarily terminates after a finite number of iterations as no active set will be generated twice. Unfortunately, this does not happen always, as the cycling example shows. In the next section we describe modifications to the SN-method, which insure global convergence. These will be based on the following observations.

Lemma 4. *Let* $\mathcal{A}, x, \mathcal{B}$ *be as above and set* $u := \min(x, b)$*. Then* $u_i = b_i \ \forall i \in \mathcal{A} \cup \mathcal{B}$.

Proof. Since $x_{\mathcal{A}} = b_{\mathcal{A}}$ we have $u_{\mathcal{A}} = b_{\mathcal{A}}$. Let $\mathcal{B}_1 = \{i \in \mathcal{A} : \alpha_i \geq 0\}$ and $\mathcal{B}_2 = \{i \notin \mathcal{A} : x_i \geq b_i\}$, hence $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$. Since $\mathcal{B}_1 \subseteq \mathcal{A}$ we have $u_{\mathcal{B}_1} = b_{\mathcal{B}_1}$. On the other hand $u_{\mathcal{B}_2} = b_{\mathcal{B}_2}$ due to the definition of u. ☐

In the following, $\mathcal{B} \subseteq \mathcal{N}$ is an arbitrary set and y=$KKT(\mathcal{B})$. We also use an arbitrary feasible point u ($u \leq b$) with $u_{\mathcal{B}} = b_{\mathcal{B}}$. The crucial structure for our modifications is the line segment joining u and y:

$$z(\lambda) := \lambda u + (1 - \lambda) y \text{ for } 0 \leq \lambda < 1. \tag{7}$$

It is rather easy to see that $J(z(\lambda))$ is strictly monotonically decreasing from u to y in case $u \neq y$. We now show that $z(\lambda)$ is in fact the optimal solution for a slightly perturbed objective function with respect to the constraint $z_{\mathcal{B}} = b_{\mathcal{B}}$.

Lemma 5. *Let* $\mathcal{B} \subseteq \mathcal{N}$ *and* y=$KKT(\mathcal{B})$*. Further let* $u \leq b$ *and* $u_{\mathcal{B}} = b_{\mathcal{B}}$*. Then* $z(\lambda)$ *given by* (7) *is a solution of*

$$\min_z \left\{ J(z) + \frac{\lambda}{2(1-\lambda)} (u - z)^T Q (u - z) : z_{\mathcal{B}} = b_{\mathcal{B}} \right\} \tag{8}$$

for all $0 \leq \lambda < 1$.

Proof. Let $\mathcal{J} := \mathcal{N} \setminus \mathcal{B}$. We substitute $z_{\mathcal{B}} = b_{\mathcal{B}}$ so that the objective function in (8) becomes

$$\text{constant} + (-\frac{\lambda}{1-\lambda} Q_{\mathcal{J}} u_{\mathcal{J}} + q_{\mathcal{J}} + Q_{\mathcal{J},\mathcal{B}} b_{\mathcal{B}})^T z_{\mathcal{J}} + \frac{1}{2(1-\lambda)} z_{\mathcal{J}}^T Q_{\mathcal{J}} z_{\mathcal{J}}.$$

Hence the first order optimality condition of (8) is

$$\frac{1}{1-\lambda} Q_{\mathcal{J}} z_{\mathcal{J}} - \frac{\lambda}{1-\lambda} Q_{\mathcal{J}} u_{\mathcal{J}} + q_{\mathcal{J}} + Q_{\mathcal{J},\mathcal{B}} b_{\mathcal{B}} = 0.$$

Solving for $z_{\mathcal{J}}$ gives

$$z_{\mathcal{J}} = \lambda u_{\mathcal{J}} - (1-\lambda)(Q_{\mathcal{J}}^{-1}(q_{\mathcal{J}} + Q_{\mathcal{J},\mathcal{B}}b_{\mathcal{B}})) = \lambda u_{\mathcal{J}} + (1-\lambda)y_{\mathcal{J}}.$$

The last equation follows from $y_{\mathcal{J}} = -Q_{\mathcal{J}}^{-1}(q_{\mathcal{J}} + Q_{\mathcal{J},\mathcal{B}}b_{\mathcal{B}})$. Moreover $u_{\mathcal{B}} = y_{\mathcal{B}} = b_{\mathcal{B}}$ and hence $z_{\mathcal{B}} = b_{\mathcal{B}}$. This shows that $z(\lambda)$ is a solution of (8). □

In the next section we propose modifications to the definition of the active set for the next iteration in case that the SN-Update is not successful. These are based on the properties of the line segment in (7).

4 SN-method with safeguards using combinatorial line search

In this section we explain our modifications to the SN-method to ensure finite termination. An iteration is started with a configuration $(\mathcal{A}, (x, \alpha), u)$. We will maintain the following properties of a configuration during all iterations.

$$Qx + q + \alpha = 0,\ x_{\mathcal{A}} = b_{\mathcal{A}},\ u = \min(x, b).\text{ If } x_i < b_i\ \forall i \notin \mathcal{A} \text{ then } [x, \alpha] = KKT(\mathcal{A}). \tag{9}$$

An iteration is terminated with a new configuration $(\mathcal{A}^+, (x^+, \alpha^+), u^+)$. We label each configuration either 'green' in the case that $[x, \alpha] = KKT(\mathcal{A})$, or 'red' otherwise.

We first test whether the SN Update is successful, meaning that the objective function decreases. If it is not successful, we carry out a safeguard update. Here are the details.

4.1 SN Update

We start a new iteration with the configuration $(\mathcal{A}, (x, \alpha), u)$ which may be either green or red, and determine the SN Update $\mathcal{B}$ by (6). We calculate $[y, \beta] = KKT(\mathcal{B})$ and set $v = \min(y, b)$. If $J(v) < J(u)$ we continue with the configuration $(\mathcal{B}, (y, \beta), v)$. In this case the conditions (9) obviously hold for this update. Moreover, the new configuration is green.

4.2 Safeguard Update

If $J(v) \geq J(u)$ we carry out a Safeguard Update. In this case we modify the SN Update in two ways. Instead of $\mathcal{B}$ given by (6) we use a slightly modified new active set $\mathcal{A}^+$. Moreover we do not insist that (x^+, α^+) is the solution of $KKT(\mathcal{A}^+)$ but only maintain (9). The new active set $\mathcal{A}^+$ will be determined differently depending on whether the current iterate x satisfies $x_i < b_i\ \forall i \notin \mathcal{A}$ (dual update) or not (primal update).

4.2.1 Definition of $\mathcal{A}^+$

Dual Update. If the SN Update was unsuccessful and $x_i < b_i$ for all $i \notin \mathcal{A}$, then we have a primal feasible x, and therefore (x, α) is by assumption (9) the solution of $KKT(\mathcal{A})$. Hence we start with the green configuration, determined by the set $\mathcal{A}$. As the solution is not optimal we must have

$$\min\{\alpha_i : i \in \mathcal{A}\} < 0.$$

Let $j := \arg\min\{\alpha_i : i \in \mathcal{A}\}$. We define

$$\mathcal{A}^+ := \mathcal{A} \setminus j. \tag{10}$$

Remark 1. *Note that the SN Update $\mathcal{B}$ would release all $i \in \mathcal{A}$ with $\alpha_i < 0$ and here we release only one of those variables. Hence $\mathcal{A}^+$ is at least as close to $\mathcal{A}$ as to $\mathcal{B}$ in the sense that*

$$|\mathcal{A}^+ \cap \mathcal{A}| \geq |\mathcal{A}^+ \cap \mathcal{B}|.$$

Primal Update. In the final case we have $x_i \geq b_i$ for at least one index $i \notin \mathcal{A}$. In this case we set

$$\mathcal{A}^+ := \mathcal{A} \cup \{i \notin \mathcal{A} : x_i \geq b_i\}. \tag{11}$$

Remark 2. *Note that the SN Update would be obtained from $\mathcal{A}^+$ by additionally removing all $i \in \mathcal{A}$ with $\alpha_i < 0$. Again $\mathcal{A}^+$ is at least as close to $\mathcal{A}$ as to $\mathcal{B}$,*

$$|\mathcal{A}^+ \cap \mathcal{A}| \geq |\mathcal{A}^+ \cap \mathcal{B}|.$$

Having defined the new active set $\mathcal{A}^+$ in both the Primal and the Dual Update, we now explain how we define the new primal-dual pair (x^+, α^+). We first observe that the current feasible solution u is on the upper bound on $\mathcal{A}$ and $\mathcal{A}^+$.

Lemma 6. *Let $\mathcal{A}, x, \mathcal{A}^+$ be as above and set $u := \min(x, b)$. Then $u_i = b_i \; \forall i \in \mathcal{A} \cup \mathcal{A}^+$.*

Proof. We always have $x_\mathcal{A} = b_\mathcal{A}$, hence $u_\mathcal{A} = b_\mathcal{A}$. In a dual update we have $\mathcal{A}^+ \subset \mathcal{A}$, so the lemma holds. In the other cases, $\mathcal{A}^+$ contains all $i \notin \mathcal{A}$ such that $x_i \geq b_i$. Hence the projection u will be on the upper bound for those i as well. □

We use the new active set $\mathcal{A}^+$ and determine its KKT-solution

$$[y, \beta] = KKT(\mathcal{A}^+).$$

The previous lemma shows that u is on the upper bound on $\mathcal{A}^+$, hence we may use u and y to define the line segment $z(\lambda)$ given in (7). The new primal-dual pair will be determined according to a combinatorial line search on the segment $z(\lambda)$ given by (7).

4.2.2 The new primal-dual pair (x^+, α^+) for the Dual Update

We first investigate the situation in case of a Dual Update, where $x_i < b_i \; \forall i \notin \mathcal{A}$, so $u = x$. We recall that in this case the SN Update $\mathcal{B} = \{i \in \mathcal{A} : \alpha_i \geq 0\} \subset \mathcal{A}$ was not successful and we have $\mathcal{A}^+ = \mathcal{A} \setminus j$, where $j = \arg\min\{\alpha_i : i \in \mathcal{A}\}$. The next lemma shows that the KKT solution $[y, \beta] = KKT(\mathcal{A}^+)$ has $y_j < b_j$ and in particular $y \neq u$.

Lemma 7. *Let $(x, \alpha) = KKT(\mathcal{A})$ with $x_i < b_i$ for $i \notin \mathcal{A}$ and suppose x is not optimal. Let the new active set $\mathcal{B} = \mathcal{A} \setminus \mathcal{C}$ be given with $\mathcal{C} \subseteq \{i \in A : \alpha_i < 0\}$ and $\mathcal{C}$ nonempty. Then there exists $j \in \mathcal{C}$ such that $y_j < b_j$ holds for $(y, \beta) = KKT(\mathcal{B})$.*

Proof. Due to the assumptions of the lemma and the definition of $\mathcal{B}$ the following implications hold

$$[x, \alpha] = KKT(\mathcal{A}) \Rightarrow \alpha_\mathcal{C} < 0, \qquad [y, \beta] = KKT(\mathcal{B}) \Rightarrow \beta_\mathcal{C} = 0. \tag{12}$$

As $\mathcal{I} \subsetneq \mathcal{J}$ we can substitute the primal variables associated with $\mathcal{I}$ by

$$x_\mathcal{I} = -Q_\mathcal{I}^{-1}(q_\mathcal{I} + Q_{\mathcal{I},\mathcal{B}} b_\mathcal{B} + Q_{\mathcal{I},\mathcal{C}} b_\mathcal{C}), \qquad y_\mathcal{I} = -Q_\mathcal{I}^{-1}(q_\mathcal{I} + Q_{\mathcal{I},\mathcal{B}} b_\mathcal{B} + Q_{\mathcal{I},\mathcal{C}} y_\mathcal{C}). \tag{13}$$

The equations $Qx + q + \alpha = 0$ and $Qy + q + \beta = 0$ hold due to the definition of $KKT(\cdot)$. Using (12) we obtain

$$Q_{\mathcal{C},\mathcal{N}} \begin{pmatrix} b_\mathcal{B} \\ b_\mathcal{C} \\ x_\mathcal{I} \end{pmatrix} + q_\mathcal{C} = -\alpha_\mathcal{C} > 0, \qquad Q_{\mathcal{C},\mathcal{N}} \begin{pmatrix} b_\mathcal{B} \\ y_\mathcal{C} \\ y_\mathcal{I} \end{pmatrix} + q_\mathcal{C} = -\beta_\mathcal{C} = 0. \tag{14}$$

Now applying (13) to (14) yields

$$\begin{aligned} &Q_{\mathcal{C},\mathcal{B}} b_\mathcal{B} + Q_\mathcal{C} b_\mathcal{C} - Q_{\mathcal{C},\mathcal{I}} Q_\mathcal{I}^{-1}(q_\mathcal{I} + Q_{\mathcal{I},\mathcal{B}} b_\mathcal{B} + Q_{\mathcal{I},\mathcal{C}} b_\mathcal{C}) + q_\mathcal{C} > \\ &Q_{\mathcal{C},\mathcal{B}} b_\mathcal{B} + Q_\mathcal{C} y_\mathcal{C} - Q_{\mathcal{C},\mathcal{I}} Q_\mathcal{I}^{-1}(q_\mathcal{I} + Q_{\mathcal{I},\mathcal{B}} b_\mathcal{B} + Q_{\mathcal{I},\mathcal{C}} y_\mathcal{C}) + q_\mathcal{C}. \end{aligned}$$

Simplifying the above inequality gives

$$(Q_{\mathcal{C}} - Q_{\mathcal{C},\mathcal{I}} Q_{\mathcal{I}}^{-1} Q_{\mathcal{I},\mathcal{C}})(b_{\mathcal{C}} - y_{\mathcal{C}}) > 0. \tag{15}$$

Therefore we have $b_{\mathcal{C}} \neq y_{\mathcal{C}}$. The matrix $(Q_{\mathcal{C}} - Q_{\mathcal{C},\mathcal{I}} Q_{\mathcal{I}}^{-1} Q_{\mathcal{I},\mathcal{C}})$ is positive definite due to the Schur-complement lemma. Hence we can conclude

$$(b_{\mathcal{C}} - y_{\mathcal{C}})^{\top} (Q_{\mathcal{C}} - Q_{\mathcal{C},\mathcal{I}} Q_{\mathcal{I}}^{-1} Q_{\mathcal{I},\mathcal{C}})(b_{\mathcal{C}} - y_{\mathcal{C}}) > 0. \tag{16}$$

Combining (15) and (16) shows that at least one component of $b_{\mathcal{C}} - y_{\mathcal{C}}$ must be positive. $\square$

In case that the solution y of $KKT(\mathcal{A}^+)$ is primal feasible, we set

$$(x^+, \alpha^+) := (y, \beta), \; u^+ := y.$$

Note that (9) holds for this update, and moreover $J(u^+) < J(u)$, as $x = u \neq y$. Thus the new configuration will be green.

If y is not feasible, then the set $\mathcal{D} = \{i : y_i > b_i\}$ is non-empty. For each $i \in \mathcal{D}$ we define

$$\lambda_i := \frac{y_i - b_i}{y_i - u_i} > 0.$$

The previous lemma with $\mathcal{C} = \{j\}$ shows that $\mathcal{D} \subseteq \mathcal{N} \setminus \mathcal{A}$. Moreover, we have $x_i < b_i \; \forall i \in \mathcal{N} \setminus \mathcal{A}$, hence $u_i = x_i < b_i$ and therefore $0 < \lambda_i < 1$. At each of the values λ_i we have $z(\lambda_i) = b_i$. In other words, at $z(\lambda_i)$, the line segment passes through the upper bound b_i. In each of these points we consider the projection $w_i := \min(b, z(\lambda_i))$ and compute $J(w_i)$. We let $z(\lambda^*)$ denote the point on the line segment whose projection yields the smallest objective value. Thus we set

$$\lambda^* = \arg\min_{i \in \mathcal{D}} J(w_i). \tag{17}$$

Determining λ^* in this way can be viewed as a combinatorial line search of the projection of $z(\lambda)$ to the feasible set with respect to the objective function J. To facilitate a good intuition of the combinatorial line search suggested above we depict a 2-dimensional variant in Figure 3.

Since $\lambda^* < 1$ we have with $w^* = \min(b, z(\lambda^*))$ that $J(w^*) < J(u)$. This is in fact the reason for defining the Dual Update condition in this specific way. We finish now the Dual Update step also for the case that $y \not\leq b$ and set

$$x^+ := z(\lambda^*), \; \alpha^+ = Qx^+ - q, \;\; u^+ := \min(x^+, b).$$

We note that also in this case condition (9) is satisfied, but the configuration is red. Moreover, after a Dual Update step we have

$$J(u^+) < J(u).$$

We summarize the previous analysis in the following lemma.

Lemma 8. *Suppose that a Dual Update step is started from the green configuration $(\mathcal{A}, (x, \alpha), u)$ where $x_i < b_i$ for $i \notin \mathcal{A}$, resulting in the updated configuration $(\mathcal{A}^+, (x^+, \alpha^+), u^+)$. Then $J(u^+) < J(u)$.*

4.2.3 The new primal-dual pair (x^+, α^+) for the Primal Update

Finally, we need to explain how we proceed in case of a Primal Update. In this case we proceed similar to the Dual Update. Thus we determine $[y, \beta] = KKT(\mathcal{A}^+)$ and use u and y to investigate the line segment $z(\lambda)$. In this case we cannot avoid the possibility of having $u^+ = u$. If $y \leq b$ we proceed with

$$(x^+, \alpha^+) \leftarrow (y, \beta), \; u^+ = y.$$

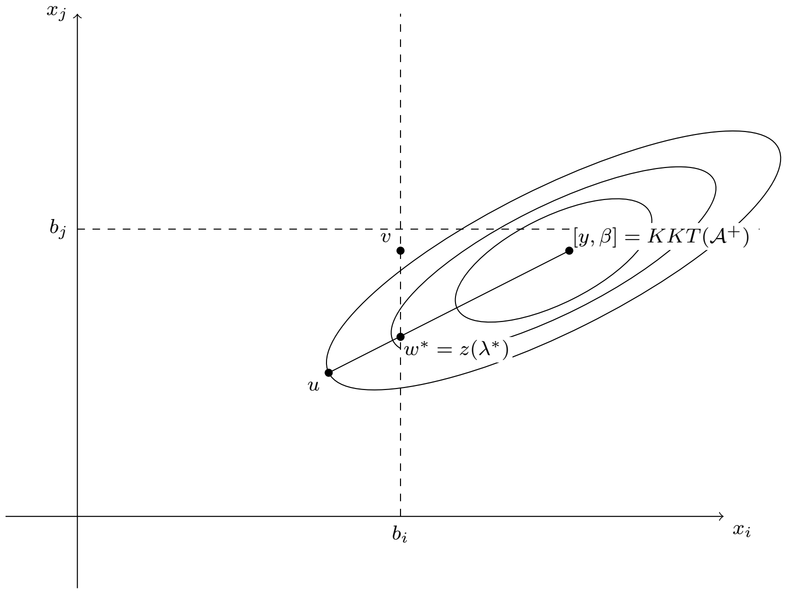


Figure 3: 2-dimensional variant of the combinatorial line search in a Dual Update: The primal-dual pair $(y, \beta) = KKT(\mathcal{A}^+)$ violates the bound b_i. The ellipses represent the level sets of J with global minimum at y. The projection v of y on the feasible region has a higher objective value than u, $J(v) > J(u)$.

In this case we have $J(u^+) \leq J(u)$ and the new configuration is green. Note that the case $u^+ = u$ is linked to the situation depicted in Figure 4 and can only occur in the Primal Update but not in the Dual Update.

Finally, if $y \not\leq b$ we determine λ^* from (17) yielding $z(\lambda^*)$. We continue with

$$x^+ \leftarrow z(\lambda^*),\ \alpha^+ = -Qx^+ - q,\ u^+ = \min(x^+, b).$$

Also in this case condition (9) holds and after a primal update we have $J(u^+) \leq J(u)$.

Summarizing the discussion so far in this section we give an algorithmic description of the modified SN-method (MSN) in Table 2. We also sketch the workings of MSN in the flow chart in Figure 5. We note in particular that a safeguard update requires two solves of a KKT system, while the SN Update requires only one such solve.

4.3 Convergence analysis

We now show global convergence of our modified SN-method. The method generates a sequence of configurations which are either green or red. We emphasize that a green configuration corresponding to some set $\mathcal{A}$ is uniquely determined by $\mathcal{A}$ in the sense that $[x, \alpha] = KKT(\mathcal{A})$ as well as $u = \min(x, b)$ uniquely follow from $\mathcal{A}$.

The key observation is that a sequence of red configurations can have length at most $n - 1$.

Lemma 9. *Let $\mathcal{C}^{(i)} = (\mathcal{A}^{(i)}, (x^{(i)}, \alpha^{(i)}), u^{(i)}),\ i = 1, \ldots, k$ be a sequence of red iterates. Then $k \leq n - 1$ and $J(u^{(1)}) \geq J(u^{(k)})$.*

Proof. We note that an SN Update moves from either a red or a green configuration into a green configuration, hence such an update can not have occured during this sequence. Similarly, a Dual Update moves from a green

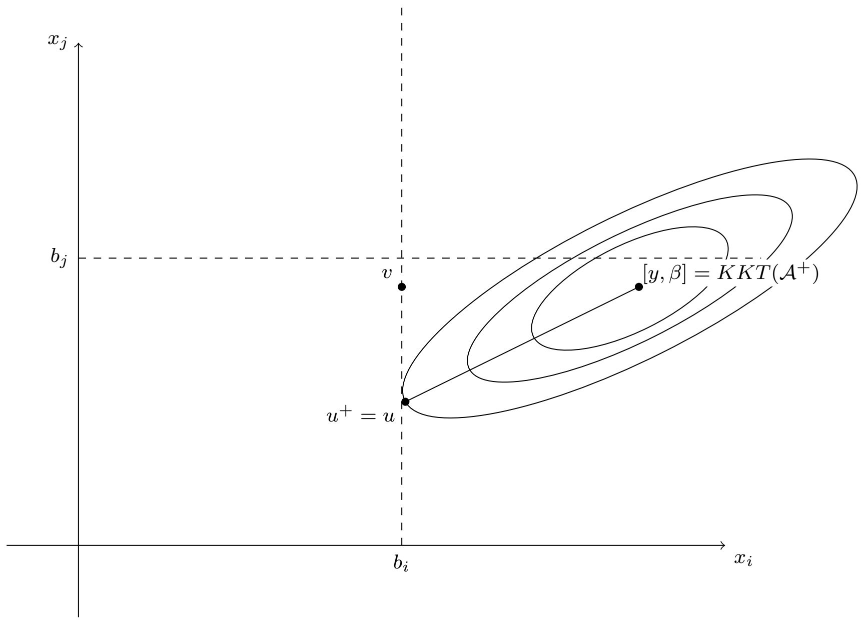


Figure 4: The case $u = z(\lambda)$ can only occur for the Primal Update.

configuration to either a red or a green configuration. Again, such an update can not have occured during this sequence. Thus we have carried out only Primal Updates in the whole sequence. But in each such update, the active set increases by at least one member, so the number of steps is bounded by $n - 1$, as an active set containing all indices gives a green configuration. Monotonicity of the values $J(u^{(i)})$ follows from the combinatorial line search. □

Lemma 10. *An update starting from a green configuration $(\mathcal{A}, (x, \alpha), u)$ to the updated configuration $(\mathcal{A}^+, (x^+, \alpha^+), u^+)$ has $J(u^+) < J(u)$.*

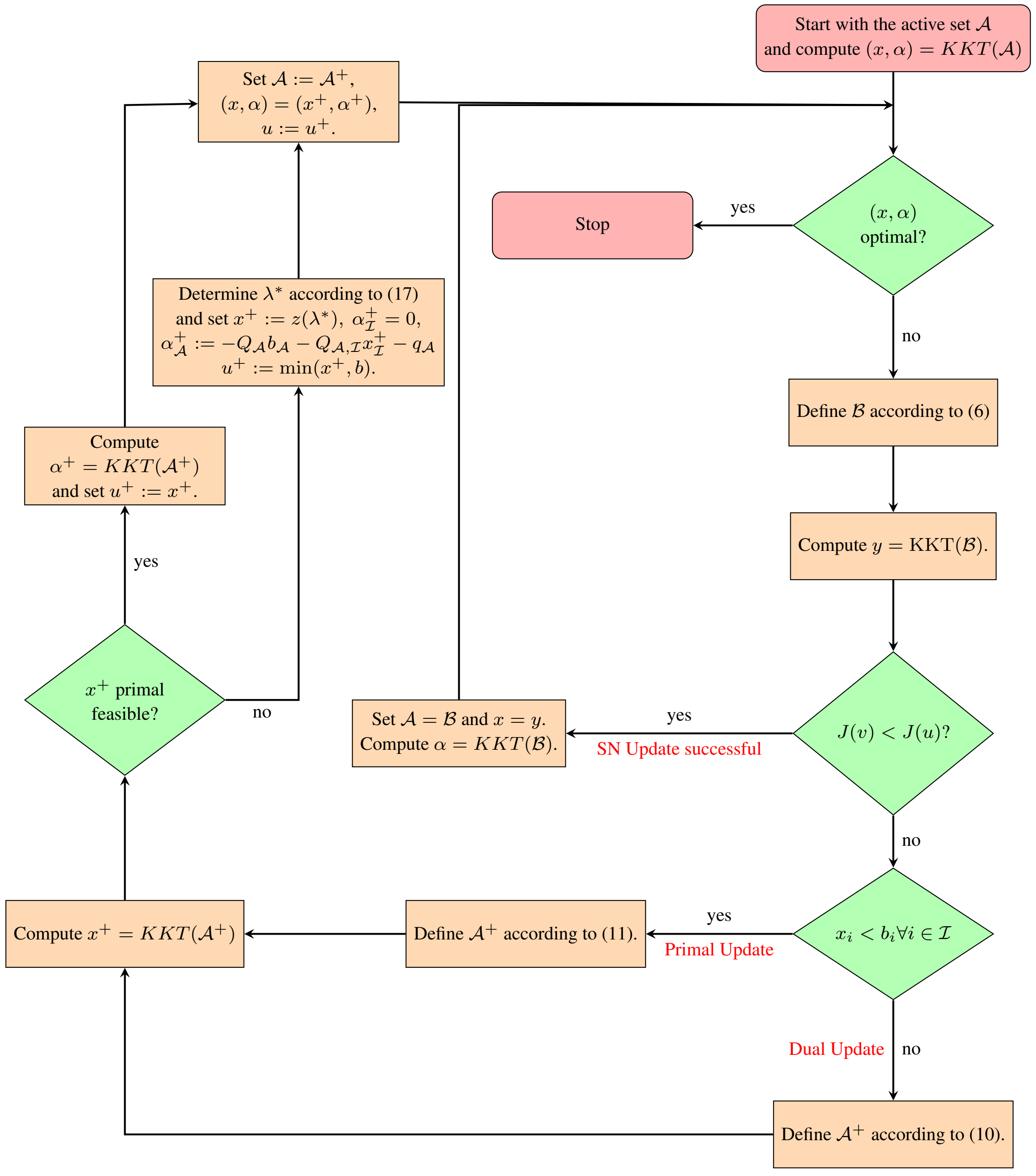
Proof. This follows from the definition in case of a SN Update, and by construction in case of a Dual Update, see Lemma 8. □

Looking at the subsequence of green iterates produced by our algorithm, we note that they yield a strictly monotonically decreasing sequence of function values $J(u)$ for feasible solutions u. Each of these u is determined (uniquely) through the active set $\mathcal{A}$, hence we never generate the same green set twice. Furthermore there can be at most $n - 1$ red configurations between two green configurations. Thus we have proved the following main theorem of this paper.

Theorem 11. *For any $Q \succ 0$ the modified SN-method terminates in a finite number of iterations with an optimal solution.*

Let us close this section with two supplemental remarks.

Remark 3. *The Primal Update might look "slow" as only one variable can leave the active set per iteration. But in practice we hardly ever have to perform the Primal (and also the Dual) Update, for details see the following two sections. In a nutshell we use Primal and Dual Update only as safeguards to escape special bad configuration of the SN-method.*



Start with the active set $\mathcal{A}$ and compute $(x, \alpha) = KKT(\mathcal{A})$
(x, α) optimal?
yes
Stop
no
Define $\mathcal{B}$ according to (6)
Compute $y = \mathrm{KKT}(\mathcal{B})$.
$J(v) < J(u)$?
yes
SN Update successful
Set $\mathcal{A} = \mathcal{B}$ and $x = y$. Compute $\alpha = KKT(\mathcal{B})$.
no
$x_i < b_i \forall i \in \mathcal{I}$
yes
Primal Update
Define $\mathcal{A}^+$ according to (11).
Dual Update
no
Define $\mathcal{A}^+$ according to (10).
Compute $x^+ = KKT(\mathcal{A}^+)$
x^+ primal feasible?
yes
Compute $\alpha^+ = KKT(\mathcal{A}^+)$ and set $u^+ := x^+$.
no
Determine λ^* according to (17) and set $x^+ := z(\lambda^*)$, $\alpha^+_{\mathcal{I}} = 0$, $\alpha^+_{\mathcal{A}} := -Q_{\mathcal{A}} b_{\mathcal{A}} - Q_{\mathcal{A},\mathcal{I}} x^+_{\mathcal{I}} - q_{\mathcal{A}}$ $u^+ := \min(x^+, b)$.
Set $\mathcal{A} := \mathcal{A}^+$, $(x, \alpha) = (x^+, \alpha^+)$, $u := u^+$.

Figure 5: Outline of the workings of MSN.

```
Modified SN-method (MSN)
Start: $\mathcal{A} \subseteq \mathcal{N},\ [x, \alpha] = KKT(\mathcal{A}),\ u = \min(x, b)$

while $[x, \alpha]$ not optimal
    Set $B := \{i \notin \mathcal{A} : x_i \geq b_i\} \cup \{i \in \mathcal{A} : \alpha_i \geq 0\}$
    Compute $y = KKT(\mathcal{B})$ and $v = \min(y, b)$
    Case 1: SN Update $J(v) < J(u)$
        Compute $\beta = KKT(\mathcal{B})$ and continue with $[\mathcal{B}, (y, \beta), v]$
    Case 2: Safeguard Update $J(v) \geq J(u)$
        Determine $\mathcal{A}^+$:
        If $x_i < b_i,\ i \notin \mathcal{A}$, then $\mathcal{A}^+ := \mathcal{A} \setminus \{j\},\ j = \arg\min_{i \in \mathcal{A}} \alpha_i$
        else $\mathcal{A}^+ = \mathcal{A} \cup \{i \notin \mathcal{A} : x_i \geq b_i\}$
        Compute $y = KKT(\mathcal{A}^+)$
        If $y \leq b$ then compute $\beta = KKT(\mathcal{A}^+)$ and continue with $[\mathcal{A}^+, (y, \beta), y]$
        else determine $\lambda^*$ according to (17)
        set $x^+ := z(\lambda^*),\ \alpha^+ = -Qx^+ - q,\ u^+ := \min(x^+, b)$ and continue with $[\mathcal{A}^+, (x^+, \alpha^+), u^+]$
endwhile
```

Table 2: Modified SN-Method (MSN) with safeguards using combinatorial line search.

Remark 4. *Let us emphasize that we cannot guarantee a polynomial running time for the modified SN-method as one might already expect for combinatorial, active set methods.*

5 Speeding-up the modified SN-method in practice

The Safeguard Update described in the previous section keeps the SN-method from cycling. On the other hand we know from practical experience that SN Updates are very efficient, provided they do not run into cycling. We therefore modify our conservative setting and try more aggressively to carry out SN Updates.

Here is our first extension for enhancing the practical performance of the modified SN-method: If the SN Update is rejected (because $J(u^+) \geq J(u)$), we may still apply the combinatorial line search to the SN-iterate in the hope of determining a point with better objective function than the previous iterate. If successful, this would avoid the Safeguard Update. Thus we extend MSN to MSNa by including the following additional case 1a.

```
Case 1a: combinatorial SN Update
    (Given $B$, $y = KKT(B)$ and case 1 failed)
    Determine $\lambda^*$ according to (17)
    Set $x^+ := z(\lambda^*),\ \alpha^+ = -Qx^+ - q,\ u^+ := \min(x^+, b)$
    If $J(u^+) < J(u)$ then continue with $[\mathcal{B}, (x^+, \alpha^+), u^+]$
```

Table 3: MSN-Method with additional combinatorial SN Update.

We observe that a successful termination of case 1a yields the red configuration $(\mathcal{B}, [x^+, \alpha^+], u^+)$. In case $J(u^+) \geq J(u)$ we continue with the Safeguard Update.

Contrary to the primal update, which can occur consecutively at most $n - 1$ times, we were not able to prove that a consecutive sequence of such new iterates has bounded length. We therefore propose to add a counter which counts consecutive executions of case-1a. We declare case-1a also as unsuccessful, if this counter exceeds 500. In our computational experiments this counter always remained significantly below this limit, independent of n.

In the following we will demonstrate that this modification has a major (positive) impact on running time on most of our benchmarking instances. Making the combinatorial SN Update instead of a Safeguard Update has two important advantages:

1. It is faster as no additional linear system has to be solved.
2. SN Updates (with or without line search) normally allow that many primal and dual indices enter and/or leave the active set such that the active set may change substantially compared to the active set of the previous iterate. Hence, in average, SN Updates allow for more progress towards the optimal solution than Safeguard Updates. This typically results in a smaller number of iterations, as demonstrated in our computational experiments, see below.

In order to give a better idea of the workings of both MSN and MSNa, we apply them to the toy examples discussed at the end of Section 2.

Example 12 (Example 1 (continued))**.** *For this example the active-set-transition-graph is depicted in Figure 6 for both MSN and MSNa. The changes with respect to the active-set-transition-graph for the SN-method described in Section 2 only result from the slight difference between SN Update rules* (4) *and* (6)*.*

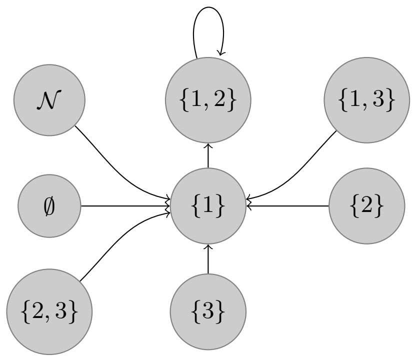


Figure 6: Active-set-transition-graph for both MSN and MSNa for Example 1.

Example 13 (Example 2 (continued))**.** *Contrary to the SN-method both MSN and MSNa terminate, see Figure 7. We now give some details about the transitions:*

- *MSN makes a Dual Update and a Primal Update when going from* $\{1,3\}$ *to* $\{2,3\}$*.*
- *MSNa makes an SN Update with combinatorial line search, a Dual Update and a Primal Update when going from* $\{1,3\}$ *to* $\{2,3\}$*.*
- *Both MSN and MSNa make a Primal Update when going from the* $\{2\}$ *to* $\{1,2,3\}$ *and a Dual Update when going from* $\{1,2,3\}$ *to* $\{2,3\}$*.*

In all other iterations the standard SN Update works, i.e. making an SN Update without combinatorial line search gives a decrease of the objective function.

Example 14 (Example 3 (continued))**.** *Again starting with the active set* $\mathcal{A} = \{1,2,3,6,9,11,12\}$*, MSNa generates the active set* $\mathcal{E} = \{1,3,7,8,11\}$ *and then already stops with the optimal active set* $\mathcal{O} = \{1,3,7,11\}$*. First the SN Update without line search generates the active set* $\mathcal{B} = \{1,7,9\}$ *that is rejected. Next we deploy an SN Update with combinatorial line-search that already yields the optimal active set.*

Starting MSN with $\mathcal{A}$ *we first obtain* $\mathcal{B}$ *via an SN Update. This SN Update is rejected and we go to the active set* $\mathcal{F} := \mathcal{A} \cup \{7\} = \{1,2,3,6,7,9,11,12\}$ *using a Primal Update.*[1] *The corresponding primal-dual pair is primal*

[1]Note that in a Primal Update also more than one index can enter the active set.

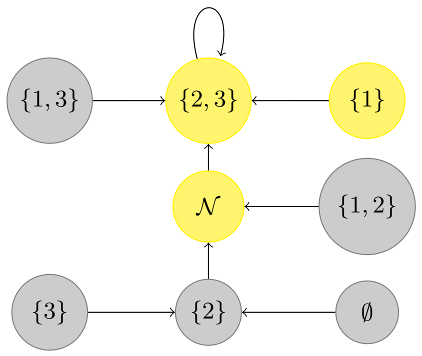


Figure 7: Active-set-transition-graph for both MSN and MSNa for Example 2.

feasible. The next active set is again $\mathcal{B}$ and hence the SN Update is again rejected. Now we have to make a Dual Update, where we release $\{2\}$ from $\mathcal{F}$ to obtain the active set $\mathcal{G} := \{1, 3, 6, 7, 9, 11, 12\}$. The corresponding primal-dual pair is again primal feasible, but $x_i < b_i,\ i \notin \mathcal{A}$ does not hold. The next SN Update that generates the active set $\mathcal{H} := \{1, 7, 8\}$ is again rejected. Now we have to make a Primal Update, where we add $\{8\}$ to $\mathcal{G}$ to obtain the active set $\mathcal{L} := \{1, 3, 6, 7, 8, 9, 11, 12\}$. The next SN Update gives $\mathcal{M} = \{1, 3, 7, 8, 11\}$ and is accepted. Finally another SN Update yields the optimal active set $\mathcal{O}$.

In our computational experiments, which we elaborate on in the following section, we noted that the occurence of safeguard updates often lead to an increase in the number of iterations. This motivated us to investigate even further possibilities to avoid safeguard updates.

Our final extension to stay with SN Updates pushes the update step case-1a to its limit. In case that the combinatorial line search in case-1a fails (because $J(u^+) \geq J(u)$), we carry out a continuous line-search to find the minimum of

$$J(\min\{z(\lambda), b\}) \text{ for } 0 \leq \lambda \leq 1. \tag{18}$$

In contrast to the safeguard updates, for the SN Updates it is possible that the combinatorial line search fails but the continuous one succeeds. Since the function evaluation of J is cheap, we perform this line-search in a heuristic way, and evaluate J at equidistant points $\lambda_i = \frac{i}{100}$ and $i = 0, \dots, 100$. Again, we need to maintain a counter as before, to ensure finite termination. In our computational experiments this counter always remained significantly below this limit of 500, independent of n.

A formal description of case-1b is given in the table below. As before this update is declared unsuccessful if $J(u^+) \geq J(u)$ or if the maximum number of consecutive case-1b iterations is reached, resulting in a safeguard update according to case 2. In the following we show on a variety of different test instances that this last modification yields the overall best performance. We denote by MSNb the inclusion of both case-1a and case-1b in MSN.

```
Case 1b: continuous SN Update
    (Given $B$, $y = KKT(B)$ and case 1 and case 1a failed)
    Determine $\lambda^*$ according to (18)
    Set $x^+ := z(\lambda^*),\ \alpha^+ = -Qx^+ - q,\ u^+ := \min(x^+, b)$
    If $J(u^+) < J(u)$ then continue with $[\mathcal{B}, (x^+, \alpha^+), u^+]$
```

Table 4: MSN-Method with additional continuous SN Update.

We close this section with computational comparisons of the original SN-method [SN] with our modified versions [MSN], [MSNa] and [MSNb] which guarantee finite termination. We also compare with the globally convergent version of the SN-method with primal feasible iterates and recursion ([RSN]) proposed in Hungerländer and Rendl [18].

In all the tables below, the column labels have the following meaning.

- The condition number of Q is estimated with the MATLAB command `condest`. Its average is given in the column labeled $cond(Q)$.
- In the column labeled *solve* we show how often in average we solve $x = KKT(\mathcal{A})$ to determine the primal inactive variables. We recall that this reflects the main computational effort of our method.
- The column labeled *fail* contains the number of trials, where [SN] ran into a cycle and failed to solve the problem.
- The column labeled *error* gives the relative error of the function *quadprog* with respect to the optimal objective value, *error* $=$ (objective value found - optimal objective value)/(optimal objective value). We do not include similar error measures for other methods as the linear systems are solved to machine accuracy 10^{-15} by Matlab.
- In some tables we also provide the average density of Q, which is denoted by *dens*.
- All experiments were conducted on an Intel Xeon 5160 processor with 3 GHz and 2 GB RAM. The corresponding Matlab code for generating all instances and solving them with the various methods discussed is available from `http://philipphungerlaender.jimdo.com/qp-code/`.

Example 15. *We close this section with an interesting class of instances taken from [25]. It is given as follows. Let*

$$L = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 2 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 2 & \dots & 2 & 1 \end{pmatrix}$$

be $n \times n$ with $n \geq 2$. We consider

$$\min \frac{1}{2} x^T L L^T x + e^T x \text{ such that } x \leq 0.$$

It is easy to verify that $x = -e_1$, and $\alpha = e - e_1$ is the unique optimal solution, with optimal active set $A_{opt} = \{2, \dots, n\}$.

We investigated the SN transition graph for $n \leq 5$ and noted that in all cases it turned out to be a complete binary tree rooted at A_{opt}. We conjecture that this is true for all n. Since a complete binary tree on 2^n nodes has 2^{n-1} leaves, we would expect that for half of all possible starting active sets, the number of SN iterations should be $n+1$. This is confirmed by our computational results given below. For instance starting with $A = N$, it always took exactly $n+1$ iterations and n system solves for the original SN-method to reach the optimal solution. This is the only example that we are aware of where the SN-method does not cycle, and takes a number of iterations proportional to n. It is interesting to note that the globalized versions of the SN method terminate with significantly fewer iterations than the original SN method.

Now let us also consider the following randomly generated problems (see [22, Section 4.1.]), where we vary the matrix Q by making it increasingly ill-conditioned. The random problems of size n = 2000 are generated so that Q_0 is a sparse positive semidefinite matrix, which is singular. The matrix Q is obtained from Q_0 by adding a small multiple of the identity matrix I:

$$Q = Q_0 + \epsilon I,$$

where $\epsilon \in \{1, 10^{-5}, 10^{-10}, 10^{-14}\}$. We state the MATLAB commands used to produce these random benchmarking instances:

```
» p = sprandn(n,n,.1)+speye(n); p = tril(triu(p,-100));
» Q0 = p*p'; Q = Q0 + ϵ*eye(n); q=rand(n,1)*20*n-10*n; b=ones(n,1);
```

	$\mathcal{A}^0 = \mathcal{N}$					$\mathcal{A}^0$ random				
n	[SN]	[RSN]	[MSN]	[MSNa]	[MSNb]	[SN]	[RSN]	[MSN]	[MSNa]	[MSNb]
	solve	solve	solve	solve	solve	solve	solve	solve	solve	solve
50	50	7	2	2	2	48.5	6.2	10.5	10.7	11.0
100	100	8	2	2	2	99.4	8.7	11.5	10.9	10.7
500	500	10	2	2	2	500.4	10.2	11.6	10.6	10.6
1000	1000	10	2	2	2	999.6	11.4	12.9	12.0	12.0

Table 5: Comparison of SN, RSN, MSN and MSNa for varying choices of n and starting active sets A_0.

cond(Q)	log(ϵ)	[SN]	[RSN]	[MSN]	[MSNa]	[MSNb]
		solve	solve	solve	solve	solve
$\approx 10^3$	0	4.84	9.48	4.84	4.84	4.84
$\approx 10^8$	-5	8.03	12.10	307.83	64.43	10.51
$\approx 10^{13}$	-10	8.05	15.06	311.86	64.94	10.69
$\approx 10^{17}$	-14	8.05	15.08	311.65	64.97	10.68

Table 6: Comparison of SN, RSN, MSNa and MSNb on sparse random data with one-sided bounds. Initial active set is $\mathcal{N}$ and $n = 2000$. Each line is the average over 2000 trials.

We note that the SN-method never cycles in these experiments even though in principle this is possible. For ill-conditioned problems the number of linear systems solved is slightly smaller for [SN] than for [MSNb] at the cost of risking to run into a cycle. It is also quite impressive that the inclusion of case-1a and case-1b to enhance [MSN] leads to a substantial drop in iterations, from 311 for [MSN] to 65 for [MSNa] and further to 11 for [MSNb]. [MSNb] and [RSN] show similar performance with a slight advantage for [MSNb] on these random instances.

As a first conclusion, we see a clear dominance from [MSN] to [MSNa] to [MSNb], and so from now on we use [MSNb] as the strongest candidate for comparison with other methods. This conclusion is supported by all our computational experiments also with other classes of instances. In the following section we provide extensive computational comparisons with other methods on random instances as well as structured problems coming from applications in mathematical physics and combinatorial optimization.

6 Computational Results

In this section we computationally compare the SN-method ([SN]) with the newly proposed globalized version [MSNb] and also with [RSN], a modification of the SN-method using recursion to avoid cycling, see Hungerländer and Rendl [18]. We also compare with the MATLAB function `quadprog` and the interior-point and active-set solvers [CPLEX-IP] and [CPLEX-AS] from ILOG CPLEX 12.1 see [19]. We do not consider further methods, which are shown in [18] to be dominated by [RSN].

6.1 Random Data

In Table 7 we compare [RSN] and [MSNb] to the interior point and active set solvers of ILOG CPLEX 12.1 [19]. We use the same random data as above with $\epsilon \in \{1, 10^{-10}\}$ for various sizes n and take the average over 10 instances.

Both [MSNb] and [RSN] are superior to both CPLEX methods in terms of computation times, where the relative gap between the methods is growing with problem size. The number of system solves for the interior-point method and also [MSNb] and [RSN] doubles from $n = 2000$ to $n = 50000$, and stays below 33 in all cases. Again [MSNb] is sligthly favourable to [RSN].

n	cond(Q)	[CPLEX-IP]		[CPLEX-AS]		[RSN]		[MSNb]	
		iterations	time	iterations	time	solve	time	solve	time
2000	$\approx 10^3$	16.2	1.34	1000	0.49	9.3	0.29	4.9	0.19
2000	$\approx 10^{13}$	21.4	1.69	1063	0.53	14.2	0.42	8.2	0.41
10000	$\approx 10^3$	20.8	8.95	5024	8.03	11.2	1.77	5.1	1.26
10000	$\approx 10^{13}$	27.1	11.25	5273	9.04	15.6	2.44	8.6	2.30
50000	$\approx 10^3$	26.2	66.78	25243	99.86	11.6	8.32	5.4	4.91
50000	$\approx 10^{13}$	32.1	80.13	26263	108.13	16.0	10.68	8.8	8.16

Table 7: Comparison of [RSN] and [MSNb] with IP and AS solvers on sparse random data with one-sided bounds. Initial active set is $\mathcal{N}$. Each line is the average over 10 trials.

Next we consider randomly generated problems in the style of [6] in Table 8. Specifically, we generated Q via Matlab's `sprandsym` routine. Both the average density and the average condition number are given in the following two tables. Then we generated the optimal solution x via the Matlab `randn` routine. With the solution in hand, we choose the objective linear term coefficients, lower and upper bounds so that at the optimal solution the number of

- the active variables and the inactive variables are roughly equal for the case with one-sided bounds and
- the active variables on the upper bound, the active variables on the lower bound and the inactive variables are all roughly equal for the case with box-constraints.

We consider the same sizes n as in [6], but allow the condition number to go up to $\approx 10^{10}$ as compared to $\approx 10^6$ in [6]. Since the code from [6] is not available to us, we can not make a direct comparison. In Table 8 we compare [SN], [RSN], [MSNb] and Matlab's function `quadprog`. Each line represents the average over 10 trials.

n	dens	cond(Q)	[SN]			[RSN]		[MSNb]			[quadprog]		
			time	iter	fail	time	solve	time	solve	max c	time	iter	error
1000	0.1	$\approx 10^2$	0.17	4.7	0	0.25	6.4	0.17	5.1	0	0.23	13.5	10^{-10}
1000	0.1	$\approx 10^6$	0.46	15.4	1	0.73	33.3	1.08	40.1	34	1.12	43.1	10^{-6}
1000	0.1	$\approx 10^{10}$	1.49	51.5	8	1.37	76.3	2.95	104.7	42	4.19	82.2	10^{-5}
5000	0.1	$\approx 10^2$	9.23	4.6	0	15.96	6.9	13.27	5.5	0	6.14	15.2	10^{-15}
5000	0.1	$\approx 10^6$	15.35	13.7	0	33.77	45.6	142.06	83.10	72	40.00	55.0	10^{-5}
5000	0.1	$\approx 10^{10}$	33.06	33.4	0	66.17	124.4	537.78	325.00	171	163.75	113.4	10^{-5}
5000	0.01	$\approx 10^2$	3.79	4.4	0	6.51	6.3	5.10	5.3	0	0.73	13.7	10^{-11}
5000	0.01	$\approx 10^6$	4.71	16.8	4	10.64	47.0	17.79	66.9	64	6.78	58.7	10^{-6}
5000	0.01	$\approx 10^{10}$	-	-	10	15.05	96.0	53.35	222.7	111	40.80	112.6	10^{-5}
10000	0.001	$\approx 10^2$	3.09	4.9	0	4.66	6.4	3.33	5.2	0	0.60	12.6	10^{-9}
10000	0.001	$\approx 10^6$	1.77	11.0	9	9.31	25.3	11.12	38.6	18	6.40	51.6	10^{-6}
10000	0.001	$\approx 10^{10}$	-	-	10	16.81	52.8	24.02	121.9	75	47.32	82.2	10^{-5}

Table 8: Comparison of [SN], [RSN], [MSNb] and Matlab's function `quadprog` on random data in the style of [6] with one-side bounds. Initial active set is $\mathcal{N}$. Each line is the average over 10 trials.

These are difficult examples for the active set methods. In particular, [SN] fails completely, once the condition number and the problem size increases. Here the globalizations of [SN] given by [RSN] and [MSNb] show their full power. It is interesting to note that [RSN] is the clear champion for the ill-conditioned cases, in which [SN] cycles. We also note that case-1b has long consecutive calls, the longest sequence given in the column labeled 'max c'. The number of system solves is also much larger for both [RSN] and [MSNb] than in all other classes of instances considered in this paper. Compared to Matlab's large scale solver, [RSN] is clearly faster and also more accurate on all problems.

A similar situation occurs if we consider the above random data also for the box-constrained case of problem (1). In the appendix, we argue that the appropriate extensions of [MSNb] to the box-constrained case insure finite termination. We summarize the performance of the different methods in Table 9. All methods considered show similar behaviour as in the case with one-sided bounds and hence the above discussion on the interpretation of the results is also valid for box-constraint problems of this type.

n	dens	cond(Q)	[SN]			[RSN]		[quadprog]			[MSNb]		
			time	iter	fail	time	solve	time	iter	error	time	solve	max c
1000	0.1	$\approx 10^2$	0.30	6.5	0	0.39	14.2	0.43	17.1	10^{-9}	0.35	10.0	0
1000	0.1	$\approx 10^6$	1.32	26.7	1	1.03	49.1	2.22	35.3	10^{-7}	1.37	50.0	39
1000	0.1	$\approx 10^{10}$	-	-	10	2.08	128.8	4.25	51.8	10^{-6}	2.99	100.1	75
5000	0.1	$\approx 10^2$	13.04	6.1	0	16.68	12.0	16.32	19.0	10^{-11}	15.17	10.1	0
5000	0.1	$\approx 10^6$	43.97	26.1	0	43.21	66.5	115.03	40.6	10^{-7}	93.94	64.6	93
5000	0.1	$\approx 10^{10}$	-	-	10	104.20	258.1	401.83	98.1	10^{-6}	349.4	231.1	193
5000	0.01	$\approx 10^2$	4.57	6.2	0	5.83	10.6	2.19	18.9	10^{-10}	5.42	9.8	1
5000	0.01	$\approx 10^6$	8.64	27.6	5	8.83	57.1	11.34	42.1	10^{-7}	11.32	54.70	36
5000	0.01	$\approx 10^{10}$	-	-	10	21.11	157.9	31.00	58.1	10^{-6}	23.60	112.6	87
10000	0.001	$\approx 10^2$	3.28	6.0	0	3.11	9.6	1.71	17.2	10^{-9}	3.02	8.4	1
10000	0.001	$\approx 10^6$	-	-	10	5.07	31.5	9.32	47.4	10^{-6}	5.41	30.7	30
10000	0.001	$\approx 10^{10}$	-	-	10	10.18	62.7	16.49	49.2	10^{-5}	11.96	90	95

Table 9: Comparison of [SN], [RSN], [MSNb] and Matlab's function `quadprog` on random data in the style of [6] with box constraints. Initial active set is $\mathcal{N}$. Each line is the average over 10 trials.

6.2 Harmonic Equation and Biharmonic Equation

We now consider problems where the data are not random, but come from applications in mathematical phyics. There are several problem types which lead to optimization problems of the form (1), where the Hessian Q_m represents a discretization of the Laplace operator, acting on a square grid of size $m \times m$. The linear term and the boundary conditions describe the precise nature of the problem. Moré and Toraldo [24] discuss several such applications. Instances of this type are also contained in the CUTEr [13] and CUTEst [14] test library and are denoted by "OBSTCL*". In this collection however we have $m \leq 125$. The elastic-plastic torsion problem as well as the journal bearing problem from [24] both have the same Hessian Q_m. We do not include computational results for these classes, as they are very similar to the obstacle problem. We summarize the results of our runs on obstacle problems with one-sided and two-sided bounds in Tables 10 and 11 respectively.

m	largest system	[SN], [MSNb]		[RSN]	
		solve	time	solve	time
64	1519	4	0.02	5	0.03
128	6385	4	0.11	5	0.16
256	26236	4	0.65	5	0.81
512	106184	4	3.97	5	5.27

Table 10: Obstacle problem from [24] with one-sided bounds. The matrix Q_m represents a discretization of the harmonic operator acting on a square grid of size $m \times m$.

m	largest system	[SN], [MSNb]		[RSN]	
		solve	time	solve	time
64	3000	5	0.08	8	0.10
128	12573	4	0.22	6	0.34
256	51484	5	1.42	7	2.48
512	208357	5	9.10	7	11.85

Table 11: Obstacle problem from [24] with box constraints. The matrix Q_m represents a discretization of the harmonic operator acting on a square grid of size $m \times m$.

As a second class of problems with a physical interpretation we consider a thin elastic square plate, clamped on its boundary with a vertical force acting on it. The plate deforms but is constrained to remain below an obstacle. Discretizing again the square into an $m \times m$ rectangular grid, we obtain a problem of the form (1), with matrix Q_m of order m^2 representing the discretized biharmonic operator. For specific details about the force acting on the plate and the obstacle, we refer to [22]. We conducted experiments for m going up to 512 and summarize the results in Table 12.

m	largest system	[SN], [MSNb]		[RSN]	
		solve	time	solve	time
8	26	6	0.00	6	0.00
16	139	7	0.01	12	0.01
32	647	10	0.04	15	0.06
64	2738	6	0.13	7	0.17
128	11251	7	0.79	8	0.92
256	45580	8	5.56	12	7.79
512	183486	12	44.12	20	70.81

Table 12: Biharmonic Equation: Comparison with [22]. Here the matrix Q_m is the discretisation of the biharmonic operator on a square $m \times m$ grid of order m^2.

SN has no difficulty with this problem and MSNb shows identical performance to SN and better performance than RSN. We note that even though the dimension m^2 as well as the largest system being solved is quite large, we are still able to solve this problem rather quickly, because of the sparse structure of Q_m.

6.3 Sparse Quadratic 0-1 Optimization Problems by Pardalos and Rodgers

The box-constrained version of the problem has also applications in combinatorial optimization. Quadratic 0-1 optimization is one of the fundamental NP-hard problems in discrete optimization. It consists of minimizing $J(x)$ on $\{0,1\}^n$, but Q need not be semidefinite. The integrality conditions on x imply that $x_i^2 = x_i$ can be used to convexify the objective function. A possible convex relaxation is to minimize

$$\frac{1}{2}x^T(Q - \lambda_{\min}(Q)I)x + (q + \lambda_{\min}(Q)e)^T x \tag{19}$$

on the box $0 \leq x \leq 1$. Both objective functions agree on 0-1 vectors x. This opens the way to solve the original integer problem by solving the relaxation (19) using Branch-and-Bound techniques. This amounts to solving a problem of type (19) in every node of the branching tree. The nodes of the branching tree indicate which of the variables are fixed to their bounds. Since the number of nodes in such branching trees may become quite large, it is critical to solve each subproblem as quickly as possible. For more details on solving quadratic 0-1 problems see e.g. [3]. In Table 13 we compare RSN and MSNb with the interior-point solver of CPLEX. Both active set methods are faster by an order of magnitude, especially for larger instances, where MSNb is slightly faster than RSN. Q was generated by the Sparse Problem Generator of Pardalos and Rodgers [28]. Its diagonal coefficients are in the range $[-100, 100]$ and the off-diagonal entries are in the range $[-50, 50]$.

Looking at all the computational results, we make the following observations:

a) Comparing the globalized version MSN, MSNa and MSNb it seems most efficient to exploit SN updates as much as possible, showing that MSNb clearly yields the overall best performance.

b) It is not very elegant to have a counter to exit sequences of type case-1a and case-1b consecutive iterations, but found no mathematical proof for these sequences to be bounded in general.

c) There is no clear dominance relation between RSN, the globalization of SN using recursion, and the new version MSNb. On instances, where SN cycles, both these modifications also take many more iterations than usual, see Tables 8 and 9.

d) It is remarkable that the globalized versions substantially improve the performance of the SN method on the class of test cases from Example 16.

e) The prize of ensuring convergence of the globalized methods is relatively small on all instances, where the SN method terminates anyway.

f) There does not seem to be a substantial difference in performance of the globalized methods in case of box constraints versus one-sided bounds.

n	cond(Q)	dens(Q)	CPLEX-IP		[RSN]		[MSNb]	
			iter	time	iter	time	iter	time
1000	$\approx 10^8$	.1	8.0	3.16	6.0	0.41	5.8	0.33
2000	$\approx 2 \cdot 10^8$	.1	8.0	38.36	6.0	2.19	5.8	1.89
5000	$\approx 5 \cdot 10^8$	.1	8.0	533.44	7.0	31.95	6.7	28.77

Table 13: Convex quadratic minimization over the unit cube, Comparison with interior point methods.

7 Conclusion

The minimization of a convex quadratic function under bound constraints is a fundamental building block in optimization. We start from the active-set method introduced by Bergounioux et al. [1, 2] and introduce yet another modification to make it globally convergent and at the same time maintain the combinatorial flavour of the original active-set method. We also show that it has a competitive performance on a variety of difficult classes of test problems. The corresponding Matlab code for generating all instances and solving them with the various methods discussed in this paper is available from `http://philipphungerlaender.jimdo.com/qp-code/`.

For future research it would be challenging to try to combine the globalized active-set method suggested in this paper with the globalized version using recursion from Hungerländer and Rendl [18] in order to achieve further performance improvements. Trying to extend the method suggested in this paper to convex QPs with general linear constraints seems to be another worthwhile direction for research.

References

[1] M. Bergounioux, K. Ito, and K. Kunisch. Primal-Dual Strategy for Constrained Optimal Control Problems. *SIAM Journal on Control and Optimization*, 37:1176–1194, 1999.

[2] M. Bergounioux, M. Haddou, M. Hintermüller, and K. Kunisch. A comparison of interior point methods and a Moreau-Yosida based active set strategy for constrained optimal control problems. *SIAM Journal on Optimization*, 11(2):495–521, 2000.

[3] A. Billionnet and S. Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109(1):55–68, 2007.

[4] R. H. Byrd, G. M. Chin, J. Nocedal, and F. Oztoprak. A family of second-order methods for convex ℓ_1 regularized optimization. *Mathematical Programming*, 2015. Online version.

[5] E. J. Candès, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[6] F. E. Curtis, Z. Han, and D. P. Robinson. A globally convergent primal-dual active-set framework for large-scale convex quadratic optimization. *Computational Optimization and Applications*, 60(2):311–341, 2015.

[7] M. D'Apuzzo and M. Marino. Parallel computational issues of an interior point method for solving large bound-constrained quadratic programming problems. *Parallel Computing*, 29:467–483, 2003.

[8] D. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006. ISSN 0018-9448.

[9] Z. Dostál and J. Schöberl. Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination. *Computational Optimization and Applications*, 30:23–43, 2005.

[10] M. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of selected topics in Signal Processing*, 1(4): 586–597, 2007.

[11] I. B. Gharbia and C. J. Gilbert. Nonconvergence of the plain newton-min algorithm for linear complementarity problems with a P-matrix. *Mathematical Programming*, 134(2):349–364, 2012.

[12] N. Goldberg and S. Leyffer. An active-set method for second-order conic-constrained quadratic programming. *SIAM Journal on Optimization*, 25(3):1455–1477, 2015.

[13] N. I. M. Gould, D. Orban, and P. L. Toint. CUTEr and sifdec: A constrained and unconstrained testing environment, revisited. *ACM Transactions on Mathematical Software*, 29(4):373–394, 2003.

[14] N. I. M. Gould, D. Orban, and P. L. Toint. CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Computational Optimization and Applications*, 60(3):545–557, 2015.

[15] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semi-smooth newton method. *SIAM Journal on Optimization*, 13(3):865–888, 2003.

[16] P. Hoyer. Non-negative sparse coding. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 557–565, 2002.

[17] P. Hungerländer. *Algorithms for convex quadratic programming*. Master Thesis, Alpen-Adria-Universität Klagenfurt, Austria, 2009.

[18] P. Hungerländer and F. Rendl. A feasible active set method for strictly convex problems with simple bounds. *SIAM Journal on Optimization*, 25(3):1633–1659, 2015.

[19] IBM. *IBM ILOG CPLEX V12.1 User's Manual for CPLEX*. URL `ftp://public.dhe.ibm.com/software/websphere/ilog/docs/optimization/cplex/ps_usrmancplex.pdf`, 2009.

[20] J. J. Júdice and F. M. Pires. A block principal pivoting algorithm for large-scale strictly monotone linear complementarity problems. *Computers and Operations Research*, 21(5):587–596, 1994.

[21] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):606–617, 2007.

[22] K. Kunisch and F. Rendl. An infeasible active set method for convex problems with simple bounds. *SIAM Journal on Optimization*, 14(1):35–52, 2003.

[23] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401: 788–791, 1999.

[24] J. Moré and G. Toraldo. On the solution of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization*, 1:93–113, 1991.

[25] K. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann Verlag, 1988.

[26] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, New York, second edition, 2006.

[27] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[28] P. M. Pardalos and G. P. Rodgers. Computational aspects of a branch and bound algorithm for quadratic zero-one programming. *Computing*, 45(2):131–144, 1990.

A APPENDIX

A.1 Convex QPs with box constraints

In this appendix we state the details for the generalization of the convergence argument from Sections 3 and 4 to convex QPs with box constraints $a \leq x \leq b$. Note that the convergence argument for the original SN-method that avoids cycling only under some strong form of diagonal dominance of Q cannot be generalized to the box-constraint case. For a detailed discussion of this issue see [17].

We introduce an additional active set $\mathcal{E}$ with $x_{\mathcal{E}} = a_{\mathcal{E}}$ and dual variables γ. Furthermore we set $\mathcal{I} = \mathcal{N} \backslash (\mathcal{A} \cup \mathcal{E})$. The KKT system for box-constrained convex QPs is given by

$$\begin{aligned} Qx + q + \alpha + \gamma &= 0, \\ \alpha \circ (b - x) &= 0, \\ \gamma \circ (a - x) &= 0, \\ b - x &\geq 0, \\ x - a &\geq 0, \\ \alpha &\geq 0, \\ \gamma &\leq 0. \end{aligned}$$

To simplify notation let us denote $KKT(\mathcal{A}, \mathcal{E})$ as the following set of equations:

$$KKT(\mathcal{A}, \mathcal{E}) \qquad Qx + q + \alpha + \gamma = 0, \quad x_{\mathcal{A}} = b_{\mathcal{A}}, \quad x_{\mathcal{E}} = a_{\mathcal{E}}, \quad \alpha_{\mathcal{I}} = \alpha_{\mathcal{E}} = \gamma_{\mathcal{I}} = \gamma_{\mathcal{A}} = 0.$$

Let us also extend the SN Update rule from (6) for the box-constraint case:

$$\mathcal{B} := \{i \in \mathcal{I} : x_i \geq b_i\} \cup \{i \in A : \alpha_i \geq 0\}, \quad \mathcal{F} := \{i \in \mathcal{I} : x_i \leq a_i\} \cup \{i \in E : \gamma_i \leq 0\}. \tag{20}$$

A.2 Some basic properties for consecutive SN-iterates in the box-constraint case

We consider two consecutive iterations of the SN-method. Thus starting with $\mathcal{A}, \mathcal{E} \subseteq N$ we get $[x, \alpha, \gamma] = KKT(\mathcal{A})$, set $u = \max(\min(x, b), a)$ and determine $\mathcal{B}$ and $\mathcal{F}$ given by (20). Then we compute $[y, \beta, \delta] = KKT(\mathcal{B}, \mathcal{F})$ and set $v = \max(\min(y, b), a)$. Thus one iteration of the SN-method moves from the configuration $(\mathcal{A}, (x, \alpha, \gamma), u)$ to the new configuration $(\mathcal{B}, (y, \beta, \delta), v)$.

We declare the new configuration **successful** if

$$J(v) < J(u).$$

Next we extend Lemmas 4 – 5 from Section 3 to the box-constraint case.

Lemma 16. *Let A, x, B be as above and set $u := \max(\min(x, b), a)$. Then $u_i = b_i \; \forall i \in \mathcal{A} \cup \mathcal{B}$ and $u_i = a_i \; \forall i \in \mathcal{E} \cup \mathcal{F}$.*

Proof. Since $x_{\mathcal{A}} = b_{\mathcal{A}},\ x_{\mathcal{E}} = a_{\mathcal{E}}$ we have $u_{\mathcal{A}} = b_{\mathcal{A}},\ u_{\mathcal{E}} = a_{\mathcal{E}}$. Let $\mathcal{B}_1 = \{i \in \mathcal{A} : \alpha_i \geq 0\}$ and $\mathcal{B}_2 = \{i \in \mathcal{I} : x_i \geq b_i\}$, hence $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$. Accordingly let $\mathcal{F}_1 = \{i \in \mathcal{E} : \gamma_i \leq 0\}$ and $\mathcal{F}_2 = \{i \in I : x_i \leq a_i\}$, hence $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$. Since $\mathcal{B}_1, \mathcal{F}_1 \subseteq \mathcal{A}$ we have $u_{\mathcal{B}_1} = b_{\mathcal{B}_1},\ u_{\mathcal{F}_1} = a_{\mathcal{F}_1}$. On the other hand $u_{\mathcal{B}_2} = b_{\mathcal{B}_2},\ u_{\mathcal{F}_2} = b_{\mathcal{F}_2}$ due to the definition of u. □

In the following, $\mathcal{B},\ \mathcal{F} \subseteq \mathcal{N}$ are arbitrary sets and $y = KKT(\mathcal{B}, \mathcal{F})$. We also use an arbitrary feasible point u ($u \leq b$) with $u_{\mathcal{B}} = b_{\mathcal{B}},\ u_{\mathcal{F}} = a_{\mathcal{F}}$. Again we work with the line segment joining u and y

$$z(\lambda) := \lambda u + (1 - \lambda) y \text{ for } 0 \leq \lambda < 1, \tag{21}$$

where $J(z(\lambda))$ is strictly monotonically decreasing from u to y in case $u \neq y$.

Lemma 17. *Let $\mathcal{B},\ \mathcal{F} \subseteq \mathcal{N}$ and $y = KKT(\mathcal{B},\mathcal{F})$. Further let $a \le u \le b$ and $u_{\mathcal{B}} = b_{\mathcal{B}},\ u_{\mathcal{F}} = a_{\mathcal{F}}$. If $u \neq y$ then $J(u) > J(z(\lambda_1)) > J(z(\lambda_2)) \ge J(y)$ for all $0 \le \lambda_2 < \lambda_1 < 1$.*

Proof. Note that y is optimal for the problem

$$\min\{J(y) : y_{\mathcal{B}} = b_{\mathcal{B}},\ y_{\mathcal{F}} = a_{\mathcal{F}}\}$$

and u is feasible for this problem and the cost function J is strictly convex. □

We next show that $z(\lambda)$ is in fact the optimal solution for a slightly perturbed objective function with respect to the constraints $x_{\mathcal{B}} = b_{\mathcal{B}}$ and $x_{\mathcal{F}} = a_{\mathcal{F}}$.

Lemma 18. *Let $\mathcal{B},\ \mathcal{F} \subseteq \mathcal{N}$ and $y = KKT(\mathcal{B},\mathcal{F})$. Further let $a \le u \le b$ and $u_{\mathcal{B}} = b_{\mathcal{B}},\ u_{\mathcal{F}} = a_{\mathcal{F}}$. Then $z(\lambda)$ given by* (21) *is a solution of*

$$\min_z \left\{ J(z) + \frac{\lambda}{2(1-\lambda)}(u-z)^T Q(u-z) : z_{\mathcal{B}} = b_{\mathcal{B}},\ z_{\mathcal{F}} = a_{\mathcal{F}} \right\} \tag{22}$$

for all $0 \le \lambda < 1$.

Proof. Let $\mathcal{J} := \mathcal{N} \setminus (\mathcal{B} \cup \mathcal{F})$. Then the objective function in (22) becomes

$$\text{constant} + \Big(-\frac{\lambda}{1-\lambda} Q_{\mathcal{J}} u_{\mathcal{J}} + q_{\mathcal{J}} + Q_{\mathcal{J},\mathcal{B}} b_{\mathcal{B}} + Q_{\mathcal{J},\mathcal{F}} a_{\mathcal{F}}\Big)^T z_{\mathcal{J}} + \frac{1}{2(1-\lambda)} z_{\mathcal{J}}^T Q_{\mathcal{J}} z_{\mathcal{J}}.$$

Hence the first order optimality condition of (22) is

$$\frac{1}{1-\lambda} Q_{\mathcal{J}} z_{\mathcal{J}} - \frac{\lambda}{1-\lambda} Q_{\mathcal{J}} u_{\mathcal{J}} + q_{\mathcal{J}} + Q_{\mathcal{J},\mathcal{B}} b_{\mathcal{B}} + Q_{\mathcal{J},\mathcal{F}} a_{\mathcal{F}} = 0.$$

Solving for $z_{\mathcal{J}}$ gives

$$z_{\mathcal{J}} = \lambda u_{\mathcal{J}} - (1-\lambda)(Q_{\mathcal{J}}^{-1}(q_{\mathcal{J}} + Q_{\mathcal{J},\mathcal{B}} b_{\mathcal{B}} + Q_{\mathcal{J},\mathcal{F}} a_{\mathcal{F}})) = \lambda u_{\mathcal{J}} + (1-\lambda) y_{\mathcal{J}}.$$

The last equation follows from $y_{\mathcal{J}} = -Q_{\mathcal{J}}^{-1}(q_{\mathcal{J}} + Q_{\mathcal{J},\mathcal{B}} b_{\mathcal{B}} + Q_{\mathcal{J},\mathcal{F}} a_{\mathcal{F}})$. Moreover $u_{\mathcal{B}} = y_{\mathcal{B}} = b_{\mathcal{B}},\ u_{\mathcal{F}} = y_{\mathcal{F}} = a_{\mathcal{F}}$ and hence $z_{\mathcal{B}} = b_{\mathcal{B}},\ z_{\mathcal{F}} = a_{\mathcal{F}}$. This shows that $z(\lambda)$ is a solution of (22). □

Next we propose modifications to the definition of the actives set in case that the SN Update is not successful. These are based on the properties of the line segment in (21).

A.3 SN-method with safeguards using combinatorial line search for box constraints

Let us explain our modifications to the SN-method to ensure finite termination for convex QPs with box-constraints. An iteration is started with a configuration $(\mathcal{A}, \mathcal{E}, (x, \alpha, \gamma), u)$. We will maintain the following properties of a configuration during all iterations.

$$\begin{aligned} &Qx + q + \alpha + \gamma = 0,\ x_{\mathcal{A}} = b_{\mathcal{A}},,\ x_{\mathcal{E}} = a_{\mathcal{E}},\ u = \max(\min(x, b), a). \\ &\text{If } a_i < x_i < b_i\ \forall i \in \mathcal{I} \text{ then } [x, \alpha, \gamma] = KKT(\mathcal{A}, \mathcal{E}). \end{aligned} \tag{23}$$

An iteration is terminated with a new configuration $(\mathcal{A}^+, \mathcal{E}^+, (x^+, \alpha^+, \gamma^+), u^+)$. We label each configuration either 'green' in the case that $[x, \alpha, \gamma] = KKT(\mathcal{A})$, or 'red' otherwise.

We first test whether the SN Update is successful. If it is not, we carry out a safeguard update. Here are the details.

A.4 SN Update

We start the iteration with the configuration $(\mathcal{A}, \mathcal{E}, (x, \alpha), u)$ which may be either green or red, and determine $\mathcal{B}$ and $\mathcal{F}$ by (20). We calculate $[y, \beta, \delta] = KKT(\mathcal{B}, \mathcal{F})$ and set $v = \max(\min(y, b), a)$. If $J(v) < J(u)$ we continue with

$$A^+ \leftarrow \mathcal{B},\ \mathcal{E}^+ \leftarrow \mathcal{F},\ (x^+, \alpha^+, \gamma^+) \leftarrow (y, \beta, \delta),\ u^+ \leftarrow v.$$

In this case the conditions (23) obviously hold for this update. Moreover, the new configuration $(\mathcal{A}^+, \mathcal{E}^+, (x^+, \alpha^+), u^+)$ is green.

A.5 Safeguard Update

If $J(v) \geq J(u)$ we carry out a Safeguard Update. In this case we modify the SN Update in two ways. Instead of $\mathcal{B}$ and $\mathcal{F}$ given by (20) we use slightly modified new actives set $\mathcal{A}^+$ and $\mathcal{E}^+$. Moreover we do not insist that $(x^+, \alpha^+, \gamma^+)$ is the solution of $KKT(\mathcal{A}^+, \mathcal{E}^+)$ but only maintain (23). The new active sets $\mathcal{A}^+$ and $\mathcal{E}^+$ will be determined differently depending on whether the current iterate x satisfies $a < x_i < b_i\ \forall i \in \mathcal{I}$ (dual update) or not (primal update).

A.5.1 Definition of $\mathcal{A}^+$ and $\mathcal{E}^+$

Dual Update. If the SN Update was unsuccessful and $a_i < x_i < b_i$ for all $i \in \mathcal{I}$, then we have a primal feasible x, and therefore (x, α, γ) is by assumption (23) the solution of $KKT(\mathcal{A}, \mathcal{E})$. Hence we start with a green configuration, determined by the sets $\mathcal{A}$ and $\mathcal{E}$. As the solution is not optimal we must have

$$\min\{\alpha_i : i \in \mathcal{A}\} < 0 \text{ or } \max\{\gamma_i : i \in \mathcal{E}\} > 0.$$

Let $j := \arg\min \{\{\alpha_i : i \in \mathcal{A}\}, -\{\gamma_i : i \in \mathcal{E}\}\}$. We define

$$\mathcal{A}^+ := \mathcal{A} \setminus j,\ \mathcal{E}^+ := \mathcal{E},\ \text{if } j \in \mathcal{A}, \qquad \mathcal{A}^+ := \mathcal{A},\ \mathcal{E}^+ := \mathcal{E} \setminus j,\ \text{if } j \in \mathcal{E}. \tag{24}$$

Remark 5. *Note that the SN Update would release all $i \in \mathcal{A}$ with $\alpha_i < 0$ and all $i \in \mathcal{E}$ with $\gamma_i > 0$. Here we release only one of those variables. Hence $\mathcal{A}^+ \cup \mathcal{E}^+$ is at least as close to $\mathcal{A} \cup \mathcal{E}$ as to $\mathcal{B} \cup \mathcal{F}$ in the sense that*

$$|\left(\mathcal{A}^+ \cup \mathcal{E}^+\right) \cap (\mathcal{A} \cup \mathcal{E})| \geq |\left(\mathcal{A}^+ \cup \mathcal{E}^+\right) \cap (\mathcal{B} \cup \mathcal{F})|.$$

Primal Update. In the final case we have $x_i \geq b_i$ or $x_i \leq a_i$ for at least one index $i \in \mathcal{I}$. In this case we set

$$\mathcal{A}^+ := \mathcal{A} \cup \{i \in \mathcal{I} : x_i \geq b_i\}, \quad \mathcal{E}^+ := \mathcal{E} \cup \{i \in \mathcal{I} : x_i \leq a_i\} \tag{25}$$

Remark 6. *Note that the SN Update would be obtained from $\mathcal{A}^+$ and $\mathcal{E}^+$ by additionally removing all $i \in \mathcal{A}$ with $\alpha_i < 0$ and all $i \in \mathcal{E}$ with $\gamma_i > 0$. Again $\mathcal{A}^+ \cup \mathcal{E}^+$ is at least as close to $\mathcal{A} \cup \mathcal{E}$ as to $\mathcal{B} \cup \mathcal{F}$ in the sense that*

$$|\left(\mathcal{A}^+ \cup \mathcal{E}^+\right) \cap (\mathcal{A} \cup \mathcal{E})| \geq |\left(\mathcal{A}^+ \cup \mathcal{E}^+\right) \cap (\mathcal{B} \cup \mathcal{F})|.$$

Having defined the new active sets $\mathcal{A}^+$ and $\mathcal{E}^+$ in both the Primal and the Dual Update, we now explain how we define the new primal-dual triple $(x^+, \alpha^+, \gamma^+)$. We first observe that the current feasible solution u is on the upper bound on $\mathcal{A}$ and $\mathcal{A}^+$ and on the lower bound on $\mathcal{E}$ and $\mathcal{E}^+$.

Lemma 19. *Let $\mathcal{A}, \mathcal{E}, x, \mathcal{A}^+, \mathcal{E}^+$ be as above and set $u := \max(\min(x, b), a)$. Then $u_i = b_i\ \forall i \in \mathcal{A} \cup \mathcal{A}^+$ and $u_i = a_i\ \forall i \in \mathcal{E} \cup \mathcal{E}^+$.*

Proof. We always have $x_{\mathcal{A}} = b_{\mathcal{A}},\ x_{\mathcal{E}} = a_{\mathcal{E}}$, hence $u_{\mathcal{A}} = b_{\mathcal{A}},\ u_{\mathcal{E}} = a_{\mathcal{E}}$. In case 2 we have $\mathcal{A}^+ \subset \mathcal{A},\ \mathcal{E}^+ \subset \mathcal{E}$, so the lemma holds. In the other cases, $\mathcal{A}^+$ contains all $i \in \mathcal{A}$ such that $x_i \geq b_i$ and contains all $i \in \mathcal{E}$ such that $x_i \leq a_i$. Hence the projection u will be on the respective bound for those i as well. □

We use the new active sets $\mathcal{A}^+$ and $\mathcal{E}^+$ and determine

$$[y, \beta, \delta] = KKT(\mathcal{A}^+, \mathcal{E}^+).$$

The previous lemma shows that u is on the upper bound on $\mathcal{A}^+$ and on the lower bound on $\mathcal{E}^+$. Hence we may use u and y to define the line segment $z(\lambda)$ given in (21). The new primal-dual triple will be determined according to a combinatorial line search on the segment $z(\lambda)$ given by (21).

A.5.2 The new primal-dual triple $(x^+, \alpha^+, \gamma^+)$ for the Dual Update

We first investigate the situation in case of a Dual Update, where $a_i < x_i < b_i \; \forall i \in \mathcal{I}$, so $u = x$. We recall that in this case the SN Update $(\mathcal{B}, \mathcal{F})$ with $\mathcal{B} = \{i \in \mathcal{A} : \alpha_i \geq 0\} \subset \mathcal{A}$ and $\mathcal{F} = \{i \in \mathcal{E} : \gamma_i \leq 0\} \subset \mathcal{E}$ was not successful. Hence we use (24) to define $(\mathcal{A}^+, \mathcal{E}^+)$. The next lemma shows that the KKT solution $[y, \beta, \delta] = KKT(\mathcal{A}^+, \mathcal{E}^+)$ has $a < y_j < b_j$ and in particular $y \neq u$.

Lemma 20. *Let $(x, \alpha, \gamma) = KKT(\mathcal{A}, \mathcal{E})$ with $a_i < x_i < b_i$ for $i \in \mathcal{I}$ and suppose x is not optimal. Let the new active sets $\mathcal{B} = \mathcal{A} \setminus \mathcal{C}$ and $\mathcal{F} = \mathcal{E} \setminus \mathcal{D}$ be given with $\mathcal{C} \subseteq \{i \in A : \alpha_i < 0\}$, $\mathcal{D} \subseteq \{i \in E : \gamma_i > 0\}$ and $\mathcal{C} \cup \mathcal{D}$ nonempty. Then there exists $j \in (\mathcal{C} \cup \mathcal{D})$ such that either $y_j < b_j$ or $a_j < y_j$ holds for $(y, \beta, \delta) = KKT(\mathcal{B}, \mathcal{F})$.*

Proof. Due to the assumptions of the lemma and the definitions of $\mathcal{B}$ and $\mathcal{F}$ the following implications hold

$$[x, \alpha, \gamma] = KKT(\mathcal{A}, \mathcal{E}) \Rightarrow \alpha_{\mathcal{C}} < 0,\ \gamma_{\mathcal{D}} > 0, \qquad [y, \beta, \delta] = KKT(\mathcal{B}, \mathcal{F}) \Rightarrow \beta_{\mathcal{C}} = \delta_{\mathcal{D}} = 0. \tag{26}$$

As $\mathcal{I} \subsetneq \mathcal{J}$ we can substitute the primal variables associated with $\mathcal{I}$ by

$$\begin{aligned} x_{\mathcal{I}} &= -Q_{\mathcal{I}}^{-1}(q_{\mathcal{I}} + Q_{\mathcal{I},\mathcal{B}} b_{\mathcal{B}} + Q_{\mathcal{I},\mathcal{C}} b_{\mathcal{C}} + Q_{\mathcal{I},\mathcal{F}} a_{\mathcal{F}} + Q_{\mathcal{I},\mathcal{D}} a_{\mathcal{D}}), \\ y_{\mathcal{I}} &= -Q_{\mathcal{I}}^{-1}(q_{\mathcal{I}} + Q_{\mathcal{I},\mathcal{B}} b_{\mathcal{B}} + Q_{\mathcal{I},\mathcal{C}} y_{\mathcal{C}} + Q_{\mathcal{I},\mathcal{F}} a_{\mathcal{F}} + Q_{\mathcal{I},\mathcal{D}} y_{\mathcal{D}}). \end{aligned} \tag{27}$$

The equations $Qx + q + \alpha + \gamma = 0$ and $Qy + q + \beta + \delta = 0$ hold due to the definition of $KKT(\cdot)$. Using (26) we obtain

$$\begin{aligned} &Q_{\mathcal{C},\mathcal{N}} \begin{pmatrix} b_{\mathcal{B}} \\ b_{\mathcal{C}} \\ a_{\mathcal{F}} \\ a_{\mathcal{D}} \\ x_{\mathcal{I}} \end{pmatrix} + q_{\mathcal{C}} = -\alpha_{\mathcal{C}} > 0, \qquad Q_{\mathcal{D},\mathcal{N}} \begin{pmatrix} b_{\mathcal{B}} \\ b_{\mathcal{C}} \\ a_{\mathcal{F}} \\ a_{\mathcal{D}} \\ x_{\mathcal{I}} \end{pmatrix} + q_{\mathcal{D}} = -\delta_{\mathcal{D}} < 0, \\ &Q_{\mathcal{C},\mathcal{N}} \begin{pmatrix} b_{\mathcal{B}} \\ y_{\mathcal{C}} \\ a_{\mathcal{F}} \\ y_{\mathcal{D}} \\ y_{\mathcal{I}} \end{pmatrix} + q_{\mathcal{C}} = -\beta_{\mathcal{C}} = 0, \qquad Q_{\mathcal{D},\mathcal{N}} \begin{pmatrix} b_{\mathcal{B}} \\ y_{\mathcal{C}} \\ a_{\mathcal{F}} \\ y_{\mathcal{D}} \\ y_{\mathcal{I}} \end{pmatrix} + q_{\mathcal{D}} = -\delta_{\mathcal{D}} = 0. \end{aligned} \tag{28}$$

Now applying (27) to (28) yields

$$\begin{aligned} &Q_{\mathcal{C},\mathcal{B}} b_{\mathcal{B}} + Q_{\mathcal{C}} b_{\mathcal{C}} + Q_{\mathcal{C},\mathcal{F}} a_{\mathcal{F}} + Q_{\mathcal{C},\mathcal{D}} a_{\mathcal{D}} - Q_{\mathcal{C},\mathcal{I}} Q_{\mathcal{I}}^{-1}(q_{\mathcal{I}} + Q_{\mathcal{I},\mathcal{B}} b_{\mathcal{B}} + \\ &Q_{\mathcal{I},\mathcal{C}} b_{\mathcal{C}} + Q_{\mathcal{I},\mathcal{F}} a_{\mathcal{F}} + Q_{\mathcal{I},\mathcal{D}} a_{\mathcal{D}}) + q_{\mathcal{C}} > Q_{\mathcal{C},\mathcal{B}} b_{\mathcal{B}} + Q_{\mathcal{C}} y_{\mathcal{C}} + Q_{\mathcal{C},\mathcal{F}} a_{\mathcal{F}} + \\ &Q_{\mathcal{C},\mathcal{D}} y_{\mathcal{D}} - Q_{\mathcal{C},\mathcal{I}} Q_{\mathcal{I}}^{-1}(q_{\mathcal{I}} + Q_{\mathcal{I},\mathcal{B}} b_{\mathcal{B}} + Q_{\mathcal{I},\mathcal{C}} y_{\mathcal{C}} + Q_{\mathcal{I},\mathcal{F}} a_{\mathcal{F}} + Q_{\mathcal{I},\mathcal{D}} y_{\mathcal{D}}) + q_{\mathcal{C}}, \end{aligned}$$

and

$$\begin{aligned} &Q_{\mathcal{D},\mathcal{B}} b_{\mathcal{B}} + Q_{\mathcal{D},\mathcal{C}} b_{\mathcal{C}} + Q_{\mathcal{D},\mathcal{D}_s} a_{\mathcal{D}_s} + Q_{\mathcal{D}} a_{\mathcal{D}} - Q_{\mathcal{D},\mathcal{I}} Q_{\mathcal{I}}^{-1}(q_{\mathcal{I}} + Q_{\mathcal{I},\mathcal{B}} b_{\mathcal{B}} + \\ &Q_{\mathcal{I},\mathcal{D}_s} a_{\mathcal{D}_s} + Q_{\mathcal{I},\mathcal{C}} b_{\mathcal{C}} + Q_{\mathcal{I},\mathcal{D}} a_{\mathcal{D}}) + q_{\mathcal{D}} < Q_{\mathcal{D},\mathcal{B}} b_{\mathcal{B}} + Q_{\mathcal{D},\mathcal{C}} y_{\mathcal{C}} + Q_{\mathcal{D},\mathcal{D}_s} a_{\mathcal{D}_s} + \\ &Q_{\mathcal{D}} y_{\mathcal{D}} - Q_{\mathcal{D},\mathcal{I}} Q_{\mathcal{I}}^{-1}(q_{\mathcal{I}} + Q_{\mathcal{I},\mathcal{B}} b_{\mathcal{B}} + Q_{\mathcal{I},\mathcal{D}_s} a_{\mathcal{D}_s} + Q_{\mathcal{I},\mathcal{C}} y_{\mathcal{C}} + Q_{\mathcal{I},\mathcal{D}} y_{\mathcal{D}}) + q_{\mathcal{D}}. \end{aligned}$$

Simplifying the above inequalities gives

$$\begin{aligned}&(Q_{\mathcal{C}} - Q_{\mathcal{C},\mathcal{I}}Q_{\mathcal{I}}^{-1}Q_{\mathcal{I},\mathcal{C}})b_{\mathcal{C}} + (Q_{\mathcal{C},\mathcal{D}} - Q_{\mathcal{C},\mathcal{I}}Q_{\mathcal{I}}^{-1}Q_{\mathcal{I},\mathcal{D}})a_{\mathcal{D}} > \\ &(Q_{\mathcal{C}} - Q_{\mathcal{C},\mathcal{I}}Q_{\mathcal{I}}^{-1}Q_{\mathcal{I},\mathcal{C}})y_{\mathcal{C}} + (Q_{\mathcal{C},\mathcal{D}} - Q_{\mathcal{C},\mathcal{I}}Q_{\mathcal{I}}^{-1}Q_{\mathcal{I},\mathcal{D}})y_{\mathcal{D}}.\end{aligned}$$

and

$$\begin{aligned}&(Q_{\mathcal{D}} - Q_{\mathcal{D},\mathcal{I}}Q_{\mathcal{I}}^{-1}Q_{\mathcal{I},\mathcal{D}})a_{\mathcal{D}} + (Q_{\mathcal{D},\mathcal{C}} - Q_{\mathcal{D},\mathcal{I}}Q_{\mathcal{I}}^{-1}Q_{\mathcal{I},\mathcal{C}})b_{\mathcal{C}} < \\ &(Q_{\mathcal{D}} - Q_{\mathcal{D},\mathcal{I}}Q_{\mathcal{I}}^{-1}Q_{\mathcal{I},\mathcal{D}})y_{\mathcal{D}} + (Q_{\mathcal{D},\mathcal{C}} - Q_{\mathcal{D},\mathcal{I}}Q_{\mathcal{I}}^{-1}Q_{\mathcal{I},\mathcal{C}})y_{\mathcal{C}}.\end{aligned}$$

Now adding up the two inequalities yields

$$\left[\begin{pmatrix} Q_{\mathcal{C}} & -Q_{\mathcal{C},\mathcal{D}} \\ -Q_{\mathcal{D},\mathcal{C}} & Q_{\mathcal{D}} \end{pmatrix} - \begin{pmatrix} Q_{\mathcal{C},\mathcal{I}} \\ -Q_{\mathcal{D},\mathcal{I}} \end{pmatrix} Q_{\mathcal{I}}^{-1} \begin{pmatrix} Q_{\mathcal{I},\mathcal{C}} & -Q_{\mathcal{I},\mathcal{D}} \end{pmatrix}\right] \begin{pmatrix} b_{\mathcal{C}} - y_{\mathcal{C}} \\ y_{\mathcal{D}} - a_{\mathcal{D}} \end{pmatrix} > 0. \tag{29}$$

The matrix

$$\begin{pmatrix} Q_{\mathcal{C}} & -Q_{\mathcal{C},\mathcal{D}} \\ -Q_{\mathcal{D},\mathcal{C}} & Q_{\mathcal{D}} \end{pmatrix} - \begin{pmatrix} Q_{\mathcal{C},\mathcal{I}} \\ -Q_{\mathcal{D},\mathcal{I}} \end{pmatrix} Q_{\mathcal{I}}^{-1} \begin{pmatrix} Q_{\mathcal{I},\mathcal{C}} & -Q_{\mathcal{I},\mathcal{D}} \end{pmatrix},$$

is positive definite. To see this we use

$$\begin{pmatrix} Q_{\mathcal{C}} & -Q_{\mathcal{C},\mathcal{D}} \\ -Q_{\mathcal{D},\mathcal{C}} & Q_{\mathcal{D}} \end{pmatrix} \succ 0, \qquad \text{and} \qquad M_1 := \begin{pmatrix} Q_{\mathcal{C}} & -Q_{\mathcal{C},\mathcal{D}} & Q_{\mathcal{C},\mathcal{I}} \\ -Q_{\mathcal{D},\mathcal{C}} & Q_{\mathcal{D}} & -Q_{\mathcal{D},\mathcal{I}} \\ Q_{\mathcal{I},\mathcal{C}} & -Q_{\mathcal{I},\mathcal{D}} & Q_{\mathcal{I}} \end{pmatrix} \succ 0,$$

in the Schur-complement lemma. The two above matrices are positive definite because

$$M_2 := \begin{pmatrix} Q_{\mathcal{C}} & Q_{\mathcal{C},\mathcal{D}} & Q_{\mathcal{C},\mathcal{I}} \\ Q_{\mathcal{D},\mathcal{C}} & Q_{\mathcal{D}} & Q_{\mathcal{D},\mathcal{I}} \\ Q_{\mathcal{I},\mathcal{C}} & Q_{\mathcal{I},\mathcal{D}} & Q_{\mathcal{I}} \end{pmatrix} \succ 0, \qquad \text{and} \qquad \begin{pmatrix} x_{\mathcal{C}} \\ x_{\mathcal{D}} \\ x_{\mathcal{I}} \end{pmatrix}^{\top} M_2 \begin{pmatrix} x_{\mathcal{C}} \\ x_{\mathcal{D}} \\ x_{\mathcal{I}} \end{pmatrix} = \begin{pmatrix} x_{\mathcal{C}} \\ -x_{\mathcal{D}} \\ x_{\mathcal{I}} \end{pmatrix}^{\top} M_1 \begin{pmatrix} x_{\mathcal{C}} \\ -x_{\mathcal{D}} \\ x_{\mathcal{I}} \end{pmatrix}.$$

Hence we have

$$\begin{pmatrix} b_{\mathcal{C}} - y_{\mathcal{C}} \\ y_{\mathcal{D}} - a_{\mathcal{D}} \end{pmatrix}^{\top} \left[\begin{pmatrix} Q_{\mathcal{C}} & -Q_{\mathcal{C},\mathcal{D}} \\ -Q_{\mathcal{D},\mathcal{C}} & Q_{\mathcal{D}} \end{pmatrix} - \begin{pmatrix} Q_{\mathcal{C},\mathcal{I}} \\ -Q_{\mathcal{D},\mathcal{I}} \end{pmatrix} Q_{\mathcal{I}}^{-1} \begin{pmatrix} Q_{\mathcal{I},\mathcal{C}} & -Q_{\mathcal{I},\mathcal{D}} \end{pmatrix}\right] \begin{pmatrix} b_{\mathcal{C}} - y_{\mathcal{C}} \\ y_{\mathcal{D}} - a_{\mathcal{D}} \end{pmatrix} > 0. \tag{30}$$

Finally due to (29) and (30) we can conclude that there exists $j \in (\mathcal{C} \cup \mathcal{D})$ such that either $y_j < b_j$ or $a_j < y_j$ holds. □

In case that the solution y of $KKT(\mathcal{A}^+, \mathcal{E}^+)$ is primal feasible, we set

$$(x^+, \alpha^+, \gamma^+) := (y, \beta, \delta), \; u^+ := y.$$

Note that (23) holds for this update, and moreover $J(u^+) < J(u)$, as $x = u \neq y$. Thus the new configuration will be green.

If y is not feasible, then the set $\mathcal{G} = (\{i : y_i > b_i\} \cup \{i : y_i < a_i\})$ is non-empty. We define

$$\lambda_i := \frac{y_i - b_i}{y_i - u_i} > 0, \text{ for } y_i > b_i, \qquad \lambda_i := \frac{a_i - y_i}{u_i - y_i} > 0, \text{ for } y_i < a_i.$$

The previous lemma shows that $\mathcal{G} \subseteq \mathcal{I}$. Moreover, we have $a_i < x_i < b_i \; \forall i \in \mathcal{I}$, hence $a_i < u_i = x_i < b_i$ and therefore $0 < \lambda_i < 1$. At each of the values λ_i we have either $z(\lambda_i) = b_i$ or $z(\lambda_i) = a_i$. In other words, at $z(\lambda_i)$, the line segment passes through either the upper bound b_i or the lower bound a_i. In each of these points we consider the projection $w_i := \max(\min(b, z(\lambda_i)), a)$ and define

$$\lambda^* = \arg\min_{i \in \mathcal{G}} J(w_i). \tag{31}$$

Determining λ^* in this way can be viewed as a combinatorial line search of the projection of $z(\lambda)$ to the feasible set with respect to the objective function J.

Since $\lambda^* < 1$ we have with $w^* = \max(\min(b, z(\lambda^*)), a)$ that $J(w^*) < J(u)$. This is in fact the reason for defining the Dual Update condition in this specific way. We finish now the Dual Update step also for the case that $y \not\leq b$ and set

$$x^+ := z(\lambda^*),\ \alpha_{\mathcal{I}}^+ = \alpha_{\mathcal{E}}^+ = \gamma_{\mathcal{I}}^+ = \gamma_{\mathcal{A}}^+ = 0,\ \alpha_{\mathcal{A}}^+ := -Q_{\mathcal{A}} b_{\mathcal{A}} - Q_{\mathcal{A},\mathcal{I}} x_{\mathcal{I}}^+ - Q_{\mathcal{A},\mathcal{E}} a_{\mathcal{E}}^+ - q_{\mathcal{A}},$$
$$\gamma_{\mathcal{E}}^+ := -Q_{\mathcal{E}} a_{\mathcal{E}} - Q_{\mathcal{E},\mathcal{I}} x_{\mathcal{I}}^+ - Q_{\mathcal{E},\mathcal{A}} b_{\mathcal{A}}^+ - q_{\mathcal{E}},\ u^+ := \max(\min(x^+, b), a).$$

We note that also in this case condition (23) is satisfied, but the configuration is red. Moreover, after a Dual Update step we have

$$J(u^+) < J(u).$$

We summarize the previous analysis in the following lemma.

Lemma 21. *Suppose that a Dual Update step is started from the green configuration $(\mathcal{A}, \mathcal{E}, (x, \alpha, \gamma), u)$ where $a_i < x_i < b_i$ for $i \in \mathcal{I}$, resulting in the updated configuration $(\mathcal{A}^+, \mathcal{E}^+, (x^+, \alpha^+, \gamma^+), u^+)$. Then $J(u^+) < J(u)$.*

A.5.3 The new primal-dual triple $(x^+, \alpha^+, \gamma^+)$ for the Primal Update

Finally, we need to explain how we proceed in case of a Primal Update. In this case we proceed similar to the Dual Update. Thus we determine $[y, \beta, \delta] = KKT(\mathcal{A}^+, \mathcal{E}^+)$ and use u and y to investigate the line segment $z(\lambda)$. In this case we cannot avoid the possibility of having $u^+ = u$. If $a \leq y \leq b$ we proceed with

$$(x^+, \alpha^+, \gamma^+) \leftarrow (y, \beta, \delta),\ u^+ = y.$$

In this case we have $J(u^+) \leq J(u)$ and the new configuration is green.

Finally, if $y \not\leq b$ or $y \not\geq a$ we determine λ^* from (31) yielding $z(\lambda^*)$. We continue with

$$x^+ \leftarrow z(\lambda^*),\ \alpha_{\mathcal{I}}^+ = \alpha_{\mathcal{E}}^+ = \gamma_{\mathcal{I}}^+ = \gamma_{\mathcal{A}}^+ = 0,\ \alpha_{\mathcal{A}}^+ := -Q_{\mathcal{A}} b_{\mathcal{A}} - Q_{\mathcal{A},\mathcal{I}} x_{\mathcal{I}}^+ - Q_{\mathcal{A},\mathcal{E}} a_{\mathcal{E}}^+ - q_{\mathcal{A}},$$
$$\gamma_{\mathcal{E}}^+ := -Q_{\mathcal{E}} a_{\mathcal{E}} - Q_{\mathcal{E},\mathcal{I}} x_{\mathcal{I}}^+ - Q_{\mathcal{E},\mathcal{A}} b_{\mathcal{A}}^+ - q_{\mathcal{E}},\ u^+ := \max(\min(x^+, b), a).$$

Also in this case condition (23) holds and after a primal update we have $J(u^+) \leq J(u)$.

Summarizing the discussion so far in this section we give an algorithmic description of the modified SN-method in Table 14.

A.5.4 Convergence analysis

We now show global convergence of our modified SN-method for convex QPs with box constraints. The method generates a sequence of configurations which are either green or red. We emphasize that a green configuraton corresponding to the active sets $\mathcal{A}$ and $\mathcal{E}$ is uniquely determined by $\mathcal{A}$and $\mathcal{E}$ in the sense that $[x, \alpha] = KKT(\mathcal{A}, \mathcal{E})$ as well as $u = \max(\min(x, b), a)$ uniquely follow from $\mathcal{A}$ and $\mathcal{E}$.

The key observation is that a sequence of red configurations can have length at most $n - 1$.

Lemma 22. *Let $\mathcal{C}^{(i)} = (\mathcal{A}^{(i)}, \mathcal{E}^{(i)}, (x^{(i)}, \alpha^{(i)}, \gamma^{(i)}), u^{(i)}),\ i = 1, \ldots, k$ be a sequence of red iterates. Then $k \leq n - 1$ and $J(u^{(1)}) \geq J(u^{(k)})$.*

Proof. We note that an SN Update moves from either a red or a green configuration into a green configuration, hence such an update can not have occured during this sequence. Similarly, a Dual Update moves from a green configuration to either a red or a green configuration. Again, such an update can not have occured during this

```
Modified SN-method (MSN) for convex QPs with box-constraints
Start: $\mathcal{A}, \mathcal{E} \subseteq \mathcal{N},\ [x, \alpha, \gamma] = KKT(\mathcal{A}, \mathcal{E}),\ u = \max(\min(x, b), a)$
while $[x, \alpha, \gamma]$ not optimal
    Set $\mathcal{B} := \{i \in \mathcal{I} : x_i \geq b_i\} \cup \{i \in \mathcal{A} : \alpha_i \geq 0\}$ and $\mathcal{F} := \{i \in \mathcal{I} : x_i \leq a_i\} \cup \{i \in \mathcal{E} : \gamma_i \leq 0\}$
    Compute $y = KKT(\mathcal{B}, \mathcal{F})$ and $v = \max(\min(y, b), a)$
    Case 1: SN Update $J(v) < J(u)$
        Compute $[\beta, \delta] = KTT(\mathcal{B}, \mathcal{F})$ and continue with $[\mathcal{B}, \mathcal{F}, (y, \beta, \delta),\ v]$
    Case 2: Safeguard Update $J(v) \geq J(u)$
        Determine $(\mathcal{A}^+, \mathcal{E}^+)$:
        If $a_i < x_i < b_i,\ i \in \mathcal{I}$, then determine $(\mathcal{A}^+, \mathcal{E}^+)$ according to (24)
        else $\mathcal{A}^+ = \mathcal{A} \cup \{i \in \mathcal{I} : x_i \geq b_i\},\ \mathcal{E}^+ = \mathcal{E} \cup \{i \in \mathcal{I} : x_i \leq b_i\}$
        Compute $y = KKT(\mathcal{A}^+, \mathcal{E}^+)$
            If $y$ is primal feasible then compute $[\beta, \delta] = KKT(\mathcal{A}^+, \mathcal{E}^+)$ and continue with $[\mathcal{A}^+, \mathcal{E}^+, (y, \beta, \delta),\ y]$
            else Determine $\lambda^*$ according to (31), set $x^+ := z(\lambda^*),\ u^+ := \max(\min(x^+, b), a),\ \alpha^+_{\mathcal{I}} = \alpha^+_{\mathcal{E}} = 0,$
            $\gamma^+_{\mathcal{I}} = \gamma^+_{\mathcal{A}} = 0,\ \alpha^+_{\mathcal{A}} := -Q_{\mathcal{A}} b_{\mathcal{A}} - Q_{\mathcal{A},\mathcal{I}} x^+_{\mathcal{I}} - Q_{\mathcal{A},\mathcal{E}} a^+_{\mathcal{E}} - q_{\mathcal{A}},\ \gamma^+_{\mathcal{E}} := -Q_{\mathcal{E}} a_{\mathcal{E}} - Q_{\mathcal{E},\mathcal{I}} x^+_{\mathcal{I}} - Q_{\mathcal{E},\mathcal{A}} b^+_{\mathcal{A}} - q_{\mathcal{E}}$
            and continue with $[\mathcal{A}^+, \mathcal{E}^+, (x^+, \alpha^+, \gamma^+), u^+]$
endwhile
```

Table 14: Modified SN-Method for convex QPs with box constraints.

sequence. Thus we have carried out only Primal Updates in the whole sequence. But in each such update, the union of our two active sets increases by at least one member, so the number of steps is bounded by $n-1$, as we have got a green configuration if all indices are contained in one of our two active sets. Monotonicity of the values $J(u^{(i)})$ follows from the combinatorial line search. ☐

Lemma 23. *An update starting from a green configuration $(\mathcal{A}, \mathcal{E}, (x, \alpha, \gamma), u)$ to the updated configuration $(\mathcal{A}^+, \mathcal{E}^+, (x^+, \alpha^+, \gamma^+), u^+)$ has $J(u^+) < J(u)$.*

Proof. This follows from the definition in case of a SN Update, and by construction in case of a Dual Update, see Lemma 21. ☐

Looking at the subsequence of green iterates produced by our algorithm, we note that they yield a strictly monotonically decreasing sequence of function values $J(u)$ for feasible solutions u. Each of these u is determined (uniquely) through the active sets $\mathcal{A}$ and $\mathcal{E}$, hence we never generate the same green set twice. Furthermore there can be at most $n-1$ red configurations between two green configurations. Thus we have proved the main theorem of this paper also for the box constraint case.

Theorem 24. *For any convex QP with box constraints the modified SN-method terminates in a finite number of iterations with an optimal solution.*