



HAL
open science

ϵ Suboptimality Based Accelerated Termination for Quadratic Programming Using Dual Decomposition

Xiang Dai, Romain Bourdais, Hervé Guéguen

► **To cite this version:**

Xiang Dai, Romain Bourdais, Hervé Guéguen. ϵ Suboptimality Based Accelerated Termination for Quadratic Programming Using Dual Decomposition. 2021. hal-03266186v2

HAL Id: hal-03266186

<https://hal.science/hal-03266186v2>

Preprint submitted on 18 Jul 2021 (v2), last revised 22 Feb 2022 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ϵ Suboptimality Based Accelerated Termination for Quadratic Programming Using Dual Decomposition

Xiang Dai · Romain Bourdais · Hervé
Guéguen

Received: date / Accepted: date

Abstract In solving quadratic programming, analytical and dual decomposition based iterative method are 2 main approaches. However, the defects are evident: the former only works with small size problems and the latter only guarantees feasibility and optimality in the limit of iterations. In this paper, we

Xiang Dai, Corresponding author

Department of Automatic Control, IETR, CentraleSupélec
Rennes, France

xiang.dai@centralesupelec.fr

Romain Bourdais,

Department of Automatic Control, IETR, CentraleSupélec
Rennes, France

romain.bourdais@centralesupelec.fr

Hervé Guéguen,

Department of Automatic Control, IETR, CentraleSupélec
Rennes, France

herve.gueguen@centralesupelec.fr

propose a proactive method by combining these 2 methods to solve the optimal solution through dynamically identifying the active inequalities constraints. Further, to faster terminate the iterative process, we propose a suboptimal method based on cone programming to deliver feasible solutions with suboptimality guarantee. In addition to the mathematical proofs provided, various random simulations illustrate the effectiveness of the suboptimal method.

Keywords Quadratic programming · Suboptimality · Dual decomposition · Active set

Mathematics Subject Classification (2000) 49J53 · 49K99 · more

1 Introduction

Quadratic programming has long tracked massive interests in the society of control system, applied mathematics, and computer science, for it encompasses a large variety of application as computational geometry, finance, process networks, robotics, telecommunications, energy, and data confidentiality, etc[1].

In brief, iterative and analytical methods are 2 main approaches to solve such a problem. The former was first studied in 1956 by Frank and Wolfe [2], and has been advanced over last decades in combination with dual decomposition and methods for general nonlinear and convex optimization[3], for instance gradient method[4], steepest descent method[5], Newton's method[6], Quasi-Newton method[7] and conjugate gradient/direction method[8], etc. Generally, the the dual decomposition based iterative process converges to the optimal so-

lution, the convergence, however, is only guaranteed in the limits of iterations, so as feasibility[7].

The analytical methods, using multi-parametric quadratic programming (mp-QP) [9], was initially proposed in [10], where the initial state was deemed as the multi-variables to form an offline mapping by partitioning its Euclidean space into neighboring critical regions. That is obtained via two steps: first, solving a Karush-Kuhn-Tucker (KKT) condition based linear programming for a polyhedron with a given feasible starting point; second, visiting the opposite side of the polyhedron border (hyperplane) one by one to form other critical regions. A great deal of related research has been developed either to extend its application scope or to improve its efficiency, including reduction of unnecessary critical region partition [11], linear independence constrains qualifications free and semi-definite hessian cases [12], pruning infeasible set during critical region partition [13], using graph traversal algorithm to cope with degeneracy cases [14]. The "ergodic" property of critical region visit, however, limits its applicability only for small size system [15].

It is worth noting that in practice, visiting all active set combinations is extraordinarily time-consuming and memory exhausting even for medium size problem[16]. In particular, as a compromise, a suboptimal but strictly feasible (in some applications of security concerns or physical limits) solution is preferred over an optimal solution in practical application. In [17], suboptimality focused on variable has been investigated; however, no definite criterion of objective value suboptimality has been demonstrated, and the method may

generally fail large suboptimality. In [18], though arbitrary suboptimality is fulfilled, solving an exponentially increasing number of QP is required to build the stopping criterion.

In this paper, combining iterative and analytical methods, we propose a proactive method to search for the optimal active set during iterations, after which the optimal solution can be solved analytically. Furthermore, we propose a suboptimal method by extending the idea of suboptimality oriented faster stop in iterative process [19] to general quadratic programming. Taking advantage of variables during iterative process, the suboptimal method can deliver suboptimal and feasible solutions with no information of the optimal active set.

The main contributions of this paper are twofold. First, the combination of iterative and analytical methods can reduce the enumerations massively in solving analytical solutions. Second, the suboptimal method, with iteration number upper bounded by the proactive method, can generate feasible solutions with predefined suboptimality without information of the optimal active set, which enables an accelerated termination of iterative process prior to find the optimal active set.

This paper is organized as follows. Section 2 sets up the optimization problem and fundamentals. Section 3 proposes a proactive method to enumerate active set in searching for the optimal solution. Section 4 illustrates transformations from active inequality constraints into equality ones and proposes

criterion to generate ϵ primal solution. Numerical experiments and results discussions are presented in Section 5. And conclusions are given in Section 6.

Notation: For $n \in \mathbb{N}_{>0}$, \mathbb{S}_+^n and \mathbb{S}_{++}^n denote semidefinite positive and definite positive matrix of size $n \times n$ respectively. The norm $\|\cdot\|$ denotes the Euclidean norm, for $x \in \mathbb{R}^n$, $R \in \mathbb{S}_+^n$, $\|x\|_R = \sqrt{x^T R x}$, $\min \text{eig}(R)$ denotes the minimal eigenvalue of R . For matrix $A \in \mathbb{R}^{m_1 \times n}$ and $B \in \mathbb{R}^{m_2 \times n}$ ($m_1, m_2 \in \mathbb{N}_+$), $A \oplus B = (A^T, B^T)^T$, and $\text{rank}(A)$ denotes the rank of A . $\mathbf{1}_n$ denotes column vector with elements being 1 of size n .

2 Problem statement and fundamentals

In this section, the linear constrained quadratic problem dealt with in the paper is formulated. Besides, ϵ primal solution, iteration mechanism and definition of optimal active set are introduced as fundamentals of methods studied later.

2.1 Problem statement and main objective

In this paper, we consider a constrained quadratic problem as below:

$$\mathcal{J}^* = \min_{\mathbf{y}} \mathcal{J}(\mathbf{y}) \quad (1a)$$

$$s.t. \mathbf{A}\mathbf{y} = \mathbf{b}, \quad (1b)$$

$$\mathbf{C}\mathbf{y} \leq \mathbf{d}, \quad (1c)$$

where $\mathcal{J}(\mathbf{y}) = \frac{1}{2} \|\mathbf{y}\|_{\mathbf{R}}^2$, $\mathbf{y} \in \mathbb{R}^{ny}$, $\mathbf{R} \in \mathbb{S}_{++}^{ny}$, $\mathbf{A} \in \mathbb{R}^{n_e \times ny}$, $\text{rank}(\mathbf{A}) = n_r$, $\mathbf{b} \in \mathbb{R}^{n_e}$, $\mathbf{C} \in \mathbb{R}^{n_{ie} \times ny}$ and $\mathbf{d} \in \mathbb{R}^{n_{ie}}$.

Definition 2.1 (Feasible set) We define the feasible set of problem (1) as Y :

$$Y = \{\mathbf{y} \in \mathbb{R}^{ny} \mid \mathbf{A}\mathbf{y} = \mathbf{b}, \mathbf{C}\mathbf{y} \leq \mathbf{d}\}. \quad (2)$$

Assumption 1 We assume that Y is compact, closed and not empty, and there is at least one \mathbf{y} in the interior of Y . We also assume that $n_r < ny$.

Definition 2.2 (ϵ suboptimal solution) \mathbf{y} is said to be an ϵ primal solution of problem (1) if and only if $\mathbf{y} \in Y$ and

$$\mathcal{J}(\mathbf{y}) - \mathcal{J}^* \leq \epsilon. \quad (3)$$

The main objective of this paper is to find an efficient way to solve an ϵ primal solution of problem (1).

2.2 Dual problem and iterative process

In this subsection, we introduce dual variables to form the dual problem associated with (1), which can be solved by a general iterative process.

To begin with, the dual problem of problem (1) is defined as:

$$g^* = \max_{\boldsymbol{\theta}, \boldsymbol{\lambda} \geq 0} g(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \max_{\boldsymbol{\theta}, \boldsymbol{\lambda} \geq 0} \min_{\mathbf{y}} \mathcal{L}(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}), \quad (4)$$

$$\mathcal{L}(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \frac{1}{2} \|\mathbf{y}\|_{\mathbb{R}}^2 + \boldsymbol{\theta}^T (\mathbf{A}\mathbf{y} - \mathbf{b}) + \boldsymbol{\lambda}^T (\mathbf{C}\mathbf{y} - \mathbf{d}),$$

where $\boldsymbol{\theta} \in \mathbb{R}^{n_e}$ and $\boldsymbol{\lambda} \in \mathbb{R}_+^{n_{ie}}$ are the dual variables associated with constraint (1b) and (1c) respectively.

Remark 2.1 Since $\mathbf{R} \in \mathbb{S}_+^{ny}$, (1b) and (1c) are linear, problem (1) is convex. As the Slater's condition is satisfied by Assumption 1, thus the strong duality holds by Slater's theorem[5], namely $\mathcal{J}^* = g^*$.

To solve dual decomposition based problem (4) in a iterative manner, we initiate a general gradient method as:

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \alpha_{\boldsymbol{\theta}}^k (\mathbf{A}\mathbf{y}^k - \mathbf{b}), \quad (5a)$$

$$\boldsymbol{\lambda}^{k+1} = \max\{0, \boldsymbol{\lambda}^k + \alpha_{\boldsymbol{\lambda}}^k (\mathbf{C}\mathbf{y}^k - \mathbf{d})\}, \quad (5b)$$

$$\mathbf{y}^{k+1} = -\mathbf{R}^{-1}(\mathbf{A}^T \boldsymbol{\theta}^{k+1} + \mathbf{C}^T \boldsymbol{\lambda}^{k+1}), \quad (5c)$$

where $\alpha_{\boldsymbol{\theta}}^k, \alpha_{\boldsymbol{\lambda}}^k \in \mathbb{R}_{>0}$ are step size associated with $\boldsymbol{\theta}^k$ and $\boldsymbol{\lambda}^k$ respectively, and (5c) is obtained by substituting $\boldsymbol{\lambda}^{k+1}$ and $\boldsymbol{\theta}^{k+1}$ into $\nabla_{\mathbf{y}} \mathcal{L}(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\lambda})=0$.

Assumption 2 We assume that $\alpha_{\boldsymbol{\theta}}^k$ and $\alpha_{\boldsymbol{\lambda}}^k$ satisfy one of step size conditions as: minimization rule, Armijo rule and diminishing step size [7], such that the sequence $\{\mathbf{y}^k\}$ converges to the optimal solution of problem (1):

$$\lim_{k \rightarrow \infty} \mathbf{y}^k = \mathbf{y}^*. \quad (6)$$

Note that in implementing iterative process (5), the primal feasibility of \mathbf{y}^k can only be guaranteed in the limit of iterations.

Let $\mathcal{P} = \{1, \dots, n_{ie}\}$, we give definitions of active constraint, active set and inactive set respectively.

Definition 2.3 (Active set) During iterative process (5), the i -th constraint of (1c): $\mathbf{C}_i \mathbf{y} \leq \mathbf{d}_i$ ($\mathbf{C} = \bigoplus_{i=1}^{n_{ie}} \mathbf{C}_i$, $\mathbf{d} = \bigoplus_{i=1}^{n_{ie}} \mathbf{d}_i$), is said to be active at \mathbf{y}^k if $\mathbf{C}_i \mathbf{y}^k \geq \mathbf{d}_i$, i.e., the equality is reached or the inequality is violated; or $\boldsymbol{\lambda}_i^k > 0$,

its corresponding dual variable is turned positive. Denote \mathcal{A}^k and \mathcal{I}^k the active and the inactive set of constraints (1c) at \mathbf{y}^k , which are defined as:

$$\mathcal{A}^k = \{i \in \mathcal{P} \mid \mathbf{C}_i \mathbf{y} \geq \mathbf{d}_i \text{ or } \lambda_i^k > 0\}, \quad (7)$$

$$\mathcal{I}^k = \mathcal{P} \setminus \mathcal{A}^k. \quad (8)$$

Let \mathbf{y}^* denote the optimal solution of problem (1), we use \mathcal{A}^* and \mathcal{I}^* to denote the optimal active and inactive set, $\mathcal{A}^* = \{i \in \mathcal{P} \mid \mathbf{C}_i \mathbf{y}^* = \mathbf{d}_i\}$ ¹, $\mathcal{I}^* = \mathcal{P} \setminus \mathcal{A}^*$.

2.3 Active set enumeration approach

A linear programming (LP) presented in [13] can be used to test if a candidate active set \mathcal{A}^2 is equal to \mathcal{A}^* , whose solution is denoted as $(\mathbf{y}^*, \boldsymbol{\lambda}^*, \mathbf{s}^*, h^*)$ if it

¹ It is felicitous to replace " \geq " with "=", and remove the dual variable check at the optimal solution.

² Note that in active set enumeration approach, \mathcal{A}^k can be any subset of \mathcal{P} .

exists.

$$\text{LP}(\mathcal{A}) : \min_{\mathbf{y}, \boldsymbol{\lambda}_{\mathcal{A}}, \mathbf{s}_{\mathcal{I}}, h} -h \quad (9a)$$

$$s.t. \mathbf{R}\mathbf{y} + \mathbf{A}^T \boldsymbol{\theta} + \mathbf{C}_{\mathcal{A}}^T \boldsymbol{\lambda}_{\mathcal{A}} = \mathbf{0}, \quad (9b)$$

$$\mathbf{A}\mathbf{y} - \mathbf{b} = \mathbf{0}, \quad (9c)$$

$$\mathbf{C}_{\mathcal{A}}\mathbf{y} - \mathbf{d}_{\mathcal{A}} = \mathbf{0}, \quad (9d)$$

$$\mathbf{C}_{\mathcal{I}}\mathbf{y} - \mathbf{d}_{\mathcal{I}} + \mathbf{s}_{\mathcal{I}} = \mathbf{0}, \quad (9e)$$

$$h \cdot \mathbf{1}_{c_{\mathcal{A}}} \leq \boldsymbol{\lambda}_{\mathcal{A}}, \quad (9f)$$

$$h \cdot \mathbf{1}_{n_{ie} - c_{\mathcal{A}}} \leq \mathbf{s}_{\mathcal{I}}, \quad (9g)$$

$$0 \leq h, \quad (9h)$$

where $\mathbf{C}_{\mathcal{A}} = \bigoplus_{i \in \mathcal{A}} \mathbf{C}_i$, $\mathbf{C}_{\mathcal{A}} \in \mathbb{R}^{c_{\mathcal{A}} \times ny}$, $\mathbf{d}_{\mathcal{A}} = \bigoplus_{i \in \mathcal{A}} \mathbf{d}_i$, $\mathbf{d}_{\mathcal{A}} \in \mathbb{R}^{c_{\mathcal{A}}}$, $\mathbf{C}_{\mathcal{I}} = \bigoplus_{i \in \mathcal{I}} \mathbf{C}_i$, $\mathbf{C}_{\mathcal{I}} \in \mathbb{R}^{(n_{ie} - c_{\mathcal{A}}) \times ny}$, $\mathbf{d}_{\mathcal{I}} = \bigoplus_{i \in \mathcal{I}} \mathbf{d}_i$, $\mathbf{d}_{\mathcal{I}} \in \mathbb{R}^{n_{ie} - c_{\mathcal{A}}}$.

Note that we do not assume that the strict complementarity condition or linear independence constraint qualification (LICQ) in this paper³. As a result, \mathcal{A}^* is not necessary to be unique, for which case $\text{LP}(\mathcal{A})$ is still valid to generate \mathbf{y}^* .

³ The notion of LICQ and strict complementarity condition are borrowed from [20]. In this paper, LICQ is said to hold if matrix $\mathbf{A}_{\mathcal{A}^k}$, $\mathbf{A}_{\mathcal{A}^k} = \mathbf{A} \oplus \mathbf{C}_{\mathcal{A}^k}$ has full row rank; strict complementarity condition is said to hold if $\lambda_i > 0$ for each $\mathbf{C}_i \mathbf{y}^* = \mathbf{d}_i$.

3 Proactive optimal active set identification method (POASIM)

In this section, combining iterative process and active set enumeration approach, we propose a proactive method to solve for the optimal solution of problem (1) by enumerating \mathcal{A}^k during the iterative process, which requires fewer enumerations in total compared to conventional analytical method.

In [21] [13], the power set of \mathcal{P} is "passively" enumerated in search of \mathcal{A}^* , which would lead to explosive growth of enumeration number as problem size increases, i.e. $\sum_{i=1}^{ny-n_r} \binom{ny-n_r}{i}$, no matter what kind of pruning technique is applied, for intrinsically it cannot offset the boost of binomial coefficient.

By contrast, as presented in Alg. 1, proactive identification of \mathcal{A}^* along iterative process (5) is proposed, which enables a proactive enumeration of candidate active sets, and avoids massive meaningless combinations, i.e. $\binom{ny-n_r}{i}$, $i = 1, \dots, ny - n_r$. Note that \mathcal{A}^k and \mathcal{A}^{k-i} ($i = 1, \dots, k$) are possible to be identical, it is sufficient to test \mathcal{A}^k when it has not been tested before.

The optimal solution of problem (1) can be solved out if $\text{LP}(\mathcal{A}^k)$ has a solution in implementing POASIM, which, however, cannot be guaranteed to happen during the iterative process. Therefore, it is one crucial drawback of POASIM, and we will show that it can be overcome by the suboptimal method proposed in the next section.

Algorithm 1 Proactive Optimal Active Set Identification Method (POASIM)

```

1: Initialize:  $\theta^{-1}$ ,  $\lambda^{-1}$ ,  $k = 0$  and  $\epsilon$ .  $\mathbf{y}^{-1}$  is obtained by (5c).
2: repeat
3:   Update primal and dual variables by (5), update  $\mathcal{A}^k$  by (7)
4:   if LP( $\mathcal{A}^k$ ) has a solution then return  $\mathbf{y}^*$ 
5:   end if
6:    $k \leftarrow k + 1$ 
7: until  $\mathbf{y}^*$  is returned

```

4 Active set based ϵ suboptimal approach

In this section, based on Definition 2.3, we dynamically convert problem (1) into only equality constrained formulation during the iteration, whose optimal solution can be solved explicitly. With that and the best dual objective value of problem (4), we can check whether an ϵ primal solution of problem (1) is available under \mathcal{A}^k .

4.1 Transforming active constraints into equality constraints

Definition 2.3 can be used to identify the active and inactive constraints of (1c) at k -th iteration. By doing so, Problem (1) can be dynamically converted into the only equality constrained problem (denote its optimizer as $\mathbf{y}_{\mathcal{A}^k}^*$) as:

$$\mathcal{J}_{\mathcal{A}^k}^* = \min_{\mathbf{y}} \mathcal{J}(\mathbf{y}) \quad (10a)$$

$$s.t. \mathbf{A}_{\mathcal{A}^k} \mathbf{y} = \mathbf{b}_{\mathcal{A}^k}, \quad (10b)$$

where $\mathbf{A}_{\mathcal{A}^k} = \mathbf{A} \oplus \mathbf{C}_{\mathcal{A}^k}$, and $\mathbf{b}_{\mathcal{A}^k} = \mathbf{b} \oplus \mathbf{d}_{\mathcal{A}^k}$.

Lemma 4.1 $\mathbf{y}_{\mathcal{A}^k}^*$ can be computed by the linear equation group below:

$$\begin{cases} \mathbf{A}_{\mathcal{A}^k} \mathbf{y} = \mathbf{b}_{\mathcal{A}^k}, \\ \mathbf{F}_{\mathcal{A}^k}^T \mathbf{R} \mathbf{y} = \mathbf{0}, \end{cases} \quad (11)$$

where $\mathbf{F}_{\mathcal{A}^k}$ is an orthonormal null space matrix of $\mathbf{A}_{\mathcal{A}^k}$ satisfying $\mathbf{A}_{\mathcal{A}^k} \mathbf{F}_{\mathcal{A}^k} = \mathbf{0}$, and $\mathbf{F}_{\mathcal{A}^k} \in \mathbb{R}^{ny \times (ny - n_r - n_k)}$, $\text{rank}(\mathbf{F}_{\mathcal{A}^k}) = n_k$.

Proof First, we have $\text{rank}(\mathbf{F}_{\mathcal{A}^k}^T) = ny - n_r - n_k$. Since $\text{rank}(\mathbf{A}_{\mathcal{A}^k}) = n_r + n_k$, and $\mathbf{R} \in \mathbb{S}_{++}^{ny}$, $\mathbf{A}_{\mathcal{A}^k} \oplus \mathbf{F}_{\mathcal{A}^k}^T \mathbf{R}$ has full row rank, which means linear equation group (11) has a unique solution.

Here, we characterize the feasible set $Y_{\mathcal{A}^k}$ of problem (10) as:

$$\begin{aligned} Y_{\mathcal{A}^k} &= \{\mathbf{y} \in \mathbb{R}^{ny} \mid \mathbf{A}_{\mathcal{A}^k} \mathbf{y} = \mathbf{b}_{\mathcal{A}^k}\} \\ &= \{\hat{\mathbf{y}}_{\mathcal{A}^k} + \mathbf{F}_{\mathcal{A}^k} \mathbf{t}_{\mathcal{A}^k} \mid \mathbf{t}_{\mathcal{A}^k} \in \mathbb{R}^{ny - n_r - n_k}\}, \end{aligned} \quad (12)$$

this characterization is based on any point $\hat{\mathbf{y}}_{\mathcal{A}^k} \in Y_{\mathcal{A}^k}$.

It is trivial that the solution of the linear equation group is a feasible solution of problem (10) since (10b) is satisfied.

Next, we prove the optimality. By (12), problem (10) is equivalent as:

$$\mathbf{J}_{\mathcal{A}^k}^* = \min_{\mathbf{t}_{\mathcal{A}^k}} \mathbf{J}(\mathbf{t}_{\mathcal{A}^k}), \quad (13)$$

where $\mathbf{J}_{\mathcal{A}^k}(\mathbf{t}_{\mathcal{A}^k}) = \frac{1}{2} \|\hat{\mathbf{y}}_{\mathcal{A}^k} + \mathbf{F}_{\mathcal{A}^k} \mathbf{t}_{\mathcal{A}^k}\|_{\mathbf{R}}^2$, and $\mathbf{J}_{\mathcal{A}^k}^* = \mathcal{J}_{\mathcal{A}^k}^*$.

Accordingly, we have

$$\nabla \mathbf{J}_{\mathcal{A}^k}(\mathbf{t}_{\mathcal{A}^k}) = \mathbf{F}_{\mathcal{A}^k}^T \mathbf{R} (\hat{\mathbf{y}}_{\mathcal{A}^k} + \mathbf{F}_{\mathcal{A}^k} \mathbf{t}_{\mathcal{A}^k}), \quad (14)$$

$$\nabla^2 \mathbf{J}_{\mathcal{A}^k}(\mathbf{t}_{\mathcal{A}^k}) = \mathbf{F}_{\mathcal{A}^k}^T \mathbf{R} \mathbf{F}_{\mathcal{A}^k}. \quad (15)$$

For $\forall \mathbf{t}_{\mathcal{A}^k} \in \mathbb{R}^{ny-n_r-c_k}$, we can assign a feasible solution of (10b), denoted as $\mathbf{y}_{\mathcal{A}^k}$, to have

$$\mathbf{y}_{\mathcal{A}^k} = \hat{\mathbf{y}}_{\mathcal{A}^k} + \mathbf{F}_{\mathcal{A}^k} \mathbf{t}_{\mathcal{A}^k}, \quad (16)$$

then we have

$$\nabla \mathbf{J}_{\mathcal{A}^k}(\mathbf{t}_{\mathcal{A}^k}) = \mathbf{F}_{\mathcal{A}^k}^T \mathbf{R} \mathbf{y}_{\mathcal{A}^k}. \quad (17)$$

Next, we have the necessary and sufficient optimality condition of unconstrained convex optimization (13) as:

$$\|\nabla \mathbf{J}_{\mathcal{A}^k}(\mathbf{t}_{\mathcal{A}^k})\| = 0. \quad (18)$$

And we can conclude the proof by (11). ■

4.2 Active set based ϵ suboptimal criterion

To emphasis the main contribution, henceforth we delineate the methodology only for case $c_k + n_r < ny$ ⁴. Let $g_{best}^k = \sup_{i \leq k, i \in \mathbb{N}_{>0}} g(\boldsymbol{\theta}^i, \boldsymbol{\lambda}^i)$. By the primal-dual theory[5], we have $g_{best}^k \leq \mathcal{J}^*$. Combining with Definition 2.2, if $\mathbf{y} \in Y$, and satisfies

$$\mathcal{J}(\mathbf{y}) - g_{best}^k \leq \epsilon, \quad (19)$$

then \mathbf{y} is an ϵ primal solution of problem (1).

⁴ The case $c_k + n_r > ny$ is unsolvable, executing $\mathcal{A}^k \leftarrow \mathcal{A}^k \setminus \{i \in \mathcal{A}^k \mid \lambda_i^k < \bar{\lambda}^k\}$ ($\bar{\lambda}^k$ denote the ny -th largest value of $\boldsymbol{\lambda}_i^k$), it can be converted into case $c_k + n_r = ny$, for which case the only solution can be obtained by (10b).

In solving an ϵ primal solution of problem (1), condition (19) enables us to further reduce the gap between g^k and \mathcal{J}^* by taking advantage of the sequence $\{g_{best}^k\}$ generated. Since the solution of problem (10) is not unique, we need the relation between g_{best}^k and $\mathbf{y}_{\mathcal{A}^k}^*$ as a base to build the criterion for ϵ suboptimality.

There are 3 possibilities between $\mathcal{J}_{\mathcal{A}^k}^*$ and g_{best}^k depending on the identification correctness of \mathcal{A}^k , and the gap between g_{best}^k and \mathcal{J}^* :

$$\mathcal{J}_{\mathcal{A}^k}^* < g_{best}^k, \quad (20a)$$

$$0 \leq \mathcal{J}_{\mathcal{A}^k}^* - g_{best}^k \leq \epsilon, \quad (20b)$$

$$\epsilon < \mathcal{J}_{\mathcal{A}^k}^* - g_{best}^k. \quad (20c)$$

Note that in case (20c), no definite ϵ suboptimality criterion can be devised without knowing \mathcal{A}^* , since even $\mathcal{J}_{\mathcal{A}^k}^*$, the optimum by far fails (19). The following lemma and corollary will be used to build criteria for ϵ suboptimality when (20a) and (20b) are satisfied respectively.

Proposition 4.1 Denote $\beta_{\mathcal{A}^k} = \min \text{eig}(\mathbf{F}_{\mathcal{A}^k}^T \mathbf{R} \mathbf{F}_{\mathcal{A}^k})$. Given $\Delta \in R_{>0}$. If \mathbf{y} satisfies (10b) and

$$\|\mathbf{F}_{\mathcal{A}^k}^T \mathbf{R} \mathbf{y}\| \leq (2\beta_{\mathcal{A}^k} \Delta)^{\frac{1}{2}}, \quad (21)$$

then we have

$$\mathcal{J}(\mathbf{y}) - \mathcal{J}_{\mathcal{A}^k}^* \leq \Delta. \quad (22)$$

Proof As \mathbf{y} satisfies (10b), it is a feasible solution of problem (10). Since problem (13) is convex and unconstrained, and $\nabla^2 \mathcal{J}_{\mathcal{A}^k}(\mathbf{t}_{\mathcal{A}^k})$ is lower bounded by

$\beta_{\mathcal{A}^k}$ (because $\mathbf{A}_{\mathcal{A}^k}$ is full row rank), then (22) holds by applying (9.10) of [5].

■

Now, consider the cone programming (CP) below, denote its optimizer, if it exists, as $(\mathbf{y}_{sub}, s_{\mathcal{I}^k}^1, h^1)$.

$$\text{CP}(\mathcal{A}^k) : \min_{\mathbf{y}, s_{\mathcal{I}^k}, h} -h \quad (23a)$$

$$s.t. \text{ (10b),}$$

$$\mathbf{C}_{\mathcal{I}^k} \mathbf{y} - \mathbf{d}_{\mathcal{I}^k} + \mathbf{s}_{\mathcal{I}^k} = \mathbf{0}, \quad (23b)$$

$$h \cdot \mathbf{1}_{n_{ie} - c_{\mathcal{A}^k}} \leq \mathbf{s}_{\mathcal{I}^k}, \quad (23c)$$

$$0 \leq h, \quad (23d)$$

$$\|\mathbf{F}_{\mathcal{A}^k}^T \mathbf{R} \mathbf{y}\| \leq (2\beta_{\mathcal{A}^k}(\epsilon + g_{best}^k - \mathcal{J}_{\mathcal{A}^k}^*))^{\frac{1}{2}}. \quad (23e)$$

Based on ϵ suboptimality criteria demonstrated in Proposition 4.1, we can use CP (\mathcal{A}^k) to obtain an ϵ primal solution of problem (1), which is illustrated in the following 2 propositions.

Proposition 4.2 *If CP (\mathcal{A}^k) has a solution and*

$$\mathcal{J}_{\mathcal{A}^k}^* - g_{best}^k \leq \epsilon, \quad (24)$$

then \mathbf{y}_{sub} is an ϵ primal solution of problem (1).

Proof First, consider case (20a), let $\delta^k = g_{best}^k - \mathcal{J}_{\mathcal{A}^k}^*$. As \mathbf{y}_{sub} satisfies (10b), (9e), (9g), (9h), we have $\mathbf{A}_{\mathcal{A}^k} \mathbf{y}_{sub} = \mathbf{b}_{\mathcal{A}^k}$, and $\mathbf{C}_{\mathcal{I}^k} \mathbf{y}_{sub} \leq \mathbf{d}_{\mathcal{I}^k}$. Since at each iteration, \mathbf{C} consists of $\mathbf{C}_{\mathcal{I}^k}$ and $\mathbf{C}_{\mathcal{A}^k}$, then by (2), $\mathbf{y}_{sub} \in Y$, then by the

primal-dual theory[5],it gives

$$0 \leq \mathcal{J}(\mathbf{y}_{sub}) - g_{best}^k. \quad (25)$$

As (23e) is satisfied by \mathbf{y}_{sub} , we have by Proposition 4.1:

$$0 \leq \mathcal{J}(\mathbf{y}_{sub}) - g_{best}^k \leq \epsilon. \quad (26)$$

As a consequence, combining $\mathbf{y}_{sub} \in Y$, (25) and (26), \mathbf{y}_{sub} is an ϵ primal solution of problem (1) by (19).

Second, consider case (20b), let $\Delta^k = \mathcal{J}_{\mathcal{A}^k}^* - g_{best}^k$, we have by (22) that: $\mathcal{J}(\mathbf{y}) - \mathcal{J}_{\mathcal{A}^k}^* \leq \epsilon - \Delta^k$. Namely, $\mathcal{J}(\mathbf{y}) - g_{best}^k \leq \epsilon$. The rest of the proof remains the same as that of case (20a). ■

In a cost ascending order, Alg. 2 presents The complete algorithm of computing an ϵ primal solution of problem (1): after obtaining $\mathcal{A}^k \mathbf{y}_{\mathcal{A}^k}^*$ at k -th iteration; firstly, check whether $\mathbf{y}_{\mathcal{A}^k}^*$ is an ϵ primal solution; if not, check whether \mathcal{A}^k is an optimal active set by $\text{LP}(\mathcal{A}^k)$; if not, check whether an ϵ primal solution can be found by CP (\mathcal{A}^k) using g_{best}^k .

4.3 Optimization properties of ASBSA

From here, we will use the following 2 lemmas to derive the lower bound of ϵ : for any value above the bound, ASBSA can terminate within finite iterations.

Lemma 4.2 *Under Assumption 2, (24) can be satisfied within finite iterations in implementing ASBSA.*

Algorithm 2 Active Set Based Suboptimal Algorithm (ASBSA)

```

1: Initialize:  $\theta^{-1}$ ,  $\lambda^{-1}$ ,  $k = 0$  and  $\epsilon$ .  $\mathbf{y}^{-1}$  is obtained by (5c).
2: repeat
3:   Update primal and dual variables by (5), update  $\mathcal{A}^k$  by (7), compute  $\mathbf{y}_{\mathcal{A}^k}^*$  by (11)
4:   if  $\mathbf{y}_{\mathcal{A}^k}^* \in Y$  then
5:     if  $\mathbf{y}_{\mathcal{A}^k}^*$  satisfies (19) then return  $\mathbf{y}_{\mathcal{A}^k}^*$ 
6:     end if
7:   end if
8:   if LP( $\mathcal{A}^k$ ) has a solution for  $\mathcal{A}^k$  then return  $\mathbf{y}^*$ 
9:   end if
10:  if  $\mathbf{y}_{\mathcal{A}^k}^*$  satisfies (24) then
11:    if CP ( $\mathcal{A}^k$ ) has a solution then return  $\mathbf{y}_{sub}$ 
12:    end if
13:  end if
14:   $k \leftarrow k + 1$ 
15: until one of  $\mathbf{y}_{\mathcal{A}^k}^*$ ,  $\mathbf{y}^*$  and  $\mathbf{y}_{sub}$  is returned

```

Proof First, consider the following problem: $\delta = \inf_{i \in \mathcal{I}^*} \{\min_{\mathbf{y}} \|\mathbf{y} - \mathbf{y}^*\| \mid \mathbf{C}_i \mathbf{y} = \mathbf{d}_i\}$. Then, as \mathbf{y}^k asymptotically converges to \mathbf{y}^* by Assumption 2, there exists a k_1 such that $\forall k \geq k_1$, $\|\mathbf{y}^k - \mathbf{y}^*\| \leq \delta$. So, for \mathcal{A}^k generated by (7), $\forall k \geq k_1$, we have $\mathcal{I}^* \cap \mathcal{A}^k = \emptyset$. Therefore, we have $\mathcal{A}^k \subset \mathcal{A}^*$ for $\forall k \geq k_1$, which means Y is a proper subset of the feasible set of problem (10), thus we have:

$$\mathcal{J}_{\mathcal{A}^k}^* \leq \mathcal{J}^*, \quad \forall k \geq k_1. \quad (27)$$

For a given $\epsilon > 0$, there exists a k_2 by Remark 2.1 such that

$$\mathcal{J}^* - g_{best}^k \leq \epsilon, \quad \forall k \geq k_2. \quad (28)$$

Consequently, combining (27) and (28), we have

$$\mathcal{J}_{\mathcal{A}^k}^* - g_{best}^k \leq \epsilon, \quad \forall k \geq \max\{k_1, k_2\}. \quad (29)$$

And this completes the proof. ■

Here, consider the following norm minimization:

$$\bar{\delta} = \min_{\mathbf{y}} \|\mathbf{F}^T \mathbf{R} \mathbf{y}\|, \quad s.t. \mathbf{y} \in Y,$$

where $\mathbf{F} \in \mathbb{R}^{ny \times (ny - n_r)}$ is a orthonormal null space matrix of \mathbf{A} satisfying $\mathbf{A} \mathbf{F} = \mathbf{0}$.

Lemma 4.3 *Let $\beta_{\mathbf{A}} = \min \text{eig}(\mathbf{F}^T \mathbf{R} \mathbf{F})$. Under Assumption 2, for $\forall \epsilon > \bar{\delta}/2\beta_{\mathbf{A}}$, an ϵ suboptimal solution of problem (1) can be generated within finite iterations in implementing ASBSA.*

Proof By Lemma 4.2, given an arbitrary $\epsilon' > 0$, there exist a k such that

$$\mathcal{J}_{\mathcal{A}^k}^* - g_{best}^k \leq \epsilon'. \quad (30)$$

Next, consider the problem:

$$\delta^k = \min_{\mathbf{y}} \|\mathbf{F}_{\mathcal{A}^k}^T \mathbf{R} \mathbf{y}\|, \quad s.t. (10b), \mathbf{C}_{\mathcal{I}^k} \mathbf{y} \leq \mathbf{d}_{\mathcal{I}^k}.$$

Observing CP (\mathcal{A}^k), for $\epsilon^k = (\delta^k)^2/2\beta_{\mathcal{A}^k} + \mathcal{J}_{\mathcal{A}^k}^* - g_{best}^k$, an ϵ^k suboptimal solution of problem (1) can be found by solving CP (\mathcal{A}^k).

By (30), we have that an $((\delta^k)^2/2\beta_{\mathcal{A}^k} + \epsilon')$ suboptimal solution of problem (1) can be generated with at most k iterations. If it can be shown that

$$(\delta^k)^2/2\beta_{\mathcal{A}^k} \leq \bar{\delta}^2/2\beta_{\mathbf{A}}, \quad (31)$$

then for $\epsilon = \bar{\delta}/2\beta_{\mathbf{A}} + \epsilon'$, an ϵ suboptimal solution of problem (1) can be generated with at most k iterations. Since ϵ' can be arbitrary small, we have that for $\forall \epsilon > \bar{\delta}/2\beta_{\mathbf{A}}$, an ϵ suboptimal solution of problem (1) can be generated within finite iteration.

We give the proof of (31) from here. If $\mathbf{A}_{\mathcal{A}^k} = \mathbf{A}$, (31) trivially holds. We consider the case $\mathbf{A}_{\mathcal{A}^k} \neq \mathbf{A}$, which indicates $\mathbf{A}_{\mathcal{A}^k} = \mathbf{A} \oplus \mathbf{C}_{\mathcal{A}^k}$. Subsequently, by $\mathbf{A}\mathbf{F} = \mathbf{0}$, $\mathbf{A}\mathbf{F}_{\mathcal{A}^k} = \mathbf{0}$, and $\mathbf{C}_{\mathcal{A}^k}\mathbf{F}_{\mathcal{A}^k} = \mathbf{0}$, the null space of $\mathbf{A}_{\mathcal{A}^k}$ is a subspace of the null space of \mathbf{A} . Since $\mathbf{F} \in \mathbb{R}^{ny \times (ny - n_r)}$ and $\mathbf{F}_{\mathcal{A}^k} \in \mathbb{R}^{ny \times (ny - n_r - c_k)}$, there exists a semi-orthogonal matrix $\mathbf{P} \in \mathbb{R}^{(ny - n_r) \times (ny - n_r - c_k)}$ with $\mathbf{P}^T \mathbf{P} = \mathbf{I}$, such that $\mathbf{F}\mathbf{P} = \mathbf{F}_{\mathcal{A}^k}$. It follows $\mathbf{F}_{\mathcal{A}^k}^T \mathbf{R} \mathbf{F}_{\mathcal{A}^k} = \mathbf{P}^T \mathbf{F}^T \mathbf{R} \mathbf{F} \mathbf{P}$. Then by Poincaré separation theorem[22],

$$\beta_{\mathbf{A}} \leq \beta_{\mathcal{A}^k}. \quad (32)$$

For any $\mathbf{y} \in Y$,

$$\|\mathbf{F}_{\mathcal{A}^k}^T \mathbf{R} \mathbf{y}\| = \|\mathbf{P}^T \mathbf{F}^T \mathbf{R} \mathbf{y}\| \leq \|\mathbf{P}^T\| \|\mathbf{F}^T \mathbf{R} \mathbf{y}\| = \|\mathbf{F}^T \mathbf{R} \mathbf{y}\|, \quad (33)$$

where the inequality uses Cauchy-Schwarz inequality, and the equality uses the property of semi-orthogonal matrix that $\|\mathbf{P}^T\| = \|\mathbf{P}\| = 1$.

Finally, (31) can be concluded by (32) and (33), and this completes the proof. ■

Remark 4.1 Note that $\epsilon > \bar{\delta}/2\beta_{\mathbf{A}}$ is a sufficient condition for ASBSA to be terminated within finite iterations. In practice, it is possible to take ϵ much lower than $\bar{\delta}/2\beta_{\mathbf{A}}$, which will be illustrated with a numerical example in Subsection 5.1.

Remark 4.2 Suppose that POASIM terminates with k iterations, since POASIM and ASBSA share iterative process (5), and $\text{LP}(\mathcal{A}^k)$ (test if \mathcal{A}^k is \mathcal{A}^* or not), then ASBSA can return \mathbf{y}^* with k iterations. Alternatively, if one of $\mathbf{y}_{\mathcal{A}^k}^*$ or \mathbf{y}_{sub} is returned prior to k -th iteration, ASBSA terminates with less than k iterations. All in all, ASBSA terminates for $\forall \epsilon > 0$ with the same or fewer iterations than that of POASIM.

In terms of ASBSA, economic computation techniques as follows can further improve the efficiency of implementation:

1. for each distinct \mathcal{A}^k , its according variables $\mathbf{C}_{\mathcal{A}^k}, \mathbf{d}_{\mathcal{A}^k}, \mathbf{C}_{\mathcal{I}^k}, \mathbf{d}_{\mathcal{I}^k}, \mathbf{y}_{\mathcal{A}^k}^*, \mathbf{F}_{\mathcal{A}^k}$ and $\mathcal{J}_{\mathcal{A}^k}^*$ can be stored, then if $\mathcal{A}^{k+i} = \mathcal{A}^k, i = 1, 2, \dots$, the above mentioned variables can be retrieved from the stored data, instead to compute from the scratch⁵;
2. we require that \mathcal{A}^k in Step 8 of ASBSA has not been tested by $\text{LP}(\mathcal{A}^k)$ before⁶. Likewise, we require that \mathcal{A}^k in Step 11 of ASBSA has not been tested by CP (\mathcal{A}^k) before;
3. at each iteration, if $\mathbf{y}_{\mathcal{A}^k}^* \in Y$ and satisfies (19), then ASBSA can be terminated without excess computation. In addition, to avoid unnecessary solving of $\text{LP}(\mathcal{A}^k)$ and $\text{CP}(\mathcal{A}^k)$, for each $\mathbf{y}_{\mathcal{A}^k}^* \in Y$ and $\mathcal{J}(\mathbf{y}) - g_{best}^{\bar{k}} > \epsilon$, denote $D_{\mathcal{A}^k} = \mathcal{J}_{\mathcal{A}^k}^* - \epsilon$, then at every $\bar{k} > k$, we can claim that $\mathbf{y}_{\mathcal{A}^k}^*$ is an

⁵ This technique is also applied to POASIM.

⁶ In the same argument, we require that \mathcal{A}^k in Step 4 of POASIM has not been tested by $\text{LP}(\mathcal{A}^k)$ before in POASIM.

ϵ suboptimal solution of problem (1) if

$$g_{best}^k \geq D_{\mathcal{A}^k}. \quad (34)$$

5 Numerical experiments

In this section, we carry out 2 groups of experiments: firstly, a single small size problem to give a clear-cut comparison of time and iteration number magnitude among the 2 algorithms proposed in this paper (POASIM and ASBSA), and 2 existing methods (Nesterov gradient descent[23] [24], Gupta method[13]); secondly, we consider 1000 randomly generated tests to demonstrate general performance between POASIM and ASBSA. The complete datasets generated in numerical experiment are available at <https://github.com/SettingTheWorld/epsilon-Suboptimality-QP>.

To detail each problem, inequality constraints (1c) are $0 \leq \mathbf{y} \leq 1$, the penalty matrix is $\mathbf{R} = \mathbf{I}$. For each test, n_e is a integer randomly drawn from uniform distribution $(0, ny)$, each entry of \mathbf{A} is randomly drawn from uniform distribution $(-0.5, 0.5)$, and each entry of \mathbf{b} is randomly drawn from uniform distribution $(0, \mathbf{A}_i \cdot \mathbf{1}_{ny})$ to make problem (1) feasible, where \mathbf{A}_i denote the i -th row of \mathbf{A} . Specifically, we adopt Nesterov gradient descent[23] for iteration (5a) (5b)⁷, which is proved to be the best first order gradient method [3]. All numerical experiments are carried out using MATLAB 2021a on a Windows 10

⁷ $\boldsymbol{\theta}^{k+1} = \hat{\boldsymbol{\theta}}^k + \frac{1}{L}(\mathbf{A}\hat{\mathbf{y}}^k - \mathbf{b})$, $\boldsymbol{\lambda}^{k+1} = \max\{0, \hat{\boldsymbol{\lambda}}^k + \frac{1}{L}(\mathbf{C}\hat{\mathbf{y}}^k - \mathbf{d})\}$, where for a vector $\boldsymbol{\nu}$, $\hat{\boldsymbol{\nu}}^k = \boldsymbol{\nu}^k + \frac{k-1}{k+2}(\boldsymbol{\nu}^k - \boldsymbol{\nu}^{k-1})$, $L = \|\mathbf{E}\mathbf{R}^{-1}\mathbf{E}^T\|_2$, and $\mathbf{E} = (\mathbf{A}^T, \mathbf{C}^T)^T$

Table 1: Performance comparison among POASIM, Nesterov gradient descent, Gupta method and ASBSA of a single test with $ny=10$, and predefined relative ϵ as 1×10^{-2}

	Gupta***	Nesterov		POASIM	ASBSA
		ϵ^*	$\epsilon + Y^{**}$		
Computation time (s)	10	5.8×10^{-3}	4.3×10^{-1}	8.9×10^{-2}	1.7×10^{-2}
# of iterations or (9) solved	911	15	74803	20	15
Primal feasible	yes	no	yes	yes	yes

* solved by (5a)-(5b), a posterior ϵ suboptimality criterion is used: $p^* - \tilde{g}(\tilde{\theta}^k, \tilde{\lambda}^k) \leq 0.01p^*$, where p^* is known as a parameter.

** solved by (5a)-(5b), ϵ suboptimality criterion: $p^* - \tilde{g}(\tilde{\theta}^k, \tilde{\lambda}^k) \leq 0.01p^*$, feasibility criterion: $\|\mathbf{A}\mathbf{y} - \mathbf{b}\| \leq 1 \times 10^{-16}$, $\mathbf{C}\mathbf{y} - \mathbf{d} \leq 1 \times 10^{-16}$, the magnitude 1×10^{-16} is computed from optimal solution \mathbf{y}^* that solved by POASIM.

*** Gupta method refers to the algorithm in [13]. In this test, 18779 candidate sets are pruned, making in total 19690 candidate sets enumerated.

PC with 2.20 GHz Core i7-8750H CPU and 16GB RAM, $LP(\mathcal{A}^k)$ and $CP(\mathcal{A}^k)$ are solved by MATLAB solver `linprog` and `coneprog` respectively.

5.1 Single test comparison among 4 methods

To initiate a perception of POASIM, ASBSA, iterative process (5) and Gupta method, here we present the comparison by a small size ($ny = 10$, $n_e = 5$) test in Table. 1. Unsurprisingly, the Gupta method shows the explosion of candi-

date sets enumerated and overwhelming defect of time, which is incompetent for larger size problems. When ϵ suboptimality is only concerned, Nesterov gradient descent discloses evident superiority in iteration number and time. However, if feasibility is required, Nesterov gradient descent requires 5000 times more iterations, which results in worse time performance than our 2 methods. POASIM and ASBSA promised to deliver optimal (obviously feasible) solutions, reveal favorable results in iteration number and time, and hence will be investigated further with randomly generated problems of larger size in the next subsection.

Note that, in this test, $\bar{\delta}/2\beta_{\mathbf{A}} = 0.0348$, if we take $\epsilon = \bar{\delta}/2\beta_{\mathbf{A}}$, then the corresponding relative suboptimality is 0.0424, which is a sufficient condition for ASBSA to be terminated within finite iterations. In fact, it is quite safe to take much smaller ϵ in practice, e.g. in the next subsection, predefined relative suboptimality is set as 0.0001, and all the tests can generate ϵ suboptimal solution within in finite iterations.

5.2 Random tests between POASIM and ASBSA

In this subsection, we use 1000 independent randomly generated linear constrained quadratic problem to test POASIM and ASBSA; for each problem, ASBSA is tested under 4 different relative suboptimality⁸: 0.0001, 0.001, 0.01 and 0.1. For each random problem, $ny = 100$, and sparsity concerning matrix

⁸ The relative suboptimality is computed as suboptimality divided by \mathcal{J}^* .

Table 2: Average and maximal relative error of ASBSA (The magnitude of 1×10^{-4} , 1×10^{-3} , 1×10^{-2} and 1×10^{-1} are omitted from the results according to the relative suboptimality referred for space-saving. As a consequence, any result presented with value less than 1 means that predefined suboptimality is fulfilled.)

Predefined Rel. Subopt.	10^{-4}	10^{-3}	10^{-2}	10^{-1}
Ave. Rel. Error	0.15	0.06	0.21	0.25
Max. Rel. Error	0.81	0.91	0.84	0.78

\mathbf{A} in equality constraints (1b) are randomly drawn from uniform distribution $(0, 1)$.

Table 2 shows that the predefined suboptimality of all random tests is fulfilled, which in general is significantly larger than the average relative error.

From Fig. 1, as predefined relative suboptimality increases from 1×10^{-4} to 1×10^{-1} , the range of boxplot decreases gradually, indicating an increasing iteration number of ASBSA. The reason behind this is the higher suboptimality, the higher tolerance of incorrectness of \mathcal{A}^k , and the higher possibility for (20a) or (20b) to occur, thus the higher probability for $\text{CP}(\mathcal{A}^k)$ to have a solution. This tendency is also shown in Table. 3, where the average number of CP calculated and the number of \mathbf{y}_{sub} grows dramatically as predefined relative suboptimality increases, which in return has an undesired effect on the time performance of ASBSA. Together with the following facts: computation time ratio of one CP to one iterate of (5) is $4 - 5 \times 10^4$, of one CP to one LP ranges

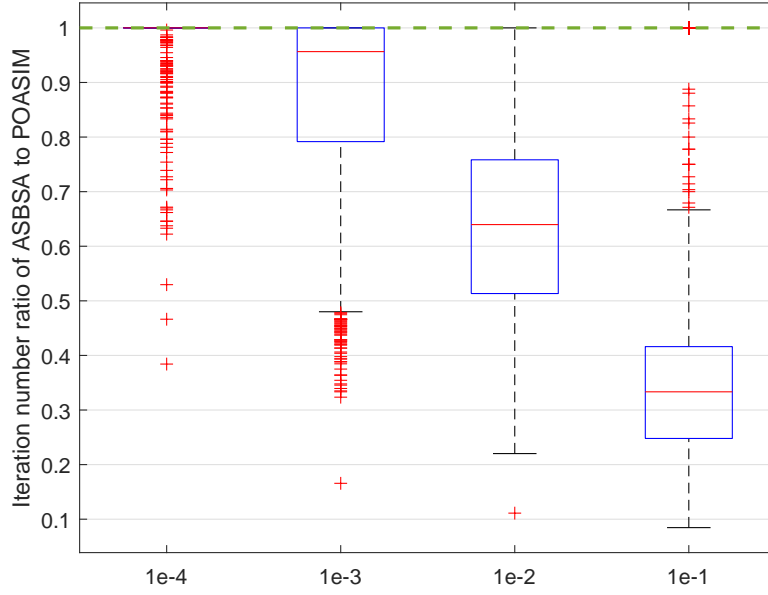


Fig. 1: Iteration number ratio of ASBSA to POASIM (Sample value exceeded $\pm 2.7\sigma$ shows as whisker, same setting for other box plots. Sample value less, greater than or equals to 1 (green horizontal line) means ASBSA consumes less, more or the same iteration as POASIM in the same test. The lower value, the better performance of ASBSA.)

from 3.1 to 5. The more CP calculated, the more likely that ASBSA consumes a longer time in total because for a \mathcal{A}^k , the calculation of CP comes after an LP is failed, resulting 2 solving process for one \mathcal{A}^k . Nevertheless, as shown in Fig. 2, ASBSA dominates time performance with comparison to POASIM, even though the superiority is blunted somewhat under 1×10^{-2} and 1×10^{-1} . Note that the considerable number of $\mathbf{y}_{\mathcal{A}^k}^*$ and \mathbf{y}^* (the first 3 columns of Ta-

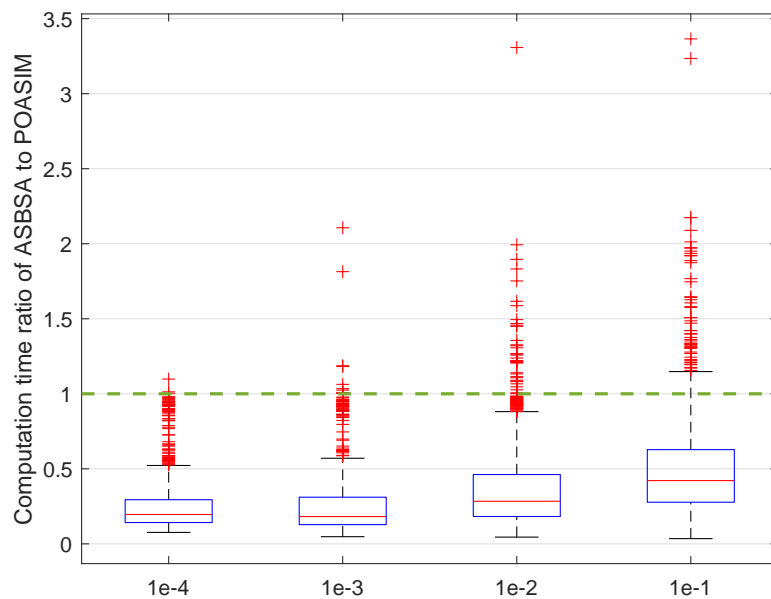


Fig. 2: Ratio of ASBSA to POASIM on computation time (Sample value less, greater than or equals to 1 (green horizontal line) means ASBSA spends less, more or the same computation time as POASIM in the same test. The lower value, the better performance of ASBSA.)

ble. 3) returned as ϵ primal suboptimal solution can also account for time supremacy of ASBSA, which spare the effort in computing CP, leading even less time demanded.

6 Conclusion

In solving quadratic programming, through the combination of active set enumeration and general gradient method, we have proposed a proactive method

Table 3: Statistics of 1000 random tests: termination by different approaches in ASBSA and calculation number of CP

Predefined Rel. Subopt.	Total # returned				Ave. # of CP
	$D_{\mathcal{A}^k}$	$\mathbf{y}_{\mathcal{A}^k}^*$	\mathbf{y}^*	\mathbf{y}_{sub}	Calculated
10^{-4}	51*	24**	899	26	0.039
10^{-3}	210	193	379	218	0.270
10^{-2}	109	126	68	697	0.972
10^{-1}	27	31	49	893	2.079

* data in column $D_{\mathcal{A}^k}$ means that the ϵ suboptimal solution is found by (34).

** data in column $\mathbf{y}_{\mathcal{A}^k}^*$ means that the ϵ suboptimal solution is found by Step 5 in ASBSA.

to obtain the optimal active set in an iterative manner, by which the optimal solution can be solved out. In the hope to terminate the iterative process faster, we have further initiated a suboptimal method based on cone programming to generate suboptimal and feasible solutions, which requires no information on the optimal active set. Furthermore, we have demonstrated the lower bound of suboptimality, which can ensure the iterative process to be terminated within finite iterations.

Through random numerical experiments, the ϵ -suboptimality and feasibility have been verified for the suboptimal method, which has moreover revealed statistical improvement of computation time and iteration number under certain predefined relative suboptimality.

Future work can be addressed in the following aspects for the 2 methods proposed: exploitation of the performance for various problem sizes and settings, application in model predictive control, comparison with primal[20] and dual[25] active set method.

References

1. Furini, F., Traversi, E., Belotti, P., Frangioni, A., Gleixner, A., Gould, N., Liberti, L., Lodi, A., Misener, R., Mittelmann, H., Sahinidis, N.V., Vigerske, S., Wiegele, A.: QPLIB: A library of quadratic programming instances. *Mathematical Programming Computation* **11**(2) (2019). DOI 10.1007/s12532-018-0147-4
2. Frank, M., Wolfe, P.: An algorithm for quadratic programming. *Naval research logistics quarterly* **3**(1-2), 95–110 (1956). DOI 10.1002/nav.3800030109
3. Nesterov, Y.: *Lectures on Convex Optimization*, vol. 137. Springer (2018)
4. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. *IMA journal of numerical analysis* **8**(1), 141–148 (1988). DOI 10.1093/imanum/8.1.141
5. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge university press (2004)
6. Josephy, N.H.: *Newton’s Method for Generalized Equations*. Tech. rep., Wisconsin Univ-Madison Mathematics Research Center (1979)
7. Bertsekas, D.P.: *Nonlinear Programming*. Athena scientific Belmont (1999)
8. Shewchuk, J.R.: *An Introduction to the Conjugate Gradient Method without the Agonizing Pain*. Carnegie-Mellon University. Department of Computer Science (1994)
9. Alessio, A., Bemporad, A.: A survey on explicit model predictive control. In: *Nonlinear Model Predictive Control*, pp. 345–369. Springer (2009)
10. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. *Automatica* **38**(1), 3–20 (2002). DOI 10.1016/S0005-1098(01)00174-1
11. Tøndel, P., Johansen, T.A., Bemporad, A.: An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica* **39**(3), 489–497 (2003)

12. Tondel, P., Johansen, T.A., Bemporad, A.: Further results on multiparametric quadratic programming. In: 42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475), vol. 3, pp. 3173–3178. IEEE (2003). DOI 10.1109/CDC.2003.1273111
13. Gupta, A., Bhartiya, S., Nataraj, P.S.V.: A novel approach to multiparametric quadratic programming. *Automatica* **47**(9), 2112–2117 (2011). DOI 10.1016/j.automatica.2011.06.019
14. Patrinos, P., Sarimveis, H.: A new algorithm for solving convex parametric quadratic programs based on graphical derivatives of solution mappings. *Automatica* **46**(9), 1405–1418 (2010). DOI 10.1016/j.automatica.2010.06.008
15. Wang, Y., Boyd, S.: Fast model predictive control using online optimization. *IEEE Transactions on control systems technology* **18**(2), 267–278 (2009). DOI 10.1109/TCST.2009.2017934
16. Mayne, D.Q.: Model predictive control: Recent developments and future promise. *Automatica* **50**(12), 2967–2986 (2014). DOI 10.1016/j.automatica.2014.10.128
17. Bemporad, A., Filippi, C.: Suboptimal explicit rhc via approximate multiparametric quadratic programming. Technical Report ETH Zurich, AUT02-07 (2002)
18. Johansen, T.A., Grancharova, A.: Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Transactions on Automatic Control* **48**(5), 810–815 (2003). DOI 10.1109/TAC.2003.811259
19. Dai, X., Bourdais, R., Guéguen, H.: Dynamic Reduction of the Iterations Requirement in a Distributed Model Predictive Control. In: 2019 IEEE 58th Conference on Decision and Control (CDC), pp. 6392–6397. IEEE (2019). DOI 10.1109/CDC40024.2019.9029783
20. Nocedal, J., Wright, S.J.: NUMERICAL OPTIMIZATION. Springer (2006)
21. Mitze, R., Mönnigmann, M.: A dynamic programming approach to solving constrained linear–quadratic optimal control problems. *Automatica* **120**, 109,132 (2020). DOI 10.1016/j.automatica.2020.109132
22. Magnus, J.R., Neudecker, H.: Matrix Differential Calculus with Applications in Statistics and Econometrics. John Wiley & Sons (2019)

-
23. Nesterov, Y.E.: A method for solving the convex programming problem with convergence rate $O(1/k^2)$. In: Dokl. Akad. Nauk Sssr, vol. 269, pp. 543–547 (1983)
 24. Giselsson, P., Doan, M.D., Keviczky, T., De Schutter, B., Rantzer, A.: Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica* **49**(3), 829–833 (2013). DOI 10.1016/j.automatica.2013.01.009
 25. Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., Diehl, M.: qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation* **6**(4), 327–363 (2014). DOI 10.1007/s12532-014-0071-1