How the augmented Lagrangian algorithm can deal with an infeasible convex quadratic optimization problem

J.Ch. Gilbert (INRIA Paris-Rocquencourt)
Joint work with
A. Chiche (INRIA, EDF, Paris VI)

July 4, 2010
Outline

The AL algorithm for a feasible convex QP

Problem structure

Convergence when the QP is infeasible

Consequences on the AL algorithm

Future works
The AL algorithm for a feasible convex QP

Problem structure

Convergence when the QP is infeasible

Consequences on the AL algorithm

Future works
The AL algorithm for a feasible convex QP

The QP to solve

- The problem to solve

\[
\begin{cases}
\inf_x q(x) \\
I \leq Ax \leq u,
\end{cases}
\]  

(1)

where

- \(q : x \in \mathbb{R}^n \mapsto q(x) = g^T x + \frac{1}{2} x^T H x\),
- \(g \in \mathbb{R}^n\) and \(H \succeq 0\) (encompasses linear optimization),
- \(A\) is \(m \times n\),
- \(l, u \in \mathbb{R}^m\) satisfy \(l < u\).

- What can do the AL algorithm when the QP is infeasible?
  - Detecting infeasibility rapidly.
  - Obtaining the solution to the closest feasible QP.
  - Finding a direction for an SQP nonlinear optimization solver.
The AL algorithm for a feasible convex QP

The QP to solve

- The problem to solve

\[
\begin{align*}
\inf_x q(x) \\
l &\leq Ax \leq u,
\end{align*}
\]

where

- \( q : x \in \mathbb{R}^n \mapsto q(x) = g^\top x + \frac{1}{2}x^\top Hx, \)
- \( g \in \mathbb{R}^n \) and \( H \succeq 0 \) (encompasses linear optimization),
- \( A \) is \( m \times n, \)
- \( l, u \in \mathbb{R}^m \) satisfy \( l < u. \)

- What can do the AL algorithm when the QP is infeasible?
  - Detecting infeasibility rapidly.
  - Obtaining the solution to the closest feasible QP.
  - Finding a direction for an SQP nonlinear optimization solver.
Towards the AL algorithm

• The problem is transformed by using an auxiliary variable $y$:

\[
\begin{align*}
\inf_x q(x) \\
l \leq Ax \leq u
\end{align*}
\rightsquigarrow
\begin{align*}
\inf_x q(x) \\
Ax = y \\
l \leq y \leq u.
\end{align*}
\]

• Equality constraints penalized by the augmented Lagrangian

\[
\ell_r(x, y, \lambda) := q(x) + \lambda^T(Ax - y) + \frac{r}{2}\|Ax - y\|^2.
\]

• At each iteration the AL algorithm solves

\[
\inf_{(x, y) \in \mathbb{R}^n \times [l, u]} \ell_r(x, y, \lambda).
\]

The minimization in $y$ could be computed analytically → AL of Rockafellar [13, 14], piecewise quadratic function.
Towards the AL algorithm

• The problem is transformed by using an auxiliary variable $y$:

$$\inf_x q(x) \quad \begin{array}{c} \text{l} \leq Ax \leq u \\ \end{array} \sim \inf_x q(x) \quad \begin{array}{c} Ax = y \\ \text{l} \leq y \leq u. \end{array}$$

• Equality constraints penalized by the augmented Lagrangian

$$\ell_r(x, y, \lambda) := q(x) + \lambda^\top(Ax - y) + \frac{r}{2}\|Ax - y\|^2.$$

• At each iteration the AL algorithm solves

$$\inf_{(x,y) \in \mathbb{R}^n \times [l,u]} \ell_r(x, y, \lambda).$$

The minimization in $y$ could be computed analytically
→ AL of Rockafellar [13, 14], piecewise quadratic function.
Towards the AL algorithm

- The problem is transformed by using an auxiliary variable $y$:

$$\begin{align*}
\inf_x q(x) \\
\quad \quad \quad \quad \quad \quad \quad l \leq Ax \leq u
\end{align*}$$

$$\sim$$

$$\begin{align*}
\inf_x q(x) \\
Ax = y \\
\quad \quad \quad \quad \quad \quad \quad l \leq y \leq u.
\end{align*}$$

- Equality constraints penalized by the augmented Lagrangian

$$\ell_r(x, y, \lambda) := q(x) + \lambda^\top (Ax - y) + \frac{r}{2} \|Ax - y\|^2.$$ 

- At each iteration the AL algorithm solves

$$\inf_{(x,y) \in \mathbb{R}^n \times [l,u]} \ell_r(x, y, \lambda).$$

The minimization in $y$ could be computed analytically

$\rightarrow$ AL of Rockafellar [13, 14], piecewise quadratic function.
The AL algorithm for feasible QPs \([11, 12, 13, 5, 1, 15, 16]\)

Set \(\lambda_0 \in \mathbb{R}^m, r_0 > 0, \rho_{\text{des}} \in ]0, 1[,\) and repeat for \(k = 0, 1, 2, \ldots\)

- Compute (if possible, exit otherwise)

\[
(x_{k+1}, y_{k+1}) \in \arg \min_{(x,y) \in \mathbb{R}^n \times [l,u]} \ell_r(x,y, \lambda_k).
\]

- Update the multipliers

\[
\lambda_{k+1} = \lambda_k + r_k(A x_{k+1} - y_{k+1}).
\]

- Stop if

\[
A x_{k+1} - y_{k+1} \simeq 0.
\]

- Update \(r_k \leadsto r_{k+1} > 0\): \(s_k := y_k - A x_k,\)

\[
L_{\text{inf},k} := \max_{1 \leq i \leq k} \left( r_i \frac{\|s_{i+1}\|}{\|s_i\|} \right) \quad \text{and} \quad r_{k+1} \geq \frac{L_{\text{inf},k}}{\rho_{\text{des}}}.
\]
Comparison with other algorithms

- **No need of factorization** → useful for large scale problems
- Can take advantage of an a priori knowledge of active constraints → useful for SQP
- **Comparison with interior point methods**
  - the algorithm must deal with bound activity at each iteration, while $Ax = y$ is satisfied asymptotically
  - the linear systems to solve are not ill-conditioned
  - no polynomiality result
- **Comparison with active set methods**
  - In case of strict complementarity, the active set is determined after a finite number of iterations
  - simple bounds allows the algorithm to use the gradient projections to change many active constraints at each iteration
  - can deal with an infeasible QP without elastic variables
Comparison with other algorithms

- **No need of factorization** → useful for large scale problems
- **Can take advantage of an a priori knowledge of active constraints** → useful for SQP

- **Comparison with interior point methods**
  - the algorithm must deal with bound activity at each iteration, while $Ax = y$ is satisfied asymptotically
  - the linear systems to solve are not ill-conditioned
  - no polynomiality result

- **Comparison with active set methods**
  - In case of strict complementarity, the active set is determined after a finite number of iterations
  - simple bounds allows the algorithm to use the gradient projections to change many active constraints at each iteration
  - can deal with an infeasible QP without elastic variables
Comparison with other algorithms

- **No need of factorization** → useful for large scale problems
- **Can take advantage of an a priori knowledge of active constraints** → useful for SQP
- **Comparison with interior point methods**
  - the algorithm must deal with bound activity at each iteration, while $Ax = y$ is satisfied asymptotically
  - the linear systems to solve are not ill-conditioned
  - no polynomiality result
- **Comparison with active set methods**
  - In case of strict complementarity, the active set is determined after a finite number of iterations
  - simple bounds allows the algorithm to use the gradient projections to change many active constraints at each iteration
  - can deal with an infeasible QP without elastic variables
Comparison with other algorithms

- **No need of factorization** → useful for large scale problems
- **Can take advantage of an a priori knowledge of active constraints** → useful for SQP
- **Comparison with interior point methods**
  - the algorithm must deal with bound activity at each iteration, while $Ax = y$ is satisfied asymptotically
  - the linear systems to solve are not ill-conditioned
  - no polynomiality result
- **Comparison with active set methods**
  - In case of strict complementarity, the active set is determined after a finite number of iterations
  - simple bounds allows the algorithm to use the gradient projections to change many active constraints at each iteration
  - can deal with an infeasible QP without elastic variables
Global linear convergence for QPs with a solution [7]

QP with solution ⇒ the dual solution set $\mathcal{S}_D \neq \emptyset$ and

\[
\forall \beta > 0, \; \exists L > 0, \; \text{dist}(\lambda_0, \mathcal{S}_D) \leq \beta \; \text{implies that} \; \\
\forall k \geq 1, \; \|Ax_{k+1} - y_{k+1}\| \leq \min \left( \frac{L}{r_k}, 1 \right) \|Ax_k - y_k\|.
\]

This property can by used in QP solvers (e.g., Qpal [8], QPlab)

- Global property → $\forall k \geq 1$, increase $r_k$ if $\frac{\|Ax_{k+1} - y_{k+1}\|}{\|Ax_k - y_k\|} > \rho_{\text{des}}$.
- For infeasible QP
  - the rule $\Rightarrow r_k \nearrow \infty$,
  - the AL subproblems become ill-conditioned,
  - difficult to decide to stop when $r_k \geq$ a universal threshold.

Main goal: extending the above estimate to infeasible QPs.
The AL algorithm for a feasible convex QP

Problem structure

Convergence when the QP is infeasible

Consequences on the AL algorithm

Future works
Problem structure

The smallest feasible shift

- It is always possible to find a shift \( s \in \mathbb{R}^m \) such that
  \[
  l \leq Ax + s \leq u \text{ is feasible for } x \in \mathbb{R}^n.
  \]

- These feasible shifts are exactly those in \( S := [l, u] + \mathcal{R}(A) \):

- The smallest feasible shift \( \bar{s} := \arg \min \{ \| s \| : s \in S \} \).
Problem structure

The smallest feasible shift

- It is always possible to find a shift $s \in \mathbb{R}^m$ such that
  $$l \leq Ax + s \leq u$$
  is feasible for $x \in \mathbb{R}^n$.

- These feasible shifts are exactly those in $\mathcal{S} := [l, u] + \mathcal{R}(A)$:

$$\mathcal{S} := [l, u] + \mathcal{R}(A)$$

- The smallest feasible shift $\bar{s} := \arg \min \{\|s\| : s \in \mathcal{S}\}$. 
**Problem structure**

The smallest feasible shift

- It is always possible to find a shift $s \in \mathbb{R}^m$ such that
  
  $$l \leq Ax + s \leq u$$
  is feasible for $x \in \mathbb{R}^n$.

- These feasible shifts are exactly those in $S := [l, u] + \mathcal{R}(A)$:

- The smallest feasible shift $\bar{s} := \arg \min \{\|s\| : s \in S\}$. 
The dual function

- The dual function $\delta : \mathbb{R}^m \to \overline{\mathbb{R}}$, defined at $\lambda \in \mathbb{R}^m$ by
  \[
  \delta(\lambda) := -\inf_{(x,y) \in \mathbb{R}^n \times [l,u]} \left( q(x) + \lambda^T(Ax - y) \right).
  \]

  - $\delta$ is convex, closed, and $\delta > -\infty$.
  - $\text{dom } \delta \neq \emptyset \iff \delta \in \text{Conv}(\mathbb{R}^m)$.

- Important function for
  - specifying when the AL algorithm is well defined,
  - giving the proximal interpretation of the AL algorithm,
  - characterizing the QP feasibility.

Let us look at that.
The dual function

- The dual function $\delta : \mathbb{R}^m \rightarrow \mathbb{R}$, defined at $\lambda \in \mathbb{R}^m$ by

$$
\delta(\lambda) := -\inf_{(x,y) \in \mathbb{R}^n \times [l,u]} \left( q(x) + \lambda^T (Ax - y) \right).
$$

- $\delta$ is convex, closed, and $\delta > -\infty$.
- $\text{dom } \delta \neq \emptyset \iff \delta \in \text{Conv}(\mathbb{R}^m)$.

- Important function for
  - specifying when the AL algorithm is well defined,
  - giving the proximal interpretation of the AL algorithm,
  - characterizing the QP feasibility.

Let us look at that.
When is the AL algorithm well defined?

- The following properties are equivalent:
  \( (i) \) \( \text{dom} \delta \neq \emptyset \quad (\iff \delta \in \text{Conv}(\mathbb{R}^m)) \),
  \( (ii) \) for some (or any) \( s \in S \), the shifted QP has a solution,
  \( (iii) \) for some (or any) \( r_k > 0 \) and \( \lambda_k \in \mathbb{R}^m \), the AL subproblem has a solution.

- The failure of these conditions can be detected at the first iteration of the AL algorithm: the first AL subproblem is unbounded and can find a direction \( d \in \mathbb{R}^n \) such that

\[
g^\top d < 0, \quad Hd = 0, \quad \text{and} \quad Ad \in [l, u]^{\infty}.
\]
When is the AL algorithm well defined?

- The following properties are equivalent:
  1. \( \text{dom} \, \delta \neq \emptyset \) \( \iff \delta \in \text{Conv}(\mathbb{R}^m) \),
  2. for some (or any) \( s \in S \), the shifted QP has a solution,
  3. for some (or any) \( r_k > 0 \) and \( \lambda_k \in \mathbb{R}^m \), the AL subproblem has a solution.

- The failure of these conditions can be detected at the first iteration of the AL algorithm: the first AL subproblem is unbounded and can find a direction \( d \in \mathbb{R}^n \) such that

\[
\begin{align*}
g^\top d &< 0, \\
Hd &\equiv 0, \quad \text{and} \\
Ad &\in [l, u]^\infty.
\end{align*}
\]
AL algorithm = proximal algorithm on $\delta$ [14]

- If $\delta \in \overline{\text{Conv}}(\mathbb{R}^m)$ and $r > 0$, then

\[ -\inf_{(x,y) \in \mathbb{R}^n \times [l,u]} \ell_{r_k}(x,y,\lambda_k) = \inf_{\lambda \in \mathbb{R}^m} \left( \delta(\lambda) + \frac{1}{2r_k} \|\lambda - \lambda_k\|^2 \right). \]

- Any solution $(x_{k+1}, y_{k+1})$ to the problem in the LHS and the unique solution $\lambda_{k+1}$ to the problem in the RHS are linked by

\[
\begin{align*}
\lambda_{k+1} &= \lambda_k + r_k(Ax_{k+1} - y_{k+1}) \\
y_{k+1} - Ax_{k+1} &\in \partial\delta(\lambda_{k+1}).
\end{align*}
\]

Hence $\lambda_{k+1} = \text{prox}_{\delta,r_k}(\lambda_k)$.

- Useful for the analysis and to clarify the algorithm.
AL algorithm = proximal algorithm on $\delta$ [14]

- If $\delta \in \text{Conv}(\mathbb{R}^m)$ and $r > 0$, then
\[
-\inf_{(x,y)\in \mathbb{R}^n \times [l,u]} \ell_r(x, y, \lambda_k) = \inf_{\lambda \in \mathbb{R}^m} \left( \delta(\lambda) + \frac{1}{2r} \|\lambda - \lambda_k\|^2 \right).
\]

- Any solution $(x_{k+1}, y_{k+1})$ to the problem in the LHS and the unique solution $\lambda_{k+1}$ to the problem in the RHS are linked by
\[
\begin{cases}
\lambda_{k+1} = \lambda_k + r_k(Ax_{k+1} - y_{k+1}) \\
y_{k+1} - Ax_{k+1} \in \partial \delta(\lambda_{k+1}).
\end{cases}
\]

Hence $\lambda_{k+1} = \text{prox}_{\delta, r_k}(\lambda_k)$.

- Useful for the analysis and to clarify the algorithm.
AL algorithm = proximal algorithm on $\delta$ [14]

- If $\delta \in \overline{\text{Conv}(\mathbb{R}^{m})}$ and $r > 0$, then

$$- \inf_{(x, y) \in \mathbb{R}^{n} \times [l, u]} \ell_{r_{k}}(x, y, \lambda_{k}) = \inf_{\lambda \in \mathbb{R}^{m}} \left( \delta(\lambda) + \frac{1}{2r_{k}} \|\lambda - \lambda_{k}\|^{2} \right).$$

- Any solution $(x_{k+1}, y_{k+1})$ to the problem in the LHS and the unique solution $\lambda_{k+1}$ to the problem in the RHS are linked by

$$\begin{cases}
\lambda_{k+1} = \lambda_{k} + r_{k}(Ax_{k+1} - y_{k+1}) \\
y_{k+1} - Ax_{k+1} \in \partial \delta(\lambda_{k+1}).
\end{cases}$$

Hence $\lambda_{k+1} = \text{prox}_{\delta, r_{k}}(\lambda_{k})$.

- Useful for the analysis and to clarify the algorithm.
AL iterates minimizing the dual function for a feasible QP

- $\delta$ is piecewise quadratic
  
  $$\delta(\lambda) = \frac{1}{2} \lambda^T S \lambda + (v + y_\lambda)^T \lambda + C^\text{st}$$

- $S_D := \arg \min \delta$

- $\partial \delta(\lambda_{k+1})$ contains
  
  $$\frac{\lambda_k - \lambda_{k+1}}{r_k} = y_{k+1} - Ax_{k+1}$$

- small $r_k$'s in the figure
Feasibility and dual function

• If $\text{dom} \delta \neq \emptyset$,

the QP is feasible $\iff \delta$ is bounded below.

• Consequence:

infeasible QP $\implies \{\lambda_k\}$ blows up.
Feasibility and dual function

- If \( \text{dom} \, \delta \neq \emptyset \),
  
  the QP is feasible \( \iff \) \( \delta \) is bounded below.

- Consequence:
  
  infeasible QP \( \implies \) \( \{ \lambda_k \} \) blows up.
Level curves of the dual function $\delta$ (infeasible QP, $H \succ 0$)
Level curves of the dual function $\delta$ (infeasible QP, $H = 0$)
Outline

The AL algorithm for a feasible convex QP

Problem structure

Convergence when the QP is infeasible

Consequences on the AL algorithm

Future works
Convergence when the QP is infeasible

- A crucial (and surprising) observation

  feasible shifts \( \rightarrow S = \mathcal{R}(\partial \delta) \) \( \leftarrow \) subgradients of \( \delta \).

  - Value function \( v(s) := \inf \{ q(x) : Ax + s \in [l, u], x \in \mathbb{R}^n \} \).
  - Clearly \( \text{dom } v = S \).
  - Then \( \delta = v^* \) (always) and \( \delta^* = v \) (for a convex QP).
  - Next \( \mathcal{R}(\partial \delta) = \overline{\mathcal{R}(\partial \delta)} = \overline{\text{dom } \delta^*} = \overline{\text{dom } v} = \overline{S} = S \).

- This identity explains the consequence \( y_k - Ax_k \rightarrow \bar{s} \) [17].

  - \( \bar{s} := \arg \min \{ ||s|| : s \in S \} \) is also the smallest subgradient of \( \delta \).
  - \( \bar{s} \in \partial \delta(\lambda) \implies \lambda \) is on the “bottom of the slide” \( \tilde{S}_D \) of \( \delta \).
  - Intuitively \( \text{dist}(\lambda_k, \tilde{S}_D) \rightarrow 0 \), hence \( y_k - Ax_k \in \partial \delta(\lambda_k) \rightarrow \bar{s} \).
Convergence when the QP is infeasible

- A crucial (and surprising) observation

  feasible shifts $\rightarrow S = \mathcal{R}(\partial \delta)$ $\leftarrow$ subgradients of $\delta$.

  - Value function $v(s) := \inf \{ q(x) : Ax + s \in [l, u], \ x \in \mathbb{R}^n \}$.
  - Clearly $\text{dom} \ v = S$.
  - Then $\delta = v^*$ (always) and $\delta^* = v$ (for a convex QP).
  - Next $\mathcal{R}(\partial \delta) = \overline{\mathcal{R}(\partial \delta)} = \overline{\text{dom} \delta^*} = \overline{\text{dom} v} = \overline{S} = S$.

- This identity explains the consequence $y_k - Ax_k \rightarrow \bar{s}$ [17].

  - $\bar{s} := \arg \min \{ \| s \| : s \in S \}$ is also the smallest subgradient of $\delta$.
  - $\bar{s} \in \partial \delta(\lambda) \quad \Rightarrow \quad \lambda$ is on the “bottom of the slide” $\tilde{S}_D$ of $\delta$.
  - Intuitively $\text{dist}(\lambda_k, \tilde{S}_D) \rightarrow 0$, hence $y_k - Ax_k \in \partial \delta(\lambda_k) \rightarrow \bar{s}$.
Proof of the convergence $s_k := y_k - Ax_k \to \bar{s}$ on a figure

The slope at $\lambda_k$ is $y_k - Ax_k$

The slope at $P_{\tilde{S}_D}\lambda_k$ is $\bar{s}$

$\text{dist}(\lambda_k, \tilde{S}_D) \to 0 \quad \text{“}\implies\text{”} \quad s_k \to \bar{s}$
Global linear convergence of \( s_k := y_k - Ax_k \to \bar{s} \)

\[
\forall \beta > 0, \ \exists L > 0, \ \text{dist}(\lambda_0, \tilde{S}_D) \leq \beta \implies \forall k \geq 1, \ \|s_{k+1} - \bar{s}\| \leq \min \left( \frac{L}{r_k}, 1^? \right) \|s_k - \bar{s}\|,
\]

where \( \tilde{S}_D \) is the “bottom of the slide”:

\[
\tilde{S}_D := \{ \lambda \in \mathbb{R}^m : \bar{s} \in \partial \delta(\lambda) \}.
\]

Comments:

- the inequality is valid from the 1st iteration \( \rightarrow \text{global} \),
- \( L \) is not known \( \rightarrow \) iterative monitoring of \( r_k \),
- \( \bar{s} \) is not known, but there is a way to go around that difficulty.
Global linear convergence of $s_k := y_k - Ax_k \to \bar{s}$

\[
\forall \beta > 0, \quad \exists L > 0, \quad \text{dist}(\lambda_0, \tilde{S}_D) \leq \beta \quad \text{implies that}
\]

\[
\forall k \geq 1, \quad \|s_{k+1} - \bar{s}\| \leq \min \left( \frac{L}{r_k}, 1? \right) \|s_k - \bar{s}\|,
\]

where $\tilde{S}_D$ is the “bottom of the slide”:

\[
\tilde{S}_D := \{\lambda \in \mathbb{R}^m : \bar{s} \in \partial \delta(\lambda)\}.
\]

Comments:

- the inequality is valid from the 1st iteration $\to$ global,
- $L$ is not known $\to$ iterative monitoring of $r_k$,
- $\bar{s}$ is not known, but there is a way to go around that difficulty.
The AL algorithm for a feasible convex QP problem structure

Convergence when the QP is infeasible

Consequences on the AL algorithm

\[ \rho \]

Augmentation parameter \( r \)

- Maximal quotient \( \rho \)
- Straight of constant value 1
- Straight of slope 1
Where does this global linear convergence come from? (I)

- Closest feasible problem
  \[
  \begin{cases}
  \inf_x q(x) \\
  l \leq Ax + \bar{s} \leq u.
  \end{cases}
  \]
- Its dual function
  \[
  \tilde{\delta}(\lambda) = \delta(\lambda) - \bar{s}^T \lambda.
  \]
- Dual solution set
  \[
  \tilde{S}_D = \partial \tilde{\delta}^{-1}(0).
  \]
- The (easy) property
  \[
  \lambda_{k+1} = \text{prox}_{\tilde{\delta}, r_k} (\lambda_k - r_k \bar{s})
  \]
  \[
  + \text{monotonicity of } \partial \tilde{\delta}(\cdot) \Rightarrow 
  \]
  \[
  \|s_{k+1} - \bar{s}\| \leq \frac{1}{r_k} \text{dist}(\tilde{\lambda}_k, \tilde{S}_D).
  \]
Where does this global linear convergence come from? (II)

The $\text{dist}(\tilde{\lambda}_k, \tilde{S}_D)$ can be estimated, thanks to the following quasi-global error bound [7] applied to the dual function $\tilde{\delta}$ of the closest feasible problem:

For any bounded set $\tilde{B} \subset \mathbb{R}^m$, there is an $L > 0$, such that

$$\forall \tilde{\lambda} \in \tilde{S}_D + \tilde{B}, \quad \forall \tilde{s} \in \partial \tilde{\delta}(\tilde{\lambda}) : \quad \text{dist}(\tilde{\lambda}, \tilde{S}_D) \leq L \|\tilde{s}\|.$$
Outline

The AL algorithm for a feasible convex QP

Problem structure

Convergence when the QP is infeasible

Consequences on the AL algorithm

Future works
Consequences on the AL algorithm

Detecting optimality of the closest feasible QP without $\bar{s}$

- Optimality conditions of the closest feasible problem:

\[
\begin{align*}
(\bar{x}, \bar{y}) & \text{ is a solution to } \\
\inf_{(x,y)} q(x) & \iff \\
Ax + \bar{s} = y & \forall \lambda \in \mathbb{R}^m: \\
l \leq y \leq u & \\
A^T(A\bar{x} - \bar{y}) = 0 & \\
P_{[l,u]}(A\bar{x}) = \bar{y}.
\end{align*}
\]

- In the AL algorithm, there hold

\[
A^T(Ax_k - y_k) \to 0 \quad \text{ and } \quad P_{[l,u]}(Ax_k) - y_k \to 0.
\]
Consequences on the AL algorithm

Detecting optimality of the closest feasible QP without \( \bar{s} \)

- Optimality conditions of the closest feasible problem:

\[
\begin{align*}
(\bar{x}, \bar{y}) \text{ is a solution to} & \\
\inf_{(x,y)} q(x) & \\
A x + \bar{s} = y & \\
l \leq y \leq u & \\
\Leftrightarrow & \\
\text{for some } \bar{\lambda} \in \mathbb{R}^m: & \\
(\bar{x}, \bar{y}) \in \arg \min_{x, y \in [l,u]} \ell_r(x, y, \bar{\lambda}) & \\
A^\top (A \bar{x} - \bar{y}) = 0 & \\
P_{[l,u]}(A \bar{x}) = \bar{y}. & 
\end{align*}
\]

- In the AL algorithm, there hold

\[A^\top (A x_k - y_k) \to 0 \quad \text{and} \quad P_{[l,u]}(A x_k) - y_k \to 0.\]
Monitoring the global linear convergence without $\bar{s}$

- Global linear convergence of $s_k := y_k - Ax_k \to \bar{s}$ at rate $\tau \in [0, 1]$. But $\bar{s}$ is unknown!

- The sequence $s'_k := s_{k+1} - s_k$ also converges globally linearly to 0 at the rate
  \[ \tau' := \frac{(1 + \tau)\tau}{1 - \tau}, \]
  which is $< 1$ if $\tau < \sqrt{2} - 1 \simeq 0.41$.

- Conclusion: monitor the rate of convergence, by observing $s'_k$ instead of $s_k$. 
Monitoring the global linear convergence without $\bar{s}$

- Global linear convergence of $s_k := y_k - Ax_k \rightarrow \bar{s}$ at rate $\tau \in [0, 1]$. But $\bar{s}$ is unknown!

- The sequence $s'_k := s_{k+1} - s_k$ also converges globally linearly to 0 at the rate
  \[ \tau' := \frac{(1 + \tau)\tau}{1 - \tau}, \]
  which is $< 1$ if $\tau < \sqrt{2} - 1 \approx 0.41$.

- Conclusion: monitor the rate of convergence, by observing $s'_k$ instead of $s_k$. 
Monitoring the global linear convergence without $\bar{s}$

- Global linear convergence of $s_k := y_k - Ax_k \to \bar{s}$ at rate $\tau \in [0, 1]$. But $\bar{s}$ is unknown!

- The sequence $s'_k := s_{k+1} - s_k$ also converges globally linearly to 0 at the rate
  
  $$\tau' := \frac{(1 + \tau)\tau}{1 - \tau},$$

  which is < 1 if $\tau < \sqrt{2} - 1 \approx 0.41$.

- Conclusion: monitor the rate of convergence, by observing $s'_k$ instead of $s_k$.  

Revisiting the AL algorithm

Set $\lambda_0 \in \mathbb{R}^m$, $r_0 > 0$, $\rho_{\text{des}} \in ]0,1[,$ and repeat for $k = 0, 1, 2, \ldots$

- Compute (if possible, exit otherwise)
  
  $$(x_{k+1}, y_{k+1}) \in \arg\min_{(x,y)\in\mathbb{R}^n \times [l,u]} \ell_r(x, y, \lambda_k).$$

- Multiplier update
  
  $$\lambda_{k+1} = \lambda_k + r_k(Ax_{k+1} - y_{k+1}).$$

- Stop if
  
  $$A^T(Ax_{k+1} - y_{k+1}) \simeq 0 \quad \text{and} \quad P_{[l,u]}(Ax_{k+1}) - y_{k+1} \simeq 0.$$ 

- Update $r_k \leadsto r_{k+1} > 0$: $s_k := y_k - Ax_k$, $s'_k := s_{k+1} - s_k$,

  $$L_{\text{inf},k} := \max_{1 \leq i \leq k} \left( r_i \frac{\|s'_{i+1}\|}{\|s'_i\|} \right) \quad \text{and} \quad r_{k+1} \geq \frac{L_{\text{inf},k}}{\rho_{\text{des}}}. $$
Using of the AL algorithm within the SQP algorithm

The SQP algorithm solves the nonlinear optimization problem

\[
\begin{align*}
\inf_x & \quad f(x) \\
\text{s.t.} & \quad l \leq c(x) \leq u,
\end{align*}
\]

by solving a sequence of QPs of the form

\[
\begin{align*}
\inf_d & \quad g^\top d + \frac{1}{2} d^\top H d \\
\text{s.t.} & \quad l' \leq Ad \leq u',
\end{align*}
\]

where \( g = \nabla f(x) \), \( 0 \preceq H \preceq \nabla^2_{xx} \ell(x, \lambda) \), \( l' := l - c(x) \), \( u' := u - c(x) \), and \( A := c'(x) \).
Suppose that the AL algorithm is used to solve the QP.

- **Two cases can occur:**
  1. the first AL subproblem has no solution and finds a $d$ s.t.
     \[ g^T d < 0, \quad H d = 0, \quad \text{and} \quad A d \in [l, u]^{-\infty}, \]
  2. the algorithm solves the closest feasible QP
     \[
     \left\{ \begin{array}{l}
     \inf_d \ g^T d + \frac{1}{2} d^T H d \\
     l' \leq A d + \bar{s} \leq u'.
     \end{array} \right.
     \]

- In both cases, $d$ is a descent direction of the merit function
  \[ \Theta_{\sigma}(x) = f(x) + \sigma \ dist(c(x), [l, u]), \]
  if $\sigma > 0$ is large enough (Burke and Han for case 2 [4, 2, 3]).
- By this strategy, there is no need of elastic variables [9, 10].
Suppose that the AL algorithm is used to solve the QP.

- Two cases can occur:
  1. the first AL subproblem has no solution and finds a $d$ s.t.
     \[ g^T d < 0, \quad H d = 0, \quad \text{and} \quad A d \in [l, u]^\infty, \]
  2. the algorithm solves the closest feasible QP
     \[
     \begin{aligned}
     \inf_d & \quad g^T d + \frac{1}{2} d^T H d \\
     l' & \leq A d + \bar{s} \leq u'.
     \end{aligned}
     \]

- In both cases, $d$ is a descent direction of the merit function
  \[ \Theta_\sigma(x) = f(x) + \sigma \ \text{dist}(c(x), [l, u]), \]
  if $\sigma > 0$ is large enough (Burke and Han for case 2 [4, 2, 3]).

- By this strategy, there is no need of elastic variables [9, 10].
Suppose that the AL algorithm is used to solve the QP.

- **Two cases can occur:**
  1. the first AL subproblem has no solution and finds a \( d \) s.t.
     \[
     g^\top d < 0, \quad Hd = 0, \quad \text{and} \quad Ad \in [l, u]^{\infty},
     \]
  2. the algorithm solves the closest feasible QP
     \[
     \inf_d \left\{ g^\top d + \frac{1}{2} d^\top Hd \right\} : \quad l' \leq Ad + \bar{s} \leq u'.
     \]

- In both cases, \( d \) is a descent direction of the merit function
  \[
  \Theta_\sigma(x) = f(x) + \sigma \, \text{dist}(c(x), [l, u]),
  \]
  if \( \sigma > 0 \) is large enough (Burke and Han for case 2 [4, 2, 3]).
- By this strategy, there is no need of elastic variables [9, 10].
Outline

The AL algorithm for a feasible convex QP

Problem structure

Convergence when the QP is infeasible

Consequences on the AL algorithm

Future works
Future works

See whether the global linear convergence result of the AL algorithm can be extended to the following cases.

- Adding a trust region or more generally

\[
\inf_x \langle g, x \rangle + \frac{1}{2} \langle x, Hx \rangle \\
A x \in C \\
x \in X,
\]

where \( C \) and \( X \) are polyhedral sets (not possible otherwise? in particular for an SDP problem?). Useful for the globalization of the SQP algorithm.

- Inexact solution of the AL subproblems.
Future works

See whether the global linear convergence result of the AL algorithm can be extended to the following cases.

- Adding a trust region or more generally

\[
\inf_x \langle g, x \rangle + \frac{1}{2} \langle x, Hx \rangle \\
Ax \in C \\
x \in X,
\]

where $C$ and $X$ are polyhedral sets (not possible otherwise? in particular for an SDP problem?). Useful for the globalization of the SQP algorithm.

- Inexact solution of the AL subproblems.
Future works

See whether the global linear convergence result of the AL algorithm can be extended to the following cases.

- **Adding a trust region** or more generally

\[
\inf_x \langle g, x \rangle + \frac{1}{2} \langle x, Hx \rangle
\]

\[
A x \in C
\]

\[
x \in X,
\]

where \(C\) and \(X\) are polyhedral sets (not possible otherwise? in particular for an SDP problem?). Useful for the globalization of the SQP algorithm.

- **Inexact solution** of the AL subproblems.
The AL algorithm for a feasible convex QP problem structure convergence when the QP is infeasible. Consequences on the AL algorithm.

A generalized saddle point result for constrained optimization. 
*Mathematical Programming*, 5, 225–234.

J.V. Burke (1989).
A sequential quadratic programming method for potentially infeasible mathematical programs. 

A robust trust region method for constrained nonlinear programming problems. 
*SIAM Journal on Optimization*, 2, 325–347.

J.V. Burke, S.-P. Han (1989).
A robust sequential quadratic programming method. 

J.D. Buys (1972).
*Dual algorithms for constrained optimization.* 

How the augmented Lagrangian algorithm can deal with an infeasible convex quadratic optimization problems. 
Technical report, INRIA, BP 105, 78153 Le Chesnay, France. 
To appear.

Global linear convergence of an augmented Lagrangian algorithm for solving convex quadratic optimization problems. 
*Journal of Convex Analysis*, 12, 45–69.

QPAL – A solver of convex quadratic optimization problems, using an augmented Lagrangian approach – Version 0.6.1. 
Rapport Technique 377, INRIA, BP 105, 78153 Le Chesnay, France.
The AL algorithm for a feasible convex QP problem structure convergence when the QP is infeasible. Consequences on the AL algorithm.


The AL algorithm for a feasible convex QP  Problem structure  Convergence when the QP is infeasible  Consequences on the AL alg

A projection method for least-squares solutions to overdetermined systems of linear inequalities.  
86, 211–236.