Quadratic Programming: Theory and Algorithms

Zola Donovan[1] and Murong Xu[1]

[1] Department of Mathematics
West Virginia University
Morgantown, WV USA

March 3, 2015

Outline

Outline

Outline

Outline

Outline

Outline

The Casino Game

The Casino Game

Example (1)

The Casino Game

Example (1)

- Suppose you are given the choice of playing one of two games at a casino.

The Casino Game

Example (1)

- Suppose you are given the choice of playing one of two games at a casino.
- Game X has a 5% chance of winning $1000, and a 95% chance of winning nothing.

The Casino Game

Example (1)

- Suppose you are given the choice of playing one of two games at a casino.
- Game X has a 5% chance of winning $1000, and a 95% chance of winning nothing.
- Game Y has a 5% chance of winning $5000.

The Casino Game

Example (1)

- Suppose you are given the choice of playing one of two games at a casino.
- Game X has a 5% chance of winning $1000, and a 95% chance of winning nothing.
- Game Y has a 5% chance of winning $5000. If you lose however, you have to pay the casino $200.

The Casino Game

Example (1)

- Suppose you are given the choice of playing one of two games at a casino.
- Game X has a 5% chance of winning $1000, and a 95% chance of winning nothing.
- Game Y has a 5% chance of winning $5000. If you lose however, you have to pay the casino $200.
- You are allowed to play this game one time.

The Casino Game

Example (1)

- Suppose you are given the choice of playing one of two games at a casino.
- Game X has a 5% chance of winning $1000, and a 95% chance of winning nothing.
- Game Y has a 5% chance of winning $5000. If you lose however, you have to pay the casino $200.
- You are allowed to play this game one time.
- Which game would you choose to play?

Portfolio Optimization

Portfolio Optimization

Example (2)

Portfolio Optimization

Example (2)

- We wish to invest $1000.00 in stocks A, B, and C for a one month period.

Portfolio Optimization

Example (2)

- We wish to invest $1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month,

Portfolio Optimization

Example (2)

- We wish to invest \$1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.

Portfolio Optimization

Example (2)

- We wish to invest \$1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.
- The rate of return of each stock is a random variable with some expected value.

Portfolio Optimization

Example (2)

- We wish to invest \$1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.
- The rate of return of each stock is a random variable with some expected value.
- Our goal is to invest in such a way that the expected end-of-month return is at least \$50.00 or 5%.

Optimization Approach

Optimization Approach

An optimization approach to the decision problems:

Optimization Approach

An optimization approach to the decision problems:

- Build a mathematical model of the decision problem.

Optimization Approach

An optimization approach to the decision problems:

- Build a mathematical model of the decision problem.
- Analyze available quantitative data to use in the mathematical model.

Optimization Approach

An optimization approach to the decision problems:

- Build a mathematical model of the decision problem.
- Analyze available quantitative data to use in the mathematical model.
- Use a numerical method to solve the mathematical model.

Optimization Approach

An optimization approach to the decision problems:

- Build a mathematical model of the decision problem.
- Analyze available quantitative data to use in the mathematical model.
- Use a numerical method to solve the mathematical model.
- Infer the actual decision from the solution to the mathematical model.

The Quadratic Programming Problem

The Quadratic Programming Problem

The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function,

The Quadratic Programming Problem

The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.

The Quadratic Programming Problem

The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.
- The general quadratic program can be stated as follows:

The Quadratic Programming Problem

The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.
- The general quadratic program can be stated as follows:

$$\text{Optimize} \quad \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} + \mathbf{c}^T \cdot \mathbf{x}$$

The Quadratic Programming Problem

The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.
- The general quadratic program can be stated as follows:

$$\begin{array}{ll} \text{Optimize} & \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} + \mathbf{c}^T \cdot \mathbf{x} \\ & \mathbf{A} \cdot \mathbf{x} \ \{\leq, =, or \geq\} \ \mathbf{b} \end{array}$$

The Quadratic Programming Problem

The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.
- The general quadratic program can be stated as follows:

$$\begin{aligned} \text{Optimize} \quad & \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} + \mathbf{c}^T \cdot \mathbf{x} \\ & \mathbf{A} \cdot \mathbf{x} \ \{\leq, =, \mathit{or} \geq\} \ \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

The Quadratic Programming Problem

The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.
- The general quadratic program can be stated as follows:

$$\begin{aligned} \text{Optimize} \quad & \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} + \mathbf{c}^T \cdot \mathbf{x} \\ & \mathbf{A} \cdot \mathbf{x} \; \{\leq, =, or \geq\} \; \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

- QPs are special classes of nonlinear optimization problems,

The Quadratic Programming Problem

The quadratic programming (QP) problem

- Quadratic programming (QP) refers to the problem of optimizing a quadratic function, subject to linear equality and inequality constraints.
- The general quadratic program can be stated as follows:

$$\begin{aligned} \text{Optimize} \quad & \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} + \mathbf{c}^T \cdot \mathbf{x} \\ & \mathbf{A} \cdot \mathbf{x} \ \{\leq, =, or \geq\} \ \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

- QPs are special classes of nonlinear optimization problems, and contain linear programming problems as special cases.

The Quadratic Programming Problem

The Quadratic Programming Problem

The quadratic programming (QP) problem

- In standard form,

The Quadratic Programming Problem

The quadratic programming (QP) problem

- In standard form, QPs may be represented as follows:

The Quadratic Programming Problem

The quadratic programming (QP) problem

- In standard form, QPs may be represented as follows:

$$\min_{x} \quad f(\mathbf{x}) := \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} + \mathbf{c}^T \cdot \mathbf{x}$$

The Quadratic Programming Problem

The quadratic programming (QP) problem

- In standard form, QPs may be represented as follows:

$$\min_{x} \quad f(\mathbf{x}) := \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} + \mathbf{c}^T \cdot \mathbf{x}$$
$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

The Quadratic Programming Problem

The quadratic programming (QP) problem

- In standard form, QPs may be represented as follows:

$$\begin{aligned} \min_{x} \quad f(\mathbf{x}) := \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} + \mathbf{c}^T \cdot \mathbf{x} \\ \mathbf{A} \cdot \mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{aligned}$$

The Quadratic Programming Problem

The Quadratic Programming Problem

Positive semidefinite matrix **Q**

- Recall that, when **Q** is a positive semidefinite matrix,

The Quadratic Programming Problem

Positive semidefinite matrix $\mathbf{Q}$

- Recall that, when $\mathbf{Q}$ is a positive semidefinite matrix, i.e., when $\mathbf{y}^T \cdot \mathbf{Q} \cdot \mathbf{y} \geq \mathbf{0}$ for all $\mathbf{y}$,

The Quadratic Programming Problem

Positive semidefinite matrix $\mathbf{Q}$

- Recall that, when $\mathbf{Q}$ is a positive semidefinite matrix, i.e., when $\mathbf{y}^T \cdot \mathbf{Q} \cdot \mathbf{y} \geq \mathbf{0}$ for all $\mathbf{y}$, the objective function of the problem is a convex function of $\mathbf{x}$.

The Quadratic Programming Problem

Positive semidefinite matrix $\mathbf{Q}$

- Recall that, when $\mathbf{Q}$ is a positive semidefinite matrix, i.e., when $\mathbf{y}^T \cdot \mathbf{Q} \cdot \mathbf{y} \geq \mathbf{0}$ for all $\mathbf{y}$, the objective function of the problem is a convex function of $\mathbf{x}$.
- In this case, a local minimizer of the objective function is also a global minimizer.

The Quadratic Programming Problem

Positive semidefinite matrix **Q**

- Recall that, when **Q** is a positive semidefinite matrix, i.e., when $\mathbf{y}^T \cdot \mathbf{Q} \cdot \mathbf{y} \geq \mathbf{0}$ for all **y**, the objective function of the problem is a convex function of **x**.
- In this case, a local minimizer of the objective function is also a global minimizer.
- The figure below shows the graph and contours of a quadratic function with a positive semidefinite **Q**.

The Quadratic Programming Problem

Positive semidefinite matrix **Q**

- Recall that, when **Q** is a positive semidefinite matrix, i.e., when $\mathbf{y}^T \cdot \mathbf{Q} \cdot \mathbf{y} \geq \mathbf{0}$ for all **y**, the objective function of the problem is a convex function of **x**.
- In this case, a local minimizer of the objective function is also a global minimizer.
- The figure below shows the graph and contours of a quadratic function with a positive semidefinite **Q**.

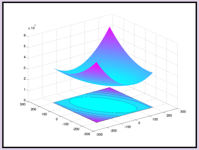


Figure: Graph and contours of a convex function

The Quadratic Programming Problem

The Quadratic Programming Problem

Indefinite or negative semidefinite matrix $\mathbf{Q}$

- When $\mathbf{Q}$ is not a positive semidefinite matrix (either indefinite or negative semidefinite),

The Quadratic Programming Problem

Indefinite or negative semidefinite matrix $\mathbf{Q}$

- When $\mathbf{Q}$ is not a positive semidefinite matrix (either indefinite or negative semidefinite), the objective function is nonconvex,

The Quadratic Programming Problem

Indefinite or negative semidefinite matrix $\mathbf{Q}$

- When $\mathbf{Q}$ is not a positive semidefinite matrix (either indefinite or negative semidefinite), the objective function is nonconvex, and may have local minimizers that are not global minimizers.

The Quadratic Programming Problem

Indefinite or negative semidefinite matrix **Q**

- When **Q** is not a positive semidefinite matrix (either indefinite or negative semidefinite), the objective function is nonconvex, and may have local minimizers that are not global minimizers.
- The figure below shows the graph and contours of a quadratic function with an indefinite **Q**.

The Quadratic Programming Problem

Indefinite or negative semidefinite matrix **Q**

- When **Q** is not a positive semidefinite matrix (either indefinite or negative semidefinite), the objective function is nonconvex, and may have local minimizers that are not global minimizers.
- The figure below shows the graph and contours of a quadratic function with an indefinite **Q**.

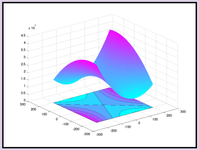


Figure: Graph and contours of a nonconvex function

The Quadratic Programming Problem

The Quadratic Programming Problem

The dual of the QP problem

- As in linear programming, we can develop a dual of quadratic programming problems.

The Quadratic Programming Problem

The dual of the QP problem

- As in linear programming, we can develop a dual of quadratic programming problems.
- The dual of f(**x**) is given below:

The Quadratic Programming Problem

The dual of the QP problem

- As in linear programming, we can develop a dual of quadratic programming problems.
- The dual of f(**x**) is given below:

$$\max_{x,y,s} \quad \mathbf{b}^T \cdot \mathbf{y} - \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x}$$

The Quadratic Programming Problem

The dual of the QP problem

- As in linear programming, we can develop a dual of quadratic programming problems.
- The dual of f(**x**) is given below:

$$\max_{x,y,s} \quad \mathbf{b}^T \cdot \mathbf{y} - \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x}$$

$$\mathbf{A}^T \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}$$

The Quadratic Programming Problem

The dual of the QP problem

- As in linear programming, we can develop a dual of quadratic programming problems.
- The dual of f(**x**) is given below:

$$\max_{x,y,s} \quad \mathbf{b}^T \cdot \mathbf{y} - \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x}$$
$$\mathbf{A}^T \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}$$
$$\mathbf{x} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0}$$

The Quadratic Programming Problem

The dual of the QP problem

- As in linear programming, we can develop a dual of quadratic programming problems.
- The dual of f(**x**) is given below:

$$\begin{aligned} \max_{x,y,s} \quad & \mathbf{b}^T \cdot \mathbf{y} - \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} \\ & \mathbf{A}^T \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c} \\ & \mathbf{x} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0} \end{aligned}$$

- Note that, unlike the case of linear programming,

The Quadratic Programming Problem

The dual of the QP problem

- As in linear programming, we can develop a dual of quadratic programming problems.
- The dual of f(**x**) is given below:

$$\begin{aligned} \max_{x,y,s} \quad & \mathbf{b}^T \cdot \mathbf{y} - \frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} \\ & \mathbf{A}^T \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c} \\ & \mathbf{x} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0} \end{aligned}$$

- Note that, unlike the case of linear programming, the variables of the primal quadratic programming problem also appear in the dual QP.

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

- *Suppose that* $\boldsymbol{x}$ *is a local optimal solution of the QP such that it satisfies* $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$,

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

- *Suppose that* $\boldsymbol{x}$ *is a local optimal solution of the QP such that it satisfies* $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$*, and assume that* $\boldsymbol{Q}$ *is a positive semidefinite matrix.*

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

- *Suppose that* $\boldsymbol{x}$ *is a local optimal solution of the QP such that it satisfies* $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$*, and assume that* $\boldsymbol{Q}$ *is a positive semidefinite matrix.*
- *Then, there exist vectors* $\boldsymbol{y}$ *and* $\boldsymbol{s}$ *such that the following conditions hold:*

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

- *Suppose that* $\mathbf{x}$ *is a local optimal solution of the QP such that it satisfies* $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$, *and assume that* $\mathbf{Q}$ *is a positive semidefinite matrix.*
- *Then, there exist vectors* $\mathbf{y}$ *and* $\mathbf{s}$ *such that the following conditions hold:*

$$\mathbf{A}^T \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

- *Suppose that $\boldsymbol{x}$ is a local optimal solution of the QP such that it satisfies $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$, and assume that $\boldsymbol{Q}$ is a positive semidefinite matrix.*
- *Then, there exist vectors $\boldsymbol{y}$ and $\boldsymbol{s}$ such that the following conditions hold:*

$$\begin{aligned} \boldsymbol{A}^T \cdot \boldsymbol{y} - \boldsymbol{Q} \cdot \boldsymbol{x} + \boldsymbol{s} &= \boldsymbol{c} \\ \boldsymbol{s} &\geq \boldsymbol{0} \end{aligned}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

- *Suppose that* $\boldsymbol{x}$ *is a local optimal solution of the QP such that it satisfies* $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$, *and assume that* $\boldsymbol{Q}$ *is a positive semidefinite matrix.*
- *Then, there exist vectors* $\boldsymbol{y}$ *and* $\boldsymbol{s}$ *such that the following conditions hold:*

$$\begin{aligned} \boldsymbol{A}^T \cdot \boldsymbol{y} - \boldsymbol{Q} \cdot \boldsymbol{x} + \boldsymbol{s} &= \boldsymbol{c} \\ \boldsymbol{s} &\geq \boldsymbol{0} \\ \boldsymbol{x}_i \cdot \boldsymbol{s}_i &= \boldsymbol{0}, \forall i \end{aligned}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker Theorem (as applied to the QP problem)

- *Suppose that* $\mathbf{x}$ *is a local optimal solution of the QP such that it satisfies* $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$*, and assume that* $\mathbf{Q}$ *is a positive semidefinite matrix.*
- *Then, there exist vectors* $\mathbf{y}$ *and* $\mathbf{s}$ *such that the following conditions hold:*

$$\begin{aligned} \mathbf{A}^T \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} &= \mathbf{c} \\ \mathbf{s} &\geq \mathbf{0} \\ \mathbf{x}_i \cdot \mathbf{s}_i &= \mathbf{0}, \forall i \end{aligned}$$

- *Futhermore,* $\mathbf{x}$ *is a global optimal solution.*

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem

Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems,

Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.

Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.
- If vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{s}$ satisfy conditions of the KKT theorem as well

Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.
- If vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{s}$ satisfy conditions of the KKT theorem as well as the primal feasibility conditions

Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.
- If vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{s}$ satisfy conditions of the KKT theorem as well as the primal feasibility conditions

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.
- If vectors **x**, **y**, and **s** satisfy conditions of the KKT theorem as well as the primal feasibility conditions

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.
- If vectors **x**, **y**, and **s** satisfy conditions of the KKT theorem as well as the primal feasibility conditions

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$

then **x** is a global optimal solution.

Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.
- If vectors **x**, **y**, and **s** satisfy conditions of the KKT theorem as well as the primal feasibility conditions

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$

  then **x** is a global optimal solution.
- In other words,

Karush-Kuhn-Tucker (KKT) Optimality Conditions

More about the KKT theorem

- The positive semidefiniteness condition related to the Hessian of the Lagrangian function in the KKT theorem is automatically satisfied for convex quadratic programming problems, and therefore is not included in the theorem above.
- If vectors **x**, **y**, and **s** satisfy conditions of the KKT theorem as well as the primal feasibility conditions

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$
$$\mathbf{x} \geq \mathbf{0}$$

  then **x** is a global optimal solution.
- In other words, all 5 conditions are both necessary and sufficient for **x**, **y**, and **s** to describe a global optimal solution of the QP problem.

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker (KKT) Optimality Conditions

KKT conditions

Karush-Kuhn-Tucker (KKT) Optimality Conditions

KKT conditions

- In a manner similar to linear programming,

Karush-Kuhn-Tucker (KKT) Optimality Conditions

KKT conditions

- In a manner similar to linear programming, the optimality conditions can be seen as a collection of conditions for:

Karush-Kuhn-Tucker (KKT) Optimality Conditions

KKT conditions

- In a manner similar to linear programming, the optimality conditions can be seen as a collection of conditions for:
  1. primal feasibility: $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$;

Karush-Kuhn-Tucker (KKT) Optimality Conditions

KKT conditions

- In a manner similar to linear programming, the optimality conditions can be seen as a collection of conditions for:
  1. primal feasibility: $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$;
  2. dual feasibility: $\mathbf{A}^T \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}$, $\mathbf{s} \geq \mathbf{0}$;

Karush-Kuhn-Tucker (KKT) Optimality Conditions

KKT conditions

- In a manner similar to linear programming, the optimality conditions can be seen as a collection of conditions for:
  1. primal feasibility: $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$;
  2. dual feasibility: $\mathbf{A}^T \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}$, $\mathbf{s} \geq \mathbf{0}$;
  3. complementary slackness: for each $i = 1, \cdots, n$ we have $\mathbf{x}_i \cdot \mathbf{s}_i = \mathbf{0}$.

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Consider the following quadratic program

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Consider the following quadratic program

$$\min \quad 2 \cdot x_1^2 + x_2^2 + 4 \cdot x_3^2$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Consider the following quadratic program

$$\begin{aligned} \min \quad & 2 \cdot x_1^2 + x_2^2 + 4 \cdot x_3^2 \\ & x_1 + 2 \cdot x_2 - x_3 \quad = \quad 6 \end{aligned}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Consider the following quadratic program

$$
\begin{aligned}
\min \quad & 2 \cdot x_1^2 + x_2^2 + 4 \cdot x_3^2 \\
& x_1 + 2 \cdot x_2 - x_3 = 6 \\
& 2 \cdot x_1 - 2 \cdot x_2 + 3 \cdot x_3 = 12
\end{aligned}
$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Consider the following quadratic program

$$
\begin{array}{lrcl}
\min & 2 \cdot x_1^2 + x_2^2 + 4 \cdot x_3^2 & & \\
 & x_1 + 2 \cdot x_2 - x_3 & = & 6 \\
 & 2 \cdot x_1 - 2 \cdot x_2 + 3 \cdot x_3 & = & 12 \\
 & x_1, x_2, x_3 & \geq & 0
\end{array}
$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Is the quadratic function convex?

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Is the quadratic function convex? Yes.

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Is the quadratic function convex? Yes.

$$\mathbf{Q} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Is the quadratic function convex? Yes.

$$\mathbf{Q} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix} \text{ which is a positive semidefinite matrix.}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Is the quadratic function convex? Yes.

$$\mathbf{Q} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix} \text{ which is a positive semidefinite matrix.}$$

- Set up the KKT conditions for the optimal solution in matrix form,

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Is the quadratic function convex? Yes.

$$\mathbf{Q} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix} \text{ which is a positive semidefinite matrix.}$$

- Set up the KKT conditions for the optimal solution in matrix form, and show how you would solve for $\mathbf{x}$ and $\mathbf{y}$.

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Our KKT conditions for the optimal solution in matrix form is as follows:

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Our KKT conditions for the optimal solution in matrix form is as follows:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Our KKT conditions for the optimal solution in matrix form is as follows:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Our KKT conditions for the optimal solution in matrix form is as follows:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Our KKT conditions for the optimal solution in matrix form is as follows:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} =$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Our KKT conditions for the optimal solution in matrix form is as follows:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ \mathbf{b} \end{bmatrix}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Our KKT conditions for the optimal solution in matrix form is as follows:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ \mathbf{b} \end{bmatrix}$$

Where

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Our KKT conditions for the optimal solution in matrix form is as follows:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ \mathbf{b} \end{bmatrix}$$

Where

$$\mathbf{Q} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix},$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Our KKT conditions for the optimal solution in matrix form is as follows:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ \mathbf{b} \end{bmatrix}$$

Where

$$\mathbf{Q} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 1 & 2 & -1 \\ 2 & -2 & 3 \end{bmatrix},$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Our KKT conditions for the optimal solution in matrix form is as follows:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ \mathbf{b} \end{bmatrix}$$

Where

$$\mathbf{Q} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 1 & 2 & -1 \\ 2 & -2 & 3 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- Our KKT conditions for the optimal solution in matrix form is as follows:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ \mathbf{b} \end{bmatrix}$$

Where

$$\mathbf{Q} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 1 & 2 & -1 \\ 2 & -2 & 3 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 6 \\ 12 \end{bmatrix}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- We get the following linear system:

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- We get the following linear system:

$$\begin{bmatrix} -4 & 0 & 0 & 1 & 2 \\ 0 & -2 & 0 & 2 & -2 \\ 0 & 0 & -8 & -1 & 3 \\ 1 & 2 & -1 & 0 & 0 \\ 2 & -2 & 3 & 0 & 0 \end{bmatrix}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- We get the following linear system:

$$\begin{bmatrix} -4 & 0 & 0 & 1 & 2 \\ 0 & -2 & 0 & 2 & -2 \\ 0 & 0 & -8 & -1 & 3 \\ 1 & 2 & -1 & 0 & 0 \\ 2 & -2 & 3 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \end{bmatrix}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- We get the following linear system:

$$\begin{bmatrix} -4 & 0 & 0 & 1 & 2 \\ 0 & -2 & 0 & 2 & -2 \\ 0 & 0 & -8 & -1 & 3 \\ 1 & 2 & -1 & 0 & 0 \\ 2 & -2 & 3 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- We get the following linear system:

$$\begin{bmatrix} -4 & 0 & 0 & 1 & 2 \\ 0 & -2 & 0 & 2 & -2 \\ 0 & 0 & -8 & -1 & 3 \\ 1 & 2 & -1 & 0 & 0 \\ 2 & -2 & 3 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 6 \\ 12 \end{bmatrix}$$

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- We get the following linear system:

$$\begin{bmatrix} -4 & 0 & 0 & 1 & 2 \\ 0 & -2 & 0 & 2 & -2 \\ 0 & 0 & -8 & -1 & 3 \\ 1 & 2 & -1 & 0 & 0 \\ 2 & -2 & 3 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 6 \\ 12 \end{bmatrix}$$

- After solving the system,

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- We get the following linear system:

$$\begin{bmatrix} -4 & 0 & 0 & 1 & 2 \\ 0 & -2 & 0 & 2 & -2 \\ 0 & 0 & -8 & -1 & 3 \\ 1 & 2 & -1 & 0 & 0 \\ 2 & -2 & 3 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 6 \\ 12 \end{bmatrix}$$

- After solving the system, we find that $\mathbf{x} = (5.045, 1.194, 1.433)$ is an optimal solution

Karush-Kuhn-Tucker (KKT) Optimality Conditions

Exercise 3

- We get the following linear system:

$$\begin{bmatrix} -4 & 0 & 0 & 1 & 2 \\ 0 & -2 & 0 & 2 & -2 \\ 0 & 0 & -8 & -1 & 3 \\ 1 & 2 & -1 & 0 & 0 \\ 2 & -2 & 3 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 6 \\ 12 \end{bmatrix}$$

- After solving the system, we find that $\mathbf{x} = (5.045, 1.194, 1.433)$ is an optimal solution with $\mathbf{y} = (7.522, 6.328)$ and $\mathbf{s} = (0, 0, 0)$.

Interior-Point Methods

Interior-Point Methods

Interior-Point Method

Interior-Point Methods

Interior-Point Method

Interior-Point Method(IPM) finds primal-dual solutions $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ by applying variants of Newton's method to the optimality conditions

Interior-Point Methods

Interior-Point Method

Interior-Point Method(IPM) finds primal-dual solutions $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ by applying variants of Newton's method to the optimality conditions and modifying the search directions and step lengths so that $\mathbf{x} \geq 0$ and $\mathbf{s} \geq 0$ are satisfied strictly at every iteration.

Interior-Point Methods

Interior-Point Method

Interior-Point Method(IPM) finds primal-dual solutions $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ by applying variants of Newton's method to the optimality conditions and modifying the search directions and step lengths so that $\mathbf{x} \geq 0$ and $\mathbf{s} \geq 0$ are satisfied strictly at every iteration.

Rewrite the Optimality Conditions

Interior-Point Methods

Interior-Point Method

Interior-Point Method(IPM) finds primal-dual solutions $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ by applying variants of Newton's method to the optimality conditions and modifying the search directions and step lengths so that $\mathbf{x} \geq 0$ and $\mathbf{s} \geq 0$ are satisfied strictly at every iteration.

Rewrite the Optimality Conditions

$$\mathbf{F}(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \begin{bmatrix} \mathbf{A}^T\mathbf{y} - \mathbf{Q}\mathbf{x} + \mathbf{s} - \mathbf{c} \\ \mathbf{A}\mathbf{x} - \mathbf{b} \\ \mathbf{X}\mathbf{S}\mathbf{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (\mathbf{x}, \mathbf{s}) \geq 0$$

Interior-Point Methods

Interior-Point Method

Interior-Point Method(IPM) finds primal-dual solutions $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ by applying variants of Newton's method to the optimality conditions and modifying the search directions and step lengths so that $\mathbf{x} \geq 0$ and $\mathbf{s} \geq 0$ are satisfied strictly at every iteration.

Rewrite the Optimality Conditions

$$\mathbf{F}(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \begin{bmatrix} \mathbf{A}^\mathsf{T}\mathbf{y} - \mathbf{Q}\mathbf{x} + \mathbf{s} - \mathbf{c} \\ \mathbf{A}\mathbf{x} - \mathbf{b} \\ \mathbf{X}\mathbf{S}\mathbf{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (\mathbf{x}, \mathbf{s}) \geq 0$$

$\mathbf{X}$ and $\mathbf{S}$ are diagonal matrices such that $\mathbf{X}_{ii} = x_i$ and $\mathbf{X}_{ij} = 0, i \neq j$, and similarly for $\mathbf{S}$.

Strategy

Strategy

Remark

Strategy

Remark

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.

Strategy

Remark

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.
- Since the system is square we can apply Newton's method.

Strategy

Remark

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.
- Since the system is square we can apply Newton's method.
- The existence of nonnegative constraints creates a difficulty, otherwise we can use Newton's method directly.

Strategy

Remark

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.
- Since the system is square we can apply Newton's method.
- The existence of nonnegative constraints creates a difficulty, otherwise we can use Newton's method directly.
- Since we are generating iterates for both the primal and dual problems, this version of IPMs are often called *primal-dual interior-point methods*.

Strategy

Remark

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.
- Since the system is square we can apply Newton's method.
- The existence of nonnegative constraints creates a difficulty, otherwise we can use Newton's method directly.
- Since we are generating iterates for both the primal and dual problems, this version of IPMs are often called *primal-dual interior-point methods*.

Strategy of Applying A Modified Newton's Method

Strategy

Remark

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.
- Since the system is square we can apply Newton's method.
- The existence of nonnegative constraints creates a difficulty, otherwise we can use Newton's method directly.
- Since we are generating iterates for both the primal and dual problems, this version of IPMs are often called *primal-dual interior-point methods*.

Strategy of Applying A Modified Newton's Method

1. Identify an initial solution $(\mathbf{x^0}, \mathbf{y^0}, \mathbf{s^0})$, which satisfies the first two constraints (linear) and $(\mathbf{x^0}, \mathbf{s^0}) > 0$, but not the third one.

Strategy

Remark

- $x_i \cdot s_i$ is a nonlinear constraint, so we cannot solve this system using linear system solution methods such as Gaussian elimination.
- Since the system is square we can apply Newton's method.
- The existence of nonnegative constraints creates a difficulty, otherwise we can use Newton's method directly.
- Since we are generating iterates for both the primal and dual problems, this version of IPMs are often called *primal-dual interior-point methods*.

Strategy of Applying A Modified Newton's Method

1. Identify an initial solution $(\mathbf{x^0}, \mathbf{y^0}, \mathbf{s^0})$, which satisfies the first two constraints (linear) and $(\mathbf{x^0}, \mathbf{s^0}) > 0$, but not the third one.
2. Generate new points $(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k})$ that also satisfy these same conditions and get progressively closer to satisfying the third constraint.

Algorithms for IPMs with pure Newton direction

Algorithms for IPMs with pure Newton direction

Definition

Algorithms for IPMs with pure Newton direction

Definition

- Feasible set: $\mathcal{F} := \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : \mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{A}^\mathsf{T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}, \mathbf{x} \geq 0, \mathbf{s} \geq 0\}$

Algorithms for IPMs with pure Newton direction

Definition

- Feasible set: $\mathcal{F} := \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : \mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{A}^\mathsf{T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}, \mathbf{x} \geq 0, \mathbf{s} \geq 0\}$
  $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}$ is a feasible point.

Algorithms for IPMs with pure Newton direction

Definition

- Feasible set: $\mathcal{F} := \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : \mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{A}^{\mathbf{T}} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}, \mathbf{x} \geq 0, \mathbf{s} \geq 0\}$
  $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}$ is a feasible point.
- Strictly feasible set:
  $\mathcal{F}^0 := \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : \mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{A}^{\mathbf{T}} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}, \mathbf{x} > 0, \mathbf{s} > 0\}$

Algorithms for IPMs with pure Newton direction

Definition

- Feasible set: $\mathcal{F} := \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : \mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{A}^{\mathbf{T}} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}, \mathbf{x} \geq 0, \mathbf{s} \geq 0\}$
  $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}$ is a feasible point.
- Strictly feasible set:
  $\mathcal{F}^0 := \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : \mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{A}^{\mathbf{T}} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}, \mathbf{x} > 0, \mathbf{s} > 0\}$
  $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^0$ is a strictly feasible solution, which lies in the *interior* of the region defined by those constraints rather than being on the boundary.

Algorithms for IPMs with pure Newton direction

Definition

- Feasible set: $\mathcal{F} := \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : \mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{A}^\mathsf{T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}, \mathbf{x} \geq 0, \mathbf{s} \geq 0\}$
  $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}$ is a feasible point.
- Strictly feasible set:
  $\mathcal{F}^0 := \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) : \mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{A}^\mathsf{T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}, \mathbf{x} > 0, \mathbf{s} > 0\}$
  $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^0$ is a strictly feasible solution, which lies in the *interior* of the region defined by those constraints rather than being on the boundary. So $\mathcal{F}^0$ is the relative interior of the set $\mathcal{F}$.

Algorithms for IPMs with pure Newton direction

Algorithms for IPMs with pure Newton direction

Exercise 4

Algorithms for IPMs with pure Newton direction

Exercise 4

Consider the quadratic programming problem given below:

Algorithms for IPMs with pure Newton direction

Exercise 4

Consider the quadratic programming problem given below:

$$\begin{aligned} \min \quad & x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\ & +2 \cdot x_1 + x_2 + 3 \cdot x_3 \end{aligned}$$

Algorithms for IPMs with pure Newton direction

Exercise 4

Consider the quadratic programming problem given below:

$$
\begin{aligned}
\min \quad & x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\
& +2 \cdot x_1 + x_2 + 3 \cdot x_3 \\
& x_1 + x_2 + x_3 = 1
\end{aligned}
$$

Algorithms for IPMs with pure Newton direction

Exercise 4

Consider the quadratic programming problem given below:

$$\begin{aligned} \min \quad & x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\ & +2 \cdot x_1 + x_2 + 3 \cdot x_3 \\ & x_1 + x_2 + x_3 = 1 \\ & x_1 - x_2 = 0 \end{aligned}$$

Algorithms for IPMs with pure Newton direction

Exercise 4

Consider the quadratic programming problem given below:

$$\begin{aligned} \min \quad & x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\ & +2 \cdot x_1 + x_2 + 3 \cdot x_3 \\ & x_1 + x_2 + x_3 = 1 \\ & x_1 - x_2 = 0 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Algorithms for IPMs with pure Newton direction

Exercise 4

Consider the quadratic programming problem given below:

$$\begin{aligned} \min \quad & x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\ & +2 \cdot x_1 + x_2 + 3 \cdot x_3 \\ & x_1 + x_2 + x_3 = 1 \\ & x_1 - x_2 = 0 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

The current primal-dual estimate of the solution $\mathbf{x^k} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$, $\mathbf{y^k} = (1, \frac{1}{2})^T$, and $\mathbf{s^k} = (\frac{3}{2}, \frac{11}{6}, \frac{10}{3})^T$.

Algorithms for IPMs with pure Newton direction

Exercise 4

Consider the quadratic programming problem given below:

$$\begin{aligned} \min \quad & x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\ & +2 \cdot x_1 + x_2 + 3 \cdot x_3 \\ & x_1 + x_2 + x_3 = 1 \\ & x_1 - x_2 = 0 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

The current primal-dual estimate of the solution $\mathbf{x^k} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$, $\mathbf{y^k} = (1, \frac{1}{2})^T$, and $\mathbf{s^k} = (\frac{3}{2}, \frac{11}{6}, \frac{10}{3})^T$. Is $(x, y, s) \in \mathcal{F}$?

Algorithms for IPMs with pure Newton direction

Exercise 4

Consider the quadratic programming problem given below:

$$\begin{aligned} \min \quad & x_1 \cdot x_2 + x_1^2 + \frac{3}{2} \cdot x_2^2 + 2 \cdot x_3^2 \\ & +2 \cdot x_1 + x_2 + 3 \cdot x_3 \\ & x_1 + x_2 + x_3 = 1 \\ & x_1 - x_2 = 0 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

The current primal-dual estimate of the solution $\mathbf{x}^\mathbf{k} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$, $\mathbf{y}^\mathbf{k} = (1, \frac{1}{2})^T$, and $\mathbf{s}^\mathbf{k} = (\frac{3}{2}, \frac{11}{6}, \frac{10}{3})^T$. Is $(x, y, s) \in \mathcal{F}$? How about $\mathcal{F}^0$?

Algorithms for IPMs with pure Newton direction

Algorithms for IPMs with pure Newton direction

Two Basic Ingredients of IPMs

Algorithms for IPMs with pure Newton direction

Two Basic Ingredients of IPMs

1. A measure that can be used to evaluate and compare the quality of alternative solutions and search directions.

Algorithms for IPMs with pure Newton direction

Two Basic Ingredients of IPMs

1. A measure that can be used to evaluate and compare the quality of alternative solutions and search directions.
2. A method to generate a better solution, with respect to the measure just mentioned, from a non-optimal solution.

Algorithms for IPMs with pure Newton direction

Algorithms for IPMs with pure Newton direction

Pure Newton Step

Algorithms for IPMs with pure Newton direction

Pure Newton Step

Assume that we have a current estimate (x^k, y^k, s^k) of the optimal solution to the problem.

Algorithms for IPMs with pure Newton direction

Pure Newton Step

Assume that we have a current estimate (x^k, y^k, s^k) of the optimal solution to the problem.

The Newton step from this point is determined by solving the following system of linear equations:

Algorithms for IPMs with pure Newton direction

Pure Newton Step

Assume that we have a current estimate (x^k, y^k, s^k) of the optimal solution to the problem.

The Newton step from this point is determined by solving the following system of linear equations:

$$\mathbf{J}(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k}) \begin{bmatrix} \Delta \mathbf{x^k} \\ \Delta \mathbf{y^k} \\ \Delta \mathbf{s^k} \end{bmatrix} = -\mathbf{F}(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k}),$$

Algorithms for IPMs with pure Newton direction

Pure Newton Step

Assume that we have a current estimate (x^k, y^k, s^k) of the optimal solution to the problem.

The Newton step from this point is determined by solving the following system of linear equations:

$$\mathbf{J}(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k}) \begin{bmatrix} \Delta \mathbf{x^k} \\ \Delta \mathbf{y^k} \\ \Delta \mathbf{s^k} \end{bmatrix} = -\mathbf{F}(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k}),$$

where $\mathbf{J}(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k})$ is the Jacobian of the function F and $[\Delta \mathbf{x^k}, \Delta \mathbf{y^k}, \Delta \mathbf{s^k}]^T$ is the search direction.

Algorithms for IPMs with pure Newton direction

Algorithms for IPMs with pure Newton direction

Pure Newton Step

Algorithms for IPMs with pure Newton direction

Pure Newton Step

First, we observe that

Algorithms for IPMs with pure Newton direction

Pure Newton Step

First, we observe that

$$\mathbf{J}(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) = \begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S}^k & 0 & \mathbf{X}^k \end{bmatrix}$$

Algorithms for IPMs with pure Newton direction

Pure Newton Step

First, we observe that

$$\mathbf{J}(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) = \begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S}^k & 0 & \mathbf{X}^k \end{bmatrix}$$

where, $\mathbf{X}^k$ and $\mathbf{S}^k$ are diagonal matrices with the components of the vectors $\mathbf{x}^k$ and $\mathbf{s}^k$ along their diagonals.

Algorithms for IPMs with pure Newton direction

Pure Newton Step

First, we observe that

$$\mathbf{J}(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) = \begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S}^k & 0 & \mathbf{X}^k \end{bmatrix}$$

where, $\mathbf{X}^k$ and $\mathbf{S}^k$ are diagonal matrices with the components of the vectors $\mathbf{x}^k$ and $\mathbf{s}^k$ along their diagonals.
Furthermore, if $(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) \in \mathcal{F}^0$, then

Algorithms for IPMs with pure Newton direction

Pure Newton Step

First, we observe that

$$\mathbf{J}(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) = \begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S}^k & 0 & \mathbf{X}^k \end{bmatrix}$$

where, $\mathbf{X}^k$ and $\mathbf{S}^k$ are diagonal matrices with the components of the vectors $\mathbf{x}^k$ and $\mathbf{s}^k$ along their diagonals.
Furthermore, if $(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) \in \mathcal{F}^0$, then

$$\mathbf{F}(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k) = \begin{bmatrix} 0 \\ 0 \\ \mathbf{X}^k\mathbf{S}^k\mathbf{e} \end{bmatrix}$$

Algorithms for IPMs with pure Newton direction

Algorithms for IPMs with pure Newton direction

Pure Newton Step

Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S}^k & 0 & \mathbf{X}^k \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^k \\ \Delta \mathbf{y}^k \\ \Delta \mathbf{s}^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{X}^k \mathbf{S}^k \mathbf{e} \end{bmatrix}.$$

Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^{\mathsf{T}} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S}^{k} & 0 & \mathbf{X}^{k} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}^{k} \\ \Delta\mathbf{y}^{k} \\ \Delta\mathbf{s}^{k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{X}^{k}\mathbf{S}^{k}\mathbf{e} \end{bmatrix}.$$

So by solving this equation system, we can find the search step for $(k+1)$th iteration.

Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S}^k & 0 & \mathbf{X}^k \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^k \\ \Delta \mathbf{y}^k \\ \Delta \mathbf{s}^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{X}^k \mathbf{S}^k \mathbf{e} \end{bmatrix}.$$

So by solving this equation system, we can find the search step for $(k+1)$th iteration.

Exercise 5

Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^{\mathbf{T}} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S}^{\mathbf{k}} & 0 & \mathbf{X}^{\mathbf{k}} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}^{\mathbf{k}} \\ \Delta\mathbf{y}^{\mathbf{k}} \\ \Delta\mathbf{s}^{\mathbf{k}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{X}^{\mathbf{k}}\mathbf{S}^{\mathbf{k}}\mathbf{e} \end{bmatrix}.$$

So by solving this equation system, we can find the search step for $(k+1)$th iteration.

Exercise 5

Consider the quadratic programming problem given in Exercise 4

Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^\mathbf{T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S}^\mathbf{k} & 0 & \mathbf{X}^\mathbf{k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^\mathbf{k} \\ \Delta \mathbf{y}^\mathbf{k} \\ \Delta \mathbf{s}^\mathbf{k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{X}^\mathbf{k}\mathbf{S}^\mathbf{k}\mathbf{e} \end{bmatrix}.$$

So by solving this equation system, we can find the search step for $(k+1)$th iteration.

Exercise 5

Consider the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $\mathbf{x}^\mathbf{k} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$,

Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x^k} \\ \Delta \mathbf{y^k} \\ \Delta \mathbf{s^k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{X^k S^k e} \end{bmatrix}.$$

So by solving this equation system, we can find the search step for $(k+1)$th iteration.

Exercise 5

Consider the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $\mathbf{x^k} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$, $\mathbf{y^k} = (1, \frac{1}{2})^T$,

Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x^k} \\ \Delta \mathbf{y^k} \\ \Delta \mathbf{s^k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{X^k S^k e} \end{bmatrix}.$$

So by solving this equation system, we can find the search step for $(k+1)$th iteration.

Exercise 5

Consider the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $\mathbf{x^k} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$, $\mathbf{y^k} = (1, \frac{1}{2})^T$, and $\mathbf{s^k} = (\frac{3}{2}, \frac{11}{6}, \frac{10}{3})^T$.

Algorithms for IPMs with pure Newton direction

Pure Newton Step

The Newton equation reduces to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x^k} \\ \Delta\mathbf{y^k} \\ \Delta\mathbf{s^k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{X^k S^k e} \end{bmatrix}.$$

So by solving this equation system, we can find the search step for $(k+1)$th iteration.

Exercise 5

Consider the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $\mathbf{x^k} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$, $\mathbf{y^k} = (1, \frac{1}{2})^T$, and $\mathbf{s^k} = (\frac{3}{2}, \frac{11}{6}, \frac{10}{3})^T$. Form and solve the Newton equation for this problem at $(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k})$.

Algorithms for IPMs with pure Newton direction

Algorithms for IPMs with pure Newton direction

Step-size Parameter

Algorithms for IPMs with pure Newton direction

Step-size Parameter

- In our case, this action may not be permissible,

Algorithms for IPMs with pure Newton direction

Step-size Parameter

- In our case, this action may not be permissible, since the Newton step may take us to a new point that does not necessarily satisfy the nonnegativity constraints

Algorithms for IPMs with pure Newton direction

Step-size Parameter

- In our case, this action may not be permissible, since the Newton step may take us to a new point that does not necessarily satisfy the nonnegativity constraints $\mathbf{x} \geq 0$ and $\mathbf{s} \geq 0$.

Algorithms for IPMs with pure Newton direction

Step-size Parameter

- In our case, this action may not be permissible, since the Newton step may take us to a new point that does not necessarily satisfy the nonnegativity constraints $\mathbf{x} \geq 0$ and $\mathbf{s} \geq 0$.
- To avoid such violations, we sill seek a *step-size parameter* $\alpha^k \in (0, 1]$ such that $\mathbf{x^k} + \alpha^k \cdot \Delta\mathbf{x^k} > 0$ and $\mathbf{s^k} + \alpha^k \cdot \Delta\mathbf{s^k} > 0$.

Algorithms for IPMs with pure Newton direction

Step-size Parameter

- In our case, this action may not be permissible, since the Newton step may take us to a new point that does not necessarily satisfy the nonnegativity constraints $\mathbf{x} \geq 0$ and $\mathbf{s} \geq 0$.
- To avoid such violations, we sill seek a *step-size parameter* $\alpha^k \in (0, 1]$ such that $\mathbf{x^k} + \alpha^k \cdot \Delta\mathbf{x^k} > 0$ and $\mathbf{s^k} + \alpha^k \cdot \Delta\mathbf{s^k} > 0$.
- Once we determine the step-size parameter, we choose the next iterate as

$$(\mathbf{x^{k+1}}, \mathbf{y^{k+1}}, \mathbf{s^{k+1}}) = (\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k}) + \alpha^k \cdot (\Delta\mathbf{x^k}, \Delta\mathbf{y^k}, \Delta\mathbf{s^k}).$$

Algorithms for IPMs with pure Newton direction

Algorithms for IPMs with pure Newton direction

Weakness of IPMs with pure Newton direction

Algorithms for IPMs with pure Newton direction

Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction ($\alpha^k \ll 1$) before violating the condition $\mathbf{x^k} + \alpha^k \cdot \Delta\mathbf{x^k} > 0$ and $\mathbf{s^k} + \alpha^k \cdot \Delta\mathbf{s^k} > 0$;

Algorithms for IPMs with pure Newton direction

Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction ($\alpha^k \ll 1$) before violating the condition $\mathbf{x^k} + \alpha^k \cdot \Delta\mathbf{x^k} > 0$ and $\mathbf{s^k} + \alpha^k \cdot \Delta\mathbf{s^k} > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Algorithms for IPMs with pure Newton direction

Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction ($\alpha^k \ll 1$) before violating the condition $\mathbf{x^k} + \alpha^k \cdot \Delta\mathbf{x^k} > 0$ and $\mathbf{s^k} + \alpha^k \cdot \Delta\mathbf{s^k} > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways

Algorithms for IPMs with pure Newton direction

Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction ($\alpha^k \ll 1$) before violating the condition $\mathbf{x^k} + \alpha^k \cdot \Delta\mathbf{x^k} > 0$ and $\mathbf{s^k} + \alpha^k \cdot \Delta\mathbf{s^k} > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways

1. They bias the search direction toward the interior of the nonnegative orthant $(\mathbf{x}, \mathbf{s}) \geq 0$

Algorithms for IPMs with pure Newton direction

Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction ($\alpha^k \ll 1$) before violating the condition $\mathbf{x^k} + \alpha^k \cdot \Delta\mathbf{x^k} > 0$ and $\mathbf{s^k} + \alpha^k \cdot \Delta\mathbf{s^k} > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways

1. They bias the search direction toward the interior of the nonnegative orthant $(\mathbf{x}, \mathbf{s}) \geq 0$ so that we can move further along the direction before one of the components of $(\mathbf{x}, \mathbf{s})$ becomes negative.

Algorithms for IPMs with pure Newton direction

Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction ($\alpha^k \ll 1$) before violating the condition $\mathbf{x^k} + \alpha^k \cdot \Delta\mathbf{x^k} > 0$ and $\mathbf{s^k} + \alpha^k \cdot \Delta\mathbf{s^k} > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways

1. They bias the search direction toward the interior of the nonnegative orthant $(\mathbf{x}, \mathbf{s}) \geq 0$ so that we can move further along the direction before one of the components of $(\mathbf{x}, \mathbf{s})$ becomes negative.
2. They keep the component of $(\mathbf{x}, \mathbf{s})$ from moving "too close" to the boundary of the nonnegative of orthant.

Algorithms for IPMs with pure Newton direction

Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction ($\alpha^k \ll 1$) before violating the condition $\mathbf{x^k} + \alpha^k \cdot \Delta\mathbf{x^k} > 0$ and $\mathbf{s^k} + \alpha^k \cdot \Delta\mathbf{s^k} > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways

1. They bias the search direction toward the interior of the nonnegative orthant $(\mathbf{x}, \mathbf{s}) \geq 0$ so that we can move further along the direction before one of the components of $(\mathbf{x}, \mathbf{s})$ becomes negative.
2. They keep the component of $(\mathbf{x}, \mathbf{s})$ from moving "too close" to the boundary of the nonnegative of orthant. Search directions computed from points that are close to the boundary tend to be distorted,

Algorithms for IPMs with pure Newton direction

Weakness of IPMs with pure Newton direction

- We often can take only a small step along the direction ($\alpha^k \ll 1$) before violating the condition $\mathbf{x^k} + \alpha^k \cdot \Delta\mathbf{x^k} > 0$ and $\mathbf{s^k} + \alpha^k \cdot \Delta\mathbf{s^k} > 0$; hence, the pure Newton direction often does not allow us to make much more progress towards a solution.

Modify the basic Newton procedure in two important ways

1. They bias the search direction toward the interior of the nonnegative orthant $(\mathbf{x}, \mathbf{s}) \geq 0$ so that we can move further along the direction before one of the components of $(\mathbf{x}, \mathbf{s})$ becomes negative.
2. They keep the component of $(\mathbf{x}, \mathbf{s})$ from moving "too close" to the boundary of the nonnegative of orthant. Search directions computed from points that are close to the boundary tend to be distorted, and little progress can be made along them.

Algorithms for IPMs with centered Newton direction

Algorithms for IPMs with centered Newton direction

The Central Path

Algorithms for IPMs with centered Newton direction

The Central Path

The central path $\mathcal{C}$ is an arc of strictly feasible points(any point in $\mathcal{C}$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm.

Algorithms for IPMs with centered Newton direction

The Central Path

The central path $\mathcal{C}$ is an arc of strictly feasible points(any point in $\mathcal{C}$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$,

Algorithms for IPMs with centered Newton direction

The Central Path

The central path $\mathcal{C}$ is an arc of strictly feasible points(any point in $\mathcal{C}$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ on the central path are obtained as solutions of the following system:

Algorithms for IPMs with centered Newton direction

The Central Path

The central path $\mathcal{C}$ is an arc of strictly feasible points(any point in $\mathcal{C}$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ on the central path are obtained as solutions of the following system:

$$\mathbf{F}(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau \cdot \mathbf{e} \end{bmatrix}, (\mathbf{x}_\tau, \mathbf{s}_\tau) > 0.$$

Algorithms for IPMs with centered Newton direction

The Central Path

The central path $\mathcal{C}$ is an arc of strictly feasible points(any point in $\mathcal{C}$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ on the central path are obtained as solutions of the following system:

$$\mathbf{F}(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau \cdot \mathbf{e} \end{bmatrix}, (\mathbf{x}_\tau, \mathbf{s}_\tau) > 0.$$

Then, the central path $\mathcal{C}$ is defined as

Algorithms for IPMs with centered Newton direction

The Central Path

The central path $\mathcal{C}$ is an arc of strictly feasible points(any point in $\mathcal{C}$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ on the central path are obtained as solutions of the following system:

$$\mathbf{F}(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau \cdot \mathbf{e} \end{bmatrix}, (\mathbf{x}_\tau, \mathbf{s}_\tau) > 0.$$

Then, the central path $\mathcal{C}$ is defined as

$$\mathcal{C} = \{(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau) : \tau > 0\}.$$

Algorithms for IPMs with centered Newton direction

The Central Path

The central path $\mathcal{C}$ is an arc of strictly feasible points(any point in $\mathcal{C}$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ on the central path are obtained as solutions of the following system:

$$\mathbf{F}(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau \cdot \mathbf{e} \end{bmatrix}, (\mathbf{x}_\tau, \mathbf{s}_\tau) > 0.$$

Then, the central path $\mathcal{C}$ is defined as

$$\mathcal{C} = \{(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau) : \tau > 0\}.$$

The third constraint can be rewritten as

Algorithms for IPMs with centered Newton direction

The Central Path

The central path $\mathcal{C}$ is an arc of strictly feasible points(any point in $\mathcal{C}$ is in $\mathcal{F}^0$) that plays a vital role in the theory of primal-dual algorithm. It is parameterized by a scalar $\tau > 0$, and the points $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ on the central path are obtained as solutions of the following system:

$$\mathbf{F}(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau \cdot \mathbf{e} \end{bmatrix}, (\mathbf{x}_\tau, \mathbf{s}_\tau) > 0.$$

Then, the central path $\mathcal{C}$ is defined as

$$\mathcal{C} = \{(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau) : \tau > 0\}.$$

The third constraint can be rewritten as

$$(x_\tau)_i \cdot (s_\tau)_i = \tau, \forall i.$$

The Central Path

The Central Path

Remark

The Central Path

Remark

- Instead of the complementary condition, we require the products $(x_\tau)_i \cdot (s_\tau)_i$ have the same value for all i.

The Central Path

Remark

- Instead of the complementary condition, we require the products $(x_\tau)_i \cdot (s_\tau)_i$ have the same value for all i.
- The system has a unique solution for every $\tau > 0$, provided that $\mathcal{F}^0$ is nonempty.

The Central Path

Remark

- Instead of the complementary condition, we require the products $(x_\tau)_i \cdot (s_\tau)_i$ have the same value for all i.
- The system has a unique solution for every $\tau > 0$, provided that $\mathcal{F}^0$ is nonempty.
- As $\tau \to 0$, the conditions defining the points on the central path approximate the set of optimality conditions more and more closely.

The Central Path

Remark

- Instead of the complementary condition, we require the products $(x_\tau)_i \cdot (s_\tau)_i$ have the same value for all i.
- The system has a unique solution for every $\tau > 0$, provided that $\mathcal{F}^0$ is nonempty.
- As $\tau \to 0$, the conditions defining the points on the central path approximate the set of optimality conditions more and more closely.

If $\mathcal{F}^0$ is nonempty, $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ will converge to an optimal solution of the problem.

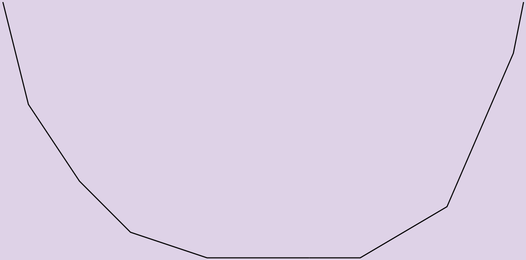
The Central Path

The Central Path

The Central Path

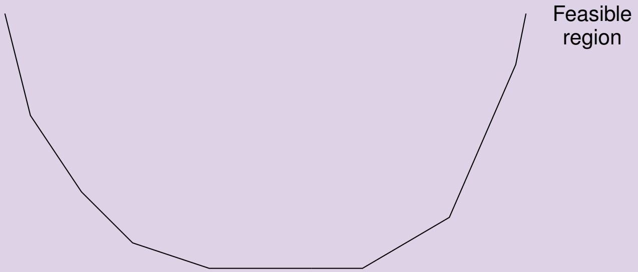
The Central Path

The Central Path



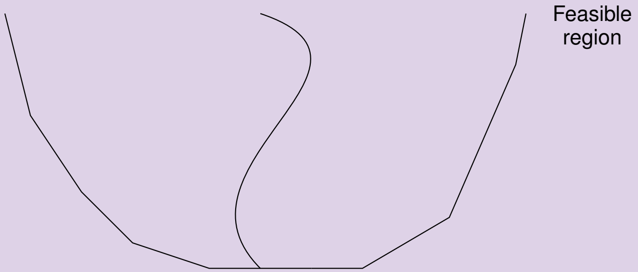
The Central Path

The Central Path



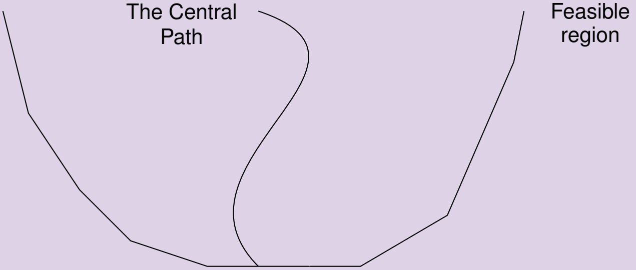
The Central Path

The Central Path



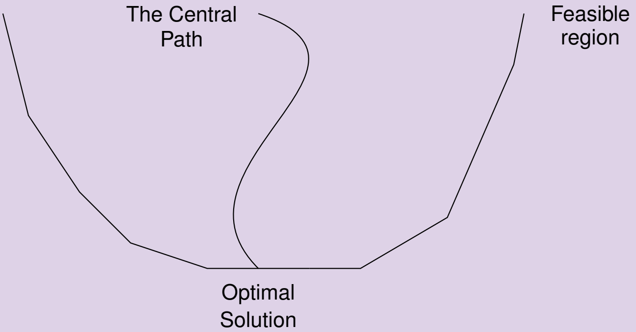
The Central Path

The Central Path



The Central Path

The Central Path



The Central Path

The Central Path

Exercise 6

The Central Path

Exercise 6

Recall the quadratic programming problem given in Exercise 4

The Central Path

Exercise 6

Recall the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution

The Central Path

Exercise 6

Recall the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $\mathbf{x^k} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$,

The Central Path

Exercise 6

Recall the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $\mathbf{x^k} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$, $\mathbf{y^k} = (1, \frac{1}{2})^T$,

The Central Path

Exercise 6

Recall the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $\mathbf{x^k} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$, $\mathbf{y^k} = (1, \frac{1}{2})^T$, and $\mathbf{s^k} = (\frac{3}{2}, \frac{11}{6}, \frac{10}{3})^T$.

The Central Path

Exercise 6

Recall the quadratic programming problem given in Exercise 4 and the current primal-dual estimate of the solution $\mathbf{x^k} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T$, $\mathbf{y^k} = (1, \frac{1}{2})^T$, and $\mathbf{s^k} = (\frac{3}{2}, \frac{11}{6}, \frac{10}{3})^T$. Verify that $(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k})$ is not on the central path.

IPMs with Centered Newton directions

IPMs with Centered Newton directions

Centered Newton directions

IPMs with Centered Newton directions

Centered Newton directions

To get over the weakness with pure Newton directions, most interior-point methods take a step toward points on the central path $\mathcal{C}$ corresponding to predetermined value of τ.

IPMs with Centered Newton directions

Centered Newton directions

To get over the weakness with pure Newton directions, most interior-point methods take a step toward points on the central path $\mathcal{C}$ corresponding to predetermined value of τ.

Since such directions are aiming for central points, they are called *centered directions*.

IPMs with Centered Newton directions

IPMs with Centered Newton directions

Description

IPMs with Centered Newton directions

Description

A centered direction is obtained by applying Newton update to the following system:

IPMs with Centered Newton directions

Description

A centered direction is obtained by applying Newton update to the following system:

$$\hat{\mathbf{F}}(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \begin{bmatrix} \mathbf{A}^{\mathsf{T}} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} - \mathbf{c} \\ \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \\ \mathbf{X} \cdot \mathbf{S} \cdot \mathbf{e} - \tau \cdot \mathbf{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

IPMs with Centered Newton directions

Description

A centered direction is obtained by applying Newton update to the following system:

$$\hat{\mathbf{F}}(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \begin{bmatrix} \mathbf{A}^{\mathbf{T}} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} - \mathbf{c} \\ \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \\ \mathbf{X} \cdot \mathbf{S} \cdot \mathbf{e} - \tau \cdot \mathbf{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Since the Jacobian of $\hat{F}$ is identical to the Jacobian of F,

IPMs with Centered Newton directions

Description

A centered direction is obtained by applying Newton update to the following system:

$$\hat{\mathbf{F}}(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \begin{bmatrix} \mathbf{A}^\mathbf{T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} - \mathbf{c} \\ \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \\ \mathbf{X} \cdot \mathbf{S} \cdot \mathbf{e} - \tau \cdot \mathbf{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Since the Jacobian of $\hat{F}$ is identical to the Jacobian of F, proceeding as the previous Newton equation,

IPMs with Centered Newton directions

Description

A centered direction is obtained by applying Newton update to the following system:

$$\hat{\mathbf{F}}(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \begin{bmatrix} \mathbf{A}^\mathsf{T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} - \mathbf{c} \\ \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \\ \mathbf{X} \cdot \mathbf{S} \cdot \mathbf{e} - \tau \cdot \mathbf{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Since the Jacobian of $\hat{F}$ is identical to the Jacobian of F, proceeding as the previous Newton equation, we obtain the following (modified) Newton equation for the centered direction:

IPMs with Centered Newton directions

Description

A centered direction is obtained by applying Newton update to the following system:

$$\hat{\mathbf{F}}(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \begin{bmatrix} \mathbf{A}^\mathbf{T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} - \mathbf{c} \\ \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \\ \mathbf{X} \cdot \mathbf{S} \cdot \mathbf{e} - \tau \cdot \mathbf{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Since the Jacobian of $\hat{F}$ is identical to the Jacobian of F, proceeding as the previous Newton equation, we obtain the following (modified) Newton equation for the centered direction:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^\mathbf{T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S}^\mathbf{k} & 0 & \mathbf{X}^\mathbf{k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_\mathbf{c}^\mathbf{k} \\ \Delta \mathbf{y}_\mathbf{c}^\mathbf{k} \\ \Delta \mathbf{s}_\mathbf{c}^\mathbf{k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau \cdot \mathbf{e} - \mathbf{X}^\mathbf{k} \cdot \mathbf{S}^\mathbf{k} \cdot \mathbf{e} \end{bmatrix}.$$

IPMs with Centered Newton directions

IPMs with Centered Newton directions

Description

IPMs with Centered Newton directions

Description

We introduce a *centering parameter* $\sigma \in [0, 1]$ and a duality gap (or average complementarity) μ defined by:

IPMs with Centered Newton directions

Description

We introduce a *centering parameter* $\sigma \in [0, 1]$ and a duality gap (or average complementarity) μ defined by:

$$\mu = \mu(\mathbf{x}, \mathbf{s}) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.$$

IPMs with Centered Newton directions

Description

We introduce a *centering parameter* $\sigma \in [0, 1]$ and a duality gap (or average complementarity) μ defined by:

$$\mu = \mu(\mathbf{x}, \mathbf{s}) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.$$

Note that, when $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ satisfy the conditions $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0$ and $\mathbf{A^T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}$, $\mathbf{s} \geq 0$, then $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ are optimal if and only if $\mu(\mathbf{x}, \mathbf{s}) = 0$.

IPMs with Centered Newton directions

Description

We introduce a *centering parameter* $\sigma \in [0, 1]$ and a duality gap (or average complementarity) μ defined by:

$$\mu = \mu(\mathbf{x}, \mathbf{s}) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.$$

Note that, when $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ satisfy the conditions $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0$ and $\mathbf{A^T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}$, $\mathbf{s} \geq 0$, then $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ are optimal if and only if $\mu(\mathbf{x}, \mathbf{s}) = 0$. If μ is large, then we are far away from the solution.

IPMs with Centered Newton directions

Description

We introduce a *centering parameter* $\sigma \in [0, 1]$ and a duality gap (or average complementarity) μ defined by:

$$\mu = \mu(\mathbf{x}, \mathbf{s}) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.$$

Note that, when $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ satisfy the conditions $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0$ and $\mathbf{A^T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}$, $\mathbf{s} \geq 0$, then $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ are optimal if and only if $\mu(\mathbf{x}, \mathbf{s}) = 0$. If μ is large, then we are far away from the solution. Therefore, μ serves as a measure of optimality for feasible points

IPMs with Centered Newton directions

Description

We introduce a *centering parameter* $\sigma \in [0, 1]$ and a duality gap (or average complementarity) μ defined by:

$$\mu = \mu(\mathbf{x}, \mathbf{s}) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.$$

Note that, when $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ satisfy the conditions $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0$ and $\mathbf{A^T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}$, $\mathbf{s} \geq 0$, then $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ are optimal if and only if $\mu(\mathbf{x}, \mathbf{s}) = 0$. If μ is large, then we are far away from the solution. Therefore, μ serves as a measure of optimality for feasible points – the smaller the duality gap, the closer the point of optimality.

IPMs with Centered Newton directions

Description

We introduce a *centering parameter* $\sigma \in [0, 1]$ and a duality gap (or average complementarity) μ defined by:

$$\mu = \mu(\mathbf{x}, \mathbf{s}) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.$$

Note that, when $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ satisfy the conditions $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0$ and $\mathbf{A}^\mathbf{T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}$, $\mathbf{s} \geq 0$, then $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ are optimal if and only if $\mu(\mathbf{x}, \mathbf{s}) = 0$. If μ is large, then we are far away from the solution. Therefore, μ serves as a measure of optimality for feasible points – the smaller the duality gap, the closer the point of optimality.

For a central point $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ we have

IPMs with Centered Newton directions

Description

We introduce a *centering parameter* $\sigma \in [0, 1]$ and a duality gap (or average complementarity) μ defined by:

$$\mu = \mu(\mathbf{x}, \mathbf{s}) := \frac{\sum_{i=1}^{n} x_i \cdot s_i}{n} = \frac{x^T \cdot s}{n}.$$

Note that, when $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ satisfy the conditions $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0$ and $\mathbf{A}^\mathbf{T} \cdot \mathbf{y} - \mathbf{Q} \cdot \mathbf{x} + \mathbf{s} = \mathbf{c}$, $\mathbf{s} \geq 0$, then $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ are optimal if and only if $\mu(\mathbf{x}, \mathbf{s}) = 0$. If μ is large, then we are far away from the solution. Therefore, μ serves as a measure of optimality for feasible points – the smaller the duality gap, the closer the point of optimality.

For a central point $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ we have

$$\mu(\mathbf{x}_\tau, \mathbf{s}_\tau) = \frac{\sum_{i=1}^{n} (x_\tau)_i \cdot (s_\tau)_i}{n} = \frac{\sum_{i=1}^{n} \tau}{n} = \tau.$$

IPMs with Centered Newton directions

IPMs with Centered Newton directions

Description

IPMs with Centered Newton directions

Description

Using a simple change in notation, the centered direction can now be describe as the solution of the following system:

IPMs with Centered Newton directions

Description

Using a simple change in notation, the centered direction can now be describe as the solution of the following system:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^\mathbf{T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S}^\mathbf{k} & 0 & \mathbf{X}^\mathbf{k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_\mathbf{c}^\mathbf{k} \\ \Delta \mathbf{y}_\mathbf{c}^\mathbf{k} \\ \Delta \mathbf{s}_\mathbf{c}^\mathbf{k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma^\mathbf{k} \cdot \mathbf{mu}^\mathbf{k} \cdot \mathbf{e} - \mathbf{X}^\mathbf{k} \cdot \mathbf{S}^\mathbf{k} \cdot \mathbf{e} \end{bmatrix},$$

IPMs with Centered Newton directions

Description

Using a simple change in notation, the centered direction can now be describe as the solution of the following system:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x_c^k} \\ \Delta \mathbf{y_c^k} \\ \Delta \mathbf{s_c^k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma^{\mathbf{k}} \cdot \mathbf{mu^k} \cdot \mathbf{e} - \mathbf{X^k} \cdot \mathbf{S^k} \cdot \mathbf{e} \end{bmatrix},$$

where $\mu^k := \mu(\mathbf{x^k}, \mathbf{s^k}) = \frac{(\mathbf{x^k})^T \cdot \mathbf{s^k}}{n}$ and $\sigma^k \in [0, 1]$ is a user defined quantity describing the ratio of the duality gap at the target central point and the current point.

IPMs with Centered Newton directions

Description

Using a simple change in notation, the centered direction can now be describe as the solution of the following system:

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x_c^k} \\ \Delta \mathbf{y_c^k} \\ \Delta \mathbf{s_c^k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma^{\mathbf{k}} \cdot \mathbf{mu^k} \cdot \mathbf{e} - \mathbf{X^k} \cdot \mathbf{S^k} \cdot \mathbf{e} \end{bmatrix},$$

where $\mu^k := \mu(\mathbf{x^k}, \mathbf{s^k}) = \frac{(\mathbf{x^k})^T \cdot \mathbf{s^k}}{n}$ and $\sigma^k \in [0, 1]$ is a user defined quantity describing the ratio of the duality gap at the target central point and the current point.
When $\sigma^k = 1$, we have a *pure centering direction*.

IPMs with Centered Newton directions

IPMs with Centered Newton directions

Remark

IPMs with Centered Newton directions

Remark

We have three ways to choose to target a central point:

IPMs with Centered Newton directions

Remark

We have three ways to choose to target a central point:

- At a lower level than our current point ($\tau < \mu(\mathbf{x}, \mathbf{s})$);

IPMs with Centered Newton directions

Remark

We have three ways to choose to target a central point:

- At a lower level than our current point ($\tau < \mu(\mathbf{x}, \mathbf{s})$);
- At the same level as our current point ($\tau = \mu(\mathbf{x}, \mathbf{s})$);

IPMs with Centered Newton directions

Remark

We have three ways to choose to target a central point:

- At a lower level than our current point ($\tau < \mu(\mathbf{x}, \mathbf{s})$);
- At the same level as our current point ($\tau = \mu(\mathbf{x}, \mathbf{s})$);
- At a higher level than our current point ($\tau > \mu(\mathbf{x}, \mathbf{s})$).

IPMs with Centered Newton directions

Remark

We have three ways to choose to target a central point:

- At a lower level than our current point ($\tau < \mu(\mathbf{x}, \mathbf{s})$);
- At the same level as our current point ($\tau = \mu(\mathbf{x}, \mathbf{s})$);
- At a higher level than our current point ($\tau > \mu(\mathbf{x}, \mathbf{s})$).

In most circumstances, the third option is not a good choice as it targets a central point that is ”farther” than the current iterate to the optimal solution.

IPMs with Centered Newton directions

Remark

We have three ways to choose to target a central point:

- At a lower level than our current point ($\tau < \mu(\mathbf{x}, \mathbf{s})$);
- At the same level as our current point ($\tau = \mu(\mathbf{x}, \mathbf{s})$);
- At a higher level than our current point ($\tau > \mu(\mathbf{x}, \mathbf{s})$).

In most circumstances, the third option is not a good choice as it targets a central point that is "farther" than the current iterate to the optimal solution.

Therefore, we will always choose $\tau \leq \mu(\mathbf{x}, \mathbf{s})$ in defining centered directions.

Generic Interior Point Algorithm

Generic Interior Point Algorithm

General Interior Point Algorithm I

With these basic concepts in hand, we can define a general primal-dual interior point algorithm.

Generic Interior Point Algorithm

General Interior Point Algorithm I

With these basic concepts in hand, we can define a general primal-dual interior point algorithm.

- Choose $(\mathbf{x^0}, \mathbf{y^0}, \mathbf{s^0}) \in \mathcal{F}^0$. For $k = 0, 1, 2, \ldots$ repeat the following steps.

Generic Interior Point Algorithm

Generic Interior Point Algorithm

General Interior Point Algorithm II

Generic Interior Point Algorithm

General Interior Point Algorithm II

- Choose $\sigma^k \in [0,1]$, let $\mu^{\mathbf{k}} = \frac{(\mathbf{x^k})^T \cdot \mathbf{s^k}}{n}$. Solve

Generic Interior Point Algorithm

General Interior Point Algorithm II

- Choose $\sigma^k \in [0,1]$, let $\mu^{\mathbf{k}} = \frac{(\mathbf{x}^{\mathbf{k}})^T \cdot \mathbf{s}^{\mathbf{k}}}{n}$. Solve

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A}^{\mathbf{T}} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S}^{\mathbf{k}} & 0 & \mathbf{X}^{\mathbf{k}} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_{\mathbf{c}}^{\mathbf{k}} \\ \Delta \mathbf{y}_{\mathbf{c}}^{\mathbf{k}} \\ \Delta \mathbf{s}_{\mathbf{c}}^{\mathbf{k}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma^{\mathbf{k}} \cdot \mu^{\mathbf{k}} \cdot \mathbf{e} - \mathbf{X}^{\mathbf{k}} \cdot \mathbf{S}^{\mathbf{k}} \cdot \mathbf{e} \end{bmatrix}.$$

Generic Interior Point Algorithm

General Interior Point Algorithm II

- Choose $\sigma^k \in [0, 1]$, let $\mu^{\mathbf{k}} = \frac{(\mathbf{x^k})^T \cdot \mathbf{s^k}}{n}$. Solve

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x_c^k} \\ \Delta \mathbf{y_c^k} \\ \Delta \mathbf{s_c^k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma^{\mathbf{k}} \cdot \mu^{\mathbf{k}} \cdot \mathbf{e} - \mathbf{X^k} \cdot \mathbf{S^k} \cdot \mathbf{e} \end{bmatrix}.$$

- Choose α^k such that

Generic Interior Point Algorithm

General Interior Point Algorithm II

- Choose $\sigma^k \in [0, 1]$, let $\mu^{\mathbf{k}} = \frac{(\mathbf{x^k})^T \cdot \mathbf{s^k}}{n}$. Solve

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x_c^k} \\ \Delta \mathbf{y_c^k} \\ \Delta \mathbf{s_c^k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma^{\mathbf{k}} \cdot \mu^{\mathbf{k}} \cdot \mathbf{e} - \mathbf{X^k} \cdot \mathbf{S^k} \cdot \mathbf{e} \end{bmatrix}.$$

- Choose α^k such that

$$\mathbf{x^k} + \alpha^k \cdot \Delta \mathbf{x^k} > 0, \mathbf{s^k} + \alpha^k \cdot \Delta \mathbf{s^k} > 0.$$

Generic Interior Point Algorithm

General Interior Point Algorithm II

- Choose $\sigma^k \in [0, 1]$, let $\mu^{\mathbf{k}} = \frac{(\mathbf{x^k})^T \cdot \mathbf{s^k}}{n}$. Solve

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x_c^k} \\ \Delta \mathbf{y_c^k} \\ \Delta \mathbf{s_c^k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma^{\mathbf{k}} \cdot \mu^{\mathbf{k}} \cdot \mathbf{e} - \mathbf{X^k} \cdot \mathbf{S^k} \cdot \mathbf{e} \end{bmatrix}.$$

- Choose α^k such that

$$\mathbf{x^k} + \alpha^k \cdot \Delta \mathbf{x^k} > 0, \mathbf{s^k} + \alpha^k \cdot \Delta \mathbf{s^k} > 0.$$

Set

Generic Interior Point Algorithm

General Interior Point Algorithm II

- Choose $\sigma^k \in [0,1]$, let $\mu^{\mathbf{k}} = \frac{(\mathbf{x^k})^T \cdot \mathbf{s^k}}{n}$. Solve

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x_c^k} \\ \Delta \mathbf{y_c^k} \\ \Delta \mathbf{s_c^k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma^{\mathbf{k}} \cdot \mu^{\mathbf{k}} \cdot \mathbf{e} - \mathbf{X^k} \cdot \mathbf{S^k} \cdot \mathbf{e} \end{bmatrix}.$$

- Choose α^k such that

$$\mathbf{x^k} + \alpha^k \cdot \Delta \mathbf{x^k} > 0, \mathbf{s^k} + \alpha^k \cdot \Delta \mathbf{s^k} > 0.$$

  Set

$$(\mathbf{x^{k+1}}, \mathbf{y^{k+1}}, \mathbf{s^{k+1}}) = (\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k}) + \alpha^k \cdot (\Delta \mathbf{x^k}, \Delta \mathbf{y^k}, \Delta \mathbf{s^k}),$$

Generic Interior Point Algorithm

General Interior Point Algorithm II

- Choose $\sigma^k \in [0, 1]$, let $\mu^\mathbf{k} = \frac{(\mathbf{x^k})^T \cdot \mathbf{s^k}}{n}$. Solve

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x_c^k} \\ \Delta \mathbf{y_c^k} \\ \Delta \mathbf{s_c^k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma^\mathbf{k} \cdot \mu^\mathbf{k} \cdot \mathbf{e} - \mathbf{X^k} \cdot \mathbf{S^k} \cdot \mathbf{e} \end{bmatrix}.$$

- Choose α^k such that

$$\mathbf{x^k} + \alpha^k \cdot \Delta \mathbf{x^k} > 0, \mathbf{s^k} + \alpha^k \cdot \Delta \mathbf{s^k} > 0.$$

  Set

$$(\mathbf{x^{k+1}}, \mathbf{y^{k+1}}, \mathbf{s^{k+1}}) = (\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k}) + \alpha^k \cdot (\Delta \mathbf{x^k}, \Delta \mathbf{y^k}, \Delta \mathbf{s^k}),$$

  and $k := k + 1$.

Starting From an Infeasible Point

Starting From an Infeasible Point

Motivation

The generic interior point method starts with a strictly feasible iterate.

Starting From an Infeasible Point

Motivation

The generic interior point method starts with a strictly feasible iterate.

It is not practical since finding such a starting point is not always a trivial task.

Starting From an Infeasible Point

Motivation

The generic interior point method starts with a strictly feasible iterate.

It is not practical since finding such a starting point is not always a trivial task.

Fortunately, however, we can accommodate infeasible starting points with a small modification of the linear system we solve in each iteration.

Starting From an Infeasible Point

Starting From an Infeasible Point

Modification

- We only require that the initial point $(\mathbf{x^0}, \mathbf{y^0}, \mathbf{s^0})$ satisfy the nonnegativity restrictions strictly: $\mathbf{x^0} > 0$ and $\mathbf{s^0} > 0$.

Starting From an Infeasible Point

Modification

- We only require that the initial point $(\mathbf{x^0}, \mathbf{y^0}, \mathbf{s^0})$ satisfy the nonnegativity restrictions strictly: $\mathbf{x^0} > 0$ and $\mathbf{s^0} > 0$.
- Then the Newton equation from an infeasible point $(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k})$ is reduced to

Starting From an Infeasible Point

Modification

- We only require that the initial point $(\mathbf{x^0}, \mathbf{y^0}, \mathbf{s^0})$ satisfy the nonnegativity restrictions strictly: $\mathbf{x^0} > 0$ and $\mathbf{s^0} > 0$.
- Then the Newton equation from an infeasible point $(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k})$ is reduced to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x^k} \\ \Delta\mathbf{y^k} \\ \Delta\mathbf{s^k} \end{bmatrix} = \begin{bmatrix} \mathbf{c} + \mathbf{Q} \cdot \mathbf{x^k} - \mathbf{A^T} \cdot \mathbf{y^k} - \mathbf{s^k} \\ \mathbf{b} - \mathbf{A} \cdot \mathbf{x^k} \\ \tau \cdot \mathbf{e} - \mathbf{X^k} \cdot \mathbf{S^k} \cdot \mathbf{e} \end{bmatrix}$$

Starting From an Infeasible Point

Modification

- We only require that the initial point $(\mathbf{x^0}, \mathbf{y^0}, \mathbf{s^0})$ satisfy the nonnegativity restrictions strictly: $\mathbf{x^0} > 0$ and $\mathbf{s^0} > 0$.
- Then the Newton equation from an infeasible point $(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k})$ is reduced to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x^k} \\ \Delta\mathbf{y^k} \\ \Delta\mathbf{s^k} \end{bmatrix} = \begin{bmatrix} \mathbf{c} + \mathbf{Q} \cdot \mathbf{x^k} - \mathbf{A^T} \cdot \mathbf{y^k} - \mathbf{s^k} \\ \mathbf{b} - \mathbf{A} \cdot \mathbf{x^k} \\ \tau \cdot \mathbf{e} - \mathbf{X^k} \cdot \mathbf{S^k} \cdot \mathbf{e} \end{bmatrix}$$

  We no longer have zeros in the first and second blocks of the right-hand-side vector since we are not assuming that the iterates satisfy

Starting From an Infeasible Point

Modification

- We only require that the initial point $(\mathbf{x^0}, \mathbf{y^0}, \mathbf{s^0})$ satisfy the nonnegativity restrictions strictly: $\mathbf{x^0} > 0$ and $\mathbf{s^0} > 0$.
- Then the Newton equation from an infeasible point $(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k})$ is reduced to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x^k} \\ \Delta \mathbf{y^k} \\ \Delta \mathbf{s^k} \end{bmatrix} = \begin{bmatrix} \mathbf{c} + \mathbf{Q} \cdot \mathbf{x^k} - \mathbf{A^T} \cdot \mathbf{y^k} - \mathbf{s^k} \\ \mathbf{b} - \mathbf{A} \cdot \mathbf{x^k} \\ \tau \cdot \mathbf{e} - \mathbf{X^k} \cdot \mathbf{S^k} \cdot \mathbf{e} \end{bmatrix}$$

We no longer have zeros in the first and second blocks of the right-hand-side vector since we are not assuming that the iterates satisfy $A \cdot x^k = b$

Starting From an Infeasible Point

Modification

- We only require that the initial point $(\mathbf{x^0}, \mathbf{y^0}, \mathbf{s^0})$ satisfy the nonnegativity restrictions strictly: $\mathbf{x^0} > 0$ and $\mathbf{s^0} > 0$.
- Then the Newton equation from an infeasible point $(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k})$ is reduced to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x^k} \\ \Delta \mathbf{y^k} \\ \Delta \mathbf{s^k} \end{bmatrix} = \begin{bmatrix} \mathbf{c} + \mathbf{Q} \cdot \mathbf{x^k} - \mathbf{A^T} \cdot \mathbf{y^k} - \mathbf{s^k} \\ \mathbf{b} - \mathbf{A} \cdot \mathbf{x^k} \\ \tau \cdot \mathbf{e} - \mathbf{X^k} \cdot \mathbf{S^k} \cdot \mathbf{e} \end{bmatrix}$$

We no longer have zeros in the first and second blocks of the right-hand-side vector since we are not assuming that the iterates satisfy $A \cdot x^k = b$ and $A^T \cdot y^k - Q \cdot x^k + s^k = c$.

Starting From an Infeasible Point

Modification

- We only require that the initial point $(\mathbf{x^0}, \mathbf{y^0}, \mathbf{s^0})$ satisfy the nonnegativity restrictions strictly: $\mathbf{x^0} > 0$ and $\mathbf{s^0} > 0$.
- Then the Newton equation from an infeasible point $(\mathbf{x^k}, \mathbf{y^k}, \mathbf{s^k})$ is reduced to

$$\begin{bmatrix} -\mathbf{Q} & \mathbf{A^T} & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{S^k} & 0 & \mathbf{X^k} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x^k} \\ \Delta\mathbf{y^k} \\ \Delta\mathbf{s^k} \end{bmatrix} = \begin{bmatrix} \mathbf{c} + \mathbf{Q} \cdot \mathbf{x^k} - \mathbf{A^T} \cdot \mathbf{y^k} - \mathbf{s^k} \\ \mathbf{b} - \mathbf{A} \cdot \mathbf{x^k} \\ \tau \cdot \mathbf{e} - \mathbf{X^k} \cdot \mathbf{S^k} \cdot \mathbf{e} \end{bmatrix}$$

  We no longer have zeros in the first and second blocks of the right-hand-side vector since we are not assuming that the iterates satisfy $A \cdot x^k = b$ and $A^T \cdot y^k - Q \cdot x^k + s^k = c$.
- Replacing the linear system in this case, the algorithms can work simultaneously.

Solving Motivating Example 2

Solving Motivating Example 2

Optimization model for Example 2: Quantifying the notion of “risk”

- Markowitz, in his Nobel prize winning work,

Solving Motivating Example 2

Optimization model for Example 2: Quantifying the notion of “risk”

- Markowitz, in his Nobel prize winning work, showed that a rational investor’s notion of minimizing risk can be closely approximated

Solving Motivating Example 2

Optimization model for Example 2: Quantifying the notion of "risk"

- Markowitz, in his Nobel prize winning work, showed that a rational investor's notion of minimizing risk can be closely approximated by minimizing the variance of the return of the investment portfolio.

Solving Motivating Example 2

Solving Motivating Example 2

Example

Solving Motivating Example 2

Example

- Recall that:

Solving Motivating Example 2

Example

- Recall that:
- We wish to invest $1000.00 in stocks A, B, and C for a one month period.

Solving Motivating Example 2

Example

- Recall that:
- We wish to invest $1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month,

Solving Motivating Example 2

Example

- Recall that:
- We wish to invest $1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.

Solving Motivating Example 2

Example

- Recall that:
- We wish to invest \$1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.
- The rate of return of each stock is a random variable with some expected value.

Solving Motivating Example 2

Example

- Recall that:
- We wish to invest \$1000.00 in stocks A, B, and C for a one month period.
- We buy a stock at some dollar amount per share in the beginning of the month, and sell it at some dollar amount per share at the end of the month.
- The rate of return of each stock is a random variable with some expected value.
- Our goal is to invest in such a way that the expected end-of-month return is at least \$50.00 or 5%.

Solving Motivating Example 2

Solving Motivating Example 2

Optimization model for Example 2: The decision variables

Solving Motivating Example 2

Optimization model for Example 2: The decision variables

- Our decision variables are $x_i, i = 1, 2, 3$, denoting the dollars invested in stock i.

Solving Motivating Example 2

Optimization model for Example 2: The decision variables

- Our decision variables are $x_i, i = 1, 2, 3$, denoting the dollars invested in stock i.
- Since we have a total of \$1000.00 to invest, then the x_i's should satisfy:

Solving Motivating Example 2

Optimization model for Example 2: The decision variables

- Our decision variables are $x_i, i = 1, 2, 3$, denoting the dollars invested in stock i.
- Since we have a total of $1000.00 to invest, then the x_i's should satisfy:

$$\sum_{i=1}^{3} x_i \leq 1000$$

Solving Motivating Example 2

Optimization model for Example 2: The decision variables

- Our decision variables are $x_i, i = 1, 2, 3$, denoting the dollars invested in stock i.
- Since we have a total of \$1000.00 to invest, then the x_i's should satisfy:

$$\sum_{i=1}^{3} x_i \leq 1000$$
$$x_i \geq 0, i = 1, 2, 3$$

Solving Motivating Example 2

Solving Motivating Example 2

Optimization model for Example 2: Covariance matrix

- Let **Q** denote the covariance matrix of rates of stock returns.

Solving Motivating Example 2

Optimization model for Example 2: Covariance matrix

- Let **Q** denote the covariance matrix of rates of stock returns.
- The classical mean-variance model consists of minimizing portfolio risk, as measured by

Solving Motivating Example 2

Optimization model for Example 2: Covariance matrix

- Let $\mathbf{Q}$ denote the covariance matrix of rates of stock returns.
- The classical mean-variance model consists of minimizing portfolio risk, as measured by

$$\frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x},$$

Solving Motivating Example 2

Optimization model for Example 2: Covariance matrix

- Let **Q** denote the covariance matrix of rates of stock returns.
- The classical mean-variance model consists of minimizing portfolio risk, as measured by

$$\frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x},$$

and subject to a set of constraints.

Solving Motivating Example 2

Optimization model for Example 2: Covariance matrix

- Let $\mathbf{Q}$ denote the covariance matrix of rates of stock returns.
- The classical mean-variance model consists of minimizing portfolio risk, as measured by

$$\frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x},$$

  and subject to a set of constraints.
- Since we want to have an expected return of at least \$50.00, then

Solving Motivating Example 2

Optimization model for Example 2: Covariance matrix

- Let **Q** denote the covariance matrix of rates of stock returns.
- The classical mean-variance model consists of minimizing portfolio risk, as measured by

$$\frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x},$$

  and subject to a set of constraints.
- Since we want to have an expected return of at least \$50.00, then

$$\sum_{i=1}^{3} \bar{r}_i \cdot x_i \geq 50$$

Solving Motivating Example 2

Optimization model for Example 2: Covariance matrix

- Let **Q** denote the covariance matrix of rates of stock returns.
- The classical mean-variance model consists of minimizing portfolio risk, as measured by

$$\frac{1}{2} \cdot \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x},$$

  and subject to a set of constraints.
- Since we want to have an expected return of at least \$50.00, then

$$\sum_{i=1}^{3} \bar{r}_i \cdot x_i \geq 50$$

  Where $\bar{r}_i$ is the expected value of the random variable corresponding to the monthly return per dollar for stock i.

Solving Motivating Example 2

Solving Motivating Example 2

Optimization model for Example 2

Solving Motivating Example 2

Optimization model for Example 2

- Using matrices and vectors, our optimization model can be compactly stated as follows:

Solving Motivating Example 2

Optimization model for Example 2

- Using matrices and vectors, our optimization model can be compactly stated as follows:

$$\min \quad \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x}$$

Solving Motivating Example 2

Optimization model for Example 2

- Using matrices and vectors, our optimization model can be compactly stated as follows:

$$\begin{aligned} \min \quad \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} & \\ \mathbf{e}^T \cdot \mathbf{x} & \leq 1000 \end{aligned}$$

Solving Motivating Example 2

Optimization model for Example 2

- Using matrices and vectors, our optimization model can be compactly stated as follows:

$$\begin{array}{rrcl} \min & \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} & & \\ & \mathbf{e}^T \cdot \mathbf{x} & \leq & 1000 \\ & \bar{\mathbf{r}}^T \cdot \mathbf{x} & \geq & 50 \end{array}$$

Solving Motivating Example 2

Optimization model for Example 2

- Using matrices and vectors, our optimization model can be compactly stated as follows:

$$\begin{array}{rrcl} \min & \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} & & \\ & \mathbf{e}^T \cdot \mathbf{x} & \leq & 1000 \\ & \bar{\mathbf{r}}^T \cdot \mathbf{x} & \geq & 50 \\ & \mathbf{x} & \geq & \mathbf{0} \end{array}$$

Solving Motivating Example 2

Optimization model for Example 2

- Using matrices and vectors, our optimization model can be compactly stated as follows:

$$\begin{aligned} \min \quad & \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} \\ & \mathbf{e}^T \cdot \mathbf{x} \leq 1000 \\ & \bar{\mathbf{r}}^T \cdot \mathbf{x} \geq 50 \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Where **x** is the decision vector of size n (n is the number of stocks, $n = 3$ in our example),

Solving Motivating Example 2

Optimization model for Example 2

- Using matrices and vectors, our optimization model can be compactly stated as follows:

$$\begin{array}{rrcl} \min & \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} & & \\ & \mathbf{e}^T \cdot \mathbf{x} & \leq & 1000 \\ & \bar{\mathbf{r}}^T \cdot \mathbf{x} & \geq & 50 \\ & \mathbf{x} & \geq & \mathbf{0} \end{array}$$

Where **x** is the decision vector of size n (n is the number of stocks, $n = 3$ in our example), **e** is an n-vector of ones,

Solving Motivating Example 2

Optimization model for Example 2

- Using matrices and vectors, our optimization model can be compactly stated as follows:

$$\begin{array}{rrcl} \min & \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} & & \\ & \mathbf{e}^T \cdot \mathbf{x} & \leq & 1000 \\ & \bar{\mathbf{r}}^T \cdot \mathbf{x} & \geq & 50 \\ & \mathbf{x} & \geq & \mathbf{0} \end{array}$$

Where $\mathbf{x}$ is the decision vector of size n (n is the number of stocks, $n = 3$ in our example), $\mathbf{e}$ is an n-vector of ones, $\bar{\mathbf{r}}$ is the n-vector of expected returns of the stocks, and

Solving Motivating Example 2

Optimization model for Example 2

- Using matrices and vectors, our optimization model can be compactly stated as follows:

$$
\begin{array}{rrcl}
\min & \mathbf{x}^T \cdot \mathbf{Q} \cdot \mathbf{x} & & \\
 & \mathbf{e}^T \cdot \mathbf{x} & \leq & 1000 \\
 & \bar{\mathbf{r}}^T \cdot \mathbf{x} & \geq & 50 \\
 & \mathbf{x} & \geq & \mathbf{0}
\end{array}
$$

  Where $\mathbf{x}$ is the decision vector of size n (n is the number of stocks, $n = 3$ in our example), $\mathbf{e}$ is an n-vector of ones, $\bar{\mathbf{r}}$ is the n-vector of expected returns of the stocks, and $\mathbf{Q}$ is the $n \times n$ covariance matrix.

Using MATLAB and Optimization Toolbox Function `quadprog`

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving our motivating example using MATLAB

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving our motivating example using MATLAB

- For our example, suppose that

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving our motivating example using MATLAB

- For our example, suppose that

$$\mathbf{Q} = \begin{bmatrix} 0.0171 & 0.0033 & 0.0012 \\ 0.0033 & 0.0059 & 0.0045 \\ 0.0012 & 0.0045 & 0.0630 \end{bmatrix}$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving our motivating example using MATLAB

- For our example, suppose that

$$\mathbf{Q} = \begin{bmatrix} 0.0171 & 0.0033 & 0.0012 \\ 0.0033 & 0.0059 & 0.0045 \\ 0.0012 & 0.0045 & 0.0630 \end{bmatrix}, \text{ and}$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving our motivating example using MATLAB

- For our example, suppose that

$$\mathbf{Q} = \begin{bmatrix} 0.0171 & 0.0033 & 0.0012 \\ 0.0033 & 0.0059 & 0.0045 \\ 0.0012 & 0.0045 & 0.0630 \end{bmatrix}, \text{ and } \quad \bar{\mathbf{r}} = \begin{bmatrix} 0.026 \\ 0.008 \\ 0.074 \end{bmatrix}$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving our motivating example using MATLAB

- For our example, suppose that

$$\mathbf{Q} = \begin{bmatrix} 0.0171 & 0.0033 & 0.0012 \\ 0.0033 & 0.0059 & 0.0045 \\ 0.0012 & 0.0045 & 0.0630 \end{bmatrix}, \text{ and } \quad \bar{\mathbf{r}} = \begin{bmatrix} 0.026 \\ 0.008 \\ 0.074 \end{bmatrix}$$

- So,

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ -0.026 & -0.008 & -0.074 \end{bmatrix}$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving our motivating example using MATLAB

- For our example, suppose that

$$\mathbf{Q} = \begin{bmatrix} 0.0171 & 0.0033 & 0.0012 \\ 0.0033 & 0.0059 & 0.0045 \\ 0.0012 & 0.0045 & 0.0630 \end{bmatrix}, \text{ and } \quad \bar{\mathbf{r}} = \begin{bmatrix} 0.026 \\ 0.008 \\ 0.074 \end{bmatrix}$$

- So,

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ -0.026 & -0.008 & -0.074 \end{bmatrix}, \text{ and}$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving our motivating example using MATLAB

- For our example, suppose that

$$\mathbf{Q} = \begin{bmatrix} 0.0171 & 0.0033 & 0.0012 \\ 0.0033 & 0.0059 & 0.0045 \\ 0.0012 & 0.0045 & 0.0630 \end{bmatrix}, \text{ and } \quad \bar{\mathbf{r}} = \begin{bmatrix} 0.026 \\ 0.008 \\ 0.074 \end{bmatrix}$$

- So,

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ -0.026 & -0.008 & -0.074 \end{bmatrix}, \text{ and } \quad \mathbf{b} = \begin{bmatrix} 1000 \\ -50 \end{bmatrix}$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving our motivating example using MATLAB

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving our motivating example using MATLAB

- We find that our optimal solution is

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving our motivating example using MATLAB

- We find that our optimal solution is $\mathbf{x}^* = \begin{bmatrix} 500 \\ 0 \\ 500 \end{bmatrix}$.

Using MATLAB and Optimization Toolbox Function `quadprog`

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving another example using MATLAB

$$\min_{x} \quad \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving another example using MATLAB

$$\begin{aligned} \min_{x} \quad & \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\ & x_1 + 3 \cdot x_2 \geq 15 \end{aligned}$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving another example using MATLAB

$$\min_{x} \quad \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2$$
$$x_1 + 3 \cdot x_2 \quad \geq \quad 15$$
$$2 \cdot x_1 + 5 \cdot x_2 \quad \leq \quad 100$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving another example using MATLAB

$$
\begin{aligned}
\min_x \quad & \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\
& x_1 + 3 \cdot x_2 \geq 15 \\
& 2 \cdot x_1 + 5 \cdot x_2 \leq 100 \\
& 3 \cdot x_1 + 4 \cdot x_2 \leq 80
\end{aligned}
$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving another example using MATLAB

$$
\begin{aligned}
\min_{x} \quad & \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 & & \\
& x_1 + 3 \cdot x_2 & \geq & \quad 15 \\
& 2 \cdot x_1 + 5 \cdot x_2 & \leq & \quad 100 \\
& 3 \cdot x_1 + 4 \cdot x_2 & \leq & \quad 80 \\
& x_1, x_2 & \geq & \quad 0
\end{aligned}
$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving another example using MATLAB

$$
\begin{aligned}
\min_{x} \quad & \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\
x_1 + 3 \cdot x_2 & \geq 15 \\
2 \cdot x_1 + 5 \cdot x_2 & \leq 100 \\
3 \cdot x_1 + 4 \cdot x_2 & \leq 80 \\
x_1, x_2 & \geq 0
\end{aligned}
$$

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving another example using MATLAB

$$\begin{array}{rcl} \min_{x} \quad \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 & & \\ x_1 + 3 \cdot x_2 & \geq & 15 \\ 2 \cdot x_1 + 5 \cdot x_2 & \leq & 100 \\ 3 \cdot x_1 + 4 \cdot x_2 & \leq & 80 \\ x_1, x_2 & \geq & 0 \end{array}$$

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 3 \\ 4 \end{bmatrix},$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving another example using MATLAB

$$\begin{aligned} \min_x \quad & \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 & & \\ & x_1 + 3 \cdot x_2 & \geq & \quad 15 \\ & 2 \cdot x_1 + 5 \cdot x_2 & \leq & \quad 100 \\ & 3 \cdot x_1 + 4 \cdot x_2 & \leq & \quad 80 \\ & x_1, x_2 & \geq & \quad 0 \end{aligned}$$

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} -1 & -3 \\ 2 & 5 \\ 3 & 4 \end{bmatrix},$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving another example using MATLAB

$$\begin{aligned} \min_x \quad & \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\ x_1 + 3 \cdot x_2 &\geq 15 \\ 2 \cdot x_1 + 5 \cdot x_2 &\leq 100 \\ 3 \cdot x_1 + 4 \cdot x_2 &\leq 80 \\ x_1, x_2 &\geq 0 \end{aligned}$$

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} -1 & -3 \\ 2 & 5 \\ 3 & 4 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -15 \\ 100 \\ 80 \end{bmatrix}$$

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving another example using MATLAB

$$
\begin{aligned}
\min_{x} \quad & \frac{1}{2} \cdot x_1^2 + 3 \cdot x_1 + 4 \cdot x_2 \\
x_1 + 3 \cdot x_2 &\geq 15 \\
2 \cdot x_1 + 5 \cdot x_2 &\leq 100 \\
3 \cdot x_1 + 4 \cdot x_2 &\leq 80 \\
x_1, x_2 &\geq 0
\end{aligned}
$$

$$
\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} -1 & -3 \\ 2 & 5 \\ 3 & 4 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -15 \\ 100 \\ 80 \end{bmatrix}
$$

We find that our optimal solution is

Using MATLAB and Optimization Toolbox Function `quadprog`

Solving another example using MATLAB

$$\begin{aligned}\min_{x} \quad \frac{1}{2}\cdot x_1^2 + 3\cdot x_1 + 4\cdot x_2 & \\ x_1 + 3\cdot x_2 &\geq 15 \\ 2\cdot x_1 + 5\cdot x_2 &\leq 100 \\ 3\cdot x_1 + 4\cdot x_2 &\leq 80 \\ x_1, x_2 &\geq 0\end{aligned}$$

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} -1 & -3 \\ 2 & 5 \\ 3 & 4 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -15 \\ 100 \\ 80 \end{bmatrix}$$

We find that our optimal solution is $\mathbf{x^*} = (0, 5)$.

References

References

References

References

References

1. Stephen J. Wright, Primal-Dual Interior-Point Methods, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

References

References

1. Stephen J. Wright, Primal-Dual Interior-Point Methods, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.
2. G. Cornuejols, R. Tütüncü, Optimization methods in Finance, Cambridge University Press, 2007.

References

References

1. Stephen J. Wright, Primal-Dual Interior-Point Methods, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.
2. G. Cornuejols, R. Tütüncü, Optimization methods in Finance, Cambridge University Press, 2007.
3. Shabbir Ahmed, The Optimization Process: An example of portfolio optimization, Fall 2002.

References

References

1. Stephen J. Wright, Primal-Dual Interior-Point Methods, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.
2. G. Cornuejols, R. Tütüncü, Optimization methods in Finance, Cambridge University Press, 2007.
3. Shabbir Ahmed, The Optimization Process: An example of portfolio optimization, Fall 2002.
4. Marnix Engels, Portfolio Optimization: Beyond Markowitz, January 13, 2004.

References

References

1. Stephen J. Wright, Primal-Dual Interior-Point Methods, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.
2. G. Cornuejols, R. Tütüncü, Optimization methods in Finance, Cambridge University Press, 2007.
3. Shabbir Ahmed, The Optimization Process: An example of portfolio optimization, Fall 2002.
4. Marnix Engels, Portfolio Optimization: Beyond Markowitz, January 13, 2004.
5. Website: https://inst.eecs.berkeley.edu/~ee127a/book/login/exa_quad_fcn_cvx.html