Qianli Deng (Shally)
E-mail: dqianli@umd.edu

Advisor: Dr. Mark A. Austin
Department of Civil and Environmental Engineering
Institute for Systems Research
University of Maryland, College Park, MD 20742
E-mail: austin@isr.umd.edu

Solving Minimax Problems with Feasible Sequential Quadratic Programming

05/06/2014

Constrained minimax problem

Minimize $\max_{i \in I^f} \{f_i(x)\}$

s.t. $x \in X$ $\quad X$ is the set of points $x \in \mathbb{R}^n$ satisfying

$$\begin{cases} bl \le x \le bu & n_B \\ g_j(x) \le 0 & j = 1, \ldots, n_{NI} \\ g_j(x) \equiv \langle c_j, x \rangle - d_j \le 0 & j = 1, \ldots, n_{LI} \\ h_j(x) = 0 & j = 1, \ldots, n_{NE} \\ h_j(x) \equiv \langle a_j, x \rangle - b_j = 0 & j = 1, \ldots, n_{LE} \end{cases}$$

Constrained minimax problem

$$\text{Minimize } \max_{i \in I^f} \{f_i(x)\}$$

*** Objectives**

s.t. $x \in X$ $\quad X$ is the set of points $x \in \mathbb{R}^n$ satisfying

$$\begin{cases} bl \le x \le bu & n_B \\ g_j(x) \le 0 & j = 1, \ldots, n_{NI} \\ g_j(x) \equiv \langle c_j, x \rangle - d_j \le 0 & j = 1, \ldots, n_{LI} \\ h_j(x) = 0 & j = 1, \ldots, n_{NE} \\ h_j(x) \equiv \langle a_j, x \rangle - b_j = 0 & j = 1, \ldots, n_{LE} \end{cases}$$

Constrained minimax problem

$$\text{Minimize } \max_{i \in I^f} \{f_i(x)\}$$ *** Objectives**

s.t. $x \in X$ $\quad X$ is the set of points $x \in \mathbb{R}^n$ satisfying

$$\begin{cases} bl \le x \le bu & n_B \\ g_j(x) \le 0 & j = 1, \ldots, n_{NI} \\ g_j(x) \equiv \langle c_j, x \rangle - d_j \le 0 & j = 1, \ldots, n_{LI} \\ h_j(x) = 0 & j = 1, \ldots, n_{NE} \\ h_j(x) \equiv \langle a_j, x \rangle - b_j = 0 & j = 1, \ldots, n_{LE} \end{cases}$$

*** Bounds**

*** Nonlinear inequality**

*** Linear inequality**

*** Nonlinear equality**

*** Linear equality**

FSQP Algorithm

- A random point
- A point fitting all linear constraints
- A point fitting all constraints
- An optimal point fitting all constraints

FSQP Algorithm

- **A random point**
- A point fitting all linear constraints
- A point fitting all constraints
- An optimal point fitting all constraints

FSQP Algorithm

- A random point
- **A point fitting all linear constraints**
  - boundary + linear inequality + linear equality
- A point fitting all constraints
- An optimal point fitting all constraints

FSQP Algorithm

- A random point

- A point fitting all linear constraints
  - boundary + linear inequality + linear equality

- **A point fitting all constraints**
  - boundary + linear inequality + linear equality + nonlinear inequality + nonlinear equality

- An optimal point fitting all constraints

FSQP Algorithm

- A random point
- A point fitting all linear constraints
  - boundary + linear inequality + linear equality
- A point fitting all constraints
  - boundary + linear inequality + linear equality + nonlinear inequality + nonlinear equality
- **An optimal point fitting all constraints**

FSQP Algorithm

Step 1. Initialization x_k – a point; H_k – Hessian matrix

↓

Step 2. A search line d_k – direction

↓

Step 3. Line search t_k – distance

↓

Step 4. Updates x_{k+1} – an update point;

H_{k+1} – an update Hessian matrix

Initialization

x_0 = an initial point fitting all constraints

H_0 = an identity matrix

$p_{0,j}$ = a constant

$k = 0$

$$f_m(x, p) = \max_{i \in I^f} \{ f_i(x) \} - \sum_{j=1}^{n_{NE}} p_j h_j(x)$$

$$j = 1, \ldots, n_{NE}$$

Computation of a search line

1. Compute d_k^0, solution of $QP(x_k, H_k, p_k)$

$$\begin{cases} \min\limits_{d^0} & \frac{1}{2}\langle d^0, H_k d^0\rangle + f'(x_k, d^0, p_k) \\ s.t. & bl \le x_k + d^0 \le bu \\ & g_j(x_k) + \langle \nabla g_j(x_k), d^0\rangle \le 0, \quad j = 1,\dots,t_i \\ & h_j(x_k) + \langle \nabla h_j(x_k), d^0\rangle \le 0, \quad j = 1,\dots,n_e \\ & \langle a_j, x_k + d^0\rangle = b_j \quad j = 1,\dots,t_e - n_e \end{cases}$$

$$f'(x,d,p) = \max_{i\in I^f}\{f_i(x) + \langle \nabla f_i(x), d\rangle\} - f_{I^f}(x) - \sum_{j=1}^{n_e} p_j \langle \nabla h_i(x), d\rangle$$

$$f_I(x) = \max_{i\in I}\{f_i(x)\}$$

2. Compute d_k^1, solution of $QP(x_k, d_k^0, p_k)$

$$\begin{cases} \min\limits_{d^1\in\mathbb{R}^n, \gamma\in\mathbb{R}} & \frac{\eta}{2}\langle d_k^0 - d^1, d_k^0 - d^1\rangle + \gamma \\ s.t. & bl \le x_k + d^1 \le bu \\ & f'(x_k, d^1, p_k) \le \gamma \\ & g_j(x_k) + \langle \nabla g_j(x_k), d^1\rangle \le \gamma \quad j = 1,\dots,n_i \\ & \langle c_j, x_k + d^1\rangle \le d_j \quad j = 1,\dots,t_i - n_i \\ & h_j(x_k) + \langle \nabla h_j(x_k), d^1\rangle \le \gamma \quad j = 1,\dots,n_e \\ & \langle a_j, x_k + d^1\rangle = b_j \quad j = 1,\dots,t_e - n_e \end{cases}$$

3. Set $d_k = (1-\rho_k)d_k^0 + \rho_k d_k^1$

$$\rho_k = \| d_k^0 \|^{\kappa} / (\| d_k^0 \|^{\kappa} + v_k)$$

$$v_k = \max(0.5, \| d_k^1 \|^{\tau 1})$$

4. Compute $\tilde{d}_k$

Stop: if $\| d_k^0 \| \le \varepsilon$ and $\sum_{j=1}^{n_e} | h_j(x_k) | \le \varepsilon_e$

Line search

Armijo line search

$$\delta_k = \begin{cases} -\left\langle d_k^0, H_k d_k^0 \right\rangle & n_i + n_e = 0 \\ f'(x_k, d_k, p_k) & n_i + n_e \neq 0 \end{cases}$$

No nonlinear constraints

$$f'(x,d,p) = \max_{i \in I^f} \{ f_i(x) + \langle \nabla f_i(x), d \rangle \} - f_{I^f}(x) - \sum_{j=1}^{n_{NE}} p_j \langle \nabla h_i(x), d \rangle$$

Compute t_k, the first number t in the sequence $\{1, \beta, \beta^2, \ldots\}$ satisfying

$$f_m(x_k + t_k d_k, p_k) \leq f_m(x_k, p_k) + \alpha t \delta_k$$

$$\begin{cases} g_j(x_k + t d_k) \leq 0, & j = 1, \ldots, n_{NI} \\ \left\langle c_j, x_k + t d_k \right\rangle \leq d_j, & j = 1, \ldots, n_{LI} \\ h_j(x_k + t d_k) \leq 0, & j = 1, \ldots, n_{NE} \end{cases}$$

$$f_m(x,p) = \max_{i \in I^f} \{ f_i(x) \} - \sum_{j=1}^{n_{NE}} p_j h_j(x)$$

Updates

$$x_{k+1} = x_k + t_k d_k$$

H_{k+1} updated by BFGS formula with Powell's modification

$$\eta_{k+1} = x_{k+1} - x_k, \qquad \gamma_{k+1} = \nabla_x L(x_{k+1}, p_k) - \nabla_x L(x_k, p_k)$$

$$\theta_{k+1} = \begin{cases} 1, & \eta_{k+1}^T \gamma_{k+1} \geq 0.2 \eta_{k+1}^T H_k \eta_{k+1} \\ \dfrac{0.8 \eta_{k+1}^T H_k \eta_{k+1}}{\eta_{k+1}^T H_k \eta_{k+1} - \eta_{k+1}^T \gamma_{k+1}} & otherwise \end{cases}$$

$$\xi_{k+1} = \theta_{k+1} \cdot \gamma_{k+1} + (1 - \theta_{k+1}) \cdot H_k \gamma_{k+1}$$

$$H_{k+1} = H_k - \frac{H_k \eta_{k+1} \eta_{k+1}^T H_k}{\eta_{k+1}^T H_k \eta_{k+1}} + \frac{\xi_{k+1} \xi_{k+1}^T}{\eta_{k+1}^T \xi_{k+1}}$$

$$p_{k+1,j} = \begin{cases} p_{k,j} & p_{k,j} + \bar{\mu}_j \geq \varepsilon_1 \\ \max\{\varepsilon_1 - \bar{\mu}_j, \delta p_{k,j}\} & otherwise \end{cases}$$

$$k = k + 1$$

Implementation

JFSQP.java

Initial.java

QP.java

MiniMax.java

Direction_d0.java

QP.java

KKT.java

Direction_d1.java

QP.java

Arcsearch.java

Check.java

BFGS_Powell.java

MiniMax.java

Direction_d0.java

QP.java

KKT.java

Direction_d1.java

QP.java

Arcsearch.java

Check.java

BFGS_Powell.java

Implementation

JFSQP.java

Initial.java

QP.java

MiniMax.java

Direction_d0.java

QP.java

KKT.java

Direction_d1.java

QP.java

Arcsearch.java

Check.java

BFGS_Powell.java

Loop 1.
Find initial feasible point

MiniMax.java

Direction_d0.java

QP.java

KKT.java

Direction_d1.java

QP.java

Arcsearch.java

Check.java

BFGS_Powell.java

Loop 2.
Find optimal point

Implementation

JFSQP.java

Initial.java

QP.java

Loop 1.
Find initial feasible point

MiniMax.java

Direction_d0.java

QP.java

KKT.java

Direction_d1.java

QP.java

Arcsearch.java

Check.java

BFGS_Powell.java

MiniMax.java

Direction_d0.java

QP.java

KKT.java

Direction_d1.java

QP.java

Arcsearch.java

Check.java

BFGS_Powell.java

Loop 2.
Find optimal point

Implementation

JFSQP.java

Initial.java

QP.java ← **Find a point fitting all linear constraints**

MiniMax.java

↑ **Find point fitting all constraints (linear + nonlinear)**

Direction_d0.java

QP.java

KKT.java

Direction_d1.java

QP.java

Arcsearch.java

Check.java

BFGS_Powell.java

MiniMax.java

↑ **Find the optimal point also fitting all constraints (linear + nonlinear)**

Direction_d0.java

QP.java

KKT.java

Direction_d1.java

QP.java

Arcsearch.java

Check.java

BFGS_Powell.java

Implementation

Three open-source Java libraries were leveraged:

1) JAMA (Version 1.0.3; 11/2012)

http://math.nist.gov/javanumerics/jama/

2) Apache Commons.Lang (Version 3.3.2; 04/2014)

http://commons.apache.org/

3) Joptimizer (Version 3.3.0; 04/2014)

http://www.joptimizer.com/

Validation

JFSQP.java

Initial.java

QP.java

MiniMax.java

Direction_d0.java

QP.java

KKT.java

Direction_d1.java

QP.java

Arcsearch.java

Check.java

BFGS_Powell.java

Validation

JFSQP.java

Initial.java

QP.java

MiniMax.java

Direction_d0.java

QP.java

KKT.java

Direction_d1.java

QP.java

Arcsearch.java

Check.java

BFGS_Powell.java

testJFSQP.java

testInitial.java

testQP.java

testMiniMax.java

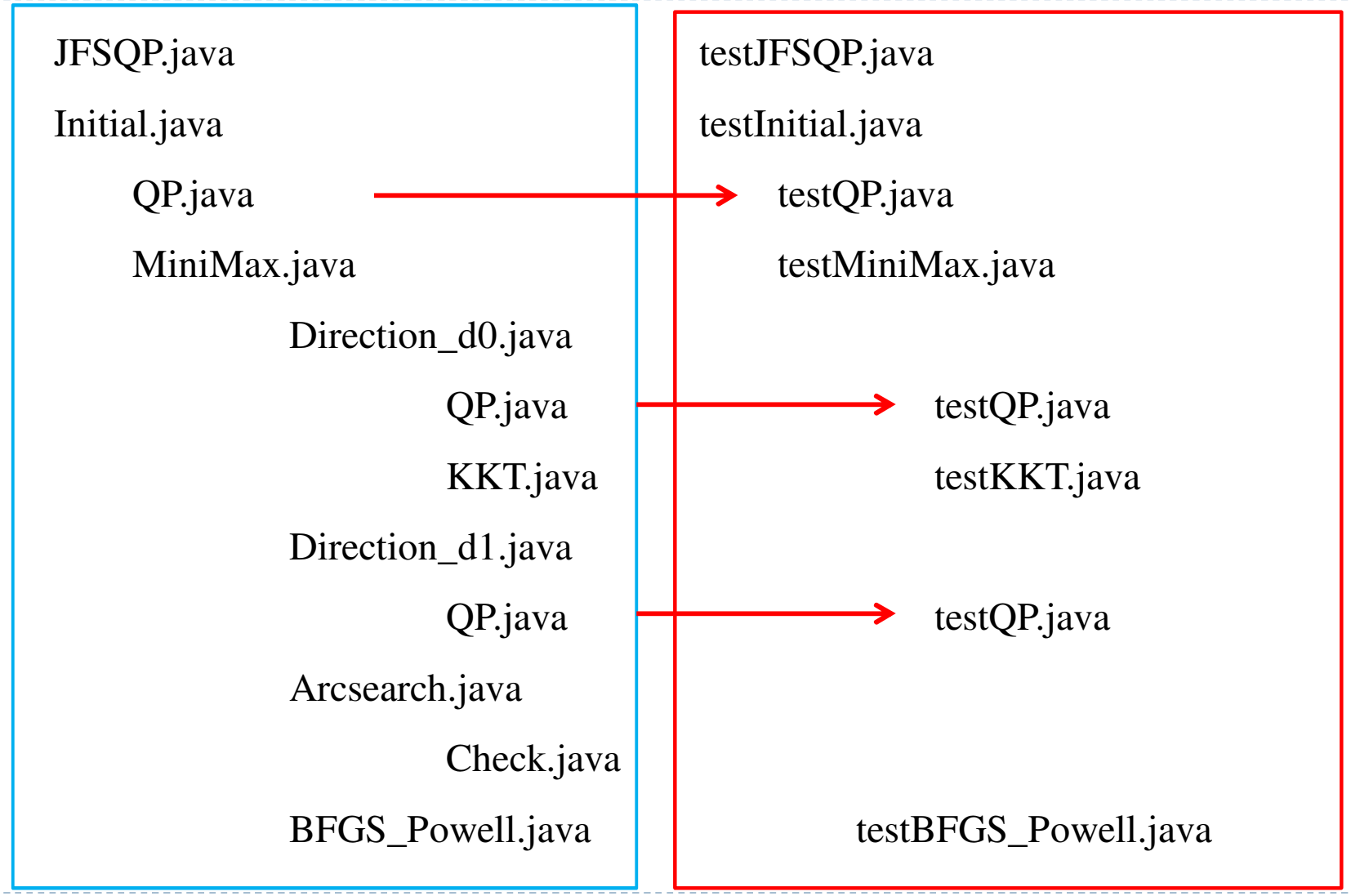
testQP.java

testKKT.java

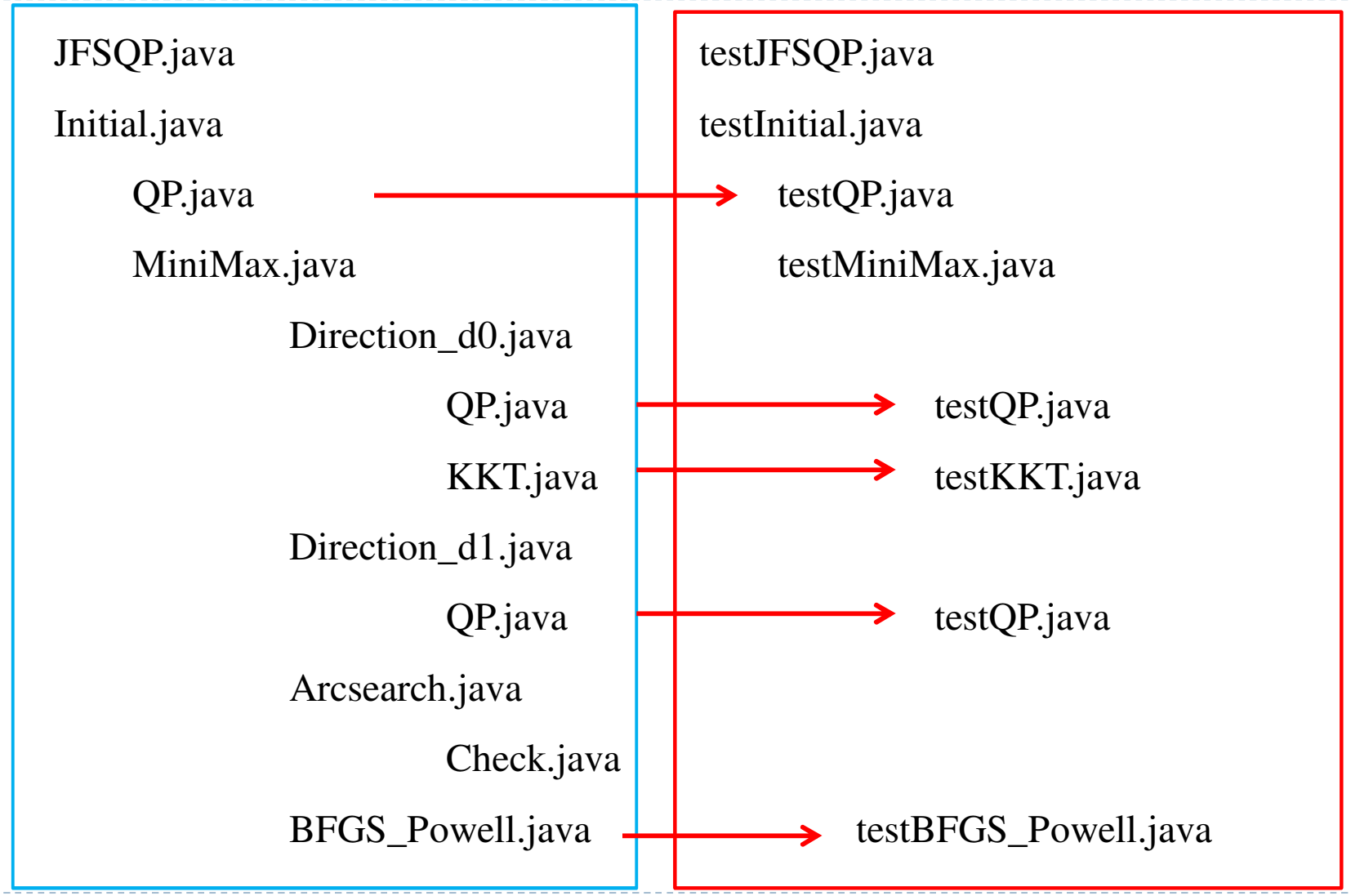
testQP.java

testBFGS_Powell.java

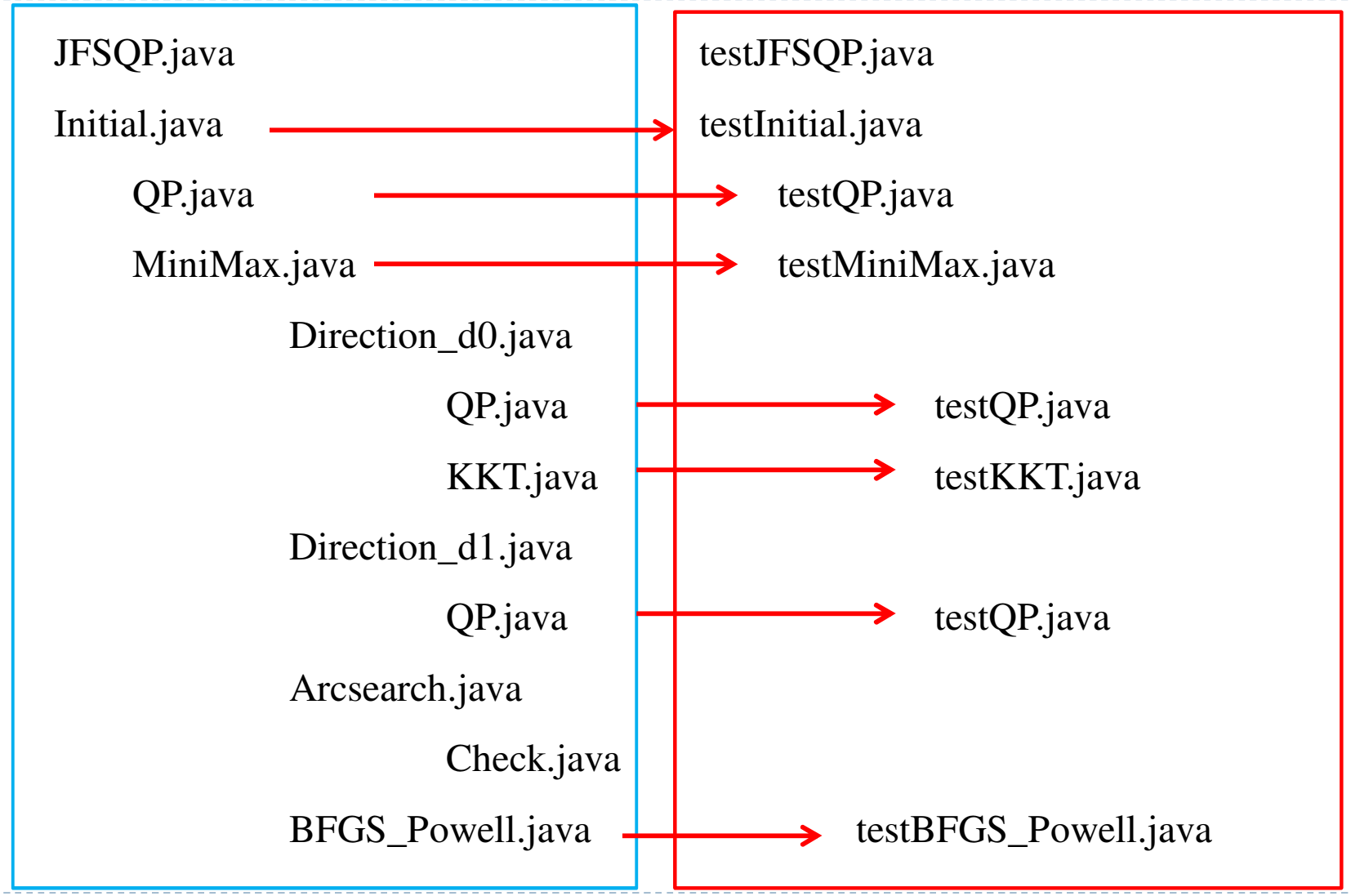
Validation


JFSQP.java
Initial.java
QP.java
MiniMax.java
Direction_d0.java
QP.java
KKT.java
Direction_d1.java
QP.java
Arcsearch.java
Check.java
BFGS_Powell.java
testJFSQP.java
testInitial.java
testQP.java
testMiniMax.java
testQP.java
testKKT.java
testQP.java
testBFGS_Powell.java

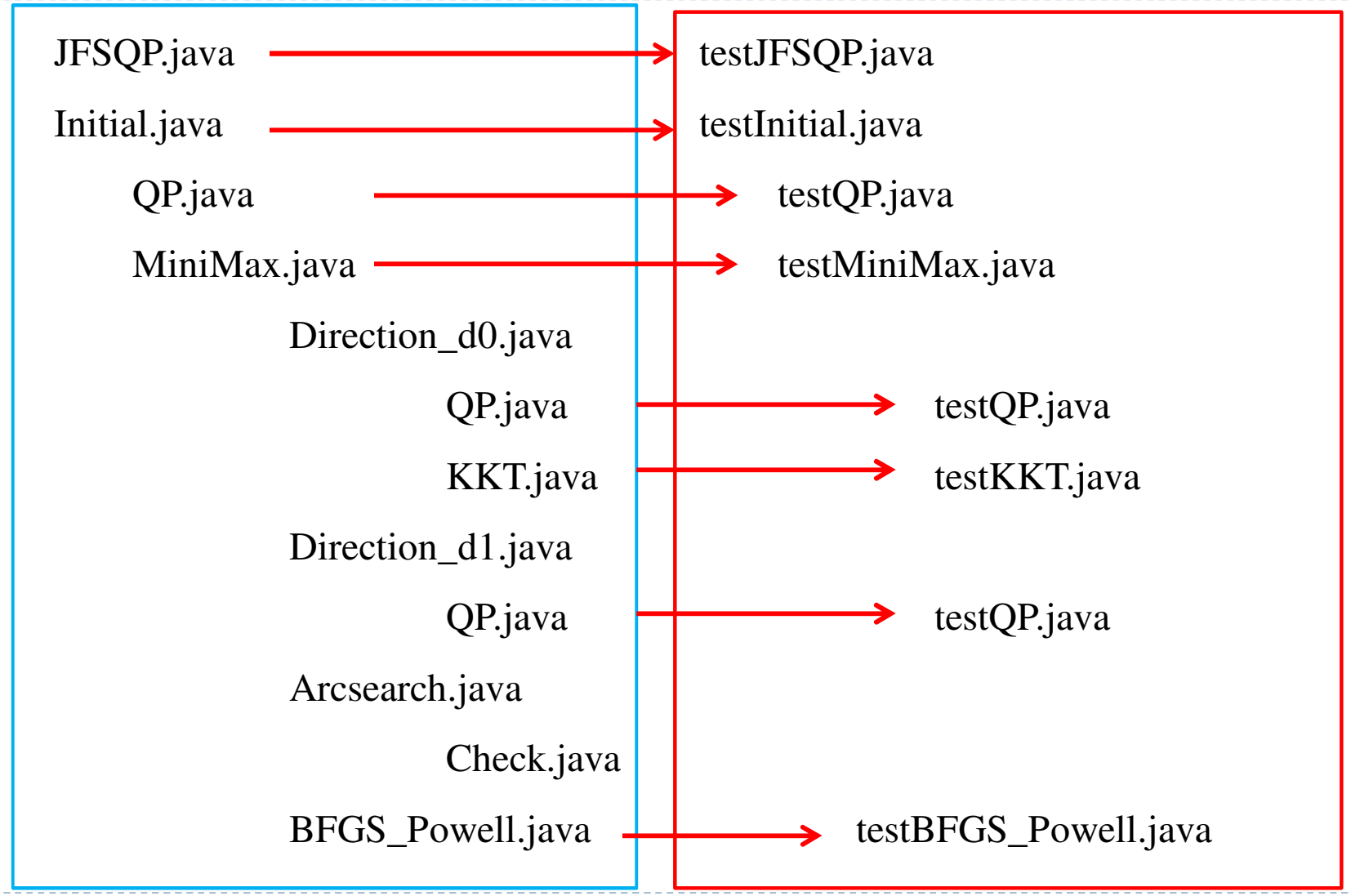
Validation


JFSQP.java
Initial.java
QP.java
MiniMax.java
Direction_d0.java
QP.java
KKT.java
Direction_d1.java
QP.java
Arcsearch.java
Check.java
BFGS_Powell.java
testJFSQP.java
testInitial.java
testQP.java
testMiniMax.java
testQP.java
testKKT.java
testQP.java
testBFGS_Powell.java

Validation


JFSQP.java
Initial.java
QP.java
MiniMax.java
Direction_d0.java
QP.java
KKT.java
Direction_d1.java
QP.java
Arcsearch.java
Check.java
BFGS_Powell.java
testJFSQP.java
testInitial.java
testQP.java
testMiniMax.java
testQP.java
testKKT.java
testQP.java
testBFGS_Powell.java

Validation


JFSQP.java
Initial.java
QP.java
MiniMax.java
Direction_d0.java
QP.java
KKT.java
Direction_d1.java
QP.java
Arcsearch.java
Check.java
BFGS_Powell.java
testJFSQP.java
testInitial.java
testQP.java
testMiniMax.java
testQP.java
testKKT.java
testQP.java
testBFGS_Powell.java

Validation

testJFSQP.java

testInitial.java

testQP.java

testMiniMax.java

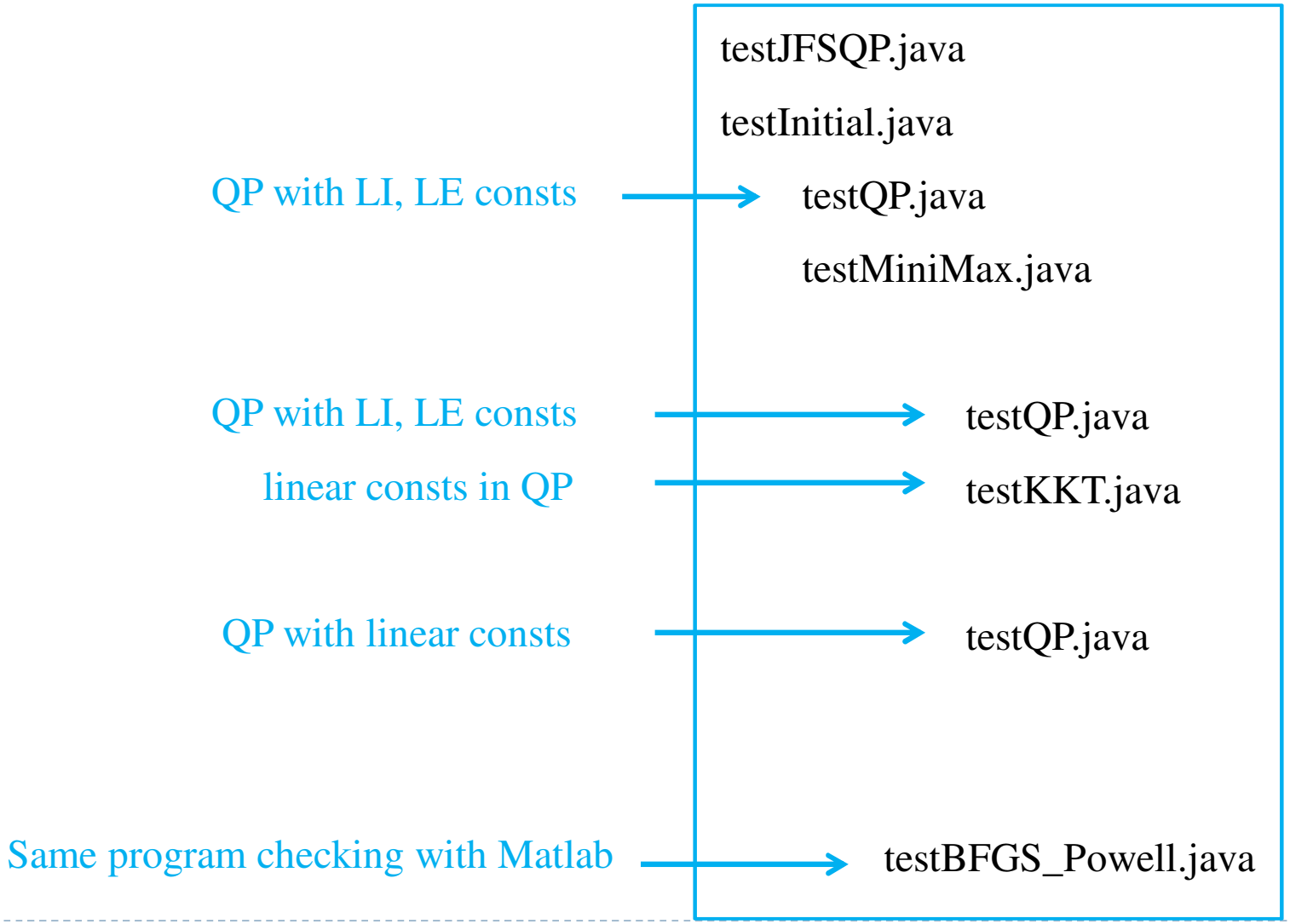
testQP.java

testKKT.java

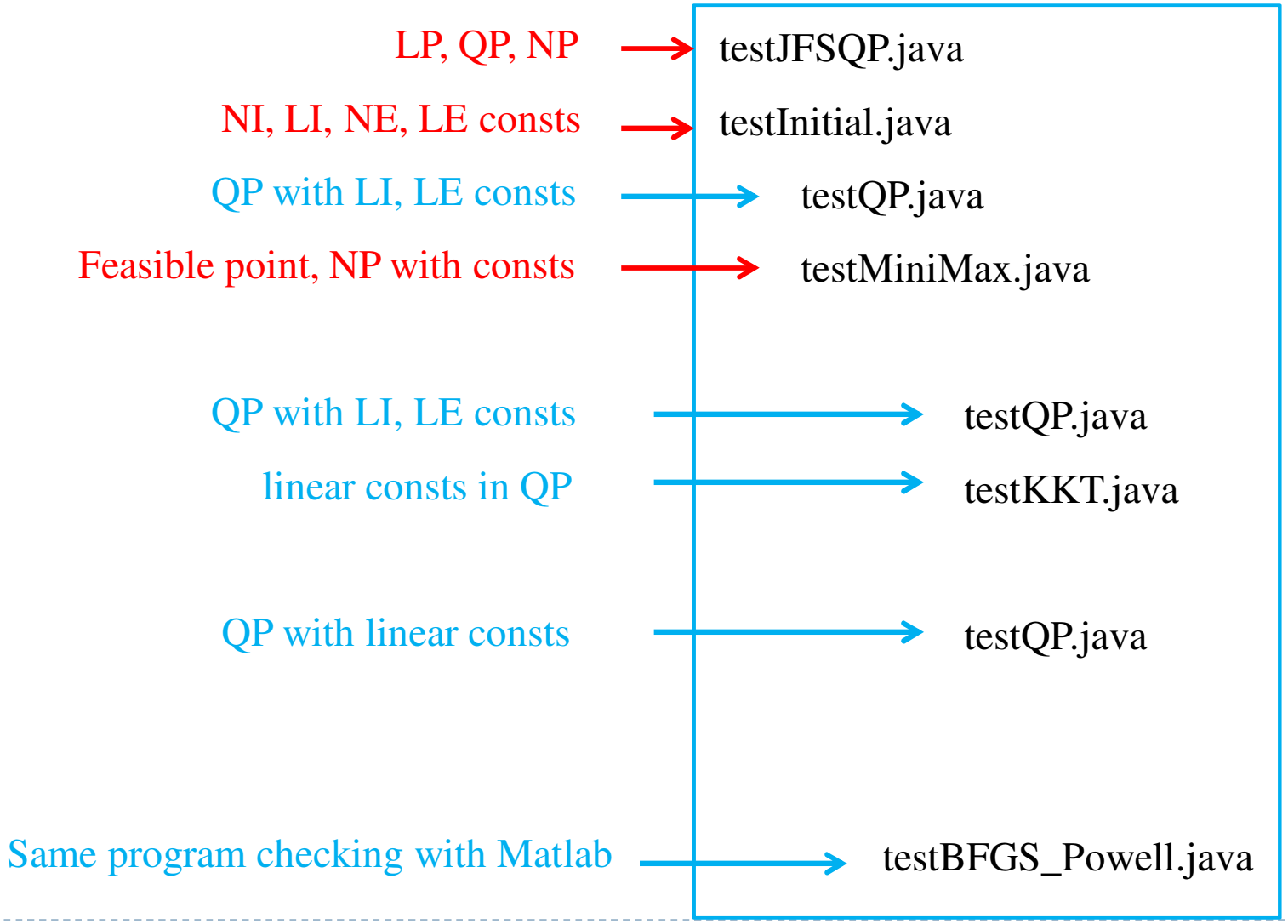
testQP.java

testBFGS_Powell.java

Validation


testJFSQP.java
testInitial.java
QP with LI, LE consts
testQP.java
testMiniMax.java
QP with LI, LE consts
testQP.java
linear consts in QP
testKKT.java
QP with linear consts
testQP.java
Same program checking with Matlab
testBFGS_Powell.java

Validation


LP, QP, NP
testJFSQP.java
NI, LI, NE, LE consts
testInitial.java
QP with LI, LE consts
testQP.java
Feasible point, NP with consts
testMiniMax.java
QP with LI, LE consts
testQP.java
linear consts in QP
testKKT.java
QP with linear consts
testQP.java
Same program checking with Matlab
testBFGS_Powell.java

Example 1

$$\min_{x \in \mathbb{R}^2} f(x) = -x_1 + x_2$$

$$\text{S.t.} \quad \begin{cases} -x_1 \le 0 \\ -x_2 \le 0 \\ x_1 - 1 \le 0 \\ x_2 - 1 \le 0 \end{cases}$$

Known minimizer:

$$x_0 = (-3, 2)^T ; f(x_0) = 5$$

$$x^* = (1, 0)^T ; f(x^*) = -1$$

Reference: Hock, Willi, and Klaus Schittkowski. "Test examples for nonlinear programming codes." *Journal of Optimization Theory and Applications* 30.1 (1980): 127-129.

Example 1

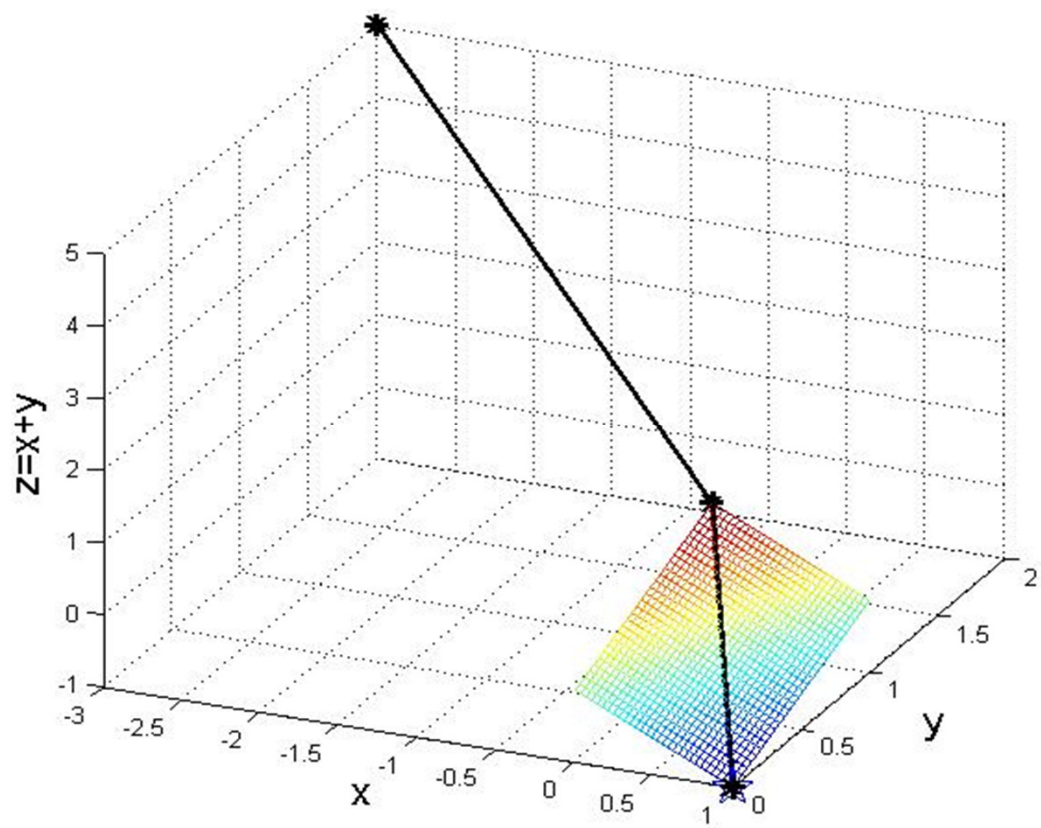
$$\min_{x \in \mathbb{R}^2} f(x) = -x_1 + x_2$$

$$\text{S.t.} \quad \begin{cases} -x_1 \le 0 \\ -x_2 \le 0 \\ x_1 - 1 \le 0 \\ x_2 - 1 \le 0 \end{cases}$$

Known minimizer:

$$x_0 = (-3, 2)^T ; f(x_0) = 5$$

$$x^* = (1, 0)^T ; f(x^*) = -1$$



Reference: Hock, Willi, and Klaus Schittkowski. "Test examples for nonlinear programming codes." *Journal of Optimization Theory and Applications* 30.1 (1980): 127-129.

Example 2

$$\min_{x \in \mathbb{R}^3} f(x) = (x_1 + 3x_2 + x_3)^2 + 4(x_1 - x_2)^2$$

$$\text{S.t.} \quad \begin{cases} x_1, x_2, x_3 \geq 0 \\ x_1^3 - 6x_2 - 4x_3 + 3 \leq 0 \\ 1 - x_1 - x_2 - x_3 = 0 \end{cases}$$

Known global minimizer: $x^* = (0,0,1)^T$; $f(x^*) = 1$

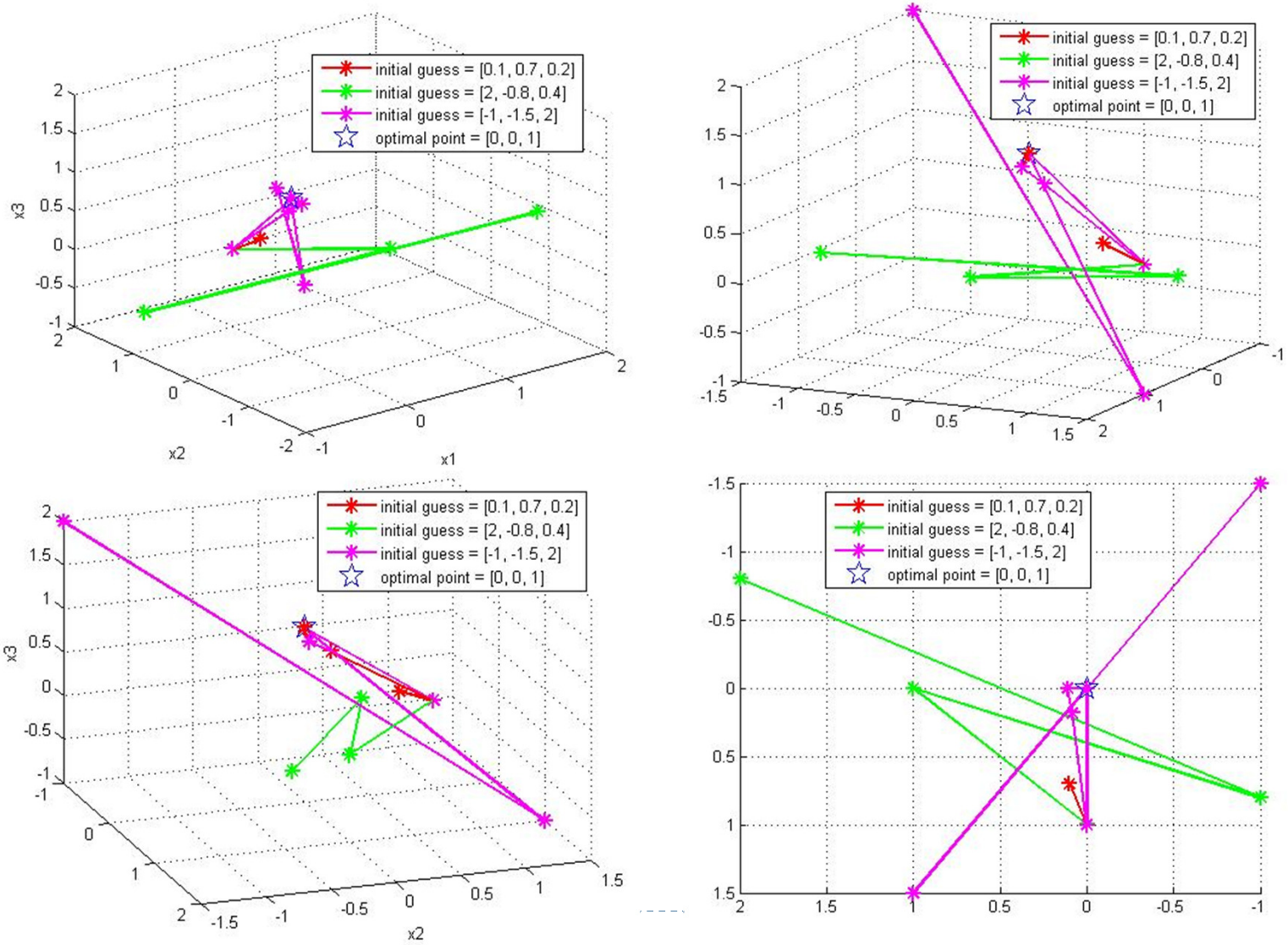
Reference: Hock, Willi, and Klaus Schittkowski. "Test examples for nonlinear programming codes." *Journal of Optimization Theory and Applications* 30.1 (1980): 127-129.

Example 2

Table 1. Results return from different initial guess

Initial guess	Objective function	Iterations for finding feasible point	Feasible point return	Iterations for finding optimal point	Final point return
$\begin{pmatrix} 0.1 \\ 0.7 \\ 0.2 \end{pmatrix}$	7.2	2	$\begin{pmatrix} 3.05637e-07 \\ 9.99999e-01 \\ 1.01789e-06 \end{pmatrix}$	29	$\begin{pmatrix} 4.35632e-05 \\ 9.67363e-08 \\ 9.99956e-01 \end{pmatrix}$
$\begin{pmatrix} 2 \\ -0.8 \\ 0.4 \end{pmatrix}$	31.36	2	$\begin{pmatrix} 1.34913e-07 \\ 9.99999e-01 \\ 7.11207e-07 \end{pmatrix}$	33	$\begin{pmatrix} 4.26647e-05 \\ 9.67349e-08 \\ 9.99957e-01 \end{pmatrix}$
$\begin{pmatrix} -1.0 \\ -1.5 \\ 2.0 \end{pmatrix}$	48.25	2	$\begin{pmatrix} 4.04903e-08 \\ 1.00000e-00 \\ 2.02531e-07 \end{pmatrix}$	32	$\begin{pmatrix} 2.45688e-05 \\ 9.81212e-08 \\ 9.99975e-01 \end{pmatrix}$

Example 2


initial guess = [0.1, 0.7, 0.2]
initial guess = [2, -0.8, 0.4]
initial guess = [-1, -1.5, 2]
optimal point = [0, 0, 1]
x1
x2
x3

Project Schedule

October	• **Literature review;** • **Specify the implementation module details;** • **Structure the implementation;**
November	• **Develop the quadratic programming module;** • **Unconstrained quadratic program;** • **Strictly convex quadratic program;** • **Validate the quadratic programming module;**
December	• **Develop the Gradient and Hessian matrix calculation module;** • **Validate the Gradient and Hessian matrix calculation module;** • **Midterm project report and presentation;**

January	• **Develop Armijo line search module;** • **Validate Armijo line search module;**
February	• **Develop the feasible initial point module;** • **Validate the feasible initial point module;** • **Integrate the program;**
March	• **Debug and document the program;** • **Validate and test the program with case application;**
April	• Add arch search variable $\tilde{d}$; • Compare calculation efficiency of line search with arch search methods;
May	• Develop the user interface if time available; • Final project report and presentation;

Deliverables

Project proposal

Algorithm description

Well-documented Java codes

Test cases: LP, QP, NP with NI, LI, NE, LE constraints

Validation results

Presentations

Project reports

Bibliography

Charalambous, Conn and AR Conn. "An Efficient Method to Solve the Minimax Problem Directly." *SIAM Journal on Numerical Analysis* 15, no. 1 (1978): 162-187.

Goldfarb, Donald and Ashok Idnani. "A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs." *Mathematical programming* 27, no. 1 (1983): 1-33.

Lawrence, Craig T and André L Tits. "Nonlinear Equality Constraints in Feasible Sequential Quadratic Programming∗." *Optimization Methods and Software* 6, no. 4 (1996): 265-282.

Lawrence, Craig T and André L Tits. "A Computationally Efficient Feasible Sequential Quadratic Programming Algorithm." *Siam Journal on optimization* 11, no. 4 (2001): 1092-1118.

Li, Wu and John Swetits. "A New Algorithm for Solving Strictly Convex Quadratic Programs." *SIAM Journal on Optimization* 7, no. 3 (1997): 595-619.

Madsen, Kaj and Hans Schjaer-Jacobsen. "Linearly Constrained Minimax Optimization." *Mathematical Programming* 14, no. 1 (1978): 208-223.

Mayne, DQ and E Polak. "Feasible Directions Algorithms for Optimization Problems with Equality and Inequality Constraints." *Mathematical Programming* 11, no. 1 (1976): 67-80.

Watson, GA. "The Minimax Solution of an Overdetermined System of Non-Linear Equations." *IMA Journal of Applied Mathematics* 23, no. 2 (1979): 167-180.