Workshop on Computational Optimization

Session-1: Linear and Nonlinear Optimization

ORSI-2018

Siddharth Agarwal, Rupak Rokade and Ashutosh Mahajan
IIT Bombay

December 16, 2018, Mumbai

FOSSEE: Free and Open Source Software for Education

- Eliminate use of proprietary/commercial software packages in education.
- Develop FOSS tools for various engineering domains.
- Save institutional and government money.
- Increase use of FOSS in education.
- Use of FOSS makes students and teachers better programmers
- FOSSEE's contributions include development for Scilab, OpenModelica, OpenFOAM, Python, ESIM, DWSIM, Drupal, etc.

Learning Objectives

Solving the following problems using Scilab FOSSEE Optimization Toolbox:

- Linear Programming
- Sensitivity Analysis for Linear Programming
- Nonlinear Programming

Introduction to Scilab

- Scilab is a free and open source software for mathematical and numerical computation.
- It is available from `www.scilab.org`
- The toolbox directly requires understanding of the following scilab capabilities:
  - Vectors and matrices
  - Functions
  - Lists
  - Using data structures
- Scilab provides extensive documentation. Simply type 'help topic' on the console.

Introduction to the FOSSEE Optimization Toolbox

- FOSSEE Optimization Toolbox (FOT) for Scilab offers several optimization routines.
- This includes:
  - Linear Programming
  - Quadratic Programming
  - Unconstrained Programming
  - Bounded Programming
  - Constrained Programming
  - Multiobjective Goal Programming
  - Integer Programming
- These routines call popular optimization libraries such as CLP, IPOPT, CBC, BONMIN, and SYMPHONY in the backend.
- The function arguments and parameters are similar to those available in Matlab.

Linear Programming: The General Form

General Linear Programming Problem form:

$$\min_x \; c^T x$$

subject to:

$$Ax \leq b,$$
$$A_{eq}x = b_{eq},$$
$$lb \leq x \leq ub,$$

where $c, \; A, \; b, A_{eq}, \; b_{eq}, \; b, \;$ and ub are given.

Linear Programming: Input Arguments

Input arguments for Scilab `linprog`:
(c, A, b, Aeq, beq, lb, ub, param)

- `c`: Vector for the coefficients in the objective function.
- `A`: Matrix of coefficients of inequality constraints.
- `b`: Right-hand side of inequality constraints.
- `Aeq`: Matrix of coefficients of equality constraints.
- `beq`: Right-hand side of equality constraints.
- `lb`: The lower bounds for x.
- `ub`: The upper bounds for x.
- `param`: List containing parameters to be set.

Linear Programming: Outputs

Outputs of `linprog`:
(xopt, fopt, exitflag, output, lambda)

- `xopt`: optimal solution of x.
- `fopt`: objective function value at xopt.
- `exitflag`: The status of execution.
- `output`: a structure containing messages from the solver.
- `lambda`: a structure containing the Lagrange multipliers at the optimal solution.

Linear Programming: A Small Example

$$\min_x \; -x_1 - 3x_2$$

Subject to:

$$\begin{aligned} x_1 + x_2 &\leq 2 \\ -0.25x_1 - x_2 &\leq 1 \\ -x_1 - x_2 &\leq -1 \\ -x_1 + x_2 &\leq 2 \end{aligned}$$

$$x_1 + 0.25x_2 = 0.5$$

$$\begin{aligned} -1 \leq x_1 &\leq 1.5 \\ -0.5 \leq x_2 &\leq \infty \end{aligned}$$

Linear Programming: A Small Example

$$c = \begin{bmatrix} -1 & -3 \end{bmatrix}$$

$$\min_x \; -x_1 - 3x_2$$

Subject to:

$$A = \begin{bmatrix} 1 & 1 \\ -0.25 & -1 \\ -1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\begin{aligned} x_1 + x_2 &\leq 2 \\ -0.25x_1 - x_2 &\leq 1 \\ -x_1 - x_2 &\leq -1 \\ -x_1 + x_2 &\leq 2 \end{aligned}$$

$$A_{eq} = \begin{bmatrix} 1 & 0.25 \end{bmatrix}$$

$$x_1 + 0.25x_2 = 0.5$$

$$lb = \begin{bmatrix} -1 \\ -0.5 \end{bmatrix}$$

$$\begin{aligned} -1 \leq x_1 &\leq 1.5 \\ -0.5 \leq x_2 &\leq \infty \end{aligned}$$

Linear Programming: A Small Example

$$c = \begin{bmatrix} -1 & -3 \end{bmatrix} \qquad \min_x \; -x_1 - 3x_2$$

Subject to:

$$A = \begin{bmatrix} 1 & 1 \\ -0.25 & -1 \\ -1 & -1 \\ -1 & 1 \end{bmatrix} \qquad \begin{aligned} x_1 + x_2 &\leq 2 \\ -0.25x_1 - x_2 &\leq 1 \\ -x_1 - x_2 &\leq -1 \\ -x_1 + x_2 &\leq 2 \end{aligned} \qquad b = \begin{bmatrix} 2 \\ 1 \\ -1 \\ 2 \end{bmatrix}$$

$$A_{eq} = \begin{bmatrix} 1 & 0.25 \end{bmatrix} \qquad x_1 + 0.25x_2 = 0.5 \qquad b_{eq} = \begin{bmatrix} 0.5 \end{bmatrix}$$

$$lb = \begin{bmatrix} -1 \\ -0.5 \end{bmatrix} \qquad \begin{aligned} -1 \leq x_1 &\leq 1.5 \\ -0.5 \leq x_2 &\leq \infty \end{aligned} \qquad ub = \begin{bmatrix} 1.5 \\ \%inf \end{bmatrix}$$

Solving using `linprog`

- Open the terminal by going into the dashboard and clicking on it.
- Type `scilab` on the terminal.
- You should see a scilab window with the toolbox loading on the console.
- Type `editor` on the console and press Enter.
- Code the objective function, constraints, and the bounds.
- Execute `linprog`.

Linear Programming: Solution to the Example

You should see the following values for xopt and fopt.

- xopt = [0, 2]
- fopt = -6

The Maximisation Problem

- The above problem was a minimisation problem with $\leq$ constraints. This is the standard format that linprog solves. However, not all problems are directly framed to be of this sort.
- To solve a maximisation problem, change the sign of the cost function terms.
- To change $\geq$ constraint to $\leq$ constraint, change the sign on the terms of the constraint

Linear Programming: Modelling and Solving

Aleph Toys assembles three types of toys: trains, trucks, and cars using three operations. The daily limits on the available times for the three operations are 430, 460, and 420 minutes respectively. The revenues per unit of toy train, truck and car are INR 30, INR 20, and INR 50 respectively. The corresponding times per train and per car are (1,3,1), (2,0,4) and (1,2,0) minutes (a zero time indicates that the operation is not used). Determine the optimum production rate to maximize profit. Check for any surplus resource availability.

Linear Programming: Modelling and Solving

Aleph Toys assembles three types of toys: trains, trucks, and cars using three operations. The daily limits on the available times for the three operations are 430, 460, and 420 minutes respectively. The revenues per unit of toy train, truck and car are INR 30, INR 20, and INR 50 respectively. The corresponding times per train and per car are (1,3,1), (2,0,4) and (1,2,0) minutes (a zero time indicates that the operation is not used). Determine the optimum production rate to maximize profit. Check for any surplus resource availability.

Approach the problem in the following manner:

- Develop a LP formulation on paper.
- Identify the relevant matrices.
- Solve it using linprog
- Does your answer make sense?

Linear Programming: Modelling and Solving

You should see the following values for xopt and fopt.

- xopt = [0 100 230]
- fopt = -13500

Question: How much time is spent on the third operation?

Linear Programming: Sensitivity Analysis

- What happens if we change the inputs?
- Dual values (λ) provide this information (for b, b_{eq}, lb, ub).
- They indicate the change in the objective function value for a unit change in a bounds of a variable or right-hand side of a constraint.

Linear Programming: Dual Information

Aleph Toys assembles three types of toys: trains, trucks, and cars using three operations. The daily limits on the available times for the three operations are 430, 460, and 420 minutes respectively. The revenues per unit of toy train, truck and car are INR 30, INR 20, and INR 50 respectively. The corresponding times per train and per car are (1,3,1), (2,0,4) and (1,2,0) minutes (a zero time indicates that the operation is not used).

Recall `xopt = [0 100 230]`

Questions (Answer without solving the modified model)

- Suppose we increase the available time for Operation-3 to 421 minutes in the original model. Will the optimal solution change? By how much?
- Suppose we increase the available time for Operation-1 to 431 minutes in the original model. Will the optimal solution change? By how much?

Linear Programming: Dual Information

One can estimate the changes without solving a new LP by observing the dual values:

$$[xopt, fopt, exitflag, output, lambda] = linprog(c, A, b, A_{eq}, b_{eq}, lb, ub);$$

- lambda.lower: duals corresponding to variable lower bounds
- lambda.upper: duals corresponding to variable upper bounds
- lambda.ineqlin: duals corresponding to inequality constraints
- lambda.eqlin: duals corresponding to equality constraints

Exercise: Try changing the rhs and variable bounds (lb and ub) and check if duals are giving the right estimates.

Note: The duals give a *good* estimate for small changes, and a lower estimate for larger changes.

Nonlinear Programming: General form

General nonlinear programming problem form.

$$\min_x \; f(x)$$

subject to:

$$Ax \leq b,$$
$$A_{eq}x = b_{eq},$$
$$c(x) \leq 0,$$
$$c_{eq}(x) = 0,$$
$$lb \leq x \leq ub,$$

where f, A, b, A_{eq}, b_{eq}, c, c_{eq}, lb, *and* ub *are given.*

You can solve nonlinear programming problems using the `fmincon` function in FOT.

Nonlinear Programming: fmincon input arguments

Inputs:

- `f`: A function representing the objective function of the problem.
- `x0`: A vector of doubles, containing the starting values for x.
- `A`: Matrix of coefficients of the linear inequality constraints.
- `b`: Right-hand side of the linear inequality constraints.
- `Aeq`: Matrix of coefficients of the linear equality constraints.
- `beq`: Right-hand side of the linear equality constraints.
- `lb`: The lower bounds for x.
- `ub`: The upper bounds for x.
- `nlc`: A function representing the nonlinear inequality and equality constraints.
- `param`: List containing parameters for the solver.

Nonlinear Programming: fmincon outputs

Outputs:
(xopt, fopt, exitflag, output, lambda)

- `xopt`: xopt is the optimal value of x.
- `fopt`: fopt is the objective function value at the optimal value of x.
- `exitflag`: The status of execution.
- `output`: A structure containing detailed information about the optimization.
- `lambda`: A structure containing the Lagrange multipliers at the optimal solution.

Nonlinear Programming: Example

$$f(x) = x_1^2 - \frac{x_1 x_2}{3} + x_2^2$$

subject to:

$$x_1 + x_2 \leq 2$$
$$x_1 + \frac{x_2}{4} \leq 1$$
$$x_1^2 + x_2^3 \leq 1$$

This example is solved on the next slide using `fmincon`.

Nonlinear Programming: Example

Objective Function:

```
function y=f(x)
        y=x(1)^2 - x(1)*x(2)/3 + x(2)^2;
endfunction
```

Initial values for x and linear constraints are defined in the standard way.
Nonlinear constraints:

```
function [c,ceq]=nlc(x)
        c = x(1)^2  + x(2)^3 -1;
        ceq = [];
endfunction
```

Calling fmincon:

```
[xopt, fopt] = fmincon(f,x0,A,b,[],[],[],[],nlc);
```

Nonlinear Programming: Another Example

$$min_x f(x) = (x_3 + 2)x_1x_2^2$$

Subject to:

$$1 - x_1^3 \frac{x_3}{71785x_2^4} \leq 0$$

$$1 - 140.45 \frac{x_1}{x_2^2 x_3} \leq 0$$

$$\frac{(4x_1 - x_2x_1)}{12566(x_2^3 x_1 - x_2^4)} + \frac{1}{5108x_2^2} - 1 \leq 0$$

$$\frac{x_1 + x_2}{1.5} - 1 \leq 0$$

$$0.05 \leq x_2 \leq 2$$

$$0.25 \leq x_1 \leq 1.3$$

$$2 \leq x_3 \leq 15$$

Nonlinear Programming: Another Example

$$min_x f(x) = (x_3 + 2)x_1 x_2^2$$

$$f(x) = (x_3 + 2)x_1 x_2^2$$

Subject to:

$$1 - x_1^3 \frac{x_3}{71785 x_2^4} \leq 0$$

$$1 - 140.45 \frac{x_1}{x_2^2 x_3} \leq 0$$

$$\frac{(4x_1 - x_2 x_1)}{12566(x_2^3 x_1 - x_2^4)} + \frac{1}{5108 x_2^2} - 1 \leq 0$$

$$c = \begin{bmatrix} 1 - x_2^3 * x_3/(71785 * x_1^4) \\ 1 - 140.45 \frac{x_1}{x_2^2 x_3} \\ \frac{(4x_1 - x_2 x_1)}{12566(x_2^3 x_1 - x_2^4)} + \frac{1}{5108 x_2^2} - 1 \end{bmatrix}$$

$$\frac{x_1 + x_2}{1.5} - 1 \leq 0$$

$$A, b = \begin{bmatrix} 0.667 & 0.667 \end{bmatrix}, \begin{bmatrix} 1 \end{bmatrix}$$

$$0.05 \leq x_2 \leq 2$$

$$0.25 \leq x_1 \leq 1.3$$

$$2 \leq x_3 \leq 15$$

$$lb, ub = \begin{bmatrix} 0.05 \\ 0.25 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 1.3 \\ 15 \end{bmatrix}$$

Nonlinear Programming: Another Example

You should see the following approximate values for xopt and fopt.

- xopt $=$ [0.052, 0.036, 10.95]
- fopt $=$ 0.0127

Nonlinear Programming: Modelling and Solving

Design a circular tank, closed at both ends, with a volume of 200 m^3.The cost is proportional to the surface area of material, which is priced at $\$400/m^2$. The tank is contained within a shed with a sloping roof, thus the height of the tank h is limited by

$$h \leq 12 - \frac{d}{2}$$

where d is the tank diameter. Formulate the minimum cost problem and solve the design problem.

Nonlinear Programming: Modelling and Solving

Design a circular tank, closed at both ends, with a volume of 200 m^3.The cost is proportional to the surface area of material, which is priced at \$400/m^2. The tank is contained within a shed with a sloping roof, thus the height of the tank h is limited by

$$h \leq 12 - \frac{d}{2}$$

where d is the tank diameter. Formulate the minimum cost problem and solve the design problem.

Approach the problem in the following manner:

- Develop a problem formulation on paper.
- Identify the relevant functions and matrices.
- Solve it using `fmincon`.
- Does your answer make sense?

Nonlinear Programming: Modelling and Solving

You should see the following approximate values.

- radius $= 3.17$ m
- height $= 6.34$ m
- volume $= 200\ m^3$
- surface area $= 189.32\ m^2$
- cost $= \$75728$

FOSSEE: Getting Involved

- Text Book Companion: Porting solved examples from textbooks to a FOSS. Visit https://cloud.scilab.in/ to view the Scilab TBC on the cloud.
- Lab Migration: Migrating a course lab to a FOSS-only lab.
- Workshops on Python.

You are welcome to participate in FOSSEE activites, via the following avenues:

- FOSSEE Internships
- Preparing Text Book Companions using FOSS
- Joining FOSSEE as a project staff member

Please visit https://fossee.in/ to know more.