Optimization in Scilab

Manas Das & Anuradha Singhania

Department of Electrical Engineering
Indian Institute of Technology Bombay
FOSSEE

27 Dec, 2011

Outline

Today's focus

- Scilab is free.

Today's focus

- Scilab is free.
- Matrix/loops syntax is similar to Matlab.

Today's focus

- Scilab is free.
- Matrix/loops syntax is similar to Matlab.
- Scilab provides all basic and many advanced tools.

Today's focus

- Scilab is free.
- Matrix/loops syntax is similar to Matlab.
- Scilab provides all basic and many advanced tools.
- Today: Optimization in Scilab optim command

Optimization

- Minimize/maximize some (more than one) objective function by varying decision variable
- Subject to constraints on functions of decision variables
- All engineering domains, economics
- Vast area in itself

Single Objective Optimization Problems

$$\min_{x} \quad f(x) \tag{1}$$

where x is n dimensional, $f(x)$ is a scalar, with/without

- Bound constraints: $x_L \leq x \leq x_U$
- Linear/Nonlinear equality constraints: $g(x) = b$
- Linear/Nonlinear inequality constraints: $g(x) \leq b$

Components of x can be real, integer or binary.
Find 'Decision Variables': x which minimize 'Obj': f

Linear optimization with Linear constraints

- Scilab function: **karmarkar**
  Calling Sequence
  x1=karmarkar(a,b,c,lb)
  - `a:matrix (n,p)`
  - `b:n-vector`
  - `c:p-vector`
  - `lb:lower bound`
- **Also be solved in scilab using linpro function**
- **linpro is in Quapro toolbox under Optimization in ATOMS,**
- **It can installed by command atomsInstall('Quapro')**

Linear optimization with Linear constraints

Example:-

Maximize

$3x_1 + x_2 + 3x_3$

for

$2x_1 + x_2 + x_3 \leq 2$

$x_1 + 2x_2 + 3x_3 \leq 5$

$2x_1 + 2x_2 + x_3 \leq 6$

$x1, x2, x3 \geq 0$

Quadratic with linear constraints

- Scilab function: **qp_solve**
  Calling Sequence
  x1=**[x [,iact [,iter [,f]]]]=qp_solve(Q,p1,C1,b,me)**
  - Q:real positive definite symmetric matrix (dimension n x n). p- real (column) vector (dimension n)
  - c:p - vector
  - lb:lower bound
  - me: number of equality constraints
- **It can also be solved in scilab using quapro function.**
- **quapro is in Quapro toolbox under Optimization in ATOMS**
- **It can installed by command atomsInstall(quapro)**

Quadratic with linear constraints

Example Minimize

$$f(x) = \frac{1}{2} x^T \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} x + \begin{bmatrix} -2 \\ -2 \end{bmatrix}^T x$$

for

$x_1 + x_2 \geq 2 + \sqrt{2}$

$-x_1 + x_2 \geq -2$

Other Optimization Tools in Scilab: I

Native (built-in) functions in Scilab:

- fminsearch: Computes unconstrained minimum of a given function using the Nelder-Mead algorithm (simplex region search).
  - Native (built-in function) to Scilab:
  - Derivative free, relies on search
  - Basic usage: [xopt,fopt]=fminsearch(costf,x0); costf should take x as input and return f as output
  - Try minimizing $f(x) = x^2 + 5$ using fminsearch
- neldermead: Computes minimum of a nonlinear function subject to bounds and nonlinear constraints
  - Option for several types of search algorithms

Unconstrained Non-linear

- Scilab function: **fminsearch**
  Calling Sequence
  `x = fminsearch ( costf , x0 )`
  - costf:The cost function. x0:The initial guess.
- computes the unconstrained minimimum of given function with the Nelder-Mead algorithm.

Unconstrained Non-linear

Example: $f(x) = x_1^2 + x_2^2 + x_1 x_2$

Optim command in Scilab

Simplified Call:
[fopt,xopt]=optim(costf,x0)
where

- *x*0: initial guess where minimum of f occurs,
- *fopt*: the optimum (minimum) value,
- *xopt*: where the optimum occurs,
- *costf*: user specified (scilab) function to compute objective function (f) and/or its gradient (g) depending on the input integer flag "ind" as:
  [f,g,ind]=costf(x,ind)

Simple example

$f(x) = x^2 + 10$
$g(x) = 2x$

- Minimize $f(x)$ starting from $x0 = 10$.
- Write a function (example "myfunction1.sci") which takes x, ind as input and returns f, g, ind as output.
- Use "optim" function to minimize.
- Answer: $fopt = 10, xopt = 0$.

Another example (Campbell et al., 2010)

$$f(x_1, x_2, x_3) = (x_1 - x_3)^2 + 3(x_1 + x_2 + x_3 - 1)^2 + (x_1 - x_3 + 1)^2$$

with gradient

$$\begin{aligned} g &= [\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}] \\ &= [2(x_1 - x_3) + 6(x_1 + x_2 + x_3 - 1) + 2(x_1 - x_3 + 1), \\ &\quad 6(x_1 + x_2 + x_3 - 1), -2(x_1 - x_3) - 2(x_1 - x_3 + 1) + 6(x_1 + x_2 + x \end{aligned}$$

Exercise plan

- Minimize the function starting with guess values $x0 = [1, 1, 1]$.
- Write a function (example "myfunction2.sci") which takes x, ind as input and returns f, g, ind as output.
- Use "optim" function to minimize.
- Answer: $xopt = [0.14, 0.27, 0.64]$, $fopt = 0.51$.

Other options with optim

Several other options/features with optim

- Several optimization algorithms: quasi-newton, conjugate gradient, etc.
- Gradient computation using finite differences: "NDcost" used with optim.
- Various stopping/diagnostic criteria: Maximum number of iterations, calls to costf, thresholds, etc.
- Use of ind to compute gradient only when required (save computational effort).
- help optim: to know more

Other Optimization Tools in Scilab: II

Native (built-in) functions in Scilab:
AI based algorithms

- Genetic algorithms: single objective optimization, multiobjective optimization, various implementations
- Simulated annealing

Optimization Toolboxes in Scilab

Can be installed easily (atoms or otherwise) or interfaces to them

- quapro: linear, quadratic programming (Q not necessarily positive definite)
- IPopt, fsqp, lp_solve, etc.

Check "http://atoms.scilab.org/categories/optimization"

Optimization Features available in Scilab

- Nonlinear optimization with the **optim** function
- Quadratic optimization with the **qpsolve** function
- Nonlinear least-square optimization with the **lsqrsolve** function
- Semidefinite programming with the **semidef** function
- Genetic algorithms with the **optim_ga** function
- Simulated annealing with the **optim_sa** function
- Linear matrix inequalities with the **lmisolver** function

Missing optimization features in Scilab

`A list of features which are not available in Scilab, but are available in toolboxes.`

- Integer parameter with linear objective solver and sparse matrices : currently available in **LPSOLVE** toolbox, based on the simplex method,
- Linear objective with sparse matrices : currently available in **LIPSOL**, based on interior points method,
- Nonlinear objective and non linear constraints : currently available in interface to **IPOPT** toolbox, based on interior point methods,
- Nonlinear objective and non linear constraints : currently available in interface to **CONMIN** toolbox, based on method of feasible directions,

References

- S. L. Campbell, J.P. Chancelier and R. Nikoukhah, "Modeling and Simulation in Scilab/Scicos", Springer, 2006.

Thank You