Week 4 – Wednesday

Mathematical Modeling (Math 420/620)

Paul J. Hurtado

16 Sept, 2015

Constrained Optimization

Ex: Maximize general $f(x)$ with general constraints. (**Hard!**)
Solution: Equivalent, unconstrained objective function?

Smooth Constraints:

Ex: Maximize continuous $f(x, y)$ w/ constraints $g_i(x, y) = c_i$
Solution: Use LaGrange (or KKT) Multipliers.

Constrained Optimization

Ex: Maximize general $f(x)$ with general constraints. (**Hard!**)
Solution: Equivalent, unconstrained objective function?

Smooth Constraints:

Ex: Maximize continuous $f(x, y)$ w/ constraints $g_i(x, y) = c_i$
Solution: Use LaGrange (or KKT) Multipliers.

Linear Constraints:

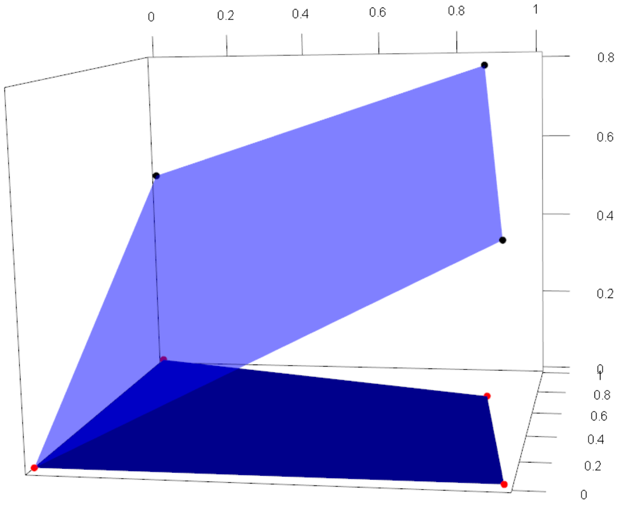
Ex: Maximize **linear** $f(x)$; linear constraints $g_i(x) \leq c_i$
Ex: Maximize **quadric** $f(x)$; linear constraints $g_i(x) \leq c_i$
Solution: Linear & Quadratic Programming, respectively

Example: Linear Programming

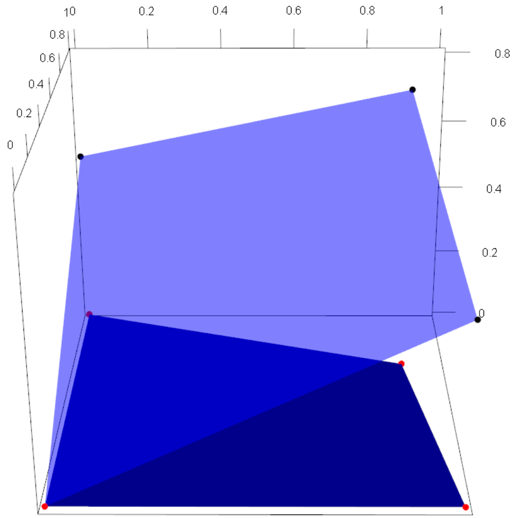
Code to plot example linear objective with linear constraints:

```
## Example of a linear objective with linear constraints
library(rgl)
x=c(0,  1, 0.9,  0,0)
y=c(0,  0, 0.7,  1,0)
z=c(0,0.5, 0.8,0.5,0)
polygon3d(x,y,z,lit=FALSE,col="blue",alpha=0.5,
          xlim=0:1, ylim=0:1, zlim=0:1)
polygon3d(x,y,z*0,lit=FALSE,col="darkblue")
points3d(x,y,rep(0,4),col="red",size=10)
points3d(x,y,z,col="black",size=10)
bbox3d(col="gray",alpha=0.5)
axes3d()
```

Example: Linear Programming



Example: Linear Programming



Gradient Methods

Examples: Broyden-Fletcher-Goldfarb-Shanno (BFGS) in R.

Step 1: Compute an approximate, or use-provided, Gradient (vector of partials).

Step 2: Compute an approximate, or use-provided, Hessian (matrix of 2nd order partials).

Step 3: Search along the line of steepest descent for a minimum.

Step 4: Choose a new point, and repeat.

Non-gradient Methods

Examples: Nelder-Mead in R.

Step 1: For a function of n variables, choose $n+1$ nearby points to form a *simplex*.

Step 2: Reflect the "worst" point through the centroid of the remaining n points.

Step 3: Stretch in that direction if it's better, contract if worse.

Step 4: Repeat steps 2-3 for the new simplex.

Nelder-Mead Simplex Algorithm

http://en.wikipedia.org/wiki/Nelder%E2%80%93Mead_method#/media/File:Nelder_Mead2.gif

Which Method?

How much do you know about your chosen objective surface?

How “smooth” is it?

How hard is it to compute objective function values?

How much computing power do you have?

Good rule of thumb: Methods like Nelder-Mead, that assume little about the objective surface, tend to work well on a broad range of problems!

Choosing Objective Functions

Homework Problem 1b and Bonus Problem