# ip

June 18, 2019

## 1 Interior Points methods

### 1.1 IPOpt

[1]:
```
using JuMP
using Ipopt
using LinearAlgebra
```

[ ]:
```
"""
Problem: Solve the Rosenbrock function
"""

# For more information see here:
# https://jump.readthedocs.org/en/latest/installation.html#getting-solvers

# Information about Ipopt options here:
# http://www.coin-or.org/Ipopt/documentation/node39.html

m = Model(with_optimizer(Ipopt.Optimizer))
# setsolver(m, IpoptSolver(tol = 1e-6, max_iter = 200, output_file = "results.
 ↪txt"))

@variable(m, x1)
@variable(m, x2)

@NLobjective(m, Min, 100(x2-x1^2)^2 + (1-x1)^2)

set_start_value(x1, -1.2)
set_start_value(x2, 1)

optimize!(m)

println("got ", objective_value(m), " at [", value(x1), ", ", value(x2), "]")
```

[ ]:
```
print(m)
```

[ ]:
```
m = Model(with_optimizer(Ipopt.Optimizer))
```

```
@variable(m, x[i=1:2])

@NLobjective(m, Min, 100(x[2]-x[1]^2)^2 + (1-x[1])^2)

set_start_value(x[1], -1.2)
set_start_value(x[2], 1)

println(m)

optimize!(m)

println("got ", objective_value(m), " at ", [value(x[1]), value(x[2])])
```

[ ]:
```
include("mle.jl")
```

[ ]:
```
n = 100

m = Model(with_optimizer(Ipopt.Optimizer))

@variable(m, x[i=1:n], start=(i/2))
@NLexpression(m, myexpr[i=1:n], sin(x[i]))
@NLconstraint(m, myconstr[i=1:n], myexpr[i] <= 0.5)
@NLobjective(m, Min, sum(x[i]^2-1 for i = 1:n))
```

[ ]:
```
m
```

[ ]:
```
optimize!(m)
```

[ ]:
```
objective_value(m)
```

[ ]:
```
sol = [value(x[i]) for i =1:100]
```

[ ]:
```
sin.(sol)
```

[ ]:
```
n = 100

m = Model(with_optimizer(Ipopt.Optimizer))

@variable(m, x[i=1:n], start=-1.0)
@NLexpression(m, myexpr[i=1:n], sin(x[i]))
@NLconstraint(m, myconstr[i=1:n], myexpr[i] <= 0.5)
@NLobjective(m, Min, sum(x[i]^2-1 for i = 1:n))
```

[ ]:
```
optimize!(m)
```

[ ]:
```
sol = [value(x[i]) for i =1:100]
```

[ ]:
```
objective_value(m)
```

[ ]:
```
sin.(sol)
```

## 1.2 Interior Points Methods for Linear Programming

Consider the program

$$\begin{aligned} \min_x \; & c^T x \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned}$$

The Lagrangian is

$$L(x, \lambda, \mu) = c^T x + \lambda^T (Ax - b) + \mu^T (-x)$$

and the duality bound is given by

$$\min_{x \in S} \max_{\mu \geq 0, \lambda} L(x, \lambda, \mu)$$

The KKT system becomes

$$\begin{aligned} c + A^T \lambda - \mu &= 0 \\ Ax &= b \\ x &\geq 0 \\ \mu_i x_i &= 0, \; i = 1, \ldots, n \\ \mu &\geq 0 \end{aligned}$$

Considering $\mu$ as a surplus vector, the first and fifth conditions can be rewritten as

$$c + A^T \lambda \geq 0$$

At $(x^*, \lambda^*, \mu^*)$,

$$L(x^*, \lambda^*, \mu^*) = c^T x^* + (\lambda^*)^T (Ax^* - b) - (\mu^*)^T x^* = c^T x^*$$

Moreover, the first KKT condition implies that at the primal-dual solution $(x^*, \lambda^*, \mu^*)$, the Lagrangian becomes

$$L(x^*, \lambda^*, \mu^*) = -b^T \lambda^*$$

as

$$\begin{aligned} L(x^*, \lambda^*, \mu^*) = c^T x^* + (\lambda^*)^T (Ax^* - b) - (\mu^*)^T x^* &= (c^T + (\lambda^*)^T A - (\mu^*)^T) x^* - (\lambda^*)^T b \\ &= (x^*)^T (c + A^T \lambda^* - \mu^*) - b^T \lambda^* \end{aligned}$$

Along with the duality bound, this means that we have to solve the problem

$$\begin{aligned} \max_\lambda \; & -b^T \lambda \\ \text{s.t. } & A^T \lambda \geq -c \end{aligned}$$

Setting $y = -\lambda$, the program can be rewritten as

$$\begin{aligned} \max_y \; & b^T y \\ \text{s.t. } & A^T y \leq c \end{aligned}$$

We retrieve the well-known dual problem is linear programming.

If we apply the logarithmic barrier to the primal linear program, we can rewrite it as

$$\begin{aligned} \min\ & c^T x - \mu \sum_{i=1}^{n} \ln x_i \\ \text{s.t. } & Ax = b, \end{aligned}$$

and the barrier Lagrangian is

$$L(x, \lambda) = c^T x + \lambda^T (Ax - b) - \mu \sum_{i=1}^{n} \ln x_i$$

and the KKT conditions are now

$$\begin{aligned} c + A^T \lambda - \mu X^{-1} e &= 0 \\ Ax &= b \end{aligned}$$

Set $s = \mu X^{-1} e$. We can rewrite the conditions as

$$\begin{aligned} c + A^T \lambda - s &= 0 \\ Ax &= b \\ s &= \mu X^{-1} e \end{aligned}$$

or

$$\begin{aligned} c + A^T \lambda - s &= 0 \\ Ax &= b \\ s_i x_i &= \mu \end{aligned}$$

The logarithmic barrier is equivalent to perturb the complementarity condition of the KKT system.

As before, set $y = -\lambda$. Solving the KKT system is equivalent to look for a solution of the linear system

$$\begin{aligned} A^T y + s - c &= 0 \\ Ax - b &= 0 \\ SXe - \mu e &= 0 \end{aligned}$$

The corresponding Newton equation is then

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = - \begin{pmatrix} A^T y + s - c \\ Ax - b \\ SXe - \mu e \end{pmatrix}$$

### 1.2.1 Example: the farmer problem

Adapted from https://www.sonoma.edu/users/w/wilsonst/Courses/Math_131/lp/Farm.html

A farmer has 10 acres to plant in wheat and rye. He has to plant at least 7 acres. However, he has only CAD1200 to spend and each acre of wheat costs CAD200 to plant and each acre of rye costs CAD100 to plant. Moreover, the farmer has to get the planting done in 12 hours and it takes an hour to plant an acre of wheat and 2 hours to plant an acre of rye. If the profit is CAD500 per acre of wheat and CAD300 per acre of rye how many acres of each should be planted to maximize profits?