In [ ]:
```
#Problem 1
```

In [2]:
```
using JuMP, Clp

m = Model(solver=ClpSolver())

@variable(m, x1 >= 0)
@variable(m, x2 >= 0)
@variable(m, x3)
@variable(m, x4 >= 0)

@objective(m, Min, x1 + x2 - 2*x3 + (1/3)x4)

@constraint(m, 5 * x1 - 3 * x2 + x4 == 7)
@constraint(m, 2*x1 + x2 + x3 >= 2)
@constraint(m, x2 + (1/2)x3 <= x1)

println(@time(solve(m)))

println("Value of X1: ", getvalue(x1))
println("Value of X2: ", getvalue(x2))
println("Value of X3: ", getvalue(x3))
println("Value of X4: ", getvalue(x4))
```

```
  0.026020 seconds (70 allocations: 5.422 KiB)
Optimal
Value of X1: 1.4
Value of X2: 0.0
Value of X3: 2.8
Value of X4: 0.0
```

In [3]:
```
using JuMP, ECOS

m = Model(solver=ECOSSolver())

@variable(m, x1 >= 0)
@variable(m, x2 >= 0)
@variable(m, x3)
@variable(m, x4 >= 0)

@objective(m, Min, x1 + x2 - 2*x3 + (1/3)x4)

@constraint(m, 5 * x1 - 3 * x2 + x4 == 7)
@constraint(m, 2*x1 + x2 + x3 >= 2)
@constraint(m, x2 + (1/2)x3 <= x1)

println(@time(solve(m)))

println("Value of X1: ", getvalue(x1))
println("Value of X2: ", getvalue(x2))
println("Value of X3: ", getvalue(x3))
println("Value of X4: ", getvalue(x4))
```

```
  0.001207 seconds (535 allocations: 34.375 KiB)
Optimal
Value of X1: 1.399999999984053
Value of X2: -3.143276621278021e-12
Value of X3: 2.8000000000083842
Value of X4: 7.030730957563056e-11

ECOS 2.0.5 - (C) embotech GmbH, Zurich Switzerland, 2012-15. Web: www.embotec
h.com/ECOS

It     pcost       dcost      gap   pres   dres    k/t    mu     step   sigma
IR    |   BT
 0  -4.716e+00  -1.499e+00  +4e+01  5e-01  5e-01  1e+00  7e+00    ---    ---
1  1  - |  -  -
 1  -4.213e+00  -3.640e+00  +3e+00  3e-02  3e-02  4e-01  6e-01  0.9485  4e-02
1  1  1 |  0  0
 2  -4.201e+00  -4.193e+00  +3e-02  3e-04  4e-04  6e-03  8e-03  0.9874  2e-04
1  1  1 |  0  0
 3  -4.200e+00  -4.200e+00  +4e-04  3e-06  4e-06  6e-05  9e-05  0.9890  1e-04
1  1  1 |  0  0
 4  -4.200e+00  -4.200e+00  +4e-06  4e-08  5e-08  7e-07  9e-07  0.9890  1e-04
1  0  0 |  0  0
 5  -4.200e+00  -4.200e+00  +5e-08  4e-10  5e-10  8e-09  1e-08  0.9890  1e-04
1  0  0 |  0  0
 6  -4.200e+00  -4.200e+00  +5e-10  5e-12  6e-12  9e-11  1e-10  0.9890  1e-04
1  0  0 |  0  0

OPTIMAL (within feastol=5.6e-12, reltol=1.2e-10, abstol=5.0e-10).
Runtime: 0.000240 seconds.
```

In [4]:
```
using JuMP, SCS

m = Model(solver=SCSSolver())

@variable(m, x1 >= 0)
@variable(m, x2 >= 0)
@variable(m, x3)
@variable(m, x4 >= 0)

@objective(m, Min, x1 + x2 - 2*x3 + (1/3)x4)

@constraint(m, 5 * x1 - 3 * x2 + x4 == 7)
@constraint(m, 2*x1 + x2 + x3 >= 2)
@constraint(m, x2 + (1/2)x3 <= x1)

println(@time(solve(m)))

println("Value of X1: ", getvalue(x1))
println("Value of X2: ", getvalue(x2))
println("Value of X3: ", getvalue(x3))
println("Value of X4: ", getvalue(x4))
```

```
INFO: Precompiling module SCS.

 13.122613 seconds (2.12 M allocations: 108.350 MiB, 0.97% gc time)
Optimal
Value of X1: 1.3999999889060737
Value of X2: -1.9221557968958916e-9
Value of X3: 2.7999999853096393
Value of X4: 7.459669340021949e-9
------------------------------------------------------------------------------
        SCS v2.0.2 - Splitting Conic Solver
        (c) Brendan O'Donoghue, Stanford University, 2012-2017
------------------------------------------------------------------------------
Lin-sys: sparse-indirect, nnz in A = 12, CG tol ~ 1/iter^(2.00)
eps = 1.00e-05, alpha = 1.50, max_iters = 5000, normalize = 1, scale = 1.00
acceleration_lookback = 20, rho_x = 1.00e-03
Variables n = 4, constraints m = 6
Cones:  primal zero / dual free vars: 1
        linear vars: 5
Setup time: 1.97e-03s
------------------------------------------------------------------------------
 Iter | pri res | dua res | rel gap | pri obj | dua obj | kap/tau | time (s)
------------------------------------------------------------------------------
     0| 1.00e+19  3.02e+19  1.00e+00 -2.10e+20  5.08e+19  6.09e+19  3.24e-03
    40| 5.29e-09  1.15e-08  3.75e-09 -4.20e+00 -4.20e+00  4.50e-16  8.75e-03
------------------------------------------------------------------------------
Status: Solved
Timing: Solve time: 8.77e-03s
        Lin-sys: avg # CG iterations: 1.76, avg solve time: 2.48e-06s
        Cones: avg projection time: 6.07e-07s
        Acceleration: avg step time: 1.28e-04s
------------------------------------------------------------------------------
Error metrics:
dist(s, K) = 2.2536e-16, dist(y, K*) = 0.0000e+00, s'y/|s||y| = -3.9264e-17
primal res: |Ax + s - b|_2 / (1 + |b|_2) = 5.2908e-09
dual res:   |A'y + c|_2 / (1 + |c|_2) = 1.1530e-08
rel gap:    |c'x + b'y| / (1 + |c'x| + |b'y|) = 3.7493e-09
------------------------------------------------------------------------------
c'x = -4.2000, -b'y = -4.2000
==============================================================================
```

In [5]:
```
# The difference in runtime seems to be the result of the differences in preci
sion of the different solvers. Clp had results of
# only two significant figures, but it ran the fastest. Clp was followed by EC
OS, then SCS in runtime; Precision increase
# followed the same order.
```

In [ ]:
```
# Problem 2
```

In [2]:
```
using JuMP, Clp

m = Model(solver=ClpSolver())

@variable(m, iron >= 0)
@variable(m, leather >= 0)

@objective(m, Max, 7*iron + 4*leather) # Total Protection per m^2

@constraint(m, 9*iron + 6*leather <= 150) # Gold constraint
@constraint(m, 5*iron + 2*leather <= 80) # Weight constraint

println(@time(solve(m)))

println("Square meters of leather: ", getvalue(leather))
println("Square meters of iron: ", getvalue(iron))
println("Total armor protection: ", getobjectivevalue(m))
```

```
  0.023016 seconds (70 allocations: 4.797 KiB)
Optimal
Square meters of leather: 2.499999999999991
Square meters of iron: 15.000000000000004
Total armor protection: 115.0
```

In [1]:
```
# Max protection parameters

materials = [:iron, :leather] # Material types

materialCost = Dict(:iron => 9, :leather => 6) # Cost per meters squared per m
aterial

materialWeight = Dict(:iron => 5, :leather => 2) # Weight per meter squared of
material

materialProtection = Dict(:iron => 7, :leather => 4) # protect per square mete
r

# Fixed values

playerGold = 150 # Max gold available
carryWeight = 80; # Max carry weight of character
```

In [4]:
```
using JuMP, Clp

m = Model(solver=ClpSolver())

@variable(m, material[materials] >= 0) # material variable

@objective(m, Max, sum(materialProtection[i] * material[i] for i in materials
))

@constraint(m, sum(materialCost[i] * material[i] for i in materials) <= player
Gold) # Gold constraint
@constraint(m, sum(materialWeight[i] * material[i] for i in materials) <= carr
yWeight) # weight restriction

status = solve(m)

println("Total protection: ", getobjectivevalue(m))
println("Iron used: ", getvalue(material[:iron]))
println("Leather used: ", getvalue(material[:leather]))
```

```
Total protection: 115.0
Iron used: 15.000000000000004
Leather used: 2.499999999999991
```

In [ ]:
```
# Problem 3
```

In [6]:
```
using JuMP, Clp

m = Model(solver=ClpSolver())

@variable(m, a >= 0)
@variable(m, b >= 0)
@variable(m, c >= 0)
@variable(m, x >= 0)

@objective(m, Max, ((x+7a-7b-c+16))) # Standard LP eq found above

@constraint(m, c <= 10) # New constraints
@constraint(m, x <= 7)
@constraint(m, a-b <= x+1)
@constraint(m, x-a+b+(c/2) == 10.5)

solution = solve(m)

println("Value of a: ", getvalue(a))
println("Value of b: ", getvalue(b))
println("Value of c: ", getvalue(c))
println("Value of x: ", getvalue(x))
println("Objective Value: ", (-1 * getobjectivevalue(m))) # Adjusted for Min -
> -Max(-1(equation))
```

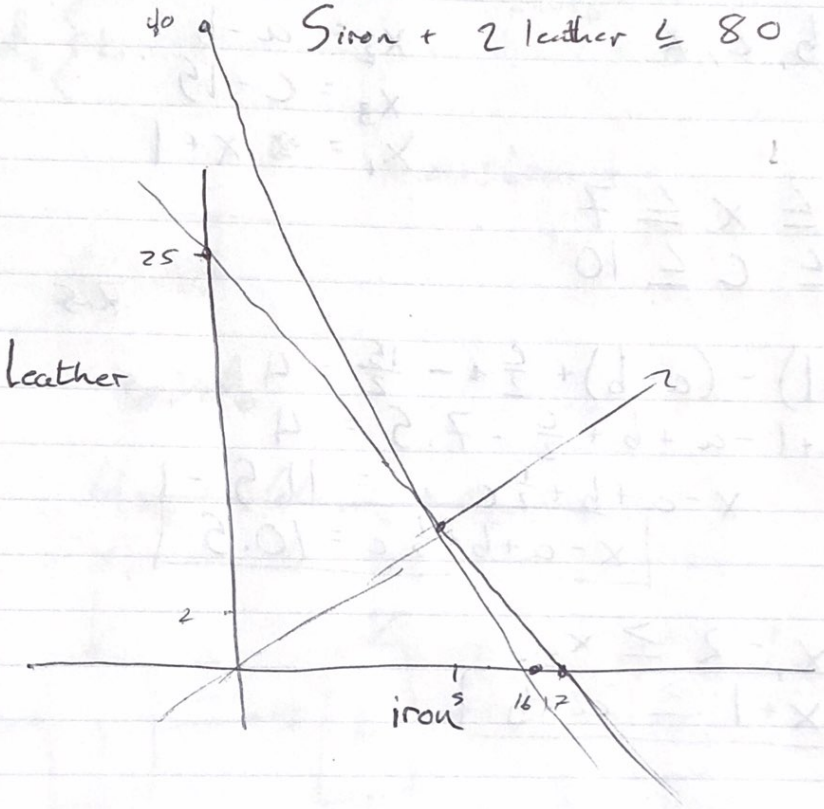
```
Value of a: 1.5
Value of b: 0.0
Value of c: 10.0
Value of x: 7.0
Min of LP: -23.5
```

Problem 2

c) $9\,\text{iron} + 6\,\text{leather} \leq 150$

$\rightarrow 3\,\text{iron} + 2\,\text{leather} \leq 50$

$5\,\text{iron} + 2\,\text{leather} \leq 80$



$\text{iron} \approx 15$

$\text{leather} \approx 2$

Problem 3

a) $-\text{Max}\,(-1)(x_1 + 7x_2 - x_3)$

A, b, c, x

$x_2 = a - b$
$x_3 = c - 15$
$x_1 = x + 1$

$0 \le x \le 7$
$0 \le c \le 10$

$(x+1) - (a-b) + \frac{c}{2} - \frac{15}{2} = 4$
$x + 1 - a + b + \frac{c}{2} - 7.5 = 4$
$x - a + b + \frac{1}{2}c = 11.5 - 1$
$\boxed{x - a + b + \frac{1}{2}c = 10.5}$

$x_1 - 2 \ge x_2$
$\boxed{x + 1 \ge a - b}$

b) $\Rightarrow -\underset{A,b,c,x}{\text{Max}}\,(-1)(x + 1 + 7a - 7b - c + 15)$

$a, b \ge 0$
$0 \le x \le 7$
$0 \le c \le 10$

$x_1 = x + 1$
$x_2 = a - b$
$x_3 = c - 15$

$x - a + b + \frac{1}{2}c = 10.5$
$x + 1 \ge a - b$