In [1]:
```
using DataFrames
using Ipopt
using JuMP
using ImplicitEquations, Plots
```

```
WARNING: Method definition midpoints(Base.Range{T} where T) in module Base at deprecated.jl:56 overwritt
en in module StatsBase at /home/boturon/.julia/v0.6/StatsBase/src/hist.jl:535.
WARNING: Method definition midpoints(AbstractArray{T, 1} where T) in module Base at deprecated.jl:56 ove
rwritten in module StatsBase at /home/boturon/.julia/v0.6/StatsBase/src/hist.jl:533.
```

Formulation and Graphical Solution

1.

An aviary wants the fowl’s food to have a certain quantity of nutrients. Two kinds of grain are available: A and B. The data are as follows:

Kind of grain	Starch	Proteins	Vitamins	Cost/kg
A	4	5	2	6.0
B	5	7	1	4.0
Minimum quantity to obtain	8	14	3	

What is the ideal composition of the fowl’s food in order to minimize the cost?

In [2]:
```
m_1 = Model(solver=IpoptSolver())
@variable(m_1, A_1 >= 0 )
@variable(m_1, B_1 >= 0)
@constraint(m_1, 8 <= 4A_1+5B_1)
@constraint(m_1, 14 <= 5A_1+7B_1)
@constraint(m_1, 3 <= 2A_1+B_1)
@objective(m_1, Min, 6A_1 + 4B_1)

print(m_1)
```

```
Min 6 A_1 + 4 B_1
Subject to
 -4 A_1 - 5 B_1 ≤ -8
 -5 A_1 - 7 B_1 ≤ -14
 -2 A_1 - B_1 ≤ -3
 A_1 ≥ 0
 B_1 ≥ 0
```

In [3]:
```
solve(m_1)
```

```
******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear optimization.
 Ipopt is released as open source code under the Eclipse Public License (EPL).
         For more information visit http://projects.coin-or.org/Ipopt (http://projects.coin-or.org/Ipop
t)
******************************************************************************

This is Ipopt version 3.12.1, running with linear solver mumps.
NOTE: Other linear solvers might be more efficient (see Ipopt documentation).

Number of nonzeros in equality constraint Jacobian...:        0
Number of nonzeros in inequality constraint Jacobian.:        6
Number of nonzeros in Lagrangian Hessian.............:        0

Total number of variables............................:        2
                     variables with only lower bounds:        2
                variables with lower and upper bounds:        0
                     variables with only upper bounds:        0
Total number of equality constraints.................:        0
Total number of inequality constraints...............:        3
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        3

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  9.9999900e-02 1.39e+01 1.04e+00  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  6.4244186e-01 1.32e+01 5.03e+00  -1.0 2.39e+00    -  1.03e-02 5.80e-02h  1
   2  4.1883368e+00 8.38e+00 5.95e+00  -1.0 2.25e+00    -  5.24e-02 3.64e-01h  1
   3  1.0623383e+01 0.00e+00 1.00e-06  -1.0 1.47e+00    -  1.00e+00 1.00e+00h  1
   4  1.0484370e+01 0.00e+00 2.00e-07  -1.7 5.78e-02    -  1.00e+00 1.00e+00f  1
   5  1.0444777e+01 0.00e+00 4.68e-04  -3.8 9.70e-02    -  1.00e+00 9.83e-01f  1
   6  1.0444448e+01 0.00e+00 1.85e-11  -5.7 1.34e-04    -  1.00e+00 1.00e+00f  1
   7  1.0444444e+01 0.00e+00 2.57e-14  -8.6 8.28e-06    -  1.00e+00 1.00e+00f  1

Number of Iterations....: 7

                                   (scaled)                 (unscaled)
Objective...............:   1.0444444345006207e+01    1.0444444345006207e+01
Dual infeasibility......:   2.5652263423313793e-14    2.5652263423313793e-14
Constraint violation....:   0.0000000000000000e+00    0.0000000000000000e+00
Complementarity.........:   2.5104998670999341e-09    2.5104998670999341e-09
Overall NLP error.......:   2.5104998670999341e-09    2.5104998670999341e-09


Number of objective function evaluations             = 8
Number of objective gradient evaluations             = 8
Number of equality constraint evaluations            = 0
Number of inequality constraint evaluations          = 8
Number of equality constraint Jacobian evaluations   = 0
Number of inequality constraint Jacobian evaluations = 8
Number of Lagrangian Hessian evaluations             = 7
Total CPU secs in IPOPT (w/o function evaluations)   =      0.227
Total CPU secs in NLP function evaluations           =      0.230

EXIT: Optimal Solution Found.
```

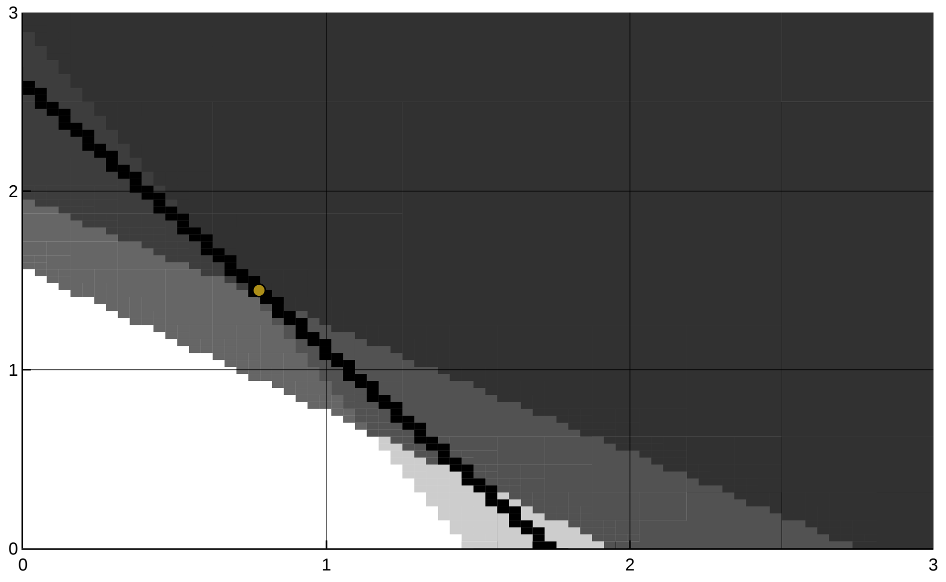
Out[3]:
```
:Optimal
```

In [4]:
```
println("Objective value: ", getobjectivevalue(m_1))
println("A = ", getvalue(A_1))
println("B = ", getvalue(B_1))
```

```
Objective value: 10.444444345006207
A = 0.7777777695463124
B = 1.4444444319320833
```

In [5]:
```
C_1_1(A,B) = 4A+5B
C_2_1(A,B) = 5A+7B
C_3_1(A,B) = 2A+B
O_1(A,B) = 6A+4B
plot((C_1_1 ≧ 8), alpha=0.6)
plot!(C_2_1 ≧ 14, alpha=0.4)
plot!(C_3_1 ≧ 3, alpha=0.2)
plot!(O_1 == getobjectivevalue(m_1), alpha=1)
scatter!([getvalue(A_1)],[getvalue(B_1)], xlims=(0,3), ylims=(0,3))
```

Out[5]:



2.

The marketing department of a company that produces metal office furniture suggests the launch of two new products: a desk and a shelf, in order to substitute current models. This department doesn't foresee any difficulty in the placement of the new shelf model in the market, but advises that the monthly production of desks shouldn't exceed 160 units. After some research conducted by the production department, we know that:

- The monthly availability of the stamping department is 720 machine hours;
- The monthly availability of the assembly and finishing department is 880 man hours;
- Each desk takes 2 hours of stamping and 4 hours of assembly and finishing;
- Each shelf takes 4 hours of stamping and 4 hours of assembly and finishing;
- The estimated gross margins per unit are 60€ for desks and 30€ for shelves.

a) Formulate the problem in order to determine the monthly production plan that maximizes the gross margin for these two new products.

In [6]:
```
m_2 = Model(solver=IpoptSolver())
@variable(m_2, D_2 >= 0 )
@variable(m_2, S_2 >= 0)
@constraint(m_2, 2D_2+4S_2 <= 720)
@constraint(m_2, 4D_2+4S_2 <= 880)
@constraint(m_2, D_2 <= 160)
@objective(m_2, Max, 60D_2+30S_2)

print(m_2)
```

```
Max 60 D_2 + 30 S_2
Subject to
 2 D_2 + 4 S_2 ≤ 720
 4 D_2 + 4 S_2 ≤ 880
 D_2 ≤ 160
 D_2 ≥ 0
 S_2 ≥ 0
```

In [7]:
```
solve(m_2)
```

```
This is Ipopt version 3.12.1, running with linear solver mumps.
NOTE: Other linear solvers might be more efficient (see Ipopt documentation).

Number of nonzeros in equality constraint Jacobian...:        0
Number of nonzeros in inequality constraint Jacobian.:        5
Number of nonzeros in Lagrangian Hessian.............:        0

Total number of variables............................:        2
                     variables with only lower bounds:        2
                variables with lower and upper bounds:        0
                     variables with only upper bounds:        0
Total number of equality constraints.................:        0
Total number of inequality constraints...............:        3
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        3

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0 -8.9999910e-01 0.00e+00 1.17e+01  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1 -5.4868280e+01 0.00e+00 6.00e+01  -1.0 4.40e+00    -  1.62e-02 1.00e+00f  1
   2 -4.3727446e+03 0.00e+00 5.78e+01  -1.0 5.49e+03    -  5.12e-04 5.23e-02f  1
   3 -9.5478552e+03 0.00e+00 5.41e+01  -1.0 5.33e+03    -  1.78e-01 6.47e-02f  1
   4 -9.5996901e+03 0.00e+00 5.08e+01  -1.0 5.73e+01    -  1.45e-01 6.04e-02f  1
   5 -9.6462086e+03 0.00e+00 3.88e+01  -1.0 2.60e+01    -  1.61e-03 2.37e-01f  1
   6 -1.1382360e+04 0.00e+00 2.36e+01  -1.0 5.92e+02    -  1.00e+00 3.91e-01f  1
   7 -1.1399724e+04 0.00e+00 6.60e+00  -1.0 3.21e+00    -  1.00e+00 7.20e-01f  1
   8 -1.1399801e+04 0.00e+00 1.00e-06  -1.0 1.01e-02    -  1.00e+00 1.00e+00f  1
   9 -1.1399994e+04 0.00e+00 2.83e-08  -2.5 1.29e-02    -  1.00e+00 1.00e+00f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
  10 -1.1400000e+04 0.00e+00 1.50e-09  -3.8 3.57e-04    -  1.00e+00 1.00e+00f  1
  11 -1.1400000e+04 0.00e+00 1.84e-11  -5.7 1.98e-05    -  1.00e+00 1.00e+00f  1
  12 -1.1400000e+04 0.00e+00 3.00e-14  -8.6 2.46e-07    -  1.00e+00 1.00e+00f  1

Number of Iterations....: 12

                                   (scaled)                 (unscaled)
Objective...............:  -1.1400000113994987e+04   -1.1400000113994987e+04
Dual infeasibility......:   2.9952483815378195e-14    2.9952483815378195e-14
Constraint violation....:   0.0000000000000000e+00    0.0000000000000000e+00
Complementarity.........:   2.5059421204791552e-09    2.5059421204791552e-09
Overall NLP error.......:   2.5059421204791552e-09    2.5059421204791552e-09


Number of objective function evaluations             = 13
Number of objective gradient evaluations             = 13
Number of equality constraint evaluations            = 0
Number of inequality constraint evaluations          = 13
Number of equality constraint Jacobian evaluations   = 0
Number of inequality constraint Jacobian evaluations = 13
Number of Lagrangian Hessian evaluations             = 12
Total CPU secs in IPOPT (w/o function evaluations)   =      0.005
Total CPU secs in NLP function evaluations           =      0.000

EXIT: Optimal Solution Found.
```

Out[7]:
```
:Optimal
```

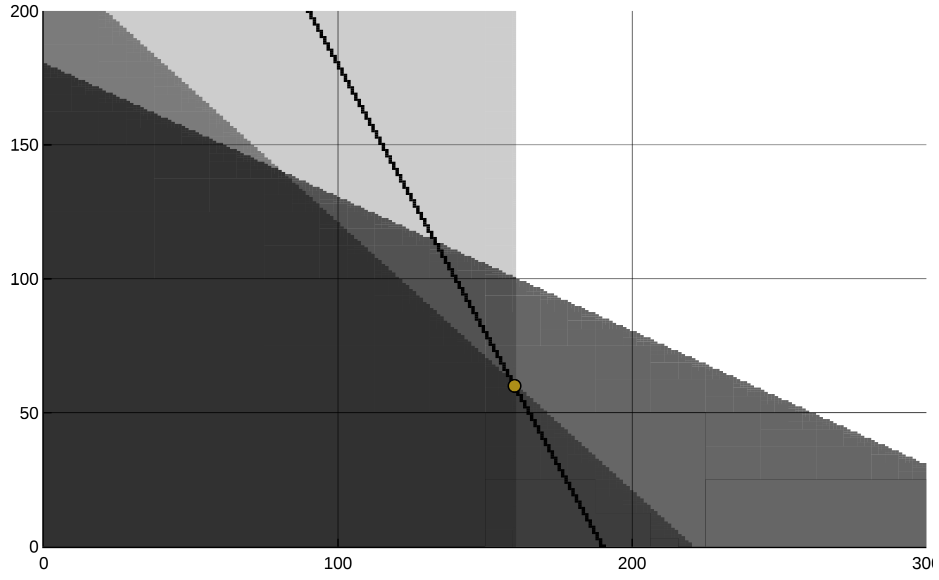
In [8]:
```
println("Objective value: ", getobjectivevalue(m_2))
println("Desk = ", getvalue(D_2))
println("Shelf = ", getvalue(S_2))
```

```
Objective value: 11400.000113994987
Desk = 160.00000159991646
Shelf = 60.00000060000003
```

b) Graphically find the optimal solution of the problem.

In [9]:
```
C_1_2(D,S) = 2D+4S
C_2_2(D,S) = 4D+4S
C_3_2(D,S) = D
O_2(D,S) = 60D+30S
plot(C_1_2 ≦ 720, alpha=0.6, xlims=(0,300), ylims=(0,200))
plot!(C_2_2 ≦ 880, alpha=0.4, xlims=(0,300), ylims=(0,200))
plot!(C_3_2 ≦ 160, alpha=0.2, xlims=(0,300), ylims=(0,200))
plot!(O_2 == getobjectivevalue(m_2), alpha=1, xlims=(0,300), ylims=(0,200))
scatter!([getvalue(D_2)],[getvalue(S_2)], xlims=(0,300), ylims=(0,200))
```

Out[9]:



3.

A production manager is planning the production of three products. For pursuing this there are four available machines. All products can be produced in all machines. The unitary production cost (in euros) is as follows:

Product	M1	M2	M3	M4
1	4	4	5	7
2	6	7	5	6
3	12	10	8	11

The time (in minutes) required for each machine to produce one unit of each product is:

Product	M1	M2	M3	M4
1	0.3	0.25	0.2	0.2
2	0.2	0.3	0.2	0.25
3	0.8	0.6	0.6	0.5

Assume that the demand for products 1, 2 and 3 is, respectively, 4.000, 5.000 and 3.000 units. For safety reasons, machines 1, 2 and 3 cannot work more than 1.500, 1.200 and 2.000 minutes, respectively.

Formulate the problem in order to minimize the total production cost.

```
In [10]: m_3 = Model(solver=IpoptSolver())
         @variable(m_3, P1M1_3 >= 0)
         @variable(m_3, P2M1_3 >= 0)
         @variable(m_3, P3M1_3 >= 0)
         @variable(m_3, P1M2_3 >= 0)
         @variable(m_3, P2M2_3 >= 0)
         @variable(m_3, P3M2_3 >= 0)
         @variable(m_3, P1M3_3 >= 0)
         @variable(m_3, P2M3_3 >= 0)
         @variable(m_3, P3M3_3 >= 0)
         @variable(m_3, P1M4_3 >= 0)
         @variable(m_3, P2M4_3 >= 0)
         @variable(m_3, P3M4_3 >= 0)
         @constraint(m_3, P1M1_3*.3 + P2M1_3*.2 + P3M1_3*.8 <= 1500)
         @constraint(m_3, P1M2_3*.25 + P2M2_3*.3 + P3M2_3*.6 <= 1200)
         @constraint(m_3, P1M3_3*.2 + P2M3_3*.2 + P3M3_3*.6 <= 2000)
         @constraint(m_3, P1M1_3 + P1M2_3 + P1M3_3 + P1M4_3 == 4000)
         @constraint(m_3, P2M1_3 + P2M2_3 + P2M3_3 + P2M4_3 == 5000)
         @constraint(m_3, P3M1_3 + P3M2_3 + P3M3_3 + P3M4_3 == 3000)
         @objective(m_3, Min, (P1M1_3*4+P2M1_3*6+P3M1_3*12)+(P1M2_3*4+P2M2_3*7+P3M2_3*10)+(P1M3_3*5+P2M3_3*5+P3M3_

         print(m_3)
```

```
Min 4 P1M1_3 + 6 P2M1_3 + 12 P3M1_3 + 4 P1M2_3 + 7 P2M2_3 + 10 P3M2_3 + 5 P1M3_3 + 5 P2M3_3 + 8 P3M3_3 +
7 P1M4_3 + 6 P2M4_3 + 11 P3M4_3
Subject to
 0.3 P1M1_3 + 0.2 P2M1_3 + 0.8 P3M1_3 ≤ 1500
 0.25 P1M2_3 + 0.3 P2M2_3 + 0.6 P3M2_3 ≤ 1200
 0.2 P1M3_3 + 0.2 P2M3_3 + 0.6 P3M3_3 ≤ 2000
 P1M1_3 + P1M2_3 + P1M3_3 + P1M4_3 = 4000
 P2M1_3 + P2M2_3 + P2M3_3 + P2M4_3 = 5000
 P3M1_3 + P3M2_3 + P3M3_3 + P3M4_3 = 3000
 P1M1_3 ≥ 0
 P2M1_3 ≥ 0
 P3M1_3 ≥ 0
 P1M2_3 ≥ 0
 P2M2_3 ≥ 0
 P3M2_3 ≥ 0
 P1M3_3 ≥ 0
 P2M3_3 ≥ 0
 P3M3_3 ≥ 0
 P1M4_3 ≥ 0
 P2M4_3 ≥ 0
 P3M4_3 ≥ 0
```

```
In [11]: solve(m_3)
```

```
This is Ipopt version 3.12.1, running with linear solver mumps.
NOTE: Other linear solvers might be more efficient (see Ipopt documentation).

Number of nonzeros in equality constraint Jacobian...:       12
Number of nonzeros in inequality constraint Jacobian.:        9
Number of nonzeros in Lagrangian Hessian.............:        0

Total number of variables............................:       12
                     variables with only lower bounds:       12
                variables with lower and upper bounds:        0
                     variables with only upper bounds:        0
Total number of equality constraints.................:        3
Total number of inequality constraints...............:        3
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        3

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  8.4999915e-01 5.00e+03 1.82e+00  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  8.0749832e+04 9.09e-13 1.25e+05  -1.0 1.25e+03    -  7.92e-06 1.00e+00h  1
   2  7.4629030e+04 9.09e-13 1.15e+05  -1.0 2.38e+04    -  4.95e-02 7.66e-02f  1
   3  7.4295768e+04 9.09e-13 1.14e+05  -1.0 1.62e+04    -  1.66e-01 1.05e-02f  1
   4  7.1087460e+04 9.09e-13 9.61e+04  -1.0 1.05e+04    -  1.70e-01 1.58e-01f  1
   5  7.0441703e+04 4.55e-13 8.76e+04  -1.0 2.24e+03    -  3.91e-01 8.89e-02f  1
   6  6.8186796e+04 4.55e-13 4.24e+04  -1.0 1.00e+03    -  4.95e-01 5.16e-01f  1
   7  6.7766935e+04 9.09e-13 1.27e+04  -1.0 2.16e+02    -  7.98e-01 7.00e-01f  1
   8  6.7668406e+04 0.00e+00 6.51e+02  -1.0 4.70e+01    -  9.97e-01 9.49e-01f  1
   9  6.7667466e+04 9.09e-13 1.00e-06  -1.0 1.27e+02    -  1.00e+00 1.00e+00f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
  10  6.7666689e+04 4.55e-13 2.83e-08  -2.5 1.27e+00    -  1.00e+00 1.00e+00f  1
  11  6.7666668e+04 9.09e-13 1.50e-09  -3.8 2.92e+00    -  1.00e+00 1.00e+00f  1
  12  6.7666667e+04 4.55e-13 1.85e-11  -5.7 6.46e-01    -  1.00e+00 1.00e+00f  1
  13  6.7666667e+04 4.55e-13 2.62e-14  -8.6 7.08e-02    -  1.00e+00 1.00e+00f  1

Number of Iterations....: 13

                                   (scaled)                 (unscaled)
Objective...............:   6.7666666599923366e+04    6.7666666599923366e+04
Dual infeasibility......:   2.6201263381153694e-14    2.6201263381153694e-14
Constraint violation....:   4.5474735088646412e-13    4.5474735088646412e-13
Complementarity.........:   2.6842170028222575e-09    2.6842170028222575e-09
Overall NLP error.......:   2.6842170028222575e-09    2.6842170028222575e-09


Number of objective function evaluations             = 14
Number of objective gradient evaluations             = 14
Number of equality constraint evaluations            = 14
Number of inequality constraint evaluations          = 14
Number of equality constraint Jacobian evaluations   = 14
Number of inequality constraint Jacobian evaluations = 14
Number of Lagrangian Hessian evaluations             = 13
Total CPU secs in IPOPT (w/o function evaluations)   =      0.007
Total CPU secs in NLP function evaluations           =      0.000

EXIT: Optimal Solution Found.
```

```
Out[11]: :Optimal
```

In [12]:
```
println("Objective value: ", getobjectivevalue(m_3))
println("Machine 1 = ", getvalue(P1M1_3))
println("Machine 1 = ", getvalue(P2M1_3))
println("Machine 1 = ", getvalue(P3M1_3))
println("Machine 2 = ", getvalue(P1M2_3))
println("Machine 2 = ", getvalue(P2M2_3))
println("Machine 2 = ", getvalue(P3M2_3))
println("Machine 3 = ", getvalue(P1M3_3))
println("Machine 3 = ", getvalue(P2M3_3))
println("Machine 3 = ", getvalue(P3M3_3))
println("Machine 4 = ", getvalue(P1M4_3))
println("Machine 4 = ", getvalue(P2M4_3))
println("Machine 4 = ", getvalue(P3M4_3))
```

```
Objective value: 67666.66659992337
Machine 1 = 3183.6135415554
Machine 1 = 0.0
Machine 1 = 0.0
Machine 2 = 816.3864584622615
Machine 2 = 0.0
Machine 2 = 1333.3333000190232
Machine 3 = 0.0
Machine 3 = 5000.000000013085
Machine 3 = 1666.6666999972176
Machine 4 = 0.0
Machine 4 = 0.0
Machine 4 = 0.0
```

4

Leary Chemical manufactures three chemicals: A, B and C. These chemical are produced via two production processes: 1 and 2. Running process 1 for an hour costs $4 and yields 3 units of A, 1 of B and 1 of C. Running process 2 for an hour costs $1 and produces 1 unit of A and 1 of B. To meet customer demands, at least 10 units of A, 5 of B and 3 of C must be produced daily.

In order to minimize the cost of meeting Leary Chemical's daily demands:

a) Formulate this managerial problem as an LP problem.

In [13]:
```
m_4 = Model(solver=IpoptSolver())
@variable(m_4, P1_4 >= 0)
@variable(m_4, P2_4 >= 0)
@constraint(m_4, 3*P1_4 + P2_4 >= 10)
@constraint(m_4, P1_4 + P2_4 >= 5)
@constraint(m_4, P1_4 >= 3)
@objective(m_4, Min, 4P1_4 + P2_4)

print(m_4)
```

```
Min 4 P1_4 + P2_4
Subject to
 3 P1_4 + P2_4 ≥ 10
 P1_4 + P2_4 ≥ 5
 P1_4 ≥ 3
 P1_4 ≥ 0
 P2_4 ≥ 0
```

In [14]:
```
solve(m_4)
```

```
This is Ipopt version 3.12.1, running with linear solver mumps.
NOTE: Other linear solvers might be more efficient (see Ipopt documentation).

Number of nonzeros in equality constraint Jacobian...:        0
Number of nonzeros in inequality constraint Jacobian.:        5
Number of nonzeros in Lagrangian Hessian.............:        0

Total number of variables............................:        2
                     variables with only lower bounds:        2
                variables with lower and upper bounds:        0
                     variables with only upper bounds:        0
Total number of equality constraints.................:        0
Total number of inequality constraints...............:        3
        inequality constraints with only lower bounds:        3
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        0

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  4.9999950e-02 9.96e+00 8.00e-01  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  2.4544222e-01 9.81e+00 3.09e+00  -1.0 3.34e+00    -  5.34e-03 2.26e-02h  1
   2  1.0447886e+00 9.18e+00 3.67e+00  -1.0 2.86e+00    -  1.52e-02 5.91e-02h  1
   3  1.4203460e+01 0.00e+00 1.53e+01  -1.0 2.83e+00    -  7.52e-02 1.00e+00h  1
   4  1.4168930e+01 0.00e+00 1.00e-06  -1.0 4.55e-02    -  1.00e+00 1.00e+00f  1
   5  1.4007161e+01 0.00e+00 1.08e-02  -2.5 1.39e-01    -  1.00e+00 9.18e-01f  1
   6  1.4000242e+01 0.00e+00 1.50e-09  -3.8 4.62e-03    -  1.00e+00 1.00e+00f  1
   7  1.4000003e+01 0.00e+00 1.84e-11  -5.7 2.09e-04    -  1.00e+00 1.00e+00f  1
   8  1.4000000e+01 0.00e+00 2.51e-14  -8.6 3.02e-06    -  1.00e+00 1.00e+00f  1

Number of Iterations....: 8

                                   (scaled)                 (unscaled)
Objective...............:   1.3999999865004634e+01    1.3999999865004634e+01
Dual infeasibility......:   2.5120766448416835e-14    2.5120766448416835e-14
Constraint violation....:   0.0000000000000000e+00    0.0000000000000000e+00
Complementarity.........:   2.5116011409572163e-09    2.5116011409572163e-09
Overall NLP error.......:   2.5116011409572163e-09    2.5116011409572163e-09


Number of objective function evaluations             = 9
Number of objective gradient evaluations             = 9
Number of equality constraint evaluations            = 0
Number of inequality constraint evaluations          = 9
Number of equality constraint Jacobian evaluations   = 0
Number of inequality constraint Jacobian evaluations = 9
Number of Lagrangian Hessian evaluations             = 8
Total CPU secs in IPOPT (w/o function evaluations)   =      0.006
Total CPU secs in NLP function evaluations           =      0.000

EXIT: Optimal Solution Found.
```

Out[14]:
```
:Optimal
```

In [15]:
```
println("Objective value: ", round(getobjectivevalue(m_4)))
println("Process 1 = ", round(getvalue(P1_4)))
println("Process 2 = ", round(getvalue(P2_4)))
println("Chemical 1 = ", round(getvalue(3*P1_4 + 2*P2_4)))
println("Chemical 2 = ", round(getvalue(P1_4 + P2_4)))
println("Chemical 3 = ", round(getvalue(P1_4 + P2_4)))
```

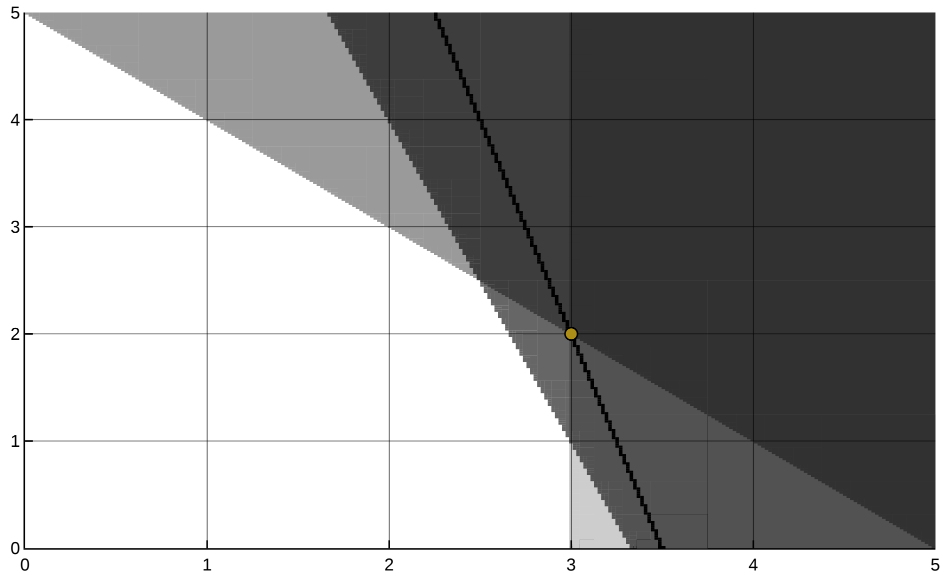
```
Objective value: 14.0
Process 1 = 3.0
Process 2 = 2.0
Chemical 1 = 13.0
Chemical 2 = 5.0
Chemical 3 = 5.0
```

b) Graphically determine a daily production plan.

In [16]:
```
C_1_4(P1,P2) = 3P1+P2
C_2_4(P1,P2) = P1+P2
C_3_4(P1,P2) = P1
O_4(P1,P2) = 4P1+P2

plot(C_1_4 ≧ 10, alpha=0.6, xlims=(0,5), ylims=(0,5))
plot!(C_2_4 ≧ 5, alpha=0.4, xlims=(0,5), ylims=(0,5))
plot!(C_3_4 ≧ 3, alpha=0.2, xlims=(0,5), ylims=(0,5))
plot!(O_4 == getobjectivevalue(m_4), alpha=1, xlims=(0,5), ylims=(0,5))
scatter!([getvalue(P1_4)],[getvalue(P2_4)], xlims=(0,5), ylims=(0,5))
```

Out[16]:



c) Confirm the solution using the Solver Excel® add-in.

You wish.

5

Mr. Silva is dealing with dollar currency and yuan currency. At 12 midnight, he can buy dollars by paying 0.25 yuans per dollar and yuans by paying 3 dollars per yuan. Let $x_1 =$ number of yuans bought (by paying dollars) and $x_2 =$ number of dollars bought (by paying yauns). Assume that both type of transactions take place simultaneously, and the only constraint is that at 12:01 a.m. Mr. Silva must have a nonnegative number of dollars and yuans. In order to help Mr. Silva maximizing the number of yauns he has after all transactions are completed:

a) Formulate the corresponding LP problem.

In [17]:
```
m_5 = Model(solver=IpoptSolver())
@variable(m_5, x1_5 >= 0)
@variable(m_5, x2_5 >= 0)

@constraint(m_5, )

@objective(m_5, Max, Y1_5)
```

```
In @constraint(m_5): Not enough arguments

Stacktrace:
 [1] include_string(::String, ::String) at ./loading.jl:522
```

In [18]:
```
m_5 = Model(solver=IpoptSolver())
@variable(m_5, x1_5 >= 0)
@variable(m_5, x2_5 >= 0)
@variable(m_5, Y0_5 >= 0)
@variable(m_5, D0_5 >= 0)
@variable(m_5, Y1_5 >= 0)
@variable(m_5, D1_5 >= 0)

@constraint(m_5, Y1_5==Y0_5 + x1_5 - .25x2_5)
@constraint(m_5, D1_5==D0_5 + -3x1_5 +x2_5)
@constraint(m_5, x1_5==Y1_5 - 3D0_5)
@constraint(m_5, x2_5==-0.25Y0_5 + D1_5)
@objective(m_5, Max, Y1_5)

print(m_5)
```

```
Max Y1_5
Subject to
 Y1_5 - Y0_5 - x1_5 + 0.25 x2_5 = 0
 D1_5 - D0_5 + 3 x1_5 - x2_5 = 0
 x1_5 - Y1_5 + 3 D0_5 = 0
 x2_5 + 0.25 Y0_5 - D1_5 = 0
 x1_5 ≥ 0
 x2_5 ≥ 0
 Y0_5 ≥ 0
 D0_5 ≥ 0
 Y1_5 ≥ 0
 D1_5 ≥ 0
```

In [19]:
```
solve(m_5)
```

```
This is Ipopt version 3.12.1, running with linear solver mumps.
NOTE: Other linear solvers might be more efficient (see Ipopt documentation).

Number of nonzeros in equality constraint Jacobian...:       14
Number of nonzeros in inequality constraint Jacobian.:        0
Number of nonzeros in Lagrangian Hessian.............:        0

Total number of variables............................:        6
                     variables with only lower bounds:        6
                variables with lower and upper bounds:        0
                     variables with only upper bounds:        0
Total number of equality constraints.................:        4
Total number of inequality constraints...............:        0
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:        0
        inequality constraints with only upper bounds:        0

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0 -9.9999900e-03 3.00e-02 1.66e+00  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  1.1665810e-01 1.39e-17 7.24e+00  -1.0 1.27e-01    -  2.66e-01 1.00e+00f  1
```

In [20]:
```
println("Objective value: ", getobjectivevalue(m_5))
println("x1, number of yuans bought (by paying dollars) = ", getvalue(x1_5))
println("x2, number of dollars bought (by paying yauns) = ", getvalue(x2_5))
```

```
Objective value: NaN
x1, number of yuans bought (by paying dollars) = NaN
x2, number of dollars bought (by paying yauns) = NaN
```

```
WARNING: Variable value not defined for x1_5. Check that the model was properly solved.
WARNING: Variable value not defined for x2_5. Check that the model was properly solved.
```

In [ ]:

In [ ]: