Example 3.6. Reconsider the family farm problem of Example 3.4. The family has 625 acres available for planting. There are 5 plots of 120 acres each and another plot of 25 acres. The family wants to plant each plot with only one crop: corn, wheat, or oats. As before, 1,000 acre–ft of water will be available for irrigation, and the farmers will be able to devote 300 hours of labor per week. Additional data are presented in Table 3.2. Find the crop that should be planted in each plot for maximum profit.

Table 3.2.

Requirements	Corn	Wheat	Oats
Irrigation (acre-ft)	3.0	1.0	1.5
Labor (person-hrs/week)	0.8	0.2	0.3
Yield ($)	400	200	250

x_1 = number of 120 acre plots of corn planted
x_2 = number of 120–acre plots of wheat planted
x_3 = number of 120–acre plots of oats planted
x_4 = number of 25–acre plots of corn planted
x_5 = number of 25–acre plots of wheat planted
x_6 = number of 25–acre plots of oats planted
w = irrigation required (acre–ft)
l = labor required (person–hrs/wk)
t = total acreage planted
y = total yield ($)

In [1]:
```
using JuMP,HiGHS
```

In [2]:
```
model=Model(HiGHS.Optimizer)
```

Out[2]:
```
A JuMP Model
├ solver: HiGHS
├ objective_sense: FEASIBILITY_SENSE
├ num_variables: 0
├ num_constraints: 0
└ Names registered in the model: none
```

In [3]:
```
@variable(model,x[1:6].>=0,Int)
```

Out[3]:
```
6-element Vector{VariableRef}:
 x[1]
 x[2]
 x[3]
 x[4]
 x[5]
 x[6]
```

In [4]:
```
print(model)
```

$$\begin{aligned} & \text{feasibility} \\ \text{Subject to} \quad & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_3 \geq 0 \\ & x_4 \geq 0 \\ & x_5 \geq 0 \\ & x_6 \geq 0 \\ & x_1 \in \mathbb{Z} \\ & x_2 \in \mathbb{Z} \\ & x_3 \in \mathbb{Z} \\ & x_4 \in \mathbb{Z} \\ & x_5 \in \mathbb{Z} \\ & x_6 \in \mathbb{Z} \end{aligned}$$

$w(x) =$ The total amount of water used
$l(x) =$ The total labor needed
$t(x) =$ The total acres planted
$y(x) =$ The yield (objective function)

In [5]:
```
# Note the function definitions that extend over two lines extend
# to the next line by placing the + at the end of the first line.
w(x)=120*(3.0*x[1]+1.0*x[2]+1.5*x[3])+
      25*(3.0*x[4]+1.0*x[5]+1.5*x[6])
l(x)=120*(0.8*x[1]+0.2*x[2]+0.3*x[3])+
      25*(0.8*x[4]+0.2*x[5]+0.3*x[6])
t(x)=120*(x[1]+x[2]+x[3])+25*(x[4]+x[5]+x[6])
y(x)=120*(400*x[1]+200*x[2]+250*x[3])+
     25*(400*x[4]+200*x[5]+250*x[6])
```

Out[5]:
```
y (generic function with 1 method)
```

In [6]:
```
@objective(model,Max,y(x))
```

Out[6]: $48000x_1 + 24000x_2 + 30000x_3 + 10000x_4 + 5000x_5 + 6250x_6$

In [7]:
```
c1=@constraint(model,w(x)<=1000)
```

Out[7]:

$$360x_1 + 120x_2 + 180x_3 + 75x_4 + 25x_5 + 37.5x_6 \leq 1000$$

In [8]:
```
c2=@constraint(model,l(x)<=300)
```

Out[8]:

$$96x_1 + 24x_2 + 36x_3 + 20x_4 + 5x_5 + 7.5x_6 \leq 300$$

In [9]:
```
c3=@constraint(model,x[1]+x[2]+x[3]<=5)
```

Out[9]:

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

In [10]:
```
c4=@constraint(model,x[4]+x[5]+x[6]<=1)
```

Out[10]:

$$x_4 + x_5 + x_6 \leq 1$$

In [11]:
```
print(model)
```

$$\begin{aligned} \max \quad & 48000x_1 + 24000x_2 + 30000x_3 + 10000x_4 + 5000x_5 + 6250x_6 \\ \text{Subject to} \quad & 360x_1 + 120x_2 + 180x_3 + 75x_4 + 25x_5 + 37.5x_6 \leq 1000 \\ & 96x_1 + 24x_2 + 36x_3 + 20x_4 + 5x_5 + 7.5x_6 \leq 300 \\ & x_1 + x_2 + x_3 \leq 5 \\ & x_4 + x_5 + x_6 \leq 1 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_3 \geq 0 \\ & x_4 \geq 0 \\ & x_5 \geq 0 \\ & x_6 \geq 0 \\ & x_1 \in \mathbb{Z} \\ & x_2 \in \mathbb{Z} \\ & x_3 \in \mathbb{Z} \\ & x_4 \in \mathbb{Z} \\ & x_5 \in \mathbb{Z} \\ & x_6 \in \mathbb{Z} \end{aligned}$$

In [12]:
```
optimize!(model)
```

```
Running HiGHS 1.13.1 (git hash: 1d267d97c): Copyright (c) 2026 under Apache
2.0 license terms
Using BLAS: blastrampoline
MIP has 4 rows; 6 cols; 18 nonzeros; 6 integer variables (0 binary)
Coefficient ranges:
  Matrix  [1e+00, 4e+02]
  Cost    [5e+03, 5e+04]
  Bound   [0e+00, 0e+00]
  RHS     [1e+00, 1e+03]
Presolving model
4 rows, 6 cols, 18 nonzeros  0s
4 rows, 6 cols, 18 nonzeros  0s
Presolve reductions: rows 4(-0); columns 6(-0); nonzeros 18(-0) - Not reduce
d
Objective function is integral with scale 0.004

Solving MIP model with:
   4 rows
   6 cols (3 binary, 3 integer, 0 implied int., 0 continuous, 0 domain fixe
d)
   18 nonzeros

Src: B => Branching; C => Central rounding; F => Feasibility pump; H => Heur
istic;
     I => Shifting; J => Feasibility jump; L => Sub-MIP; P => Empty MIP; R =
> Randomized rounding;
     S => Solve LP; T => Evaluate node; U => Unbounded; X => User solution;
Y => HiGHS solution;
     Z => ZI Round; l => Trivial lower; p => Trivial point; u => Trivial upp
er; z => Trivial zero

        Nodes      |    B&B Tree     |            Objective Bounds
|  Dynamic Constraints |       Work
Src  Proc. InQueue |  Leaves   Expl. | BestBound       BestSol
Gap |   Cuts   InLp Confl. | LpIters     Time

 z       0       0         0   0.00%   inf             -0                La
rge        0      0      0         0     0.0s
 J       0       0         0   0.00%   inf             150000            La
rge        0      0      0         0     0.0s
 R       0       0         0   0.00%   162500          155000            4.
84%        0      0      0         4     0.0s
 L       0       0         0   0.00%   162250          162250            0.
00%       10      3      3         8     0.0s
         1       0         1 100.00%   162250          162250            0.
00%       10      3      3         8     0.0s

Solving report
  Status            Optimal
  Primal bound      162250
  Dual bound        162250
  Gap               0% (tolerance: 0.01%)
  P-D integral      0.00268704875823
  Solution status   feasible
                    162250 (objective)
                    0 (bound viol.)
```

```
                       3.99680288865e-15 (int. viol.)
                       0 (row viol.)
 Timing                0.04
 Max sub-MIP depth     1
 Nodes                 1
 Repair LPs            0
 LP iterations         8
                       0 (strong br.)
                       4 (separation)
                       0 (heuristics)
```

In [13]:

```
objective_value(model)
```

Out[13]:

```
162250.00000000003
```

In [14]:

```
value(x)
```

Out[14]:

```
6-element Vector{Float64}:
  0.9999999999999998
  2.0
  2.0000000000000004
  3.9968028886505635e-15
 -0.0
  0.999999999999996
```

Solution was $[1, 2, 2, 0, 0, 1]$ with optimal yield 162250.

In [ ]: