Lab 3 Solutions

2025-06-04

Due Date

Wednesday, 10/16/24, 9:00pm

Setup

The following code should go at the top of most Julia scripts; it will load the local package environment and install any needed packages. You will see this often and shouldn't need to touch it.

```julia
import Pkg
Pkg.activate(".")
Pkg.instantiate()
```

```julia
using JuMP # optimization modeling syntax
using HiGHS # optimization solver
```

Overview

Exercise (3 points)

Your task is to decide how much lumber to produce to maximize profit from wood sales. You can purchase wood from a managed forest, which consists of spruce (320,000 bf) and fir (720,000 bf). Spruce costs $0.12 per bf to purchase and fir costs $0.08 per bf.

At the lumber mill, wood can be turned into plywood of various grades (see Table 1 for how much wood of each type is required for and the revenue from each grade). Any excess wood is sent to be recycled into particle board, which yields no revenue for the mill.

Table 1: Wood inputs and revenue by plywood grade. S refers to spruce inputs, F fir inputs.

Plywood Grade	Inputs (bf/bf plywood)	Revenue ($/1000 bf)
1	0.5 (S) + 1.5 (F)	400
2	1.0 (S) + 2.0 (F)	520
3	1.5 (S) + 2.0 (F)	700

First, we need to identify our decision variables. While there are several options, we will use G_i, the amount of each grade the mill produces (in 1000 bf).

Using these decision variables, formulate a linear program to maximize the profit of the mill subject to the supply constraints on spruce and fir.

Solution:

With these decision variables, we need to calculate the amount of profit yielded by a bf of each grade of plywood.

1. G1 sells for $400 per 1000 bf and consumes 500 bf of spruce and 1500 bf of fir, costing $60 from spruce and $120 from fir, yielding a profit of $220/(1000 bf).
2. G2 sells for $520 per 1000 bf and consumes 1000 bf of spruce and 2000 bf of fir, costing $120 from spruce and $160 from fir, yielding a profit of $240/(1000 bf).
3. G3 sells for $700 per 1000 bf and consumes 1500 bf of spruce and 2000 bf of fir, costing $180 from spruce and $160 from fir, yielding a profit of $360/(1000 bf).

Then the objective is

$$\max_{G_1,G_2,G_3} 220G_1 + 240G_2 + 360G_3.$$

We can only purchase 320,000 bf of spruce and 720,000 bf of fir. These become the following constraints (in terms of 1,000 bf):

$$\begin{aligned} 0.5G_1 + G_2 + 1.5G_3 &\leq 320 \\ 1.5G_1 + 2G_2 + 2G_3 &\leq 720 \end{aligned}$$

Finally, we have non-negativity (or boundary) constraints on G_i:

$$G_1, G_2, G_3 \geq 0.$$

This makes the final optimization problem:

$$
\begin{aligned}
\max_{G_1,G_2,G_3} \quad & 220G_1 + 240G_2 + 360G_3 \\
\text{subject to} \quad & \\
& 0.5G_1 + G_2 + 1.5G_3 \leq 320 \\
& 1.5G_1 + 2G_2 + 2G_3 \leq 720 \\
& G_1, G_2, G_3 \geq 0
\end{aligned}
\tag{1}
$$

Next, we want to convert the optimization problem (1) into `JuMP` syntax, which we do below.

```
forest_model = Model(HiGHS.Optimizer) # initialize model object
@variable(forest_model, G[1:3] >= 0) # define variables and add
 ↪  non-negativity constraints
@objective(forest_model, Max, 220G[1] + 240G[2] + 360G[3])
@constraint(forest_model, spruce, 0.5G[1] + G[2] + 1.5G[3] <= 320)
@constraint(forest_model, fir, 1.5G[1] + 2G[2] + 2G[3] <= 720)
print(forest_model) # this outputs a nicely formatted summary of the model so
 ↪  you can check your specification
```

```
Max 220 G[1] + 240 G[2] + 360 G[3]
Subject to
 spruce : 0.5 G[1] + G[2] + 1.5 G[3]   320
 fir : 1.5 G[1] + 2 G[2] + 2 G[3]   720
 G[1]   0
 G[2]   0
 G[3]   0
```

Next, to optimize, use the `optimize!()` function:

```
optimize!(forest_model)
```

```
Running HiGHS 1.7.2 (git hash: 5ce7a2753): Copyright (c) 2024 HiGHS under MIT licence terms
Coefficient ranges:
  Matrix [5e-01, 2e+00]
  Cost   [2e+02, 4e+02]
  Bound  [0e+00, 0e+00]
  RHS    [3e+02, 7e+02]
Presolving model
2 rows, 3 cols, 6 nonzeros  0s
2 rows, 3 cols, 6 nonzeros  0s
```

```
Presolve : Reductions: rows 2(-0); columns 3(-0); elements 6(-0) - Not reduced
Problem not reduced by presolve: solving the LP
Using EKK dual simplex solver - serial
  Iteration        Objective     Infeasibilities num(sum)
          0    -8.1999929744e+02 Ph1: 2(8.5); Du: 3(819.999) 0s
          2     1.1200000000e+05 Pr: 0(0) 0s
Model   status      : Optimal
Simplex   iterations: 2
Objective value     :  1.1200000000e+05
HiGHS run time      :          0.00
```

```julia
@show value.(G);
@show objective_value(forest_model);
```

```
value.(G) = [352.0, 0.0, 96.0]
objective_value(forest_model) = 112000.0
```

So the optimal result is to produce 96,000 bf of grade 3 plywood and 352,000 bf of grade 1 plywood, which yields a profit of $112,000.

```julia
@show shadow_price(spruce);
@show shadow_price(fir);
```

```
shadow_price(spruce) = 80.0
shadow_price(fir) = 120.0
```

Both the spruce and fir constraints are binding, as evidenced by their non-zero shadow prices.

Why does it make sense that we aren't producing any grade 2 plywood even though it produces more profit than grade 1? The production of grade 3, which is the highest-profit grade, consumes 144,000 bf of spruce and 192,000 bf of fir. If we produced as much grade 2 plywood as possible, which would be 176,000 bf, we would make a profit of $76,800. Since we can produce far more grade 1 plywood, the extra $20/(1,000 bf) profit from producing grade 2 is not enough to compensate for the loss of production volume.

References

Put any consulted sources here, including classmates you worked with/who helped you.