540HW3

Jordan Hilton

January 22, 2019

Problem 4.1

We're going to start by formulating and building a model; here's the model's source data for reference.

Characteristic	Chairs	Desks	Frames	Tables	Available
Profit	20	14	3	16	
Fabrication	6	2	1	4	1440
Assembly	8	6	1	8	1440
Machining	6	4	1	25	2000
Painting	7	10	2	12	1000
Wood	40	25	5	16	9600

a (Formulation)

Let's start by defining our choice variables:

- Chairs = # of Chairs to Make
- Desks = # of Desks to Make
- Frames = # of Frames to Make
- Tables = # of Tables to Make

Now let's formulate the LP:

$$\begin{aligned}
\text{Maximize } & 20 * Chairs + 14 * Desks + 3 * Frames + 16 * Tables \\
\text{subject to } & 6 * Chairs + 2 * Desks + Frames + 4 * Tables \leq 1440 \\
& 8 * Chairs + 6 * Desks + Frames + 8 * Tables \leq 1440 \\
& 6 * Chairs + 4 * Desks + Frames + 25 * Tables \leq 2000 \\
& 7 * Chairs + 10 * Desks + 2 * Frames + 12 * Tables \leq 1000 \\
& 40 * Chairs + 25 * Desks + 5 * Frames + 16 * Tables \leq 9600 \\
& Chairs, Desks, Frames, Tables \geq 0
\end{aligned}$$

b (Implementation)

Let's implement our model in OMPR:

```
model <- MIPModel() %>%
  add_variable(Chairs, type = "continuous", lb = 0) %>%
  add_variable(Desks, type = "continuous",lb = 0) %>%
  add_variable(Frames, type = "continuous", lb = 0) %>%
  add_variable(Tables, type = "continuous", lb = 0) %>%

  set_objective(20*Chairs + 14*Desks + 3*Frames + 16*Tables, "max") %>%

  add_constraint(6*Chairs+2*Desks+Frames+4*Tables <= 1440) %>% #fabrication
  add_constraint(8*Chairs+6*Desks+Frames+8*Tables <= 1440) %>% #assembly
```

```
  add_constraint(6*Chairs+4*Desks+Frames+25*Tables <= 2000) %>% #machining
  add_constraint(7*Chairs+10*Desks+2*Frames+12*Tables <= 1000) %>% #painting
  add_constraint(40*Chairs+25*Desks+5*Frames+16*Tables <= 9600) #wood
```

c and d (Solution and interpretation)

Now let's examine the values for our decision variables, the optimal objective value, and the row and column duals.

```
solution<-solve_model(model, with_ROI(solver = "glpk"))
solution$status
```

```
## [1] "optimal"
```

```
solution$solution
```

```
##   Chairs    Desks   Frames   Tables
## 142.8571   0.0000   0.0000   0.0000
```

```
solution$objective_value
```

```
## [1] 2857.143
```

So our optimal solution is to make only chairs, of which we have enough materials to make 142.8571, and the total profit we will make is $2,857.14.

e Reduced Costs and Shadow Prices

Let's take a look at our reduced costs and shadow prices:

```
rowduals<-data.frame(ShadowPrices=solution$solution_row_duals(), row.names=(c("Fabrication", "Assembly"
pander(rowduals, caption="Shadow Prices")
```

Table 2: Shadow Prices

	ShadowPrices
Fabrication	0
Assembly	0
Machining	0
Painting	2.857
Wood	0

```
columnduals<-data.frame(ReducedCosts=solution$solution_column_duals())
pander(columnduals, caption="Reduced Costs")
```

Table 3: Reduced Costs

	ReducedCosts
Chairs	0
Desks	-14.57
Frames	-2.714
Tables	-18.29

We see that our only constrained resource is paint, so we would be unaffected by a small reduction in other resources and we would achieve additional profit with a small increase in our supply of paint. We also see that we'd take a small opportunity cost loss in profit by making tables, frames, or desks, with tables being the worst option, losing 18 dollars in profit for every table we chose to make.

f Changing the Objective Function

Since we know that our current plan is to only make chairs, let's increase the profitability of frames by slightly more than the current reduced costs of frames to get our model to decide to make some frames. I'll change the profit coefficient for frames from 3 to 6, so that the formal objective function would read:

$$\text{Maximize } 20 * Chairs + 14 * Desks + 6 * Frames + 16 * Tables$$

and our R model looks like:

```
modelf <- MIPModel() %>%
  add_variable(Chairs, type = "continuous", lb = 0) %>%
  add_variable(Desks, type = "continuous",lb = 0) %>%
  add_variable(Frames, type = "continuous", lb = 0) %>%
  add_variable(Tables, type = "continuous", lb = 0) %>%

  set_objective(20*Chairs + 14*Desks + 6*Frames + 16*Tables, "max") %>%

  add_constraint(6*Chairs+2*Desks+Frames+4*Tables <= 1440) %>% #fabrication
  add_constraint(8*Chairs+6*Desks+Frames+8*Tables <= 1440) %>% #assembly
  add_constraint(6*Chairs+4*Desks+Frames+25*Tables <= 2000) %>% #machining
  add_constraint(7*Chairs+10*Desks+2*Frames+12*Tables <= 1000) %>% #painting
  add_constraint(40*Chairs+25*Desks+5*Frames+16*Tables <= 9600) #wood
```

Let's look at the new solution:

```
solutionf<-solve_model(modelf, with_ROI(solver = "glpk"))
solutionf$status
```

```
## [1] "optimal"
```

```
solutionf$solution
```

```
## Chairs  Desks Frames Tables
##      0      0    500      0
```

```
solutionf$objective_value
```

```
## [1] 3000
```

You can see that now we're supposed to make some frames, and that our overall profit has increased from $2,857 to $3,000.

g Changing Resource Usage

Let's try removing our paint usage for tables, since that is our constrained resource- we can sell unpainted tables to use up our spare resources of wood, etc. The paint constraint would change to:

$$7 * Chairs + 10 * Desks + 2 * Frames \leq 1000$$

and our R model looks like this (remember that we're changing the profitability of frames back to 3):

```
modelg <- MIPModel() %>%
  add_variable(Chairs, type = "continuous", lb = 0) %>%
  add_variable(Desks, type = "continuous",lb = 0) %>%
  add_variable(Frames, type = "continuous", lb = 0) %>%
```

```
  add_variable(Tables, type = "continuous", lb = 0) %>%

  set_objective(20*Chairs + 14*Desks + 3*Frames + 16*Tables, "max") %>%

  add_constraint(6*Chairs+2*Desks+Frames+4*Tables <= 1440) %>% #fabrication
  add_constraint(8*Chairs+6*Desks+Frames+8*Tables <= 1440) %>% #assembly
  add_constraint(6*Chairs+4*Desks+Frames+25*Tables <= 2000) %>% #machining
  add_constraint(7*Chairs+10*Desks+2*Frames <= 1000) %>% #painting
  add_constraint(40*Chairs+25*Desks+5*Frames+16*Tables <= 9600) #wood
```

Let's look at the new solution:

```
solutiong<-solve_model(modelg, with_ROI(solver = "glpk"))
solutiong$status
```

```
## [1] "optimal"
```

```
solutiong$solution
```

```
##    Chairs     Desks    Frames    Tables
## 142.85714   0.00000   0.00000  37.14286
```

```
solutiong$objective_value
```

```
## [1] 3451.429
```

Great, now we get most of our profit from our painted chairs but some additional profit from our unpainted tables, which are using up some of our spare resources.

h Changing Available Resources

This time let's triple our supply of paint, since that's the constrained resource. The relevant line of the formal model would change as follows:

$$7 * Chairs + 10 * Desks + 2 * Frames + 12 * Tables \leq 3000$$

and our R model looks like this:

```
modelh <- MIPModel() %>%
  add_variable(Chairs, type = "continuous", lb = 0) %>%
  add_variable(Desks, type = "continuous",lb = 0) %>%
  add_variable(Frames, type = "continuous", lb = 0) %>%
  add_variable(Tables, type = "continuous", lb = 0) %>%

  set_objective(20*Chairs + 14*Desks + 3*Frames + 16*Tables, "max") %>%

  add_constraint(6*Chairs+2*Desks+Frames+4*Tables <= 1440) %>% #fabrication
  add_constraint(8*Chairs+6*Desks+Frames+8*Tables <= 1440) %>% #assembly
  add_constraint(6*Chairs+4*Desks+Frames+25*Tables <= 2000) %>% #machining
  add_constraint(7*Chairs+10*Desks+2*Frames+12*Tables <= 3000) %>% #painting
  add_constraint(40*Chairs+25*Desks+5*Frames+16*Tables <= 9600) #wood
```

Let's look at the new solution:

```
solutionh<-solve_model(modelh, with_ROI(solver = "glpk"))
solutionh$status
```

```
## [1] "optimal"
```

```r
solutionh$solution
```

```
## Chairs  Desks Frames Tables
##      0      0   1440      0
```

```r
solutionh$objective_value
```

```
## [1] 4320
```

This change causes us to create frames instead of chairs, and significantly increases our overall profit. Note that when we run the rowduals another resource should be constrained now:

```r
rowdualsh<-data.frame(ShadowPrices=solutionh$solution_row_duals(), row.names=(c("Fabrication", "Assembly
pander(rowdualsh, caption="New Shadow Prices")
```

Table 4: New Shadow Prices

	ShadowPrices
Fabrication	1
Assembly	2
Machining	0
Painting	0
Wood	0

As expected, now we have shadow prices for Fabrication and Assembly but not Painting.

i Summary Table

Let's create a summary table of all of our solution:

```r
objectivevalues<-c(solution$objective_value, solutionf$objective_value, solutiong$objective_value, solu

solutiontable<- solution$solution %>%
  rbind(solutionf$solution) %>%
  rbind(solutiong$solution) %>%
  rbind(solutionh$solution) %>%
  cbind(objectivevalues)

rownames(solutiontable)<-c("Base Solution", "Part F", "Part G", "Part H")

pander(solutiontable)
```

	Chairs	Desks	Frames	Tables	objectivevalues
Base Solution	142.9	0	0	0	2857
Part F	0	0	500	0	3000
Part G	142.9	0	0	37.14	3451
Part H	0	0	1440	0	4320

My favorite is solution G, where we got to use our unconstrained resources. Of course the profit is higher in some models just because I gave us magically more resources or lower consumption to work with. Let's play the same game with row duals just for fun:

```r
objectivevalues<-c(solution$objective_value, solutionf$objective_value, solutiong$objective_value, solu
```

```
rowdualsf<-data.frame(ShadowPrices=solutionf$solution_row_duals(), row.names=(c("Fabrication", "Assembly

rowdualsg<-data.frame(ShadowPrices=solutiong$solution_row_duals(), row.names=(c("Fabrication", "Assembly

rowdualstable<- rowduals %>%
  cbind(rowdualsf) %>%
  cbind(rowdualsg) %>%
  cbind(rowdualsh)

colnames(rowdualstable)<-c("Base Solution", "Part F", "Part G", "Part H")

pander(rowdualstable, caption="Comparing Row Duals")
```

Table 6: Comparing Row Duals

	Base Solution	Part F	Part G	Part H
Fabrication	0	0	0	1
Assembly	0	0	2	2
Machining	0	0	0	0
Painting	2.857	3	0.5714	0
Wood	0	0	0	0

Here we can see that in my favorite solution G, our new constraining resource for our unpainted tables is assembly time- so we could play some more by deciding to sell unassembled desks, etc.

Problem 4.2

Just very quickly, I'm going to run our generalized transport model from HW2, look at the row and column duals, and answer the question:

```
NW<- 4
ND<- 4
PathNames<-list(c("XCP", "XCS", "XCM", "XCA", "XBS", "XBM", "XBA", "XEP", "XES", "XEM", "XEA", "XDP", "X

WarehouseNames <- list(c("Chicago", "Beaverton", "Eugene", "Dallas"))


DistributorNames<- list(c("PDX", "SEA", "MSP", "ATL"))


PathCosts <- matrix(c(20, 21, 8, 12, 6, 7, 18, 24, 8, 10, 22, 28, 16, 26, 15, 5),
              ncol=ND,byrow=TRUE,dimnames=c(WarehouseNames,DistributorNames))

Supply<- matrix(c(500, 500, 500, 600),
              ncol=NW,dimnames=c("Supply",WarehouseNames))

Demand <- matrix(c(700, 500, 500, 600),
              ncol=ND,dimnames=c("Demand",DistributorNames))

transmodel2 <- MIPModel() %>%
  add_variable (x[i, j], i=1:NW,j=1:ND, type="continuous", lb=0) %>%
  set_objective (sum_expr(sum_expr(PathCosts[i, j] * x[i, j] , i=1:NW), j=1:ND ), "min") %>% #minimize
  add_constraint (sum_expr(x[i, j], j=1:ND) == Supply[i],  i=1:NW) %>%  # NW constraints on supply
```

```r
  add_constraint (sum_expr(x[i, j], i=1:NW) <= Demand[j],  j=1:ND) %>%  # ND constraints on demand
  solve_model(with_ROI(solver = "glpk"))
```

We have the solution to the model, let's make tables of the row and column duals:

```r
transcolduals<-matrix(transmodel2$solution_column_duals(), ncol=ND, dimnames=(c(WarehouseNames,Distribu
pander(transcolduals, caption="Increased transportation prices")
```

Table 7: Increased transportation prices

	PDX	SEA	MSP	ATL
Chicago	0	0	0	3
Beaverton	0	0	24	29
Eugene	0	1	26	31
Dallas	0	9	11	0

These are the 16 column duals corresponding to our 16 decision variables. We should interpret the 26 for the Eugene-MSP path to mean that it would cost us an addition 26 dollars if we were forced to transport one unit from Eugene to MSP.

```r
transrowtable<-data.frame(RowDuals=transmodel2$solution_row_duals(), row.names=c("Chicago", "Beaverton"
pander(transrowtable)
```

	RowDuals
Chicago	21
Beaverton	7
Eugene	9
Dallas	17
PDX	-1
SEA	0
MSP	-13
ATL	-12

Here are the row duals, which should each correspond to a constraint, and to which I've attached the names for the appropriate constraints. Remember that we're trying to minimize cost here, so smaller row duals are better. I think the story is that we should allocate an additional unit of warehouse supply to Beaverton, since we'd be able to ship it somewhere for 7 dollars; and if we had an additional unit of distributor capacity, we should allocate it to MSP, since we'll be able to ship there for 13 dollars cheaper than wherever we're shipping our last unit currently.