ETM540 Homework#6 - Ship Loading

Mala Daryanani

November 6, 2018

Exercise:

You are responsible for loading a ship with 4 holds with 3 different cargos. The cargos have different profits, volumes, and weights.

- The ship must maintain balance for seaworthiness with the following
- Left and right hold weights must be with 20% of each other
- Front and back holds must each be between 20% and 30% of the total

Your goal is to find the plan for loading the ship that generates the best profit

Load the required libraries:

```
library (pander, quietly = TRUE)
library (magrittr, quietly = TRUE) #Used for pipes/dplyr
library(dplyr, quietly = TRUE)
library (ROI, quietly = TRUE)
library (ROI.plugin.glpk, quietly = TRUE)
library (ompr, quietly = TRUE)
library (ompr.roi, quietly = TRUE)
```

```
Cargo_names <- c("Rice", "Wheat", "Beans")
Profit <- matrix(c(10.3, 12, 15), ncol = 3,byrow=1, dimnames = list("Profit/Tons", Cargo_names))
Desity <- matrix(c(1.2, 1, 1.4), ncol = 3, byrow=1, dimnames = list("Desity Tons/m^3", Cargo_names))
Avail_tons <- matrix(c(100,150,200), ncol = 3, byrow=1, dimnames = list("Availability tons", Cargo_name

pander(rbind(Desity, Avail_tons, Profit))
```

	Rice	Wheat	Beans
Desity Tons/m^3	1.2	1	1.4
Availability tons	100	150	200
Profit/Tons	10.3	12	15

```
Cargo_capacity_names <- c("F","R","B","L")
Cargo_capacity_each_side <- matrix(c(80,70,70,80), ncol = 4, byrow=1,
                                   dimnames = list("Capacity m^3", Cargo_capacity_names))
pander(Cargo_capacity_each_side)
```

	F	R	B	L
Capacity m^3	80	70	70	80

Part 1: Mixed Items (Rice,Wheat,Beans) distributed in 4 holds (Front,Right,Back,Left)

```
Model_load_cargo <- MIPModel() %>%

  #xij- Number of 'i' tons (Rice=1,Wheat=2,Beans=3) in j (front=1, right=2, back=3, left=4)
  add_variable (x[i,j], i=1:3, j=1:4, type="continuous", lb=0) %>%

  set_objective (sum_expr(sum_expr(Profit[i] * x[i,j] , i=1:3),j=1:4), "max") %>%

  #Contraint1: Available tons of each item (Rice,Wheat,Bean)
  add_constraint(x[1,1]+x[1,2]+x[1,3] +x[1,4] <= 100) %>%
  add_constraint(x[2,1]+x[2,2]+x[2,3] +x[2,4] <= 150) %>%
  add_constraint(x[3,1]+x[3,2]+x[3,3] +x[3,4] <= 200) %>%

  #Constaint2: Front holds 20% to 30% of Total(450tons) weight
  add_constraint (x[1,1]+x[2,1]+x[3,1] >= 90) %>%
  add_constraint (x[1,1]+x[2,1]+x[3,1] <= 135) %>%

  #Constraint3: Back holds 20% to 30% of Total (450Tons) weight
  add_constraint (x[1,3]+x[2,3]+x[3,3] >= 90) %>%
  add_constraint (x[1,3]+x[2,3]+x[3,3] <= 135) %>%

  #Constraint4: volume m^3 = Weight/Density, hold capacity (m^3) of each side
  add_constraint(0.83*x[1,1] + x[2,1] + 0.714*x[3,1] <= 80) %>%
  add_constraint(0.83*x[1,2] + x[2,2] + 0.714*x[3,2] <= 70) %>%
  add_constraint(0.83*x[1,3] + x[2,3] + 0.714*x[3,3] <= 70) %>%
  add_constraint(0.83*x[1,4] + x[2,4] + 0.714*x[3,4] <= 80) %>%

  #Constraint5: Left and right hold weights must be within 20% of each other
  add_constraint(0.8*(x[1,2]+x[2,2]+x[3,2]) <= (x[1,4]+x[2,4]+x[3,4])) %>%
  add_constraint((x[1,4]+x[2,4]+x[3,4]) <= 1.2*(x[1,2]+x[2,2]+x[3,2])) %>%

  solve_model(with_ROI(solver = "glpk"))

Model_load_cargo
```

```
## Status: optimal
## Objective value: 4920.4
```

```
solution_table <- Model_load_cargo$solution
pander(solution_table)
```

Table 3: Table continues below

x[1,1]	x[2,1]	x[3,1]	x[1,2]	x[2,2]	x[3,2]	x[1,3]	x[2,3]	x[3,3]
96.39	0	0	0	31.63	53.74	3.614	18.6	67.78

x[1,4]	x[2,4]	x[3,4]
0	23.97	78.48

Part2: Allowed only one type of item to be stored in each hold

```
Model_load_unique_cargo <- MIPModel() %>%

  add_variable (x[i,j], i=1:3, j=1:4, type="continuous", lb=0) %>%
  add_variable (y[i,j], i=1:3, j=1:4, type="binary") %>%

  set_objective (sum_expr(sum_expr(Profit[i]*x[i,j], i=1:3),j=1:4), "max") %>%

  #Constraint1:
  add_constraint(x[1,1] + x[1,2] + x[1,3] + x[1,4] <= 100) %>%
  add_constraint(x[2,1] + x[2,2] + x[2,3] + x[2,4] <= 150) %>%
  add_constraint(x[3,1] + x[3,2] + x[3,3] + x[3,4] <= 200) %>%

  #Constraint2:
  add_constraint (x[1,1] + x[2,1] + x[3,1] >= 90) %>%
  add_constraint (x[1,1] + x[2,1] + x[3,1] <= 135) %>%

  #Constraint3:
  add_constraint (x[1,3] + x[2,3] + x[3,3] >= 90) %>%
  add_constraint (x[1,3] + x[2,3] + x[3,3] <= 135) %>%

  #Constraint4: Sum of 'y' in each hold is 1 to ensure only 1 item is selected
  add_constraint(sum_expr(y[i,1], i=1:3) == 1) %>%
  add_constraint(sum_expr(y[i,2], i=1:3) == 1) %>%
  add_constraint(sum_expr(y[i,3], i=1:3) == 1) %>%
  add_constraint(sum_expr(y[i,4], i=1:3) == 1) %>%

  #Constraint5: Consider Big'M' theory from Part1-constaint4
  add_constraint(x[1,1] <= 96*y[1,1]) %>%
  add_constraint(x[1,2] <= 84*y[1,2]) %>%
  add_constraint(x[1,3] <= 84*y[1,3]) %>%
  add_constraint(x[1,4] <= 96*y[1,4]) %>%

  add_constraint(x[2,1] <= 80*y[2,1]) %>%
  add_constraint(x[2,2] <= 70*y[2,2]) %>%
  add_constraint(x[2,3] <= 70*y[2,3]) %>%
  add_constraint(x[2,4] <= 80*y[2,4]) %>%

  add_constraint(x[3,1] <= 112*y[3,1]) %>%
  add_constraint(x[3,2] <=  98*y[3,2]) %>%
  add_constraint(x[3,3] <=  98*y[3,3]) %>%
  add_constraint(x[3,4] <= 112*y[3,4]) %>%

  #Constraint6: Left & right cargo within 20% of each other
  add_constraint(0.8*(x[1,2] + x[2,2] + x[3,2]) <= (x[1,4] + x[2,4] + x[3,4])) %>%
  add_constraint((x[1,4] + x[2,4] + x[3,4]) <= 1.2*(x[1,2] + x[2,2] + x[3,2])) %>%

  solve_model(with_ROI(solver = "glpk"))

Model_load_unique_cargo
```

```
## Status: optimal
## Objective value: 4888.8
```

```r
solution_table_unique <- Model_load_unique_cargo$solution
pander(solution_table_unique)
```

Table 5: Table continues below

x[1,1]	x[2,1]	x[3,1]	x[1,2]	x[2,2]	x[3,2]	x[1,3]	x[2,3]	x[3,3]
96	0	0	0	0	98	0	0	98

Table 6: Table continues below

x[1,4]	x[2,4]	x[3,4]	y[1,1]	y[2,1]	y[3,1]	y[1,2]	y[2,2]	y[3,2]
0	80	0	1	0	0	0	0	1

y[1,3]	y[2,3]	y[3,3]	y[1,4]	y[2,4]	y[3,4]
0	0	1	0	1	0

Important notes:

- Max profit with mixed cargo in each hold is $4920.4
- Max Profit with unique cargo in each hold is $4888.8
- The profit decreases when we select unique cargo/hold.
- Part2 of selecting unique cargos is based on Big M theory where we push other variables to zero to check the max available to assign for that particular item.
- 'y' variable is binary such that $x \leq M*y$. Implies if 'x' variable goes beyond 'M' value, this equation will force y=0 and so x becomes 0.