

Reproducible Research

The following document shows how the statistical and graphics program *R* is used to analyse data for this project. Set the working directory to where the output files for `MyCarrierPlanInterpreter.java` are. These files are available at the github repository for this project. The outputs for `MyCarrierPlanInterpreter.java` can be found at <https://github.com/fransbrooks/BPJ420FinalYearProject/tree/master/CarrierPlanInterpreter>. For the purposes of this document an analysis of day 3's demand will be shown. The relevant directories and file names can be changed to analyse the demand for any specific day. The directory specified should contain all the files produced as output for a given day of demand data. This should result in a directory for a given day of demand containing 15 .csv files.

```
setwd("/home/frans/Documents/BPJ420/CarrierPlanInterpreter/Day3")
```

Load the ggplot2 package for plotting data and load the xtable package to generate LaTeX code for tables.

```
library(ggplot2)
library(xtable)
```

Fleet Composition

Running-Fixed-Cost and Variable Cost

Read in the .csv file containing data about the fleet composition for running-fixed-cost and variable cost

```
setwd("/home/frans/Documents/BPJ420/CarrierPlanInterpreter/Day3")
FSMinfoRunFixAndVar <- read.csv("Day3_80_vehicleTypesFixedRunningAndVariablesValues.csv")
FSMinfoRunFixAndVar
```

##	vehId	score	numOfAct	totDist	totTravelTime	totActTime
## 1	truck_12_1	-759298	89	60788	3164	26700
## 2	truck_3_1_0	-759298	38	65717	2931	14100
## 3	truck_3_1	-759298	22	342276	18644	9900

Make separate tables for each type of truck for running-fixed-cost and variable cost. Do this by subsetting the data read in above. Subset by searching for a string within the `vehId` column of the `FSMinfoRunFixAndVar` data.frame, in order to identify the vehicle type. The string identifying the vehicle type is searched for by making use of the `grep` function.

```
RunFixAndVarThreeTonners <- FSMinfoRunFixAndVar[grep("truck_3",FSMinfoRunFixAndVar$vehId),]
RunFixAndVarSixTonners <- FSMinfoRunFixAndVar[grep("truck_6",FSMinfoRunFixAndVar$vehId),]
RunFixAndVarSevenTonners <- FSMinfoRunFixAndVar[grep("truck_7",FSMinfoRunFixAndVar$vehId),]
RunFixAndVarTwelveTonners <- FSMinfoRunFixAndVar[grep("truck_12",FSMinfoRunFixAndVar$vehId),]
RunFixAndVarFifteenTonners <- FSMinfoRunFixAndVar[grep("truck_15",FSMinfoRunFixAndVar$vehId),]
RunFixAndVarThreeTonners
```

##	vehId	score	numOfAct	totDist	totTravelTime	totActTime
## 2	truck_3_1_0	-759298	38	65717	2931	14100
## 3	truck_3_1	-759298	22	342276	18644	9900

Make a data.frame indicating how many vehicles of each type was specified for running-fixed-cost and variable cost.

Make a character vector containing the names of the columns for the data.frame and create an empty data.frame.

```
ColumnNames <- c("3Tonners", "6Tonners", "7Tonners", "12Tonners", "15Tonners")
TruckTotals <- data.frame(matrix(nrow=0, ncol=5))
```

Find the number of rows for each truck type from the data frames by using the length function. This will give the number of trucks for each truck type specified in the fleet composition generated by solving the FSMVRP. Append the number of vehicles specified to the data.frame.

```
TruckTotals <- rbind(c(length(RunFixAndVarThreeTonners$vehId),
  length(RunFixAndVarSixTonners$vehId),
  length(RunFixAndVarSevenTonners$vehId),
  length(RunFixAndVarTwelveTonners$vehId),
  length(RunFixAndVarFifteenTonners$vehId)))
```

Assign the column names to the data.frame TruckTotals in order to see how many of each vehicle type was specified in the solution to the FSMVRP. Name the row according to the cost components that were considered in order to generate the fleet composition.

```
colnames(TruckTotals) <- ColumnNames
rownames(TruckTotals) <- "RunFixAndVar"
TruckTotals
```

```
##           3Tonners 6Tonners 7Tonners 12Tonners 15Tonners
## RunFixAndVar      2         0         0         1         0
```

Variable Cost

Read in the .csv file containing data about the fleet composition for variable cost only.

```
setwd("/home/frans/Documents/BPJ420/CarrierPlanInterpreter/Day3")
FSMinfoVar <- read.csv("Day3_80_vehicleTypesOnlyVariablesValues.csv")
```

Make separate data frames for each type of truck for variable cost only.

```
VarThreeTonners <- FSMinfoVar[grepl("truck_3",FSMinfoVar$vehId),]
VarSixTonners <- FSMinfoVar[grepl("truck_6",FSMinfoVar$vehId),]
VarSevenTonners <- FSMinfoVar[grepl("truck_7",FSMinfoVar$vehId),]
VarTwelveTonners <- FSMinfoVar[grepl("truck_12",FSMinfoVar$vehId),]
VarFifteenTonners <- FSMinfoVar[grepl("truck_15",FSMinfoVar$vehId),]
```

Append the data for using variable cost only to the TruckTotals data.frame and change the row names to reflect the newly added data.

```
TruckTotals <- rbind(TruckTotals, c(length(VarThreeTonners$vehId),
  length(VarSixTonners$vehId), length(VarSevenTonners$vehId),
  length(VarTwelveTonners$vehId), length(VarFifteenTonners$vehId)))
rownames(TruckTotals) <- c("RunFixAndVar",
  "VarOnly")
```

Running-Fixed-Cost

Read in the .csv file containing data about the fleet composition for running-fixed-cost only.

```
setwd("/home/frans/Documents/BPJ420/CarrierPlanInterpreter/Day3")
FSMinfoRunFix <- read.csv("Day3_80_vehicleTypesOnlyFixedRunningsumValues.csv")
```

Make separate data frames for each type of truck for fixed cost only.

```
RunFixThreeTonners <- FSMinfoRunFix[grepl("truck_3",FSMinfoRunFix$vehId),]
RunFixSixTonners <- FSMinfoRunFix[grepl("truck_6",FSMinfoRunFix$vehId),]
RunFixSevenTonners <- FSMinfoRunFix[grepl("truck_7",FSMinfoRunFix$vehId),]
RunFixTwelveTonners <- FSMinfoRunFix[grepl("truck_12",FSMinfoRunFix$vehId),]
RunFixFifteenTonners <- FSMinfoRunFix[grepl("truck_15",FSMinfoRunFix$vehId),]
```

Append the data for using variable cost only to the TruckTotals data.frame.

```
TruckTotals <- rbind(TruckTotals, c(length(RunFixThreeTonners$vehId),
  length(RunFixSixTonners$vehId), length(RunFixSevenTonners$vehId),
  length(RunFixTwelveTonners$vehId), length(RunFixFifteenTonners$vehId)))
rownames(TruckTotals) <- c("RunFixAndVar",
  "VarOnly", "RunFixOnly")
```

Fixed Cost and Variable Cost

Read in the .csv file containing data about the fleet composition for fixed cost and variable cost.

```
setwd("/home/frans/Documents/BPJ420/CarrierPlanInterpreter/Day3")
FSMinfoFixAndVar <- read.csv("Day3_80_vehicleTypesAbsFixAndVarsumValues.csv")
```

Make separate data frames for each type of truck for fixed and variable cost.

```
FixAndVarThreeTonners <- FSMinfoFixAndVar[grepl("truck_3",
  FSMinfoFixAndVar$vehId), ]
FixAndVarSixTonners <- FSMinfoFixAndVar[grepl("truck_6",
  FSMinfoFixAndVar$vehId), ]
FixAndVarSevenTonners <- FSMinfoFixAndVar[grepl("truck_7",
  FSMinfoFixAndVar$vehId), ]
FixAndVarTwelveTonners <- FSMinfoFixAndVar[grepl("truck_12",
  FSMinfoFixAndVar$vehId), ]
FixAndVarFifteenTonners <- FSMinfoFixAndVar[grepl("truck_15",
  FSMinfoFixAndVar$vehId), ]
```

Append the data for using fixed and variable cost to the TruckTotals data.frame.

```
TruckTotals <- rbind(TruckTotals, c(length(FixAndVarThreeTonners$vehId),
  length(FixAndVarSixTonners$vehId), length(FixAndVarSevenTonners$vehId),
  length(FixAndVarTwelveTonners$vehId),
  length(FixAndVarFifteenTonners$vehId)))
rownames(TruckTotals) <- c("RunFixAndVar",
  "VarOnly", "RunFixOnly", "FixAndVar")
```

Fixed Cost

Read in the .csv file containing data about the fleet composition for fixed cost only.

```
setwd("/home/frans/Documents/BPJ420/CarrierPlanInterpreter/Day3")
FSMinfoFix <- read.csv("Day3_80_vehicleTypesOnlyAbsFixsumValues.csv")
```

Make separate data frames for each type of truck for fixed cost only.

```
FixThreeTonners <- FSMinfoFix[grepl("truck_3",FSMinfoFixAndVar$vehId),]
FixSixTonners <- FSMinfoFix[grepl("truck_6",FSMinfoFixAndVar$vehId),]
FixSevenTonners <- FSMinfoFix[grepl("truck_7",FSMinfoFixAndVar$vehId),]
FixTwelveTonners <- FSMinfoFix[grepl("truck_12",FSMinfoFixAndVar$vehId),]
FixFifteenTonners <- FSMinfoFix[grepl("truck_15",FSMinfoFixAndVar$vehId),]
```

Append the data for using fixed cost only to the TruckTotals data.frame

```
TruckTotals <- rbind(TruckTotals, c(length(FixThreeTonners$vehId),
  length(FixSixTonners$vehId), length(FixSevenTonners$vehId),
  length(FixTwelveTonners$vehId), length(FixFifteenTonners$vehId)))
rownames(TruckTotals) <- c("Running-Fixed-Cost and Variable Cost",
  "Variable Cost", "Running-Fixed-Cost",
  "Fixed and Variable Cost", "Fixed Cost")
```

Finalise the TruckTotals data.frame

Add a new column which shows total number of vehicles for each cost scenario by using the rowSums function. Append the new column to the TruckTotals data.frame by using the cbind function. The TruckTotals data.frame now shows the fleet composition using each set of cost parameters for day 3's demand data.

```
TruckTotals <- cbind(TruckTotals, rowSums(TruckTotals))
ColumnNames <- c("3Tonners", "6Tonners",
  "7Tonners", "12Tonners", "15Tonners",
  "Total")
colnames(TruckTotals) <- ColumnNames
TruckTotals
```

##	3Tonners	6Tonners	7Tonners	12Tonners
## Running-Fixed-Cost and Variable Cost	2	0	0	1
## Variable Cost	0	0	0	0
## Running-Fixed-Cost	1	2	0	0
## Fixed and Variable Cost	2	0	0	1
## Fixed Cost	2	0	0	1
##	15Tonners	Total		
## Running-Fixed-Cost and Variable Cost	0	3		
## Variable Cost	3	3		
## Running-Fixed-Cost	0	3		
## Fixed and Variable Cost	0	3		
## Fixed Cost	0	3		

Generate L^AT_EX Code for Fleet Compositions

Use the xtable package to generate a table in L^AT_EX code for use in L^AT_EX documents.

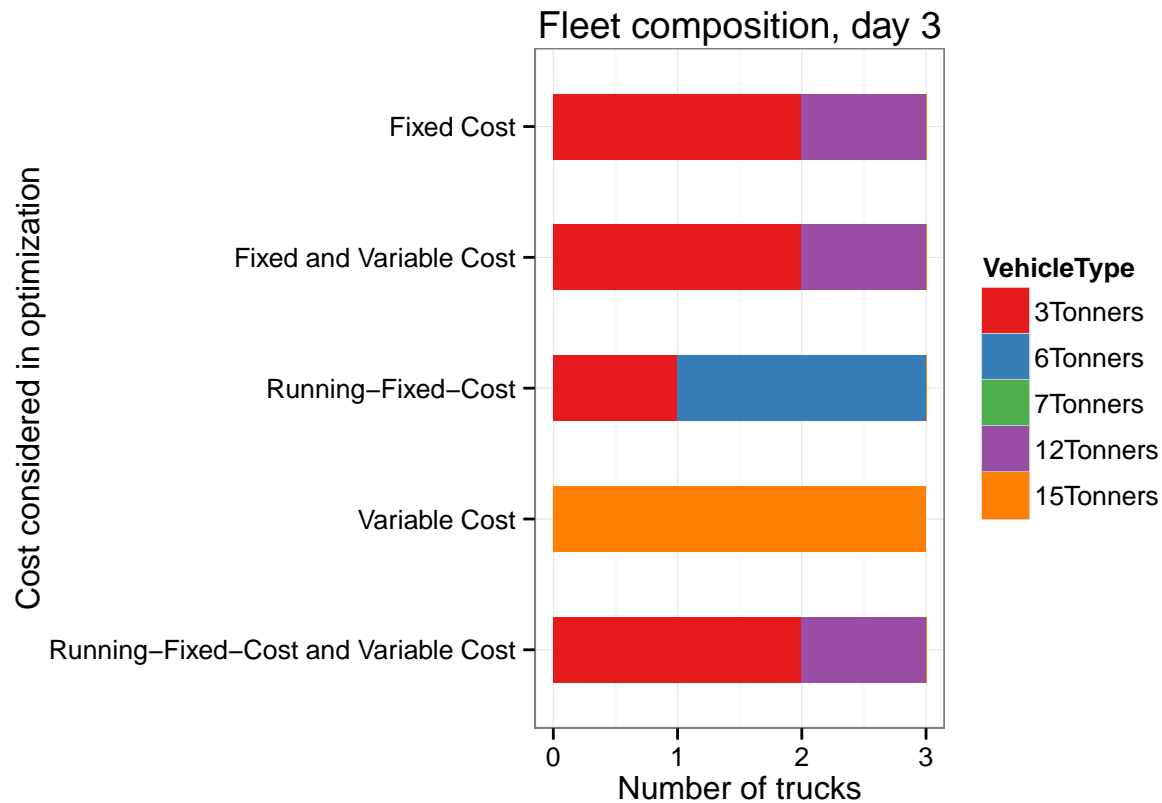
```
xtable(TruckTotals)

## % latex table generated in R 3.1.1 by xtable 1.7-3 package
## % Mon Sep 22 05:36:45 2014
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrrrr}
## \hline
## & 3Tonners & 6Tonners & 7Tonners & 12Tonners & 15Tonners & Total \\
## \hline
## Running-Fixed-Cost and Variable Cost & 2.00 & 0.00 & 0.00 & 1.00 & 0.00 & 3.00 \\
## Variable Cost & 0.00 & 0.00 & 0.00 & 0.00 & 3.00 & 3.00 \\
## Running-Fixed-Cost & 1.00 & 2.00 & 0.00 & 0.00 & 0.00 & 3.00 \\
## Fixed and Variable Cost & 2.00 & 0.00 & 0.00 & 1.00 & 0.00 & 3.00 \\
## Fixed Cost & 2.00 & 0.00 & 0.00 & 1.00 & 0.00 & 3.00 \\
## \hline
## \end{tabular}
## \end{table}
```

Fleet compositions plot

Change the TruckTotals data.frame into the ‘long format’ using the melt from the reshape2 package. This makes it possible to plot the fleet compositions for all the costs in one plot by means of stacked bars. The Totals column must be removed from TruckTotals, because the total size of the fleet will be represented by the stacked bar. Spaces in the column names are also removed, so the columns can be referred to in the ggplot function. Referring to column names with spaces in the ggplot function generates an error.

```
TruckTotals <- TruckTotals[, -6] #Remove the 'Totals' column
library(reshape2)
m.TruckTotals <- melt(TruckTotals)
colnames(m.TruckTotals) <- c("CostParameters", "VehicleType", "NumOfVehicles")
StackPlot <- ggplot(data = m.TruckTotals, aes(x = CostParameters, y = NumOfVehicles,
  fill = VehicleType)) + geom_bar(stat = "identity", width = 0.5)
StackPlot <- StackPlot + coord_flip() + xlab("Cost considered in optimization") +
  ylab("Number of trucks") + scale_fill_brewer(palette = "Set1")
StackPlot <- StackPlot + theme_bw() + ggtitle("Fleet composition, day 3")
StackPlot
```



Distance travelled plot

Add a column called “Cost” to each data.frame which contains the information for each model run. Add a variable to each row of the Cost column which describes the cost components considered in the model. Because the cost components used is the same for each data.frame, each row can be assigned the same value for the Cost variable. Create a new data.frame, FSMinfo. This data.frame will be appended with all the data from each model run, so the distance information can be plotted in a single plot.

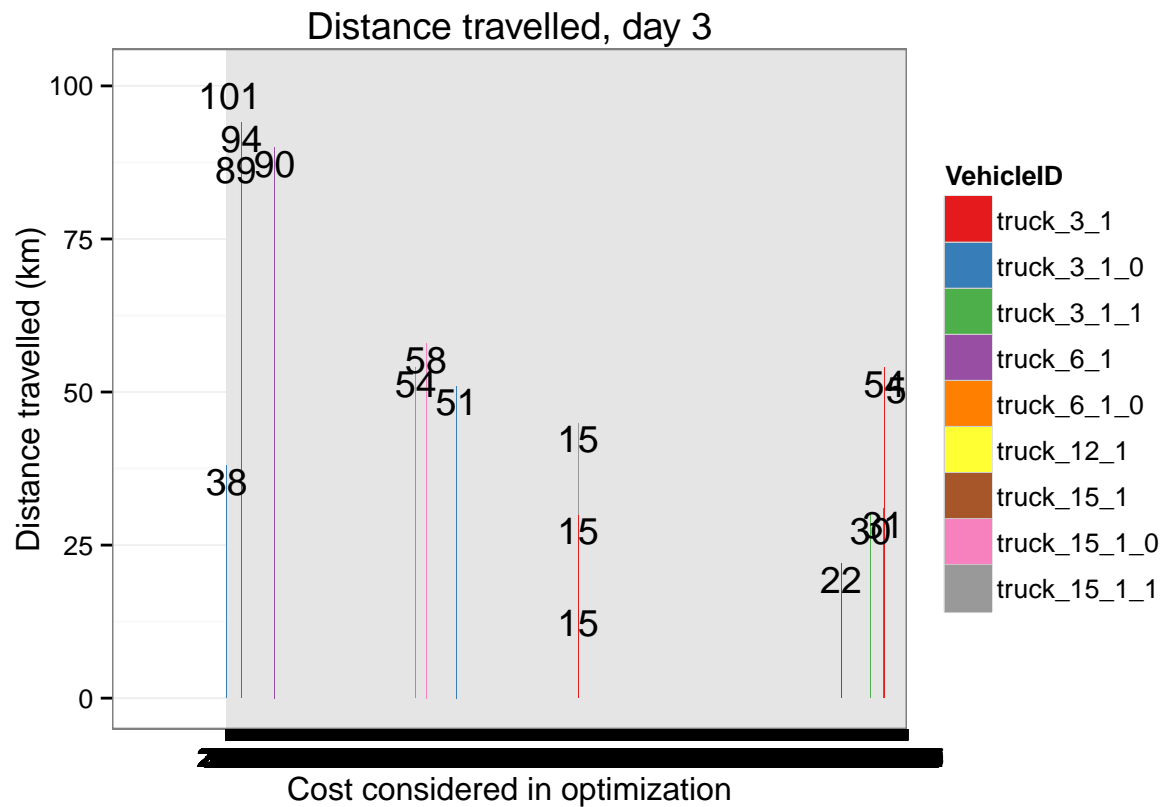
```
FSMinfoFix["Cost"] <- "Fixed cost"
FSMinfoFixAndVar["Cost"] <- "Fixed and variable cost"
FSMinfoRunFix["Cost"] <- "Running-fixed-cost"
FSMinfoRunFixAndVar["Cost"] <- "Running-fixed-cost and variable cost"
FSMinfoVar["Cost"] <- "Variable cost"
FSMinfo <- data.frame()
FSMinfo <- rbind(FSMinfoFix, FSMinfoFixAndVar,
  FSMinfoRunFix, FSMinfoRunFixAndVar, FSMinfoVar)
colnames(FSMinfo) <- c("VehicleID", "NumOfActivities",
  "TotalDistKM", "TotalTravelTimeHours",
  "Cost")
FSMinfo$VehicleID <- factor(x = FSMinfo$VehicleID,
  c("truck_3_1", "truck_3_1_0", "truck_3_1_1",
    "truck_6_1", "truck_6_1_0", "truck_12_1",
    "truck_15_1", "truck_15_1_0", "truck_15_1_1"))
FSMinfo$TotalDistKM <- round(FSMinfo$TotalDistKM,
  0)
DistanceStackPlot <- ggplot(data = FSMinfo,
```

```

aes(x = Cost, y = TotalDistKM, fill = VehicleID)) +
  geom_bar(stat = "identity", width = 0.5)
DistanceStackPlot <- DistanceStackPlot +
  scale_fill_brewer(palette = "Set1") +
  xlab("Cost considered in optimization") +
  ylab("Distance travelled (km)")
DistanceStackPlot <- DistanceStackPlot +
  theme_bw() + ggtitle("Distance travelled, day 3") +
  geom_text(aes(label = TotalDistKM), position = "stack",
    hjust = 0.5, vjust = 1.1)
DistanceStackPlot <- DistanceStackPlot +
  scale_x_discrete(labels = c(`Fixed and variable cost` = "Fixed and \nvariable cost",
    `Running-fixed-cost and variable cost` = "Running-fixed- \ncost and \nvariable cost",
    `Running-fixed-cost` = "Running-fixed- \ncost"))
DistanceStackPlot

```

ymax not defined: adjusting position using y instead



The code used to generate this .pdf plot for the project's final document is identical, but for an unknown reason the plot does not display correctly in this .pdf document, which is generated by 'knitting' this R Markdown document to a .pdf file.