To: Dr. Gregory Macfarlane

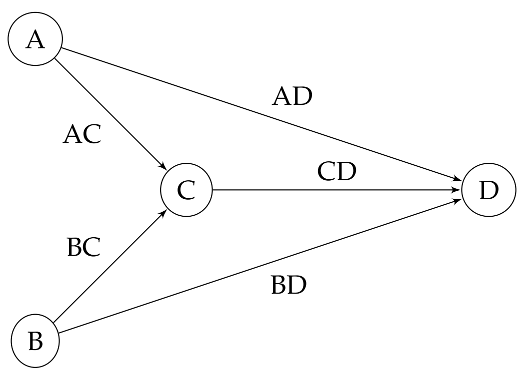
From: Dr. Gregory Macfarlane

Subject: Route Assignment Lab Solution

Date: March 12, 2014

```
library(knitr)
library(plyr)
# make \Sexpr{} output look pretty
knit_hooks$set(inline = function(x) {
   prettyNum(x, big.mark=",")
})
options(width=60)
# set knitr options
opts_chunk$set(cache=TRUE, echo=TRUE,
              tidy=FALSE, size="small", autodep=TRUE)
```

You have been given[1] the following network:



7000 vehicles travel from A to D, and 5000 from B to D (node C generates no trips). Link travel time functions are given as

$$
\begin{aligned}
t_{AD} &= 20.0 + 0.01V_{AD} \\
t_{AC} &= 10.0 + 0.005V_{AC} \\
t_{CD} &= 12.0 + 0.005V_{CD} \\
t_{BC} &= 7.25 + 0.005V_{BC} \\
t_{BD} &= 20.0 + 0.01V_{BD}
\end{aligned}
$$

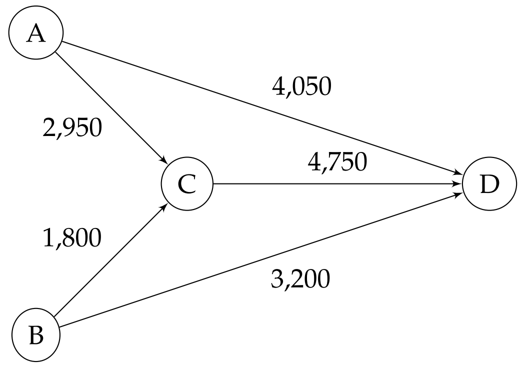
1. Solve for the user equilibrium (UE) link flows and travel times (HINT: write and solve a set of simultaneous equations that explicitly define the UE conditions). Demonstrate that your solution is the user equilibrium by showing through example that all UE conditions are satisfied.

```
userequilibrium <- function(unknowns,costfunctions, info=FALSE){
  volumes <- vector("numeric", length=5)
```

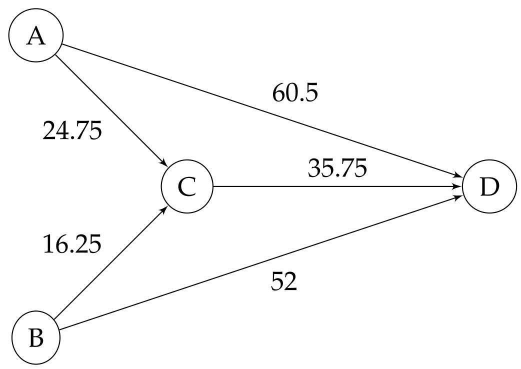
[1] This is adapted from Dr. John Ivan at UConn

```
  # put in volume constraints; these constraints make only two unkowns
  volumes[1] <- unknowns[1]
  volumes[2] <- 7000 - unknowns[1]
  volumes[4] <- 5000 - unknowns[2]
  volumes[5] <- unknowns[2]
  volumes[3] <- volumes[2] + volumes[4]

  # compute travel time on each link
  times <- rowSums(costfunctions * cbind(1, volumes))

  # compute path travel times

  traveltimes <- c(times[1], times[2] + times[3], times[3]+times[4],
                   times[5])

  # User Equilibrium condition:
  #   At equilibrium, all available paths will have precisely
  #   equal travel times. Want to iterate until minimum is acheived.
  #   (if all equal, max(traveltimes) - min(traveltimes) = 0)
  stat <- max(traveltimes) - min(traveltimes)


  if(info==TRUE){
    return(list("Link Volumes"= volumes,
                "Link Times" = times,
                "Path Times" = traveltimes))
  }else{
    return(stat)
  }
}

costfunctions <- matrix(c(20,0.01,10,0.005,12,0.005,7.25,0.005,20, 0.01),
                nrow=5, ncol=2, byrow=TRUE)

# Find the volumes that define user equilibrium (equal travel times).
optimize <- nlm(userequilibrium,p=c(500,500), costfunctions=costfunctions,
                typsize= c(2000,2000))
optimize$estimate

## [1] 4050 3200

# What are the link times and volumes at this equilibrium?
results <- userequilibrium(optimize$estimate, costfunctions, info=TRUE)
results

## $Link Volumes
## [1] 4050 2950 4750 1800 3200
##
## $Link Times
## [1] 60.50 24.75 35.75 16.25 52.00
##
## $Path Times
## [1] 60.5 60.5 52.0 52.0
```

Link volumes:



Link travel times:



Another way to solve it Another way to get this answer would be to create a matrix of the all of the equations and solve for its solution.

```
# left-hand matrix A
system <- matrix(c(1, 0, 0, 0, 0,-0.01, 0,     0,     0,     0,
                   0, 1, 0, 0, 0, 0,   -0.005, 0,     0,     0,
                   0, 0, 1, 0, 0, 0,    0,    -0.005, 0,     0,
                   0, 0, 0, 1, 0, 0,    0,     0,    -0.005, 0,
                   0, 0, 0, 0, 1, 0,    0,     0,     0,    -0.01,
                  -1, 1, 1, 0, 0, 0,    0,     0,     0,     0,
                   0, 0, 1, 1,-1, 0,    0,     0,     0,     0,
                   0, 0, 0, 0, 0, 0,    1,    -1,     1,     0,
                   0, 0, 0, 0, 0, 1,    1,     0,     0,     0,
                   0, 0, 0, 0, 0, 0,    0,     0,     1,     1),
                 nrow=10,ncol=10, byrow=TRUE)
# right-hand vector b
constraints <- matrix(c(20, 10, 12, 7.25, 20, 0, 0,0,7000,5000),
                      nrow=10, ncol=1)
# compute solution x, Ax=b
solve(system, constraints)
```

```
##          [,1]
##  [1,]   60.50
##  [2,]   24.75
##  [3,]   35.75
##  [4,]   16.25
##  [5,]   52.00
##  [6,] 4050.00
##  [7,] 2950.00
```

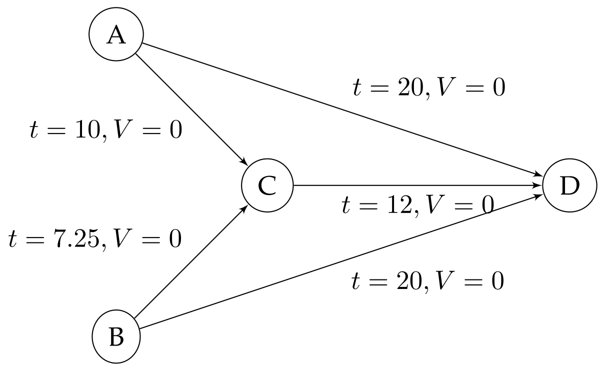
```
##  [8,] 4750.00
##  [9,] 1800.00
## [10,] 3200.00
```

As you can see, this leads to identical results.

2. Perform four iterations of All Or Nothing (AON) assignment on the network and O/D volumes. Show the link flows and travel times at the end of each iteration.

```
volumes <- c(0,0,0,0,0)
times <- rowSums(costfunctions * cbind(1, volumes))
Volumes <- list()
```

All-or-Nothing Iteration 0:

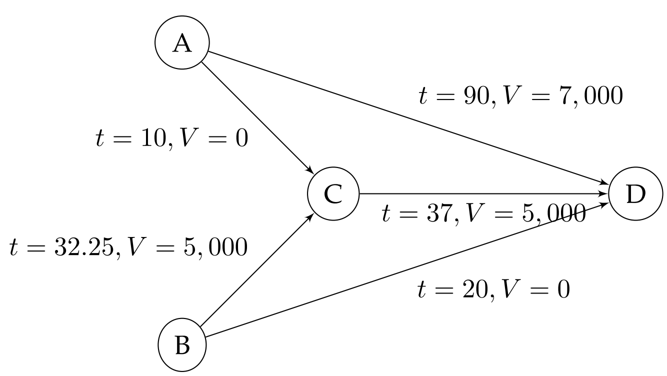


Total travel times are:

AD 20
ACD 22
BCD 19.25
BD 20

```
volumes <- c(7000, 0, 5000, 5000, 0)
Volumes[[1]] <- volumes
times <- rowSums(costfunctions * cbind(1, volumes))
```

All-or-Nothing Iteration 1:

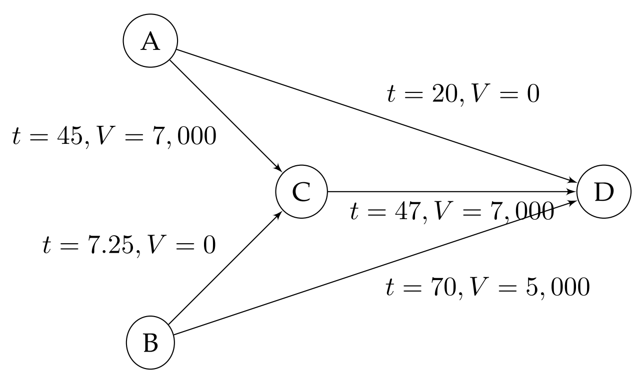


Total travel times are:

AD 90
ACD 47
BCD 69.25
BD 20

```
volumes <- c(0, 7000, 7000, 0, 5000)
Volumes[[2]] <- volumes
times <- rowSums(costfunctions * cbind(1, volumes))
```

All-or-Nothing Iteration 2:

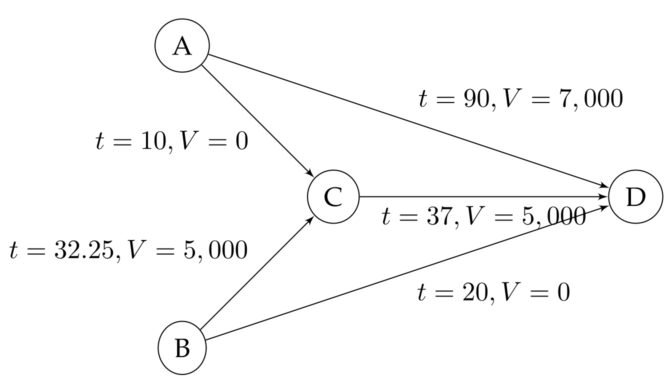


Total travel times are:

AD 20
ACD 92
BCD 54.25
BD 70

```
volumes <- c(7000, 0, 5000, 5000, 0)
Volumes[[3]] <- volumes
times <- rowSums(costfunctions * cbind(1, volumes))
```

All-or-Nothing Iteration 3:

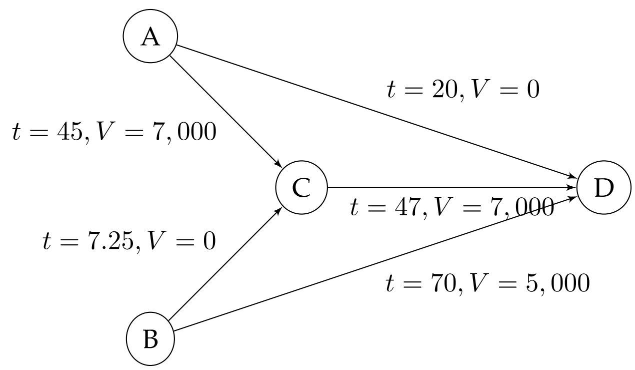


Total travel times are:

AD 90
ACD 47
BCD 69.25
BD 20

```
volumes <- c(0, 7000, 7000, 0, 5000)
Volumes[[4]] <- volumes
times <- rowSums(costfunctions * cbind(1, volumes))
```

All-or-Nothing Iteration 4:

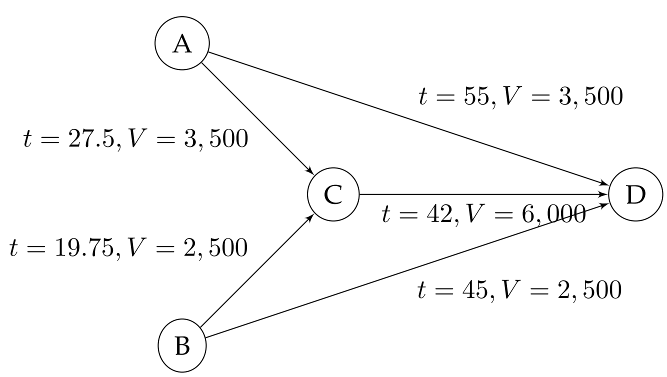


Total travel times are:

AD 20
ACD 92
BCD 54.25
BD 70

```
volumes <- colMeans(laply(Volumes, as.matrix))
times <- rowSums(costfunctions * cbind(1, volumes))
```

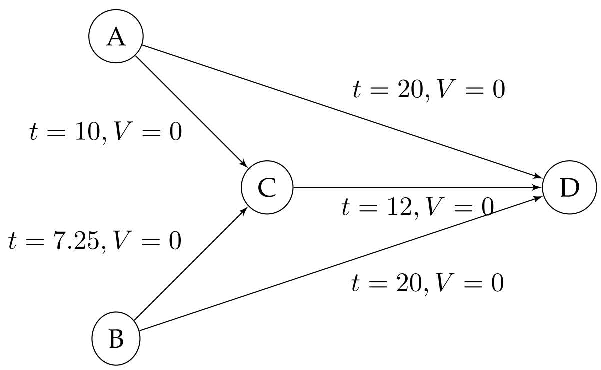
All-or-Nothing Final Average:



3. Perform an incremental assignment, using trip table increments of 25% for each step. Show the link flows and travel times at the end of each incremental assignment.

```
volumes <- c(0,0,0,0,0)
times <- rowSums(costfunctions * cbind(1, volumes))
```

Incremental Iteration 0:

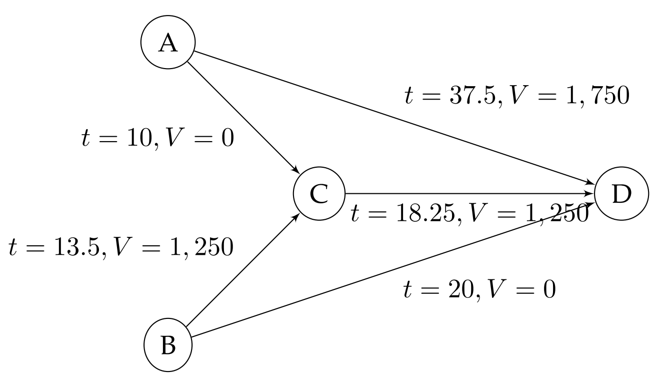


Total travel times are:

AD 20
ACD 22
BCD 19.25
BD 20

```
volumes <- c(1750,0,1250,1250,0)
times <- rowSums(costfunctions * cbind(1, volumes))
```

Incremental Iteration 1:

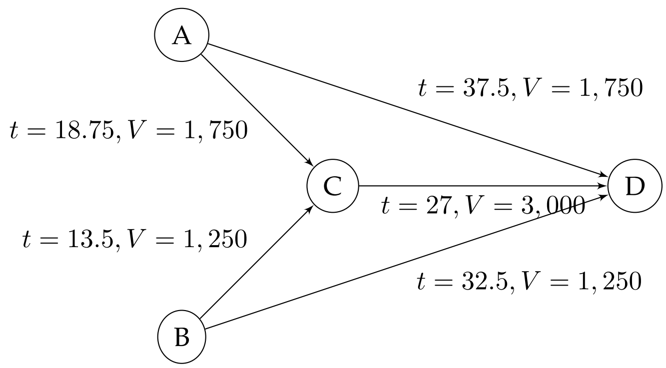


Total travel times are:

AD 37.5
ACD 28.25
BCD 31.75
BD 20

```
volumes <- c(1750,1750,3000,1250,1250)
times <- rowSums(costfunctions * cbind(1, volumes))
```

Incremental Iteration 2:



Total travel times are:

AD 37.5

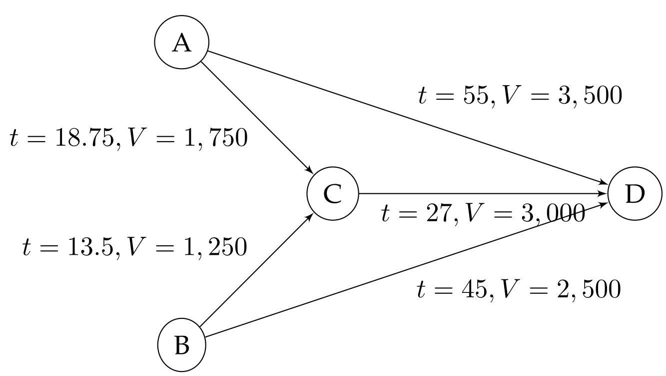
ACD 45.75

BCD 40.5

BD 32.5

```
volumes <- c(3500,1750,3000,1250,2500)
times <- rowSums(costfunctions * cbind(1, volumes))
```

Incremental Iteration 3:



Total travel times are:

AD 55

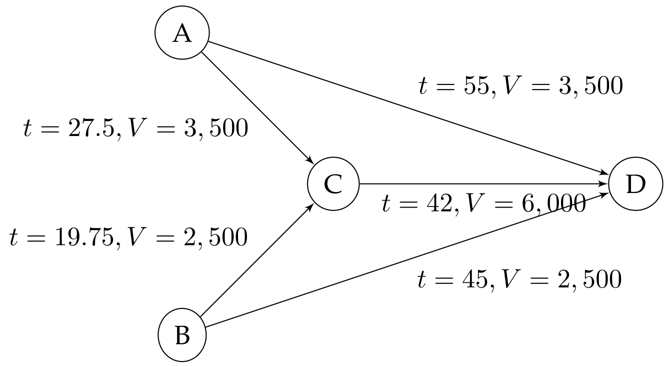
ACD 45.75

BCD 40.5

BD 45

```
volumes <- c(3500,3500,6000,2500,2500)
times <- rowSums(costfunctions * cbind(1, volumes))
```

Incremental Iteration 4 (Final Assignment):



Total travel times are:

AD 55

ACD 69.5

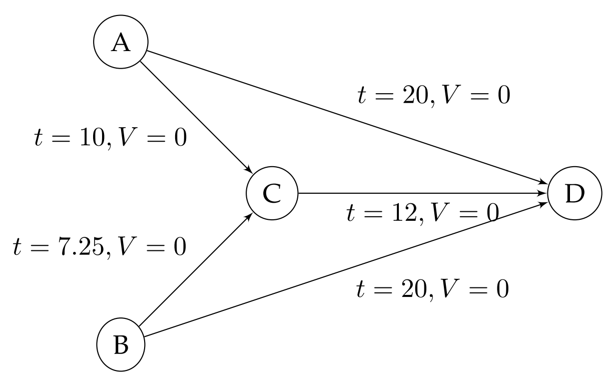
BCD 61.75

BD 45

4. Assign trips using the FHWA assignment heuristic. Show the link flows and travel times for four assignments, and the final average assignment and resulting travel times.

```
volumes <- c(0,0,0,0,0)
Times <- list()
Volumes <- list()
Times[[1]] <- rowSums(costfunctions * cbind(1, volumes))
times <- Times[[1]]
```

FHWA Iteration 0:



Total travel times are:

AD 20

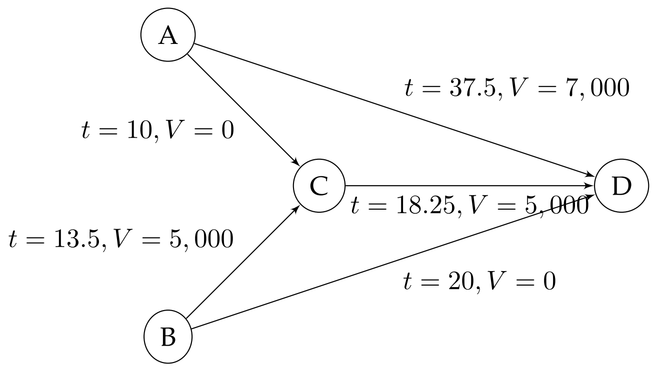
ACD 22

BCD 19.25

BD 20

```
volumes <- c(7000,0,5000,5000,0)
Volumes[[1]] <- volumes
Times[[2]] <- rowSums(costfunctions * cbind(1, volumes))
times <- 0.75 * Times[[1]] + 0.25 *Times[[2]]
```

FHWA Iteration 1:



Total travel times are:

AD 37.5

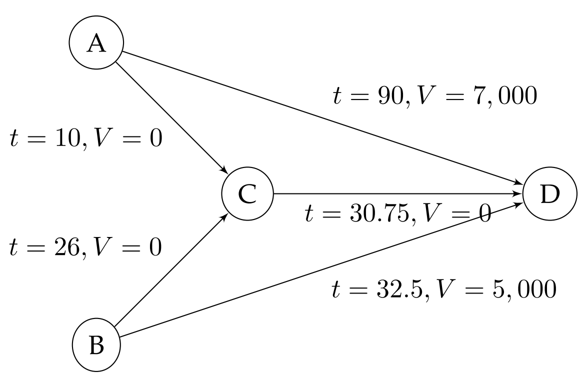
ACD 28.25

BCD 31.75

BD 20

```
volumes <- c(7000,0,0,0,5000)
Volumes[[2]] <- volumes
Times[[3]] <- rowSums(costfunctions * cbind(1, volumes))
times <- 0.75 * Times[[2]] + 0.25 *Times[[3]]
```

FHWA Iteration 2:

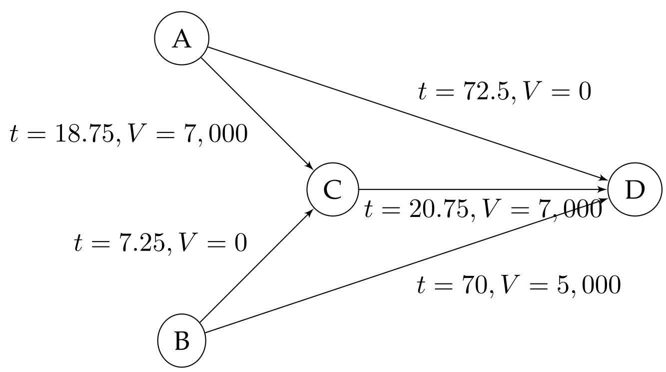


Total travel times are:

AD 90
ACD 40.75
BCD 56.75
BD 32.5

```
volumes <- c(0,7000,7000,0,5000)
Volumes[[3]] <- volumes
Times[[4]] <- rowSums(costfunctions * cbind(1, volumes))
times <- 0.75 * Times[[3]] + 0.25 *Times[[4]]
```

FHWA Iteration 3:

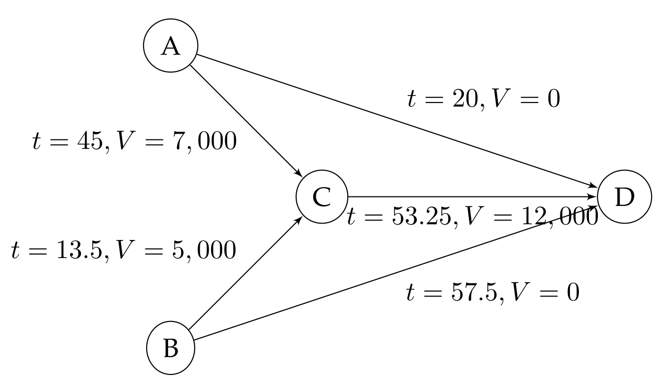


Total travel times are:

AD 72.5
ACD 39.5
BCD 28
BD 70

```
volumes <- c(0,7000,12000,5000,0)
Volumes[[4]] <- volumes
Times[[5]] <- rowSums(costfunctions * cbind(1, volumes))
times <- 0.75 * Times[[4]] + 0.25 *Times[[5]]
```

FHWA Iteration 4:



Total travel times are:

AD 20

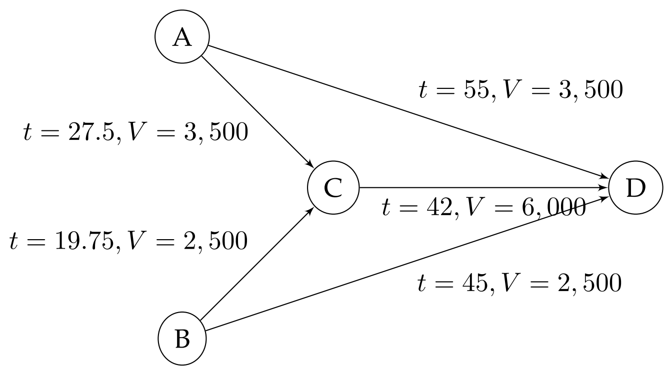
ACD 98.25

BCD 66.75

BD 57.5

```r
volumes <- colMeans(laply(Volumes, as.matrix))
times <- rowSums(costfunctions * cbind(1, volumes))
```

FHWA Averages:



Total travel times are:

AD 55

ACD 69.5

BCD 61.75

BD 45

5. Compare these four traffic assignment heuristic approaches to the UE assignment and to each other. How do the resulting flow patterns differ (cite specific differences)? Which one comes closest to the UE flows?

6. State in words the general theory underlying each of the heuristic approaches. Which one do you prefer and why? Consider accuracy, ease of computation and the underlying theory.