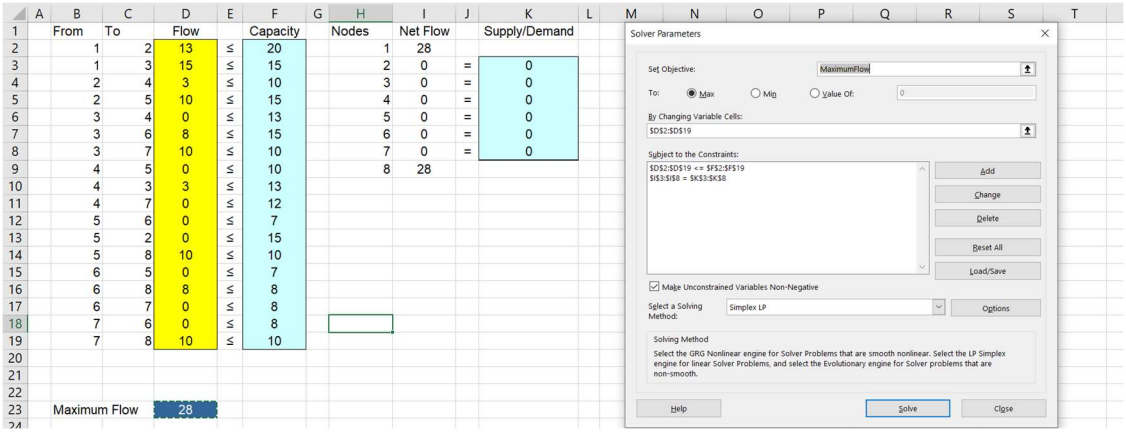
Willy Wonka

Thomas Courtney

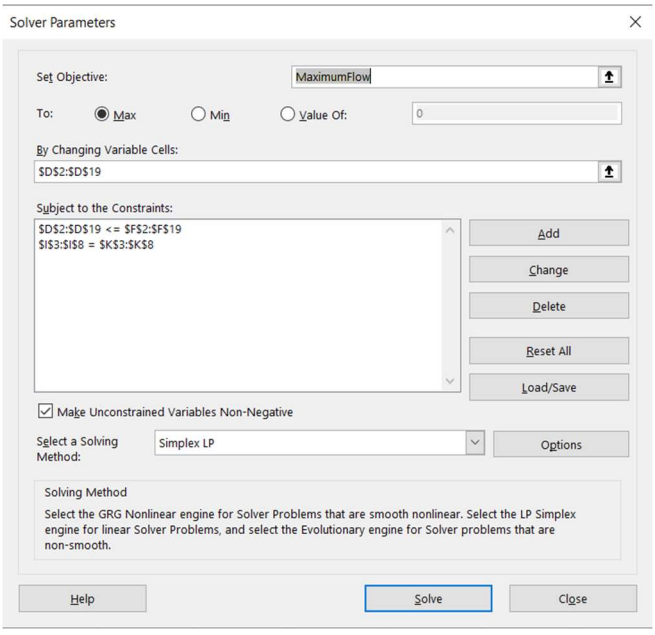
12/1/2020

#1

Below is an example of my Excel model I did to answer the question. Additionally, there is an example of my solver constraint.



	From	To	Flow		Capacity	Nodes	Net Flow		Supply/Demand
	1	2	13	≤	20	1	28		
	1	3	15	≤	15	2	0	=	0
	2	4	3	≤	10	3	0	=	0
	2	5	10	≤	15	4	0	=	0
	3	4	0	≤	13	5	0	=	0
	3	6	8	≤	15	6	0	=	0
	3	7	10	≤	10	7	0	=	0
	4	5	0	≤	10	8	28		
	4	3	3	≤	13				
	4	7	0	≤	12				
	5	6	0	≤	7				
	5	2	0	≤	15				
	5	8	10	≤	10				
	6	5	0	≤	7				
	6	8	8	≤	8				
	6	7	0	≤	8				
	7	6	0	≤	8				
	7	8	10	≤	10				
	Maximum Flow		28						



#2

I then solve the same algorithm as I did in the excel model below using R. I get the same answer, being 28. Which confirms accuracy of both.

```
nodes <- 1:9
maps<- matrix(c(1,2,20,
                1,3,15,
                2,4,10,
                2,5,15,
                3,4,13,
                3,6,15,
                3,7,10,
                4,3,13,
                4,5,10,
                4,7,12,
                5,2,15,
                5,6,7,
                5,8,10,
                6,5,7,
                6,7,8,
                6,8,8,
                7,6,8,
                7,8,10), byrow=TRUE, ncol = 3)

maxValue<-maxFlowFordFulkerson(nodes,maps,source.node=1,sink.node = 8)
maxValue
```

```
## $s.cut
## [1] 1 2 3 5 4 6 7
##
## $t.cut
## [1] 8 9
##
## $max.flow
## [1] 28
```

Sensitivity Analysis

I then run the sensitivity analysis to determine if there is a limit to the increasing the expansion factor of K at 3. This also matches the answer I received in Python.

```
k <-1:100
max.flows<- c()

for(i in k){
  mapsx<-matrix(c(1,2,i*20,
                  1,3,i*15,
                  2,4,10,
```

```
                  2,5,15,
                  3,4,13,
                  3,6,15,
                  3,7,10,
                  4,3,13,
                  4,5,10,
                  4,7,12,
                  5,2,15,
                  5,6,7,
                  5,8,i*10,
                  6,5,7,
                  6,7,i*8,
                  6,8,i*8,
                  7,6,i*8,
                  7,8,10), byrow=TRUE, ncol = 3)
  x<-graph_from_edgelist(mapsx[,1:2],directed=T)
  E(x)$capacity<-mapsx[,3]
  tf<-max_flow(x,source=1, target=8)
  max.flows[i]<-tf$value
}

sensitivity<- cbind(k, max.flows)
max_k<-which.max(sensitivity[,2])
a<-sensitivity[max_k,]
a

##         k max.flows
##         3        62
```

#3

I then run the reduced row echelon form below.

```
A = maps
library(pracma)
rref(A)

##       [,1] [,2] [,3]
##  [1,]    1    0    0
##  [2,]    0    1    0
##  [3,]    0    0    1
##  [4,]    0    0    0
##  [5,]    0    0    0
##  [6,]    0    0    0
##  [7,]    0    0    0
##  [8,]    0    0    0
##  [9,]    0    0    0
## [10,]    0    0    0
## [11,]    0    0    0
## [12,]    0    0    0
## [13,]    0    0    0
```

```
## [14,]    0    0    0
## [15,]    0    0    0
## [16,]    0    0    0
## [17,]    0    0    0
## [18,]    0    0    0
```

```
round(rref(A), 2)
```

```
##       [,1] [,2] [,3]
##  [1,]    1    0    0
##  [2,]    0    1    0
##  [3,]    0    0    1
##  [4,]    0    0    0
##  [5,]    0    0    0
##  [6,]    0    0    0
##  [7,]    0    0    0
##  [8,]    0    0    0
##  [9,]    0    0    0
## [10,]    0    0    0
## [11,]    0    0    0
## [12,]    0    0    0
## [13,]    0    0    0
## [14,]    0    0    0
## [15,]    0    0    0
## [16,]    0    0    0
## [17,]    0    0    0
## [18,]    0    0    0
```