# Homework 7

Ben Ridenhour

08-Nov-21

## Contents



For the Chapter 7 homework, you were asked to perform the optimization (maximization and minimization) using the Simplex algorithm of various linear programs in the book. The questions are given on pg. 278 of the book, and you were to do problems 8–12. Additionally, you were asked plot each system and the progression of the algorithm at each step.

### The Simplex Algorithm

First, while not specified, it would probably be easiest to code the algorithm up as a function. For every system, we need to specify the following:

1. The matrix **A**

2. The cost (optimization) vector **c**
3. The starting coordinates for the optimization **x**
4. Whether we want to perform maximization or minimization

and

5. (Optionally) Whether to return the points the algorithm passed through.

Let's create this function first:

```
Simplex <- function(A, cost, x, Max = T, vals = F){

  Cols <- seq_len(ncol(A))
  B <- which(x!=0)
  N <- setdiff(Cols, B)
  out <- matrix(x, nrow = 1)

  enter <- 1

  while(! is.null(enter)){
    lambda <- solve(t(A[,B])) %*% cost[B]
    sn <- cost[N] - t(A[,N])%*% lambda #for maximization: continue as long as max(sn) > 0

    sn.pos <- sn[sn>0]
```

```
    sn.neg <- sn[sn<0]

    enter <- NULL

    if(Max & length(sn.pos > 0)) enter <- N[which.max(sn)]
    if(!Max & length(sn.neg > 0)) enter <- N[which.min(sn)]

    if( is.null(enter) ) break

    d <- solve(A[,B]) %*% A[,enter]
    ratios <- x[B]/d #get the ratios for the ratio test

    multiplier <- min(ratios[d > 0]) #value of entering variable
    exit <- B[which(ratios == multiplier)]

    x[enter] <- multiplier
    x[B] <- x[B] - d*multiplier

    out <- rbind(out, x)

    B <- sort(union(setdiff(B, exit), enter))
    N <- setdiff(Cols, B)
  }

  rownames(out) <- NULL
  if(vals) return(out)

  return(out[nrow(out),])
}
```

Alright, we should be able to get everything we need from that function

## Problem 8

The linear program to optimize is

$$\begin{aligned} \text{Optimize} \quad & 2x + 3y \\ \text{subject to} \quad & \\ & 2x + 3y \geq 6 \\ & 3x - y \leq 15 \\ & -x + y \leq 4 \\ & 2x + 5y \leq 27 \\ & x, y \geq 0 \end{aligned}$$

Note that the first constraint equation is $\geq$ and needs to be converted to $\leq$, giving $-2x - 3y \leq -6$.

```
A <-  c(
  -2,  -3, 1, 0, 0, 0,
  3, -1, 0, 1, 0, 0,
  -1, 1, 0, 0, 1, 0,
  2, 5, 0, 0, 0, 1
)
```

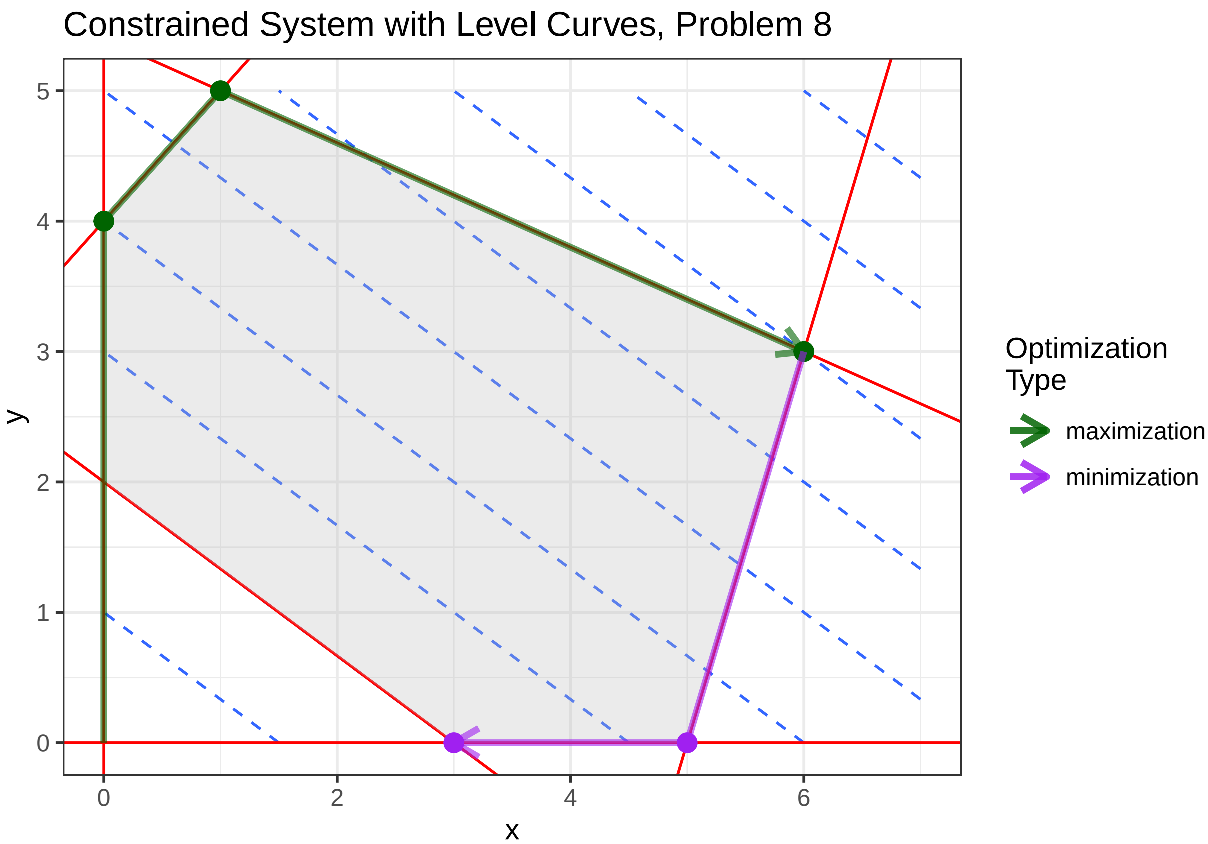
```r
A <- matrix(A, nrow = 4, byrow = T)
C <- c(2, 3, 0, 0, 0, 0)
x <- c(0, 0, -6, 15, 4, 27)

maximum <- Simplex(A,C,x,vals = T) #last row has the maximum
maximum[nrow(maximum),]
```

```
## [1]  6  3 15  0  7  0
```

```r
minimum <- Simplex(A,C,maximum[nrow(maximum),], Max = F, vals=T) #last row has the minimum
minimum[nrow(minimum),]
```

```
## [1]  3  0  0  6  7 21
```

```r
##plot it

library(ggplot2)

### A plot of the system

data <- merge(data.frame(x=seq(0,7,length=50)), data.frame(y=seq(0,5,length=50)))
data$z <- 2*data$x + 3*data$y
shadeMe <- data.frame(x=c(3,5,6,1,0,0),y=c(0,0,3,5,4,2))

maxD<- data.frame(x = maximum[,1], y = maximum[,2])
minD<- data.frame(x = minimum[,1], y = minimum[,2])

ggplot(data, aes(x,y)) + geom_contour(aes(z=z), bins = 10, linetype =2) + geom_abline(aes(intercept = 2
  geom_hline(aes(yintercept = 0 ), col="red") +
  geom_vline(aes(xintercept = 0), col="red") +
  geom_abline(aes(intercept = -15, slope = 3), col="red") +
  geom_abline(aes(intercept = 4, slope = 1), col="red") +
  geom_abline(aes(intercept = 27/5, slope = -2/5), col="red") +
  geom_polygon(data = shadeMe, aes(x,y), fill = "gray", alpha = 0.3) +
  geom_path(data=maxD, aes(color = "darkgreen"), alpha = 0.6, lwd = 1.2, arrow = arrow(length = unit(0.
  geom_point(data = maxD[-1,], col="darkgreen", size =3) +
  geom_path(data=minD, aes(color = "purple"), alpha = 0.6, lwd = 1.2, arrow = arrow(length = unit(0.15,
  geom_point(data = minD[-1,], col="purple", size =3) +
  theme_bw() + ggtitle("Constrained System with Level Curves, Problem 8") +
  scale_color_identity(name = "Optimization\nType",labels = c("maximization", "minimization"), guide =
```

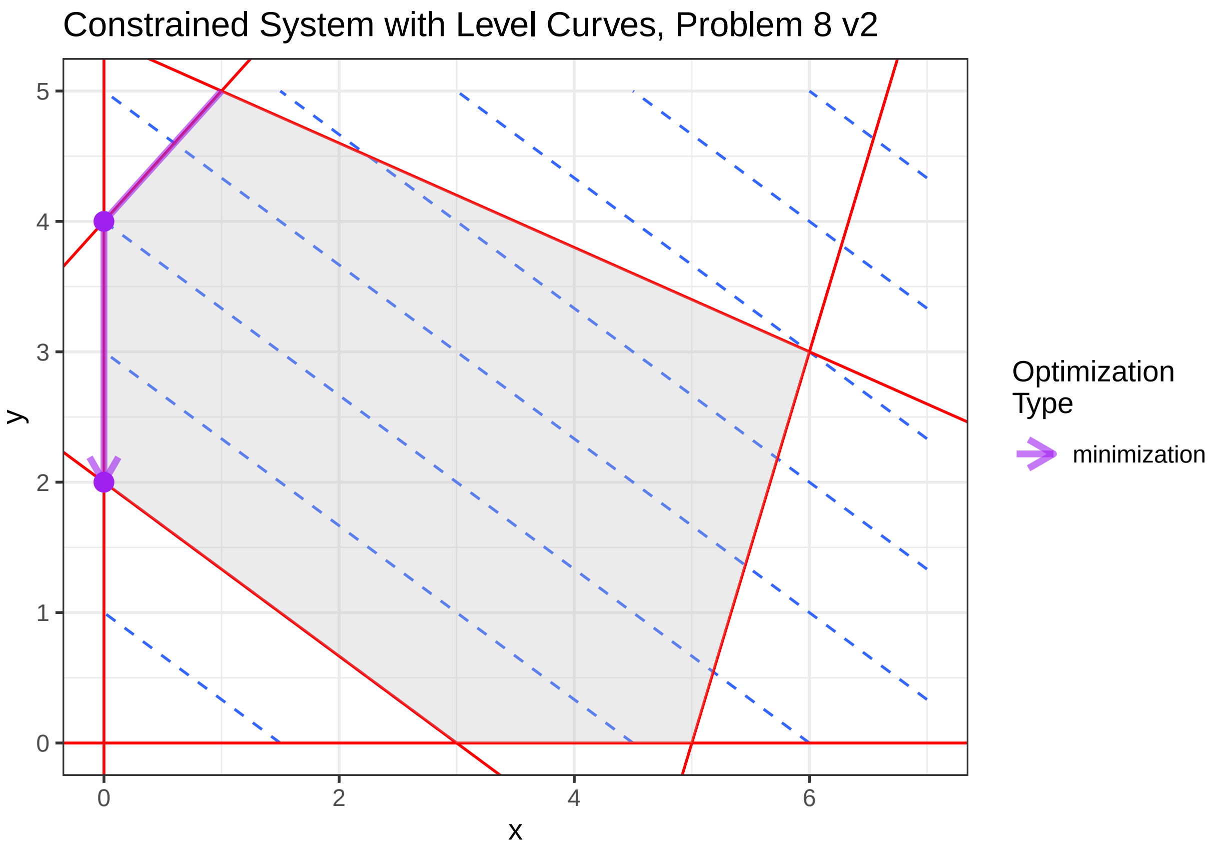



The algorithm has appropriate identify the maximum that occurs at $(6, 3)$ and a minimum that occurs at $(3, 0)$. Let's quickly try the minimization starting from a different extreme point, specifically $\mathbf{x}^T = \begin{bmatrix} 1 & 5 & -11 & 17 & 0 & 0 \end{bmatrix}$.

```
x <- c(1, 5, 11, 17, 0, 0)

minimum <- Simplex(A,C,x, Max = F, vals=T) #last row has the minimum
minimum[nrow(minimum),]
```

```
## [1]  0  2  0 17  2 17
```

```
minD<- data.frame(x = minimum[,1], y = minimum[,2])

ggplot(data, aes(x,y)) + geom_contour(aes(z=z), bins = 10, linetype =2) + geom_abline(aes(intercept = 2
  geom_hline(aes(yintercept = 0 ), col="red") +
  geom_vline(aes(xintercept = 0), col="red") +
  geom_abline(aes(intercept = -15, slope = 3), col="red") +
  geom_abline(aes(intercept = 4, slope = 1), col="red") +
  geom_abline(aes(intercept = 27/5, slope = -2/5), col="red") +
  geom_polygon(data = shadeMe, aes(x,y), fill = "gray", alpha = 0.3) +
  geom_path(data=minD, aes(color = "purple"), alpha = 0.6, lwd = 1.2, arrow = arrow(length = unit(0.15,
  geom_point(data = minD[-1,], col="purple", size =3) +
  theme_bw() + ggtitle("Constrained System with Level Curves, Problem 8 v2") +
  scale_color_identity(name = "Optimization\nType",labels = c("minimization"), guide = "legend")
```




Starting from this alternate extreme point, the Simplex algorithm identifies a different minimum $(0, 2)$. Is this okay? Why?

### Problem 9

The linear program to optimize is

$$\begin{aligned} \text{Optimize} \quad 6x + 4y \\ \text{subject to} \\ -x + y \leq 12 \\ x + y \leq 24 \\ 2x + 5y \leq 80 \\ x, y \geq 0 \end{aligned}$$

Rather than starting the at the origin (which is somewhat boring for this system), let's start at a different extreme point, specifically $\mathbf{x}^T = \begin{bmatrix} 0 & 12 & 0 & 12 & 20 \end{bmatrix}$

```
A <-  c(
  -1, 1, 1, 0, 0,
  1, 1, 0, 1, 0,
  2, 5, 0, 0, 1
)

A <- matrix(A, nrow = 3, byrow = T)
C <- c(6, 4, 0, 0, 0)
x <- c(0, 12, 0, 12, 20)
```

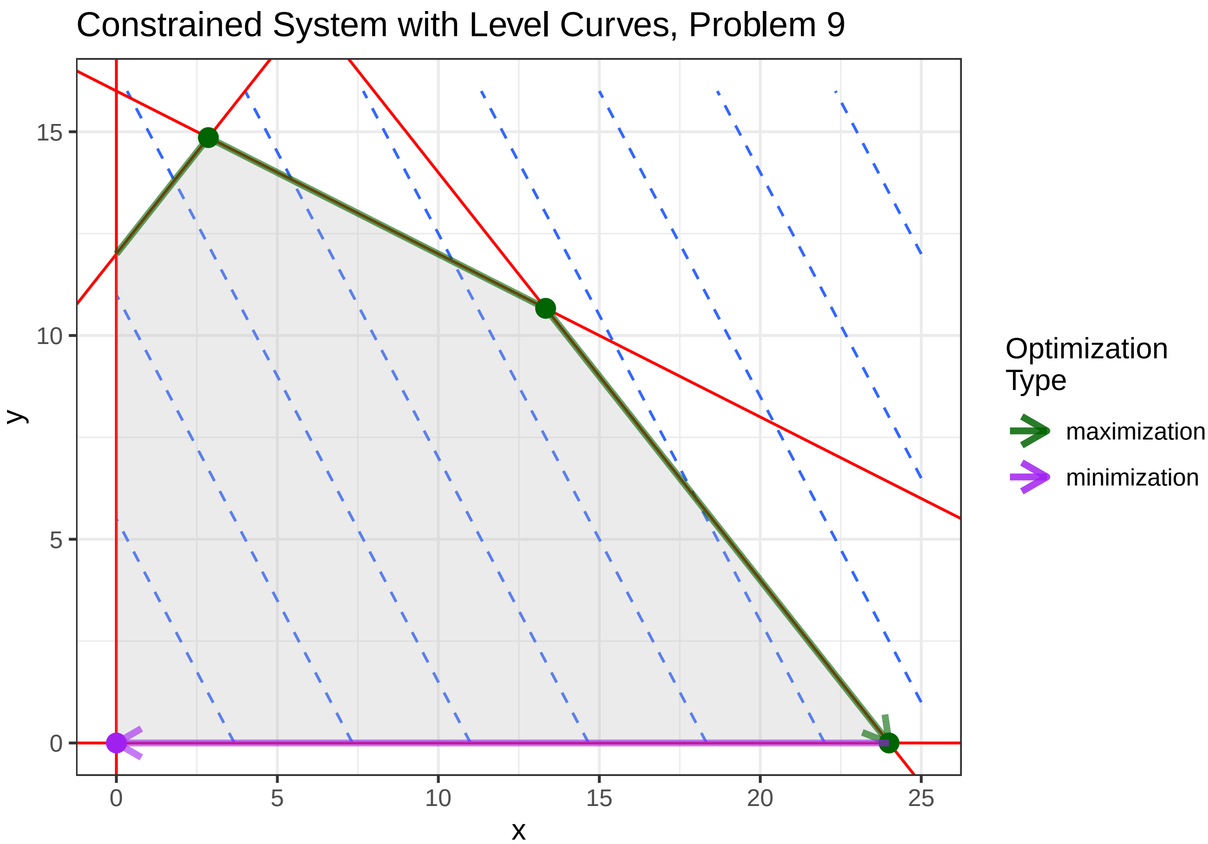
```r
maximum <- Simplex(A,C,x,vals = T) #last row has the maximum
maximum[nrow(maximum),]
```

```
## [1] 24  0 36  0 32
```

```r
minimum <- Simplex(A,C,maximum[nrow(maximum),], Max = F, vals=T) #last row has the minimum
minimum[nrow(minimum),]
```

```
## [1]  0  0 12 24 80
```

```r
### A plot of the system
data <- merge(data.frame(x=seq(0,25,length=50)), data.frame(y=seq(0,16,length=50)))
data$z <- 6*data$x + 4*data$y
shadeMe <- data.frame(x=c(20/7,40/3,24,0,0),y=c(104/7,32/3,0,0,12))

maxD<- data.frame(x = maximum[,1], y = maximum[,2])
minD<- data.frame(x = minimum[,1], y = minimum[,2])

ggplot(data, aes(x,y)) + geom_contour(aes(z=z), bins = 10, linetype =2) + geom_abline(aes(intercept = 1
  geom_hline(aes(yintercept = 0 ), col="red") +
  geom_vline(aes(xintercept = 0), col="red") +
  geom_abline(aes(intercept = 24, slope = -1), col="red") +
  geom_abline(aes(intercept = 16, slope = -2/5), col="red") +
  geom_polygon(data = shadeMe, aes(x,y), fill = "gray", alpha = 0.3) +
  geom_path(data=maxD, aes(color = "darkgreen"), alpha = 0.6, lwd = 1.2, arrow = arrow(length = unit(0.
  geom_point(data = maxD[-1,], col="darkgreen", size =3) +
  geom_path(data=minD, aes(color = "purple"), alpha = 0.6, lwd = 1.2, arrow = arrow(length = unit(0.15,
  geom_point(data = minD[-1,], col="purple", size =3) +
  theme_bw() + ggtitle("Constrained System with Level Curves, Problem 9") +
  scale_color_identity(name = "Optimization\nType",labels = c("maximization", "minimization"), guide =
```




Again, we see that simple algorithm appropriately identifies both the minimum $(0,0)$ and maximum $(24,0)$ by examining the values of the optimization function at different vertices of the convex polytope.

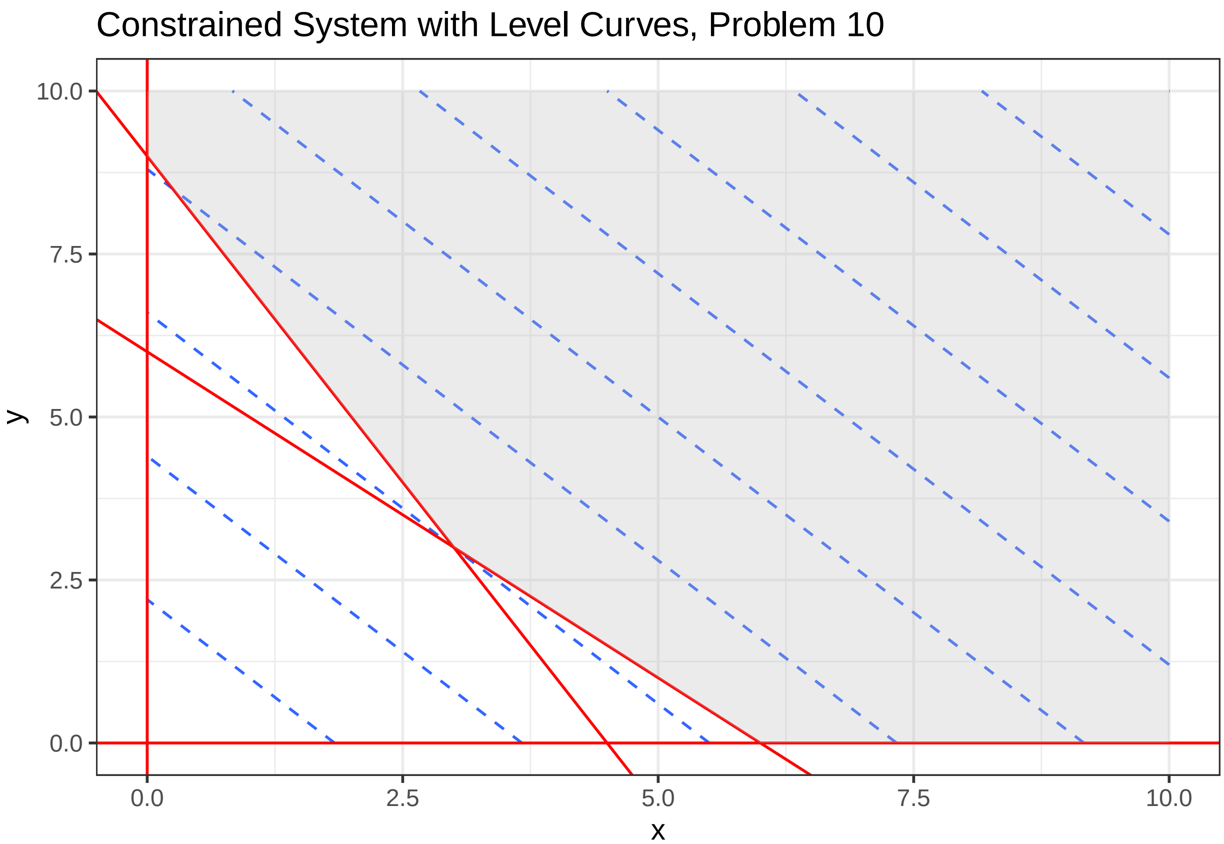
## Problem 10

The linear program to optimize is

$$\begin{aligned} \text{Optimize} \quad & 6x+5y \\ \text{subject to} & \\ x+y &\geq 6 \\ 2x+y &\geq 9 \\ x,y &\geq 0 \end{aligned}$$

Because all of the equations are expressed as $\geq$, let's plot this system first and see what it looks like.

```
### A plot of the system
data <- merge(data.frame(x=seq(0,10,length=50)), data.frame(y=seq(0,10,length=50)))
data$z <- 6*data$x + 5*data$y
shadeMe <- data.frame(x=c(3,6,10,10,0,0),y=c(3,0,0,10,10,9))

ggplot(data, aes(x,y)) + geom_contour(aes(z=z), bins = 10, linetype =2) + geom_abline(aes(intercept = 6
  geom_hline(aes(yintercept = 0 ), col="red") +
  geom_vline(aes(xintercept = 0), col="red") +
  geom_abline(aes(intercept = 9, slope = -2), col="red") +
  geom_polygon(data = shadeMe, aes(x,y), fill = "gray", alpha = 0.3) +
  theme_bw() + ggtitle("Constrained System with Level Curves, Problem 10") +
  scale_color_identity(name = "Optimization\nType",labels = c("maximization", "minimization"), guide =
```




Obviously, there is a bit of problem with this system: it is **unbounded**. Let's try our algorithm and see what it does:

```
A <-  c(
  -1, -1, 1, 0,
  -2, -1, 0, 1
  )

A <- matrix(A, nrow = 2, byrow = T)
C <- c(6, 5, 0, 0)
x <- c(0, 0, -6, -9)

maximum <- Simplex(A,C,x,vals = T) #fails due to unbounded problem
```

```
## Warning in min(ratios[d > 0]): no non-missing arguments to min; returning Inf
```

```
## Error in solve.default(t(A[, B])): 'a' (3 x 2) must be square
```

The algorithm fails to find a maximum due to the unbounded nature of the problem. Let's try a minimum starting from the origin:

```
x <- c(0, 0, -6, -9)
minimum <- Simplex(A,C,x, Max = F, vals=T) #last row has the minimum
minimum[nrow(minimum),]
```

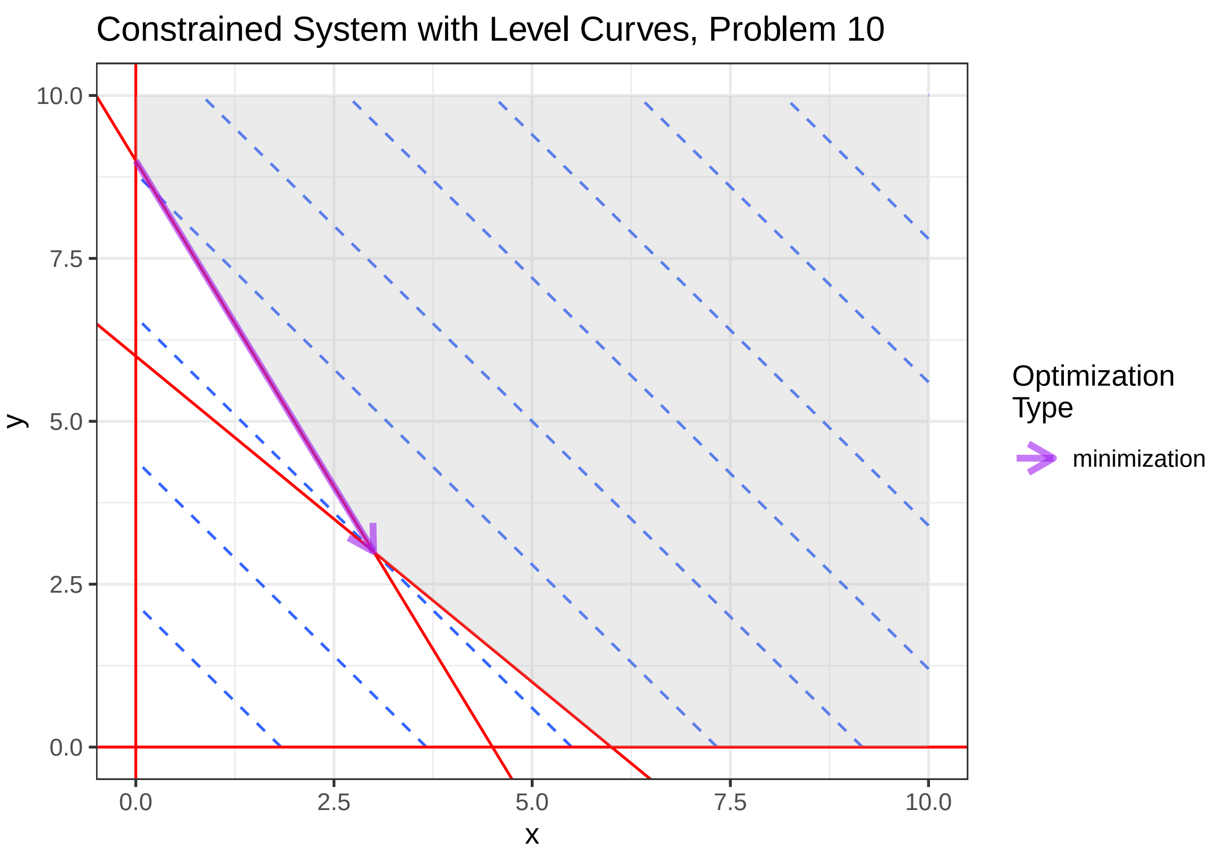
```
## [1]  0  0 -6 -9
```

The algorithm failed to move from the origin. **Why?**

```
x <- c(0,9,3,0)
minimum <- Simplex(A,C,c(0,9,3,0), Max = F, vals=T) #last row has the minimum
minimum[nrow(minimum),]
```

```
## [1] 3 3 0 0
```

```
minD<- data.frame(x = minimum[,1], y = minimum[,2])

ggplot(data, aes(x,y)) + geom_contour(aes(z=z), bins = 10, linetype =2) + geom_abline(aes(intercept = 6
  geom_hline(aes(yintercept = 0 ), col="red") +
  geom_vline(aes(xintercept = 0), col="red") +
  geom_abline(aes(intercept = 9, slope = -2), col="red") +
  geom_polygon(data = shadeMe, aes(x,y), fill = "gray", alpha = 0.3) +
  geom_path(data=minD, aes(color = "purple"), alpha = 0.6, lwd = 1.2, arrow = arrow(length = unit(0.15,
  theme_bw() + ggtitle("Constrained System with Level Curves, Problem 10") +
  scale_color_identity(name = "Optimization\nType",labels = c("minimization"), guide = "legend")
```




Our algorithm now works **if** we start it at a feasible extreme point. Starting the algorithm at a non-feasible extreme point erroneously leaves the algorithm at a non-feasible point because the value of the optimization function is less there than at the actual constrained minimum $(3, 3)$.

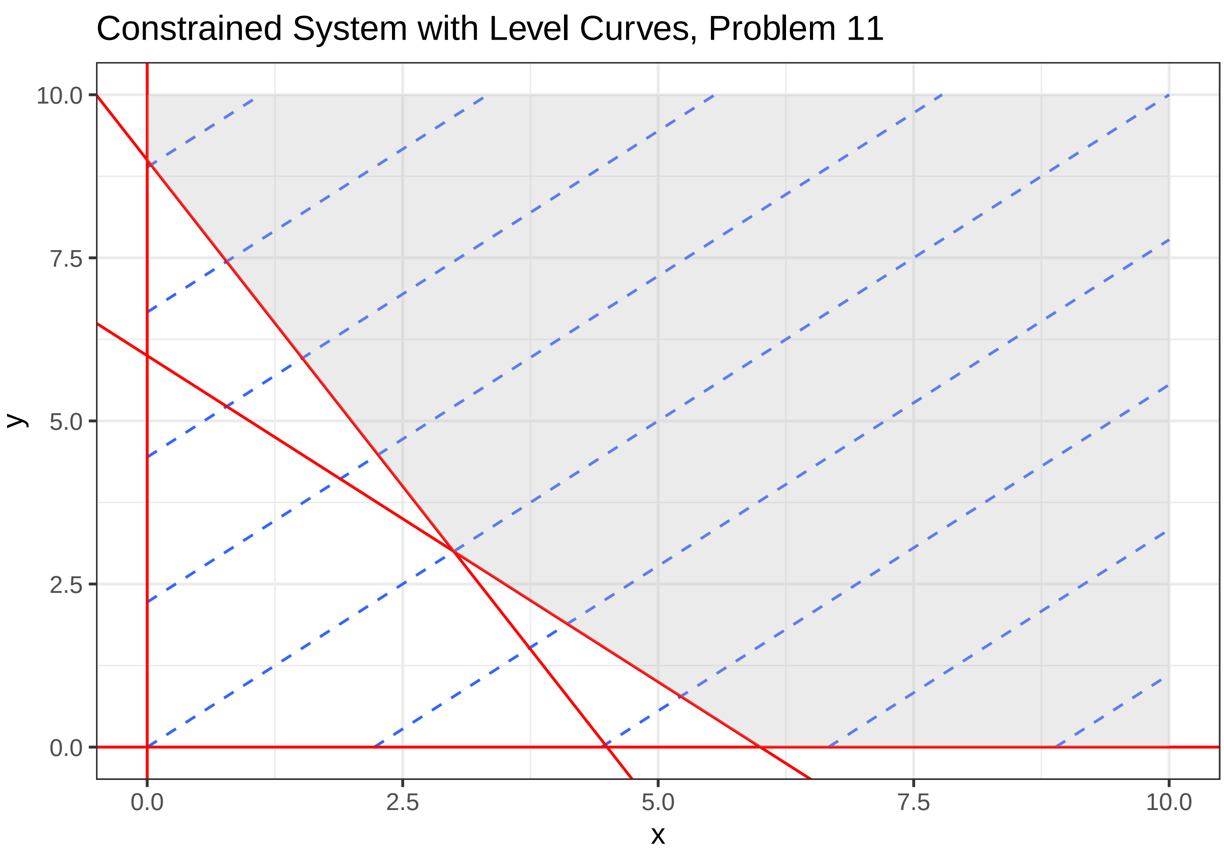
## Problem 11

Problem 11 is a slight modification of Problem 10. The optimization function for Problem 11 is:

$$\text{Optimize} \quad x - y$$

with the same constraints. Let's plot this system:

```r
### A plot of the system
data <- merge(data.frame(x=seq(0,10,length=50)), data.frame(y=seq(0,10,length=50)))
data$z <- data$x  - data$y
shadeMe <- data.frame(x=c(3,6,10,10,0,0),y=c(3,0,0,10,10,9))

ggplot(data, aes(x,y)) + geom_contour(aes(z=z), bins = 10, linetype =2) + geom_abline(aes(intercept = 6
  geom_hline(aes(yintercept = 0 ), col="red") +
  geom_vline(aes(xintercept = 0), col="red") +
  geom_abline(aes(intercept = 9, slope = -2), col="red") +
  geom_polygon(data = shadeMe, aes(x,y), fill = "gray", alpha = 0.3) +
  theme_bw() + ggtitle("Constrained System with Level Curves, Problem 11") +
  scale_color_identity(name = "Optimization\nType",labels = c("maximization", "minimization"), guide =
```




Clearly, the only difference here are the level curves. Will this matter to our algorithm? To avoid the problem with not starting at a feasible points let's use $\mathbf{x}^T = \begin{bmatrix} 0 & 9 & 3 & 0 \end{bmatrix}$ as our starting point.

```r
A <-  c(
  -1, -1, 1, 0,
  -2, -1, 0, 1
  )

A <- matrix(A, nrow = 2, byrow = T)
C <- c(1,-1, 0, 0)
x <- c(0, 9, 3, 0)

maximum <- Simplex(A,C,x,vals = T) #fails due to unbounded problem
```

```
## Warning in min(ratios[d > 0]): no non-missing arguments to min; returning Inf
```

```
## Error in solve.default(t(A[, B])): 'a' (3 x 2) must be square
```

So we can see the maximization still fails. Let's try the minimization:

```
minimum <- Simplex(A,C,x,Max = F, vals = T)
```

```
## Warning in min(ratios[d > 0]): no non-missing arguments to min; returning Inf
```

```
## Error in solve.default(t(A[, B])): 'a' (3 x 2) must be square
```

The minimization now also fails. **Why?**

## Problem 12

The final linear program for optimization was specified as follows:

$$\begin{aligned} \text{Optimize} \quad 5x + 3y \\ \text{subject to} \\ 1.2x + 0.6y \leq 24 \\ 2x + 1.5y \leq 80 \\ x, y \geq 0 \end{aligned}$$

Our system looks relatively straight-forward for optimization. Note that the extreme points of the **convex polytope** are $(0, 0)$, $(20, 0)$, and $(0, 40)$. Let's try it:

```
A<- c(1.2, 0.6, 1, 0,
      2, 1.5, 0, 1
      )

A <- matrix(A, nrow =2 , byrow=T)
C <- c(5, 3, 0, 0)
x <- c(0, 0, 24, 80)

maximum <- Simplex(A,C,x,vals=T)
maximum[nrow(maximum), ]
```

```
## [1]  0 40  0 20
```

```
minimum <- Simplex(A, C, maximum[nrow(maximum),], Max = F, vals = T)
minimum[nrow(minimum),]
```

```
## [1]  0  0 24 80
```

```
### A plot of the system
data <- merge(data.frame(x=seq(0,45,length=50)), data.frame(y=seq(0,65,length=50)))
data$z <- 5*data$x + 3*data$y
shadeMe <- data.frame(x=c(0, 20, 0),y=c(0, 0, 40))

maxD<- data.frame(x = maximum[,1], y = maximum[,2])
minD<- data.frame(x = minimum[,1], y = minimum[,2])

ggplot(data, aes(x,y)) + geom_contour(aes(z=z), bins = 10, linetype =2) + geom_abline(aes(intercept = 4
  geom_hline(aes(yintercept = 0 ), col="red") +
  geom_vline(aes(xintercept = 0), col="red") +
  geom_abline(aes(intercept = 80/1.5, slope = -2/1.5), col="red") +
  geom_polygon(data = shadeMe, aes(x,y), fill = "gray", alpha = 0.3) +
  geom_path(data=maxD, aes(color = "darkgreen"), alpha = 0.6, lwd = 1.2, arrow = arrow(length = unit(0.
  geom_point(data = maxD[-1,], col="darkgreen", size =3) +
```

```
  geom_path(data=minD, aes(color = "purple"), alpha = 0.6, lwd = 1.2, arrow = arrow(length = unit(0.15,
  geom_point(data = minD[-1,], col="purple", size =3) +
  theme_bw() + ggtitle("Constrained System with Level Curves, Problem 12") +
  scale_color_identity(name = "Optimization\nType",labels = c("maximization", "minimization"), guide =
```