DATA 609 HW9

Bin Lin

2017-10-21

Page 385:

1a. Using the definition provided for the movement diagram, determine whether the following zero-sum games have a pure strategy Nash equilibrium. If the game does have a pure strategy Nash equilibrium, state the Nash equilibrium. Assume the row player is maximizing his payoffs which are shown in the matrices below.

a.

		Colin	
		C1	C2
Rose	R1	10	10
	R2	5	0

Figure 1

To determine if a movement diagram has pure strategy Nash equilibrium, we have to draw a vertical arrow from the smaller to the largest row value in each column, and we draw a horizontal arrow from the smaller to the largest column value in each row. As shown in figure 1, all arrows point into the up left corner (R1, C1) and no arrow exits that outcome. This indicates that neither player can unilaterally improve by departing unilaterally from its strategy associated with that outcome. For Rose, she would definitely choose R1 strategy to maximize the outcome. That is why we have a pure strategy Nash equilibrium.

c.

		Pitcher	
		Fastball	Knuckleball
Batter	Guesses fastball	.400 0.6	0.9 .100
	Guesses knuckleball	.300 0.7	0.75 .250

Figure 2

1c: As shown in figure 2, all arrows point into the lower right hand corner and no arrow exits that outcome. This indicates that neither player can unilaterally improve by departing unilaterally from its strategy associated with that outcome. For Pitcher, he or she sould throw knuckleball to have success rate of 0.75. For batter, he or she should guess knuckleball to have success rate of 0.25. So then, there is a pure strategy Nash equilibrium.

Page 404: 2. For problems a-g build a linear programming model for each player's decisions and solve it both geometrically and algebraically. Assume the row player is maximizing his payoffs which are shown in the matrices below.

For Rose, she wants to Maximize A

A <= 10x + 5(1-x) Colin C1 Strategy

A <= 10x Colin C2 Strategy

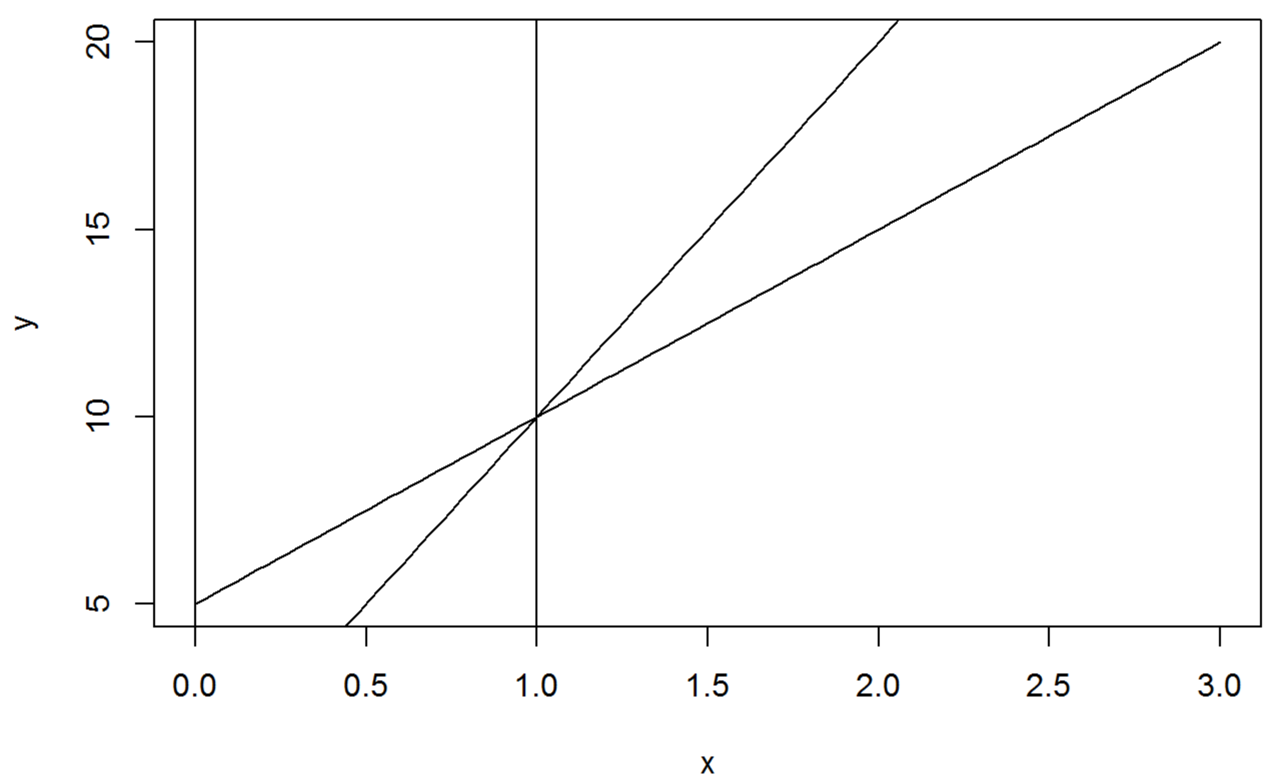
x >= 0

x <= 1

As shown in the following graph, lines are intersect at (1, 10). Thereore, for rose, she has to choose R1 strategy to maximize outcome, there is no other better alternative.

```
rose1 <- function(x){10 * x + 5 *(1 - x)}
rose2 <- function(x){10 * x}

curve(rose1, from=0, to=3, xlab="x", ylab="y")
curve(rose2, from=0, to=3, xlab="x", ylab="y", add = TRUE)
abline(v = 1)
abline(v = 0)
```



For colin, she wants to Minimize A

A >= 10y + 10(1-y) = 10 Rose R1 Strategy

A >= 5y Rose R2 Strategy

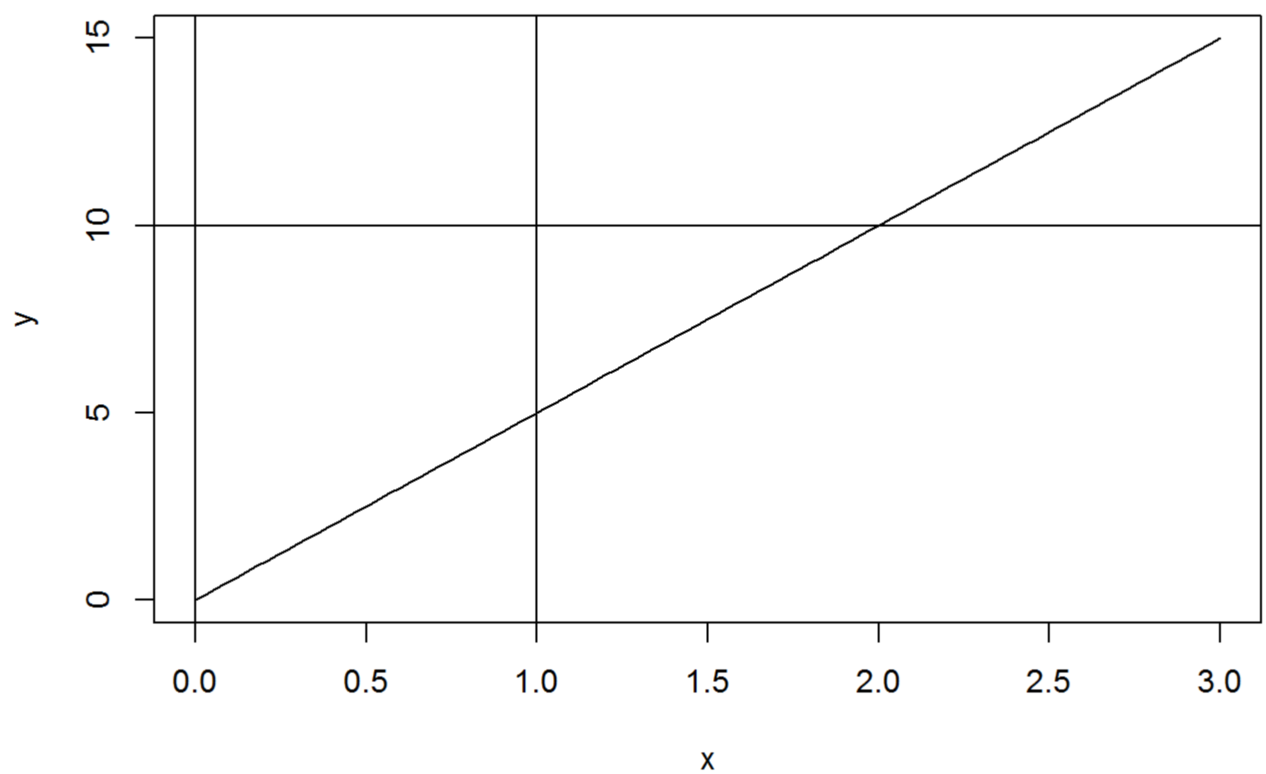
y >= 0

y <= 1

As shown in the following graph, lines are intersect at (1, 10). Thereore, for rose, she has to choose R1 strategy to maximize outcome, there is no other better alternative.

```
colin2 <- function(y){5 * y}

curve(colin2, from=0, to=3, xlab="x", ylab="y")
abline(a = 10, b = 0)
abline(v = 1)
abline(v = 0)
```



According to the graph, if Rose adopts pure R1 strategy, the minimum value for A is 10 no matter what strategy Colin choose. If Rose adopts pure R2 strategy, the minimum value for A is 0 when y = 0.

3. We are considering three alternatives A, B, or C or a mix of the three alternatives under uncertain conditions of the economy. The payoff matrix is as follows: Set up and solve both the investor's and the economy's game.

Alternative	Conditions #1	#2	#3
A	3000	4500	6000
B	1000	9000	2000
C	4500	4000	3500

Figure 3

Investor's Game:

V = Profit

x1 = Portion to place in A

x2 = Portion to place in B

1 - x1 - x2 = Portion to place in C

The linear programs are as follows:

To Maximize V

V <= 3000x1 + 1000x2 + 4500(1 - x1 - x2) Condition #1

V <= 4500x1 + 9000x2 + 4000(1 - x1 - x2) Condition #2

V <= 6000x1 + 2000x2 + 3500(1 - x1 - x2) Condition #3

x1; x2; (1 - x1 - x2) >= 0

x1; x2; (1 - x1 - x2) <= 1

V >= 0

```
library("lpSolve")
```

```
## Warning: package 'lpSolve' was built under R version 3.3.2
```

```
f.obj <- c(1, 1, 1)

row1 = c(-1500, -3500, -1)
row2 = c(500, 5000, -1)
row3 = c(2500, -1500, -1)
row4 = c(1, 0, 0)
row5 = c(0, 1, 0)
row6 = c(0, 0, 1)

f.con <- rbind(row1, row2, row3, row4, row5, row6)
f.dir <- c(">=", ">=", ">=", ">=", ">=", ">=")
f.rhs <- c(-4500, -4000, -3500, 0, 0, 0)

lp ("max", f.obj, f.con, f.dir, f.rhs)
```

```
## Success: the objective function is 4125.25
```

```
lp ("max", f.obj, f.con, f.dir, f.rhs)$solution
```

```
## [1]    0.25    0.00 4125.00
```

Based on the solution, we should allocate 25% of investment on alternative A, nothing on alternative B, and 75% on alternative C to obtain profit of 4125.

Nature's Game:

V = Profit

y1 = Portion to place in

y2 = Portion to place in B

1 - y1 - y2 = Portion to place in C

The linear programs are as follows:

To Minimize V

V <= 3000y1 + 4500y2 + 6000(1 - y1 - y2) Alternative A

V <= 1000y1 + 9000y2 + 2000(1 - y1 - y2) Alternative B

V <= 4500y1 + 4000y2 + 3500(1 - y1 - y2) Alternative C

y1; y2; (1 - y1 - y2) >= 0

y1; y2; (1 - y1 - y2) <= 1

V >= 0

```
f.obj <- c(1, 1, 1)

row1 = c(-3000, -1500, -1)
row2 = c(-1000, 7000, -1)
row3 = c(1000, 500, -1)
row4 = c(1, 0, 0)
row5 = c(0, 1, 0)
row6 = c(0, 0, 1)

f.con <- rbind(row1, row2, row3, row4, row5, row6)
f.dir <- c("<=", "<=", "<=", ">=", ">=", ">=")
f.rhs <- c(-6000, -2000, -3500, 0, 0, 0)

lp ("min", f.obj, f.con, f.dir, f.rhs)
```

```
## Success: the objective function is 4125.625
```

```
lp ("min", f.obj, f.con, f.dir, f.rhs)$solution
```

```
## [1]    0.625    0.000 4125.000
```

Therefore, the nature game shoud play 0.625 condition 1, no condition 2 and 0.375 condition 3.