**Appendix B:**

**Code for multiple forms using OMPR**

# Load packages and data ---------------------------------------------------

library(dplyr)

library(tidyr)

library(openxlsx)

library(ompr)

library(ompr.roi)

library(ROI.plugin.glpk)

library(readxl)

Bank\_Collection <- read\_excel("Demo data.xlsx", sheet = "Metadata")

enemies <- read\_excel("Demo data.xlsx", sheet = "Enemies")

enemies <- subset(enemies, select= -c(TYPE))

Bank\_Collection$Content <- substr(Bank\_Collection$Comp, 3, 5)

Bank\_Collection$Content[Bank\_Collection$Content=="1.1"] <- 1

Bank\_Collection$Content[Bank\_Collection$Content=="1.2"] <- 2

Bank\_Collection$Content[Bank\_Collection$Content=="1.3"] <- 3

Bank\_Collection$Content[Bank\_Collection$Content=="1.4"] <- 4

Bank\_Collection$Content[Bank\_Collection$Content=="1.5"] <- 5

Bank\_Collection$Content[Bank\_Collection$Content=="2.1"] <- 6

Bank\_Collection$Content[Bank\_Collection$Content=="2.2"] <- 7

Bank\_Collection$Content[Bank\_Collection$Content=="2.3"] <- 8

Bank\_Collection$Content <- as.numeric(Bank\_Collection$Content)

Bank\_Collection$B <- as.numeric(Bank\_Collection$IRTb)

OP\_items <- subset(Bank\_Collection, Status == "Scored Items (Sco)")

OP\_items$seq <- seq(1:nrow(OP\_items))

attach(OP\_items)

attach(enemies)

# Define Variables --------------------------------------------------------

n\_items <- nrow(OP\_items) #Number of items in pool (I)

n\_forms <- 10 #Number of test forms (F)

tif\_points <- 1 #Number of theta points at which information if controlled (J)

form\_length <- 100 #Length of both test forms (N)

n\_content <- c(10, 50, 11, 1, 6, 10, 5, 7) #Numbers of items required from k=1:8 content categories (Nc)

# Set TIF parameters ------------------------------------------------------

theta <- c(-0.33) #Define θ points at which information is controlled

Info <- array(0,c(n\_items,tif\_points)) #Define empty matrix with item information function values

#Fill matrix with item information function values (Rasch model)

for(j in 1:tif\_points){

P <-exp(theta[j]-B)/(1+exp(theta[j]-B))

Q <- 1-P

Info[,j] <- P\*Q

}

# Content Categories ------------------------------------------------------

Vc <- list()

for(k in 1:length(n\_content)){

Vc[[k]] <- c(1:n\_items)[Content==k]

}

# Enemy Items -------------------------------------------------------------

enemies$itemA.index<-match(enemies$ItemA, OP\_items$ID)

enemies$itemB.index<-match(enemies$ItemB, OP\_items$ID)

Ve <- list()

library(purrr)

Ve<-enemies[,3:4] %>% purrr::transpose()

Ve<-lapply(Ve, unlist, use.names=FALSE)

Ve<-unique(lapply(Ve, sort))

Ve<- Filter(function(x) length(x)>1, Ve)

# Begin constructing the MIP model ----------------------------------------

start\_time <- Sys.time()

# this is a slight difference between lpSolveAPI and ompr!!!!!!!!!!!!!!!!!!!!

q <- as.vector(Info[,1]) # makes the data.frame into a vector for the solver

model <- MIPModel() %>%

add\_variable(x[i,j], i = 1:n\_items, j=1:n\_forms, type = "binary") %>%

set\_objective(sum\_expr(q[i]\*x[i,j], i=1:n\_items, j=1:n\_forms), "max") %>%

add\_constraint(sum\_expr(x[i,j], i=1:n\_items) == form\_length, j=1:n\_forms) %>%

# content cats 1-8

add\_constraint(sum\_expr( x[i,j], i=Vc[[1]]) >= 10, j=1:n\_forms) %>%

add\_constraint(sum\_expr( x[i,j], i=Vc[[2]]) >= 50, j=1:n\_forms) %>%

add\_constraint(sum\_expr( x[i,j], i=Vc[[3]]) >= 11, j=1:n\_forms) %>%

add\_constraint(sum\_expr( x[i,j], i=Vc[[4]]) >= 1, j=1:n\_forms) %>%

add\_constraint(sum\_expr( x[i,j], i=Vc[[5]]) >= 6, j=1:n\_forms) %>%

add\_constraint(sum\_expr( x[i,j], i=Vc[[6]]) >= 10, j=1:n\_forms) %>%

add\_constraint(sum\_expr( x[i,j], i=Vc[[7]]) >= 5, j=1:n\_forms) %>%

add\_constraint(sum\_expr( x[i,j], i=Vc[[8]]) >= 7, j=1:n\_forms)

#item on only one form

for (k in 1:n\_items){

model<-model %>%

add\_constraint(sum\_expr(x[k,j], j=1:n\_forms) <= 1)

}

# enemy items

for(e in 1:length(Ve)){

model<- model %>%

add\_constraint(sum\_expr( x[i,j], i=Ve[[e]]) <= 1, j=1:n\_forms)

}

# SOLVE MODEL -------------------------------------------------------------

model <- model %>%

solve\_model(with\_ROI(solver = "glpk", verbose = TRUE))

end\_time <- Sys.time()

end\_time - start\_time

# Merge Solution with Item Collection -------------------------------------

matching <- model %>%

get\_solution(x[i,j]) %>%

filter(value ==1) %>%

select(i, j) %>%

rowwise() %>%

ungroup %>%

mutate(selected = 1)

p <-pivot\_wider(matching,

id\_cols = i,

names\_from = j,

values\_from = selected,

names\_sep = "\_",

names\_prefix = "Form")

zz <- merge(OP\_items, p, by.x="seq", by.y="i", all.x= TRUE)

zz$Form1[is.na(zz$Form1)] <- 0

zz$Form2[is.na(zz$Form2)] <- 0

zz$Form3[is.na(zz$Form3)] <- 0

zz$Form4[is.na(zz$Form4)] <- 0

zz$Form5[is.na(zz$Form5)] <- 0

zz$Form6[is.na(zz$Form6)] <- 0

zz$Form7[is.na(zz$Form7)] <- 0

zz$Form8[is.na(zz$Form8)] <- 0

zz$Form9[is.na(zz$Form9)] <- 0

zz$Form10[is.na(zz$Form10)] <- 0

# Plot TIF ----------------------------------------------------------------

x <- seq(-3,3, .01) #Define theta axis

ItemInfo <- array(0,c(n\_items, length(x))) #Define empty matrix with item information function values along theta axis for all items

for (j in 1:length(x)){

P <-exp(x[j]-B)/(1+exp(x[j]-B))

Q <- 1-P

ItemInfo[,j] <- P\*Q

} #Calculate item information function values along theta axis for all items

y <- matrix(0,length(x),10) #Define empty matrix with item information function values along theta axis for all items

for (j in 1:length(x)){

y[j,1] = sum(zz$Form1\*ItemInfo[,j])

y[j,2] = sum(zz$Form2\*ItemInfo[,j])

y[j,3] = sum(zz$Form3\*ItemInfo[,j])

y[j,4] = sum(zz$Form4\*ItemInfo[,j])

y[j,5] = sum(zz$Form5\*ItemInfo[,j])

y[j,6] = sum(zz$Form6\*ItemInfo[,j])

y[j,7] = sum(zz$Form7\*ItemInfo[,j])

y[j,8] = sum(zz$Form8\*ItemInfo[,j])

y[j,9] = sum(zz$Form9\*ItemInfo[,j])

y[j,10]= sum(zz$Form10\*ItemInfo[,j])

}

#Plot information functions

#jpeg("TIF.jpg")

plot(x,y[,1], type="l", lty=1, xlab='Theta',ylab='Information', col="gray1", ylim=c(0, 25))

lines(x,y[,2], type="l", lty=1, xlab='Theta',ylab='Information', col="gray2")

lines(x,y[,3], type="l", lty=1, xlab='Theta',ylab='Information', col="gray3")

lines(x,y[,4], type="l", lty=1, xlab='Theta',ylab='Information', col="gray4")

lines(x,y[,5], type="l", lty=1, xlab='Theta',ylab='Information', col="gray5")

lines(x,y[,6], type="l", lty=1, xlab='Theta',ylab='Information', col="gray6")

lines(x,y[,7], type="l", lty=1, xlab='Theta',ylab='Information', col="gray7")

lines(x,y[,8], type="l", lty=1, xlab='Theta',ylab='Information', col="gray8")

lines(x,y[,9], type="l", lty=1, xlab='Theta',ylab='Information', col="gray9")

lines(x,y[,10], type="l", lty=1, xlab='Theta',ylab='Information', col="gray10")

abline(v = theta, col="red", lwd=3, lty=2)

content\_table <- zz %>%

group\_by(Content) %>%

summarize(F1 = sum(Form1),

F2 = sum(Form2),

F3 = sum(Form3),

F4 = sum(Form4),

F5 = sum(Form5),

F6 = sum(Form6),

F7 = sum(Form7),

F8 = sum(Form8),

F9 = sum(Form9),

F10 = sum(Form10))

# checking to make sure that each item is assigned to only 1 form

zz$totals <- apply(zz[,12:21], 1, sum)