ETM640ProjectDraft

Randolph Anderson

March 4, 2019

Introduction

The 532d Training Squadron (532 TRS) is a unit in the United States Air Force (USAF). The 532 TRS trains the nation's intercontinental ballistic missile (ICBM) operators. The unit also trains maintenance forces for ICBM and air launched cruise missile (ALCM) systems. 532 TRS instructors receive professional development (ProD) to increase their knowledge of nuclear enterprise. Professional development involves sending one or more nstructors on a trip to a nuclear facility in the United States.While on their trip, instructors receive tours and briefings from onsite personnel. One purpose of providing professional development is to increase effectiveness of instructors. Instructors can help motivate students by illuminating the importance of what students are learning. However, instructors also perform jobs that do not involve actively teaching students. When in these other jobs instructors are not in the best position to integrate what they learned into their instruction.

Problem Definition

Some instructors do not receive professional development until most of their instructing time is over. As a result, they never get a chance to share what they have learned with the students. Since the goal of ProD is, in part, to increase the effectiveness of instructors, the unit may be able to do a better job assigning instructors to professional development trip in a way that balances several objectives (goals).

Literature Review

In this linear programming problem we are ensuring that we properly use funds in the 532d TRS. We are optimizing that all instructors get an opportunity to go to at least one professional development every year. This ensures better instruction and more rounded instructors. This is very similar to the endeavor of the Compstat used by the NYPD [1]. This showed that resources should be allocated to areas with more reported crimes [1]. Like this idea we are showing that we should allocate professional development money to areas where it would improve overall instruction. This linear programming model will ensure that we make the most of the resources in our organization. We are also focusing on goal programming. The GP technique was first used by Charnes and Cooper in 1960s. The Rubber Plantation example shows how the owner of the process will give the priority to the goals that are set in the project [2]. We are using weighted constraints that determine who is going on professional development trips based on the budget, improving instruction, minimum and maximum trip targets, and finally merit. We are going to be using data from the previous year so we are assuming that the budget will be the same for years in the future.

Defining Goals

The squadron has several goals for the professional development program. This project aims balance the following goals:

Goal 1: (Instructor Preference) Instructor preference should be strongly considered when assigning instructors to trips.

Goal 2. (Budget) Aim to spend exactly the budgeted funds for professional development activities

Goal 3. (Prioritize Active Instructors) Maxmize the number of Active instructors sent on Professional development trips

Goal 4. (Equitable opportunities) Each instructor should receive approximately equal number of professional development trips

Goal 5. (Merit) Top performing instructors deserving of merit should be matched to special trips (sometimes referred to as "glory trips") as a reward for outstanding performance.

Goal Programming Model

Goal programming is an optimization techniques to solve problems with multiple objectives. Additionally, Goal programming allows for flexibility in meeting constraints by allowing constraints to be deviated/broken in order to balance multiple goals. The model will determine which instructor to send on which ProD trip for each quarter (3 month period). Goal programming models can be formulated in many different ways. For example, some models might aim to minimized the sum of deviations from the goal targets, while other models, may be set up to minimize the largest deviation (minimax model). This project will use a version of the minimax model. A weighted percentage based minimax model is selected for the following reasons and assumptions. It is assumed that harm from deviations may increase rapidly such that it may not be advantage to allow one goal to float to far from target. For example, if objective 3 was allowed to deviate it could create harmful perceptions of unfairness. The percentage based minimax model was selected because harm of deviation was assumed to be more closely related to percentage deviation then nominal deviation. Weights were applied because squadron leadership has communicated that a loose prioritization between the goals. Weights will allow the model to capture these priorities into the results.

Mathematical Formulation

Assumptions:

The model was built under the following assumptions:

1. Instructors are available to go on any professional development trip they are assigned
2. All planned trips and number of open slots do not change througout time period under consideration
3. Active instructors pass valuable information to students following a professional development trip

Parameters

i : Instructor i $\in \{0, 1, 2, \ldots, N\}$
t : Trips t $\in \{0, 1, 2, \ldots, M\}$
g : Goal g $\in \{0, 1, 2, \ldots, G\}$
d : Deviation d $\in \{0, 1, 2, \ldots, D\}$
$P_{i,t}$: Preference of employee i for trip t
C_t : TDY cost per person
SA_i : Slots available
S_i : Active or inactive status of i-th instructor
IM_i : Instructor merit
TM_t : Trip merit status
TB : Travel budget
GT_g : Goal Target for the g-th goal
W_g^- : Weight for minus deviation the g-th goal
W_g^+ : Weight for postive deviation the g-th goal
$D_{d,g}^-$: d-th amount minus deviation for the g-th goal target
$D_{d,g}^+$: d-th amount postive deviation for the g-th goal target

Decision Variable There are i instructors and t trips. The model aims to determine which instructors to send on which trip. A decision is made to either send/not send a given instructor for each trip. $A[i,t]$ captures the decision for each instructor trip combination. $A[i,t]$ is binary (either 0 or 1). A 0 indicates that the i th instructor was not assigned to the t th trip. While a 1 indicates that the i th intructor is assigned to the t th trip. For example, A[1,2]=1 indicates the decision to send instructor 1 to Pro-D trip 2. While A[12,4] = 0 indicates that the 12th instructor is not assigned to 4 th trip.

$A_{i,t}$: Assignment of the i-th instructor to the t-th trip

Objective Weighted percent MiniMax objective function.

$$\text{Min: } Q$$
$$\text{subject to } \frac{w_g^- D_{d,g}^- + w_g^+ D_{d,g}^+}{GT_g} \leq Q \; \forall \; d, g$$

Weights

Once we are happy with the weights we should fill this out explaining the values choosen.

Goal Constraints **Updated 15 Mar 19**

The first goal is maximize the assignment of instructors to their preferred trip. This goal is implemented into the model through the following contraint. The target T_1 equals 10*n, where n is the number of instructors.

$$\sum_i \sum_t P_{i,t} A_{i,t} + D_{1,1}^- - D_{1,1}^+ = GT_1$$

The second goal is to use approxmately the full budget for professional development activies.

$$\sum_i \sum_t C_i A_{i,t} + D_{1,2}^- - D_{1,2}^+ = GT_2$$

The third goal is to maxmize the number of active instructors sent on Professional development trips

$$\sum_t S_i A_{i,t} + D_{i,3}^- - D_{i,3}^+ = S_i GT_3 \; \forall \; i$$

The fourth goal is for each instructor to receive approximately equal number of professional development trips.

$$\sum_t A_{i,t} + D_{i,4}^- - D_{i,4}^+ = GT_4 \; \forall \; i$$

The fifth goal is for each instructor with merit status to be matched with a “special trip” trip prioritize for instructors deserving merit.

$$\sum_t IM_i TM_t A_{i,t} + D_{i,5}^- - D_{i,5}^+ = GT_5 IM_i \; \forall \; i$$

Hard Constraints 1. The total cost must be less than or equal to the total budget. Total budget is a hard constraint the target budget is always less than or equal to the total budget.

$$\sum_i \sum_t C_i * A_{i,t} \leq TB$$

2. Each trip must have less than or equal to the max number of slots filled by instructors

$$\sum_i A_{i,t} \leq SA_t \; \forall \; t$$

```
suppressPackageStartupMessages(library (magrittr, quietly = TRUE))     #Used for pipes/dplyr
suppressPackageStartupMessages(library (pander, quietly = TRUE))       # for nicely formated tables
suppressPackageStartupMessages(library(dplyr, quietly=TRUE))           # For data structure manipulation
suppressPackageStartupMessages(library(ROI, quietly=TRUE))
suppressPackageStartupMessages(library(ROI.plugin.symphony, quietly=TRUE))  # R Optimization Interface package
suppressPackageStartupMessages(library(ROI.plugin.glpk, quietly=TRUE)) # Connection to glpk as solver
suppressPackageStartupMessages(library(ompr, quietly=TRUE))            # Optimization Modeling using R
suppressPackageStartupMessages(library(ompr.roi, quietly=TRUE))        # Connective tissue
suppressPackageStartupMessages(library(tidyr))
suppressPackageStartupMessages(library(tidyr))
```

Data Input

```
# reads in the instructor data from a csv file.
RawInstructorData <- as.matrix(read.table(file="InstructorData.csv", row.names=1,sep=","))
colnames(RawInstructorData) <- c("Instructor","Top Performer","Active/Inactive")
RawInstructorData
```

```
##    Instructor    Top Performer Active/Inactive
## 1  "Carswell"    "0"           "1"
## 2  "Bostic"      "0"           "1"
## 3  "Frank"       "0"           "1"
## 4  "Schell"      "0"           "1"
## 5  "McLarty"     "0"           "1"
## 6  "Johnson"     "1"           "1"
## 7  "Parsons"     "0"           "1"
## 8  "Ramie"       "0"           "1"
## 9  "McKenzie"    "0"           "0"
## 11 "Corrigan, B" "0"           "0"
## 10 "Corrigan, P" "0"           "1"
## 12 "Moselle"     "0"           "1"
## 13 "Payne"       "0"           "0"
## 15 "Boyd"        "0"           "0"
## 14 "Meno"        "1"           "1"
## 16 "Hudson"      "0"           "0"
## 17 "Moore"       "0"           "0"
## 18 "Hagstrom"    "0"           "0"
## 22 "Spratt"      "0"           "0"
## 19 "Temple"      "0"           "0"
## 20 "Westfall"    "0"           "0"
## 21 "Wyatt, B"    "0"           "1"
## 23 "Wyatt, J"    "0"           "1"
## 24 "Romanofski"  "1"           "1"
## 26 "Cook"        "1"           "1"
## 25 "Glass"       "0"           "0"
## 27 "Dalrymple"   "0"           "0"
## 28 "White, C"    "0"           "1"
## 30 "Chaney"      "0"           "0"
## 29 "Gunther"     "0"           "0"
## 33 "Anderson"    "1"           "1"
## 31 "Kuhls"       "0"           "0"
## 32 "Turner"      "0"           "0"
## 35 "Fox"         "0"           "0"
## 34 "Raines"      "0"           "0"
## 36 "Dean"        "1"           "0"
## 37 "Burnside, S" "0"           "0"
```

```
# reads in the instructor data from a csv file.
RawTripData <- as.matrix(read.table(file="C:/Users/rpatr/Documents/ProDTripData.csv", row.names=1,sep=","))
colnames(RawTripData) <- c("Trip Name","MeritBased","instrSlotsAvailable","Cost/Instr","TotalCost")
RawTripData
```

```
##    Trip Name                       MeritBased instrSlotsAvailable
## 1  "Los Alamos/Sandia National Lab" "0"        "4"
## 2  "Nuc 200"                       "1"        "4"
## 3  "Y-12"                          "0"        "6"
## 4  "NNSS"                          "0"        "4"
## 5  "PANTEX"                        "0"        "2"
## 6  "Bangor NAS"                    "0"        "3"
## 7  "NMCC"                          "1"        "2"
## 8  "LLNL"                          "0"        "5"
## 9  "AF WERX"                       "1"        "3"
## 10 "Kingsbay"                      "0"        "4"
## 11 "AFA Symposium"                 "1"        "1"
## 12 "Honeywell Tour"                "0"        "4"
## 13 "Nuclear Symposium"             "1"        "4"
##    Cost/Instr TotalCost
## 1  "1276.56"  " 5106.24"
## 2  "1282.94"  " 5131.76"
## 3  "1858.48"  "11150.88"
## 4  " 448.47"  " 1793.88"
## 5  "1267.52"  " 2535.04"
## 6  "1288.12"  " 3864.36"
## 7  "2310.44"  " 4620.88"
## 8  "1048.10"  " 5240.50"
## 9  "2555.28"  " 7665.84"
## 10 "1160.29"  " 4641.16"
## 11 "3318.60"  " 3318.60"
## 12 "1380.22"  " 5520.88"
## 13 "1841.13"  " 7364.52"
```

```
# reads in the instructor data from a csv file.
RawInstrPrefData <- as.matrix(read.table(file="C:/Users/rpatr/Documents/InstructorPreferenceData.csv", row.names=1,sep=","))
N <- length(RawInstrPrefData[,1])
M <- length(RawInstrPrefData[1,])
ColNames <- lapply(list(rep("T",M)),paste0,1:M) # Trip numbers: T1, T2, ...
RowNames <- lapply(list(rep("I",N)),paste0,1:N) # Instructor numbers: I1, I2, ...
dimnames(RawInstrPrefData) <- c(RowNames,ColNames)
RawInstrPrefData
```

```
##     T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13
## I1   0  0  0  0  0  0  0  0  5   0   6   0  10
## I2   0 10  0  8  6  5  0  0  0   0   0   0   0
## I3   0  0  5  0  0  0  0  8  0  10   0   6   0
## I4   0  0 10  0  9  0  5  0  6   0   0   0   0
## I5   0  0  0  0  0  5  0  0 10   0   6   0   0
## I6   5  0  5  0 10  0  0  0  0   0   0   0   0
## I7  10  0  0  0  0  9  0  0  0   5   0   0   0
## I8   0  5  0  0  0  0  0  0  0   0  10   0   6
## I9   0 10  0  8  0  9  0  0  0   0   0   0   0
## I10  0  0  0  0  0 10  0  6  0   8   0   0   0
## I11  0  0  0  9  0  5  0 10  0   0   0   0   0
## I12  0  0  0 10  6  5  8  0  0   0   0   0   0
## I13  0  0  0  0  0  0 10  0  0   5   0   7   0
## I14  9  7  0  0  0  0  0  0  0   0   0   8  10
## I15  0 10  0  0  0  0  5  0  0   0   0   0   0
## I16  0  0  5  0  0  0  0  0 10   0   5   0   0
## I17  0  0 10  0  0  5  0  0  9   5   0   0   0
## I18  0  0  0  0  0  0  0  0  0   0  10   9   5
## I19 10  0  0  0  5  0  5  0  0   0   0   0   0
## I20  0  0  0  0  0  6 10  8  0   0   0   0   0
## I21  0  0  6  8  0  0  0 10  0   0   0   0   0
## I22  0  0  0 10  0  0  0  0  0   5   0   6   0
## I23  5  0  6  0  0  6  0  0  0  10   0   0   0
## I24  0 10  0  0  0  0  6  0  8   0   0   0   0
## I25  0  0  6  0  0  5  0  0  0   0   0  10   5
## I26  9  5  0 10  0  0  0  0  0   0   0   0   0
## I27  0 10  0  0  0  5  0  7  0   0   0   0   0
## I28  0  0  6  0  0  0  0  0 10   0   0   0   7
## I29  5  0 10  0  0  0  9  0  0   0   0   0   0
## I30 10  0  0  9  0  0  0  7  0   0   6   0   0
## I31  0  0  0  0  0  0  0  0  0   0  10   9   8
## I32  0  0  6  0  0  8  0  0  0   0   0   0  10
## I33  0 10  0  0  8  5  0  0  0   0   0   0   0
## I34  0  0  0  0 10  0  0  7  0   5   0   0   0
## I35  8  0  6  0  0  0  0  0  0  10   0   0   0
## I36  0  0  0  0  0 10  0  0  6   0   0   0   0
## I37  5  6  0  0  7  0  0  0  0   0   0  10   0
```

Goal Targets

1. Goal 1 Target = 10*N, where N is the total number of instructors and 10 is the highest preference rating possible.
2. Goal 2 Target = 67954.50 dollars, changes year to year based on available funds
3. Goal 3 Target = Total Spots Available / Total number of active instructors. (send as many active instructors as possible)
4. Goal 4 Target = Total Spots Available / N (N = total number of instructors)
5. Goal 5 Target = Total Merit Spots Available / Total Number of instructors with merit status

$$
\begin{aligned}
i &: \text{Instructor i } \in \{0, 1, 2, \ldots, N\} \\
t &: \text{Trips t } \in \{0, 1, 2, \ldots, M\} \\
g &: \text{Goal g } \in \{0, 1, 2, \ldots, G\} \\
d &: \text{Deviation d } \in \{0, 1, 2, \ldots, D\} \\
P_{i,t} &: \text{Preference of employee i for trip t} \\
C_t &: \text{TDY cost per person} \\
SA_t &: \text{Slots available for trip t} \\
S_i &: \text{Active or inactive status of i-th instructor} \\
IM_i &: \text{Instructor merit} \\
TM_t &: \text{Trip merit status} \\
TB &: \text{Travel budget} \\
GT_g &: \text{Goal Target for the g-th goal} \\
W_g^- &: \text{Weight for minus deviation the g-th goal} \\
W_g^+ &: \text{Weight for postive deviation the g-th goal} \\
D_{d,g}^- &: \text{d-th amount minus deviation for the g-th goal target} \\
D_{d,g}^+ &: \text{d-th amount postive deviation for the g-th goal target}
\end{aligned}
$$

```
G = 5 # number of goals
TB <- 188000 #budget for all unit funded travel.

N = length(RawInstrPrefData[,1]) # number of instructors
M = length(RawInstrPrefData[1,]) # number of trips

P <- RawInstrPrefData # preference matrix P_{i,t}
C <- as.numeric(RawTripData[,4]) # Cost per person for a given trip t

S <- as.numeric(RawInstructorData[,3]) #Status of instructor (active =1, non-active =0)
TS <- sum(S) # Total number of individuals with active instructor Status (TS)

IM <-  as.numeric(RawInstructorData[,2]) #Instructor Merit Status
TIM <- sum(IM) # Total Instructors with Merit Status

TM <-  as.numeric(RawTripData[,2]) #Trip Merit Status
TTM <- sum(TM) # Total Trips with Merit Status

SA <- as.numeric(RawTripData[,3]) # Slots Available
TSA <- sum(SA) #Total Slots Available

#Goals
GT1 <- 10*TSA # where SA is the total number of slots available (SA) and 10 is the highest preference rating possible.
GT2 <- 67954.50 # Squadron ProD Target Budget changes year to year based on available funds
GT3 <- TSA/TS  #Total Slots Available / Total Active Instructors
GT4 <- TSA/N #Total Spots Available / N (N = total number of instructors)
GT5 <- TTM/TIM #Total Merit Spots Available / Total Number of instructors with merit status
GT <- rbind(GT1,GT2,GT3,GT4,GT5) # Goal Target for the g-th goal

#Weights (see section on weights for justification)
Wm <- as.matrix(c(1,1,1,2,3)) # w_g^-
Wp <- as.matrix(c(0,1,0,1,1)) # w_g^+
```

Goal Programming Code *Updated 16 Mar 19***

```
model <- MIPModel() %>%

  # VARIABLES
  add_variable (Q, type = "continuous", lb = 0) %>%
  add_variable (A[i, t], i=1:N, t=1:M, type = "binary") %>%
  add_variable (Dm[d,g], d=1:N, g=1:G, type = "continuous", lb = 0) %>%
  add_variable (Dp[d,g], d=1:N, g=1:G, type = "continuous", lb = 0) %>%

  #OBJECTIVE
  set_objective(Q, "min")%>%

  #GOAL CONSTRAINTS
  #Goal 1 (Instructor Preference)
  add_constraint(sum_expr(P[i,t]*A[i,t],i=1:N, t=1:M) + Dm[1,1]- Dp[1,1] == GT[1])%>%

  #Goal 2 (Budget)
  add_constraint(sum_expr(C[t]*A[i,t],i=1:N, t=1:M) + Dm[1,2] - Dp[1,2] == GT[2])%>%

  #Goal 3 (Active Instructors)
  add_constraint(sum_expr(S[i]*A[i,t],t=1:M) + Dm[i,3] - Dp[i,3] == S[i]*GT[3],i=1:N)%>%

  #Goal 4 (Equitable)
  add_constraint(sum_expr(A[i,t],t=1:M) + Dm[i,4] - Dp[i,4] == GT[4],i=1:N)%>%

  #Goal 5 (Merit)
  add_constraint(sum_expr(IM[i]*TM[t]*A[i,t],t=1:M) + Dm[i,5] - Dp[i,5] == IM[i]*GT[5],i=1:N)%>%


  #HARD CONSTRAINTS
  # 1. Total Travel Budget Constraint
  add_constraint(sum_expr(C[t]*A[i,t],i=1:N,t=1:M) <= TB)%>%

  # 2. Slots Open for Trip Constraint
  add_constraint(sum_expr(A[i,t], i=1:N) <= SA[t], t=1:M)%>%


  #OBJECTIVE FUNCTION CONTRAINTS
  add_constraint((Wm[g]*Dm[d,g]+Wp[g]*Dp[d,g])/GT[g] <= Q, g=1:G,d=1:N)

result <- solve_model(model, with_ROI(solver = "symphony",gap_limit = 32.39, verbosity =1))
```

RESULTS

```
# Assignment Result (AR) of each i-th instructor to t-th trips AR[i,t].
AR <- result %>%
  get_solution(A[i,t]) %>%
  filter(value == 1) %>%
  select(i, t)

DmRaw <- result %>%
  get_solution(Dm[d,g]) %>%
  select(d, g, value)
DpRaw <- result %>%
  get_solution(Dp[d,g]) %>%
  select(d, g, value)

DmRawF <- result %>%
  get_solution(Dm[d,g]) %>%
  filter(value > 0) %>%
  select(d, g, value)

DpRawF <- result %>%
  get_solution(Dp[d,g]) %>%
  filter(value > 0) %>%
  select(d, g, value)

DmR <- matrix(rep(0, each=N*G), ncol=G)
DpR <- matrix(rep(0, each=N*G), ncol=G)

for(k in 1:nrow(DmRaw))
{
  DmR[DmRaw[k,1],DmRaw[k,2]]=DmRaw[k,3]
  DpR[DpRaw[k,1],DpRaw[k,2]]=DpRaw[k,3]
}


#TRIP SUMMARY CHART CREATION
# Text name of the t-th trip
TripNames <- RawTripData[,1]

# Total Instructors Assigned (TIA) to the t-th trip
TIApT <- tabulate(as.numeric(AR[,2]),nbins = M)

# Total Cost Per Trip
TCpT <- C*TIApT

# Total Num of top performers going on the t-th trip
TIMpT <- matrix(rep(0, each=M), ncol=1)

# Manual counting of all the top performers for each trip.
for(k in 1:nrow(AR))
{
  if(IM[AR[k,1]]>=1)
  {
    TIMpT[AR[k,2]] = TIMpT[AR[k,2]] + 1
  }
}
TripSummaryChart <- cbind(TripNames,TIApT,SA,TIMpT,TM,C,TCpT)
colnames(TripSummaryChart)<-list("Names", "TIApT", "SA","TIMpT", "TM", "C","TC")
#END OF TRIP SUMMARY CHART CREATION



#INSTRUCTOR SUMMARY CHART CREATION
#Instructor ID 1... N
InstrID <- matrix(c(1:N),ncol=1)

# Total Merit per Instructor (TMTpI)
TMTpI <- matrix(rep(0, each=N), ncol=1)

# Manual counting of merit trips per instructor
for(k in 1:nrow(AR))
{
  if(TM[AR[k,2]]>=1)
  {
    TMTpI[AR[k,1]] = TMTpI[AR[k,1]] + 1
  }
}
# Total Trips per Instructor (TMTpI)
TTpI <- matrix(rep(0, each=N), ncol=1)

# Manual counting of merit trips per instructor
for(k in 1:nrow(AR))
{
```

```
    TTpI[AR[k,1]] = TTpI[AR[k,1]] + 1
}

InstructorChart <- cbind(InstrID,IM,TMTpI,S,TTpI)
colnames(InstructorChart) <- list("InstrID","IM","TMTPI","S","TTPI")
#END OF INSTRUCTOR SUMMARY CHART CREATION


#GOAL PERFORMANCE SUMMARY CHART
# Find max Q for each goal
MQ <- matrix(rep(0, each=G), ncol=1)
MQ[1] = (Wm[1]*DmR[1,1]+Wp[1]*DpR[1,1])/GT[1]
MQ[2] = (Wm[2]*DmR[1,2]+Wp[2]*DpR[1,2])/GT[2]

 for(k in 1:nrow(DmR))
{
    if(MQ[3] < (Wm[3]*DmR[k,3]+Wp[3]*DpR[k,3])/GT[3])
    {
      MQ[3] =(Wm[3]*DmR[k,3]+Wp[3]*DpR[k,3])/GT[3]
    }

    if(MQ[4] < (Wm[4]*DmR[k,4]+Wp[4]*DpR[k,4])/GT[4])
    {
      MQ[4] =(Wm[4]*DmR[k,4]+Wp[4]*DpR[k,4])/GT[4]
    }

    if(MQ[5] < (Wm[5]*DmR[k,5]+Wp[5]*DpR[k,5])/GT[5])
    {
      MQ[5] =(Wm[5]*DmR[k,5]+Wp[5]*DpR[k,5])/GT[5]
    }
}

#Non-Zero Deivations minus/postive (NZDms/NZDps)
NZDms <- as.matrix(tabulate(DmRawF[,2], nbins=G))
NZDps <- as.matrix(tabulate(DpRawF[,2], nbins=G))

#Sum of all Deviations minus/postive for the g-th goal.
SumDms <- matrix(rep(0, each=G), ncol=1)
SumDps <- matrix(rep(0, each=G), ncol=1)
 for(k in 1:G)
{
 SumDms[k]= sum(DmR[,k])
 SumDps[k]= sum(DpR[,k])
}

#Average deviation amount for the g-th goal
AvgDm <- matrix(rep(0, each=G), ncol=1)
AvgDp <- matrix(rep(0, each=G), ncol=1)
 for(k in 1:G)
{
 AvgDm[k] = SumDms[k]/NZDms[k]
 AvgDp[k] = SumDps[k]/NZDps[k]
}

GoalChart <- cbind(GT,MQ,NZDms,SumDms,AvgDm, NZDps, SumDps, AvgDp)
colnames(GoalChart)<-list("Target","maxQ","NZDms","SumDms","AvgDm","NZDps","SumDps","AvgDp")
rownames(GoalChart)<-list("G1","G2","G3","G4","G5")
#END OF GOAL CHART CREATION**



#WEIGHTS CHART CREATION**
WeightsByGoal <- t(cbind(Wm,Wp))
colnames(WeightsByGoal)<-list("G1:InstrPref","G2:Budget","G3:ActiveInstr","G4:Equitable", "G5:Merit")
rownames(WeightsByGoal)<-list("Wm","Wp")
#END WEIGHTS CHART CREATION**




#Assignments and Preferences

#assignment Result (AR) with preference (ARwPref) of instructor for trip
ARwPref <- matrix(rep(0, each=3*nrow(AR)), ncol=3)
InstrPrefDataWithAssignments <- RawInstrPrefData
for(k in 1:nrow(AR))
{
   ARwPref[k,1] = AR[k,1]
   ARwPref[k,2] = AR[k,2]
   ARwPref[k,3] = RawInstrPrefData[AR[k,1],AR[k,2]]
   InstrPrefDataWithAssignments[AR[k,1],AR[k,2]] = paste0(InstrPrefDataWithAssignments[AR[k,1],AR[k,2]], "A")
```

```
}
ARwPref <- cbind(ARwPref[,2],ARwPref[,1], ARwPref[,3])
colnames(ARwPref) <- list("Trip","Assigned Instructor"," Instructor Preference")

# End of Assignments and Preferences




#display the formated results
print(paste0("Objective value= ", objective_value(result)))
```

```
## [1] "Objective value= 0.630434782608696"
```

```
#
 pander(WeightsByGoal,
        caption="Weights")
```

Weights

	G1:InstrPref	G2:Budget	G3:ActiveInstr	G4:Equitable	G5:Merit
Wm	1	1	1	2	3
Wp	0	1	0	1	1

```
#
 pander(GoalChart,
        caption="Goal Performance")
```

Goal Performance

	Target	maxQ	NZDms	SumDms	AvgDm	NZDps	SumDps	AvgDp
G1	460	0.6304	1	290	290	1	75	75
G2	67955	5.886e-07	0	0	NA	1	0.04	0.04
G3	2.706	0.6304	17	20	1.176	0	0	NA
G4	1.243	0.6087	28	6.811	0.2432	9	6.811	0.7568
G5	0.8333	0.2	0	0	NA	6	1	0.1667

```
pander(TripSummaryChart ,
        caption="Trip Data")
```

Trip Data

Names	TIApT	SA	TIMpT	TM	C	TC
Los Alamos/Sandia National Lab	4	4	0	0	1276.56	5106.24
Nuc 200	4	4	2	1	1282.94	5131.76
Y-12	6	6	1	0	1858.48	11150.88
NNSS	4	4	0	0	448.47	1793.88
PANTEX	2	2	0	0	1267.52	2535.04
Bangor NAS	3	3	0	0	1288.12	3864.36
NMCC	2	2	1	1	2310.44	4620.88
LLNL	5	5	0	0	1048.1	5240.5
AF WERX	3	3	1	1	2555.28	7665.84
Kingsbay	4	4	0	0	1160.29	4641.16
AFA Symposium	1	1	0	1	3318.6	3318.6
Honeywell Tour	4	4	0	0	1380.22	5520.88
Nuclear Symposium	4	4	2	1	1841.13	7364.52

```
#
pander(InstructorChart,
       caption="Instructor Data")
```

Instructor Data

InstrID	IM	TMTPI	S	TTPI
1	0	0	1	1
2	0	1	1	2
3	0	1	1	1
4	0	0	1	2
5	0	1	1	1
6	1	1	1	1
7	0	0	1	2
8	0	0	1	2
9	0	0	0	1
10	0	0	0	1
11	0	0	1	1
12	0	1	1	2
13	0	0	0	1
14	0	1	0	1
15	1	1	1	2
16	0	1	0	1
17	0	0	0	1
18	0	0	0	1
19	0	0	0	1
20	0	0	0	1
21	0	0	0	1
22	0	0	1	2
23	0	0	1	2
24	1	1	1	1
25	1	1	1	1
26	0	0	0	1
27	0	0	0	1
28	0	1	1	2
29	0	0	0	1
30	0	0	0	1
31	1	1	1	1
32	0	0	0	1
33	0	1	0	1
34	0	0	0	1
35	0	0	0	1
36	1	1	0	1
37	0	0	0	1

```
pander(InstrPrefDataWithAssignments,
       caption="Preference Chart with Instructor Assignments (IA)")
```

Preference Chart with Instructor Assignments (IA)

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
I1	0	0	0A	0	0	0	0	0	5	0	6	0	10
I2	0A	10A	0	8	6	5	0	0	0	0	0	0	0
I3	0	0	5	0	0	0	0	8	0	10	0A	6	0
I4	0	0	10A	0	9A	0	5	0	6	0	0	0	0
I5	0	0	0	0	0	5	0A	0	10	0	6	0	0

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
I6	5	0A	5	0	10	0	0	0	0	0	0	0	0
I7	10A	0	0	0	0	9A	0	0	0	5	0	0	0
I8	0A	5	0A	0	0	0	0	0	0	0	10	0	6
I9	0	10	0	8A	0	9	0	0	0	0	0	0	0
I10	0	0	0	0	0	10	0	6	0	8A	0	0	0
I11	0	0	0	9	0	5	0	10A	0	0	0	0	0
I12	0	0	0	10	6	5	8	0	0A	0	0	0A	0
I13	0	0	0	0	0	0	10	0	0	5	0	7A	0
I14	9	7	0	0	0	0	0	0	0	0	0	8	10A
I15	0	10	0A	0	0	0	5	0	0A	0	0	0	0
I16	0	0	5	0	0	0	0	0	10A	0	5	0	0
I17	0	0	10A	0	0	5	0	0	9	5	0	0	0
I18	0	0	0	0	0	0	0	0	0	0	10	9A	5
I19	10	0	0	0A	5	0	5	0	0	0	0	0	0
I20	0	0	0	0	0	6	10	8A	0	0	0	0	0
I21	0	0	6	8	0	0	0	10A	0	0	0	0	0
I22	0A	0	0	10	0	0	0	0A	0	5	0	6	0
I23	5	0	6	0	0	6A	0	0	0	10A	0	0	0
I24	0	10	0	0	0	0	6	0	8	0	0	0	0A
I25	0	0	6	0	0	5	0A	0	0	0	0	10	5
I26	9	5	0	10A	0	0	0	0	0	0	0	0	0
I27	0	10	0	0	0	5	0	7A	0	0	0	0	0
I28	0	0	6	0	0	0	0	0	10	0A	0	0	7A
I29	5	0	10A	0	0	0	9	0	0	0	0	0	0
I30	10	0	0	9A	0	0	0	7	0	0	6	0	0
I31	0	0A	0	0	0	0	0	0	0	0	10	9	8
I32	0	0	6	0	0	8A	0	0	0	0	0	0	10
I33	0	10A	0	0	8	5	0	0	0	0	0	0	0
I34	0	0	0	0	10A	0	0	7	0	5	0	0	0
I35	8	0	6	0	0	0	0	0	0	10A	0	0	0
I36	0	0	0	0	0	10	0	0	6	0	0	0	0A
I37	5	6	0	0	7	0	0	0	0	0	0	10A	0

```
pander(ARwPref,
       caption="Assignments and Preferences")
```

Assignments and Preferences

Trip	Assigned Instructor	Instructor Preference
1	2	0
1	7	10
1	8	0
1	22	0
2	2	10
2	6	0
2	31	0
2	33	10
3	1	0
3	4	10
3	8	0

Trip	Assigned Instructor	Instructor Preference
3	15	0
3	17	10
3	29	10
4	9	8
4	19	0
4	26	10
4	30	9
5	4	9
5	34	10
6	7	9
6	23	6
6	32	8
7	5	0
7	25	0
8	11	10
8	20	8
8	21	10
8	22	0
8	27	7
9	12	0
9	15	0
9	16	10
10	10	8
10	23	10
10	28	0
10	35	10
11	3	0
12	12	0
12	13	7
12	18	9
12	37	10
13	14	10
13	24	0
13	28	7
13	36	0

REFERENCES

[1] COMPSTAT: Its Origins, Evolution, and Future in Law Enforcement, Bureau of Justice Assistance Police Executive Research Forum, Washington D.C., 2013 https://www.bja.gov/Publications/PERF-Compstat.pdf (https://www.bja.gov/Publications/PERF-Compstat.pdf)

[2] A Goal Programming Approach to Rubber Plantation Planning in Tripura, Department of Mathematics Assam University, Silchar, India, 2012 http://www.m-hikari.com/ams/ams-2012/ams-121-124-2012/nandiAMS121-124-2012.pdf (http://www.m-hikari.com/ams/ams-2012/ams-121-124-2012/nandiAMS121-124-2012.pdf)