Winter 2020

Project Management Assignment Tool Using R and Shiny

Ben Stewart
Portland State University

Hoseok Jung
Portland State University

Moses Rawar
Portland State University

Citation Details

Group Project Report

Project Management Assignment Tool
Using R and Shiny

Course Title   : Operation Research

Course number   : ETM 540

Instructor   : Dr. Tim Anderson

Term   : Winter 2020

Team   : Ben Stewart, Hoseok Jung, Moses Rawar

Table of Contents

Abstract

Linear programs such as the R markdown language play an important role in helping to find an optimized solution in various fields of society. This area extends from the optimal distribution channels of coffee shops, which are closely related to our lives, to very important areas such as the deployment of military forces. Through this paper, we will try to find ways to maximize the efficiency and performance of a company by properly allocating project managers suitable for the projects performed by each company according to their capabilities and the requirements of the projects.

Introduction

Organizations need to have the right way to assign projects to project managers to be able to get the best result and to maximize the project. The organization needs to assign a project to the right project manager according to its ability and capacity in doing projects. Our project is about developing a decision support model for the project manager Assignment using an R Markdown and a Shiny library. The reference material is from "A Decision Support Model for Project Management Assignments" (by Peerasit Patanakul, Dragan Milosevic and Timothy R. Anderson). With this tool, an individual or an organization could easily assign project managers in a proper way of assigning the project. And we have used a method that can input each necessary input value through a library called the Shiny in a more convenient and user-friendly way. The method of implementing Shiny requires additional work, which may be somewhat complicated, but it is a very convenient tool because the user can intuitively assign and upload numbers of data through Excel.

Literature Review

In the Reference literature, in many organizations, the task of assigning a project to a project manager as a project is often seen by management as a major challenge. It must be done appropriately since assigning the right project manager to the project is one of the critical success factors in project management. Despite the fact that project allocation is very important in terms of company efficiency or performance, and that inappropriate allocation can reduce project success rates and negatively impact organizational performance, research in this area has been fairly limited, so this study has begun to resolve it. The company needs to understand the strategic importance of the project, and prioritize the projects considering the elements of the organizational

strategy. In order to allocate projects, it is necessary to judge whether the project manager's competency and technical competencies are consistent with the project requirements. The organization's strategic characteristics and project manager's resource capacities used in this document may not apply equally to all companies, so this approach may not apply to many other organizations' project assignments.

In this literature, we have assigned several projects to manage. The study also has begun by exploring the best way to assign multiple project managers to this project. All 12 projects and 6 project managers are applied to the study. Of these, five existing projects and one new project have been properly reassigned or newly assigned to these six project managers. How the organization utilizes the allocated resources to achieve maximum efficiency within given constraints has been studied. Four types of constraints are used in this study, and details will be discussed below.

Project Objective and Constraints

A. The objective function :

The objective function of the model is developed by considering project priorities and levels of matching between project managers and projects.

$$Max \sum_{i \in \{A,B,...F\}} \sum_{j=1}^{6} \sum_{k=1}^{4} (\alpha_{ijk} \ G_k \ V_{jk} \ W_{ij} \ X_{ij})$$

✔ $\alpha_{i,j,k}$: Coefficient of effectiveness of PM i in managing project j's contribution to goal k
✔ $G_k \ and \ V_{j,k}$: Strategic importance of the projects in the assignment
✔ $W_{i,j}$: Correspondence level between PM i and project j
✔ $X_{i,j}$: Binary decision variables. If project j should be assigned to project manager i, decision variables would be equal to 1.

This objective function is expressed as follows in R markdown.

```
set_objective (sum_expr(a[i,j,k] * g[k] * v[j,k] * w[i,j] * x[i,j],
                    i=1:NPM, j=1:NProject, k=1:Goal ), "max") %>%
```

Also, the decision variables of this project are implemented as follows. All of them are binary decision variables, and in the case of variables other than X, these are rather secondary variables to solve the constraints.

```
projectmodel <- MIPModel() %>%
  add_variable (x[i, j], i=1:NPM, j=1:NProject,
                type="binary") %>%
  add_variable (r[i], i=1:NPM,
                type="binary") %>%
  add_variable (z[i], i=1:NPM,
                type="binary") %>%
```

The objective of this project is to obtain proper value of $X_{i,j}$, binary variables. That means, project manager i is assigned to the appropriate project j, to get maximum efficiency. And here, numerical values for calculating the importance of a project using a somewhat complex data such as coefficient of effectiveness, the strategic importance of the projects, correspondence level, etc., and assigning an appropriate project manager are used for the objective function.

The coefficient of effectiveness of project manager i in managing project j's contribution to goal k is valued between 0 and 1 when there is any discontinuity in the management of project j. And this value of 0 means a complete failure in discontinuity in the management, and a value of 1 tells the full effectiveness of the project manager.

α_{ijk}	P1				P2				P3			
	G1	G2	G3	G4	G1	G2	G3	G4	G1	G2	G3	G4
PM A	1	1	1	1	1	0.9	1	0.8	1	0.8	0.95	1
PM B	0.9	0.9	1	0.7	1	1	1	1	1	0.8	0.95	1
PM C	0.9	0.9	1	0.7	1	0.9	1	0.8	1	1	1	1
PM D	0.9	0.9	1	0.7	1	0.9	1	0.8	1	0.8	0.95	1
PM E	0.9	0.9	1	0.7	1	0.9	1	0.8	1	0.8	0.95	1
PM F	0.9	0.9	1	0.7	1	0.9	1	0.8	1	0.8	0.95	1
α_{ijk}	P4				P5				P6			
	G1	G2	G3	G4	G1	G2	G3	G4	G1	G2	G3	G4
PM A	0.6	1	1	0.8	1	1	1	1	1	1	1	1
PM B	0.6	1	1	0.8	1	1	1	1	1	1	1	1
PM C	0.6	1	1	0.8	1	1	1	1	1	1	1	1
PM D	1	1	1	1	1	1	1	1	1	1	1	1
PM E	0.6	1	1	0.8	1	1	1	1	1	1	1	1
PM F	0.6	1	1	0.8	1	1	1	1	1	1	1	1

<Coefficient of effectiveness>

To define the strategic importance and prioritize the projects, relative contribution of organizational goal k to the mission(G_k) and relative contribution of project to the goal($V_{j,k}$) are represented as follows. In addition, for the organization, four organizational goals are set with the same relative contribution of 25% each. Also, the contribution of the project to each goal is set differently. For example, the contribution of each project to the goal 1(to create new markets) is set as the highest with Project 1 being 36.51 and the lowest with project 5(3.08). This contribution of goals and relative contribution can be used by changing the set value according to the environment of a company.

G1: To create new markets	G2: To create business alliances	G3: To improve customer satisfaction	G4: To improve technological innovation

Project 1	Project 2	Project 3	Project 4	Project 5	Project 6

	Goal 1	Goal 2	Goal 3	Goal 4
Percent relative contribution of goals to the mission (G_k)	25	25	25	25

Percent relative contribution of projects to the goal (V_{jk})	Goal 1	Goal 2	Goal 3	Goal 4
P1	36.51	16.67	16.67	21.60
P2	9.13	16.67	16.67	17.58
P3	5.51	16.67	16.67	9.20
P4	22.00	16.67	16.67	21.57
P5	3.08	16.67	16.67	17.28
P6	23.78	16.67	16.67	12.77

< Strategic importance of the projects in the assignment >

The correspondence level(W_{ij}) between project manager i and project j or matching score based on project requirements and competencies of project managers is set as the following table. The competencies are made based on the groups of technical, administrative/process, intrapersonal, interpersonal, and business/strategic competencies. In the case of Project manager C, all the necessary requirements of all projects can be satisfied, so a score of level 100 is gotten for all projects. On the contrary, project manager F seems to have to allocate relatively insignificant projects because it lacks the most competencies.

W_{ij}	P1	P2	P3	P4	P5	P6
PM A	59.7	63.4	63.9	54.6	76.1	71.9
PM B	97.8	97.8	97.8	95.6	97.8	97.8
PM C	100.0	100.0	100.0	100.0	100.0	100.0
PM D	99.3	99.3	99.3	99.3	99.3	99.3
PM E	80.3	87.2	84.5	77.4	96.5	95.7
PM F	47.2	57.1	54.3	42.2	77.8	65.5

< Correspondence level >

B. Constraints – Resource Availability

The four kinds of constraints are formulated from the criteria regarding organizational and personal limitations. The major constraints used in this project are "resource availability", "fixed assignment", "project assignment constraint" and "project manager assignment constraint".

a. Resource availability constraint

The constraint is used to ensure that each project manager will not be assigned to have excessive workload. So each project manager cannot perform work beyond his or her resource availability, and time demand of project and switchover-time loss are used for the work hours. The working hours will vary depending on each project manager and the number of projects in charge.

$$\sum_{j=1}^{6} D_{i,j}X_{i,j} + S_i \leq A_i \quad \forall\ i$$

✔ $D_{i,j}$: Time demand of project j for project manager i
✔ A_i : Resource availability of project manager i
✔ S_i : Switchover-time loss of project manager i

Each time demand is tabulated as below, and the required values on the table are entered through Shiny. And the time loss that occurs when executing several projects at the same time is calculated by the formula. We implemented this constraint with the R markdown command as shown below. The method of calculating the switchover-time loss is rather complicated, so the constraint is relatively long and not easy to implement.

```
add_constraint (sum_expr(d[i,j] * x[i,j], j=1:NProject)
              + 1.5 * ((sum_expr(x[i,j], j=1:NProject) +  n[i]) + r[i] - 1)
              + 4.5 * z[i]
              <= A[i], i=1:NPM) %>%
    #Constraint 1) Resource availability constraints - 1

add_constraint (1 - 0.25 * (sum_expr(x[i,j], j=1:NProject) +  n[i])
              >= r[i], i=1:NPM) %>%
    #Constraint 1) Resource availability constraints - 2

add_constraint (((sum_expr(x[i,j], j=1:NProject) +  n[i]) + r[i] - 1) / 4
              <= z[i], i=1:NPM) %>%
    #Constraint 1) Resource availability constraints - 3

add_constraint (((sum_expr(x[i,j], j=1:NProject) +  n[i]) + r[i] - 1)
              >= z[i], i=1:NPM) %>%
    #Constraint 1) Resource availability constraints - 4
```

It can be seen that the project time demand varies depending on the capabilities of each project manager. In the case of project manager C, as mentioned earlier, the project execution time is relatively short because of his or her high capability. Also, in the case of project manager F, it can be seen that the time demand for execution is longer than other managers for all projects.

D_{ij}^{*}	P1	P2	P3	P4	P5	P6
PM A	276	240	208	170	208	156
PM B	240	216	184	136	184	120
PM C	240	216	184	136	184	120
PM D	240	216	184	136	184	120
PM E	276	240	208	170	208	156
PM F	420	324	276	234	249	216

* Time demand of project j from project manager i
(person-hours per 12 weeks)

< **Time demand of projects for each project manager** >

In our project, the existing workload and available resource availability for each 12-week period are given as follows. Each manager can be assigned additional projects within the available time.

	PM A	PM B	PM C	PM D	PM E	PM F
Regular work hours*	720	720	720	720	720	720
Percent of non-project work	20%	20%	20%	20%	20%	20%
Effective capacity* (E_i)	576	576	576	576	576	576
Existing workload* (L_i) after taking off reassignments	204	180	240	240	205	216
Resource availability* (A_i)	372	396	336	336	371	360

* Unit: Person-hours per planning horizon (12 weeks)

< **Resource availability of project managers** >

9

b. Special requirement constraint

For some projects, certain project managers may need to be assigned to meet stakeholder needs or other requirements.

$$\sum_{i \,\in\, \{A,B,...F\}} SC_{ij} X_{ij} = 1 \qquad \forall \quad j$$

If project manager i has special competencies to respond to specific project j, $SC_{i,j}$ can be numbered as 1 and this project can be assigned to it. And this condition can be easily implemented in the program as below.

```
add_constraint (sum_expr(sc[i] * x[i,1], i=1:NPM)
                == 1) %>%
```

c. Project assignment constraint

All projects must be assigned to the manager, and it can be assigned to only one project manager.

$$\sum_{i \,\in\, \{A,B,...F\}} X_{i,j} = 1 \qquad \forall \quad j$$

We have expressed this constraint in R code.

```
add_constraint (sum_expr(x[i,j], i=1:NPM)
                == 1, j=1:NProject) %>%
```

10

d. Project manager assignment constraint

On the contrary with the above constraint, all project managers need to take a project, and each manager should take only one project.

$$\sum_{j=i}^{m} X_{i,j} = 1 \qquad \forall \quad i$$

In R code, this constraint is implemented as follows.

```
add_constraint (sum_expr(x[i,j], j=1:NProject)
                == 1, i=1:NPM)
```

Using Shiny

The decision to implement Shiny was made in the effort to create a tool that can be useful to those without access or experience with coding in R. Through the use of Shiny, we are able to wrap the core R coding in a wrapper that allows for a web-based tool that intuitively allows comma separated value (.csv) files to be loaded without any visibility of the underlying code working to produce results. The comma separated value files can be easily created using Excel or similar spreadsheet programs. Additionally, Shiny allows for a clean user interface that displays the loaded data files as well as the solution output from the core r code.

Critically, the core R code was coded without any values hardcoded. Instead, the code required variables and matrices to be loaded from an external source. This allowed the Shiny inputs gathered from a user (numbers and matrices) to be passed along to the inputs of the core R code. The core R code also adapted to different data set sizes (i.e. the number of projects, the number of project managers and the number of company goals), allowing unique data to be input without adjustments needed within the R coding.

The user interface used for the Shiny interface, as seen in Appendix A, consisted of a side panel on the left for the user to input their numbers and datasets in .csv format and a main panel that either shows what data has been loaded via the side panel or the solution created by the tool, depending on whether 'Input Data' or 'Show Solution' is selected. The 'conditionalPanel' coding allowed the Main Window to display different outputs depending on user selection.

```
# Condition Panel for Solution

conditionalPanel(
        condition = "input.Selection == 2",
        h2("Solution From Data:", align = "center"),
        br (),
        h4("Objective Value:", textOutput("SolutionObjective")),
        br (),
        h4("Project Manager Assignments :", tableOutput("SolutionMatrix"))
    )
```

<Conditional Panel Code, Selecting Output for Main Window>

Having the ability to view the input data allowed a verification step where the user could confirm that the correct .csv files had been loaded and matched expectations. When 'Show Solution' is selected, the recommendation for the matching up of Project Managers with the most appropriate Projects is displayed in an easy to read format as seen in the below image.

A Decision Support Model for Project Manager Assignments

Select Show Solution After Data Entry

Show Solution

# of Projects

6

# of Project Managers

6

# of Goals

4

g matrix csv

Browse...   Data For OR Team

Solution From Data:

Objective Value:
[1] 8786.419

Project Manager Assignments:

PM	Project
PM_C	Project_1
PM_B	Project_2
PM_A	Project_3
PM_D	Project_4
PM_F	Project_5
PM_E	Project_6

<Output Panel from Shiny>

12

Conclusions

As society develops, the success of projects undertaken by each organization is very important. Research has shown that projects undertaken by organizations that underestimate project management have been failing more than 67%. And these failures are often initiated by project managers. Therefore, when conducting a project, the competencies of each project manager should be closely reviewed in various aspects such as technical, administrative, intrapersonal, and interpersonal. In addition, appropriate project managers should be assigned according to the importance and requirements of each project. Of course, it is important to continually manage and monitor performance even after these allocations.

Through this group project, we have studied the assignment to each project of the project manager that can be used by each company. As a result, through the R markdown we learned in the class this semester and our newly approached Shiny tool, we are able to successfully obtain the same results as those analyzed in the reference papers. And we also go further and develop a more general model to use the tool in other companies and organizations. Also Shiny provides a platform that users can access and use more easily through the tool, and enables input and results to be used intuitively. In the future research, it will be possible to use this tool in various fields by adding or modifying more or new various constraints. In addition, it may be possible to examine whether or not there is a practical tool by studying the cases of companies that used this tool actually.

References

Peerasit Patanakul, Dragan Milosevic, Timothy R. Anderson(2004), "An integer-programming model for assigning projects to project managers" Proceedings of PICMET'04: Portland International Center for Management of Engineering and Technology

Peerasit Patanakul, Dragan Milosevic, Timothy R. Anderson(2007), "A Decision Support Model for Project Manager Assignments" Engineering and Technology Management Faculty Publications and Presentations.

Appendix A. Shiny user interface

A Decision Support Model for Project Manager Assignments

Select Show Solution After Data Entry

Input Data

# of Projects
6

# of Project Managers
6

# of Goals
4

g matrix csv
Browse... Data For OR Team I
Upload complete

v matrix csv
Browse... Data For OR Team I
Upload complete

a matrix csv
Browse... Data For OR Team I
Upload complete

w matrix csv
Browse... Data For OR Team I
Upload complete

d matrix csv
Browse... Data For OR Team I
Upload complete

A, n, s, sc matrix csv
Browse... Data For OR Team I
Upload complete

Data Tables

Goal Value
[1] 4
NPM Value
[1] 6
NProject Value
[1] 6

g matrix data

Goal.1	Goal.2	Goal.3	Goal.4
0.25	0.25	0.25	0.25

v matrix data

Goal.1	Goal.2	Goal.3	Goal.4
36.51	16.67	16.67	21.60
9.13	16.67	16.67	17.58
5.51	16.67	16.67	9.20
22.00	16.67	16.67	21.57
3.08	16.67	16.67	17.28
23.78	16.67	16.67	12.77

a matrix data

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
1.00	1.00	1.00	0.60	1.00	1.00	1.00	0.90	0.80	1.00	1.00
0.90	1.00	1.00	0.60	1.00	1.00	0.90	1.00	0.80	1.00	1.00
0.90	1.00	1.00	0.60	1.00	1.00	0.90	0.90	1.00	1.00	1.00
0.90	1.00	1.00	1.00	1.00	1.00	0.90	0.90	0.80	1.00	1.00
0.90	1.00	1.00	0.60	1.00	1.00	0.90	0.90	0.80	1.00	1.00
0.90	1.00	1.00	0.60	1.00	1.00	0.90	0.90	0.80	1.00	1.00

w matrix data

P1	P2	P3	P4	P5	P6
59.70	63.40	63.90	54.60	76.10	71.90
97.80	97.80	97.80	95.60	97.80	97.80
100.00	100.00	100.00	100.00	100.00	100.00
99.30	99.30	99.30	99.30	99.30	99.30
80.30	87.20	84.50	77.40	96.50	95.70
47.20	57.10	54.30	42.20	77.80	65.50

d matrix data

P1	P2	P3	P4	P5	P6
276	240	208	170	208	156
240	216	184	136	184	120
240	216	184	136	184	120
240	216	184	136	184	120
276	240	208	170	208	156
420	324	276	234	249	216

A, n and sc matrices data

A.Matrix	n.Matrix	sc.Matrix
372	1	1
396	1	0
336	1	1
336	1	1
371	1	0
360	1	0

Appendix B. Entire R code

```r
# Define UI for app ----
ui <- fluidPage(

  # App title ----
  titlePanel("A Decision Support Model for Project Manager Assignments"),

  # Sidebar layout with input and output definitions ----
  sidebarLayout(

    # Sidebar panel for inputs ----
    sidebarPanel(


      # Switch Between Data Entry and Solving
      selectInput("Selection", h5("Select Show Solution After Data Entry"),
                      choices = list("Input Data" = 1, "Show Solution" =
2)),

      # Inputs for problem data
      numericInput("NProjectx",
                      h6("# of Projects"),
                      value = NULL),

      numericInput("NPMx",
                      h6("# of Project Managers"),
                      value = NULL),

      numericInput("NGoalsx",
                      h6("# of Goals"),
                      value = NULL),


      #Requesting CSV Data Files for Program
      fileInput("Gcsv", h6("g matrix csv")),

      fileInput("Vcsv", h6("v matrix csv")),

      fileInput("SmAcsv", h6("a matrix csv")),

      fileInput("Wcsv", h6("w matrix csv")),

      fileInput("Dcsv", h6("d matrix csv")),
```

```r
    fileInput("ANSSCcsv", h6("A, n, sc matrix csv"))


),

# Main panel for displaying outputs ----
mainPanel(
    #isReady <- input$Selection,

    # Conditional Panel for Data Entry
    conditionalPanel(
    condition = "input.Selection == 1",



    h2("Data Tables", align = "center"),

    h5("Goal Value", align = "left"),
    tableOutput("vGoal"),
    h5("NPM Value", align = "left"),
    tableOutput("vNPM"),
    h5("NProject Value", align = "left"),
    tableOutput("vNProject"),


    h5("g matrix data", align = "left"),
    tableOutput("gMatrix"),
    h5("v matrix data", align = "left"),
    tableOutput("vMatrix"),
    h5("a matrix data", align = "left"),
    tableOutput("aMatrix"),
    h5("w matrix data", align = "left"),
    tableOutput("wMatrix"),
    h5("d matrix data", align = "left"),
    tableOutput("dMatrix"),
    h5("A, n and sc matrices data", align = "left"),
    tableOutput("AnsscMatrix")
    ),

    #Conditional Panel for Solution
    conditionalPanel(
    condition = "input.Selection == 2",
    h2("Solution From Data:", align = "center"),
    br(),
    h4("Objective Value:", textOutput("SolutionObjective")),
    br(),
```

```r
        h4("Project Manager Assignments:", tableOutput("SolutionMatrix"))

      )



    )
  )

)

# Core code to display input data or display solution to Project Manager
server <- function(input, output) {

#  DISPLAY TABLES FROM INPUTS

  output$gMatrix <- renderTable({
   GcsvFile <- input$Gcsv

    if (is.null(GcsvFile))
      return(NULL)

    read.csv(GcsvFile$datapath,header = TRUE)
 })
 #Displaying g matrix

 output$vMatrix <- renderTable({
   VcsvFile <- input$Vcsv

    if (is.null(VcsvFile))
      return(NULL)

    read.csv(VcsvFile$datapath,header = TRUE)
 })
 #Displaying v matrix

  output$aMatrix <- renderTable({
   SmAcsvFile <- input$SmAcsv

    if (is.null(SmAcsvFile))
      return(NULL)

    array(c(unlist(read.csv(SmAcsvFile$datapath,FALSE))),dim=c(input$NPMx,
input$NProjectx, input$NGoalsx))
 })
```

```r
#Displaying a matrix


  output$wMatrix <- renderTable({
   WcsvFile <- input$Wcsv

    if (is.null(WcsvFile))
      return(NULL)

    read.csv(WcsvFile$datapath,header = TRUE)
 })
#Displaying w matrix

 output$dMatrix <- renderTable({
    DcsvFile <- input$Dcsv

    if (is.null(DcsvFile))
      return(NULL)

    read.csv(DcsvFile$datapath,header = TRUE)
 })
#Displaying d matrix

 output$AnsscMatrix <- renderTable({
    ANSSCcsvFile <- input$ANSSCcsv

    if (is.null(ANSSCcsvFile))
      return(NULL)

   read.csv(ANSSCcsvFile$datapath,header = TRUE)

 })
#Displaying A, n and sc matrices


#  DISPLAY GOAL, NPM, NProject FROM DATA DIMENSIONS

 output$vGoal <- renderPrint({
    Goal <- input$NGoalsx

    if (is.null(Goal))
      return(NULL)

    Goal
 })
```

```r
#Displaying Goal count

  output$vNPM <- renderPrint({
   NPM <- input$NPMx

    if (is.null(NPM))
      return(NULL)

   NPM
})
#Displaying Goal count


  output$vNProject <- renderPrint({
   NProject <- input$NProjectx

    if (is.null(NProject))
      return(NULL)

   NProject
})
#Displaying Goal count

output$SolutionMatrix <- renderTable({

NProject <- input$NProjectx
NPM   <- input$NPMx
Goal <- input$NGoalsx
ProjNames <- lapply(list(rep("Project_",NProject)),paste0,1:NProject)
                # Project names
PMNames<- lapply(list(rep("PM_",NPM)),
                 paste0,LETTERS[1:NPM])
                # Project managet names

GcsvFile <- input$Gcsv
g <- matrix(c(unlist(read.csv(GcsvFile$datapath,header = TRUE))),ncol = Goal)

  #percent relative contribution of goals to the mission

VcsvFile <- input$Vcsv
v <- matrix(c(unlist(read.csv(VcsvFile$datapath,header = TRUE))),ncol=Goal)
  #percent relative contribution of projects to the goal

SmAcsvFile <- input$SmAcsv
a <- array(c(unlist(read.csv(SmAcsvFile$datapath,FALSE))),dim=c(NPM,
```

```
NProject, Goal))
    #the coefficient of effectiveness in discontinuity management of PM in
projects


WcsvFile <- input$Wcsv
w <- matrix(c(unlist(read.csv(WcsvFile$datapath,header =
TRUE))),ncol=NProject)
    #correspondence matrix



DcsvFile <- input$Dcsv
d <- matrix(c(unlist(read.csv(DcsvFile$datapath,header =
TRUE))),ncol=NProject)
    #time demand of projects per planning horizon



ANSSCcsvFile <- input$ANSSCcsv
ANSSC <- matrix(c(unlist(read.csv(ANSSCcsvFile$datapath,header =
TRUE))),ncol=3)

A <- ANSSC[,1]
    #resource availability of project managers

n <- ANSSC[,2]
    #total number of exisitng projects

sc <- ANSSC[,3]
    #special assignment constraints
    #SC_A1 = SC_C1 = SC_D1 = 1
    #SC_B1 = SC_E1 = SC_F1 = 0

projectmodel <- MIPModel() %>%
  add_variable (x[i, j], i=1:NPM, j=1:NProject,
                type="binary") %>%
  add_variable (r[i], i=1:NPM,
                type="binary") %>%
  add_variable (z[i], i=1:NPM,
                type="binary") %>%


  set_objective (sum_expr(a[i,j,k] * g[k] * v[j,k] * w[i,j] * x[i,j],
                          i=1:NPM, j=1:NProject, k=1:Goal ), "max") %>%


  add_constraint (sum_expr(d[i,j] * x[i,j], j=1:NProject) + 1.5 *
```

```r
((sum_expr(x[i,j], j=1:NProject) +  n[i]) + r[i] - 1) + 4.5 * z[i]
                  <= A[i], i=1:NPM) %>%
      #Constraint 1) Resource availability constraints - 1

  add_constraint (1 - 0.25 * (sum_expr(x[i,j], j=1:NProject) +  n[i])
                  >= r[i], i=1:NPM) %>%
      #Constraint 1) Resource availability constraints - 2

  add_constraint (((sum_expr(x[i,j], j=1:NProject) +  n[i]) + r[i] - 1) / 4
                  <= z[i], i=1:NPM) %>%
      #Constraint 1) Resource availability constraints - 3

  add_constraint (((sum_expr(x[i,j], j=1:NProject) +  n[i]) + r[i] - 1)
                  >= z[i], i=1:NPM) %>%
      #Constraint 1) Resource availability constraints - 4




  #Constraint 2) Project type mix - NOT used

  #Constraint 3) Project phase mix - NOT used




  #Constraint 4) Maximum number of projects

  add_constraint (sum_expr(x[i,j], j=1:NProject)
                  <= 2, i=1:NPM) %>%
   ###### All project managers limit the number of projects they can do
simultaneously to not more than two




  add_constraint (sum_expr(sc[i] * x[i,1], i=1:NPM)
                  == 1) %>%
      #Constraint 5) Special assignment constraints

  add_constraint (sum_expr(x[i,j], i=1:NPM)
                  == 1, j=1:NProject) %>%
      #Constraint 6) Fixed assignment - Project assignment constraints

  add_constraint (sum_expr(x[i,j], j=1:NProject)
                  == 1, i=1:NPM)
```

```r
    #Constraint 6) Fixed assignment - Project manager assignment
constraints



  #Constraint 7) Project interdependency - NOT used


#Solve
projectmodel_result <- solve_model(projectmodel, with_ROI(solver = "glpk"))

x_solution <- t(as.matrix(as.numeric(get_solution(projectmodel_result,
x[i,j])[,4])))

x_solution_result <- matrix(x_solution, ncol=NProject, dimnames=c(PMNames,
ProjNames))

#Transform table of '0's and '1's into a table of Project Managers and
Projects
indices <- data.frame(ind= which(x_solution_result == 1, arr.ind=TRUE))
indices$rnm <- rownames(x_solution_result)[indices$ind.row]
indices$cnm <- colnames(x_solution_result)[indices$ind.col]

Final_Results <- indices [,3:4]

colnames(Final_Results)<- c("PM","Project")

Final_Results

 })



output$SolutionObjective <- renderPrint({

NProject <- input$NProjectx
NPM   <- input$NPMx
Goal <- input$NGoalsx
ProjNames <- lapply(list(rep("Project_",NProject)),paste0,1:NProject)
            # Project names
PMNames<- lapply(list(rep("PM_",NPM)),
              paste0,LETTERS[1:NPM])
            # Project managet names

GcsvFile <- input$Gcsv
g <- matrix(c(unlist(read.csv(GcsvFile$datapath,header = TRUE))),ncol = Goal)
```

```r
   #percent relative contribution of goals to the mission

VcsvFile <- input$Vcsv
v <- matrix(c(unlist(read.csv(VcsvFile$datapath,header = TRUE))),ncol=Goal)
   #percent relative contribution of projects to the goal

SmAcsvFile <- input$SmAcsv
a <- array(c(unlist(read.csv(SmAcsvFile$datapath,FALSE))),dim=c(NPM,
NProject, Goal))
    #the coefficient of effectiveness in discontinuity management of PM in
projects

WcsvFile <- input$Wcsv
w <- matrix(c(unlist(read.csv(WcsvFile$datapath,header =
TRUE))),ncol=NProject)
    #correspondence matrix


DcsvFile <- input$Dcsv
d <- matrix(c(unlist(read.csv(DcsvFile$datapath,header =
TRUE))),ncol=NProject)
    #time demand of projects per planning horizon


ANSSCcsvFile <- input$ANSSCcsv
ANSSC <- matrix(c(unlist(read.csv(ANSSCcsvFile$datapath,header =
TRUE))),ncol=3)

A <- ANSSC[,1]
    #resource availability of project managers

n <- ANSSC[,2]
    #total number of exisitng projects

sc <- ANSSC[,3]
    #special assignment constraints
    #SC_A1 = SC_C1 = SC_D1 = 1
    #SC_B1 = SC_E1 = SC_F1 = 0

projectmodel <- MIPModel() %>%
  add_variable (x[i, j], i=1:NPM, j=1:NProject,
                type="binary") %>%
  add_variable (r[i], i=1:NPM,
                type="binary") %>%
```

```r
add_variable (z[i], i=1:NPM,
              type="binary") %>%


set_objective (sum_expr(a[i,j,k] * g[k] * v[j,k] * w[i,j] * x[i,j],
                        i=1:NPM, j=1:NProject, k=1:Goal ), "max") %>%


add_constraint (sum_expr(d[i,j] * x[i,j], j=1:NProject) + 1.5 *
((sum_expr(x[i,j], j=1:NProject) +  n[i]) + r[i] - 1) + 4.5 * z[i]
                <= A[i], i=1:NPM) %>%
    #Constraint 1) Resource availability constraints - 1

add_constraint (1 - 0.25 * (sum_expr(x[i,j], j=1:NProject) +  n[i])
                >= r[i], i=1:NPM) %>%
    #Constraint 1) Resource availability constraints - 2

add_constraint (((sum_expr(x[i,j], j=1:NProject) +  n[i]) + r[i] - 1) / 4
                <= z[i], i=1:NPM) %>%
    #Constraint 1) Resource availability constraints - 3

add_constraint (((sum_expr(x[i,j], j=1:NProject) +  n[i]) + r[i] - 1)
                >= z[i], i=1:NPM) %>%
    #Constraint 1) Resource availability constraints - 4




#Constraint 2) Project type mix - NOT used

#Constraint 3) Project phase mix - NOT used



#Constraint 4) Maximum number of projects

add_constraint (sum_expr(x[i,j], j=1:NProject)
                <= 2, i=1:NPM) %>%
  ###### All project managers limit the number of projects they can do
simultaneously to not more than two



add_constraint (sum_expr(sc[i] * x[i,1], i=1:NPM)
```

25

```r
                == 1) %>%
      #Constraint 5) Special assignment constraints

  add_constraint (sum_expr(x[i,j], i=1:NPM)
                == 1, j=1:NProject) %>%
      #Constraint 6) Fixed assignment - Project assignment constraints

  add_constraint (sum_expr(x[i,j], j=1:NProject)
                == 1, i=1:NPM)
      #Constraint 6) Fixed assignment - Project manager assignment
constraints



  #Constraint 7) Project interdependency - NOT used



projectmodel_result <- solve_model(projectmodel, with_ROI(solver = "glpk"))

objective_solution <- objective_value(projectmodel_result)

objective_solution


 })


}
```
```{r} shinyApp(ui = ui, server = server) ```