WATER RESOURCES MANAGEMENT

LECTURE 5 – LINEAR PROGRAMMING

INTRODUCTION

- Resource Allocation using Linear Programming
- Setting up an LP model
- Solving LP using R
  - Install Package(s)
  - Load the Library
  - Build the Model

RESOURCE ALLOCATION USING LP

- Linear Programming is a mathematical programming tool to search for optimal solutions to allocation problems that have
  - Linear Objective Function (Objective is a weighted linear combination of the decision variables)
  - Linear Constraint Set
  - Non-negative decision variables

OBJECTIVE FUNCTION

- The objective function (merit function) is the mathematical expression of the benefit, utility, or cost incurred for various decisions. In LP it must be a linear combination of the decision variables.
- Economic cost is a common example:
- $\text{COST}(\mathbf{x}) = w_1 * x_1 + w_2 * x_2 + \ldots + w_N * x_N$

COST COEFFICIENTS

- COST($\mathbf{x}$) = $w_1 * x_1 + w_2 * x_2 + \ldots + w_N * x_N$
- The weights (w_1,w_2, etc.) are called the cost (utility, benefit) coefficients
- The variables (x_1,x_2, etc.) are called the decision or allocation variables. They represent how much of something is allocated at some unit cost (the cost coefficient).
- For example if x_1 is the volume of water delivered in liters and w_1 is the price \$1.00/liter
  - Then the cost of 6.2 liters is $w_1 * x_1 = (1.0)(6.2) = \$6.20$

CONSTRAINTS

- The amount of a decision variable that is available or can be used is called a constraint.
- Suppose that we only have 100 Liters of water available to deliver, then the constraint would be $x_1 <= 100$; that is, we can deliver anywhere from 0 to 100 liters, but not more (or less)
- The entire set of constraints is called the constraint set. It must be comprised of linear combinations of the decision variables (for an LP solution).

THE LP MODEL

- The combination of the objective function, and the constraint set, along with a directive to either minimize (make small) or maximize (make big) the objective function is the linear program.

EXAMPLE

Construction company contracted to excavate 6-foot and 18-foot wide trenches. Can transport no more than $10,000 yd^3/day of excavation material from the site because of a limited supply of dump trucks. To meet the construction schedule, the company must excavate at least 1,600 yd^3/day from the 6-foot trench and at least 3,000 yd^3/day from the 18-foot trench.

EXAMPLE

The company has 12 heavy equipment operators that can operate either a Backhoe Type 1 or Backhoe Type 2. The company has a total of 12 of each type of backhoe available – unused machines can be assigned to another job.

Backhoe Type 1 can excavate 200 yd^3/day from a 6-foot trench at a cost of $394 per machine day. Backhoe Type 2 can excavate 1,000 yd^3/day from an 18-foot trench at a cost of $1,110 per machine day

CONSTRUCTION MANAGEMENT EXAMPLE

What is the best allocation of operators (machines) to minimize daily cost and meet scheduling requirements?

SETTING UP A LINEAR PROGRAM

- As with all allocation problems (regardless of linearity) we need a goal.
- In this example the goal is to minimize the daily machine cost, so the cost (objective) function is expressed as

$$COST(\mathbf{x}) = \$394x_1 + \$1110x_2$$

Where,

x_1 is the number of operators assigned to a Type 1 machine

x_2 is the number of operators assigned to a Type 2 machine.

SETTING UP A LINEAR PROGRAM

- Next we need to explicitly state the constraint set.
- The first constraint is on the total amount of material that can be transported off the site as a function of machine count – in this case

$$200\ x_1 + 1000\ x_2 <= 10{,}000 \text{ (Dump Trucks)}$$

SETTING UP A LINEAR PROGRAM

- The next constraint is on minimum trenching requirements for each trench width

$$200\,x_1 \geq 1{,}600 \quad (\text{6-foot trench})$$

$$1000\,x_2 \geq 3{,}000 \quad (\text{18-foot trench})$$

SETTING UP A LINEAR PROGRAM

- The next constraint is on the total number of operators and supply of machines available

$$x_1 + x_2 <= 12 \text{ (Operators)}$$

$$x_1 <= 12 \text{ (Type 1 Available)}$$

$$x_2 <= 12 \text{ (Type 2 Available)}$$

SETTING UP A LINEAR PROGRAM

- The last constraint is non-negativity

$$x_1 >= 0$$

$$x_2 >= 0$$

CONSTRUCTION MANAGEMENT LINEAR PROGRAM

- Lastly we need to decide if we are minimizing or maximizing the objective function – in this example, it is minimization.
- Next we will write the entire model at once, the result is the linear programming problem

CONSTRUCTION MANAGEMENT LINEAR PROGRAM

Min $COST(\mathbf{x}) = \$394x_1 + \$1110x_2$

Subject to

$200\,x_1 + 1000\,x_2 \le 10{,}000$		(Dump Trucks)
$200\,x_1 \ge 1{,}600$		(6-foot trench)
$1000\,x_2 \ge 3{,}000$		(18-foot trench)
$x_1 + x_2 \le 12$		(Operators)
$x_1 \le 12$		(Type 1 Available)
$x_2 \le 12$		(Type 2 Available)

CONSTRUCTION MANAGEMENT LINEAR PROGRAM

- The last two constraints are redundant (in this example!), so the LP is

Min COST(**x**) = $\$394x_1 + \$1110x_2$

Subject to

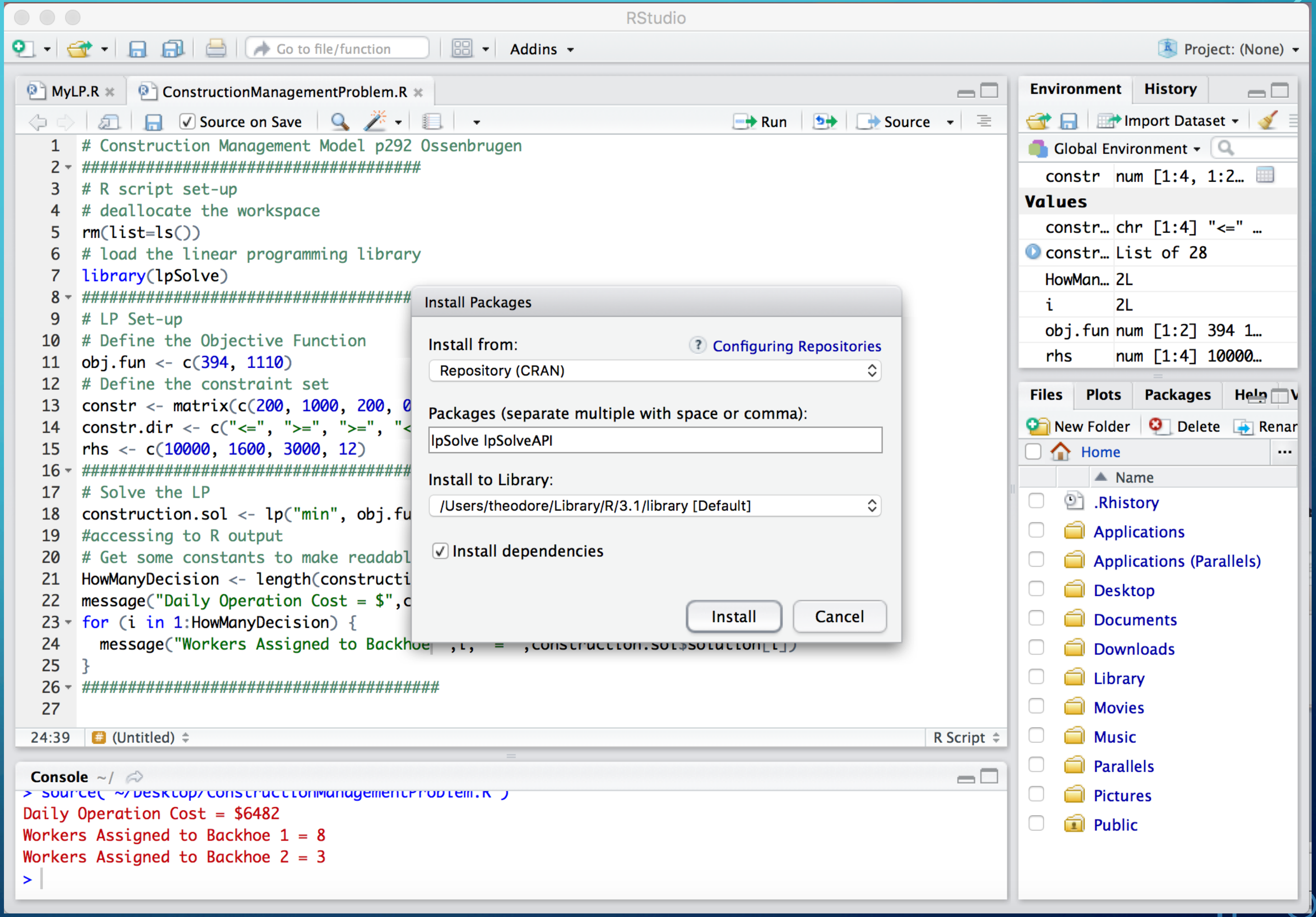
$200\,x_1 + 1000\,x_2 <= 10{,}000$	(Dump Trucks)
$200\,x_1 >= 1{,}600$	(6-foot trench)
$1000\,x_2 >= 3{,}000$	(18-foot trench)
$x_1 + x_2 <= 12$	(Operators)

SOLVING THE LINEAR PROGRAM

- Really simple LP can be solved by inspection; if there are only 2 variables, they can also be solved graphically
- For larger problems usually a variant of the SIMPLEX algorithm (Dantzig's algorithm with lexicographical pivoting) is used. The details of the algorithm are described in the readings.
- Here we will use R as a tool to solve the LP and find the decision variable values and the associated cost of the decision(s).

OBTAIN THE REQUIRED PACKAGES

- We will need the packages:
- **LpSolve**, and **LpSolveAPI**
- In R Studio simply run the pagakge installer and it will get the packages from the CRAN



TRANSLATING THE LINEAR PROGRAM TO THE R SCRIPT

- **lpSolve** has a particular syntax; for small problems we can type the parts directly.
  - Larger problems write script to generate the LP from an input file

TRANSLATING THE LINEAR PROGRAM TO THE R SCRIPT

- First, clear the R workspace and load the library

```r
# Construction Management Model
#####################################
# R script set-up
# deallocate the workspace
rm(list=ls())
# load the linear programming library
library(lpSolve)
#####################################
```

TRANSLATING THE LINEAR PROGRAM TO THE R SCRIPT

- Then construct the objective function
  - In the example the cost coefficient for Backhoe 1 is $394 per … and Backhoe 2 is $1110 per …. These weights are supplied to the objective function as a vector.

```
#######################################
# LP Set-up
# Define the Objective Function
obj.fun <- c(394, 1110)
```

TRANSLATING THE LINEAR PROGRAM TO THE R SCRIPT

- Then construct the constraint set – the constraint coefficient matrix, inequalities, and right-hand-side are entered as separate objects

$$
\begin{aligned}
200x_1 + 1000x_2 &\le 10{,}000 \\
200x_1 + 0x_2 &\ge 1{,}600 \\
0x_1 + 1000x_2 &\ge 3{,}000 \\
x_1 + x_2 &\le 12
\end{aligned}
$$

```r
# Define the constraint set
constr <- matrix(c(200, 1000, 200, 0, 0, 1000, 1, 1), ncol = 2, byrow=TRUE)
constr.dir <- c("<=", ">=", ">=", "<=")
rhs <- c(10000, 1600, 3000, 12)
########################################
```

TRANSLATING THE LINEAR PROGRAM TO THE R SCRIPT

- Now we can call the solver and have it search for values that satisfy the constraint set and minimize the objective function
- Actually not much to the R script (but there is a lot going on behind the scene).

```r
######################################
# Solve the LP
construction.sol <- lp("min", obj.fun, constr, constr.dir, rhs, compute.sens = TRUE)
######################################
```

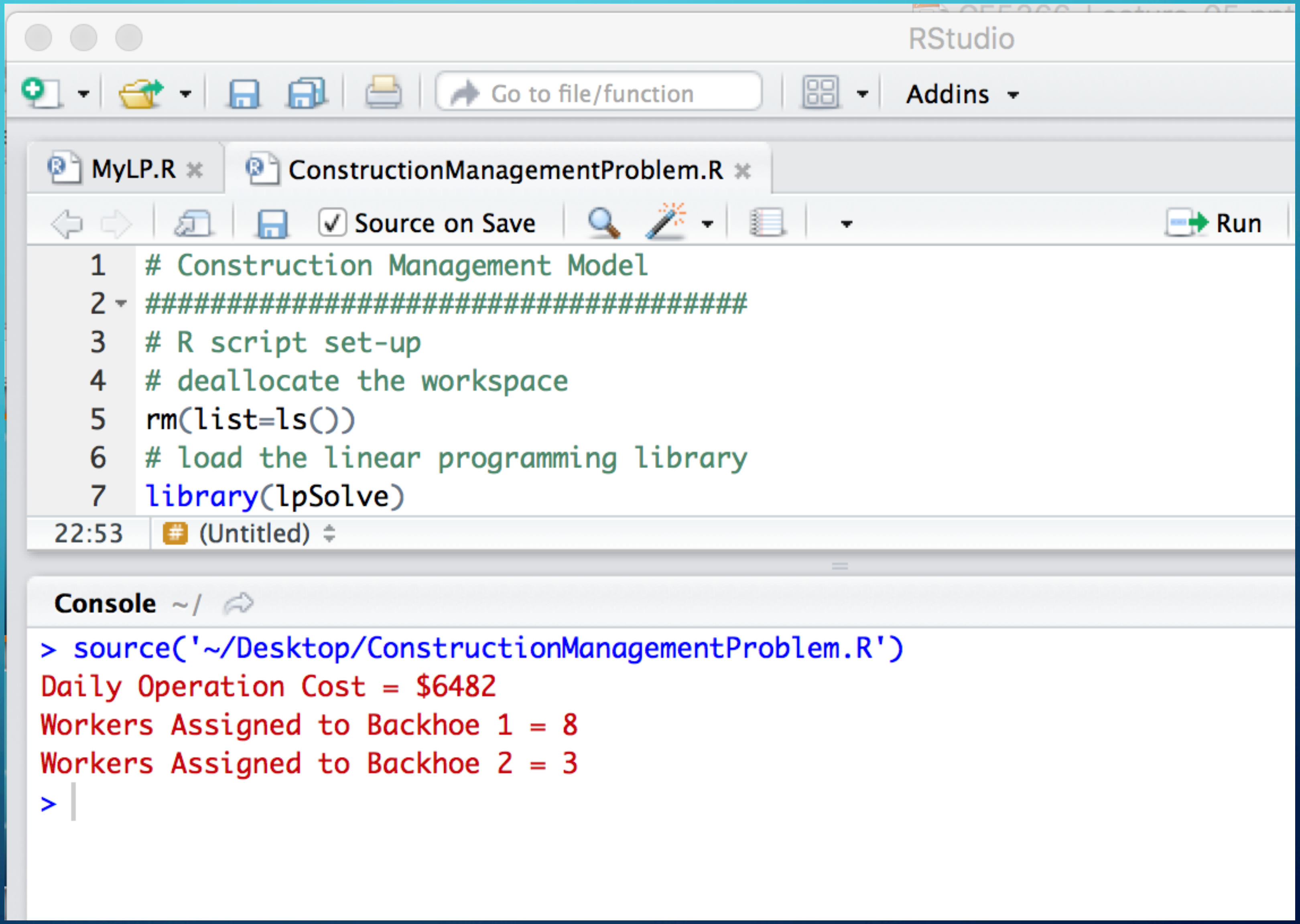
TRANSLATING THE LINEAR PROGRAM TO THE R SCRIPT

- Lastly, interrogate the solution object (**construction.sol**) and generate some meaningful output for the analyst to interpret

```r
######################################
# Generate meaningful output
# Get some constants to make readable output
HowManyDecision <- length(construction.sol$solution)
message("Daily Operation Cost = $",construction.sol$objval) #objective function value
for (i in 1:HowManyDecision) {
  message("Workers Assigned to Backhoe ",i," = ",construction.sol$solution[i])
}
#######################################
```

LP SOLUTION TO THE CONSTRUCTION MANAGEMENT EXAMPLE

- Run the R script, and examine the results!

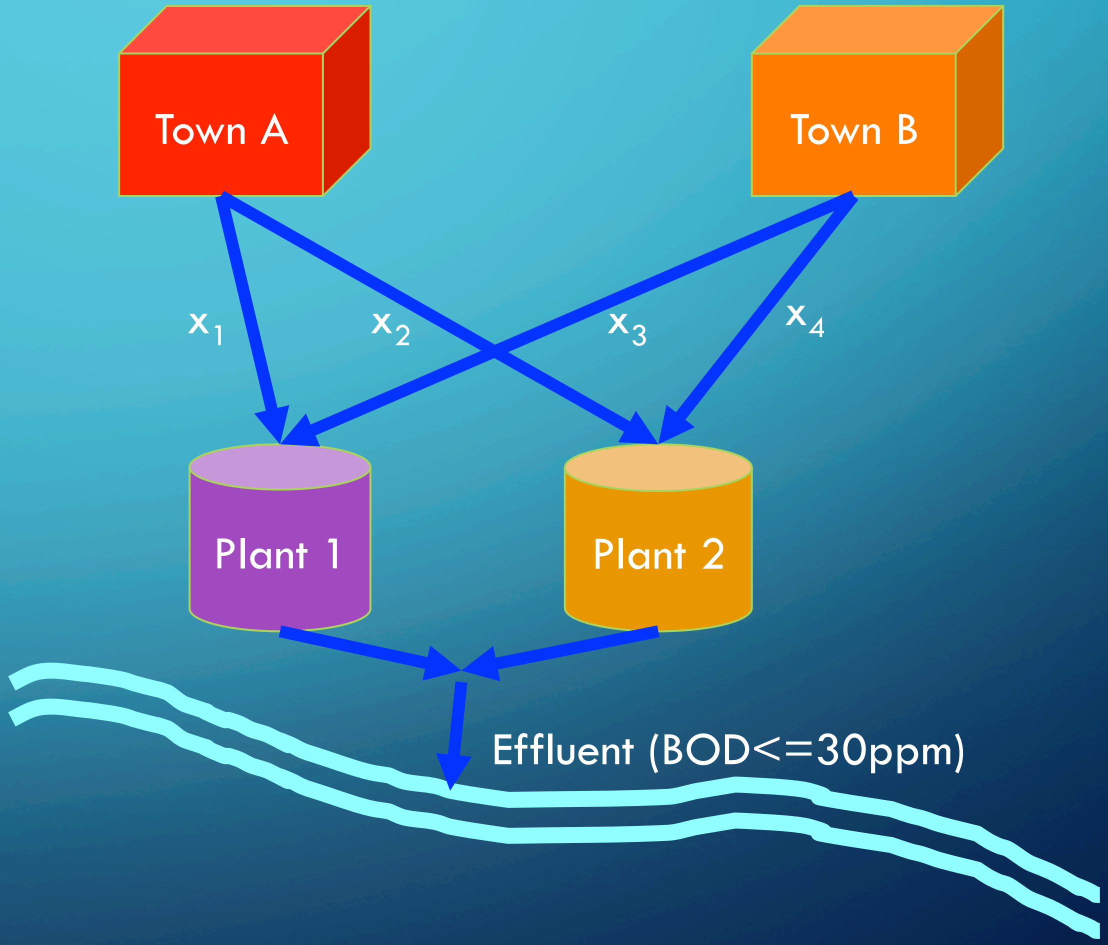


WASTEWATER TREATMENT PLANT ALLOCATION

- The construction management example illustrates how to set up a LP model – it could have been solved graphically – now we examine an example that cannot not be solved graphically (using 2D-paper)

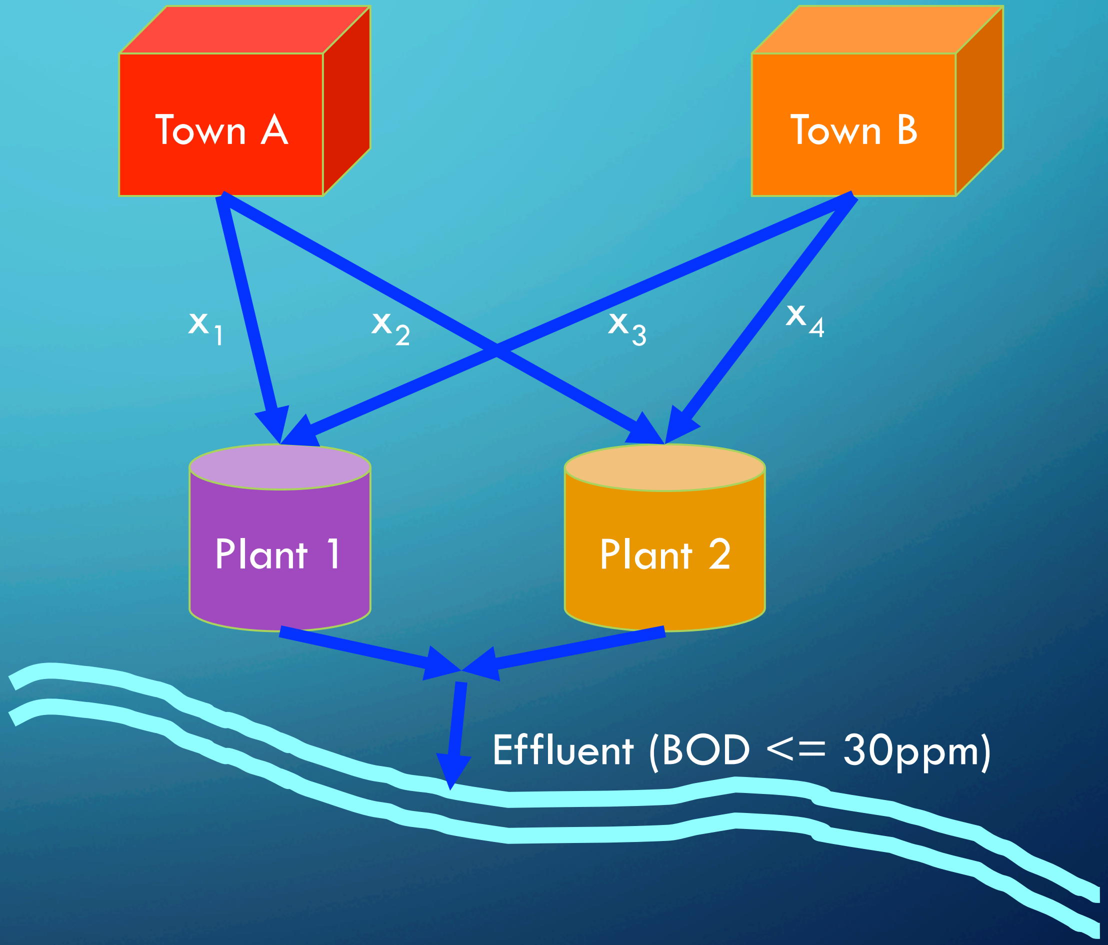
WASTEWATER TREATMENT PLANT ALLOCATION

- Town A produces 3 MGD of BOD=200ppm wastewater per day
- Town B produces 2 MGD of BOD=200ppm wastewater per day



WASTEWATER TREATMENT PLANT ALLOCATION

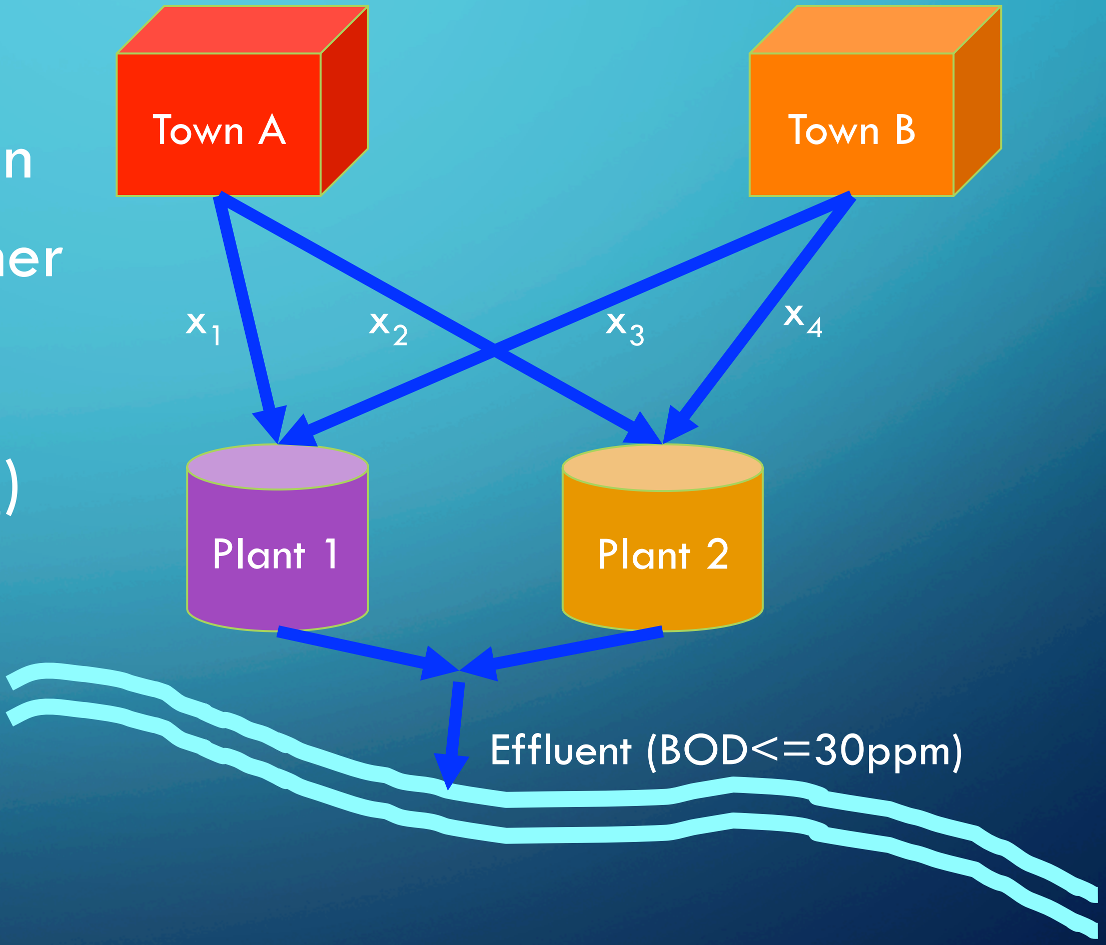
- Plant 1 can treat 3 MGD and remove 90% of incoming BOD
- Plant 2 can treat 4 MGD and remove 80% of incoming BOD



WASTEWATER TREATMENT PLANT ALLOCATION

- The regional treatment operator can allocate wastewater flows from either town to either plant
- Unit Costs for each pipeline ($x_1 \ldots x_4$) are:

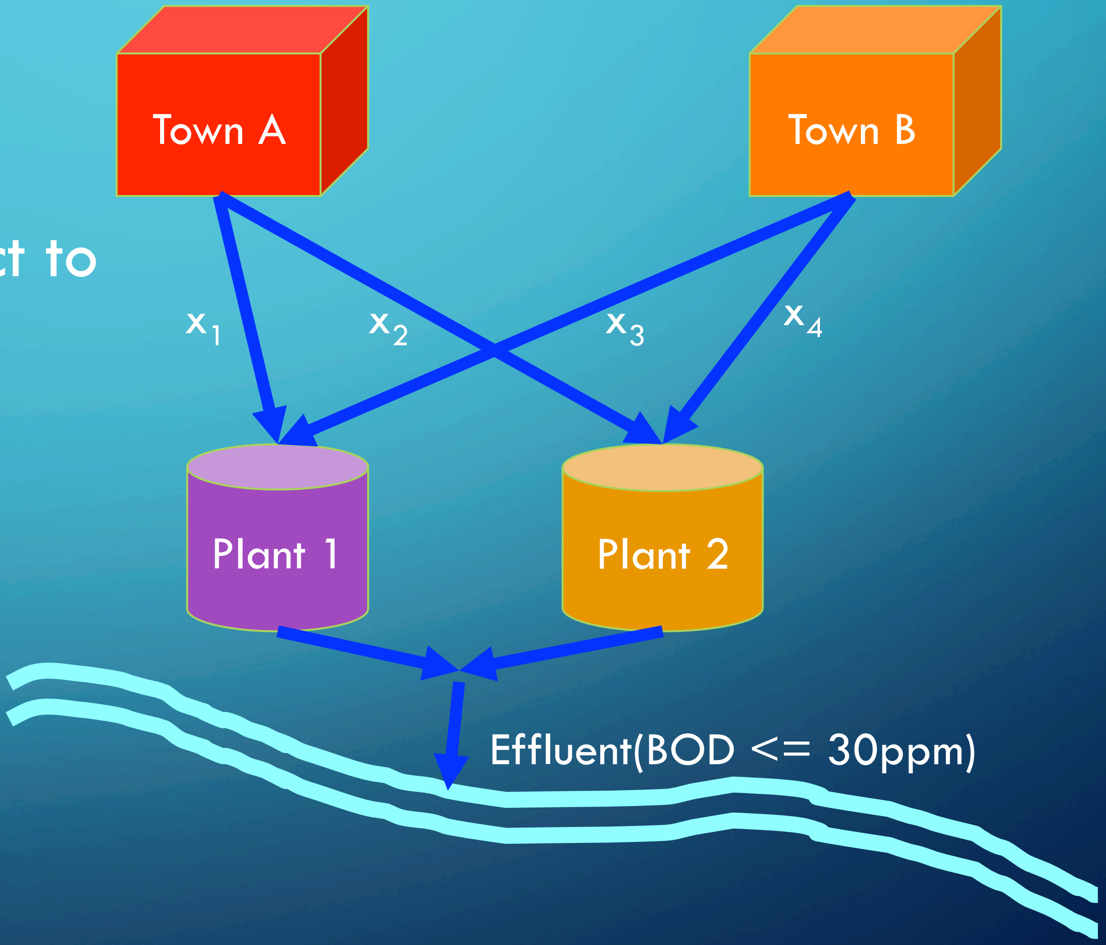
Pipeline	$1000/MGD-year
1	$46
2	$50
3	$55
4	$40



WASTEWATER TREATMENT PLANT ALLOCATION

- What allocation minimizes the treatment and pumping costs subject to the requirement that the effluent concentration not exceed 30 ppm BOD?

Pipeline	$1000/MGD-year
1	$46
2	$50
3	$55
4	$40

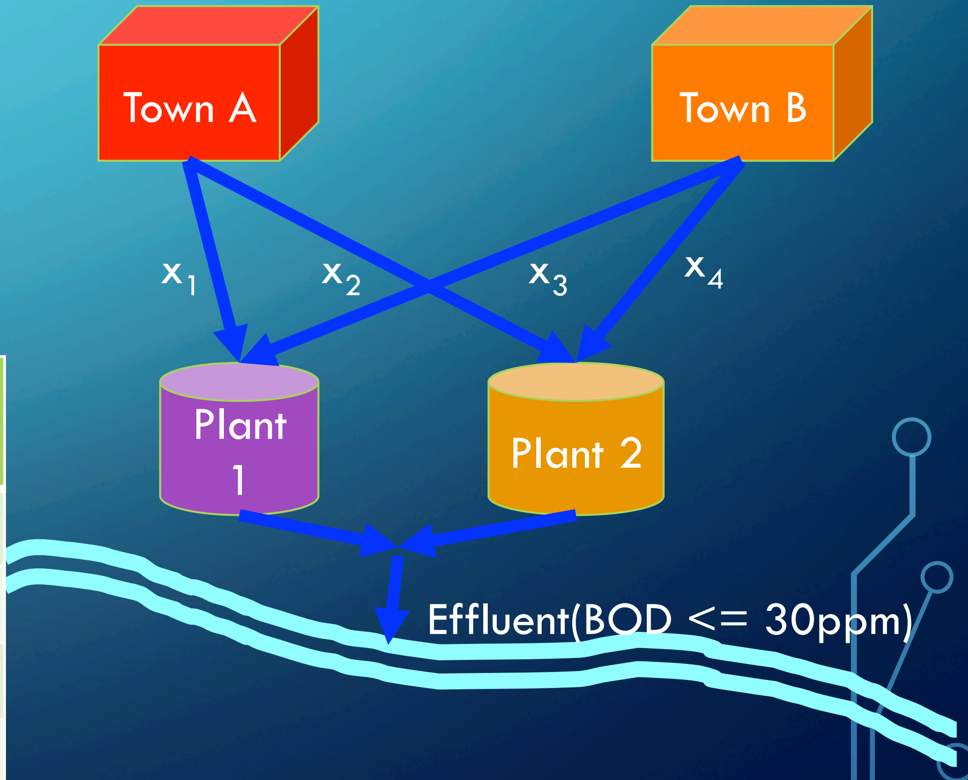


WASTEWATER TREATMENT PLANT ALLOCATION

- Write the objective function

$$C(x) = 46x_1 + 50x_2 + 55x_3 + 40x_4$$

Pipeline	$1000/MGD-year
1	$46
2	$50
3	$55
4	$40



WASTEWATER TREATMENT PLANT ALLOCATION

- Write the objective function

$$C(x) = 46x_1 + 50x_2 + 55x_3 + 40x_4$$

- Water quality constraint

$$20x_1 + 40x_2 + 20x_3 + 40x_4 \le 150$$

(BOD*FLOW after treatment)

WASTEWATER TREATMENT PLANT ALLOCATION

- Write the objective function

$$C(x) = 46x_1 + 50x_2 + 55x_3 + 40x_4$$

- Water quality constraint

$$20x_1 + 40x_2 + 20x_3 + 40x_4 \leq 150$$

(BOD*FLOW after treatment)

- Treatment Plant Capacity

$$1x_1 + 0x_2 + 1x_3 + 0x_4 \leq 3$$

$$0x_1 + 1x_2 + 0x_3 + 1x_4 \leq 4$$

WASTEWATER TREATMENT PLANT ALLOCATION

- Write the objective function

$$C(x) = 46x_1 + 50x_2 + 55x_3 + 40x_4$$

- Water quality constraint

$$20x_1 + 40x_2 + 20x_3 + 40x_4 \le 150$$

(BOD*FLOW after treatment)

- Treatment Plant Capacity

$$1x_1 + 0x_2 + 1x_3 + 0x_4 \le 3$$
$$0x_1 + 1x_2 + 0x_3 + 1x_4 \le 4$$

- All water must be treated

$$1x_1 + 1x_2 + 0x_3 + 0x_4 \ge 3$$
$$0x_1 + 0x_2 + 1x_3 + 1x_4 \ge 2$$

WASTEWATER TREATMENT PLANT ALLOCATION

- The next step is to translate the Linear Program model into R Script
- First the objective function

```
# Wastewater Treatment Plant Allocation
#####################################
# R script set-up
# deallocate the workspace
rm(list=ls())
# load the linear programming library
library(lpSolve)
#####################################
# LP Set-up
# Define the Objective Function
obj.fun <- c(46,50,55,40)
```

WASTEWATER TREATMENT PLANT ALLOCATION

- The constraint set

```
# Define the constraint set
constr <- matrix(c(20,40,20,40,1,0,1,0,0,1,0,1,1,1,0,0,0,0,1,1), ncol = 4, byrow=TRUE)
constr.dir <- c("<=", "<=", "<=", ">=", ">=")
rhs <- c(150, 3, 4, 3, 2)
######################################
```

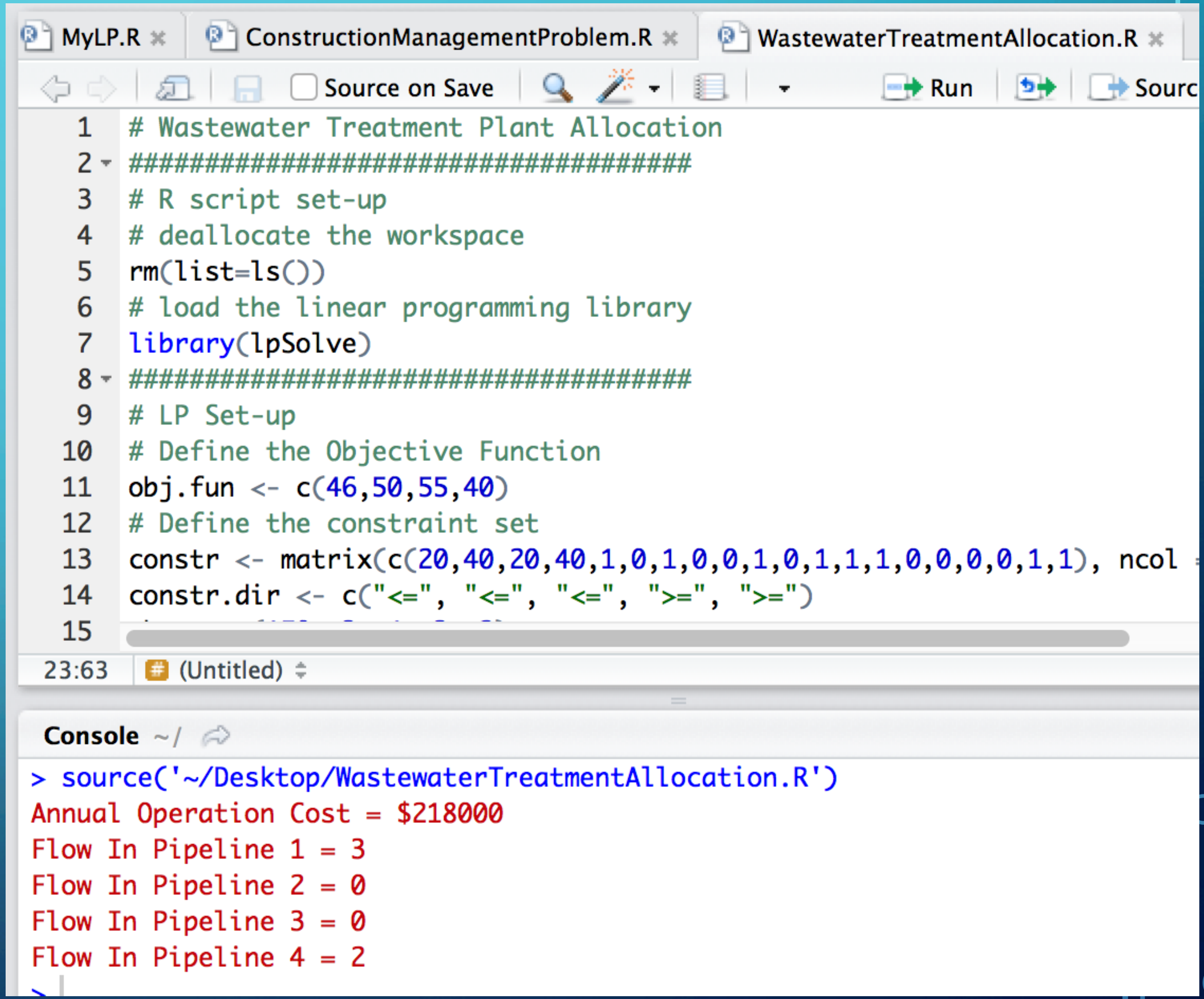
WASTEWATER TREATMENT PLANT ALLOCATION

- Solve the LP and write results

```
######################################
# Solve the LP
wastewater.sol <- lp("min", obj.fun, constr, constr.dir, rhs, compute.sens = TRUE)
######################################
# Generate meaningful output
# Get some constants to make readable output
HowManyDecision <- length(wastewater.sol$solution)
message("Daily Operation Cost = $",wastewater.sol$objval) #objective function value
for (i in 1:HowManyDecision) {
  message("Flow In Pipeline ",i," = ",wastewater.sol$solution[i])
}
#######################################
```

WASTEWATER TREATMENT PLANT ALLOCATION

- Run the script



SUMMARY

- Linear Programming as a tool to allocate resources (make decisions)
- Structure of an LP
- Simple Examples
- Solved using R and LpSolve package