

Projects Manual for Developing Spreadsheet-Based Decision Support Systems

Using Excel and VBA for Excel

Sandra D. Ekşioğlu

**Industrial and Systems Engineering
Mississippi State University, Mississippi State**

Michelle M.H. Şeref

**Decision and Information Sciences
Warrington College of Business
University of Florida, Gainesville**

Ravindra K. Ahuja

**Industrial and Systems Engineering
University of Florida, Gainesville &
Innovative Scheduling, Inc. Gainesville**

Wayne L. Winston

**Operations and Decision Technologies
Kelly School of Business
Indiana University, Bloomington**

Contents

PROJECT 1	Snow Disposal Assignment Problem
PROJECT 2	Bakery Distribution System
PROJECT 3	Blending Problem
PROJECT 4	Helicopter Flight Scheduling
PROJECT 5	The Diet Problem
PROJECT 6	Dairy Routing Problem
PROJECT 7	The News Vendor Problem - I
PROJECT 8	Machine Scheduling Problem
PROJECT 9	Production Management at a Stills Mill
PROJECT 10	Fire Station Location Problem
PROJECT 11	Decision Support System for Hospital Management
PROJECT 12	Hot to Prepare the Best Cookies...
PROJECT 13	Quality Control Support System
PROJECT 14	Spare Management Support System
PROJECT 15	Statistical Process Control
PROJECT 16	Support System for a Car Rental Company
PROJECT 17	Support System for a Bus Transportation Company
PROJECT 18	Production Management Support System
PROJECT 19	Cash Flow Analysis
PROJECT 20	Capital Budgeting
PROJECT 21	The Lockbox Problem

PROJECT 22	Simulation: An Application in Manufacturing
PROJECT 23	Support System for an Agricultural Business
PROJECT 24	Support System for Production Scheduling
PROJECT 25	Task Management Support System
PROJECT 26	Markov Chains
PROJECT 27	Support System for a Job Assignment Problem
PROJECT 28	Modeling Queuing Systems
PROJECT 29	Making Decisions about Advertising
PROJECT 30	Support System for the Multi-Period Capital Budgeting Problem
PROJECT 31	Understanding the Central Limit Theorem
PROJECT 32	Game of Words
PROJECT 33	Simulating the Check-In Process in an Airport
PROJECT 34	Decision Support System for Product Pricing
PROJECT 35	Managing Product Quality
PROJECT 36	Aggregate Production Planning Problem
PROJECT 37	Reliability Analysis
PROJECT 38	Constraint Shortest Path Problem
PROJECT 39	Decision Making Under Uncertainty
PROJECT 40	Economic Lot-Sizing Problem
PROJECT 41	Facility Location Problem
PROJECT 42	Knapsack Problem
PROJECT 43	Estimating the Return on Investments
PROJECT 44	Option Pricing
PROJECT 45	Implementing a Combinatorial Auction Algorithm
PROJECT 46	Tanker Scheduling Problem
PROJECT 47	Traveling Salesman Problem
PROJECT 48	Vehicle Routing Problem
PROJECT 49	Managing Financial Instruments: Bonds
PROJECT 50	Managing the Inspection Process in a Production Line
PROJECT 51	Staff Management at a Call Center
PROJECT 52	Choosing a Transportation Mode
PROJECT 53	Random Number Generator

PROJECT 54	Generating Random Variables
PROJECT 55	Board Game: Connect Four
PROJECT 56	Board Game: Lights Puzzle
PROJECT 57	Board Game: Tic-Tac-Toe
PROJECT 58	Board Game: TacTix
PROJECT 59	Board Game: Diamonds
PROJECT 60	Board Game: Dots
PROJECT 61	Board Game: Collector
PROJECT 62	Board Game: Tag
PROJECT 63	The News Vendor Problem - II
PROJECT 64	Warehouse Layout Problem
PROJECT 65	Personnel Assignment Problem
PROJECT 66	Degree-Constraint Minimum Spanning Tree Problem
PROJECT 67	Multi-Item Production Planning Problem
PROJECT 68	Land Management Problem
PROJECT 69	Managing Customer Relationships
PROJECT 70	Bin Packing Problem
PROJECT 71	Managing Inventories
PROJECT 72	Material Requirements Planning (MRP)
PROJECT 73	Lot-sizing in MRP Systems
PROJECT 74	Library Support System
PROJECT 75	Managing a Call Center

1 Snow Disposal Assignment Problem

Problem Description

Snow removal and disposal are important activities in many cities around the world. In order to facilitate the traffic flow in urban areas that receive heavy snowfalls, snow is first plowed from streets and sidewalks and then hauled to disposal sites. A city is typically divided into many sectors that are cleared of snow concurrently. Because transportation of snow is expensive, it has become an important strategic problem.

This case study considers the snow removal and disposal problem in the city of Montreal. For the purpose of this study, the city is divided into 60 sectors and 20 disposal sites. On average, about 300,000 truckloads of snow are hauled from sectors to disposal sites each year. Approximately 660 trucks are used in Montreal to haul the snow. There are four different types of disposal sites: river sites, quarry sites, sewer chutes, and surface sites. The city has three river sites along the St. Lawrence River, the Francon quarry site, ten sewer chutes, and six surface disposal sites. Spreadsheet 1 presents the volume of snow that each type of disposal site can hold and the corresponding handling costs. The average cost is about \$0.24 (Canadian) per cubic meter of snow.

The snow disposal assignment problem finds the best assignment of snow removal sectors to snow disposal sites for a given set of sectors and cities. We assume that the cost of serving each sector is proportional to the straight-line distance between the centroids of the sector and the assigned disposal site. Therefore, we seek an assignment that minimizes the sum of the distance from the centroid of each sector to the centroid of each disposal site.

The following is the mixed integer programming (MIP) formulation of the snow disposal assignment problem.

$$\begin{aligned}
 & \min \sum_{i=1}^n \sum_{j=1}^m d_{ij} v_i x_{ij} \\
 & \text{s.t.} \\
 & \sum_{i=1}^n v_i x_{ij} \leq V_j \quad \forall j \\
 & \sum_{i=1}^n r_i x_{ij} \leq R_j \quad \forall j \\
 & \sum_{j=1}^m x_{ij} = 1 \quad \forall i \\
 & x_{ij} \in \{0,1\} \quad \forall i, j.
 \end{aligned}$$

Where, d_{ij} presents the distance from the centroid of sector i to the centroid of disposal site j ; V_j presents the annual capacity of site j (m^3); v_i presents the annual volume of snow in sector i (m^3); r_i presents the snow removal rate in sector i (m^3/hr); R_j presents the maximum snow receiving rate of site j (m^3/hr); x_{ij} is a binary variable that takes a value of 1 if sector i is assigned to disposal site j , and 0 otherwise.

The snow disposal problem for the city of Montreal has been solved by Campbell et al. (1995). For the purpose of this project, the student will have to solve and demonstrate the performance of the heuristic proposed on randomly generated problems. The user will be asked to provide the following information: the number of sectors and the number of sites available. The rest of the information (e.g., annual volume of snow in each sector, snow removal rate, etc.) could be randomly generated or read from a file.

Excel Spreadsheets

1. Build a spreadsheet that presents the distance from the centroid of each sector to each disposal site (in km).
2. Build a spreadsheet that presents estimates on the annual volume of snow in each sector (m³/year).
3. Build a spreadsheet that presents the hourly snow removal rate in each sector (m³/hour).

A Heuristic Approach

The snow disposal problem can be formulated as an integer-programming problem. This formulation can be viewed as a generalized assignment problem. One can solve this problem to optimality; however, the solution times are far too long. Fast heuristic procedures can be used to solve the problem, allow interactive use, and facilitate sensitivity analysis. We present in here a heuristic that can be used to solve the problem. This heuristic consists of the following two phases: (1) Assign each sector to a disposal site based on a penalty defined as the difference between the best and the second best assignment. (2) Perform a two-opt exchange procedure in which sectors are considered two at a time and reassigned if the new assignment decreases the total value of the objective function.

(A) Penalty-based assignment

- (1) For each sector i , calculate its penalty:
 Penalty (i) = annual quantity of snow in sector i multiplied by the difference between the distance to the closest acceptable site and the distance to the second closest acceptable site. A site is acceptable to a sector if the annual capacity is not exceeded by the assignment. Sort the sectors in decreasing order of penalties. Assign the sectors (starting with the ones at the top of the list) to sites. Adjust each site's residual capacity accordingly, and recalculate the penalty of a sector if one of its two closest sites becomes unacceptable.
- (2) Assign to a dummy site the sectors that could not be assigned to the existing sites because of capacity restrictions. Set the distance from the dummy site to each sector to infinity
- (3) Assign to a dummy site the sectors that could not be assigned to the existing sites because of capacity restrictions. Set the distance from the dummy site to each sector to infinity.

(B) Two-opt exchange

- (4) For each pair of sectors i and j , already assigned to sites x and y , identify a pair of sites k and l such that $k \neq x$ or $l \neq y$. Assign sector i to site k and sector j to site l if the total distance traveled decreases, and the annual capacity of site k and l is not violated.

- (5) Repeat step 4 until no further improvement can be obtained.

User Interface

1. Build a welcome form.
2. Build a form that allows the user to define the size of the problem and generate or read the data needed. The form should allow the user to enter the total number of disposal sites and snow sectors. Insert an option group to allow the user to choose a method of generating the problem data. The two options are to generate the data randomly or to upload the data from a file. Insert an option group that allows the user to choose a method of solving this problem: MIP formulation or the heuristic approach. Insert a command button that, when clicked on, solves the snow disposal problem using the data and solution method chosen by the user.
3. Build a form that presents the final solution (the assignment of sectors to sites). Insert a command button that, when clicked–on, performs sensitivity analysis of the solution with respect to the total snow volume. Insert an option group that allows the user to choose a report to review.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the final assignment of each sector to the disposal sites, as presented by the heuristic. For each disposal site, report the available capacity. Report the total cost of the assignment.
2. Report the final assignment of each sector to the disposal sites, as presented by the MIP solution. For each disposal site, report the available capacity. Report the total cost of the assignment.
3. The assignment of snow sectors to disposal sites is based on estimates about the total snow volume. Report the total cost of the current assignment in the case that the total snowfall amount changes by $\pm 5\%$, $\pm 10\%$, and $\pm 20\%$.
4. Modify the heuristic so that it considers capacity constraints of the three river sites. Report the total cost of disposing snow as the volume of snow that can be handled by the river sites decreases by 5%, 10%, and 15 %.

Reference

Campbell, J.F. and Langevin, A., "The Snow Disposal Assignment Problem," *Journal of the Operational Research Society*, Vol. 46, pp. 919-929, 1995.

2 Bakery Distribution System

Problem Description

The bakery shop is part of a large food service chain that provides meals to a county school system and other customers. The bakery is responsible for delivering products to over 50 delivery points. The manager of this bakery shop is concerned about delivering on time. The main reasons for this concern are not only the penalty he has to pay for late deliveries, but also maintaining customer satisfaction. There are profits in minimizing the travel distance or maximizing the truck load, but achieving timeliness at low cost does more than maximize profits of the bakery shop, as it also keeps customers satisfied.

The manager of the bakery shop is interested in building a decision support system to identify delivery routes that guarantee on-time delivery. Currently the manager is just guessing and physically testing the delivery routes. This practice makes the addition of new delivery routes expensive.

Excel Spreadsheets

- The manager used the straight-line distance between delivery points as an estimate of the travel time. He used such an estimate because the travel time between delivery points has a very small variance, and travel time is only a small portion of the total time required to complete a route. The following spreadsheet presents the expected travel time as a function of the distance traveled.

Straight-line Distance (miles)	Expected Travel Time (minutes)	Std. Deviation of Travel Time (minutes)
1.2	8	2
2.1	10	2.1
3.5	12	2.4
5.3	14	2.6
7.8	16	3
10	18	3.2

- Loading and unloading times count for a big portion of the total time required to complete a route. These times depend on the volume of products delivered. Create the following spreadsheet, which presents the expected loading and unloading times as a function of the load size.

Load (Not to exceed)	Expected Unload Time (minutes)	Std. Deviation of Unload Time (minutes)
5	11	2
10	15	3
15	22	2
18	25	2

3. Build a spreadsheet that keeps the following information about each customer: name, location (X and Y coordinates), and average weekly demand.
4. Build a spreadsheet that keeps the following historical data about each order: order date, customer name, ordered amount, expected delivery time/date, and actual delivery time/date.

It is likely that the manager would reject unfamiliar or overly complicated solutions. For this reason we suggest the following nearest neighbor heuristic to pick routes and estimate completion times.

The Nearest Neighbor Heuristic

The heuristic works as follows:

1. Calculate the Euclidean distance between the bakery shop and each delivery point, as well as the distance between every two delivery points.
2. Build delivery routes the following way: Starting with the bakery shop, (a) identify the nearest delivery point; (b) calculate the total of travel and delivery time to this point; (c) check if the total time exceeds the promised delivery time. If it does not, continue this process by choosing the nearest neighbor (delivery point). When the total time exceeds the promised delivery time, do not deliver to that node. The required delivery should be performed by another route. Continue building delivery routes until the total demand is satisfied.

Simulation

When solving this problem, one should consider the uncertainties in the time it takes to load/unload the trucks and in travel time. However, the heuristic proposed above assumes that the data is deterministic. For this reason, perform a simulation study to assure the reliability of the results from the heuristic. Run the simulation 100 times. In each run, the load/unload times and travel times are calculated using the information provided in Spreadsheets 1 and 2 as follows: assume that the distribution of the travel time and load/unload operations is normal; the expected travel time and load/unload times present the mean of the distribution, and the corresponding standard deviations present the standard deviation of the distribution. For example, if the load does not exceed 5 units, generate the unload time from a normal distribution with a mean of 11 minutes and a standard deviation of 2 minutes.

The objective of this simulation study is to identify the probability that the schedule we propose (the solution from the heuristic procedure) will guarantee on-time deliveries.

User Interface

1. Build a welcome form.
2. Build a form that allows the user to do the following:
 - a. Enter the total number of customers that are going to receive a delivery.
 - b. Enter the following information about each order: customer name, order size, and expected delivery time and date.

- c. Save the information about orders.
- d. Solve the problem by running the nearest neighbor heuristic and report the final results.
- e. Run the simulation. The user could be prompted to enter the following information related to the simulation: the total number of iterations and the seed. The default number of iterations is 100.

Add command buttons, list boxes, combo boxes, option buttons, subforms, etc. as needed.

3. Build a form that presents the results of the final solution. Allow users to change the order of delivery in the final solution if they choose to do so. Provide the user with the cost and expected delivery time of the new route. Insert a command button that, when clicked on, opens the summary reports created.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the weekly delivery routes (as provided by the heuristic). Present in this report the total delivery costs and the expected delivery times.
2. Build a histogram of the customer demand using historical data.
3. Report a summary of the results from the simulation analysis. Include the following in this summary: the probability that the promised delivery times will be met by the schedule, expected delivery times and costs, etc.
4. Based on historical data about demands and costs, build a report to present the following statistics:
 - a. Actual weekly cost of distribution during the last 52 weeks.
 - b. Average weekly demand.

Reference

Martin, E., "Centralized Bakery Reduces Distribution Costs Using Simulation," *Interfaces*, Vol. 28: 4, pp 38-46, 1998.

3 Blending Problem

Problem Description

A refinery produces three types of gasoline, A, B, and C, which are blends of three crude oils that have a different octane number, cost, and availability. Relevant data about gasoline (final products) and crude oil (ingredients) for the current month is saved in a spreadsheet. Note the following: (a) The current production cost of gasoline is \$4/barrel; (b) The production capacity of the refinery is 14,000 barrels; (c) It is estimated that if we spend \$1 on advertising each type of gas, demand will increase by 10 barrels/day.

The following is the blending rule for this product: If we take x_i barrels of crude oil i ($i = 1, 2, 3$) and pour it all together, then the sulfur percentage of the blend is as follows:

$$\frac{0.5x_1 + 2x_2 + 3x_3}{x_1 + x_2 + x_3}$$

We assume that octane level blends in a similar way.

Below we present a mathematical formulation of this problem:

$$\text{Max} : -\sum_{i=1}^3 p_i c_i + \sum_{j=1}^3 q_j g_j - \sum_{j=1}^3 a_j - f \sum_{j=1}^3 g_j$$

Subject to :

$$\sum_{j=1}^3 x_{ij} = c_i \quad i = 1,2,3$$

$$\sum_{i=1}^3 x_{ij} = g_j \quad j = 1,2,3$$

$$g_j = d_j + \delta a_j \quad j = 1,2,3$$

$$\sum_{j=1}^3 g_j \leq C$$

$$0.5x_{1j} + 2x_{2j} + 3x_{3j} \leq s_j \sum_{i=1}^3 x_{ij} \quad j = 1,2,3$$

$$0.5x_{1j} + 2x_{2j} + 3x_{3j} \leq O_j \sum_{i=1}^3 x_{ij} \quad j = 1,2,3$$

$$x_{ij} \geq 0 \quad i, j = 1,2,3.$$

Where, c_i : presents the number of barrels of crude i purchased; p_i : presents the price paid per barrel of crude i ; g_j : presents the number of barrels of gas j produced; q_j : presents the price per barrel of gas j ; f : presents the unit production cost; d_j : presents the demand for gas j ; x_{ij} : presents the number of barrels of crude i used to produce gas j ; a_j : presents the dollars spent on advertising gas j ; δ : presents the increase in demand for each dollar spend in advertising; O_j : presents the octane number of gas j ; s_j :

presents the sulfur content of gas j ; and C: presents the production capacity of the refinery;

The price of gas, crude oil, and the corresponding demands change every week. Build a decision support system that will enable the company to solve this problem on weekly bases and perform sensitivity analysis with respect to demand and price.

Excel Spreadsheets

1. Build the following Excel spreadsheet, which contains information about the current week's price of gasoline, price of crude oil, demand for gasoline, and supply of crude oil.

Product	Price (\$/barrel)	Sulfur Content	Octane	Demand
A	70	At least 1%	10	3,000
B	60	At least 2%	8	10,000
C	50	At least 1%	6	1,000
Ingredient	Cost (\$/barrel)			Availability
1	45	0.5%	12	5,000
2	35	2%	6	5,000
3	25	3%	8	5,000

2. Build an Excel spreadsheet that keeps historical data about demand for gasoline, price of gasoline, supply of crude oil, and price of crude oil.

User Interface

1. Build a welcome form.
2. The octane number and sulfur content are the only two parameters that do not change. Build a form that allows the user to update prices, ingredient availability, and product demand. Insert a command button that, when clicked on, solves the problem using the Excel solver.
3. Build a form that provides the solution to this problem. Include command buttons. When the user clicks on a button, one of the following actions is performed:
 - a. A sensitivity analysis with respect to demand. Allow demand to vary from -20% to +20% of the current period's demand.
 - b. A sensitivity analysis with respect to gasoline price. Allow price to vary from -20% to +20% of the current period's price.
 - c. A sensitivity analysis with respect to crude oil cost. Allow costs to vary from -20% to +20% of the current period's costs.
 - d. A sensitivity analysis with respect to production capacity. Allow the capacity to vary from -20% to +20% of the current period's capacity.

In this form include a list box that allows the user to select and view the reports generated during the sensitivity analysis.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms

created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Build a summary report of the solution found using the Excel solver.
2. Report a blend of crude oil that maximizes the contribution to profit, assuming an unlimited market demand for the three products.
3. Build a report that summarizes the results of the sensitivity analysis.

Reference

Winston, L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

Mobile Producing Nigeria (MPN) has an offshore oil field consisting of approximately 45 platforms. Each day certain platforms must be visited to regulate flow rates, and a number of technicians must be transported between specific pairs of platforms. The number of technicians to fly from one platform to another depends on the daily operations. All transportation is performed by helicopter. Because the operating costs of a helicopter are high, MPN wants to identify the daily routes of the helicopter that satisfies all the requirements, while minimizing the total distance flown. In scheduling the routes, the specialists realized that another important constraint they have to account for is the helicopter capacity.

The helicopter routing problem is basically a Euclidean traveling salesman problem with additional (precedence and capacity) constraints. Therefore, it is natural to consider existing TSP heuristics in developing a solution for this problem. However, due to capacity constraints, all the sites may not be visited in a single trip. The inputs of this problem are a set of trips (a trip consists of a pickup platform, a drop-off platform, and a load) and a maximum helicopter capacity. The output of this problem is a set of helicopter tours that satisfies all requirements at minimum cost. Below we describe a heuristic that can be used to schedule helicopter routes.

A Heuristic Approach

The heuristic approach consists of two steps described below. The heuristic initially identifies a feasible set of helicopter trips and then improves these trips by interchanging the sequence of the nodes visited.

Step 1: Constructing a tour.

A tour consists of an ordered set of sites visited. Because of the capacity constraints, all sites may not be visited in a single tour. Consider the sites to be visited as the nodes of a network. Pick one of the nodes of the network as the starting node. Obviously this will be the ending node of the tours as well. Check the set of trips (problem input), and among the trips that start at node s , select one. Keep count of the available space you have in the helicopter. Proceed this way until you have scheduled all the trips without violating capacity constraints.

Step 2: Improving the tour.

Given a solution from the first step, improve its quality by considering interchanges of the nodes visited in a particular tour. Keep doing the interchanges until no better solution can be found. After every interchange, update the total cost from the trips.

For more details on developing the heuristic approach described above, see Timlin and Pulleyblank (1992).

Given the problem and the heuristic described above, build a decision support system for MPN. This system would help the company schedule helicopter trips in a cost efficient manner.

Excel Spreadsheets

1. Build a spreadsheet that presents the following information about each helicopter: the maximum capacity and operating costs such as fuel consumption per km (in \$).
2. This spreadsheet presents a distance matrix among the 45 platforms.

User Interface

1. Build a welcome form.
2. Build a form that allows the user to do the following:
 - a. Input the total number of trips to be scheduled.
 - b. Identify the pickup platform, the drop-off platform, the pickup load, and the drop-off load for each trip. Here we describe one way to complete this task: Insert a command button that, when clicked on, submits the information about the total number of trips. Upon submission, four list boxes are created. The first list box contains the names of the 45 platforms. The second and third list boxes present the pickup and drop-off platforms, respectively. The size of these list boxes depends on the total number of trips. Initially, both list boxes are empty. The user selects a platform from the first list box. Insert two command buttons, one that adds the selected platform to the list of pickup platforms and another that adds the selected platform to the list of drop-off platforms. The user manually enters the load sizes in the fourth list box.
 - c. Update the information about helicopters.
 - d. Update the information about platforms.
 - e. Run the heuristic and presents the corresponding solution. Details about solution presentation are given in form number 3, which we describe below.
 - f. Open the reports presented below.

Use text boxes, list boxes, command buttons, subforms, etc. as needed.

3. Build a form that presents the following:
 - a. The final solution from the heuristic. For each tour, present the platforms visited and the load picked up and dropped off.
 - b. A graphical representation of the final solution.

In this form, insert a command button that, when clicked on, animates the movement (adds/deletes arcs in the network of platforms) of the helicopter based on the results from the final solution. Insert another command button that, when clicked on, performs a sensitivity analysis of the final solution with respect to the helicopter operating costs. For the purpose of this analysis allow the operating costs to vary from -20% to +20% of the current operating costs. Insert another command button that, when clicked on, performs a sensitivity analysis of the final solution with respect to the helicopter capacity. For the purpose of this analysis allow the capacity to vary from -20% to +20% of the current helicopter capacity.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the final solution of the heuristic. Include the following in this report: the total number of helicopters needed and corresponding capacity; the pickup and drop off platform, load carried, and available capacity for each tour.
2. Report the results from the sensitivity analysis.

Reference

Timlin, M., and Pulleyblank, W., "Precedence Constraints Routing and Helicopter Scheduling: Heuristic Design." *Interfaces*, Vol. 22:3, pp: 100-111, 1992.

Problem Description

The goal of the diet problem is to find the cheapest combination of foods that will satisfy all the daily nutritional requirements of a person. This problem has often been used in Operations Research textbooks to illustrate the use of linear programming to solve real-life types of problems. The corresponding linear programming minimizes the cost of food, subject to nutritional requirements. Below, we give the linear programming formulation of the problem. In this formulation we include constraints that regulate the number of calories and the amount of vitamins, minerals, fats, sodium, and cholesterol in the diet.

Optimization Model

Let *Food* and *Nutrition* be sets of foods and nutrition we consider in this problem. Assume that there are f different types of food in *Food* and n different types of *Nutrition*. The following is a linear programming formulation of the problem:

$$\text{Min} : \sum_{j=1}^f c_j x_j$$

Subject to :

$$nlb_i < \sum_{j=1}^f nt_{ij} x_j \leq nub_i \quad i = 1, \dots, n$$

$$flb_j < x_j \leq fub_j \quad j = 1, \dots, f.$$

where: c_j is the cost of food j ; x_j is the amount of food j consumed; nt_{ij} is the amount of nutrition i in food j ; nlb_i is the minimum amount of nutrition i required per day; nub_i is the maximum amount of nutrition i allowed to be consumed per day; flb_j is the minimum amount of food j desired per day; fub_j is the maximum amount of food j desired per day.

Excel Spreadsheets

1. Build a spreadsheet that presents the following data about different types of food: price, weight, calories, amount of cholesterol, and vitamin content per serving.
2. Build a spreadsheet that presents data about the minimum and maximum amounts of nutrition i required per day for a healthy diet.
3. Build a spreadsheet that presents historical data about the food consumed by the user on daily bases: total amount of calories consumed and total amount of nutrition taken by nutrition type.

User Interface

1. Build a welcome form.
2. Build a form that allows the user to do the following:

- a. Select the type of food and the corresponding amount consumed. Here we describe one way to complete this task: Insert two list boxes in this form. The first list box presents a list of food items. The second list box, which is initially empty, presents the following data: the names of food items selected by the user and the corresponding minimum and maximum amounts desired per day. The user selects the name of a particular food item from the first list box and then clicks on a command button that adds the selected food in the second list box. The user enters manually the minimum and maximum amounts desired for each food item.
- b. Update the data presented in Spreadsheets 1 and 2.
- c. Add data about the food consumed in a day to Spreadsheet 3.
- d. Solve the problem and open Form 3, described below, which presents the solution of this problem.

Use text boxes, list boxes, command buttons, subforms, etc. as needed.

3. Build a form that does the following:
 - a. Presents details about the final solution of this problem, such as the total price of the diet, the amount of food j consumed ($j = 1, \dots, f$), the total amount of calories gained, and the total amount of nutrition taken per nutrition type. The user is allowed to update this information (if the user actually consumes different food types and quantities from what was proposed by the solution). Insert a command button that, when clicked on, does the following: re-calculates the total price of the diet, the amount of food consumed, the total amount of calories gained, etc; and stores the information in Spreadsheet 3.
 - b. Enables the user to perform a sensitivity analysis with respect to changes in the price of food.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Build a report that presents details about the final solution to the diet problem.
2. Build a report that presents details about the sensitivity analysis.
3. Plot the following:
 - a. The total amount of calories consumed daily.
 - b. The total amount of money spent daily.
 - c. The total amount of a particular nutrition type (e.g., vitamin C) taken daily.

Reference

Winston, L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

Central Lechera Asturiana (CLAS) Company is one of the largest distributors of dairy products in the north part of Spain. The company has 700 employees and distributors throughout the country and processes about 1.3 million liters of milk every day. Its products are distributed through a network of 19 distributors spread around the country, each serving a specific area. Each distributor supervises a number of team leaders (sales promoters). Each team leader coordinates the activities of five to ten vendors. Each vendor deals directly with the clients. A vendor works with 5,000 to 15,000 clients, which are mainly retailers, supermarkets, and hypermarkets.

Designing a logistic network for distributing the dairy products produced by CLAS is a challenging problem because the company is large and has many clients located in different regions. There are a large number of possible assignments of shops to vendors and many possible daily delivery routes. CLAS and other companies that offer dairy products are in need of efficient delivery routes because dairy products sell high-volume, low-margin and have limited shelf life. A poor supply chain design would generate additional costs because of overtime and extra miles traveled to satisfy demand in a timely manner.

To help the company manage the distribution of clients among the promoters and vendors, you are assigned to build a decision support tool. The system should help the managers to make the following decisions: the number of times a client should be visited, the assignment of clients to vendors, and the routes a vendor should perform in a particular day. This problem is complex; therefore, we propose a hierarchical decomposition-based heuristic to solve it. This procedure solves the problem by breaking it into simpler sub-problems, solving the sub-problems, and using the solution of sub-problems to generate a solution to the overall problem. Below we describe the steps of this heuristic. For more details see Adenso-Diaz et al. (1998).

A Heuristic Approach

- Step 1.** Determine the total number of visits to each client per week. This decision is based on the buying potential of each client, overall capacity of the logistic network, etc.
- Step 2.** Distribute the clients among sales promoters. Each distributor has five to ten vendor teams, each headed by a promoter. In doing the assignment of clients to sales promoters, you should consider their geographical location, number of loads demanded by each client, etc.
- Step 3.** Distribute the clients assigned to a sales promoter among the team's vendors seeking geographically balanced assignments that are reasonable in terms of number of visits, distance traveled, number of loads transported, and delivery time.
- Step 4.** At this point, a vendor knows which clients it will serve (output of **Step 3**) and how many times a week they are to be visited (output of **Step 1**). Given this information, the system should provide to each client a schedule of visits that satisfies these conditions: the number of visits to a client coincides with the

desired number of visits; and the vendor visits a similar number of shops every day (the number of clients visited in a day is balanced throughout the week).

- Step 5.** The system should build a report that lists the clients in the order in which they are visited each day of the week. In creating such a list, consider that the clients should be visited within a time window, and the number of miles traveled should be minimized.

Note: In **Step 2** we determine the best distribution of clients among sales promoters. The problem presented in this step, as well as in **Steps 3** and **4**, is a special version of the clustering problem. These problems are quite challenging. We suggest that the students use simple heuristics to perform these tasks. For example, start with an initial assignment, and then make feasible modifications to the solution (such as passing clients from one promoter to the other) to improve the results. This procedure continues until no better exchange can be identified.

Excel Spreadsheets

1. Build a spreadsheet that presents the following information about each client: name, X and Y coordinates of client location, average weekly demand, etc.
2. Build a spreadsheet that keeps historical data of client orders.
3. Build a spreadsheet that keeps the following information about each vendor: name, X and Y coordinates of vendor location, etc.
4. Build a distance matrix from each client to each vendor.
5. Build a spreadsheet that keeps information about each sales promoter: name, X and Y coordinates of sales promoter location, etc.
6. Build a spreadsheet that keeps information about each distributor: name, X and Y coordinates of distributor location, etc.

User Interface

1. Build a welcome form.
2. Build a form that will enable the user to add/delete/update the data used in this problem. The following are instructions for creating this form.
 - a. *Update the average weekly demand of a client.* To complete this task, insert a combo box that lists the name of each client. Insert a text box where the user can type the value of the average weekly demand. Insert a command button that, when clicked on, submits the new value of demand for the selected client.
 - b. *Delete the data about a client.* To complete this task, insert a combo box that lists the name of each client. Insert a command button that, when clicked on, deletes the information about the selected client from the database.
 - c. *Add a new client to the database.* To complete this task, insert text boxes where the user can type in the data required about the new client (such as name, location, etc.). Insert a command button that, when clicked on, adds the data about the new client in the database.

Insert a command button that, when clicked on, solves the problem using the heuristic approach and opens Form 3, described below.

3. Build a form that presents the results from solving this problem and allows the user to perform a sensitivity analysis with respect to problem data. The following are some instructions for creating this form.
 - a. Use a list box to present the sales promoters assigned to each distributor, the vendors assigned to each sales promoter, and the clients assigned to each vendor. Provide a graphical representation of the solution, where nodes represent vendors and clients, and arcs represent the assignment of clients to vendors.
 - b. Use a list box to present the names of the clients. The user should be able to select more than one client at a time from this list. Insert two text boxes where the user can type a lower and an upper bound in the percentage change of clients' demand (e.g., -20% to +20% changes from the current level). Insert a command button that, when clicked on, results in the following: a sensitivity analysis is performed with respect to demand changes of the selected clients; and a report is opened that contains the results of this analysis.
 - c. Insert a command button that, when clicked on, provides the following statistics about the data collected: overall (clients) average demand, standard deviation of demand, and maximum and minimum demand.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Build a report that presents details about the final solution of the heuristic. The report should include the following information about each client: name, vendor's name that serves the client, timing of each visit, quantity ordered, etc.
2. Build a report that presents details about the sensitivity analysis with respect to demand.

Reference

Adenso-Diaz B., Gonzales, M., and Garcia, E., "A Hierarchical Approach to Managing Dairy Routing." *Interfaces*, Vol: 28:2, 1998.

7 The News Vendor Problem - I

Problem Description

“Java Hot” is a small coffee shop in Gainesville. The shop is owned by Jane, who enjoys serving coffee and baked goods to her clients. Because the environment of the shop is very welcoming and Jane is responsive to her customers, the business has grown fast. However, Jane is concerned about the financial viability of the store for the following reason. The major products that she sells in her shop are coffee and bagels. Every morning Jane buys bagels from a local bakery for \$0.25 each and sells them for \$1.00 each. Since she cannot return unsold bagels or sell them the next day, she wants to know how many bagels to order every day so that she will not face situations such as having too many leftovers or not having enough bagels. As she has already identified her problem, she is trying to figure out what to do.

Jane’s problem can be formulated as the News Vendor’s Problem. We want to build a decision support system for Jane to help her in making the right decision. Below we propose an optimization approach and a simulation model.

The Optimization Approach

The optimization approach suggests that Jane orders every time the quantity q^* that satisfies the following:

$$F(q^*) = \frac{c_u}{c_o + c_u}. \quad (1)$$

Where, $F()$ is the demand distribution function; c_u (\$1.00) is the unit under-stocking cost; and c_o is the unit overstocking cost (\$0.25). Jane kept record of the number of bagels sold in the last 50 days (see Spreadsheet 1).

The Simulation Model

1. Identify the distribution of demand for bagels ($F(q)$) using the data in Spreadsheet 1. Calculate the corresponding mean and standard deviation.
2. Run the following simulation 50 times. In each simulation run, do the following:
 - a. Randomly generate daily demand from distribution $F(q)$ with mean and standard deviation as calculated above.
 - b. Calculate costs/profits assuming that Jane ordered an amount q^* of bagels using Equation (1).

Present the average costs/profits over the 50 runs.

3. Perform a scenario analysis. In each scenario, either change the distribution function $F(q)$ (e.g., normal, uniform, exponential, etc.) keeping the mean and standard deviation the same, or change the standard deviation and mean of $F(q)$. For each

scenario, compare the demand generated to historical data, and calculate the corresponding total costs/profits. The information from the scenario analysis will help Jane decide how many bagels to order every day.

Excel Spreadsheets

1. The following spreadsheet presents the daily demand for bagels during the last 50 days.

Day	Sales	Day	Sales	Day	Sales	Day	Sales	Day	Sales
1	100	11	106	21	98	31	96	41	94
2	99	12	95	22	113	32	97	42	93
3	106	13	102	23	100	33	108	43	97
4	102	14	92	24	106	34	101	44	102
5	97	15	99	25	97	35	99	45	100
6	109	16	91	26	97	36	91	46	84
7	88	17	105	27	92	37	118	47	92
8	98	18	108	28	107	38	106	48	87
9	109	19	101	29	104	39	107	49	87
10	104	20	98	30	115	40	101	50	111

User Interface

1. Build a welcome form.
2. Build a form that will enable Jane to add data to the database, view the results of the optimization and simulation models, and view the results of the scenario analysis. The form is comprised of the following:
 - a. A text box that allows Jane to type in the daily demand for bagels.
 - b. A command button that, when clicked –on, adds the daily demand in Spreadsheet 1.
 - c. A command button that, when clicked on, performs statistical analysis of the input data (Spreadsheet 1) and opens the corresponding reports.
 - d. A frame titled “Order Quantity” that includes the following: a combo box that allows the user to select a distribution for demand; and a text box that, upon the selection of demand distribution, presents the optimal order quantity.
 - e. A frame titled “Simulation Study” that includes the following: (i) a combo box that allows the user to select a distribution for demand; (ii) three text boxes where the user can type in the mean and standard deviation of demand and the total number of simulation runs. The default values in the text boxes are the mean and standard deviation of demand calculated using the data in Spreadsheet 1. The default number of simulation runs is 100. In each simulation run 50 demands are randomly generated; (iii) a command button that, when clicked on, runs the simulation and presents the total costs/profits of the coffee shop.

Reports

1. Build a report that presents details about the distribution of demand for bagels. Prepare a histogram using the data in Spreadsheet 1. Provide the following statistics about the demand: mean, standard deviation, median, etc.
2. Graph the relationship between order quantity ($q^* = 0, \dots, \text{maximum demand}$) and the total expected costs/profits. This graph should help Jane (visually) to identify the order quantity that gives the minimum total cost.
3. Based on the information about demand distribution, report the optimal order quantity using the equation presented above.
4. Present a summary report of the results from the simulation runs.

Reference

- Nahmias, S., "Production and Operations Analysis." *Irwin Book Team*, 3rd Ed., 1997.
- Winston, L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

8

Machine Scheduling Problem

Problem Description

A machine shop produces products *A* and *B*. Product *A* is available in two options, A_1 and A_2 , while product *B* is available in three options, B_1 , B_2 , and B_3 . The shop produces these products using an appropriate combination of three machines. The machines can be used in any order. The shop has signed contracts with clients that demand Q_1 units of item *A* and Q_2 units of item *B* per week. The objective is to determine the most economical way of satisfying the contracts. The data is summed up in Spreadsheet 1.

This problem is formulated as a linear programming problem. The Excel solver can be used to solve this linear program. Let x_1, \dots, x_5 be, respectively, the number of units of product $A_1, A_2, B_1, B_2,$ and B_3 produced in one week. Below we present a linear programming formulation of the problem:

$$\min : 2x_1 + 2.5x_2 + 5x_3 + 4x_4 + 4x_5$$

Subject to :

$$x_1 + x_2 = Q_1$$

$$x_3 + x_4 + x_5 = Q_2$$

$$0.5x_1 + 0.4x_3 + 0.4x_4 \leq 38$$

$$0.4x_2 + 0.3x_3 + 0.6x_5 \leq 31$$

$$0.2x_1 + 0.2x_2 + 0.3x_4 + 0.3x_5 \leq 34$$

$$x_i \geq 0, i = 1, \dots, 5$$

The first two constraints are the production requirements constraints. The next three constraints are the machine time requirements constraints.

Excel Spreadsheet

1. Build the following spreadsheet.

Product	Option	Unit production time on machine			Unit Cost
		1	2	3	
A	1	0.5	-	0.2	2
	2	-	0.4	0.2	2.5
B	1	0.4	0.3	-	5
	2	0.4	-	0.3	4
	3	-	0.6	0.3	4
Hrs. Available		38	31	34	

User Interface

1. Build a welcome form.
2. Build a form that enables the user to add/delete/update the data used in this problem. The following are instructions for creating this form.

- a. Insert two text boxes where the user can type in the amount of Q1 and Q2 that should be produced in the current week.
 - b. Insert a command button that, when clicked on, opens Spreadsheet 1 and allows the user to update the information about costs, machine hours available, etc.
 - c. Insert a command button that, when clicked on, solves the problem using the Excel solver and opens Form 3, described below.
3. Build a form that presents the solution of the machine scheduling problem. In the same form, provide tools to enable the user to perform a sensitivity analysis with respect to problem data. Insert a frame that has four option buttons, one for each problem data: unit cost, hours available, quantity required, and production time. Insert a command button that, when clicked on, performs the sensitivity analysis and opens the corresponding reports.

Reports

1. Build a report that presents the weekly production schedule that minimizes costs, given the weekly requirements for the two products. Report the utilization of each machine.
2. Build a report that presents the results from the sensitivity analysis.

Reference

Winston, L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

9 Production Management at Still Mills

Problem Description

Improving responsiveness and reliability of order fulfillment has become a challenge for integrated steel mills (ISMs) because of the increase of market competition in recent years. As a result, ISMs have shifted from using pure make-to-order systems to hybrid make-to-stock/make-to-order systems. This change will enable companies to match certain customer orders using existing semi-finished inventories, and, as a result, reduce the time it takes to fulfill the orders. However, choosing which semi-finished products should be made-to-stock is a difficult task. ISMs carry several thousand designs of steel slabs. It seems reasonable to choose for make-to-stock only the slabs that have a high probability to match customer orders. We want to build a decision support system to help ISMs with these types of decisions. Below we present an integer programming formulation of this problem.

$$\begin{aligned} \max : & \sum_{i=1}^{|C|} \sum_{j=1}^{|S|} r_{ij} y_{ij} \\ \text{Subject to :} & \\ & \sum_{j=1}^{|S|} y_{ij} \leq 1 \quad i = 1, \dots, |C| \\ & \sum_{j=1}^{|S|} x_j \leq k \\ & y_{ij} \leq x_j \leq 1 \quad i = 1, \dots, |C|; j = 1, \dots, |S| \\ & y_{ij}, x_j \in \{0, 1\} \quad i = 1, \dots, |C|; j = 1, \dots, |S|. \end{aligned}$$

Where, k is the maximum number of designs that are to be made-to-stock; S is the set of potential slab designs; C is the set of orders; r_{ij} is the benefit of having a slab j to cover an order i ; x_j takes the value 1 if slab $j \in S$ is chosen, and 0 otherwise; y_{ij} takes the value 1 if order i is assigned to slab j , and 0 otherwise.

Given r_{ij} , we want to choose at most k designs that maximize the total reward. For each feasible application of slab design to an order, the ISM earns a non-negative reward equal to the size of the order. For the non-feasible allocations, the ISM gets no reward.

Excel Spreadsheets

1. Build a spreadsheet that presents historical information about the orders received. For each order, record the date, product ordered, and order quantity.
2. Build a spreadsheet that keeps information about the current inventory of still slabs.
3. Build a spreadsheet that presents the benefit of having a particular slab assigned to a particular order.

User Interface

1. Build a welcome form.

2. Build a form that includes the following controls:
 - a. A text box where the user can type in the maximum number of designs that are to be made-to-order (k).
 - b. A command button that, when clicked on, opens Spreadsheet 1 and allows the user to add a new order or update/delete an existing order.
 - c. Insert a command button that, when clicked on, solves the integer programming problem using the Excel solver and provides a solution.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report details about the final solution of this optimization problem.
2. Report the sensitivity of the final solution with respect to changes in the maximum number of designs that should be made-to-order.

Reference

Denton, B., Gupta, D., and Jawahir, K., "Managing Increasing Product Variety at Integrated Steel Mills." *Interfaces*, Vol. 33:2, pp. 41-53, 2003.

Problem Description

The local government of Gainesville is concerned about finding a location to build a new fire station. They believe the station should be located in such a way that allows the firefighters to respond in a timely manner to emergencies. Facts that influence this decision are as follows: (a) the risk of fire is not the same in all areas. Industrial parts of the town are more vulnerable to fire; (b) population is not spread equally around the town, and, as a result, there are parts of the town that are more populated than others. It has been shown that the frequency of incidents is higher in the most populated areas.

We would like to develop a decision support system that will help the local government to decide about the location for the new fire station. We present a mathematical formulation of this problem.

Optimization Model

The objective of this problem is to determine a location for the fire station that will keep the response time under a pre-determined maximum response time. It seems logical to locate the fire station close to the most populated areas of the town and close to the areas that are vulnerable to fire. The input data for this problem would be the location of households and businesses. However, since it is difficult to collect and manage the data for each household, we group the population into N population hubs. Each population hub is represented by a set of co-ordinates. Let P_i be the population of hub i ($i = 1, \dots, |N|$). The maximum response time of hub i is $T_{\max-i}$. There are F possible locations in the town that are feasible for building the fire station ($F \subseteq N$). T_{if} is the time it takes to reach the population of hub i from location f . We use Euclidean distances to measure the distance between every two locations. We assume that the average travel time between two locations is proportional to the distance between them. We assume that the costs of building and maintaining the fire station are of secondary concern in this problem, and therefore we do not include them in this formulation. The decision variables are the binary variables z_f ($f = 1, \dots, |F|$) that take the value 1 if the fire station is located at location f , and 0 otherwise. The minimum response time to reach population i from a prospective fire station location f is given by $t_i = \min \{T_{if} \mid f \in F \text{ and } z_f = 1\}$.

$$\begin{aligned} \min : & \sum_{i \in N} t_i \\ \text{Subject to :} & \\ & t_i = \min_{f \in F, z_f = 1} \{T_{if}\} \leq T_{\max-i} \quad i = 1, \dots, |N| \\ & \sum_{f \in F} z_f = k \\ & z_f \in \{0, 1\} \quad f = 1, \dots, |F|. \end{aligned}$$

Note the following: In our problem we state that the local government is planning to build one fire station ($k = 1$). In the case that the problem is not feasible for $k = 1$ (the maximum response time is not satisfied), the user can increase k until a feasible solution to this problem is found.

Excel Spreadsheets

1. Build a spreadsheet that provides the following information about each population hub: co-ordinates of its location, corresponding population, frequency of incidents, and maximum required response time.
2. Build a spreadsheet that presents the set of all feasible locations for the fire station.
3. Build a spreadsheet that presents the distance among potential fire station locations and population hubs.

User Interface

1. Build a welcome form.
2. Build a form that enables the user to add/delete/update the data used in this problem. The following are instructions for creating this form.
 - a. Insert a text box where the user can type in the total number of fire stations to be built.
 - b. Insert two text boxes where the user can choose a lower bound (lb_k) and an upper bound (ub_k) of the values that k takes. In this case the problem will be solved ($ub_k - lb_k + 1$) times. Provide a table that presents for each value of k the corresponding total response time.
 - c. Insert a command button(s) that, when clicked on, opens one of the spreadsheets of the database and allows the user to add/update/delete the corresponding information.
 - d. Insert a command button that, when clicked on, solves the problem using the Excel solver.
3. Build a form that presents the following details about the final solution.
 - a. Provide a graphical presentation of the final solution. Population hubs and fire station locations are the nodes of the graph. The arcs of the graph represent the assignment of population hubs to fire stations.
 - b. Insert a frame that has two option buttons. The option buttons allow the user to choose the type of sensitivity analysis to be performed, a sensitivity analysis with respect to k or T_{Max} . Insert a command button that, when clicked on, performs the sensitivity analysis and returns the results.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Build a report that presents details about the final solution. The report should indicate the best location(s) for the fire station and the corresponding response time.
2. Build a report that provides details about the sensitivity analysis.

Reference

Winston, L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

Hospitals keep records of their inpatients and outpatients. The Outdoor Patient Division (OPD) keeps detailed records about the patients and their visits. Hospitals are interested in using this information to support their decisions about recruiting personnel (doctors and nurses) and preparing their weekly working schedule. The aim of this project is to build a decision support system that would help the hospitals with data storage and data analysis. This system will enable the managers to answer the following questions:

1. Is the patient waiting time and service in a particular department within the desired limits?
2. Is the workload of doctors and nurses of a particular department within the required limits?
3. How many doctors and nurses should be on duty in a particular department at different times and days of the week in order to have the waiting time of patients within a desired range?

Simulation

A simulation study is necessary to determine the total number of doctors and nurses that should be hired in a particular department. We assume that during a particular visit the patient is served by a single department. Depending on the diagnoses, during a visit the patient may see more than one doctor and one nurse. If the doctor/nurse is not available upon patient arrival, the patient will wait.

Using the data from the spreadsheets, calculate the following for each department for each hour of the day and each day of the week: the inter-arrival time of outpatients, the average number of doctors and nurses on duty, and the average service time for each doctor. Identify the distribution of the service time. Run the simulation model using the parameters. During a simulation run, do the following: (a) generate patient arrival; (b) assign patients to a particular department; (c) generate the number of nurses and doctors needed to serve the patient; (d) if the doctor/nurse is available, assign the patient to the available doctor/nurse; otherwise the patient will wait to be assigned; (e) generate service time of the patient with each doctor and nurse.

Use the results from this simulation to calculate the following: average, maximum, and minimum patient waiting time and service time and total number of patients waiting during a particular time of day, each day of the week, etc.

Perform a sensitivity analysis with respect to the total number of doctors and nurses. For each department, increase/decrease the total number of doctors and nurses available at different hours of the day, all days of the week. Re-run the simulation until the patient waiting time and staff overload are within the requirements of the hospital. This analysis will help us identify the total number of doctors and nurses needed in each department. The results may indicate that, instead of hiring, we could transfer nurses from the over-staffed departments to the under-staffed departments.

Summarize the main results and report statistics collected from the simulation study. To learn more about simulation, input, and output analysis, see Winston (1994).

Excel Spreadsheets

1. Build a spreadsheet that keeps the following information about the departments: name of the department, total number of doctors and nurses, and a list of the services provided.
2. Build a spreadsheet that keeps the following information about doctors: name, department, specialty, daily working schedule, etc.
3. Build a spreadsheet that keeps the following information about nurses: name, department, daily working schedule, etc.
4. Build a spreadsheet that keeps the following information about patients: name, birthday, address, telephone number, family doctor, etc.
5. Build a spreadsheet that presents the following information about patients' visits: patient name, scheduled visit day and time, arrival time, waiting time, duration of the visit, services provided, departments visited, etc.

User Interface

1. Build a welcome form.
2. Build a form that enables the user to add/delete/update the database of this problem. For this purpose, insert a data entry frame. Include five option buttons in this frame. The option buttons enable the user to choose to open each spreadsheet and add/delete/update their information. Provide validation rules to ensure the validity of data entered.
3. Build a form that allows the user to search the database and collect the following information: the total number of doctors and nurses working in a department at a particular time and day of the week; the total number of patients waiting to be served in a department at a particular time and day of the week; the peak hours for each department; the department with the highest patient waiting time; etc.
4. Build a form that allows the user to perform statistical analyses of certain systems performance measures. For this purpose, insert a combo box that enables the user to select a systems performance measure to investigate (e.g., patient waiting time, patient service time, average number of working hours for doctors, average number of working hours for nurses, etc). Insert a command button that, when clicked on, performs a statistical analysis of the performance measure selected and reports the corresponding results: mean, standard deviation, 95% confidence interval, etc.
5. Build a form that enables the user to update the parameters required during the simulation study. Insert two text boxes to allow the user to enter the total number of entities (patients) to be generated during a simulation run and to enter the length of the warm-up period. Insert a command button that, when clicked on, performs the simulation study and opens the report that presents the corresponding results.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Prepare a summary report that presents the following statistics about the hospital: the total number of doctors and nurses, the total number of departments, the average patient service time, the average number of patients served per day, the average number of working hours per day, etc.
2. Prepare a summary report for each department that presents the following statistics: the average patient service time, the average daily workload of doctors and nurses, etc.
3. Prepare a report that contains the following information about each department: the total number of doctors and nurses on duty every day of the week and the average patient waiting time at a particular hour of the day for each day of the week.
4. Report a list of the departments that do not satisfy the requirements set by the hospital in terms of patient waiting time, patient service time, and staff workload.
5. Prepare a report that summarizes the results from the simulation analysis. For each department, report the total number of doctors and nurses who should be on duty at a particular hour of the day for each day of the week such that the total waiting time of patients and staff overload is within the required standards.

Reference

Winston, L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

Who does not like cookies? While cookie preparation is more of an art, we can use science to help polish this art and bake better cookies. Maggie is fond of cookies and every chance she gets, she bakes some. Recently, she has been using a new recipe to bake chocolate chip cookies. Unfortunately, she is not satisfied with their quality, but she does not want to give up. She is pretty confident about the quality and quantity of all but two ingredients that are used in preparing the cookies, sugar and salt. She has been using different brands of sugar and salt, she has changed the quantity used, but still she is not happy with the results.

As an engineer, Maggie wants to build a decision support system to help improve the recipe. She baked cookies using the following four different combinations of sugar and salt: 1 cup of white sugar and no salt; 1 cup of white sugar and 1 teaspoon of salt; 1 cup of regular sugar and no salt; 1 cup of regular sugar and 1 teaspoon of salt. She had her friends try the cookies and evaluate their taste on a scale of 1 to 10, with 1 being the worst and 10 the best. Maggie used the following model to collect and analyze the data from the surveys:

Model

In order to identify how sugar or salt (individually) or their interaction influences the taste of cookies, we will perform an ANOVA analysis. Below we describe the main steps of this analysis:

1. Decide on the total number of replications.
2. In the problem description we identified four types of experiments to be performed (Experiment 1: use 1 cup of white sugar and no salt; etc). Number the experiments from one to four. For each replication, randomly generate a sequence of experiments to be performed. For example, during Replication 1, we first bake cookies using Recipe 1, then we bake again using Recipes 4, 2, and 3. During Replication 2, we bake cookies using Recipes 4, then 3, 2 and 1; etc. Assign a code to each batch of cookies baked in each experiment of each replication performed.
3. Build a spreadsheet that presents the feedback from the surveys for each batch of cookies prepared.
4. Use the data analysis tools in Excel to run a Two Factor ANOVA with replicates.
5. Use the results from the ANOVA analysis to identify the F-Statistic and degrees of freedom for the two factors and their interaction. Use the degrees of freedom and the user-defined confidence level to determine the significance of each factor and their interaction to the taste of chocolate chip cookies.
6. Graph the average responses for each factor combination. The significant interactions are the ones that lack parallelism of the lines.
7. For the significant factors, perform the Tukey or Fischer LSD tests.
8. Check the adequacy of the model using the normal probability plots, the run order plots, etc.

The results of the data analyses enable us to identify the factors that influence the taste of cookies. One can choose to repeat the procedure using different levels of the factors that are of concern. For more details about experimental design and ANOVA analysis, see Montgomery (1997).

Excel Spreadsheets

1. Build a spreadsheet that presents the following information about each batch of cookies baked: batch number, replication number, and experiment number.
2. Build a spreadsheet that presents the results of the survey.

User Interface

1. Build a welcome form.
2. Build a form that allows the user to update the data used to perform the ANOVA analyses. For this purpose use the following:
 - a. A combo box that enables the user to choose the number of levels for each factor. In the problem description we identified two factors to be observed during the experiments. For each factor we identified two levels (e.g., no salt or 1 teaspoon of salt). Excel can only handle ANOVA analysis with two factors, but there is no limit on the number of levels for each factor.
 - b. A text box where the user can type in the total number of replications.
 - c. Command buttons that, when clicked on, open the Excel spreadsheets and allow the user to add/update/delete the data on these spreadsheets.
 - d. A frame that has two option buttons. The option buttons enable the user to choose the type of test (Tukey or Fischer LSD) to be performed.
 - e. A command button that, when clicked on, performs the ANOVA analysis and presents the corresponding results.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the results from the ANOVA analysis in a tabular form.
2. Present the factors and iterations that have been identified as significant for this experiment. For each factor (or iteration) report the corresponding p value.
3. Graph the average responses for each factor combination.
4. Prepare box plots and scatter diagrams using the data collected for each factor.
5. Present the factors that have most impacted the taste of the chocolate chip cookies. Use the results from the ANOVA analysis and the Tukey and Fischer LSD tests to identify the corresponding optimal levels.

Reference

Montgomery, C.D., "Design and Analysis of Experiments", *John Wiley & Sons*, 4th Ed., 1997.

13 *Quality Control Support System*

Problem Description

PuttOn Co. is a company that designs and produces various clothing lines. The company uses the Internet and a catalogue to advertise and sell its products. Every season the company offers new designs and prepares a catalogue featuring these items. Recently, the management has noticed an increase in the returns of a particular style of sweater that was sold during the Christmas season. The management is concerned about this phenomenon and wants to identify the problem areas and repair the mistakes.

The return policy requires that the customer indicate the reasons for returning the product. The customers are provided a number of reasons to choose from. This data will then be used to identify the reasons for the quality failure. The aim of this project is to build a decision support system that will facilitate the data analysis.

Excel Spreadsheets

- The following spreadsheet lists the reasons for returning the product.

1 Large	2 Small	3 Quality
10 overall	20 overall	30 seem defective
11 chest/bust	21 chest	31 material defective
12 waist	22 waist	32 assembly defective
13 seat/hips	23 seat	33 instruction missing
14 rise	24 hips	34 color does not match
15 too long	25 too short	35 finish not acceptable
16 too wide	26 too narrow	36 parts missing
17 too loose	27 too tight	37 does not work

- Build a spreadsheet that presents the data about each return. Below we present a sample of the data collected from the returned sweaters.

Cust. ID	Primary Cause	Secondary Cause	Tertiary Cause	Procurement
1	12	34	37	Internet
4	21	22	25	Catalogue
5	16	15	35	Christmas catalogue

User Interface

- Build a welcome form.
- Build a data entry form for Spreadsheets 1 and 2.
- Build a form that enables the user to update the data in Spreadsheet 2. Include five combo boxes to allow the user to choose which field of the table to update. Include a text box where the user enters the identification number of the customer whose data will be updated. Include a text box where the user types the new value. Insert a

command button that, when clicked on, updates the data in Spreadsheet 2 using the information given by the user.

4. Build a form that enables the user to analyze the data in Spreadsheet 2. In this form, include the following:
 - a. A frame that has two option buttons. One option button allows the user to sort the information in Spreadsheet 2 based on the primary cause of return. Within each group the information is sorted based on the secondary cause of return and then based on the tertiary cause of return. The second option button allows the user to sort the information in Spreadsheet 2 based on the procurement mode. Within each group sort the information by primary, secondary, and tertiary causes of return. Include a command button that, when clicked on, sorts the information in Spreadsheet 2 based on the user's request and opens the spreadsheet.
 - b. A frame that includes a number of option buttons. Option buttons allow the user to view the following statistics: the five reasons that were selected most of the time as a primary reason for return; the five reasons that were selected the least number of times as a primary reason for return; the five reasons that were selected most of the time as a secondary reason for return; the five reasons that were selected most of the time as the tertiary reason for return, etc. Insert a command button that, when clicked on, identifies the statistics described above and returns the results.
 - c. A command button that, when clicked on, performs a Pareto analysis using the algorithm described below and presents the corresponding results. Pareto analysis is used very often to identify product/process quality problems. This analysis indicates that 20% of the reasons cause 80% of the quality problems. In here we describe the steps one should follow to complete the Pareto analysis: (i) calculate the total number of returns; (ii) for each cause of return, calculate the total number of customers that have selected it as the primary cause of return; (iii) for each cause of return, calculate the percentage of returns, sort the information in descending order of the percentage return, and calculate the cumulative percentages; (iv) repeat steps (i) to (iii) for the secondary and tertiary causes of return.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Build a summary report with the results from the Pareto analysis.
2. Build a histogram that presents the frequency of returns for each reason of return. In the same graph include the cumulative frequency curve.
3. Report the top three problem areas that are causing the return of sweaters.

Reference

Montgomery, D.C., Runger, G.C., "Applied Statistics and Probability for Engineers", 3rd Ed., *John Wiley & Sons*, 2003.

Problem Description

Manufacturing companies deploy a large number of equipments. Therefore, managing spares for proper and timely maintenance is a challenging problem. Generally, spares are categorized into two main categories: fast moving and slow moving parts. Fast moving parts are those that are very often used, and slow moving parts are those that are rarely required. Companies keep record of spares required for a particular type of equipment maintenance, spares received and used in the past, forthcoming major maintenances scheduled, potential vendors, and the spare parts supplied by each vendor.

The aim of this project is to build a decision support system that keeps record of the following: spares used, the current inventory level for each part, scheduled maintenances, etc. The system should enable the managers to identify the weekly demand for each part, parts to be ordered and corresponding order quantities, equipments that should be replaced, etc.

Excel Spreadsheets

1. Build a spreadsheet that keeps the following information about each spare: name, unit cost, on-hand inventory, equipment/s where it can be used and quantity used, name of the vendor that supplies the part, and the corresponding lead-time.
2. Build a spreadsheet that presents the timetable of the maintenances /replacements of equipment.
3. Build a spreadsheet that presents historical information about the spare parts used. The spreadsheet should include the part name, date used, quantity used, and the name of the equipment used in.

User Interface

1. Build a welcome form.
2. Build data entry forms that will enable the user to add/update/delete the records in the spreadsheets of this database.
3. Build a form that enables the user to analyze the data. In this form include the following:
 - a. A combo box that allows the user to view one of the following statistics: the five spare parts that have been used the most in the past; the five spare parts that have been used the least in the past; the five spare parts that have the highest average inventory level; spare parts that are out of stock. Insert a command button that, when clicked on, performs the analysis and displays the statistics selected by the user.
 - b. A list box that presents the name of the spares in this database. Insert a check box to enable the user to choose whether or not to perform a goodness-of-fit test (the Chi-square test). Insert a command button that, when clicked on, builds a histogram, calculates the mean and standard deviation, and performs the goodness-of-fit test (if chosen by the user) for the spare selected by the user.

The histogram of the demand for a spare part, together with its mean and standard deviation, help the user to identify the distribution of demand. For details about how to perform the Chi-square test, see Law and Kelton (2000).

- c. A combo box to enable the user to select a spare part. Insert a command button that, when clicked on, lists all the equipments that use the selected part.
- d. A text box where the user types a date. Insert a command button that, when clicked on, lists details about the maintenances scheduled on the selected date.
- e. A command button that, when clicked on, lists all the equipments for which the current year's expenses on maintenance are higher than 30% of the value of the equipment. The management may decide to replace such equipment.
- f. A command button that, when clicked on, runs the simulation model described below and presents the results.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the current inventory level for each spare part.
2. Report the annual physical consumption for each part and the annual \$ amount spent for each part.
3. List the parts that have been used the most in the current year. For each part, present the total consumption in the current year.
4. List the parts that have been used the least during the current year. For each part, present the total consumption in the current year and the current inventory level.
5. Report the unit cost of the spare parts. Sort the data by unit cost.
6. Considering the current inventory level for each part and the scheduled maintenances, calculate and report the spares requirement for the current month.
7. Report the results from the simulation study and sensitivity analysis.

Simulation Model

A simulation study is necessary to determine when and how many parts to order. We assume the following: there is one week lead-time, there is a fixed cost for each order, and the unit inventory holding cost is equal to 20% of the unit cost of the spare part. Note that a particular part can be used during breakdowns and scheduled maintenances of equipments. Use the historical information to determine the breakdown pattern of equipments. The objective of this simulation is to identify the size and frequency of each order such that the down time of equipments and costs are minimized.

Perform a sensitivity analysis with respect to the batch size and frequency of orders. For each spare part, increase/decrease the size of the order and the pattern of the orders (order every week, every other week, every month, etc.). Rerun the simulation for each scenario. Calculate the average downtime of equipments under each scenario and the corresponding total costs. The results from this study will help the management to decide about the size and frequency of orders.

To learn more about how to build a simulation model and perform output analysis, see Winston (1994).

Reference

Law, A.M., Kelton, W.D., "Simulation Modeling and Analysis," 3rd Ed., *McGraw-Hill*, 2000.

Winston L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

15 *Statistical Process Control*

Problem Description

Statistical process control (SPC) is the application of statistical techniques to determine whether the output of a process or service conforms to the product design. SPC employs a number of techniques to measure the current quality of products and detect whether the process itself has changed in a way that affects the quality. Some of these methods are control charts for variables (R-chart and \bar{x} -chart), control charts for attributes (p-chart and c-chart), process capability ratio, and process capability index.

The R-chart and \bar{x} -chart are used to monitor the mean and variability of process distribution. The p-chart is used to control the proportion of defective products or services generated by the process. The c-chart is used to control the number of defects when more than one defect could be present in a product or service. The process capability ratio is equal to the ratio of the tolerance width and 6 standard deviations. The process capability index measures the potential for a process to generate defective outputs relative to either upper or lower specifications. The last two measures indicate the ability of a process to meet the design specifications for a product or service.

The aim of this project is to build a decision support system that enables the managers to use SPC methods to determine whether the quality of the products produced meets the specifications and to identify what causes the problem in the case when these specifications are not met. To learn more about SPC techniques, the reader is referred to Krajewski and Ritzman (2002).

Excel Spreadsheets

1. The following spreadsheet presents the factors for calculating the three-sigma limits for the R-chart and \bar{x} -chart.

Sample Size (n)	Factor for UCL and LCL for \bar{x} -charts (A ₂)	Factor for LCL for R-charts (D ₃)	Factor for UCL for R-charts (D ₄)
2	1.880	0.000	3.267
3	1.023	0.000	2.575
4	0.729	0.000	2.282
5	0.577	0.000	2.115
6	0.483	0.000	2.004
7	0.419	0.076	1.924
8	0.373	0.136	1.864
9	0.337	0.184	1.816
10	0.308	0.223	1.777

User Interface

1. Build a welcome form.
2. Build a form that includes a frame that has four option buttons. The option buttons enable the user to select one of the following SPC techniques: control charts for

variables, control charts for attributes (the p-chart), control charts for attributes (the c-chart), and the process capability ratio/index. Insert a command button that, when clicked on, submits the user's choice.

- a. If the user selects the first option button, display two text boxes where the user can type in the sample size and number of replications. Insert a command button that, when clicked –on, displays a table that has as many rows as the number of replications and as many columns as the sample size. The user types in the results from the observations in this table. Insert a command button that, when clicked on, does the following: submits the data entered by the user; calculates \bar{R} , $\bar{\bar{x}}$, UCL_R and LCL_R , $UCL_{\bar{x}}$ and $LCL_{\bar{x}}$; creates the R-chart and \bar{x} -chart; and plots the observations on these charts.
- b. If the user selects the second option button, display three text boxes where the user can type in the sample size, the number of replications, and the confidence level for this analysis. Insert a command button that, when clicked on, displays a table that has one column and as many rows as the number of replications. The user types in this table the proportion of defective products for each replication. Insert a command button that, when clicked on, does the following: submits the data entered by the user; calculates \bar{p} , σ_p , UCL_p and LCL_p ; creates the p-chart; and plots the observations on the chart.
- c. If the user selects the third option button, display two text boxes where the user can type in the number of units (products) observed and the confidence level for this analysis. Insert a command button that, when clicked on, displays a table that has one column and as many rows as the number of observations. The user types in this table the number of defects for each unit (product). Insert a command button that, when clicked on, does the following: submits the data entered by the user; calculates \bar{c} , UCL_c and LCL_c ; creates the c-chart; and plots the observations on the chart.
- d. If the user selects the fourth option button, display four text boxes where the user can type in the sample size, number of replications, and upper and lower specifications for the product. Insert a command button that, when clicked on, displays a table that has as many rows as the number of replications and as many columns as the sample size. The user types in the observations in this table. Insert a command button that, when clicked on, does the following: submits the data entered by the user and calculates σ , $\bar{\bar{x}}$, process capability ratio and process capability index.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the following statistics when control charts for variables are used to monitor the mean and variability of the process distribution: \bar{R} , $\bar{\bar{x}}$, UCL_R and LCL_R , $UCL_{\bar{x}}$ and $LCL_{\bar{x}}$. Draw the R-chart and \bar{x} -chart and plot the observations on these charts.

2. Report the following statistics when a p -chart is used to monitor the proportion of defective products or services: \bar{p} , σ_p , UCL_p and LCL_p . Draw the p -chart and plot the observations on the chart.
3. Report the following statistics when a c -chart is used to monitor the number of defects in a product or service: \bar{c} , UCL_c and LCL_c . Draw the c -chart and plot the observations on the chart.
4. Report the following statistics when the user is interested to know the ability of the process to meet the design specifications for a product or service: σ , $\bar{\bar{x}}$, process capability ratio, and process capability index.

Reference

Krajewski, J.L., Ritzman, P.L., "Operations Management: Strategy and Analysis," *Prentice Hall*, 6th Ed., 2002.

16 Support System for a Car Rental Company

Problem Description

A rental car company owns a number of vehicles. The vehicles, when not rented, are located in a number of parking lots around town. The company does not own these parking lots. The company is charged differently for each vehicle. The rate charged depends on the following: the size and type of the vehicle, the location of the parking lot used, the duration of parking, and the time of parking. The rates are lower during the weekend and at night.

The management is concerned about the parking expenses. The management wants to find out the best assignment of vehicles to parking lots so that parking expenses are minimized. We describe a mathematical model to solve this problem. The aim of this project is to build a decision support system that will help the managers to solve the problem of assigning vehicles to parking lots and perform sensitivity analysis with respect to different parameters. The sensitivity analysis will give more insights about how fluctuations in prices and (parking lots') capacity will affect the optimal assignment of vehicles to parking lots.

Integer Programming Model

We propose a mathematical model to solve the problem. The following is the notation used in our problem formulation:

- z_j presents the capacity of parking lot j .
- c_{ijk} presents the cost for parking vehicle i at lot j in period k .
- λ_{ijk} takes value 1 if vehicle i exits lot j in period k ; and 0 otherwise.

The decision variable for this model is as follows:

- x_{ijk} takes value 1 if vehicle i is parked in parking lot j in period k ; and 0 otherwise.

The following is an integer programming formulation of this problem:

$$\min \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K c_{ijk} x_{ijk}$$

subject to :

$$\sum_{j=1}^J x_{ijk} = 1 \quad i = 1, \dots, I; k = 1, \dots, K, \quad (1)$$

$$\sum_{i=1}^I x_{ijk} \leq z_j \quad j = 1, \dots, J; k = 1, \dots, K, \quad (2)$$

$$\sum_{i=1}^I x_{ijk} \leq z_j - \sum_{i=1}^I \sum_{t=1}^{k-1} x_{ijt} + \sum_{i=1}^I \sum_{t=1}^k \lambda_{ijt} \quad j = 1, \dots, J; k = 1, \dots, K, \quad (3)$$

$$x_{ijk} \in \{0,1\} \quad i = 1, \dots, I; j = 1, \dots, J; k = 1, \dots, K. \quad (4)$$

The objective is to minimize the total cost of parking. Constraints (1) show that a vehicle can be parked in only one parking lot in a particular period of time. Constraints (2) show that the total number of vehicles parked in a particular parking lot in a period should not

exceed the capacity of the parking lot. Constraints (3) show that the total number of vehicles assigned to parking lot j in period k must be at most as much as the number of parking spots available. The number of parking spots available is equal to parking lot capacity minus the total number of vehicles assigned to parking lot j from time 0 to $k-1$ plus the total number of vehicles that left the parking lot from time 0 to k . Constraints (4) indicate that the decision variables are binary variables.

Excel Spreadsheets

1. Build a spreadsheet that presents the following information about each parking lot: name, address, capacity, and cost charged for parking a particular vehicle type during different times of the day and different days of the week.
2. Build a spreadsheet that presents the following information about each car: identification number, current location, time it entered the current location, and time it left the location.

User Interface

1. Build a welcome form.
2. Build data entry forms that will allow the user to add/update/delete the records saved in the database.
3. The managers are interested to search the database to find out the following: (i) the parking lots that offer the best prices during the weekend, at night, during working hours, and in the evening; (ii) the parking lots that offer the best prices for a particular type of vehicle; (iii) the parking lot that has the largest number of available spots during the weekend or during working hours, etc. Build a search form to accomplish this task. The following suggestions will help you to design the form.
 - a. Build a frame that has three option buttons. The option buttons enable the user to select one of the following search criteria: best prices, best times, or most available. Insert two command buttons: the OK and Cancel buttons. When the user clicks on the OK button, Form (3.b) is open.
 - b. This form specifies the search criteria. Depending on the selection that the user made in (3.a), one of the following frames appears.
 - c. Build a frame labeled "best prices." The frame has three option buttons named "By Vehicle," "By Parking Lot," and "By Time." Insert a command button that, when clicked on, searches the database for the parking lots that offer the best prices for each vehicle type, the best price overall parking lots, or the best price for each time period, depending on the user's choice.
 - d. Build a frame labeled "Best Times." Insert two combo boxes that allow the user to select a vehicle and a parking lot. Insert a command button that, when clicked on, presents the best (cheapest) time periods for the selected vehicle in the selected parking lot. The combo boxes should have an option called "All" that, when selected by the user, presents correspondingly the best time periods for all vehicles in a particular parking lot or the best time periods for a particular vehicle in all parking lots.
 - e. Build a frame labeled "Most Available." Insert a combo box that allows the user to pick a particular time period. Include a command button that, when clicked on, presents the five parking lots that have the greatest number of available spots.

- f. Insert a command button that, when clicked on, runs the integer programming model and opens Form 4, described below.
4. Build a form that presents the solution of the integer programming model. Insert a frame labeled "Sensitivity Analysis." In this frame include a combo box to enable the user to choose the parameter (C_{ijk}) with respect to which to perform the sensitivity analysis. Insert a command button that, when clicked on, performs the sensitivity analysis and opens the corresponding report.s

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the results of the integer programming model. Report the total monthly expenses of the company for parking.
2. Report the results of the sensitivity analysis.
3. Report for each vehicle the parking lot(s) that offers the best rates.
4. Report the parking lots that offer the best rates overall.
5. Report for each parking lot the rates during the weekend.
6. Plot the hourly utilization of each parking lot during working hours.
7. Plot the hourly utilization of each parking lot during the weekend.

Problem Description

Regional Transit System (RTS) is a company that provides bus transportation service in Gainesville, Florida. The service offered by RTS is free for students and employees of the University of Florida. Therefore, it is the most widely used mode of transportation. Increased popularity of this service has caused a few problems. When a bus is full, it does not stop at the bus station. Students complain about missing buses because of the availability of the seats. In particular Routes 16 and 20 are very busy, as they pass by residential areas populated with students.

The managers of RTS were concerned about complaints they had received. From interviews with bus drivers, the managers found out that buses were full at about 8-9am in the morning or 4-6pm in the afternoon. However, there were time periods during the day when the buses moved almost empty. Also, in certain times during the day, some routes were much busier than others.

The management is interested in improving the performance of RTS. They have thought of a few actions that can be taken to improve the situation. For example, they could buy new buses and/or re-organize the current bus schedule. In the case that more buses are needed, the management needs to know how many buses should be purchased. In the case that the schedule needs re-organizing, they need to figure out the new schedule.

The aim of this project is to build a decision support system that will help the managers understand what is wrong with the current way of operating the system and identify ways to improve the service. We describe a simulation model that will help in the decision-making process.

Simulation Model

1. Use past data to identify the distribution of the arrival of students at the bus stop during peak hours and off-peak hours.
2. Use past data to identify the distribution of the number of students not being served at each bus stop in a particular time period.
3. From here on we will refer to a route r ($r = 1, \dots, R$, R is the total number of routes) as a sequence of stops that starts with stop r_1 and ends at stop r_1 . Trip $s = r*t$ ($s = 1, \dots, T*R$, T is the total number of time periods) travels route r during time period t . For example trip 1 corresponds to the travel in route 1 that starts at 6:00am and ends 6:20am, assuming it takes 20min to travel route 1.
4. Build a simulation model using Excel. This model does the following: (i) randomly generates the number of students who show up at a bus station during different pickup times; and (ii) for every route, it calculates the available bus capacity after every bus stop. To build the simulation, consider the following data that is available to you: the total number of bus routes, total number of trips per route, total number of buses available, total number of stops per route, travel time between two stops, distribution of student arrivals in each bus stop, and distribution of student departures in each bus stop. The bus service goes on from 6:00 in the morning to 12:00 the next morning. If the bus capacity is reached, the bus will not stop at the next bus stop.

5. Run the simulation model using the current number of buses and the current schedule. Calculate the average, minimum, and maximum student waiting time at each bus stop during peak hours and off-peak hours. Calculate the average bus utilization in each route. Give a graphical representation of the total number of students waiting at a bus stop in different hours during the day.
6. Rerun the simulation model for the following scenarios: (a) Change the number of buses operating in a particular route. Note that the most problematic routes are routes 16 and 20. (b) Increase the number of buses operating during the peak hours. (c) Decrease the number of buses operating during off-peak hours. For each scenario, calculate the average, minimum, and maximum student waiting time at each bus stop during peak hours and off-peak hours. Calculate the average bus utilization in each route. Give a graphical representation of the total number of students waiting at a bus stop in different hours during the day.

To learn more about simulation modeling and analysis, see Winston (1994) and Law and Kelton (2000).

Excel Spreadsheets

1. Build a spreadsheet that has as many rows as the total number of trips and as many columns as the total number of stops (over all the routes). Each cell of the table presents the time (during the day) a bus traveling a particular trip stops at a particular bus stop. This table presents the current bus schedule.
2. Build a spreadsheet that presents for each bus the corresponding capacity and the number of the route it travels.
3. Build a spreadsheet that records historical data about the number of students who come to a bus station during peak hours.

User Interface

1. Build a welcome form.
2. Build data entry forms to add/update/delete records in the database.
3. Build a form that allows the user to run the simulation model and report the results. The following are suggestions to help you design the form.
 - a. Insert a frame labeled, "Simulation Model." The frame includes the following:
 - A text box labeled, "Number of Replications," where the user types in the total number of replications for the simulation study.
 - A text box labeled, "Warm-up Period," where the user types in the length of the warm-up period.
 - A command button that, when clicked on, runs the simulation model using the parameters defined by the user, and when the simulation terminates, opens Form 3.b, described below.
 - b. Build a form where the results from the simulation analysis are presented. The simulation analysis provides the following statistics: average waiting time at each bus station during peak hours and off-peak hours, capacity utilization for each bus in different hours during the day, etc.

4. Build a form called “Sensitivity Analysis.” The following are suggestions to help you design the form.
 - a. Insert a frame labeled, “Additional Buses.” The frame includes a combo box listing the possible trips and a text box. The user chooses a trip and then types in the text box the total number of buses to be added on the selected trip. The user can either type in a positive or a negative number in order to increase/decrease the number of buses on a particular trip.
 - b. Insert a frame titled, “Reschedule a Trip.” The frame includes a combo box that lists all the trips. Insert a command button that, when clicked on, displays a table that has one row and as many columns as the total number of stops of the trip selected by the user. The user types in the new schedule for the selected trip.
 - c. Insert a command button that, when clicked on, runs the simulation, taking into account the information provided by the user about additional buses or trips rescheduled.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Report:

1. Report on the result of the simulation study. The report should include the following: average, minimum, and maximum waiting time at each bus stop during peak hours and off-peak hours; total number of students waiting at a bus station; total number of students who could not take the bus because the bus capacity was reached; bus utilization in different times of the day.
2. Report the results from the sensitivity analysis. Note that the results should not be overridden every time the simulation model runs. Instead, keep all the records from the sensitivity analysis.
3. Graph the following for each simulation run: the total number of students waiting at a bus stop in different time periods, total waiting time per bus stop, and bus capacity utilization in different time periods.

Reference

- Law, A.M., Kelton, W.D. “Simulation Modeling and Analysis,” 3rd Ed., *McGraw-Hill*, 2000.
- Winston L.W. “Operations Research: Applications and Algorithms.” *Duxbury Press*, 3rd Ed., 1994.

Problem Description

COMP Co. manufactures three types of computers: personal computers, laptops, and main frames. Two major operations in their production lines are final product assembly and testing. The management is concerned about these two processes, as they involve scarce resources. This issue is making the process of scheduling production challenging.

The aim of this project is to create a decision support system that will help the managers build a production schedule for a period of n weeks, considering resource constraints in the product assembly and testing processes. For the purpose of this project, the user can enter the data manually or read the data from a file.

The Mathematical Model

The following is a linear programming model for the production planning problem. The decision variables for this problem are as follows:

x_{ij} : total number of product j produced during week i ,
 I_{ij} : total amount of product j in inventory at the end of week i ,

$$\min : \sum_{i=1}^n \sum_{j=1}^3 (c_{ij}x_{ij} + h_{ij}I_{ij})$$

subject to:

$$x_{ij} + I_{i-1,j} = d_{ij} + I_{ij} \quad i = 1, \dots, n, j = 1, \dots, 3 \quad (1)$$

$$\sum_{j=1}^3 x_{ij} \leq \gamma \quad i = 1, \dots, n \quad (2)$$

$$\sum_{j=1}^3 \alpha_j x_{ij} \leq \lambda \quad i = 1, \dots, n \quad (3)$$

$$x_{ij}, I_{ij} \geq 0 \quad i = 1, \dots, n, j = 1, \dots, 3. \quad (4)$$

Where,

h_{ij} : unit holding cost of product j during week i ,
 d_{ij} : demand for product j during week i ,
 c_{ij} : unit production cost of product j during week i ,
 α_j : capacity usage coefficient for product j ,
 γ : production (assembling) capacity,
 λ : testing equipment capacity.

The objective is to minimize the total costs and production and inventory holding costs. The first set of constraints presents the flow conservation constraints, the second and third set of constraints are the capacity constraints, and last set of constraints are the non-negativity constraints.

Excel Spreadsheets

1. Build a spreadsheet that presents the unit production cost for each product in every week of the planning horizon.
2. Build a spreadsheet that presents the unit inventory holding cost for each product in every week of the planning horizon.
3. Build a spreadsheet that presents the demand for each product in every week of the planning horizon.
4. Build a spreadsheet that presents the capacity usage coefficient for each product.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions for building this form.
 - a. Insert a frame called "Problem Data." The frame includes two option buttons. The user has the choice to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:
 - i. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - ii. If the user chose to enter the data manually, a text box should appear where the user types in the total number of weeks. After the number of weeks is submitted, four tables appear. In the first table (with dimensions 3 by n) the user types in the unit production cost, in the second table (3 by n) the user types in the unit inventory holding costs, in the third table (3 by n) the user types in the weekly demands, and in the fourth table (1 by n) the user types in the capacity usage coefficients.
 - b. Insert two text boxes where the user can type in the production capacity in the assembly line and the capacity of the testing equipments.
3. Build a form that enables the user to solve the mathematical formulation described above, presents the solution found, and performs a sensitivity analysis with respect to problem parameters. The following are suggestions for building this form.
 - c. Insert a command button that, when clicked on, solves the mathematical formulation of the problem and presents the corresponding solution.
 - d. Insert a frame called "Sensitivity Analysis." Within this frame, insert a sub-frame called "Choose a Parameter." Insert six option buttons that enable the user to choose a parameter (unit production cost, unit holding cost, demand, capacity usage coefficient, assembling capacity, or testing equipment capacity) for the sensitivity analysis. Insert a combo box called "Choose a Product." The combo box enables the user to choose a product for the sensitivity analysis. For example, the user may want to test the sensitivity of the solution with respect to changes on the demand for laptops. Insert a combo box called "Choose a Week." The number of items in the combo box is equal to the total number of weeks in the planning horizon. For example, the user might be interested in testing the sensitivity of the solution with respect to changes on the demand for laptops during the week of Christmas.

- e. Insert a command button that, when clicked on, runs the simulation study using the parameters already defined by the user and opens the corresponding reports.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the total number of personal computers, laptops, and main frames that should be produced every week so that the total production and inventory holding costs are minimized and capacity requirements are not violated.
2. Report the total number of personal computers, laptops, and main frames held in the inventory in each week of the planning horizon.
3. Give a graphical representation of the weekly production and inventory held during the planning period.
4. Report the results from the simulation study.

Reference

Winston, L.W. "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

19 Cash Flow Analysis

Problem Description

A manufacturing company located in Rochester produces a wide range of consumer products. The company is planning to produce a new kind of diaper. The managers are concerned about the success of this new incentive. They have estimated production, inventory, and transportation costs as well as the monthly demand for diapers. However, since this is a new product, the management has expressed the following concerns:

1. Are the forecasts about future demand, price, and costs reliable?
2. Is this product worthy to be invested in?
3. What will be the return from such an investment?
4. How long will this project take to break even?
5. How are the changes in the inflation rate going to affect the profitability (net present value) of this project?

The aim of this project is to build a decision support system that would enable the company to store and analyze the data collected. The results from the data analysis should help the managers to make their decision about producing the product.

Decision Support Tools

In order to answer the questions posed above one can use the following decision support tools:

1. *Forecasts* of future demand and inflation rate. To learn about different forecasting methods, we refer the students to Winston (1994).
2. *Net present value analysis, breakeven analysis, internal rate of return, and payback period*. For details we refer the students to Park (2003).
3. *Sensitivity analysis* with respect to demand, inflation rate, unit price, and unit production and inventory holding costs. To learn about sensitivity analysis, we refer the students to Winston (1994).
4. *Simulation model*. For details about how to build a simulation model using Excel, we refer the students to Winston (1994).

Simulation Model

To build a simulation model for this problem, follow these steps: (a) use historical data to identify the distribution of demand, price, and cost for similar products; (b) create different scenarios by randomly generating demand, price, costs, and interest rate for each period of the planning horizon; (c) calculate the net present value, internal rate of return, and payback period for this project. Re-run steps (b) and (c) of this algorithm as many times as defined by the user.

At the end of the simulation, report the expected net present value, expected internal rate of return from the investment, and expected payback period. Calculate the probability of having an internal rate of return higher than 5%. Calculate a probability of having a

negative return from this project. Calculate the probability of a payback period longer than two years.

Excel Spreadsheets

1. Build a spreadsheet that presents historical data about sales of a similar type of diaper.
2. Build a spreadsheet that presents historical data about unit production costs and inventory holding costs of a similar type of diaper.

User Interface

1. Build a welcome form.
2. Build a form that enables the user to analyze historical data. The following are suggestions for building this form.
 - a. Insert a frame that is called "Select a Forecasting Method." The frame has four option buttons that enable the user to select one of the following forecasting methods: moving-average, exponential smoothing, Holt's method, or Winter's method.
 - b. Insert a frame that is called "Select a Parameter to Forecast." The frame includes a combo box that enables the user to select whether to forecast demand, inflation, interest rate, unit price, unit production cost, or unit inventory holding cost. Insert a text box where the user can type in the total number of periods to be forecasted.
 - c. Insert a command button titled "Forecast!" that, when clicked on, submits the user's choice about the forecasting method and the parameter to be forecasted.
 - d. Insert a frame that is called "Economic Analysis of Data." The frame has three option buttons that enable the user to select one of the following economic analysis tools: net present value, internal rate of return, or payback period. Insert a command button that, when clicked on, submits the information and performs the data analysis.
 - e. Insert a frame that is called "Sensitivity Analysis." The frame includes two combo boxes. One of them enables the user to select a parameter (demand, price, inflation, costs), and the other enables the user to select a period of time. Insert two text boxes where the user can type in a lower and an upper bound on the parameter fluctuation. For example, the user might be interested to know how changes from -10% to +10% in demand during the Christmas period will affect the net present value of the project. Insert a command button that, when clicked on, submits the information and performs the sensitivity analysis with respect to the parameters selected.
 - f. Insert a command button called "Simulation Model" that, when clicked on, opens a new frame where the user identifies the parameters for the simulation analysis. Insert a text box where the user can type in the total number of simulation runs. Insert a command button that, when clicked on, runs the simulation and opens the report with the corresponding results.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms

created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the forecasts of the monthly demand, unit price, unit costs, and inflation during the next year.
2. Report the expected net present value of the project using the forecasted data about demand, costs, inflation, and interest rate.
3. Report the expected payback period and expected internal rate of return using the forecasted data.
4. Report the results from the simulation analysis.
5. Report the sensitivity of the expected rate of return with respect to a 10% fluctuation on demand, price, interest rate, and costs. Report the changes in the expected net present value of the project as demand changes from -10% to +10% in steps of 1%.
6. Graph the relationship between the net present value of the project and the interest rate.

Reference

Park, C. "Contemporary Engineering Economics." *Prentice Hall*, 3rd Ed., 2003.

Winston, L.W. "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

20 Capital Budgeting

Problem Description

Very often companies are in a situation where they have to decide where to invest their funds. There are many opportunities available; however, companies do not have the money and the time to invest in every single one. Companies analyze their alternatives and, depending on the available budget and expected return of each investment, decide how to allocate the funds.

The aim of this project is to build a decision support system that would enable companies to use a number of decision tools in the process of making the final decision about how to allocate their funds.

Below we present a number of decision support tools that can be used by the companies to choose among alternatives. We group these tools as follows:

1. *Economic analysis tools*, such as payback period, discounted payback period, net present value, internal rate of return, modified internal rate of return, profitability index. To learn more about these topics, we refer the students to Park (2003).
2. *Sensitivity analysis* with respect to demand, inflation rate, unit price, unit production cost, and unit inventory holding cost. To learn about sensitivity analysis, we refer the students to Winston (1994).
3. *Simulation model*. To learn about building simulation models using Excel, we refer the students to Winston (1994).

Simulation Model

In order to build a simulation model for this problem, we assume that the following data is available for each investment alternative: the initial investment, the distribution of annual returns, and the distribution of the interest rate.

The model consist of the following steps: (a) for each alternative, create a scenario by randomly generating the return and the interest rate in each period of the planning horizon; (b) calculate the net present value, internal rate of return, and payback period for each alternative. Re-run steps (a) and (b) of this algorithm for as many times as defined by the user.

At the end of the simulation, report the expected net present value, expected internal rate of return, and expected payback period for each alternative. Use the internal rate of return analysis and the net present value analysis to compare the alternatives. Calculate the probability of having an internal rate of return higher than 5%. Calculate a probability of having a negative return from this project. Calculate the probability of having a payback period longer than two years.

Excel Spreadsheets

1. Build a spreadsheet that presents expected returns for each alternative during each period of the planning horizon.

2. Build a spreadsheet that presents expected interest rates for each period of the planning horizon.
3. Build a spreadsheet that presents the initial investment amount for each alternative.

User Interface

1. Build a welcome form.
2. Build a form that enables the user to compare different investment alternatives. The following are suggestions for building this form.
 - a. Insert a text box where the user can type in the budget that is available.
 - b. Insert a frame called “Problem Data.” The frame has two option buttons that enable the user to select whether to read the data from a file (spreadsheet) or enter the data manually. Insert a command button that, when clicked –on, performs one of the following actions depending on the user’s selection:
 - i. If the user chose to read the data from a file, a text box appears that enables the user to type in the name of the file.
 - ii. If the user chose to enter the data manually, two text boxes appear. One of the text boxes enables the user to type in the total number of alternatives and the other the length of the planning horizon. Insert a command button that, when clicked on, submits the user’s input and creates three tables with the appropriate dimensions where the user can type in the expected returns for each alternative during each period of the planning horizon and the interest rate per period.
 - c. Insert a frame that is called “Economic Analysis of Alternatives.” The frame has six option buttons that enable the user to choose one of the following economic analysis tools: payback period, discounted payback period, net present value, internal rate of return, modified internal rate of return, and profitability index per alternative. Insert a command button that, when clicked on, submits the user’s selection and performs the data analysis.
 - d. Insert a frame that is called “Compare Alternatives Using:”. The frame has two option buttons that enable the user to choose one of the following economic analysis tools to compare the alternatives: present value analysis or internal rate of return analysis.
 - e. Insert a frame that is called “Sensitivity Analysis.” The frame includes two combo boxes. One of them enables the user to select a parameter (returns, inflation) and the other enables the user to select a time period. Insert two text boxes where the user can type in a lower and an upper bound on the parameter fluctuation. For example, the user might be interested to know how changes from -10% to +10% in the expected return during the Christmas period will affect the decision about choosing one alternative instead of another. Insert a command button that, when clicked –on, submits the user’s input and performs the sensitivity analysis with respect to the parameter chosen.
 - f. Insert a command button called “Simulation Model” that, when clicked on, opens a form where the user can identify the parameters for the simulation model. Insert a text box where the user can type in the total number of simulation runs. Insert a command button that, when clicked –on, runs the simulation model and opens the report with the corresponding results.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the following for each alternative: net present value, internal rate of return, payback period, discounted payback period, modified internal rate of return, and profitability index.
2. Report the results from using the present value method and internal rate of return method to compare the alternatives.
3. Report on the sensitivity of the final solution with respect to a 10% fluctuation on demand, interest rate, and returns. Show whether the decision about choosing a particular alternative is affected by this change.
4. Report the results from the simulation analysis.
5. Graph the relationship between the net present value of each alternative and the interest rate.

Reference

Park, C. "Contemporary Engineering Economics." *Prentice Hall*, 3rd Ed., 2003.

Winston, L.W. "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

Operating in a dynamically changing environment poses many challenges to companies. One of the problems most companies have to deal with is delays: delays in receiving an order, delays in processing an order, delays in submitting an order, delays in submitting a payment, delays in receiving a payment, etc. Dealing with delays is a very sensitive issue, as it impacts the financial standings of the company.

These days, companies receive payments (checks) from customers all over the United States. The checks are then sent to the bank, and the money is finally transferred to the company's account. Due to the time it takes to ship the checks using the U.S. Postal Service and the time needed to process the checks in the bank, there is a variable delay from the date when the check is postmarked by the customer to the time the check is cleared (the company actually receives the money). The delay time is even longer when the checks are mailed across the country. For example, a check mailed in Pittsburgh to a company located in Pittsburgh is cleared within two days; however, a check mailed in Pittsburgh to a company located in Los Angeles could take at least five days to clear. These delays cost the company, as it could use the money elsewhere. Companies have discovered one way to clear checks as quickly as possible. To speed up the process, companies open offices in different cities around the country. These offices, called lockboxes, handle a company's checks in a timely manner.

The aim of this project is to build a decision support system that would help the managers decide where to locate the lockboxes such that the total time it takes to clear the checks received by customers is minimized.

The Model

In this project we aim to identify locations for the lockboxes such that the total travel time of the checks is minimized. This is a location problem. We provide an integer programming formulation for this problem.

We use the following notation:

l_{ij} : is the lost interest due to the assignment of the region i to lockbox j . This is calculated by multiplying the average daily return of the cash inflow by the number of days that it takes to clear a check.

c_j : is the cost of opening lockbox j .

The decision variables are as follows:

y_j : is a binary variable that takes the value 1 if lockbox j is opened and 0 otherwise.

x_{ij} : is a binary variable that takes the value 1 if region i mails checks to lockbox j and 0 otherwise.

The integer programming model:

$$\min : \sum_{i=1}^I \sum_{j=1}^J (l_{ij}x_{ij} + c_j y_j)$$

subject to :

$$\sum_{i=1}^I x_{ij} = y_j \quad j = 1, \dots, J, \quad (1)$$

$$\sum_{j=1}^J x_{ij} = 1 \quad i = 1, \dots, I, \quad (2)$$

$$x_{ij}, y_j \in \{0,1\} \quad i = 1, \dots, I; j = 1, \dots, J. \quad (3)$$

The objective is to minimize the cost of opening and assigning customers to lockboxes. The first set of constraints shows that an open lockbox is assigned to exactly one region. The second set of constraints shows that a region should be assigned to exactly one lockbox. The third set of constraints is the integrality constraints.

Excel Spreadsheets

1. Build a spreadsheet that presents the lost interests l_{ij} of assigning lockbox i to region j , for all i and j .
2. Build a spreadsheet that presents the cost (c_j) of opening lockbox j .

User Interface

1. Build a welcome form.
2. Build a form that enables the user to add/update/delete the data in the database.

The following are suggestions for building this form. Insert a frame called “Problem Data.”

Insert two option buttons in this frame to enable the user to choose whether to type in the data or read the data from a file. Insert a command button that, when clicked on, performs one of the following actions depending on the user’s selection:

- a. If the user chose to read the data from a file, a text box appears that enables the user to type in the name of the file.
 - b. If the user chose to enter the data manually, two text boxes appear. One of the text boxes enables the user to type in the total number of regions (I) and the other the total number of lockboxes (J). Insert a command button that, when clicked on, submits the user’s input and creates two tables with the appropriate dimensions where the user can type in lost interests of assigning a lockbox to a region and the cost of opening a lockbox.
3. Build a form titled “Data Analysis.” The following are suggestions for building this form. Insert a command button that, when clicked on, uses the Excel solver to solve the mixed integer programming formulation of this problem and presents the results. Insert a frame called “Sensitivity Analyses.” The frame includes two combo boxes. The first combo box allows the user to select one of the parameters c_j ($j = 1, \dots, J$), and the second combo box allows the user to select one of the parameters l_{ij} ($i = 1, \dots, I; j = 1, \dots, J$) for the sensitivity analysis. Insert a command button that, when clicked on,

runs the sensitivity analysis with respect to the selected parameter(s) and opens a report that presents the results of the analysis.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the optimal locations of the lockboxes, the optimal assignment of regions to lockboxes, and the optimal assignment cost.
2. Give a graphical representation of the final solution.
3. Report the results of the sensitivity analysis.

Reference

Winston L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

Simulation is a widely used tool that helps managers to make decisions. Simulation has shown to be valuable in modeling and analyzing complex processes and systems that cannot be represented using optimization models. In this project we describe how to build a decision support system for a simple manufacturing environment. We use simulation for modeling and analyzing the system.

For the purpose of this project, we consider a production facility that has four workstations. In the process of transforming the raw materials into final products, all the workstations are visited sequentially. The inter-arrival time of the raw materials at the facility follows an exponential distribution. The processing time at a particular workstation follows a lognormal distribution. The length of the queue in front of each workstation is limited by the space available. There is an inspector in each workstation. There is a certain probability that a product needs rework or is scrap. Scraps are disposed and the products that need rework are sent back.

The aim of this project is to build a model to simulate this facility, verify and validate the model, and analyze the results. For the purpose of analyzing the performance of the system, calculate the average/minimum/maximum queue length in each workstation, the average/minimum/maximum waiting time in each queue, the average throughput, the average number of defects per workstation, the total amount of scrap, etc. For details on how to build a simulation model, verify and validate the model, and analyze the results, we refer the students to Winston (1994) and Law and Kelton (2000).

User Interface

1. Build a welcome form.
2. Build a form titled "Simulation Model" that includes the following:
 - a. A frame titled "Inter-arrival Time." The frame includes a combo box that enables the user to choose a distribution (such as exponential, uniform, etc.) for the inter-arrival time of the raw materials. As the user makes a selection, text boxes appear where the user can type in the parameters of the selected distribution. Insert a text box where the user can type in the batch size.
 - b. A frame titled "Processing Time at Workstations." The frame includes four combo boxes. The combo boxes enable the user to pick a distribution for each workstation. As the user makes a selection, text boxes appear where the user can type in the parameters of each distribution.
 - c. A frame titled "Probability of Defective Items." The frame includes eight text boxes where the user can type in the probability of scrap and rework at each workstation.
 - d. Three text boxes where the user can type in the total number of iterations, the length of the iteration, and the length of the warm-up period.
 - e. A frame titled "Animation." The frame includes a graphical representation of the model.

- f. A command button that, when clicked on, submits the information provided by the user, runs the simulation, animates the simulation, and opens Form 3, described below.
3. Build a form titled "Summary of Results." This form includes a table that presents the following statistics from the simulation study: the average/minimum/ maximum queue length in each workstation, the average/minimum/maximum waiting time in each queue, the average throughput, the average number of defects per workstation, and the total number of scrapped items. Insert a command button titled "Rerun the Simulation." When the user clicks on this button, Form 2 opens. The user can change some of the parameters on this form and run the simulation model again. The purpose of creating the summary table and allowing the user to rerun the simulation is to allow the managers to examine the performance of the system, identify things that should change, make the changes, and see the results.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the following statistics from the simulation study: the average cycle time, the average throughput, the average queue length for each workstation, the average waiting time in each queue, the total amount of scrap, and the utilization of each workstation.
2. Give a graphical representation of the utilization of each workstation with respect to time.
3. For each workstation, give a graphical representation of the queue length with respect to time.

Reference

Law, A.M., Kelton, W.D., "Simulation Modeling and Analysis," 3rd Ed., *McGraw-Hill*, 2000.

Winston L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

The aim of this project is to build a support system that will enable agricultural businesses to make better decisions. The following are some of the issues a farmer is concerned about.

1. Land is an expensive and scarce resource. A farmer has to decide during the planting season how much land to allocate to each crop.
2. The farmer has to decide how much to produce. This is not an easy decision because of the uncertainties of demand and production yield. Production yields depend on unpredictable factors such as weather, insect population, etc.
3. The farmer has to identify how many employees to hire.

In order to help the farmer identify an optimal allocation of resources to each crop, we use linear programming. Linear programming has shown to be a powerful optimization tool that has been used to model several other problems. Below we describe the linear programming model.

Linear Programming Model

We use the following notation:

$A:$	presents the total number of acres available for harvesting
$d_i:$	presents the demand for crop i
$l_i:$	presents the expected yield (per acre) of crop i harvested
$h_i:$	presents the number of working hours required to harvest one acre of crop i
$H:$	presents the maximum number of working hours that can be purchased at a rate of \$ r /hour
$c_i:$	presents the cost of planting one acre of crop i .

The decision variables are as follows:

$x_i:$	presents the total number of acres allocated to crop i
$z_i:$	presents the total number of working hours needed to harvest crop i .

The objective is to minimize the total costs of planting and harvesting the crops. The first set of constraints shows that the whole available land should be planted. The second set of constraints shows that the total number of working hours allocated to harvesting the crops should not be greater than the total number of working hours available. The third set of constraints shows that production of a particular crop should be at least as large as the corresponding demand. The last set of constraints is the non-negativity constraints. To learn more about linear programming we refer the students to Winston (1994).

$$\min : \sum_{i=1}^n (c_i x_i + r_i z_i)$$

Subject to :

$$\sum_{i=1}^n x_i = A \quad (1)$$

$$\sum_{i=1}^n h_i x_i \leq H \quad (2)$$

$$l_i x_i \leq d_i \quad i = 1, \dots, n, \quad (3)$$

$$x_i, z_i \geq 0 \quad i = 1, \dots, n. \quad (4)$$

Excel Spreadsheets

1. Build a spreadsheet that presents the annual demand for crops.
2. Build a spreadsheet that presents the cost of planting one acre of land of a particular crop i for $i = 1, \dots, n$.
3. Build a spreadsheet that presents the production yield for crop i for $i = 1, \dots, n$.
4. Build a spreadsheet that presents the total number of hours required to harvest one acre of crop i for $i = 1, \dots, n$.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions for building the form. Insert a frame titled "Problem Data." The frame includes two option buttons. The user has the choice to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, a text box should appear where the user types in the total number of crops. After the number of crops is submitted, four tables appear with dimensions 1 by n . In the first table the user types in the annual demand for each crop, and in the second table the user types in cost of planting each crop. In the third table the user types in the expected yield of each crop. In the fourth table the user types in the total number of hours required to harvest one acre of each crop.

Insert a text box where the user can type in the total number of acres of land available. Insert two text boxes where the user can type in the total number of hours that can be purchased and the corresponding hourly payment rate. Insert a command button that, when clicked on, submits the data for this problem, solves the corresponding linear programming problem, and opens Form 3, described below.

3. Build a form titled "Analyzing the Problem." Insert two frames in this form. The first frame, titled "Reports," has a number of option buttons that enable the user to choose to open one of the summary reports described below. The second frame, titled "Sensitivity Analysis," has four combo boxes, one for each of these parameters: demand, planting cost, yield, and number of working hours per crop. These combo

boxes enable the user to select a particular parameter for the sensitivity analysis. For example, the user may want to know the sensitivity of the final solution with respect to demand for corn. Include a command button that, when clicked on, performs the sensitivity analysis with respect to the selected parameter and presents the results.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the total number of acres allocated to each crop.
2. Report the total number of working hours needed to harvest each crop.
3. Report the total costs of planting and harvesting each crop.
4. Report the unmet demand for each crop.
5. Report the results from the sensitivity analysis.

Reference

Winston L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

The aim of this project is to create a decision support system that will help the process of scheduling production in a manufacturing environment. We present here a simple example that demonstrates the challenges faced by the managers of a wood furniture manufacturing company when preparing a production schedule. The company produces a number of products such as different designs of wood desks, chairs, drawers, etc. The raw material used in the production process is oak wood. In the process of producing the final product, the raw materials (wood logs) have to go through a number of machineries. Not every machine is used in the process of producing a particular product. There is a limitation in the number of products that can be processed in each machine because of capacity constraints. There are limitations in the availability of raw materials, and finally there are limitations in the total number of final products produced because of the storage space.

The managers are concerned about allocating the resources available (raw material and machines) to final products. Since demand for their products has been growing lately, the managers are interested in identifying ways to increase production. The following are alternatives they want to explore: increase the amount of raw materials available in the case that machines are not fully utilized, buy new machines, rent more space, etc.

Below we present a mixed-integer programming model that solves this resource allocation problem.

Mixed-Integer Programming Model

We use the following notation:

M_j :	the production capacity of machine j (in working hours)
k :	the total amount of raw material available
c_i :	the cost of manufacturing one unit of product i
s_j :	the fixed cost to set up machine j
l_i :	the amount of raw material required to produce one unit of product i
t_{ij} :	the total amount of processing time of product i on machine j
u_i :	the upper bound on the total amount of product i produced.

The decision variables are as follows:

x_i :	the total number of product i produced.
y_j :	a binary variable equal to 1 if machine j is set up and 0 otherwise.

$$\min : \sum_{i=1}^P c_i x_i + \sum_{j=1}^n s_j y_j$$

Subject to :

$$\sum_{i=1}^P t_{ij} x_i \leq M_j y_j \quad j = 1, \dots, n, \quad (1)$$

$$\sum_{i=1}^P l_i x_i = k \quad (2)$$

$$0 \leq x_i \leq u_i \quad i = 1, \dots, P, \quad (3)$$

$$y_j \in \{0,1\} \quad j = 1, \dots, n, \quad (4)$$

The objective is to minimize the total production costs, manufacturing costs, and machine set-up costs. The first set of constraints shows that the total amount of time machine j is used should be less than the corresponding production capacity. The second set of constraints shows that the raw material available should be used. The third set of constraints shows that there is an upper bound on the amount of product i produced.

Excel Spreadsheets

1. Build a spreadsheet that presents the unit manufacturing costs and the production upper bound for product i ($i = 1, \dots, P$).
2. Build a spreadsheet that presents the fixed set-up costs and production capacity for machine j ($j = 1, \dots, n$).
3. Build a spreadsheet that presents the amount of raw material used to produce one unit of product i ($i = 1, \dots, P$).
4. Build a spreadsheet that presents the total amount of processing time product i requires on machine j ($i=1, \dots, P; j=1, \dots, n$).

User Interface

1. Build a welcome form.
2. Build a data entry form. The following suggestions help to design this form. Insert a frame titled "Problem Data." The frame includes two option buttons. The user has the choice to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, two text boxes should appear where the user types in the total number of products and the total number of machines. After this information is typed in, four tables appear. In the first table (with dimensions 2 by P) the user types in the unit manufacturing cost and the upper bound of production quantity for each product; in the second table (2 by n) the user types in the fixed set-up costs and production capacity for each machine; in the third table (1 by P) the user types in the amount of raw material used to produce one unit of each product; in the fourth table (n by P) the user types in the total amount of processing time each product needs on each machine.

Insert a text box where the user can type in the total amount of raw material available. Insert a command button that, when clicked on, submits the data for this problem, solves the corresponding linear programming problem, and opens Form 3, described below.

3. Build a form titled "Problem Analysis." Insert two frames in this form. The first frame, titled "Reports," has a number of option buttons that enable the user to choose to open one of the summary reports described below. The second frame, titled "Sensitivity Analysis," has four combo boxes, one for each of these parameters: unit manufacturing cost, machine set-up cost, machine processing time for each product, and machine capacity. These combo boxes enable the user to choose a particular parameter to perform a sensitivity analysis. For example, the user may want to know what would happen to the total production costs and production quantity if more machines were purchased. Include a command button that, when clicked on, performs the sensitivity analysis with respect to the selected parameter.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the optimal amount that should be produced for each product.
2. Report the total number of machine hours required to produce each product.
3. Report the total production costs.
4. Report the utilization of each machine.
5. Report the results from the simulation analysis.

Source

Winston L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

A big challenge faced by many service agencies and government organizations is managing a variety of tasks with a limited budget. Often, the budget available is not enough to support all tasks; therefore, organizations have to invest in only the most profitable ones.

The aim of this case study is to build a decision support system that will enable a government organization to select the most important from a variety of tasks. Note that organizations make official commitments to fund certain tasks; hence, those commitments must be honored. However, there are several other tasks that should undergo a selection process. The value of each task is measured based on a set of criteria. Some of the tasks have multiple budget levels, such as minimum, planned, and accelerated levels. Different budget levels for a task have a different value to the organization. However, only one of the available levels of a particular task can be included in the final portfolio, making the tasks mutually exclusive. Below we present an integer programming formulation of this problem.

Integer Programming Model

We use the following notation:

T	the total number of tasks
I	the total number of evaluation measures
P	the set of “must-fund” tasks, $ P < T$
B	the total available budget
b_t	the budget for task t
x_{it}	the score of task t on the evaluation measurement i
w_i	the weight assigned to the i -th evaluation measure
D_s	sets of mutually exclusive dependent tasks; $s = 1, 2, \dots, S$.

The following are our decision variables:

y_t	is a binary variable that takes the value 1 if task t is in the portfolio, and 0 otherwise.
-------	---

The objective is to maximize the performance of the system. The first set of constraints shows that the number of tasks selected is limited by the availability of the budget. The second set of constraints shows that the tasks to which the government has made a commitment should be selected. The third set of constraints shows that at most one of the mutually exclusive tasks should be selected. The final set presents the integrality constraints.

$$\max : \sum_{t=1}^T y_t \sum_{i=1}^I w_i x_{it}$$

subject to :

$$\sum_{t=1}^T b_t y_t \leq B \quad (1)$$

$$\sum_{t \in P} y_t = |P| \quad (2)$$

$$\sum_{t \in D_s} y_t \leq 1 \quad s = 1, \dots, S, \quad (3)$$

$$y_t \in \{0,1\} \quad t = 1, \dots, T. \quad (4)$$

Excel Spreadsheets

1. Build a spreadsheet that presents the budget requested for each task.
2. Build a spreadsheet that presents the name of each evaluation criterion and the corresponding weight.
3. Build a spreadsheet that presents the score that each task has received in each evaluation criterion.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions that help in designing this form. Insert a frame named "Problem Data." The frame includes two option buttons. The user has the choice to select whether to read the data from a file or manually enter the data in the database. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, two text boxes should appear where the user types in the total number of tasks and total number of evaluation measures. After this information is submitted, three tables appear. In the first table (with dimensions 1 by T) the user types in the budget requested for each task. In the second table (with dimensions 2 by I) the user types in the name and weight for each evaluation criteria. In the third table (with dimensions I by T) the user types in the score that tasks receive in each evaluation criterion.

Insert a text box where the user can type in the total budget available. Insert a frame named "Must-Fund Tasks" that includes a list box and a command button. The list box presents all the tasks under consideration. The user can select more than one task from this list as the "must-fund" tasks. The command button allows the user to submit the names of the selected tasks. Insert a frame named "Mutually Exclusive Tasks" that includes a list box and a command button. The list box presents all the tasks under consideration. The user selects the mutually exclusive tasks and clicks on the command button to submit this information. Insert a command button that, when clicked on, uses

the Excel solver to solve the integer programming formulation of this problem and opens Form 3, described below.

3. Build a form titled "Problem Analyses." This form includes two frames. The first frame, titled "Select a Report," includes a number of check boxes that enable the user to select to open one of the reports described below. The second frame is titled "Select a Parameter." This frame includes the following: a text box where the user can type in the total budget; a combo box that allows the user to choose one of the listed weights w_i ($i = 1, \dots, I$); and a combo box that allows the user to choose one of the listed tasks for the sensitivity analysis. Insert a command button that, when clicked on, runs the sensitivity analysis with respect to the selected parameter(s) and opens the corresponding reports.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the portfolio of tasks selected by the Excel solver.
2. Report the results from the sensitivity analysis with respect to the total budget, the weight assigned to each evaluation criterion, the score that a task received in each criterion, etc.

Reference

Winston L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

26 Markov Chains

Problem Description

Markov chains have been widely used to model stochastic processes. Markov chains have been applied in areas such as education, marketing, health services, finance, accounting, etc. The aim of this project is to build a decision support system that would enable the user to answer a number of questions related to Markov chains.

Markov chains are a special type of discrete time stochastic process that evaluate the probability of a system being in a particular state at time $t+1$ by only using the knowledge of the state of the system at time t (i_t) and disregarding the states that the system had to pass through on the way to i_t .

$$\begin{aligned} P(X_{t+1} = i_{t+1} | X_t = i_t, X_{t-1} = i_{t-1}, \dots, X_1 = i_1, X_0 = i_0) \\ = P(X_{t+1} = i_{t+1} | X_t = i_t). \end{aligned} \quad (1)$$

Where, X_t is the relevant system characteristic at time t . The following formula shows how to calculate the probability that the system will be at state j in time $t+n$, given that the system is at state i at time t .

$$P(X_{t+n} = j | X_t = i) = P(X_n = j | X_0 = i) = P_{ij}(n). \quad (2)$$

Where, $P_{ij}(n)$ is the n -state probability of transition from state i to state j . $P_{ij}(n)$ is the ij^{th} element of the transition probability matrix P^n . Note that for $n=0$, $P_{ij}(0) = P(X_0 = j | X_0 = i)$, therefore $P_{ij}(0) = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases}$. It might happen that we do not know the state of the chain at time 0, therefore,

$$\text{Probability of being in state } j \text{ at time } n = \sum_{i=1}^s q_i P_{ij}(n) \quad (3)$$

Where, q_i is the probability that the chain is at state i in time 0; and s presents the total number of states. A state i is an absorbing state if $P_{ii} = 1$. Given the transition matrix P for an s -state ergodic chain, the steady state probabilities ($\pi = [\pi_1, \pi_2, \dots, \pi_s]$) are such that:

$$\lim_{n \rightarrow \infty} P^n = \begin{bmatrix} \pi_1 & \pi_2 & \dots & \pi_s \\ \pi_1 & \pi_2 & \dots & \pi_s \\ \vdots & \vdots & \ddots & \vdots \\ \pi_1 & \pi_2 & \dots & \pi_s \end{bmatrix}. \quad (4)$$

To learn more about Markov chains we refer the students to Winston (1994).

Excel Spreadsheets

Build a spreadsheet that presents the transition probability matrix P^n .

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are instructions to help you design this form. Insert a frame called “Problem Data.” The frame includes two option buttons. The user has the choice to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, a text box should appear where the user types in the total number of states. Once the total number of states is submitted, a table with dimensions s by s appears. The user types in the transition probabilities.
3. Build a form titled “Analyzes.” The following are suggestions to help design this form.
 - a. Insert a frame titled “Calculate n -Step Transition Probabilities.” Include two option buttons on the frame, named “I know the state of the Markov chain at time 0” and “I don’t know the state of the Markov chain at time 0.”
 - i. If the user chose the first option button, the following controls appear: a combo box that enables the user to pick a current state i ; a combo box that enables the user to pick the desired state (state j) after a number of periods; and a text box where the user can type in the number of periods (n) to move from i to j . Insert a command button that, when clicked on, submits the data and returns $P_{ij}(n)$.
 - ii. If the user chose the second option button, build a form that, in addition to the controls described in part 3.b.i, has a list box where the user can type in the probability that the original state is state i (type in q_i for $i = 1, \dots, s$).
 - b. Insert a frame titled “Is State i Reachable from State j ?” The frame includes two combo boxes that allow the user to select states i and j and a command button that, when clicked on, submits the information and returns an answer to the question of whether state i can be reached from state j .
 - c. Insert a command button titled “Calculate Steady-State Probabilities” that, when clicked on, returns the steady-state probabilities of the Markov chain.
 - d. Insert a command button named “List the Absorbing States” that, when clicked on, returns the absorbing states of the system. If the system has at least one absorbing state, a form appears that has three check buttons. The text next to the first check button states, “If the chain begins in a given transient state, and before we reach an absorbing state, what is the expected number of times that each state will be entered?” The text next to the second check button states, “How many periods do we expect to spend in a given transient state before absorbing takes place?” The text next to the third check button states, “If a chain begins in a given transient state, what is the probability that we end up in each absorbing state?” Include a

command button that, when clicked on, submits the question selected by the user and returns an answer.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Give a network representation of the transition probabilities. In this network, nodes represent a state, and there is a directed arc between states i and j if state j can be reached from state i . Use text boxes to present the probability of reaching node j from node i . Interpret the results.
2. Report all the absorbing states of the system. Interpret the results.
3. Report the steady-state probabilities of the Markov chain. Interpret the results.
4. Report the answers to the questions posed in Form 3.f.

Reference

Winston L.W., "Operations Research: Applications and Algorithms." Duxbury Press, 3rd Ed., 1994.

Problem Description

Assigning jobs to machines is a problem faced by many manufacturing companies. Usually, a job requires a considerable amount of time to process in a particular machine. Companies are concerned about the total time it takes to complete all jobs.

The aim of this project is to build a decision support system that would enable the managers to decide about assigning jobs to machines in such a way that the total time it takes to complete all jobs is minimized. For the purpose of this project we assume that one job can be completed by a single machine and a machine can work only at one job at a time. We give an integer programming formulation and describe an algorithm that can be used to solve the problem.

Integer Programming Model

We use the following notation:

t_{ij} : the time required by machine i to process job j

Our decision variables are as follows:

x_{ij} : takes the value 1 if machine i is assigned to job j , and 0 otherwise.

$$\min : \sum_{i=1}^I \sum_{j=1}^J t_{ij} x_{ij}$$

Subject to:

$$\sum_{i=1}^I x_{ij} = 1 \quad j = 1, \dots, J, \quad (1)$$

$$\sum_{j=1}^J x_{ij} \leq 1 \quad i = 1, \dots, I, \quad (2)$$

$$x_{ij} \in \{0,1\} \quad i = 1, \dots, I; j = 1, \dots, J. \quad (3)$$

The objective is to minimize the total machine processing time. The first and second set of constraints show that all jobs should be assigned to machines, a job uses a single machine, and a machine can process at most one job. The third set is the integrality constraints. This is a general assignment problem that can be solved using the transportation simplex method. However, a simpler and efficient heuristic that can be used is the Hungarian method. Below we present the main steps of the Hungarian method. To learn more about the Hungarian method we refer the students to Bazara *et al.* (1990).

Hungarian Method

Step 1:

Initialize the processing time matrix M ($m_{ij} \in M$, $i = 1, \dots, I; j = 1, \dots, J$). The matrix presents the time it takes to process each job in each machine.

Find the minimum element of each row ($m_i^* = \min m_{ij}$ for $i = 1, \dots, I$).

Create M^* such that $M^* = \begin{bmatrix} m_{11} - m_1^* & m_{12} - m_1^* & \dots & m_{1J} - m_1^* \\ m_{21} - m_2^* & m_{22} - m_2^* & \dots & m_{2J} - m_2^* \\ \vdots & \vdots & \ddots & \vdots \\ m_{I1} - m_I^* & m_{I2} - m_I^* & \dots & m_{IJ} - m_I^* \end{bmatrix}$

Find the minimum element of each column ($m_j^{**} = \min m_{ij}$ for $j = 1, \dots, J$).

Create the reduced cost matrix M^{**} such that

$$M^{**} = \begin{bmatrix} m_{11}^* - m_1^{**} & m_{12}^* - m_2^{**} & \dots & m_{1J}^* - m_J^{**} \\ m_{21}^* - m_1^{**} & m_{22}^* - m_2^{**} & \dots & m_{2J}^* - m_J^{**} \\ \vdots & \vdots & \ddots & \vdots \\ m_{I1}^* - m_1^{**} & m_{I2}^* - m_2^{**} & \dots & m_{IJ}^* - m_J^{**} \end{bmatrix}$$

Step 2:

Draw the minimum number of lines (horizontal and/or vertical) that are needed to cover all zeros in the reduced-cost matrix. If the minimum number of lines is equal to the order of the matrix, an optimal solution is available among the covered zeroes in the matrix. If the minimum number of lines is less than the order of the matrix, go to Step 3.

Step 3:

Select the smallest (non-zero) element in the reduced-cost matrix that is uncovered by the lines drawn in Step 2. Subtract this element from each uncovered element of the reduced-cost matrix and add it to each element that is covered by two lines. Return to Step 2.

Excel Spreadsheets

Build a spreadsheet that presents the processing time of each job in each machine.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following suggestions help in designing the form. Insert a frame that includes two option buttons. The user has the choice to select whether to read the data from a file or manually enter the data in the database. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, two text boxes should appear where the user types in the total number of jobs (J) and the total number of machines (I). Once this information is submitted, a table with dimensions I by J appears. The user types in the processing times in this table.

Insert a command button that, when clicked –on, solves the problem using the data provided by the user and the algorithm described above and opens Form 3, described below.

3. Build a data analysis form. The following are suggestions for designing this form. Insert two frames in this form. The first frame, titled “Reports,” has a number of option buttons that enable the user to choose to open one of the summary reports described below. The second frame, titled “Sensitivity Analysis,” has two combo boxes. The first combo box allows the user to choose a job (say, job j), and the second combo box allows the user to choose a machine (say, machine i) for the purpose of the sensitivity analysis. For example, the user may want to know the sensitivity of the final solution with respect to the processing time of job j in machine i . Include a command button that, when clicked on, performs the sensitivity analysis with respect to the selected parameter.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the optimal assignment of the machines to jobs.
2. Give a graphical representation of the assignment of jobs to machines. Use a bi-partite graph where one set of nodes presents the jobs and the other set of nodes presents the machines. The arcs represent the assignment of a job to a machine.
3. Report the results of the sensitivity analysis.

Reference

Bazara, S.M., Jarvis, J.J., Sherali, H.D. “Linear Programming and Network Flows.” *John Wiley & Sons, Inc.*, 2nd Ed., 1990.

28 Modeling Queuing Systems

Problem Description

The notion of a queue as a waiting line for service or to use a resource is familiar to most of us. Queues occur if there is an imbalance between the number of requests for a resource and resource capacity. We use mathematical models to analyze waiting lines. Based on their specific characteristics, such as customer arrival, service patterns, and the number of servers available, queues are classified into M/M/1 queuing systems, M/M/s queuing systems, etc. The M/M/1 is the simplest queuing model. In this model, the distribution of arrivals in the system is exponential, the distribution of service time is exponential, and the system has a single server.

The aim of this project is to build a decision support system that enables the user to analyze M/M/1 queuing systems with the help of mathematical models. The reason for studying this particular queuing system is we very often face such systems in everyday life. For example, the University of Florida has an information desk located at the center of the campus. At the information desk a student is available to answer questions. Students wait in line to be served. The line has unlimited capacity. Problems with this system arise particularly in the beginning of the academic year when the newcomers fill up the queue. Students have to wait for a long time to be served, and the long queue itself disturbs the traffic on campus. The university administration is concerned about this problem; therefore, it is considering purchasing automated response machines that will significantly reduce the service time. In this process the administration needs to figure out whether the reduction on the queue length and waiting time and changes in the customer satisfaction from the new service are worth the cost of purchasing the equipment and training the service persons.

Modeling M/M/1 Queuing Systems

We use the following notation:

r	the arrival rate to the queue
s	the service rate
i	the traffic intensity ($i = r / s$)
P_n	the probability that there are exactly n customers either waiting in the queue or being served at any point in time ($P_n = (1 - i)^n$)
L	the number of customers in the system
L_q	the expected length of the queue
W	the average time spent in the system.

Note that (a) a particular time unit should be identified and used throughout the calculations; (b) the system stability i should be less than 1, which implies that the arrival rate should be less than the service rate of the system; (c) $L = i / (1 - i)$; (d) $L_q = i^2 / (1 - i)$; (e) $W = L / r$; (f) $P(\text{time in queue} > t) = ie^{-s(1-i)t}$, $t > 0$. To learn more about Queuing Theory, we refer the students to Winston (1994) and Winston and Albright (2002).

Solution Algorithm

Step 1

Collect data about students' arrival and service time at the information desk. Use this data to simulate the system AS IS. To learn about how to build a simulation model in order to replicate an M/M/1 queuing system by using Microsoft Excel, we refer the students to Winston (1994). Collect the following statistics from the simulation: average/minimum/maximum queue length, average waiting time in the queue, average time in the system (t), and average utilization of the server.

Step 2

Build a simulation model considering that automated machines are used instead of a server. To run this simulation, use the data collected about students' arrival at the information desk and estimates about the service time at the automated response machines. Collect the following statistics for the new system: average/minimum/maximum queue length, average waiting time in the queue, average total time in the system (t_{new}), and average utilization of the machines.

Step 3

Calculate:
$$X = \frac{\sum_{i=1}^n (t_i - t_i^*)}{n}$$
; where t_i is the time spent currently in the system by student i ,

and t_i^* is the time spent in the system by student i if the new system is in place. Thus, X is the average time saved per student. Calculate the total savings per unit of time ($X * r$). Calculate the total costs saved using the new system $X * r * m$, where m is the subjective cost of having a student wait for one unit of time in the system. Calculate the cost of using the automated response machines per unit of time (Y). One can calculate Y based on the amortization of the machines per time unit. For the purpose of this project, one can use straight-line amortization. To learn more about straight-line amortization, we refer the students to Park (2003). If $Y < X * r * m$, we can say that the addition of the automated machines is beneficial and the administration should implement it.

Excel Spreadsheets

1. Build a spreadsheet that presents the arrival time of the students in the system.
2. Build a spreadsheet that presents the service time at the information desk.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions for designing this form. Insert a frame that includes two option buttons. The option buttons enable the user to select whether to read the data from a file or enter the data manually. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.

- b. If the user chose to enter the data manually, a text box should appear where the user types in the total number of observations Z . Once this information is submitted, a table with dimensions Z by 2 appears. The user types in the arrival time of each observation and the corresponding service time.

Insert two text boxes where the user can type in the purchasing cost of automated response machines and the total number of machines to be purchased. Insert a text box where the user can type in the cost of having a student wait for one unit of time in the system.

3. Insert a frame titled "Simulate the M/M/1 Queuing System." The frame includes the following:
 - a. A sub-frame titled "Simulate the Current System." The frame includes three text boxes where the user can type in the simulation length, the total number of simulation runs, and the length of the warm-up period. Insert a command button titled "Run the Simulation" that, when clicked on, runs the simulation model for the current system (runs Step 1 of the solution algorithm), prepares the corresponding reports, and prompts the user to open these reports.
 - b. A sub-frame titled "Simulate the New System." The frame includes three text boxes where the user can type in the simulation length, the total number of simulation runs, and the length of the warm-up period. Insert a command button titled "Run the Simulation" that, when clicked on, runs the simulation model for the new system (runs Step 2 of the solution algorithm), prepares the corresponding reports, and prompts the user to open these reports.
 - c. Insert a command button that, when clicked on, compares the two systems (runs Step 3 of the solution algorithm) and opens Form 4, described below.
4. Build a form that summarizes the results from analyzing this M/M/1 queuing system. The following are suggestions for designing this form.
 - a. Insert a frame titled "Reports" that has a number of option buttons. The option buttons enable the user to choose to open one of the summary reports described below.
 - b. Insert a frame titled "Sensitivity Analysis" that has the following: two combo boxes to allow the user to select a distribution for the service time and the arrival of the customer in the queue; text boxes to allow the user to enter the corresponding parameters of arrival time and service time; a text box where the user can type in the total number of automated response machines to be purchased. Include a command button that, when clicked on, performs the sensitivity analysis with respect to the selected parameter. For example, the user might be interested to know the sensitivity of the solution with respect to service time, the number of automated machines purchased, etc.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the following statistics:
 - a. The average/minimum/maximum number of customers in the current system.

- b. The average/minimum/maximum number of customers in the new system.
- c. The average/minimum/maximum waiting time in the current system.
- d. The average/minimum/maximum waiting time in the new system.
- e. The average/minimum/maximum queue length in the current system.
- f. The average/minimum/maximum queue length in the new system.
- g. The average/minimum/maximum time spent in the current system.
- h. The average/minimum/maximum time spent in the new system.
- i. The probability that a customer would spend more than 10 minutes in the queue in the current system.
- j. The probability that a customer would spend more than 10 minutes in the queue in the new system.
- k. Server utilization in the current system.
- l. Server utilization in the new system.
- m. The average (waiting and service) time saved per student when using the new system.
- n. The total (waiting and service) savings per unit of time when using the new system ($X * r$).
- o. The total costs savings per unit of time when using the new system ($X * r * m$).
- p. The equipment amortization per unit of time.

Based on these statistics, demonstrate whether the new system is better than the current one.

2. Report the results from the sensitivity analysis.

Reference

- Park, C., "Contemporary Engineering Economics." Prentice Hall, 3rd Ed., 2003.
- Winston, L.W., "Operations Research: Applications and Algorithms." Duxbury Press, 3rd Ed., 1994.
- Winston, L.W., Albright, S.C., "Practical Management Science: Spreadsheet Modeling & Applications Update," Duxbury Press, 2002.

Problem Description

The head of the marketing department in an automotive manufacturing company is planning a campaign to advertise their newest model. Based on his past experience, he believes that advertisement campaigns directly affect sales. However, there is a limited budget allocated to advertising.

The marketing team has identified a few alternatives; however, further investigations are necessary to identify the best approach to successfully complete this campaign. Two major magazines (A and B) were identified for advertising the new product. The length of the advertisement period in each magazine could be up to six, 10, or 16 days. The rates charged differ by magazine and length of the advertising period. The team has estimated the number of people who would read the advertisement by magazine and advertisement period length.

The main objective of this project is to build a decision support system that will enable the marketing manager to allocate the limited funds in such a way as to get the maximum out of the advertisement campaign. We provide an integer programming model to solve this problem. Note that the model we present below is built in such a way that reflects the requirements described in this project; however, the students are encouraged to generalize the problem.

Integer Programming Model

We use the following notation:

R	the total budget available
a_i	the expected number of exposures obtained by placing an advertisement in magazine A for the i -th time interval (<i>note</i> : interval 1 – up to 6 days; interval 2 – up to 10 days; interval 3 – up to 16 days)
b_i	the expected number of exposures obtained by placing an advertisement in magazine B for the i -th time interval
c_{ai}	the cost of placing an advertisement in magazine A during the i -th time interval
c_{bi}	the cost of placing an advertisement in magazine B during the i -th time interval.

The decision variables are as follows:

x_i	a binary variable that takes the value 1 if an advertisement is placed with magazine A during the i -th time interval, and 0 otherwise,
y_i	a binary variable that takes the value 1 if an advertisement is placed with magazine B during the i -th time interval, and 0 otherwise.

$$\max : \sum_{i=1}^3 (a_i x_i + b_i y_i)$$

Subject to :

$$\sum_{i=1}^3 (c_{ai} x_i + c_{bi} y_i) \leq R \quad (1)$$

$$\sum_{i=1}^3 x_i \leq 1 \quad (2)$$

$$\sum_{i=1}^3 y_i \leq 1 \quad (3)$$

$$x_i, y_i \in \{0,1\} \quad i = 1, \dots, 3. \quad (4)$$

The objective of this model is to maximize the total number of exposures. The first set of constraints shows that the amount spent in this advertisement campaign is limited by the budget. The second and third set of constraints show that at most one time interval will be selected to advertise in each magazine. The reason is that the cost paid per day decreases as we go from the 1st advertisement interval (up to 6 days) to the 2nd and from the 2nd to the 3rd interval, which has the smallest daily rates. The final set of constraints shows that our decision variables are binary variables.

Excel Spreadsheets

1. Build a spreadsheet that presents the expected number of exposures obtained by placing an advertisement in magazines A and B during different time intervals.
2. Build a spreadsheet that presents the cost of placing an advertisement in magazines A and B during different time intervals.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions for designing this form. Insert a frame that has two option buttons. The user has the choice to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked-on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, two tables appear with dimensions 2 by 6 where the user types in the expected exposures (for both magazines and all time intervals) and the corresponding cost of advertising.

Insert a text box where the user can type in the budget allocated to the advertisement campaign. Insert a command button that, when clicked –on, submits the problem data, uses the Excel solver to solve the corresponding integer programming problem, and opens Form 3, described below.

3. Build a form to present the results of the data analyses. The following are suggestions to help you design this form.

- a. Insert a frame titled “Reports” that has a number of option buttons to enable the user to select to open one of the summary reports described below.
- b. Insert a frame titled “Sensitivity Analysis” that includes the following: (i) a checkbox to allow the user to select whether to perform a sensitivity analysis with respect to the budget assigned to this project; (ii) two combo boxes that allow the user to choose a magazine and a time period for the purpose of the sensitivity analysis. For example, the user might be interested to know the sensitivity of the final solution with respect to the estimated response from advertising during time interval i in magazine j ($j = A, B$). Insert a command button that, when clicked –on, performs the sensitivity analysis with respect to the selected parameters and prompts the user to open the corresponding report.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the optimal solution (the magazine(s) to advertise in and the corresponding time interval) from the optimization model.
2. Report the cost of the optimal solution.
3. Report the results from the sensitivity analysis.

Reference

Winston, L.W., “Operations Research: Applications and Algorithms.” *Duxbury Press*, 3rd Ed., 1994.

Support System for the Multi-Period Capital Budgeting Problem

Problem Description

Capital budgeting is a procedure that allocates the resources of a business to prospective projects under consideration. Projects usually require initial investments that may span more than a single time period. Companies expect future returns from their investments. The objective is to maximize the net return from these investments. The available resources of a company are usually limited, and that makes the allocation process challenging. When analyzing the possible alternatives (projects), companies consider the fact that some projects are mutually exclusive and some are mandatory inclusive projects. That means only one of the mutually exclusive projects should be selected, and if one project is selected, all its mandatory inclusive projects should be selected as well.

The aim of this project is to build a decision support system that would enable a company to identify potential projects to invest in. Below we present an integer programming model to solve the multi-period capital budgeting problem.

Integer Programming Model

We use the following notation:

I	the total number of projects available
R_i	the net return from project i
T	the length of the planning horizon
B_t	the amount of money available for investment in period t ($t = 1, \dots, T$)
F_{it}	the investment made on project i in period t
M_j	the j^{th} set of mutually exclusive projects ($j = 1, \dots, J$ and $J \leq I$),
E_l	the l^{th} set of mandatory inclusive projects ($l = 1, \dots, L$ and $L \leq I$).

Note that R_i presents the net present value of the cash inflows and outflows for project i .

Our decision variables are as follows:

x_i	is a binary variable that takes the value 1 if project i is selected and 0 otherwise
y_l	is a binary variable that takes the value 1 if set l of mandatory inclusive projects is selected and 0 otherwise.

The objective of this problem is to maximize the net return from the investment. The first set of constraints shows that the total amount invested in each period should not exceed the budget available. The second set of constraints shows that at most one of the projects that are mutually exclusive should be selected. The third and fourth set of constraints show that if one of the projects is selected, all its mandatory inclusive projects should be selected as well. The fifth set of constraints is the integrality constraints.

$$\max : \sum_{i=1}^I R_i x_i$$

Subject to :

$$\sum_{i=1}^I F_{it} x_i \leq B_t \quad t = 1, \dots, T, \quad (1)$$

$$\sum_{i \in M_j} x_i \leq 1 \quad j = 1, \dots, J, \quad (2)$$

$$\sum_{i \in E_l} x_i = L y_l \quad l = 1, \dots, L, \quad (3)$$

$$x_i \leq y_l \quad i \in E_l, l = 1, \dots, L, \quad (4)$$

$$x_i \in \{0, 1\} \quad i = 1, \dots, I. \quad (5)$$

Excel Spreadsheets

1. Build a spreadsheet that presents the expected cash flows generated from each investment during the planning horizon.
2. Build a spreadsheet that presents the budget available for investment in each time period.
3. Build a spreadsheet that presents the funds required for each investment during the planning horizon.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions for designing this form. Insert a frame that includes two option buttons. The option buttons enable the user to select whether to read the data from a file or manually enter the data in the database. Include a command button that, when clicked on, performs these actions.
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, two text boxes appear where the user types in the total number of projects and the length of the planning horizon. Once this information is submitted, three tables appear. The first table (with dimensions T by I) enables the user to type in the expected cash flows from each investment during the planning horizon. The second table (with dimensions T by 1) enables the user to type in the budget available for investment in each time period. The third table (with dimensions T by I) enables the user to type in the funds required for each investment in each period.

Insert a frame titled "Select the Mutually Exclusive Projects." The frame includes a list box and a command button. The list box presents all the available projects. The user should be able to select at least one project at a time from this list. The user clicks on the command button to submit the mutually exclusive projects selected from the list. Insert a frame named "Select the Mandatory Inclusive Projects." The frame includes a list box and a command button. The list box presents all the available projects. The user should be able to select at least one project at a time from the list. The user clicks the

command button to submit the mandatory inclusive projects selected from the list. Insert a command button that, when clicked on, submits the problem data, uses the Excel solver to solve the corresponding integer programming problem, and opens Form 3, described below.

3. Build a form to present the results of the data analyses. The following are suggestions to help you design this form.
 - a. Insert a frame titled "Reports" that has a number of option buttons to enable the user to choose to open one of the summary reports described below.
 - b. Insert a frame titled "Sensitivity Analysis." Include the following controls in this frame:
 - i. Two text boxes where the user can type in a time period and a project name.
 - ii. Option buttons that allow the user to select whether to perform a sensitivity analysis with respect to the budget available in a time period t , investment required by project i in period t , etc.
 - iii. Insert a command button that, when clicked on, performs the sensitivity analysis with respect to the selected parameter and prompts the user to open the corresponding report. For example, the user may choose to perform a sensitivity analysis with respect to changes on the funds required by project i in period t .

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report a list of the projects that should be selected. For each of the selected projects, present the investments required and the net return.
2. Report the maximum return from the investments.
3. Report the results from the sensitivity analysis.

Reference

Winston, L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

The Central Limit Theorem is a theorem used widely in statistics. Formally, it is stated as follows: Let X_1, X_2, \dots, X_n be a random sample of size n taken from a population (either finite or infinite) with mean μ and finite variance σ^2 . Let \bar{X} be the sample mean. The limiting form of the distribution of $z = \frac{\bar{X} - \mu}{\sigma\sqrt{n}}$, $F_n(z)$, approaches $\Phi(z)$ as n approaches infinity:

$$\lim_{n \rightarrow \infty} F_n(z) = \Phi(z).$$

Where, $\Phi(z)$ is the distribution function of a normal random variable with $\mu = 0$ and $\sigma^2 = 1$ (standard normal distribution).

The Central Limit Theorem tells us that if we are sampling from a population that has an unknown distribution, the distribution of the sample mean will be normal with mean μ and variance σ^2/n , if the sample size n is large.

The aim of this project is to build a support system that will be used in a statistics course to help students understand the fundamentals of the Central Limit Theorem.

Example

The following is an example that demonstrates how the Central Limit Theorem works. Let Y be the outcome from tossing a die. Note that Y is uniformly distributed. There is a equal probability (1/6) that Y takes any of the values in set $S=\{1,2,3,4,5,6\}$. The mean value μ of Y is 3.5, and the variance is $\sigma^2 = 35/12$. If S_n is the sum of outcomes when n dices are rolled, and if n is "large," the distribution of random variable

$$\gamma = \frac{S_n - n\mu}{\sigma\sqrt{n}},$$

should approximate the standard normal distribution. As a result, S_n itself should have a normal distribution with mean $3.5*n$ and variance $35*n/12$.

The system we build allows the computer to randomly generate the outcome from tossing a die and calculates S_n , γ and the corresponding means and standard deviations.

User Interface

1. Build a welcome form.
2. Build a form that enables the user to perform the experiment described above. The following are suggestions for designing this form.
 - a. Insert a text box where the user types in the sample size n .
 - b. Insert a text box where the user types in the number of times the dice should be tossed, m .

- c. Insert a command button that, when clicked on, does the following:
 - i. Randomly generates n integer numbers uniformly distributed in the interval $[1, 6]$ using Excel functions.
 - ii. Calculates the sum (S_n) of the numbers generated.
 - iii. Calculates $\gamma = \frac{S_n - n\mu}{\sigma\sqrt{n}}$.
 - iv. Records the results in the summary table described in part 2.d.
 Repeat steps 2.c.i to 2.c.iv m times.

d. Build a table that records the following summary results: run number, S_n and γ .

- e. Insert a command button that, when clicked on, does the following:
 - i. Calculates the mean of S_n and γ using the data in Table 2.d.
 - ii. Calculates the variance of S_n and γ using the data in Table 2.d.

3. Insert a summary table where the user records the mean and variance of S_n and γ for different values of n .

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Plot the probability curve for the sampling distribution of the sample mean for a single die throw ($n = 1$).
2. Plot the probability distribution curves for the sampling distribution of the sample mean for all of the incremental sample sizes ($n = 2, \dots, n^*$).
3. If a curve-fitting feature is available in MS Excel, then for each of the above plots, calculate the probability that the plot conforms to the normal distribution, and report on what you find.
4. Present the probabilities obtained in each of the above incremental experiments in tabular form.

Reference

Montgomery, D.C., Runger, G.C., "Applied Statistics and Probability for Engineers," 3rd Ed., John Wiley & Sons, 2003.

32 Game of Words

Problem Description

This game will help the students to enhance their programming skills as well as improve their vocabulary. The object of the game is to fill in the blanks to complete the word selected by your opponent. The players may choose either to play the game until the player guesses the word or to limit the number of guesses by a pre-specified number. Below we present a simple algorithm that can be used to model this game.

Algorithm 1

In this algorithm we use the following notation:

w_p	the word submitted by the p -th player. $W = \{w_p \mid p = 1, \dots, P\}$.
w_{pm}	the m -th letter of the word assigned to player p
$L(w_p)$	the number of letters of word w_p

Step 1: Initialization

Input N (total number of guesses)
 Input M (length of the word)
 Input P (total number of players)

Step 2:

For $p = 1$ to P
 Input word w_p
 Assign w_p to player j ($j = 1, \dots, P, p \neq j$).

Step 3:

```

For  $n = 1$  to  $N$ 
  For  $p = 1$  to  $P$ 
    Input letter  $l$ 
    For  $m = 1$  to  $M$ 
      If letter  $l = w_{pm}$ 
        Remove letter  $l$  from word  $w_p$ 
        Print "You have a match!"
        If  $L(w_p) = 0$ 
          Then STOP, player  $p$  is the winner
        Else
          Print "No match!"
           $m = m + 1$ 
           $p = p + 1$ 
         $n = n + 1$ 
    For  $p = 1$  to  $P$ 
      Find  $Z = \min(L(w_p))$ 
      Player  $Z$  is the winner.
      Report the word  $w_z$ .
  
```

Algorithm 2**Step 1:** Initialization.

Input N (total number of guesses)
 Input M (length of the word)
 Input P (total number of players)

Step 2:

For $p = 1$ to P
 Input word w_p
 Assign w_p to player j ($j = 1, \dots, P, p \neq j$).

Step 3:

```

For  $n = 1$  to  $N$ 
  For  $p = 1$  to  $P$ 
    Input word  $l$ 
    If word  $l = w_p$ 
      Then STOP, player  $p$  is the winner
      Print "Congratulations! Word  $l$  is a match!"
    Else
      Print "No match!"
   $p = p + 1$ 
 $n = n + 1$ 

```

User Interface

1. Build a welcome form.
2. Create a data entry form. The following are suggestions to help you design the form. In this form insert a frame titled "Input the data" that includes the following:
 - a. A text box where the user types in the total number of guesses (N) allowed.
 - b. A text box where the user types in the length of the words (M) selected by the players.
 - c. A text box where the user types in the total number of players (P).
 - d. As the total number of players is submitted, $2 \cdot P$ text boxes appear where the players type in their names and the selected word.
 - e. A command button that enables the players to submit the words selected. After the submission, the text boxes become empty.
 - f. A frame titled "Choose an algorithm." The frame includes two option buttons that enable the user to choose to play the game using one of the two algorithms described above.
 - g. Insert a frame titled "Play the game." The frame includes the following controls:
 - i. P sub-frames. The title of each sub-frame is the name of the player who will use the corresponding frame. Include a text box in each sub-frame where the user can type in the letter/word chosen.
 - ii. If the user is playing the game using the first algorithm, include in each sub-frame a table consisting of one row and M columns. As the user finds the

letters of the word, these letters appear in the appropriate position on the table. For example, if the word the player has to figure out is “student” and the player found two correct letters of this word (say t and d), the table would show the following: _ t _ d _ _ t.

- iii. If the user is playing the game using the second algorithm, include a list that presents all the words submitted by the player so far.
- iv. Insert a command button that, when clicked on, submits the letter/word and prompts the user whether the letter/word chosen was a match or not.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the name of the winner of the game, the word that player successfully guessed, and the name of the player who submitted the winning word.
2. Build a summary report that presents the following information about the games played using algorithm 1: the name of the winner, the total number of players, the total number of guesses made until we had a winner, and the winning word.
3. Build a summary report that presents the following information about the games played using algorithm 2: the name of the winner, the total number of players, the total number of guesses made until we had a winner, and the winning word.

Problem Description

In this project we use simulation as a decision support tool. The aim of the project is to use simulation to analyze the current status of the check-in process in an airport and identify ways to improve its performance. The following are issues faced by the managers of the airport:

1. Most of the flights are scheduled in the morning; as a result there is large number of passengers entering the airport in the morning compared to late afternoon or evening. Scheduling working hours for employees becomes difficult, as there is a variance in the workload in different hours of the day.
2. Long queues in front of the check-in counters disturb the traffic of passengers.
3. Long waiting times to be served by a check-in clerk frustrates passengers.
4. Long passenger waiting times delay flights.

Below we describe a model to simulate the check-in process. The student should collect data, run the simulation model, and then use the results from the simulation runs to identify ways to improve the current process.

For details about building and analyzing a simulation model, we refer the students to Law and Kelton (2000) and Winston (1994).

Simulation Model

The system we are simulating works as follows: passengers arrive at the check-in line. There is a maximum of ten check-in counters in the airport. All the counters are open during the peak hours. There are two different types of passengers entering the line: passengers who have luggage and passengers who don't. It takes longer to serve passengers who have luggage. The served passengers walk to the gate. Depending on the location of the boarding gate and the amount of carry-on luggage, the time it takes to reach the gate varies. The passengers can have at most two carry-on luggage items. Passengers are required to check in at least half an hour prior to boarding time.

Step 1

Collect data about the passengers' arrival and service time at a check-in counter and the time it takes to reach a gate from the check-in area. Use this data to identify the distribution (and the corresponding parameters) of the inter-arrival time of the passengers in the system, service time, and walking time to the gate. This information will be used to simulate the system.

Step 2

Simulate the system. Build a model to simulate the system. To learn about building a simulation model using Excel, we refer the students to Winston (1994). Collect the following statistics: average/minimum/maximum queue length, average/minimum/ maximum waiting time in the queue, average total time in the system (t), average utilization of the check-in clerk, and average/minimum/maximum service time at a check-in counter.

Excel Spreadsheets

Build a spreadsheet that presents the following data collected about each passenger: amount of luggage, amount of carry-on luggage, arrival time at the check-in counter, service start time, departure time from check-in counter, and arrival time at the gate.

User Interface

1. Create a welcome form.
2. Create a data analysis form. The following are suggestions to help you design this form.
 - a. Insert a frame titled “Analyze the Input Data.” The frame includes a command button that, when clicked on, opens the spreadsheet where the data about inter-arrival time, service time, and walk time are recorded. The frame includes a second command button that, when clicked on, creates a histogram using the data collected and calculates the mean, standard deviation, etc. for inter-arrival, service, and walk times. This information helps the user to identify a distribution for inter-arrival, service, and walk times.
 - b. Insert a frame titled “Run the Simulation.” The frame includes the following:
 - i. A text box where the user types in the total number of simulation runs.
 - ii. A text box where the user types in the length of a simulation run.
 - iii. A text box where the user types in the length of the warm-up period.
 - iv. Three combo boxes that enable the user to select a distribution for the inter-arrival time, service time, and walk time. Upon the selection of a distribution, text boxes appear where the user types in the corresponding parameters (such as mean, standard deviation, etc.).
 - v. A command button that, when clicked on, takes as input the information provided by the user; randomly generates passenger arrival, service, and walking times; and after the completion of the simulation runs, opens Form 3, described below.
3. Create a form that summarizes the results of the simulation study and allows the user to perform a sensitivity analysis. Insert two frames in this form. The first frame, titled “Reports,” allows the user to select to open one of the reports presented below. The second frame, called “Output Analysis,” enables the user to analyze the system and identify better scenarios to manage the check-in process. This frame includes the following:
 - a. A text box where the user types in the total number of check-in counters. Currently there are ten check-in counters, and the user can analyze the impact of increasing (or decreasing) the number of check-in counters on the total performance of the system.
 - b. A table where the user types in the total number of clerks working at the check-in counters during different hours of the day. More clerks are needed during the peak hours. The management is interested in identifying a working schedule for the employees that increases system performance while minimizing the labor costs.
 - c. Two option buttons that allow the user to choose whether there will be a single queue in the system where the customers wait to be served or as many queues as the number of check-in clerks working.

Insert a command button that, when clicked on, takes as input the information submitted by the user in parts 3.a, 3.b, and 3.c, runs the simulation, and reports the corresponding statistics.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the following statistics:
 - a. The average/minimum/maximum number of passengers in the system.
 - b. The average/minimum/maximum waiting time.
 - c. The average/minimum/maximum serving time by passenger type.
 - d. The average/minimum/maximum walking time by passenger type.
 - e. The average/minimum/maximum queue length.
 - f. The average/minimum/maximum time spent in the system.
 - g. The probability that a customer would spend more than 20 minutes waiting in the queue.
 - h. The average utilization of a clerk.
 - i. The average number of passengers served during the peak hours.
 - j. The average number of passengers served during the non-peak hours.
2. Report the results from the sensitivity analysis. For each scenario, present the following: the average/minimum/maximum queue length, waiting time, service time, time spent in the system, and clerk utilization.
3. Give a graphical representation of the queue length during the day.
4. Give a graphical representation of clerk utilization during the day.

Reference

- Law, A.M., Kelton, W.D., "Simulation Modeling and Analysis," 3rd Ed., *McGraw-Hill*, 2000.
- Winston, L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

Pricing products is a challenging process that managers have to deal with quite often. Different factors impact the price of a product. For example, costs and demand affect price. Normally, the price we pay for a product is higher than the cost of manufacturing and delivering the product. Formally, the price of a product is determined by demand and supply. If the demand for a product is high and its supply is low, the customer would be willing to pay a higher price. Pricing the products well is crucial for a company, as it directly affects the profits. Higher prices imply higher profit margins. However, if the price of a product is too high, it will negatively affect its demand, and as a result it will negatively affect profits of the company.

The price of a product varies with time. Reasons for such a fluctuation could be changes in the demand for the product, changes in the supply because of competition, and changes in the production and delivery costs. The aim of this project is to build a decision support system that will enable the companies to price their products considering important factors such as demand and costs. Below we present a mathematical model that can be used to price products.

Mathematical Model

We use the following notation:

I	total number of products
p_{1i}	price of product i on-season
p_{2i}	price of product i off-season
c_i	cost of producing product i
m_1	production capacity on-season
m_2	production capacity off-season.

The decision variables are as follows:

d_{1i}	on-season demand for product i
d_{2i}	off-season demand for product i .

The demand functions on- and off-season are described below.

$$d_{1i} = \alpha_{1i} - \beta_{1i} * p_{1i} + \gamma_{1i} p_{2i}$$

$$d_{2i} = \alpha_{2i} + \beta_{2i} * p_{1i} - \gamma_{2i} p_{2i}$$

Where, coefficients α_{1i} , α_{2i} present the demand for product i if the on- and off-season prices were equal to zero; β_{1i} , β_{2i} present the change of demand with respect to one unit change of the on-season price, and γ_{1i} , γ_{2i} present the change of demand with respect to one unit change of the off-season price.

The following is a mathematical formulation of the problem:

$$\max : \sum_{i=1}^I (d_{1i} p_{1i} + d_{2i} p_{2i} - c_i (d_{1i} + d_{2i}))$$

Subject to :

$$\sum_{i=1}^I d_{1i} \leq m_1 \quad (1)$$

$$\sum_{i=1}^I d_{2i} \leq m_2 \quad (2)$$

$$d_{1i} = \alpha_{1i} - \beta_{1i} * p_{1i} + \gamma_{1i} p_{2i} \quad \text{for } i = 1, \dots, I, \quad (3)$$

$$d_{2i} = \alpha_{2i} + \beta_{2i} * p_{1i} - \gamma_{2i} p_{2i} \quad \text{for } i = 1, \dots, I, \quad (4)$$

$$d_{1i}, p_{1i}, d_{2i}, p_{2i} \geq 0 \quad \text{for } i = 1, \dots, I. \quad (5)$$

The objective is to maximize profits. The first and second sets of constraints show that total demand is bounded by production capacity limits. The third and fourth sets of constraints present the demand function on- and off-season. The last set of constraints is the non-negativity constraints.

Excel Spreadsheet

1. Build a spreadsheet that presents historical data about demand and prices on- and off-season for all the products produced.
2. Build a spreadsheet that presents the cost of producing the products.

User Interface

1. Build a welcome form.
2. Build a data analysis form. The following are suggestions to help you design this form:
 - a. Insert a text box where the user types in the total number of products considered.
 - b. Insert a frame titled "Regression Analysis." The frame includes a text box where the user types in the location of Spreadsheet 1. Insert a command button that, when clicked on, uses the regression analysis tools of Excel to identify the relationship between prices and demand. The results from this analysis are used to identify coefficients α_{1i} , α_{2i} , β_{1i} , β_{2i} , γ_{1i} and γ_{2i} ($i = 1, \dots, I$).
 - c. Insert two text boxes where the user types in production capacity on- and off-season.
 - d. Insert a command button that, when clicked on, solves the mathematical formulation of the pricing problem using the Excel solver and opens Form 3, described below.
3. Build a form to present the results of this study. In this form include two frames. The first frame has a number of option buttons that enable the user to choose to open any of the reports described below. The second frame, titled "Sensitivity Analysis," has a number of option buttons that enable the user to choose a parameter to perform a sensitivity analysis. One can perform a sensitivity analysis with respect to production capacity, demand, price, costs, etc.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record

navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the optimal price for each product.
2. Report the optimal profit found from solving the mathematical formulation of the problem.
3. Report the results from the sensitivity analysis.
4. Report the results from regression analysis. Report the values found for the coefficients α_{1i} , α_{2i} , β_{1i} , β_{2i} , γ_{1i} and γ_{2i} .
5. Graph the relationship between on-season demand and on-season price for a particular product.
6. Graph the relationship between off-season demand and off-season price for a particular product.

Reference

Montgomery, D.C., Runger, G.C., "Applied Statistics and Probability for Engineers," 3rd Ed., John Wiley & Sons, 2003.

Winston, L.W., "Operations Research: Applications and Algorithms." Duxbury Press, 3rd Ed., 1994.

Problem Description

A new challenge that businesses are facing these days is increased customer expectations. Because of the competition, customers expect products of good quality at a good price. This fact pressures companies to find ways to improve the quality of their products and processes. The aim of this project is to build a decision support system that will enable the user to identify ways to improve product quality while keeping costs down.

For the purpose of this project we give an example, propose a mathematical model, and demonstrate how to build a support system to manage the quality problems. The objective is to minimize costs while keeping product quality within certain limits.

Example

“Rock Co.” is a company that produces Flowable Fill, a ready-mix product used by many construction companies. The quality of this product is determined by its curing time, strength, excavatability, etc. The product is produced using cement, fly ash, admixtures, etc. The quality characteristics of this product are influenced by the amount of raw materials used. The engineers have identified models that express the relationship between the quality characteristics of the final product and the quantity of raw materials used. The managers are interested in identifying the best combination of ingredients to produce the final product.

Mathematical Model

We use the following notation:

I	total number of products
J	total number of ingredients
K	total number of product characteristics (e.g., strength, curing time, etc.)
d_i	demand for product i
c_j	unit production cost of ingredient j
q_{ik}	level of characteristic k on product i
B_j	total amount of ingredient j available
lb_{ik}	minimum requirement of characteristic k in product i
ub_{ik}	maximum requirement of characteristic k in product i .

The decision variables are as follows:

u_{ij}	amount of ingredient j needed to produce one unit of product i .
----------	--

The level of characteristic k on product i depends on the amount and mixture of the ingredients used to produce the final product. Below we present the equation that is used to quantify q_{ik} .

$$q_{ik} = \alpha_{ik} + \sum_{j=1}^J \left(\beta_{ijk} * u_{ij} + \sum_{l=j}^J \delta_{ijkl} * u_{ij} * u_{il} \right). \quad (1)$$

Where α_{ik} , β_{ijk} , δ_{ijkl} are coefficients that demonstrate the relationship between product quality and quantity of the ingredients used.

The following is a mathematical formulation of this problem.

$$\min : \sum_{i=1}^I \sum_{j=1}^J c_j u_{ij}$$

Subject to :

(1)

$$\sum_{i=1}^I u_{ij} \leq B_j \quad j = 1, \dots, J, \quad (2)$$

$$lb_{ik} \leq q_{ik} \leq ub_{ik} \quad i = 1, \dots, I; k = 1, \dots, K, \quad (3)$$

$$\sum_{j=1}^J u_{ij} = d_i \quad i = 1, \dots, I. \quad (4)$$

$$u_{ij} \geq 0 \quad i = 1, \dots, I; j = 1, \dots, J. \quad (5)$$

The objective is to minimize costs. The second set of constraints shows that the amount of ingredient j used to produce the final products is bounded by the amount of ingredient j available. The third set of constraints shows that product characteristics should be within certain limits to ensure good quality. The fourth set of constraints shows that demand should be satisfied. The last set of constraints is the non-negativity constraints.

Excel Spreadsheets

1. Build a spreadsheet that presents historical data about the amount of ingredient j ($j = 1, \dots, J$) used to produce one unit of product i (u_{ij}). For each combination of ingredients in product i , present the corresponding level of characteristic k achieved (q_{ik}).
2. Build a spreadsheet that presents the demand for all final products.
3. Build a spreadsheet that presents the unit cost and availability of each ingredient.
4. Build a spreadsheet that presents the upper and lower bound on the requirement about the quality (characteristic k , for $k = 1, \dots, K$) of each product.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. In this form insert two option buttons. The user has the choice to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.

- b. If the user chose to enter the data manually, three text boxes appear where the user types in the following: the total number of products considered, the total number of ingredients, and the total number of product characteristics. Upon submission of this information, four tables appear. The user types in these tables the information presented in the spreadsheets described above.
3. Build a data analysis form. The following are suggestions to help you design this form.
 - a. Include a text box where the user types in the location of Spreadsheet 1. Insert a command button that, when clicked on, uses the regression analysis tools of Excel to identify the relationship between product characteristics and the amount of each ingredient used. The results from this analysis are used to identify coefficients α_{ijk} , β_{ijk} , and δ_{ijkl} ($i = 1, \dots, I$; $j = 1, \dots, J$; $k = 1, \dots, K$; $l = j, \dots, J$).
 - b. Insert a command button that, when clicked –on, solves the problem using the mathematical model presented above and opens Form 4, described below.
 4. Build a form to present the results of this study. Include a frame that has a number of option buttons to allow the user to open the reports described below. Include another frame, titled “Sensitivity Analysis,” that has a number of option buttons to allow the user to select a parameter for the sensitivity analysis. For example, the user might be interested in identifying the sensitivity of the optimal solution with respect to the availability of an ingredient (for example, cement).

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Build a report that presents the total cost of producing the final products that have quality characteristics within the pre-specified levels.
2. Report the name of the ingredients and the corresponding quantities that should be used to produce product i ($i = 1, \dots, I$) in such a way that total costs are minimized and the desired quality level is maintained.
3. Report the results from the sensitivity analysis.

Reference

Montgomery, D.C., Runger, G.C., “Applied Statistics and Probability for Engineers,” 3rd Ed., John Wiley & Sons, 2003.

Winston, L.W., “Operations Research: Applications and Algorithms.” Duxbury Press, 3rd Ed., 1994.

Problem Description

Aggregate production planning is the process of determining the optimal production quantity, workforce size, overtime, machine capacity, inventory level, amount to be subcontracted, and backlogging that optimizes production costs in a manufacturing environment. The aim of this project is to build a decision support system that enables the managers to prepare their production plans. For the purpose of this project, we present a simple example and provide an optimization model that solves the corresponding aggregate production planning problem.

“Red Tomato Tools” is a manufacturer of gardening equipment. Their operations consist of assembling parts into multipurpose gardening tools. Their capacity is determined mainly by the size of their workforce. The demand for their product is highly seasonal. Demand for gardening tools is high in spring and almost zero in winter. Demand seasonality makes the process of planning production quite challenging. Below we present a mathematical model that enables us to solve this planning problem. The model assumes no limitations in the amount of inventory carried, amount of final product subcontracted, and amount of stock-out.

Mathematical Model

We use the following notation:

T	length of planning horizon
d_t	demand in period t ($t = 1, \dots, T$)
α	total number of working hours per employee in period t
w_t	hourly salary in period t
β_t	total number of hours needed to produce one unit of the final product in period t
h_t	cost of hiring one employee at the beginning of period t
l_t	cost of laying off an employee at the beginning of period t
p_t	unit production (material) cost in period t
i_t	unit inventory holding cost in period t
s_t	unit stock-out cost in period t
c_t	cost of purchasing one unit of final product from a subcontractor in period t
o_t	unit overtime cost in period t .

The decision variables are as follows:

W_t	size of the workforce in period t
H_t	total number of employees hired in the beginning of period t
L_t	total number of layoffs in the beginning of period t

P_t	quantity produced in period t
I_t	inventory in the end of period t
S_t	unsatisfied demand in the end of period t
C_t	amount of final product produced by subcontractors in period t
O_t	amount of overtime used in period t .

The objective is to minimize the total cost. The total cost consists of labor, material, inventory holding, overtime, stock-out, subcontracting, hiring, and layoff costs.

$$\min : \sum_{t=1}^T (w_t \alpha_t W_t + o_t O_t + h_t H_t + l_t L_t + i_t I_t + s_t S_t + p_t P_t + c_t C_t)$$

Subject to :

$$W_{t-1} + H_t - L_t = W_t \quad (1)$$

$$\alpha_t W_t + O_t \geq \beta_t P_t \quad t = 1, \dots, T, \quad (2)$$

$$I_{t-1} + P_t + C_t = d_t + I_t - S_t + S_{t-1} \quad t = 1, \dots, T, \quad (3)$$

$$W_t, O_t, H_t, L_t, I_t, S_t, P_t, C_t \geq 0 \quad t = 1, \dots, T. \quad (4)$$

The first set of constraints is the workforce conservation constraints. The second set of constraints is the labor capacity limitation constraints. The third set of constraints is the inventory balance constraints. The final set of constraints is the non-negativity constraints.

Excel Spreadsheets

1. Build a spreadsheet that presents the expected unit costs (e.g., cost of materials, inventory holding cost, etc.) during each period of the planning horizon.
2. Build a spreadsheet that presents demand in each period of the planning horizon.
3. Build a spreadsheet that presents historical data about the amount of inventory.
4. Build a spreadsheet that presents historical data about the total number of employees, the number of employees hired, and the number of employees laid off.
5. Build a spreadsheet that presents historical data about the amount of final product subcontracted.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. In this form insert two option buttons. These option buttons allow the user to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, a text box appears where the user types in the total number of periods in the planning horizon. Upon submission of

this information, five tables appear. The user types in these tables the data required.

3. Insert a data analysis form. In this form insert a command button that, when clicked on, solves the problem using the mathematical model presented above and opens two frames. The first frame includes a number of option buttons that allow the user to open any of the reports described below. The second frame, titled "Sensitivity Analysis," has a number of option buttons that allow the user to select a parameter for the sensitivity analysis. The user is interested in identifying the sensitivity of the optimal solution with respect to demand, costs, etc.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the optimal workforce size, number of new hires, and number of layoffs, production quantity, inventory level, amount subcontracted, and amount of overtime.
2. Report the optimal total cost of production as found from solving the mathematical model.
3. Report the results of the sensitivity analysis.

Reference

Winston, L.W., "Operations Research: Applications and Algorithms." Duxbury Press, 3rd Ed., 1994.

37 Reliability Analysis

Problem Description

A dimension of the quality of a product's design is reliability. Reliability refers to the probability that the product will be functional when used. For example, reliability of a product being 95% means that 95 out of 100 products produced will perform as intended for a stated period of time, under specified operating conditions.

Failure rate is a measure used to ascertain reliability of a product or a process. For products that must be replaced because they are dysfunctional, a relevant measure of interest is the *mean time to failure* (MTTF), and for products that may be repaired and put to service again, a relevant measure of interest is *mean time between failures* (MTBF). In order to calculate MTTF and MTBF, ideally, we would consider a large number of products to be operated and tested until failure and record the time of the failure for each. The information about MTTF and MTBF helps ascertain reliability. However, it is time consuming and costly to collect enough data to build a probability distribution and cumulative distribution curve of time to failure or time between failures. One way we can deal with this problem is by analyzing a smaller set of data and identifying a distribution that approximates the distribution of time to failure, such as the Exponential, Weibull, or Gamma distributions. Once we have identified a distribution, we can easily calculate failure rate, MTTF, etc.

The aim of this project is to build a decision support system that would enable the user to estimate the reliability of products/processes. For the purpose of this project, we use a simple example and describe a number of statistical analysis tools that can be used.

Example

A company that produces electric bulbs is concerned about the reliability of this product. The inspection team has already collected data about bulb failures. We use statistical analysis tools to calculate the reliability of this product. Below we present a model to use for this purpose.

Model

The following notation is used:

$f(t)$	probability of failure at time t
$F(t)$	probability of failure up to time t
$R(t)$	probability of surviving up to time t ($R(t) = 1 - F(t)$)
$r(t)$	failure rate at time t ($r(t) = f(t)/R(t)$).

In the case that the distribution function of failure is approximated by the Exponential distribution:

$$\begin{aligned}
 f(t) &= \lambda e^{-\lambda t}, \\
 F(t) &= 1 - e^{-\lambda t}, \\
 R(t) &= e^{-\lambda t}, \\
 r(t) &= \lambda.
 \end{aligned}$$

$\frac{1}{\lambda}$ is the mean and standard deviation of the exponential distribution. Note that if the mean and standard deviation for a data set are not equal, the corresponding distribution is not exponential. In the context of our problem, λ is the failure rate and $\frac{1}{\lambda}$ is the mean time between failures. Note that the failure rate is constant.

In the case that the distribution function of failure is approximated by the Weibull distribution:

$$\begin{aligned}
 f(t) &= \beta \alpha^{-\beta} t^{\beta-1} e^{-\left(\frac{t}{\alpha}\right)^\beta}, \\
 F(t) &= 1 - e^{-\left(\frac{t}{\alpha}\right)^\beta}, \\
 R(t) &= e^{-\left(\frac{t}{\alpha}\right)^\beta}, \\
 r(t) &= \beta \alpha^{-\beta} t^{\beta-1}.
 \end{aligned}$$

Parameters α and β define the scale and shape of this distribution. The expected value for

Weibull distribution is $E(t) = \frac{\alpha}{\beta} \Gamma\left(\frac{1}{\beta}\right)$, and the variance is

$$V(t) = \frac{\alpha^2}{\beta} \left[2\Gamma\left(\frac{2}{\beta}\right) - \frac{1}{\beta} \left(\Gamma\left(\frac{1}{\beta}\right) \right)^2 \right].$$

To learn more about Exponential and Weibull distributions, we refer the students to Mood et al. (1974).

Excel Spreadsheet

Build a spreadsheet that presents the data collected about each bulb investigated: the time the bulb was set on operation and failure time. Calculate time to failure.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. In this form insert two option buttons. These option buttons allow the user to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, a text box appears where the user types in the total number of observations (n). Upon submission of this information,

a table appears. The user types in this table the start time and failure time of the bulbs observed.

3. Build a data analysis form. The following are suggestions to help you design this form.
 - a. Insert a command button that, when clicked –on, creates a histogram using the data collected and calculates and presents the corresponding mean, standard deviation, etc. This information helps the user to identify a distribution for time to failure.
 - b. Insert a combo box that enables the user to select a distribution of failure time (e.g., Exponential, Weibull, Gamma, etc).
 - c. Insert a command button that, when clicked –on, performs a goodness-of-fit test (e.g., the Chi-square test) and presents the corresponding results. This test indicates whether the distribution selected by the user approximates the real distribution of time to failure. To learn more about the Chi-square test, we refer the students to Winston (1994).
 - d. Insert a frame titled “Reliability Analysis.” The frame includes the following:
 - i. Insert a combo box that enables the user to select a distribution (e.g., Exponential, Weibull, or Gamma) to approximate the distribution of time to failure. Upon the selection of a distribution, text boxes appear where the user types in the parameters of the selected distribution. For example, if Exponential distribution is chosen, a single text box appears where the user types in the value of λ .
 - ii. Insert a command button that, when clicked –on, takes as input the information provided by the user and calculates MTTF.
 - iii. Insert a text box where the user can type in a time period, t .
 - iv. Insert a command button that, when clicked on, returns the following: the probability of failure at time t , the survival rate at time t , the total number of failures up to time t , the failure rate at time t , etc.
4. Build a form that allows the user to access the reports created.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Plot a histogram of time to failure.
2. Plot the cumulative distribution function of time to failure.
3. Plot the failure rate function.
4. Plot the cumulative distribution function of survival ($R(t)$).
5. Report the results from the data analyses (e.g., mean and standard deviation of the observations, the results of the Chi-square test, etc.)
6. Report the results from reliability analysis.

Reference

Mood, A.M., Graybill, F.A., Boes, D.C., "Introduction to the Theory of Statistics." *McGraw Hill*, 3rd Ed., 1974.

Winston, L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

A well-known solution approach to optimization problems is Lagrangean relaxation. Lagrangean relaxation can be used to solve linear, non-linear, integer programming, and combinatorial optimization problems.

In this project we present the constraint shortest path problem, give a Lagrangean relaxation-based algorithm, and finally build a decision support system that would enable the user to solve the problem using the Lagrangean relaxation method. To learn more about the constraint shortest path problem and the Lagrangean relaxation method, we refer students to Ahuja *et al.* (1993).

Problem Formulation

The constraint shortest path problem (P) is a shortest path problem with additional capacity constraints (constraints (2)). Below we present a formulation of this problem.

$$\min : \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (P)$$

Subject to :

$$\sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} x_{ij} = \begin{cases} 1 & \text{for } i = 1 \\ 0 & \text{for } i \in N - \{1, n\} \\ -1 & \text{for } i = n \end{cases} \quad (1)$$

$$\sum_{(i,j) \in A} t_{ij} x_{ij} \leq T \quad (2)$$

$$x_{ij} \in \{0,1\} \quad \text{for all } (i,j) \in A. \quad (3)$$

Where, c_{ij} is the cost of using arc $(i, j) \in A$, n is the total number of nodes of the network, T is the capacity of the shortest path, and t_{ij} is the capacity usage coefficient for arc $(i, j) \in A$. The objective is to find the shortest path between nodes 1 and n of the network that does not violate the capacity constraints.

The Lagrangean relaxation method relaxes the problem by moving the complicating constraints to the objective function. The resulting optimization problem is the (unconstrained) shortest path problem. The solution to the relaxation problem is a lower bound for the capacitated shortest path problem.

The following is the Lagrangean relaxation problem ($LR(\mu)$):

$$\min : \sum_{(i,j) \in A} c_{ij} x_{ij} + \mu \left(\sum_{(i,j) \in A} t_{ij} x_{ij} - T \right) = \sum_{(i,j) \in A} (c_{ij} + \mu t_{ij}) x_{ij} - \mu T \quad (LR(\mu))$$

Subject to :

$$(1) \text{ and } (3)$$

is the Lagrangean multiplier.

The objective function value of $(LR(\mu))$ depends on the value of the multipliers μ . For any value of μ , $(LR(\mu))$ is a lower bound for P ($LR(\mu) \leq P^*$). We are interested in identifying the highest lower bound for P^* . We do that by identifying the multiplier μ that maximizes the Lagrangean relaxation problem.

$$LR^* = \max_{\mu} LR(\mu).$$

For any value of μ , $LR(\mu) \leq LR^* \leq P^*$. To solve problem LR^* we use the following approach. Let μ^0 be any initial choice of the Lagrangean multipliers. We determine the subsequent values μ^k for $k = 1, 2, \dots$, of the Lagrangean multipliers as follows:

$$\mu^{k+1} = \mu^k + \theta_k \left(\sum_{(i,j) \in A} t_{ij} x_{ij} - T \right). \quad (4)$$

Where,

$$\theta_k = \frac{\lambda_k [UB - LR(\mu^k)]}{\left\| \sum_{(i,j) \in A} t_{ij} x_{ij} - T \right\|}. \quad (5)$$

UB is an upper bound on the optimal objective function value P^* . Practitioners usually choose λ_k by starting with $\lambda_k = 2$ and then reducing λ_k by a factor of 2 if the best Lagrangean objective function value found so far has failed to improve in a specific number of iterations.

Lagrangean Relaxation Algorithm

Step 0

Set $k = 0$; $\mu^k = 0$; $\lambda_k = 2$;

- Find a feasible solution for problem $P^{(1)}$. Set UB equal to the corresponding objective function value.

Step 1

- Solve problem $LR(\mu_k)^{(2)}$.

- Keep the best lower bound found so far.

- Check if the solution to $LR(\mu_k)$ satisfies constraints (3) of P . If it does, this is a feasible solution to problem P as well. Calculate the corresponding objective function value.

- Keep the best upper bound found so far.

- If the best lower bound found is equal to the best upper bound, Stop. We have solved the capacitated shortest path problem.

Step 2

- If the solution to $LR(\mu_k)$ has not improved in the last g iterations (for example, $g = 5$), reduce λ_k by a factor of 2.

- Calculate θ_k .

- Calculate μ^{k+1} using (4).
- $k = k + 1$.
- If the solution to $LR(\mu_k)$ has not improved in the last G iterations (for example, $G = 10$), *Stop*. Report the best lower bound (solution of $LR(\mu_k)$) and upper bound (solution of P) found so far.
- Else go to Step 1.

(1) To find a feasible solution for problem P , one can follow a simple approach. Modify problem (P): replace c_{ij} with t_{ij} in the objective function and remove constraints (3). The modified problem is an un-capacitated shortest path problem. If the objective function value of the modified problem is less than or equal to T , the solution found is a feasible solution to problem P ; otherwise problem P is infeasible. One can use Dijkstra's algorithm to solve the shortest path problem. For a description of this algorithm, we refer the students to Ahuja et al. (1993).

(2) Problem $LR(\mu_k)$ is the classical shortest path problem. One can use Dijkstra's algorithm to solve this problem.

Excel Spreadsheets

Build a spreadsheet that presents the following data about the arcs of the network: arc number, tail node, head node, cost, etc.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. In this form insert two option buttons. These option buttons allow the user to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked –on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, two text boxes appear where the user types in the total number of arcs (m) and nodes (n) of the network. Upon submission of this information, a table appears where the user enters the following data about each arc: head node, tail node, cost, and capacity.

Insert a text box where the user can type in the capacity requirement (T) for this problem.

3. Build a form that allows the user to solve the problem and report the results. The following are suggestions to help you design this form. Insert a frame titled "Solve the Problem." This frame has two option buttons to allow the user to select either the Lagrangean relaxation-based algorithm described above or the Excel solver to solve the problem. Insert a command button that, when clicked on, solves the problem and opens two frames. The first frame, titled "Reports," has a number of option buttons that allow the user to open any of the reports described below. The second frame, titled "Sensitivity Analysis," has a number of option buttons that allow the user to select a parameter for the sensitivity analysis. The user is interested in identifying the sensitivity of the solution with respect to capacity requirement T , arc cost, etc.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Give a graphical representation of the solution to the constraint shortest path problem.
2. Present the results from the Lagrangean relaxation algorithm. For each iteration, present the following: k , μ^k , λ_k , θ_k , the best upper bound, the best lower bound, and $LR(\mu^k)$.
3. Present the results from the sensitivity analysis.

Reference

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., "Network Flows: Theory, Algorithms and Applications." *Prentice Hall*, 1993.

Problem Description

Managers deal with a number of decisions about processes, capacity, location, inventory, etc. The process of making a decision is complicated because of the uncertainties of the future. This implies that the future outcomes of alternatives that we consider today are in doubt.

The aim of this project is to build a decision support system that allows the user to make decisions under uncertainty. We describe a systematic approach that can be used by managers in the process of making decisions. To learn more about decision making under uncertainty, we refer the students to Krajewski and Ritzman (2002).

Solution approach

Step 1:

List the feasible alternatives.

Step 2:

List the events that have an impact on the outcome of each alternative but are not under managers' control.

Step 3:

Estimate the payoff for each alternative in each event.

Step 4:

Estimate the likelihood of each event using past data, executive opinion, or other forecasting methods.

Step 5:

Select a decision rule to evaluate the alternatives, such as choosing the alternative with the lowest expected cost or choosing the alternative with the maximum expected profits.

The following are decision rules to help the managers select an alternative.

Maximin: Choose the alternative that is the “best of the worst.” This rule is for the pessimist who anticipates the “worst case” return for each alternative.

Maximax: Choose the alternative that is the “best of the best.” This rule is for the optimist who anticipates the “best case” return for each alternative.

Laplace: Choose the alternative with the best-weighted payoff. To find the weighted payoff, give equal importance to each event. For example, if there are n events, the probability assigned to each event is $1/n$. This rule is for the realist.

Minimax: Choose the alternative that gives the best “worst regret.” Calculate a table of regrets in which the rows represent the alternatives and the columns represent the events. A regret is the difference between a given payoff and the best payoff in the same column. For an event, it shows how much is lost by picking an alternative to the one that is best for this event. The regret can be lost profit or increased cost, depending on the situation.

User Interface

1. Build a welcome form.

2. Build a data analysis form. The following are suggestions to help you design this form.
 - a. Insert two text boxes where the user types in the total number of events (n) and the total number of alternatives (m). Upon submission of this information a table appears. This table has dimensions m by n . The user types in this table the payoff of each alternative in each event. Label the rows by the name of alternatives. Label the columns by the name of events.
 - b. Insert a command button titled "Enter the Likelihood of Events." When the user clicks on this button, a table with dimensions 1 by n appears. The user types in this table the likelihood of each event, if this information is available.
 - c. Insert a frame named "Selection Rules." The frame includes five option buttons and a command button. The option buttons enable the user to select one of the following decision rules: maximin, maximax, Laplace, minimax regret and expected payoff. When the user clicks on the command button, the decision rule chosen applies, and the user is prompted about the best alternative found. Note that the expected payoff decision rule applies in the case that the user has provided information about the likelihood of the events.
3. Build a form that allows the user to access the reports created.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Present the table of regrets for the proposed alternatives.
2. Report the best alternative found using the maximin, maximax, Laplace, minimax regret and expected payoff methods.

Reference

Krajewski, L.J., Ritzman, L.P., "Operations Management: Strategy and Analysis." *Prentice Hall*, 6th Ed., 2002.

Problem Description

Economic lot sizing is a well-known problem in the area of production and operations management, as it has a variety of applications. For example, the economic lot-sizing problem is the core problem in aggregate production planning in MRP systems.

The economic lot-sizing problem is defined as follows: Given the demand, unit production cost, and unit inventory holding cost for a commodity, and given production capacities and set-up costs for each time period over a finite, discrete-time horizon, find a production schedule that satisfies demand at minimum cost. This model assumes a fixed and a variable component of production costs.

Let T be the length of the planning horizon, and c_t , h_t , s_t , b_t denote the unit production cost, the unit inventory holding cost, the set-up cost, and demand in period t ($t = 1, \dots, T$), respectively. The following are the decision variables for this problem:

q_t : amount produced in period t
 I_t : inventory level at the end of period t

$$y_t = \begin{cases} 1 & \text{if production occurs in period } t \\ 0 & \text{otherwise.} \end{cases}$$

A mixed-integer programming formulation of the classical economic lot-sizing problem is as follows:

$$\min : \sum_{t=1}^T (c_t q_t + s_t y_t + h_t I_t)$$

Subject to :

$$q_t + I_{t-1} = b_t + I_t \quad 1 \leq t \leq T \quad (1)$$

$$q_t \leq b_t y_t \quad 1 \leq t \leq T \quad (2)$$

$$I_1 = I_T = 0 \quad (3)$$

$$y_t \in \{0,1\} \quad 1 \leq t \leq T \quad (4)$$

$$y_t, q_t \geq 0 \quad 1 \leq t \leq T \quad (5)$$

The objective of this problem is to minimize the total production and inventory costs during the planning horizon. The first set of constraints is the flow conservation constraints. The second set of constraints shows that if production occurs in period t , the total amount produced could be as high as demand during periods t to T . Constraint (3) shows that the initial inventory level and the inventory level in the end of the planning horizon are equal to zero. Constraint (4) is the integrality constraints, and constraint (5) is the non-negativity constraints.

The aim of this project is to build a decision support system that enables the user to build a production plan by solving the economic lot-sizing problem. Below we present a dynamic programming algorithm to solve the economic lot-sizing problem. This algorithm solves the problem in $O(T^2)$. The special structure of the optimal solutions to economic lot-sizing

problem explains why the dynamic programming algorithm works. In this project we simply describe the algorithm. To learn more about the economic lot-sizing problem and dynamic programming algorithm, and to understand why this algorithm works, we refer the students to Nemhauser and Wolsky (1999) and Wagner and Whitin (1958).

Dynamic Programming Algorithm

Let $v(t)$ be the minimum cost of a solution for periods $1, \dots, t$. If $\tau \leq t$ is the last period in which production occurs, then $q_\tau \leq b_{\tau t}$ and $I_{\tau-1} = 0$. This implies that the problem can be divided into two smaller sub-problems and the least cost solution $v(\tau - 1)$ is optimal for the first sub-problem (periods $1, \dots, \tau - 1$). This leads to the following recursive function:

$$v(t) = \min_{1 \leq \tau \leq t} \left\{ v(\tau - 1) + s_\tau + c_\tau b_{\tau t} + \sum_{j=\tau}^t h_j b_{j+1, t} \right\}$$

with $v(0) = 0$.

Calculating $v(t)$ for $t = 1, \dots, T$ leads to the optimal solution $v(T)$ of the economic lot sizing problem.

Excel Spreadsheets

Build a spreadsheet that present the following data: c_t, h_t, s_t, b_t for $t = 1, \dots, T$.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. In this form insert two option buttons. These option buttons allow the user to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, a text box appears where the user can type in the length of the planning horizon T . Upon the submission of this data a table appears. The table has dimensions T by 4 and allows the user to type in c_t, h_t, s_t, b_t for $t = 1, \dots, T$.
3. Build a form that allows the user to understand the economic lot-sizing problem by looking at an example. This form includes the following:
 - a. A problem statement.
 - b. Problem formulation.
 - c. The optimal objective function value and optimal solution.
 - d. A graphical representation of the optimal solution.
4. Build a form that allows the user to solve the problem and view the results. In this form include three frames. In the first frame include two option buttons and a command

button. The option buttons allow the user to select a method to solve the problem. The two methods available are the dynamic programming algorithm and the Excel solver. When the command button is clicked –on, the problem is solved using the method selected by the user. The second frame has a number of option buttons that allow the user to open any of the reports described below. The third frame, titled “Sensitivity Analysis,” has option buttons that enable the user to choose a parameter for the sensitivity analysis. The user is interested in testing the sensitivity of the optimal solution with respect to demand, costs, etc.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the following results:
 - a. The optimal production plan.
 - b. The optimal inventory level in each period of the planning horizon.
 - c. The optimal inventory holding costs.
 - d. The optimal production costs.
 - e. The optimal set-up costs.
 - f. The optimal total costs.
2. Report the results from the sensitivity analysis.
3. Give a network representation of the optimal solution.

Reference

Nemhauser, L.G., Wolsey, A.L., “Integer and Combinatorial Optimization.” *John Wiley & Sons*, 1999.

Wagner, H.M, Whitin, T.M., “Dynamic Version of the Economic Lot Size Model.” *Management Science*, Volume 5, pg. 89-96, 1958.

41 Facility Location Problem

Problem Description

The facility location problem is a well-known problem in the areas of production and operations management and combinatorial optimization. The problem finds an optimal location of facilities considering facility construction costs, transportation costs, etc. This problem is very popular because it is faced by many companies. A large number of researchers have studied this problem and proposed solution approaches.

The purpose of this project is to build a decision support system that helps the managers decide where to locate a facility. We give a mathematical formulation of the un-capacitated facility location problem and describe a primal-dual algorithm. We do not give details about why this algorithm works. To learn more about the problem and the primal-dual algorithm, we refer the student to Francis et al. (1974) and Erlenkotter (1978).

Problem Formulation

Let I and J denote a set of facilities and customers, respectively ($i = 1, \dots, m$ and $j = 1, \dots, n$), and let f_i denote the fixed cost of locating a facility at location i . Let c_{ij} be the cost of supplying customer j from facility i . The decision variables are as follows:

x_{ij} : denotes the fraction of customer j 's demand satisfied by facility i

$$y_i = \begin{cases} 1 & \text{if we locate a facility at location } i \\ 0 & \text{otherwise.} \end{cases}$$

The following is a mixed-integer linear programming formulation of the un-capacitated facility location problem.

$$\min : \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

Subject to :

$$\sum_{i \in I} x_{ij} = 1 \quad \text{for all } j \in J, \quad (1)$$

$$y_i - x_{ij} \geq 0 \quad \text{for all } i \in I, j \in J, \quad (2)$$

$$x_{ij} \geq 0 \quad \text{for all } i \in I, j \in J, \quad (3)$$

$$y_i \in \{0, 1\} \quad \text{for all } i \in I. \quad (4)$$

The objective is to minimize the total costs. The first set of constraints shows that the demand of customer j (for $j = 1, \dots, J$) will be satisfied. The second set of constraints shows that if a facility is not located at a potential location i , there will be no shipments from that location. The third set of constraints is the non-negativity constraints, and the last set of constraints is the integrality constraints.

The following is the dual formulation of the linear programming relaxation of the un-capacitated facility location problem.

$$\max: \sum_{j \in J} v_j$$

Subject to:

$$\sum_{j \in J} w_{ij} \leq f_i \quad \text{for all } i \in I, \quad (5)$$

$$v_j - w_{ij} \leq c_{ij} \quad \text{for all } i \in I, j \in J, \quad (6)$$

$$w_{ij} \geq 0 \quad \text{for all } i \in I, j \in J. \quad (7)$$

v_j are the dual variables for constraint set (1), and w_{ij} are the dual variables for constraint set (2).

Dual Ascent Algorithm

Initialization:

- Let $J^+ \subseteq J$ denote the set of customers whose v_j 's are eligible to change, initially $J^+ = J$.
- Set $v_j = c_{1j}$ for all $j \in J$.
- Calculate: $s_i = f_i - \sum_{j \in J} \max\{0, v_j - c_{ij}\}$.
- Let $k(j) = \min\{k : v_j < c_{kj}\}$ for all $j \in J$.

Step 1:

Set $j = 1$ and $\delta = 0$.

Step 2:

If $j \notin J^+$, go to Step 6.

Step 3:

Set $\Delta_j = \min_{i \in I} \{s_i : v_j - c_{ij} \geq 0\}$

Step 4:

If $\Delta_j > c_{k(j),j} - v_j$, then set: $\Delta_j = c_{k(j),j} - v_j$; $\delta = 1$ and $k(j) = k(j) + 1$.

Step 5:

For each $i \in I$, such that $v_j \geq c_{ij}$ decrease s_i by Δ_j . Increase v_j by Δ_j .

Step 6:

If $j \neq n$, let $j = j + 1$ and return to Step 2.

Step 7:

If $\delta = 1$, return to Step 1, otherwise stop.

The dual adjustment procedure is used in the case that the dual variables found from the dual ascent algorithm do not satisfy the complementary slackness conditions.

Excel Spreadsheet

1. Build a spreadsheet that presents the fixed cost of opening a facility (f_i for $i \in I$).
2. Build a spreadsheet that presents the variable unit cost c_{ij} (for $i \in I$ and $j \in J$).

User Interface

1. Create the welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. In this form insert two option buttons. These option buttons allow the user to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, a text box appears where the user types in the total number of facilities (m) and the total number of customers (n). Upon submission of this information two tables appear. The first table with dimensions m by 1 allows the user to type in the fixed cost of opening a facility. The second table with dimensions m by n allows the user to type in the total variable cost of supplying each customer from different suppliers.
3. Build a form that allows the user to understand the facility location problem by looking at an example. This form includes the following:
 - a. A problem statement.
 - b. A problem formulation.
 - c. Present how the dual algorithm was implemented for solving this example.
 - d. The optimal objective function value and the corresponding optimal solution.
 - e. A graphical representation of the optimal allocation of customers to facilities.
4. Build a form that allows the user to solve the problem and view the results. In this form include three frames. In the first frame include two option buttons and a command button. The option buttons allow the user to select a method to solve the problem. The two methods available are the dual ascent algorithm and the Excel solver. When the command button is clicked on, the problem is solved using the method selected by the user. The second frame has a number of option buttons that allow the user to open any of the reports described below. The third frame, titled "Sensitivity Analysis," has option buttons that enable the user to choose a parameter for the sensitivity analysis. The user is interested in testing the sensitivity of the optimal solution with respect to fixed cost, variable unit costs, etc.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the following results:

- a. The optimal allocation of customers to facilities.
 - b. The optimal allocation costs, consisting of fixed and variable costs.
 - c. The fixed facility location costs.
 - d. The total variable costs.
2. Report the results from the sensitivity analysis.
 3. Give a graph representation of the optimal allocation of customers to facilities.

Reference

Erlenkotter, D., "A Dual-Based Procedure for the Uncapacitated Facility Location." *Operations Research*, Vol 26(6), pg. 992-1009, 1978.

Francis, R.L., McGinnis, F.L., Jr., White, J.A., "Facility Layout and Location: An Analytical Approach." *Prentice Hall*, 2nd Ed., 1974.

Problem Description

The knapsack problem is well known in operations research literature. This problem arises whenever there is a resource allocation problem with financial constraints. For example, given that you have a fixed budget, how do you select what things you should buy? Assume that everything has a cost and value. We seek assignments that provide the most value for a given budget.

The term *knapsack problem* invokes the image of a backpacker who is constrained by a fixed-size knapsack and so must fill it only with the most useful items. The classical knapsack problem assumes that each item must be put entirely in the knapsack or not included at all. It is this 0/1 property that makes the knapsack problem difficult. Suppose the backpacker must choose among p different items. Each item i ($i = 1, \dots, p$) has a weight w_i (in pounds) and a utility u_i to the backpacker. The objective is to maximize the utility of the backpacker's trip subject to the weight limitation that the backpacker can carry no more than W pounds. The following is an integer formulation of this problem.

$$\max : \sum_{i=1}^p u_i x_i$$

Subject to :

$$\sum_{i=1}^p w_i x_i \leq W$$

$$x_i \in \{0,1\} \quad \text{for all } i.$$

The aim of this project is to build a decision support system that enables the user to solve the knapsack problem. We describe a simple dynamic programming algorithm to solve this problem. We do not give details about why this algorithm works. To learn more about the knapsack problem and the dynamic programming algorithm, we refer the students to Ahuja *et al.* (1993) and Winston (1994).

Dynamic Programming Algorithm

Build a table that has dimensions p by W . The elements of this table $d(i,j)$ present the maximum utility of the selected items if we restrict our selection to the items 1 through i and impose a weight restriction of j . Our objective is to determine $d(p,W)$.

We can determine $d(p,W)$ by calculating $d(i,j)$ for increasing values of i ; and, for a fixed value of i , for increasing values of j . The following is the recursive relationship that we use for this purpose:

$$d(i, j) = \max\{d(i-1, j), u_i + d(i-1, j - w_i)\}.$$

When carrying out these computations, record the decision corresponding to each $d(i,j)$ (i.e., whether $x_i = 0$ or $x_i = 1$). These decisions allow us to construct the solution for any $d(i,j)$, including the desired solution for $d(p,W)$.

Excel Spreadsheets

Build a spreadsheet that presents the weight (w_i) and utility (u_i) of item i for $i = 1, \dots, p$.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. In this form insert two option buttons. These option buttons allow the user to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, a text box appears where the user types in the total number of items (p). Upon submission of this information, a table appears. The table has dimensions p by 2. The user types in this table the weight and utility of each item i for $i = 1, \dots, p$.
3. Build a form that allows the user to understand the knapsack problem by looking at an example. This form includes the following:
 - a. A problem statement.
 - b. A problem formulation.
 - c. Present how the dynamic programming algorithm was implemented for solving this example.
 - d. The optimal objective function value and the corresponding optimal solution.
4. Build a form that allows the user to solve the problem and view the results. In this form include three frames. In the first frame include two option buttons and a command button. The option buttons allow the user to select a method to solve the problem. The two methods available are the dynamic programming algorithm and the Excel solver. When the command button is clicked on, the problem is solved using the method selected by the user. The second frame has a number of option buttons that allow the user to open any of the reports described below. The third frame, titled "Sensitivity Analysis," has option buttons that enable the user to choose a parameter for the sensitivity analysis. The user is interested in testing the sensitivity of the optimal solution with respect to the size of the knapsack, the utility of items, etc.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the following results:
 - a. The optimal solution (x_i for $i = 1, \dots, p$).
 - b. The optimal objective function value (utility).
2. Report the results from the sensitivity analysis.

Reference

Ahuja, R.K., Magnanti, T.L., Orlin, J.B. "Network Flows: Theory, Algorithms and Applications." *Prentice Hall*, 1993.

Winston, L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

One can predict the return on an investment (e.g., stocks) based on the information about the market return. The responsiveness of a stock's return to changes in the market return is presented by the coefficient β . In other terms, a change in the market return of 1% will increase/decrease the return on the stock by $\beta\%$. A high (either positive or negative) value of β shows that the stock is very sensitive to the business cycle. Stocks with larger β value are riskier and therefore yield higher returns/losses than those with smaller values of β . The value of the coefficient β is used in pricing a stock. If we can estimate the value of the coefficient β accurately, then we can identify the overvalued/undervalued stocks and use this information to make profitable investments.

The aim of this project is to build a decision support system that enables the user to make an investment decision based on the information about market return and the sensitivity of stocks to market return. We propose a mathematical model to accurately estimate coefficient β .

Mathematical Model

Step 1: Error Estimation Procedure:

1. Calculate the market return and the return of the stocks of interest during a specific time period.
2. Use a *simple linear regression* analysis to predict the return on stock s (R_{si}) based on the information about the market return (R_{mi}) for period i ($i = 1, \dots, n$). The following is a linear model that presents the relationship between the market return and the return of a particular stock:

$$R_{si} = \alpha + \beta R_{mi} + \varepsilon \quad (1)$$

ε presents the random error term. Coefficients α and β are the constants representing the intercept and slope of the linear function (1). Coefficients α and β need to be estimated.

3. Let $\bar{\alpha}$ and $\bar{\beta}$ be the estimates for coefficients α and β of the linear function (1). The estimated return for the selected stock in period i ($E(R_{si})$) is equal to the following:

$$E(R_{si}) = \bar{\alpha} + \bar{\beta} R_{mi}. \quad (2)$$

The error from estimating the return of stock in period i (e_i) is equal to the difference between the actual and the estimated return of the stock.

$$e_i = R_{si} - E(R_{si}). \quad (3)$$

Step 2: Estimate coefficients $\bar{\alpha}$ and $\bar{\beta}$:

The user can choose one of the following methods to accurately estimate the coefficients $\bar{\alpha}$ and $\bar{\beta}$.

1. Sum of Squared Errors:

This approach chooses $\bar{\alpha}$ and $\bar{\beta}$ in such a way that the sum of squared errors over all observations is minimized. To learn more about linear regression and sum of squares method, we refer the students to Winston (1994).

2. Weighted Sum of Squares Error:

Often, the most recent observations influence future returns more than older observations do, as recent observations reflect current market conditions. To incorporate this idea, a greater weight is assigned to the squared errors of recent observations. For example: to the observations from the current year assign a weight equal to $(0.95)^0 = 1$; to last year's observations assign a weight equal to $(0.95)^1 = 0.95$; and to observations n years old assign a weight equal to $(0.95)^n$. This approach then chooses $\bar{\alpha}$ and $\bar{\beta}$ in such a way that the weighted sum of squared errors over all observations is minimized.

3. Least Absolute Deviations (LAD):

This approach chooses $\bar{\alpha}$ and $\bar{\beta}$ in such a way that the sum of the absolute errors for all observations is minimized.

4. MinMax Approach:

This method minimizes the absolute value of the maximum estimated error over all observations.

Excel Spreadsheets

Build a spreadsheet that presents the return of the selected stocks (m stocks) and market return during the last n periods.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form.
 - a. Insert a frame titled "Choose a Stock and a Market Index." The frame consists of two combo boxes. The first combo box presents a list of stocks. The second combo box presents a number of market indexes. The user can choose a market index and the stock whose return will be estimated.
 - b. Insert a text box where the user can type in the total number of time periods n to be considered when predicting the expected return of the selected stock.
3. Build a form that allows the user to understand the problem by looking at an example. This form includes the following:
 - a. A problem statement.

- b. The values of coefficients $\bar{\alpha}$ and $\bar{\beta}$ that are calculated using the methods described above.
 - c. The expected return for the selected stock using the simple linear regression method.
4. Build a form that allows the user to solve the problem and view the results. Insert a frame that has a number of option buttons. The option buttons enable the user to choose one of the methods we described above to estimate coefficients $\bar{\alpha}$ and $\bar{\beta}$. Insert a command button that, when clicked –on, uses the method selected by the user to estimate coefficients $\bar{\alpha}$ and $\bar{\beta}$ and presents the expected return of the selected stock. Insert a frame titled “Reports.” This frame includes a number of option buttons that enable the user to choose to open the reports described below.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the estimated values for coefficients α and β that were obtained using the methods described above.
2. Report the expected return of the selected stock found by using all the methods described above. For each method, present the following: the mean square error, the standard deviation, and the mean absolute deviation.
3. Plot the following in the same graph: the actual observations of the return of the selected stock over the last n time periods, the predicted return of the selected stock, and the value of the market index over the same time frame.

Reference

Winston, L.W., “Operations Research: Applications and Algorithms.” *Duxbury Press*, 3rd Ed., 1994.

Problem Description

A *stock option* is a contract that gives its owner the right, but not the obligation, to buy (call) or sell (put) a specified number of stocks for a fixed price –(the exercise or strike price) until a particular date (the expiration date). The buyer pays a nonrefundable fee (the premium) to the seller (the writer).

In the case that the actual price of the stock exceeds the exercise price, a call option results in a profit. This profit is equal to the difference between the actual and exercise prices. In the case that the actual price of the stock is lower than the exercise price, the owner will not use the option. The maximum owner's loss is equal to the premium paid for the option. The writer of an option, however, has to bear the risk of an unfavorable change in the price of the stock. As the stock options generate cash flows, they are traded in the market at a fixed price (different from the premium). Black and Scholes derived a formula to price stock options.

The aim of this project is to build a decision support system that would enable the user to decide about pricing a stock option. Below we present a mathematical model that can be used for this purpose.

Mathematical Model

We use the following notation:

p	is the current price of the stock
E	is the exercise price of the option,
T	is the time until the expiration date of the option
k	is the risk-free interest rate
s	is the standard deviation of the annual rate of return of the stock.

The price of the option is calculated as follows:

$$\text{Option Price} = p * N(d_1) - E * e^{(-T*k)} * N(d_2).$$

Where, $d_1 = \frac{\ln(p/E) + (k + s^2/2) * T}{s\sqrt{T}}$, $d_2 = d_1 - s\sqrt{T}$ and $N(.)$ is the cumulative standard normal distribution.

Note the following:

$\delta = N(d_1)$ is a measure of the sensitivity of the calculated option value to small changes of the share price.

$$\gamma = \frac{d^2(\text{Option Price})}{dp^2} = \frac{e^{(-d^2/2)}}{ps\sqrt{2\pi T}}$$

is a measure of the sensitivity of δ to small changes of the share price.

$$\theta = \frac{d(\text{Option Price})}{dT} = \frac{\frac{ps}{e^{(d^2/2)}}}{2\sqrt{2\pi T}} - \frac{kE}{e^{kT}} N(d - s\sqrt{T})$$
 measures the sensitivity of option value to small changes in time till maturity.

$$v = \frac{\frac{p\sqrt{T}}{e^{(d^2/2)}}}{\sqrt{2\pi}}$$
 measures the sensitivity of option value to small changes of volatility.

Excel Spreadsheets

Create a spreadsheet that presents the following information about n different stocks: the current price of the stock, the exercise price of the option, the expiration date of the option, and the standard deviation of the annual rate of return of the stock.

To get the latest information about stocks and their corresponding prices, we refer the students to the following websites: <http://finance.yahoo.com> and <http://www.nyse.com>.

User Interface

1. Build a welcome form.
2. Build a form that allows the user to understand the problem by looking at an example. This form includes the following:
 - a. A problem statement.
 - b. The mathematical formulation of the Black and Scholes option-pricing model.
 - c. The option price found using the Black and Scholes model.
3. Build a form to allow the user to select a stock and calculate its option price. The following are suggestions to help you design this form.
 - a. Insert a combo box that presents a list of stocks. This combo box enables the user to select a stock.
 - b. Insert a text box that allows the user to type in the risk-free interest rate.
 - c. Insert a frame that has four option buttons. The option buttons enable the user to choose one of the measures presented above: δ , γ , θ or v .
 - d. Insert a command button that, when clicked on, presents the option price found using the Black and Scholes model.
4. Build a form to enable the user to access the reports described below. This form includes a number of option buttons that give the user options to choose from.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the option price for the stocks presented in Spreadsheet 1. Use the Black and Scholes model to calculate the option price.
2. Report for each stock the following measures: δ , γ , θ and ν .
3. Graph the relationship between the stock price and the call premium for different values of T (time until expiration).
4. Graph the relationship between the stock price and the call premium for different values of s (standard deviation of the stock return).

Reference

<http://bradley.bradley.edu/~arr/bsm/model.html>

<http://finance.yahoo.com>

<http://www.nyse.com>

Problem Description

XYZ, Inc. has allocated a budget to provide food for children attending primary and middle schools in Chile. The company awards contracts to eligible firms for providing this service. Lately, the company has not been very successful in assigning contracts to quality bidders. The schools have been complaining about the quality of food served.

The aim of this project is to build a decision support system that will help XYZ, Inc. improve the quality of the companies chosen to serve the schools. We provide a mathematical model to solve the problem of assigning contracts to bidders.

Integer Programming Model

For modeling purposes, we divide the country into a number of geographical areas. Each geographical area is divided into territorial units (TUs). Firms bid to get contracts for a single or a combination of TUs in order to exploit economies of scale.

The following are some business rules that need to be followed in the order presented.

1. The interested firms are registered with XYZ, Inc.
2. XYZ, Inc. evaluates the firms registered from a managerial, financial, technical, and legal viewpoint. The firms that do not satisfy the minimum reliability requirements are ruled out. XYZ, Inc. assesses the capacity of the firms that do meet the minimum requirements. The firms are classified based on their financial and operating capacity as well as technical and managerial capabilities.
3. The firms that meet the minimum requirements bid for the contract confidentially. Their bids are assigned a code and stored in the system. Each bid must include the following:
 - a. The TUs being applied for.
 - b. The price of three food structures for each meal type.
 - c. Nutritional information of all foods offered.
4. The nutritional information of the meals offered is compared to the standards set by XYZ, Inc. The firms that meet the requirements are qualified. The qualified firms compete on prices and reputation. The winning firms are then chosen using the following integer programming model.

The following is the notation used in the integer programming model:

R	is the set of geographic regions in the country
I	is the set of TUs
K	is the set of participating firms
J	is the set of bids submitted
$c_j(f, d)$	is the cost of bid j for a given food f structure and demand level d
w_k	is the weight assigned to firm k ($k \in K$) according to its performance rating
$e(j)$	is the firm presenting bid j ($j \in J$)

$u(j)$	is the set of TUs in bid j
$O(k,r)$ ($r \in R$)	is the set of bids presented by firm k that include TUs belonging to region r
$M(k)$	is the maximum number of TUs acceptable for firm k . This limit depends on the size of the firm.

The decision variables are as follows:

X_j	takes the value 1 if bid j ($\forall j \in J$) is accepted and 0 otherwise
Y_{kr}	takes the value 1 if firm k will serve TUs in region r ($\forall k \in K, \forall r \in R$) and 0 otherwise
Z_k	takes the value 1 if firm k has at least one bid accepted and 0 otherwise.

$$\min : \sum_{j \in J} c_j(f,d)w_{e(j)}X_j$$

Subject to :

$$\sum_{j:i \in u(j)} X_j \geq 1 \quad \forall i \in I, \quad (1)$$

$$\sum_{j:e(j)=k} \|u(j)\| X_j \leq M(k) \quad \forall k \in K, \quad (2)$$

$$Y_{kr} - \sum_{j:j \in O(k,r)} X_j \leq 0 \quad \forall k \in K; r \in R, \quad (3)$$

$$\sum_{j:j \in O(k,r)} X_j - \|O(k,r)\| Y_{kr} \leq 0 \quad \forall k \in K, \forall r \in R, \quad (4)$$

$$Z_k - \sum_{r \in R} Y_{kr} \leq 0 \quad \forall k \in K, \quad (5)$$

$$\sum_{r \in R} Y_{kr} - \|R\| \cdot Z_k \leq 0 \quad \forall k \in K, \quad (6)$$

$$X_j, Y_{kr}, Z_k \in \{0,1\} \quad \forall j \in J; k \in K; r \in R. \quad (7)$$

The objective is to minimize the weighted cost of the accepted bids. The cost of a bid is weighted based on the performance rating of the bidder. The first set of constraints shows that to each TU will be assigned a company (the winner of a bid) to offer food services for their schools. The second set of constraints shows that the number of TUs assigned to a firm is limited. The third set of constraints shows that firm k will be assigned to serve TUs in region r ($Y_{kr}=1$) only if its bid is accepted ($X_j = 1$). The fourth set of constraints shows that in the case firm k is assigned to serve region r ($Y_{kr}=1$), the total number of bids that are accepted (that belong to firm k and serve region r) will be at most equal to the total number of bids submitted by firm k for region r . The fifth set of constraints shows that if firm k won at least one bid ($Z_k=1$), there will be at least one region r assigned to this firm. The sixth set of constraints shows that if firm k won at least one bid ($Z_k=1$), the total number of regions assigned to that firm will be bounded by the total number of regions to be served. The last set of constraints is the integrality constraints.

Excel Spreadsheets:

1. Build a spreadsheet that presents the data about the regions to be covered and their corresponding TUs.
2. Build a spreadsheet that presents the evaluation criteria related to managerial, financial, technical, and legal aspects of businesses.
3. Build a spreadsheet that presents the following data about the firms registered for the bidding process: the maximum number of TUs that can be assigned to this firm, the weight assigned (by XYZ, Inc.) on each performance criteria, etc. The maximum number of TUs assigned to a firm is based on the firm's financial and operating capacity.
4. Build a spreadsheet that presents the data about technical project requirements, such as the nutritional requirements of different meals.
5. Build a spreadsheet that presents the following data about each bid: the name of the firm bidding, the TUs the firm is bidding for, the price of different food structures offered for each meal, and the nutritional values of each food structure. Assign a code to each bid.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form.
 - a. Insert a frame titled "Regions" that has three option buttons and a command button. The option buttons enable the user to choose whether to add/delete/update the information about a region in the database. The command button allows the submission of the selection made by the user. If the user selected to add a region, then a text box appears where the user types-in the following: the name of the region, its location, and the names of the corresponding TUs. If the user selected to delete a region, a text box and a (delete) command button appear. The user types in the text box the name of the region and clicks on the command button to delete the information about the selected region from the database. If the user selects to update the information about a region, a text box and a command button appear. The user types in the text box the name of the region and clicks on the command button. In response, the corresponding spreadsheet is opened, and the cursor is located at the row that presents the data about the selected region.
 - b. Insert a frame titled "Firms." The frame includes three option buttons that enable the user to choose whether to add/delete/update the information presented in the database about the firms bidding.
 - c. Insert a frame titled "Bids." The frame includes three option buttons and a combo box. The user selects one of the bids listed in the combo box. The option buttons allow the user to select whether to add/delete/update the data presented in the database about the selected bid.
 - d. Insert a frame titled "Performance Criteria." The frame includes three option buttons that allow the user to choose whether to add/delete/update the information presented in the database about each performance criteria.
3. Build a form that allows the user to analyze the bids posted. The following are suggestions to help you design this form:

- a. Insert a command button that, when clicked on, returns the names of the firms that satisfy the minimum reliability requirements.
 - b. Insert a command button that, when clicked on, lists the names of the firms qualified and the corresponding weight assigned to each one based the performance criteria. Sort the information in descending order of the performance weight.
 - c. Insert a command button titled "See an Example." When the user clicks on this button, a frame opens that includes the following:
 - i. A problem statement.
 - ii. A formulation of the corresponding auction-bidding mathematical model.
 - iii. The optimal assignment of regions to firms.
 - d. Insert a command button that, when clicked on, uses Excel to solve the mathematical model for the auction bidding problem and opens Form 3, described below.
4. Build a form that allows the user to perform a sensitivity analysis. In this form insert a list box to present the parameters that can be used for the sensitivity analyses. For example, the user might be interested to know the sensitivity of the optimal assignment to the maximum number of TU's acceptable for a firm, etc.
 5. Build a form that allows the user to open and view the reports presented below. Include a number of option buttons to enable the user to select one of the reports.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. List the firms that meet the minimum reliability requirements.
2. List the names of the qualified firms. For each firm, present the weight assigned by XYZ, Inc. based on the performance criteria. Sort this information in descending order of the performance criteria.
3. Present the optimal assignment of firms to TUs.
4. Present the results from the sensitivity analysis.

Reference

Epstein, R., Henríquez, L., Catalán, J., Weintraub, Y.G., Martínez, C., "A Combinatorial Auction Improves School Meals in Chile." *Interfaces*, Vol. 32, Nr. 6, 2002.

Problem Description

A transportation company uses its boats to ship perishable goods among different cities located in the Pacific coast. Since the cargo is perishable, the customers are concerned about delivery times. The cargo should reach the destination not later than the required delivery date. The company wants to determine the minimum number of ships needed to meet required delivery dates. In modeling and solving this problem, one should consider the time it takes to load/unload a shipment and the time it takes to reach the next destination port.

The aim of this project is to build a decision support system that enables the user to identify the total number of boats needed to meet the required delivery times. Below we present a network flow formulation of this problem.

Network Flow Model

The tanker-scheduling problem can be formulated as a network flow problem. The corresponding network contains a node for each shipment and an arc from node i to node j if it is possible to deliver shipment j after completing shipment i ; that is, the start time of shipment j is no earlier than the delivery time of shipment i plus the corresponding unload time and travel time from the destination of shipment i to the origin of shipment j . We split each node i into two nodes i' and i'' and add the arc (i', i'') . Set a lower bound for each shipment arc (i', i'') to 1. Add a source node s in this network that is connected to the origin of each shipment i' , and add a sink node t that is connected to each destination node i'' . Set the capacity of each arc in the network equal to 1. In this network each directed path from source s to sink t corresponds to a feasible schedule for a single ship. As a result, a feasible flow of value v in this network decomposes into schedules of v boats; and our problem reduces to identifying a feasible flow of minimum value. Note that the zero flow is not feasible because shipment arcs have lower bounds equal to 1 unit.

The following notation is used:

v	presents the number of boats needed
A	presents the set of arcs in the network
A^*	presents the set of shipment arcs (i', i'') of the network
N	presents the set of the nodes in the network.

The decision variables are as follows:

x_{ij}	takes value 1 if arc (i,j) is used and 0 otherwise.
----------	---

$$\begin{aligned} & \min : v \\ & \text{Subject to :} \\ & \sum_{\{j:(i,j) \in A\}} x_{ij} - \sum_{\{j:(j,i) \in A\}} x_{ij} = \begin{cases} v & \text{for } i = s \\ 0 & \text{for } i \in N - \{s, t\} \\ -v & \text{for } i = t \end{cases} \quad (1) \\ & x_{ij} = 1 \quad \text{for all } (i, j) \in A^* \quad (2) \\ & x_{ij} \in \{0,1\} \quad \text{for all } (i, j) \in A. \quad (3) \end{aligned}$$

Augmenting Path Algorithm

The following is an algorithm that can be used to solve the tanker-scheduling problem. To learn more about network flow problems and the augmenting path algorithm, we refer the students to Ahuja et al. (1993).

Let G denote a network with a set N of nodes and a set A of arcs. Network $G(x)$ is called a residual network, and it consists of the arcs with positive residuals. An augmenting path P is a directed path from the source s to sink t in the residual network $G(x)$. We define the residual capacity of an augmented path (δ) as the minimum residual capacity of this path ($\min \{r_{ij} : (i,j) \in P\}$).

```

begin
  x = 0
  while G(x) contains a directed path from node s to node t do
    begin
      identify an augmenting path P from node s to node t
       $\delta = \min \{r_{ij} : (i,j) \in P\}$ 
      augment  $\delta$  units of flow along P and update G(x)
    end
  end

```

Excel Spreadsheets

1. Build a spreadsheet that presents for each shipment the origin, the destination, and the delivery date.
2. Build a spreadsheet that presents the loading/unloading time for shipment.
3. Build a spreadsheet that presents the travel time between ports.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. In this form include two option buttons. The user has the choice to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box appears where the user types in the name of the file.

- b. If the user chose to enter the data manually, three text boxes appear where the user types in the total number of shipments (n), the total number of origins (n_o) and the total number destination ports (n_d). Upon submission of this information, three tables appear. The first table, with dimensions 4 by n , allows the user to type in the shipment number, shipment origin and destination, and delivery date. The second table, with dimensions n_o by n_d , allows the user to type in the loading/unloading time for shipments that move from a port of origin to a destination port. The third table, with dimensions n_o by n_d , allows the user to type in the travel time between origin-destination ports.
3. Build a form that presents the following details about a related example. The user can understand how the algorithm works by using this form.
 - a. A problem statement.
 - b. A network flow formulation of this problem.
 - c. Present how the augmenting path algorithm was implemented for solving this example.
 - d. A graphical representation of the optimal solution.
 - e. The optimal number of boats needed. The optimal assignment of shipment(s) to boats.
4. Build a form that allows the user to solve the problem and perform a sensitivity analysis. Give the user the option to select either the Excel solver or the augmenting path algorithm to solve the problem. Give the user a number of options (problem parameters to choose from) for the sensitivity analyses.
5. Build a form to allow the user to access the reports described below. For this purpose use option buttons.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Present the optimal number of boats required. Present the shipment(s) assigned to each boat.
2. Give a graphical representation of the optimal solution.

Reference

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., "Network Flows: Theory, Algorithms and Applications." Prentice Hall, 1993.

Problem Description

The traveling salesman problem is a well-known problem in the area of network and combinatorial optimization. This problem is easy to state: “Starting from his home base, node 1, a salesman wishes to visit each of several cities represented by nodes $2, \dots, n$, exactly once and return home, doing so at the lowest travel cost” (see Ahuja et al. 1993).

The simplicity of this problem and its complexity to solve have attracted the attention of many researchers over a long period of time. The first mathematical model related to the traveling salesman problem was studied in the 1800s. Researchers have paid attention to this problem because it is a generic core model that captures the combinatorial essence of most routing problems.

The purpose of this project is to create a decision support system that enables the user to solve the traveling salesman problem. We give a mixed-integer programming formulation of this problem and describe two heuristics that can be used to find feasible solutions. To learn more about the traveling salesman problem, we refer the students to Ahuja et al. (1993) and Winston (1994).

Mixed Integer Formulation

The following notation is used:

A	the set of arcs of the network
n	the total number of nodes
N	the node-arc incidence matrix
V	the set of nodes of the network
c_{ij}	the cost of using arc (i,j) ($(i,j) \in A$)
b	the demand vector.

The decision variables are:

x_{ij}	the flow on arc (i,j) ,
y_{ij}	takes the value 1 if the salesman travels from city i to city j .

The objective is to minimize the total travel cost. The first and second sets of constraints show that each city is visited exactly once. The salesman will enter and leave a city exactly once. The third set of constraints makes sure that the network is connected and therefore each city will be visited. The fourth set of constraints is the sub-tour elimination constraints. The fifth set of constraints is the non-negativity constraints, and the sixth set of constraints is the integrality constraints.

$$\min : \sum_{(i,j) \in A} c_{ij} y_{ij}$$

Subject to:

$$\sum_{1 \leq j \leq n} y_{ij} = 1 \quad i = 1, \dots, n, \quad (1)$$

$$\sum_{1 \leq i \leq n} y_{ij} = 1 \quad j = 1, \dots, n, \quad (2)$$

$$Nx = b, \quad (3)$$

$$x_{ij} \leq (n-1)y_{ij} \quad (i, j) \in A, \quad (4)$$

$$x_{ij} \geq 0 \quad (i, j) \in A, \quad (5)$$

$$y_{ij} \in \{0,1\} \quad (i, j) \in A. \quad (6)$$

Heuristic Approaches

1. **Nearest Neighbor:** Start with an arbitrary node i_1 and construct a path $i_1, i_2, \dots, i_j, i_{j+1}, \dots, i_n$, where $i_{j+1} = \arg(\min\{c_{i_j k} : k \in V \setminus \{i_1, i_2, \dots, i_j\}\})$, with ties broken arbitrary. Complete the path to a tour by adding arc (i_1, i_n) .
2. **Nearest Insertion:** Given a tour T and a node $i \in V \setminus T$, let $d(i, T) = \min_{j \in T} c_{ij}$ and $i^* = \arg(\min\{d(i, T) : i \in V \setminus T\})$. Suppose $j^* = \arg(\min\{c_{i^* j} : j \in T\})$. Thus i^* is the closest node to T , and j^* is the node in T that is closest to i^* . Now construct a subtour $T \cup \{i^*\}$ by inserting i^* between j^* and one of its neighbors in T ; that is, if (j_1, j^*) and (j^*, j_2) are arcs of T and $c_{i^* j_1} \leq c_{i^* j_2}$, insert i^* between j_1 and j^* . This process terminates with a tour, but again we cannot guarantee that it will be the best tour.

To learn more about the nearest neighbor and nearest intersection heuristics as well as other heuristic approaches for the traveling salesman problem, we refer the students to Nemhauser and Wolsey (1999).

Excel Spreadsheets

1. Build a spreadsheet that presents the node-arc incidence matrix.
2. Build a spreadsheet that presents the cost of each arc of the network.

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. In this form include two option buttons. The option buttons enable the user to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:

- a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, two text boxes appear where the user types in the total number of nodes (n) and the total number of arcs (m) of the network. Upon submission of this information, two tables appear. The first table, with dimensions n by m , allows the user to enter the node-arc incidence matrix. The second table, with dimensions 1 by m , allows the user to type in the cost of traveling in a particular arc.
3. Build a form that presents the following details about a related example. The user can understand how the algorithm works by using this form.
 - a. The problem statement.
 - b. A graphical representation of the corresponding network.
 - c. The network flow formulation.
 - d. Present how the heuristic approaches were implemented to solve this example.
 - e. The optimal objective function value and the optimal tour.
 - f. A graphical representation of the optimal solution.
 4. Build a form that allows the user to solve the problem and perform a sensitivity analysis. Give the user the option to select either the Excel solver or the heuristic approaches described above to solve the problem. Give the user a number of options (problem parameters to choose from) for the sensitivity analyses.
 5. Build a form to allow the user to access the reports described below. For this purpose use option buttons.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Give a graphical representation of the corresponding network.
2. Give a graphical representation of the optimal solution (tour).
3. Give a graphical representation of the solutions (tours) found using the two heuristic approaches (nearest neighbor and nearest intersection).
4. Present the total cost of the tour found from the following:
 - a. Solving the problem optimally using the Excel solver.
 - b. Solving the problem using the nearest neighbor algorithm.
 - c. Solving the problem using the nearest intersection algorithm.
5. Present the results from the sensitivity analysis.

Reference

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., "Network Flows: Theory, Algorithms and Applications." *Prentice Hall*, 1993.

Nemhauser, G.L., Wolsey, L.A., "Integer and Combinatorial Optimization." *John Wiley & Sons*, 1999.

Winston, L.W., "Operations Research: Applications and Algorithms." *Duxbury Press*, 3rd Ed., 1994.

Problem Description

The vehicle routing problem is well known in the area of combinatorial optimization. The main reason for such popularity is this problem has a number of applications. For example, the problem of scheduling the delivery of products to a customer using a fleet of vehicles can be modeled as a vehicle routing problem. This problem is a generalization of the traveling salesman problem.

The vehicle routing problem states the following: “**Given:** (a) a fleet of K capacitated vehicles domiciled at a common depot, say node 1; (b) a set of customer sites $j = 1, \dots, n$ each with a prescribed demand d_j ; and (c) the cost c_{ij} of traveling from location i to j ; **Find** the set of routes for delivering (picking up) the goods to customer sites at the minimum cost. We assume that the fleet of vehicles is homogeneous and that each vehicle has a capacity of u units.” (Ahuja *et al.* (1993)).

The aim of this project is to build a decision support system that will enable companies to identify a schedule for delivering products to customers by solving the vehicle routing problem. Below we present an integer programming formulation of this problem. To learn more about the vehicle routing problem, we refer the students to Ahuja *et al.* (1993).

Integer Programming Formulation

The following notation is used:

A	the set of arcs of the network
K	the total number of vehicles
u	the capacity of a vehicle
n	the total number of nodes
c_{ij}	the cost of using arc (i,j) ($(i,j) \in A$)
d_i	the demand at node i .

The decision variables are as follows:

x_{ij}^k	takes the value 1 if vehicle k travels from site i to site j
y_{ij}	takes the value 1 if some vehicle travels from site i to site j .

The objective is to minimize the total cost of delivering the products to customers. The first set of constraints shows that there is at most one vehicle traveling from site i to site j (for all $(i,j) \in A$). The second and third sets of constraints show that each customer site is visited exactly once. A vehicle will enter and leave a city exactly once. The fourth and fifth set of constraints show that there will be exactly K vehicles leaving the depot and exactly K vehicles entering the depot. The sixth set of constraints shows that vehicle capacity should not be violated. The seventh set of constraints is the sub-tour elimination constraints. The eighth set of constraints is the integrality constraints.

$$\min : \sum_{k=1}^K \sum_{(i,j) \in A} c_{ij} x_{ij}^k$$

Subject to :

$$\sum_{k=1}^K x_{ij}^k = y_{ij} \quad (i, j) \in A, \quad (1)$$

$$\sum_{j=1}^n y_{ij} = 1 \quad i = 1, \dots, n, \quad (2)$$

$$\sum_{i=1}^n y_{ij} = 1 \quad j = 1, \dots, n, \quad (3)$$

$$\sum_{j=1}^n y_{1j} = K \quad (4)$$

$$\sum_{i=1}^n y_{i1} = K \quad (5)$$

$$\sum_{i=2}^n \sum_{j=1}^n d_i x_{ij}^k \leq u \quad k = 1, \dots, K, \quad (6)$$

$$\sum_{i \in Q} \sum_{j \in Q} y_{ij} \leq |Q| - 1 \quad \forall Q \subseteq \{2, 3, \dots, n\} \quad (7)$$

$$y_{ij}, x_{ij}^k \in \{0, 1\} \quad (i, j) \in A, k = 1, \dots, K. \quad (8)$$

Heuristic Solution Approaches

Nearest Neighbor:

Let V^* denote the set of customer sites that have not been visited yet by a vehicle.

1. Start at the depot (node 1) and construct a path consisting of nodes $1, i_1, \dots, i_j, 1$ where $i_j = \arg(\min \{c_{i_{j-1}k} : k \in V^*\})$; $u \geq d_{i_1} + d_{i_2} + \dots + d_{i_j}$, and any other node $s \in V^*$ is such that $u < d_{i_1} + d_{i_2} + \dots + d_{i_j} + d_s$.

2. Repeat Step 1 until all the sites are visited.

The total number of paths constructed presents the total number of vehicles needed to deliver the products to customers.

To learn more about the nearest neighbor heuristic as well as other heuristic solution approaches for the traveling salesman problem, we refer the students to Chopra and Meindl (2001).

Excel Spreadsheets

1. Build a spreadsheet that presents the demand at each customer site.
2. Build a spreadsheet that presents the cost of traveling from customer site i to site j .

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. In this form include two option buttons. The option buttons enable the user to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, a text box appears where the user can type in the total number of customer sites (n). Upon submission of this information, two tables appear. The first table, with dimensions 1 by n , allows the user to enter the data about demand in each customer site. The second table, with dimensions n by n , enables the user to type in the cost of traveling from one site to the others.
3. Build a form that presents the following details about a related example. The user can understand how the nearest neighbor algorithm works by using this form.
 - a. The problem statement.
 - b. A graphical representation of the corresponding network.
 - c. Integer programming formulation.
 - d. Present how the nearest neighbor algorithm was implemented to solve this example.
 - e. The optimal number of vehicles needed to deliver the products to customers and the corresponding transportation costs.
 - f. A graphical representation of the optimal solution (vehicle tours).
4. Build a form that allows the user to solve the problem and perform a sensitivity analysis. Give the user the option to select either the Excel solver or the nearest neighbor algorithm to solve the problem. Give the user a number of options (problem parameters to choose from) for the sensitivity analyses.
5. Build a form to allow the user to access the reports described below. For this purpose use option buttons.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Give a graphical representation of the optimal solution (vehicle tours).
2. Give a graphical representation of the solution (vehicle tours) found using the nearest neighbor approach.
3. Present the total transportation cost and the total number of vehicles used when the following occurs:
 - a. The problem is solved optimally using Excel.

- b. The problem is solved using the nearest neighbor approach.
4. Present the results from the sensitivity analysis.

Reference

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., "Network Flows: Theory, Algorithms and Applications." *Prentice Hall*, 1993.

Chopra, S., Meindl, P., "Supply Chain Management: Strategy, Planning, and Operation." *Prentice Hall*, 2001.

Problem Description

Bonds are financial instruments issued by agencies of the federal government, by state and local governments, and by corporations. A bond is an obligation by the bond issuer to pay money to the bond holder according to rules specified at the time the bond is issued. Usually, a bond pays a specific amount, called face value, at the date of maturity. In addition, the bonds make periodic coupon payments. The coupon amount is a percentage of the face value amount, and the period between coupon payments is usually less than a year.

The aim of this project is to build a decision support system that helps users manage their investment in bonds. We provide some insights about managerial issues related to bonds, such as pricing a bond, the relationship between bond prices and yield to maturity, bond duration, etc. To learn more about this topic, we refer the students to Luenberger (1998).

Bonds

The internal rate of return of a bond at the current price is called *yield to maturity*. The price of a bond with face value F that makes m coupon payments of C/m each year, n periods remaining, and yield to maturity value of λ , is calculated as follows:

$$P = \frac{F}{[1 + (\lambda/m)]^n} + \sum_{k=1}^n \frac{C/m}{[1 + (\lambda/m)]^k} \quad (1)$$

Formula (1) helps to understand the relationship between bond price, yield, coupon, and time to maturity. For example, there is a negative relationship between the price and the yield of a bond. The *price-yield* curve presents the relationship between the price and the yield of a bond.

Another measure of yield is the *current yield*. Current yield measures of the annual return of a bond. Current yield is defined as follows:

$$CY = \frac{\text{annual coupon payment}}{\text{bond price}} \times 100. \quad (2)$$

The *duration* of a bond gives a direct measure of the sensitivity of the bond to interest rates. For a bond that makes payments m times per year, with the payment in period k equal to c_k , and n periods remaining, the duration is defined as follows:

$$D = \frac{\sum_{k=1}^n (k/m)c_k / [1 + (\lambda/m)]^k}{PV}, \quad (3)$$

where λ is the yield to maturity and

$$PV = \sum_{k=1}^n \frac{c_k}{[1 + (\lambda/m)]^k} \quad (4)$$

The change in the price of a bond due to a small change in yield (or vice versa) can be estimated using the following formula:

$$\Delta P \approx -\frac{1}{1 + (\lambda/m)} DP \Delta \lambda \quad (5)$$

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. Insert four text boxes. The first text box allows the user to type in the face value of the bond, F . The second text box allows the user to enter the total number of coupon payments made per year, m . The third text box enables the user to type in the coupon payments made within a year, C . The fourth text box enables the user to type in the maturity date of the bond. The maturity date is used to calculate the number of time periods left until maturity n .
3. Build a data analysis form. In this form include a number of option buttons and a command button. The option buttons allow the user to choose whether to calculate the price of the bond P , the yield to maturity λ , the current yield CY , or the bond duration D . If the user chose to calculate the price of the bond, P , a text box appears where the user types in the value of λ . If the user chose to calculate the yield to maturity, a text box appears where the user types in the price of the bond. When the user clicks on the command button, the chosen index is calculated and the corresponding result is presented.
4. Build a form to enable the user to perform a sensitivity analysis with respect to problem parameters. In this form insert two option buttons, a text box, and a command button. The option buttons enable the user to choose whether to identify the sensitivity of a bond's price with respect to the bond's yield or vice versa. The user types in the text box the amount of change. When the user clicks on the command button, the sensitivity analysis is performed and the corresponding result is presented.
5. Build a form that presents the following details about a related example.
 - a. The problem statement.
 - b. The formulas used to calculate a bond's duration, yield to maturity, current yield, and price. Report the corresponding results.
 - c. Graphs 1, 2, and 3, described in the Reports section below.
6. Build a form to allow the user to access the reports described below. For this purpose use option buttons.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the price, yield to maturity, duration, and current yield of the bond.
2. Graph the price-yield curves for bonds that have a maturity of 30 years and coupon rates equal to 15%, 10%, 5%, and 0%. Express prices as a percentage of par. Interpret the results.
3. Graph the price-yield curves for bonds that have a coupon rate of 10% and maturity equal to 30, 10, and 3 years. Express prices as a percentage of par. Interpret the results.
4. Graph the relationship between the duration of a bond yielding 5% and the bond maturity for bonds with coupon rates equal to 1%, 2%, 5%, and 10%. Consider years to maturity equal to 1, 2, 5, 10, 25, 50, 100, and infinity. Interpret the results.

Reference

Luenberger, D.G., "Investment Science." *Oxford University Press*, 1998.

Problem Description

Production lines consist of an ordered sequence of n production stages. The products start being processed at the 1st stage of the production line in batches of size $B \geq 1$. The products are then processed in all stages of the production line until they are finally ready to be delivered to customers. An issue that managers face in managing production lines is identifying where to place inspection stations. Inspection stations could ideally be placed next to each production stage. However, this is not a common practice, as it is expensive. The managers are interested in identifying an optimal inspection plan that specifies at which stages to inspect a product so that the total of production and inspection costs is minimized.

The aim of this project is to build a decision support system that will help the user to identify where to locate inspection stations in a production line. Below we present a mathematical formulation for this problem.

Mathematical Formulation

We formulate this problem as a shortest path problem on a network with $n + 1$ nodes. The network contains an arc (i, j) for each node pair i and j for which $i < j$; therefore, the network

has $\frac{n(n+1)}{2}$ arcs. We assume that the inspection identifies every defective product and

the defects are not repairable. The defective products are scrapped. The probability of producing a defective product at stage i is α_i . The production line must end with an inspection station so that defective products are not delivered to customers.

The following notation is used:

A	the set of arcs of the network
N	the set of nodes of the network
p_i	the unit production cost in stage i
f_{ij}	the fixed cost of inspecting a batch after stage j , given that the batch was last inspected in stage i
g_{ij}	the variable unit cost of inspecting in stage j , given that the batch was last inspected in stage i .

Note that the inspection costs at stage j depend on the time when the batch was inspected last because the inspector looks for defects incurred at the intermediate stages.

Let $B(i) = B \prod_{k=1}^i (1 - \alpha_k)$ denote the expected number of non-defective products at the end of stage i . We associate cost c_{ij} with any arc (i, j) of the network:

$$c_{ij} = f_{ij} + B(i)g_{ij} + B(i) \sum_{k=i+1}^j p_k.$$

The integer programming formulation is as follows:

$$\min : \sum_{(i,j) \in A} c_{ij} x_{ij}$$

Subject to:

$$\sum_{\{j(i,j) \in A\}} x_{ij} - \sum_{\{i(i,j) \in A\}} x_{ij} = \begin{cases} n-1 & \text{for } i = s \\ -1 & \text{for all } i \in N - \{s\} \end{cases} \quad (1)$$

$$x_{ij} \in \{0,1\} \quad \text{for all } (i,j) \in A \quad (2)$$

The objective is to minimize the total production and inspection costs. The first set of constraints is the flow conservation constraints. Our decision variables take values 0 or 1. x_{ij} takes the value 1 if j is the next stage to be inspected after stage i has been inspected; and x_{ij} takes the value 0 otherwise.

The shortest path problem can be solved using Dijkstra's algorithm. To learn more about the shortest path problem and Dijkstra's algorithm, we refer the students to Ahuja *et al.* (1993).

Excel Spreadsheets

1. Build a spreadsheet that presents the fixed and variable production costs in all stages of the production line.
2. Build a spreadsheet that presents the probability of defects in all stages of the production line.
3. Build a spreadsheet that presents the cost of inspecting a batch after stage j , given that the batch was last inspected in stage i .

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. In this form include two option buttons. The option buttons enable the user to select whether to read the data from a file or manually enter the data. Include a command button that, when clicked –on, performs these actions:
 - a. If the user chose to read the data from a file, a text box should appear where the user types in the name of the file.
 - b. If the user chose to enter the data manually, a text box appears where the user types in the total number of production stages (n). Upon the submission of this information, three tables appear. The first table, with dimensions 1 by n , allows the user to type in production costs. The second table, with dimensions 1 by n , allows the user to type in the probability of defects in each production stage. The third table, with dimensions $2*n$ by $n(n+1)/2$, enables the user to type in the inspection costs for each arc (i,j) .
3. Build a form that presents the following details about a related example. The user can understand how the nearest neighbor algorithm works by using this form.
 - a. The problem statement.
 - b. The integer programming formulation.
 - c. The solution found using Dijkstra's algorithm.

- d. A graphical representation of the solution.
 - e. The optimal location of the inspection stations.
 - f. Present how Dijkstra's algorithm was implemented to solve this example.
4. Build a form that allows the user to solve the problem and perform a sensitivity analysis. Give the user the option to select either the Excel solver or Dijkstra's algorithm to solve the problem. Insert a command button that, when clicked on, submits the user's selection and solves the problem. Give the user a number of options (problem parameters to choose from) for the sensitivity analyses.
 5. Build a form to allow the user to access the reports described below. Use option buttons.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the expected number of non-defective products at the end of stage i ($B(i)$) for all $i \in N$.
2. Report the total (production and inspection) costs on arc (i,j) (c_{ij}) for all $(i,j) \in A$.
3. Report the optimal location of the inspection stations and the optimal production and inspection costs.
4. Give a graphical representation of the optimal solution.
5. Present the results from the sensitivity analysis.

Reference

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., "Network Flows: Theory, Algorithms and Applications." *Prentice Hall*, 1993.

Problem Description

“Green River” hospital has a call center that helps to schedule doctors’ appointments for patients (callers), schedule lab work, register new patients, etc. The call center operates from 7am to 9pm during the weekdays. As more patients are using the call center, a number of issues have arisen. Often, patients call to schedule an appointment about a health condition without knowing exactly to which department it should be directed. As a result, they spend a significant amount of time on the phone being transferred from one department to the other until they contact the most appropriate department, deferring other callers. The hospital has received complaints from patients about the long waiting times to schedule an appointment. The managers are concerned about the inefficiency of operations at the call center. They think that the call center is understaffed; however, they are not sure about how many employees to hire and how to schedule their working hours. A team of employees was assigned to observe the operations at the call center. The team observed the number of phone calls received during the operating hours. They considered seven time periods, each two hours long. Table 1 presents the expected number of calls per time period.

Hour	Expected No. Of Calls
7 A.M. - 9 A.M.	55 calls
9 A.M. - 11 A.M.	90 calls
11 A.M. - 1 P.M.	80 calls
1 P.M. - 3 P.M.	93 calls
3 P.M. - 5 P.M.	85 calls
5 P.M. - 7 P.M.	45 calls
7 P.M. - 9 P.M.	20 calls

The team found out that an operator is able to process on average six calls per hour. The call center has part-time and full-time employees. The full-time employees work eight hours a day, and they can either begin working at 7A.M., 9A.M., 11A.M., or 1P.M. Full-time employees get paid \$14 per hour for work before 5P.M. and \$16 per hour after 5P.M. The part-time employees work only four hours a day, and they can begin work either at 1P.M., 3P.M., or 5P.M. Part-time employees get paid \$10 per hour for work before 5P.M. and \$12 per hour after 5P.M.

The aim of this project is to build a decision support system to manage the staff of the call center. We provide a mathematical model to solve this problem.

Integer Programming Model

The decision variables for this problem are as follows:

- x_1 The total number of full-time employees who work from 7A.M. to 3P.M. Their salary is \$112 per day.
- x_2 The total number of full-time employees who work from 9A.M. to 5P.M. Their salary is \$112 per day.
- x_3 The total number of full-time employees who work from 11A.M. to 7P.M. Their salary is \$116 per day.

- x_4 The total number of full-time employees who work from 1P.M. to 9P.M. Their salary is \$120 per day.
- x_5 The total number of part-time employees who work from 1P.M. to 5P.M. Their salary is \$40 per day.
- x_6 The total number of part-time employees who work from 3P.M. to 7P.M. Their salary is \$44 per day.
- x_7 The total number of part-time employees who work from 5P.M. to 9P.M. Their salary is \$48 per day.

$$\min : 112x_1 + 112x_2 + 116x_3 + 120x_4 + 40x_5 + 42x_6 + 44x_7 + 46x_8 + 48x_9$$

Subject to :

$$x_1 \geq 55/6$$

$$x_1 + x_2 \geq 90/6$$

$$x_1 + x_2 + x_3 \geq 80/6$$

$$x_1 + x_2 + x_3 + x_4 + x_5 \geq 93/6$$

$$x_2 + x_3 + x_4 + x_5 + x_6 \geq 85/6$$

$$x_3 + x_4 + x_6 + x_7 \geq 45/6$$

$$x_4 + x_7 \geq 20/6$$

$$x_1, \dots, x_7 \geq 0$$

$$x_1, \dots, x_7 \text{ int}$$

The objective is to minimize the total cost of operating the call center.

User Interface

1. Build a welcome form.
2. Build a data analysis form. The following are suggestions to help you design this form.
 - a. Insert a command button that, when clicked on, solves the problem using Excel.
 - b. Insert a frame titled "Reports" that includes a number of option buttons. The option buttons allow the user to select any of the reports described below. Insert a command button that, when clicked –on, opens the report selected by the user.
 - c. Insert a frame titled "Sensitivity Analysis." This frame allows the user to select a parameter to perform the sensitivity analysis. The user may be interested in observing the sensitivity of the optimal solution with respect to the following: changes in the hourly rate payment of full-time and part-time employees; distribution of phone calls received throughout the day; etc. Include in this frame text boxes, option buttons, check boxes, combo boxes, and command buttons as needed.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the optimal number of full-time and part-time employees to work in each shift. Report the optimal cost of operating the call center.
2. Present the results from the sensitivity analysis.

Reference

Hillier, F.S., Lieberman, G.J., "Introduction to Operations Research." 7th Ed., *McGraw Hill*, 2001.

Problem Description

Eastern Electric (EE) is a major appliance manufacturer with a large plant in the Chicago area. EE purchases the motors for the appliances from Westview Motors, located in Dallas. Demand at EE has been relatively constant in the last several years. This year's estimate of demand for appliances is 150,000 units; therefore, EE is planning to purchase 150,000 motors at \$110/motor. Each motor weights about 10 pounds, and EE has traditionally purchased in lots of 3,000 motors. Westview ships each EE order within one day of receiving it (order processing time is one day). EE carries a safety inventory equal to 50% of the average demand for motors during the delivery time.

Every year the plant manager at EE revises the decision about the mode of transportation to use for shipping the motors from Dallas. The following updates announced in the beginning of the year make the revision necessary: (i) transportation companies update their price per quantity shipped; (ii) EE announces the forecasts of annual demand and price of appliances; (iii) production managers at EE suggest new lot sizes; (iv) inventory unit holding costs are updated. EE collects a number of proposals from transportation companies. The proposals are analyzed, and based on the results of this analysis, the managers decide about the proposal that will be accepted.

The following table presents details about the proposals submitted last year. Note that one *cwt* is equal to 100 pounds.

Carrier	Range of Quantity Shipped (in cwt)	Shipping Cost (\$/cwt)
AM Railroad	200 +	6.50
Northeast Trucking	100 +	7.50
Golden Freightways	50 – 150	8.00
Golden Freightways	150 – 250	6.00
Golden Freightways	250 – 400	4.00

EE has been undergoing a similar process during the last 10 years. Data has been collected and stored in an Excel spreadsheet. The managers are interested in analyzing the data collected so far to identify trends in transportation and inventory costs. The managers want to know whether the lot size of 3,000 units is optimal, and if not, to identify the optimal order quantity. EE's annual cost of holding inventories is estimated to be 25% of the value of the product.

The aim of this project is to build a decision support system that enables the user to do the following: analyze the data to identify the optimal order quantity; choose a carrier; and calculate total (inventory plus transportation) costs. The selection of a carrier depends mainly on the following cost components: cycle inventory, safety inventory, in-transit inventory and transportation costs.

The Model

Cycle inventory = $0.5 \times \text{Lot size}$

Lead time = (Transit time) + (Order processing time at Westview)

Safety stock = $0.5 * (\text{Demand during lead time})$

In-transit inventory = Demand during transit time

Total inventory = Cycle inventory + Safety inventory + In-transit inventory

Inventory holding costs = $(\text{Total Inventory}) * (\text{Unit inventory holding costs})$

Excel Spreadsheets

1. Build a spreadsheet that presents historical data about the annual demand for appliances.
2. Build a spreadsheet that presents historical data about transportation and inventory costs.
3. Build a spreadsheet that presents the names of carriers and their proposed rates.

User interface

4. Build a welcome form.
5. Build a data analysis form. The following are suggestions to help you design this form.
 - a. Insert a frame titled "Add/Delete/Update Proposals." The frame includes three option buttons and a command button. The option buttons allow the user to choose whether to add, delete, or update the data about proposals.
 - i. If the user selects to add a new proposal, text boxes appear where the user types in the name of the carrier, the range of the quantity shipped, the unit shipping cost, and the length of the transit time. The user clicks on a command button to add the information in Spreadsheet 3.
 - ii. If the user selects to delete a proposal, a text box appears where the user types in the identification number of the proposal to be deleted. The user clicks on a command button to delete the information about the selected proposal.
 - iii. If the user chooses to update a proposal, a text box appears where the user submits the identification number of the proposal. When the user clicks on a command button, the row that corresponds to the selected proposal in Spreadsheet 3 appears. The user updates the information.
 - b. Insert a text box that allows the user to type in the quantity ordered by EE. The default value for this text box is 3,000 units.
 - c. Insert a text box that allows the user to type in the inventory holding cost at EE. The default value for this text box is 25%.
 - d. Insert a text box that allows the user to type in the unit purchasing cost for the motors. The default value for this text box is \$110/motor.
 - e. Insert a text box that allows the user to type in the annual demand for appliances. The default value for this text box is 150,000 units.
 - f. Insert a command button that, when clicked on, considers the data entered by the user and calculates the following: cost of transportation, cycle inventory, safety stock, in-transit inventory, total inventory cost, and total costs.

- g. Insert a command button titled "Proposal Evaluation Last Year." When the user clicks on this button, a frame opens that includes the following:
 - i. A list of proposals submitted last year together with data about range of quantity shipped and unit shipping costs.
 - ii. For each proposal, present the following: cost of transportation, cycle inventory, safety stock, in-transit inventory, total inventory cost, and total costs.
 - iii. Present the proposal that was selected to be the best.
- h. Insert a frame titled "Reports." This frame includes a number of option buttons that allow the user to open the reports described below.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. For each proposal, report the cost of transportation, cycle inventory, safety stock, in-transit inventory, total inventory cost, and total costs.
2. Lot size impacts inventory and transportation costs. Let Q be the lot size provided by the user. Calculate the total inventory costs and transportation cost for the following lot sizes: $(1/5)*Q$, $(2/5)*Q$, $(3/5)*Q$, $(4/5)*Q$, Q , $(6/5)*Q$, $(7/5)*Q$, $(8/5)*Q$, $(9/5)*Q$, $2*Q$. This information will help the managers to understand whether the current lot has an optimal size.
3. Use the data from Report 2 to graph the following relations:
 - a. Lot size versus transportation costs.
 - b. Lot size versus inventory costs.
 - c. Lot size versus total costs.
4. Use the data presented in Spreadsheets 1 and 2 to build the following graphs:
 - a. Annual demand during the last 10 years.
 - b. Total transportation costs during the last 10 years.
 - c. Total inventory costs during the last 10 years.
 - d. Unit transportation and inventory costs during the last 10 years.

Reference

Chopra, S., Meindl, P., "Supply Chain Management: Strategy, Planning and Operation," *Prentice Hall*, 2001.

Shapiro, R.D., Heskett, J.L., "Logistics Strategy: Cases and Concepts," St. Paul, Minnesota: *West publishing Co.*, 1985.

Problem Description

Simulating a system that inherits random components requires a method of generating random numbers. The aim of this project is to introduce the students to the concept of random numbers, demonstrate how to build a random number generator, illustrate how they work, show some of the problems they exhibit, and, finally, provide some statistical tools to test their performance.

A random number generator is an algorithm that is used to generate random variates that are independent and uniformly distributed in the interval $[0, 1]$ (we denote it by IID and $U(0,1)$). A few algorithms have been developed for generating random numbers. Although the problem sounds easy, practice has shown that the generators of random numbers do not work properly. In most of the cases the numbers generated are not independent, or the cycle length is short (the same stream of random numbers is generated).

We describe two different algorithms. The first, called Linear Congruential Algorithm (LCG), was introduced in 1951. This is a simple algorithm and is easy to understand. However, its quality heavily depends on the choice of the parameters a , c , and m . The second algorithm is the Combined Multiple Recursive Generator (CMRG). This is the algorithm used by the simulation software ARENA. Excel also provides a generator of random numbers. Excel has a function called *rand()* that generates random numbers. To learn more about random numbers and random number generators, we refer the students to Law and Kelton (2000).

The Chi-square test is an empirical test to check the quality (independence) of the random numbers generated by using a particular generator.

Algorithm 1: Combined Multiple Recursive Generator

```

For i = 3 to n
   $X_i = (1403580X_{i-2} - 810728X_{i-3})(\text{mod } 4294967087)$ 
   $Y_i = (527612Y_{i-1} - 1370589X_{i-3})(\text{mod } 4294944443)$ 
   $Z_i = (X_i - Y_i)(\text{mod } 4294967087)$ 
  If ( $Z_i > 0$ )
     $U_i = Z_i/4294967088$ 
  If ( $Z_i = 0$ )
     $U_i = 4294967087/4294967088$ 
   $i = i + 1$ 
End i

```

Algorithm 2: Linear Congruential Generator

```

For i = 1 to n
   $Z_i = (aZ_{i-1} + c)(\text{mod } c)$ 
   $U_i = Z_i/m$ 
   $i = i + 1$ 
End i

```

Empirical Test: Chi-square test

1. H_0 : U_i are IID $U(0,1)$
2. Divide $[0,1]$ into k sub-intervals of equal length
3. f_j is the number of U_i 's in the j^{th} interval
4. Generate U_1, U_2, \dots, U_n
5. Calculate

$$\chi^2 = \frac{k}{n} \sum_{j=1}^k \left(f_j - \frac{n}{k}\right)^2$$

$$\chi_{k-1, 1-\alpha}^2 \approx (k-1) \left\{ 1 - \frac{2}{9(k-1)} + z_{1-\alpha} \sqrt{\frac{2}{9(k-1)}} \right\}$$

6. If $(\chi^2 > \chi_{k-1, 1-\alpha}^2)$ we reject the Null Hypothesis (H_0)

User Interface

1. Build a welcome form.
2. Build a data analysis form. The following are suggestions to help you design this form.
 - a. Insert a frame that includes three option buttons. The option buttons allow the user to choose whether the stream of random numbers will be generated using the Linear Congruential Generator, the Combined Multiple Recursive Generator, or the Excel *rand()* function.
 - i. If the user chose to use the Linear Congruential Generator, four text boxes appear where the user can type the value of the coefficients c , m , a , and Z_0 .
 - ii. If the user chose to use the Combined Multiple Recursive Generator, six text boxes appear where the user can type in the value of the coefficients X_0 , X_1 , X_2 , Y_0 , Y_1 , and Y_2 .
 - b. Insert a text box where the user can type in the total number of random numbers to be generated (n).
 - c. Insert a check box that enables the user to choose whether to perform the Chi-square test to check the performance of the generator.
 - d. Insert a command button that, when clicked on, generates the random numbers using the algorithm selected by the user.
3. Build a form to allow the user to open the reports described below. Use option buttons.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report streams of 30 random numbers generated using the Linear Congruential Generator, the Combined Multiple Recursive Generator, and Excel.
2. Report the result from the Chi-square test for each stream.
3. Plot the stream of random numbers generated.
4. Present a stream of 30 random numbers generated using the Linear Congruential Generator for the following choice of parameters: $a = 5$, $c = 3$, and $m = 16$. Interpret the results.
5. Present a stream of 30 random numbers generated using the Linear Congruential Generator for the following choice of parameters: $a = 5$, $c = 3$, and $m = 2^{31} - 1$. Compare these results with the results found in Report 4.

Reference

Law, A.M., Kelton, W.D., "Simulation Modeling and Analysis." 3rd Ed., *McGraw-Hill*, 2000.

Problem Description

Simulating a system that has any random aspects at all must involve sampling, or generating random variates from a probability distribution. Generating a random variate means obtaining an observation on a random variable from a desired distribution. The aim of this project is to demonstrate how to build a random number generator and provide statistical tools to test its performance. We will use the Chi-square test and plots to check the performance of the generators.

We provide algorithms to generate random numbers from the following distributions: uniform (discrete and continuous); exponential; normal; Weibull; m-Erlang; and triangular. To learn more about how to generate random variates, we refer the students to Law and Kelton (2000). Note that the very first step for all the generators we present is generating a random number IID and $U(0,1)$. The students are suggested to use the Excel function *rand()* to generate random numbers.

Continuous Uniform: $U(a,b)$

```

For i = 1 to n
  Generate  $U_i \sim U(0,1)$ 
   $X_i = a + (b - a) U_i$ 
   $i = i + 1$ 
End i

```

Discrete Uniform: $U(a,b)$

```

Calculate  $p = 1/(b-a+1)$ 
For i = 1 to n
  Generate  $U_i \sim U(0,1)$ 
  For j = 1 to  $b-a+1$ 
    If  $U_i \leq j*p$ 
       $X_i = a-1+j$ 
       $j = b-a+2$ 
    Else
       $j = j + 1$ 
  End j
   $i = i + 1$ 
End i

```

Exponential: $\text{Exp}(b)$

```

For i = 1 to n
  Generate  $U_i \sim U(0,1)$ 
   $X_i = -b*\ln(U_i)$ 
   $i = i + 1$ 
End i

```

m-Erlang: m-Erlang(b)**For i = 1 to n**Generate $U_{i1}, U_{i2}, \dots, U_{im}$ as IID and $U(0,1)$

$$X_i = -(b/m) * \ln\left(\prod_{j=1}^m U_{ij}\right)$$

 $i = i + 1$ **End i****Standard Normal:** $N(0,1)$ **For i = 1 to n**S1: Generate U_{i1} and U_{i2} as IID and $U(0,1)$

$$V_1 = 2 * U_{i1} - 1$$

$$V_2 = 2 * U_{i2} - 1$$

$$W = V_1^2 + V_2^2$$

If $W > 1$

Go to S1

Else

$$Y = \sqrt{(-2 * \ln(W)) / W}$$

$$X_i = V_1 * Y$$

$$X_{i+1} = V_2 * Y$$

$$i = i + 2$$

End i**Weibull:** Weibull(a,b)**For i = 1 to n**Generate $U_i \sim U(0,1)$

$$X_i = b * (-\ln(U_i))^{1/a}$$

$$i = i + 1$$

End i**Normal:** $N(\mu, \sigma^2)$ **For i = 1 to n**Generate $U_i \sim N(0,1)$

$$X_i = \mu + U_i * \sigma$$

$$i = i + 1$$

End i**Triangular:** triang(a,b,c)**For i = 1 to n**Generate $U_i \sim U(0,1)$

$$Z = (c-a)/(b-a)$$

If $(U_i \leq Z)$

$$X_i = \sqrt{Z * U_i}$$

Else

$$X_i = 1 - \sqrt{(1-Z) * (1-U_i)}$$

$$X_i = a + (b-a) * X_i$$

$$i = i + 1$$

End i

Empirical Test: Chi-square test

1. H_0 : U_i are IID $U(0,1)$
2. Divide $[0,1]$ into k sub-intervals of equal length
3. f_j is the number of U_i 's in the j^{th} interval
4. Generate U_1, U_2, \dots, U_n
5. Calculate

$$\chi^2 = \frac{k}{n} \sum_{j=1}^k \left(f_j - \frac{n}{k}\right)^2$$

$$\chi_{k-1,1-\alpha}^2 \approx (k-1) \left\{ 1 - \frac{2}{9(k-1)} + z_{1-\alpha} \sqrt{\frac{2}{9(k-1)}} \right\}$$

6. If $(\chi^2 > \chi_{k-1,1-\alpha}^2)$ we reject the Null Hypothesis (H_0)

User Interface

1. Build a welcome form.
2. Build a data analysis form. The following are suggestions to help you design this form.
 - a. Insert a frame that includes seven option buttons. The option buttons enable the user to choose whether to generate a stream of numbers from uniform (discrete or continuous), exponential, normal, Weibull, m-Erlang, or triangular distribution. Depending on the choice of the distribution, a number of text boxes appear where the user types in the parameters of the distribution. For example, if the user wants to generate a stream of random numbers from the uniform continuous distribution in the interval $[2,5]$, the user has to specify that the parameters for this distribution are $a = 2$ and $b = 5$.
 - b. Insert a text box where the user types in the total number of random numbers to be generated (n).
 - c. Insert a check box that enables the user to choose whether to perform the Chi-square test to check the performance of the generator.
 - d. Insert a command button that, when clicked on, generates the random variates from the distribution selected by the user.
3. Build a form to allow the user to open the reports described below. Use option buttons.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report streams of 30 random variates generated using the algorithms described above.
2. Report the result from the Chi-square test for each stream.
3. Plot the stream of random variates generated.

Reference

Law, A.M., Kelton, W.D., "Simulation Modeling and Analysis." 3rd Ed., *McGraw-Hill*, 2000.

Problem Description

The aim of this project is to build a simple board game called “Connect Four.” The game is played with two players or a player and a computer. The student should design a simple algorithm that generates computer moves when the game is played between a player and the computer. Below we describe how to play the game.

The game takes place in a 7 by 6 rectangular board. Each player has 21 turns/men to play. Players take turns dropping a man from the top of the board in one of the seven columns. The man falls down and fills the lowest unoccupied square. A player cannot drop a man in a certain column if it is already full (i.e., it contains six men). The players decide who will begin the game. The object of this game is to connect four men vertically, horizontally, or diagonally. If the board is filled and no one has aligned four men, then the game is over. The total number of moves is 42.

User Interface

1. Build the welcome form.
2. Build a form that includes the following controls:
 - a. Build a combo box that allows the player to choose whether to play on a 7 by 6 or 10 by 9 board. Upon selection, the play board appears on the Excel spreadsheet. The slots/cells of the play board should be squared. A snapshot of the board is given below.

- b. Insert a combo box that allows the player to pick a color for the board.
- c. Insert a combo box that allows the player to choose whether the game will be played between the player and the computer or between two players. Upon submission of this information, text boxes appear for the player(s) to type in the name(s).
- d. Insert a combo box that enables the players to pick their own color. For example, in the play board presented above, one of the players has selected blue and the other yellow.
- e. Insert a combo box that allows the players to choose who will start the game.
- f. The green cells in the top of each column of the play board displayed above present command buttons. When the player clicks on one of the buttons, the empty slot of the corresponding column should be filled with the respective color of the player. The player should get an error message if the player attempts to fill in a slot

in a column that is completely filled. When the game is played between the computer and a player, the first time the computer plays, randomly select any of the columns and fill the bottom square with the selected color.

- g. Insert text boxes that present the following: the total number of moves for each player, the name of the player who should play next, and the total number of moves until the end of the game.
- h. Display a message box that presents the name of the winner and the total number of moves. If the board is filled and no one has aligned four men, then the game is drawn. This will happen after 42 moves when the game is played on a 7 by 6 board.
- i. Insert a command button that enables the player to choose whether to replay the game or to close the program.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

Report the following information about the games played so far: the names of the players, the name of the winner, and the total number of moves.

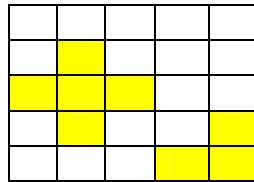
Problem Description

The aim of this project is to build a simple board game called “Lights Puzzle.” Developing this game will enhance students’ ability to program using Visual Basic for Excel.

The game is played with a single player. The game takes place in a 5 by 5 or 10 by 10 board. The player clicks on one of the slots/cells of the play board. The slot/cell that the player clicks on is lighted. The objective of this game is to turn on all the lights on the board. It is somewhat difficult to do so because turning on a specific light results in all adjacent lights to change their state. The adjacent lights that were on will turn off, and the ones that were off will turn on.

User Interface

1. Build the welcome form.
2. Build a form that includes the following controls:
 - a. Build a combo box that allows the player to choose whether to play on a 5 by 5 or 10 by 10 board. Upon selection, the play board appears on the Excel spreadsheet. The slots/cells of the play board should be squared. A snapshot of the board is given below. During the game, when the player clicks on a cell, the following happens: the selected cell changes color, the adjacent cells change color, and the cells are checked to see if they are all lighted.



- b. Insert a combo box that allows the player to pick a color for the board.
- c. Insert a combo box that allows the player to pick a color for the lighted cells. For example, in the play board presented above, the player has chosen yellow.
- d. Insert text boxes that present the total number of moves made so far and the name of the player.
- e. Display a message box that presents the name of the winner and the total number of moves when the game is over.
- f. Insert a command button that enables the player to choose whether to replay the game or close the program.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Problem Description

The aim of this project is to build a simple board game called “Tic Tac Toe.” Developing this game will enhance students’ ability to program using Visual Basic for Excel. The game is played with two players or a player and a computer. The student should implement the algorithms described below to generate computer moves when the game is played between a player and the computer. Below we describe how to play the game.

The game is usually played in a 3 by 3 board. The players take turns in playing the game. In each move the player marks one of the slots/cells. The goal is to get three marks in one row, horizontally, vertically or diagonally. The first player to get three marks in one row wins the game. The total number of moves for this game is 9.

User Interface

1. Build the welcome form.
2. Build a form that includes the following items:
 - a. Build a 3 by 3 play board in an Excel spreadsheet. The slots/cells of the play board should be squared. A snapshot of the board is given below.

X	0	
	0	
X		

- b. Insert a combo box to allow the player to pick a color for the board.
- c. Insert a combo box that allows each player to pick a representing mark. For example, in the play board presented above, one of the players has chosen the cross and the other the circle.
- d. Insert a combo box that allows the player to choose whether the game will be played between the player and the computer or between two players. Upon submission of this information, text boxes appear for the player(s) to type in the name(s). In the case that the player chose to play with the computer, a combo box appears that allows the player to choose the difficulty level of the game: beginner or advanced. For each level, we describe below algorithms that can be used by the computer to play the game.
- e. Insert a combo box that allows the players to choose who will start the game.
- f. Insert text boxes that present the following: the total number of moves for each player, the name of the player who should play next, and the total number of moves until the end of the game.

- g. Display a message box that presents the name of the winner and the total number of moves. If the board is filled and no one has aligned three marks in a row, then the game is drawn.
- h. Insert a command button that enables the player to choose whether to replay the game or close the program.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

Report the following information about the games played so far: the names of the players, the name of the winner, and the total number of moves.

Algorithms

There are a number of algorithms available to decide about the moves of the computer (which you would program) in response to the moves of the player during this game. If implemented properly, it would be very difficult for the computer to lose. Below we present two heuristic approaches to automate the computer's moves, one for the beginner level and the other for the advanced level.

Beginner Level Algorithm:

1. Check if there is a move that the computer can make so that there will be 3 marks in one row, horizontally, vertically or diagonally.
 - a. If so, fill in the relevant square.
 - b. Else, check for a move that will block a win for the other player and fill in the relevant square.
 - c. Else, fill in the square that lies on the row/column with the maximum number of cells free of marks.

Advanced Level Algorithm:

The "Tic-Tac-Toe" board game can easily be represented as a decision tree (see Figure 1). The tree for this game is very small; therefore, it is easy to make moves (add marks on the board) until you hit a board that ends the game, unlike games such as chess where the corresponding "game tree" is very big.

A board with no marks is used as the root node of the tree. Whenever it is the computer's turn, check each available move and assign the resulting board a score. The score indicates whether the board is in favor of the computer (a positive score), in favor of the player (a negative score), or neither (0). Each possible move is then added as a child to the root. The computer makes the move that takes the board from its current position to the selected branch of the tree.

The following is the minimax algorithm:

1. List all possible moves available to the computer.
2. List all the possible moves the player can make given that the computer made a move.

3. Find the maximum board value for each case, assuming that is the move the player will make.
4. Make the move that will give the smallest maximum for the player.

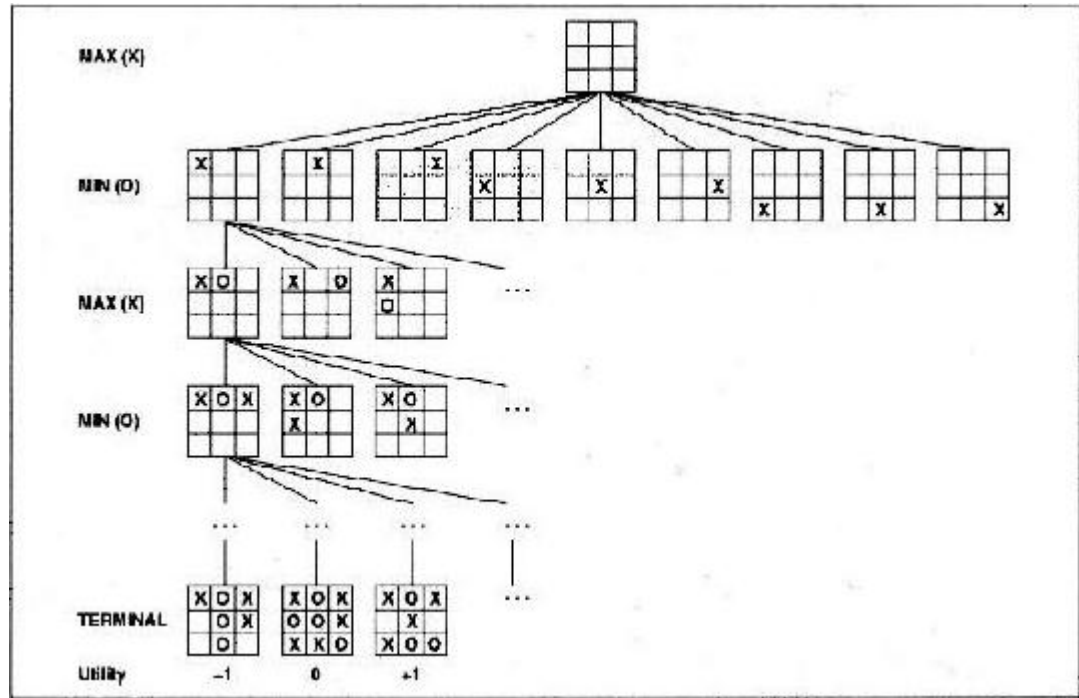


Figure 1. Decision Tree for the "Tic Tac Toe" Board Game.

Problem Description

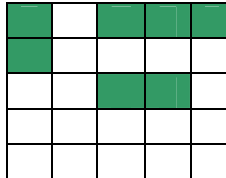
The aim of this project is to build a simple board game called “TacTix.” Developing this game will enhance students’ ability to program using Visual Basic for Excel. The game is played with two players or a player and a computer. The student should develop an algorithm, based on the game logic described below, that generates computer moves when the game is played between a player and the computer.

The game can be played in square boards of size 5 by 5 through 15 by 15. The players take turns in playing the game. In each move, the player removes slots/cells from either a single row or a single column. One can remove as few as one or as many as three slots/cells on each turn. The slots/cells must be on adjacent squares (e.g., you could not remove only two corner slots/cells). The goal is to force the other player to remove the last slot/cell.

Note that it has been proven that the player who moves second can always win if the player plays perfectly. So, if the player makes the first move, the computer should always win. Try it!

User Interface

1. Build the welcome form.
2. Build a form that includes the following controls:
 - a. Insert a combo box that allows the players to select the size of the play board. Upon the submission of this information, the play board appears on an Excel spreadsheet. The slots/cells of the play board should be squared. A snapshot of a 5 by 5 play board is given below.



- b. Insert a combo box that allows the player to pick a color for the board.
- c. Insert a combo box that allows the player to pick a color for the slots/cells that are removed. For example, in the play board presented above, the cells that have been removed are presented by the green color.
- d. Insert a combo box that allows the player to choose whether the game will be played between the player and the computer or between two players. Upon submission of this information, text boxes appear for the player(s) to type in the name(s). In the case that the player chose to play with the computer, a combo box appears that allows the player to choose the difficulty level of the game: beginner or advanced. For each level, we describe below algorithms that can be used by the computer to play the game.
- e. Insert a combo box that allows the players to choose who will start the game.

- f. Insert text boxes that present the total number of moves for each player and the name of the player who should play next.
- g. Display an error message if one of the following happens: the player/computer selected more than 3 slots/cells to remove; the selected slots/cells are not in the same row or not in the same column or are not adjacent; the same slot/cell is selected more than once; the player/computer skipped the turn.
- h. Display a message box that presents the name of the winner and the total number of moves when the game is over.
- i. Insert a command button that allows the player to select whether to replay the game or close the program.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

Report the following information about the games played so far: the names of the players, the name of the winner, and the total number of moves.

Game Logic

We briefly describe algorithms that can be used to model the movements of the computer when playing the “TacTix” game.

Beginner Level

In each turn, allow the computer to randomly pick up to three non-empty adjacent slots/cells from the play board.

Advanced Level

For the advanced level case, think of the game in the following way.

- a. If it is your turn to play and there is a single cell in the play board, the computer wins.
- b. If it is the computer’s turn to play and there are two empty cells in the play board, the computer will pick one and will win.
- c. If it is the computer’s turn to play and there are three empty cells in the play board, the computer will pick two and will win.
- d. If it is the computer’s turn to play and there are four empty cells in the play board, the computer will pick three and will win.

Therefore, if the computer can get the player to leave two to four cells at the end of the player’s turn, it can easily win.

- e. Let us assume there are five cells in the play board and it is the player’s turn. The player can remove one to three cells, leaving two to four cells in the play board for the computer to play. Based on the results from parts b., c., and d., we know that in this case the computer can win. Therefore, if the computer can leave five cells when it becomes the player’s turn, it wins.

Using the same reasoning, if the computer can leave nine cells when it becomes the player's turn, it wins. The student can use the same reasoning to figure out the other critical point/s.

The strategy described above starts with the last winning position for the computer and moves backwards to identify a good starting position. This is a well-known problem-solving heuristic called "The Backward-Forward Method."

Problem Description

The aim of this project is to build a simple board game called “Diamonds.” Developing this game will enhance students’ ability to program using Visual Basic for Excel. The game is played with two players or a player and a computer. The student should develop an algorithm, based on the game logic described below, to generate computer moves when the game is played between a player and the computer.

The game is usually played in a 10 by 6 playing board. To play the game, the players need 12 tokens (six tokens per player, and the tokens are numbered 1 through 6), 48 diamonds, and two dice. The first die has six sides, each one numbered (1 through 6). The other (called the Piece die) has its six sides labeled as Pawn, King, Queen, Horse, Bishop, and Castle.

The game starts with the following set-up of the board: each player places tokens in the first row of the player’s side in a random way, and the 48 diamonds are set (one on each square) on the board. The players take turns in playing the game. On the player’s turn, the player throws the two dice. The numbered die shows which of the (numbered) tokens will be moving; the Piece die shows what type of movement should be made (below we present all the possible movements). Then, the player lands one of the tokens on a square that could be empty or occupied by a diamond or the other player’s token. If the square is already occupied by a token/diamond, the player collects the token/diamond from the square and sets there the player’s own token. The first player to collect 25 diamonds wins the game. If both players collect 24 diamonds, the game is a tie. The maximum number of tokens that a player can capture from the opponent is five.

During the course of the game, if a player rolls the dice and the number that rolls up is a number depicting a token that has been collected by the opponent, then the player does not get to move on that turn. It is then the opponent’s turn to roll the dice for a move.

Token Movements

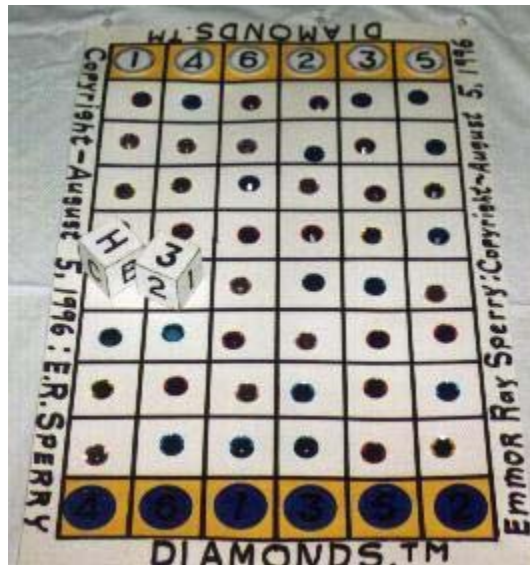
The following are the types of movements a player can make. The movements depend on the letter shown when the Piece die is thrown.

- a. Pawn: the player moves one square at a time forward, backward, or diagonally.
- b. King: the player moves one square at a time horizontally, vertically, or diagonally.
- c. Queen: the player moves forward, backward, horizontally, vertically, or diagonally up to the length or width of the playing board for as long as it is a straight run without any diamonds or tokens between the start and the finish of the move.
- d. Castle: the player moves forward, backward, horizontally, or vertically up to the length or width of the playing board for as long as it is a straight run without any diamonds or tokens between the start and the finish of the move.
- e. Bishop: the player moves forward, backward, or diagonally up to the length or width of the playing board for as long as it is a straight run without any diamonds or tokens between the start and the finish of the move.
- f. Horse: the player moves in any "L" shape on the playing board. A horse can either move one square up/down and two squares to the right/left; or two squares up/down and one square to the right/left; or one square to the right/left and two squares up/down;

or two squares to the right/left and one square up/down. A horse may jump over other tokens and diamonds during a move in order to land on an empty square or to capture a token/diamond.

User Interface

1. Build the welcome form.
2. Build a form that includes the following items:
 - a. Build a 10 by 6 play board in an Excel spreadsheet. The slots/cells of the play board should be squared. A snapshot of the board is given below.



- b. Insert a combo box that allows the player(s) to pick a color for the board.
- c. Insert a combo box that allows the player(s) to pick a color and a shape for the diamonds. For example, in the play board presented above the diamonds are presented by small black circles.
- d. Insert a combo box that allows the player(s) to pick a color and a shape for the tokens. For example, in the play board presented above, the tokens are presented with large black/white circles.
- e. Insert a combo box that allows the player to choose whether the game will be played between the player and the computer or between two players. Upon submission of this information, text boxes appear for the player(s) to type in the name(s).
- f. Insert a combo box that allows the players to choose who will start the game.
- g. Insert a command button named "Play."
 - i. If it is the player's turn, when the player clicks on the "Play" button, the Visual Basic code randomly generates an integer from 1 to 6 and randomly picks one of the following six letters: P, K, Q, C, B, or H. The player then makes a move (uses the mouse to move a token) based on the number and

letter received from tossing the dice. An error message should appear if the player makes an illegal move.

- ii. If it is the computer's turn to play, an integer from 1 to 6 is randomly generated and one of the following six letters, P, K, Q, C, B, or H, is randomly picked. A move (of the computer's token) is made based on the number and letter received from tossing the dice. When programming the moves of the computer, give priority to moves that collect tokens/diamonds.
- h. Insert text boxes that present the following: the total number of moves for each player, the total number of diamonds and tokens collected, and the name of the player who should play next.
- i. Display a message box that presents the name of the winner, the total number of moves, and the total number of diamonds and tokens collected when the game is over. Note that the game is over if we have a winner (one of the players collected 25 diamonds) or if we have a tie (each player collected 24 diamonds).
- j. Insert a command button that allows the player to choose whether to replay the game or close the program.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

Report the following information about the games played so far: the names of the players, the name of the winner, the total number of tokens/diamonds collected by each player, and the total number of moves.

Problem Description

The aim of this project is to build a board game called “Dots.” Developing this game will enhance students’ ability to program using Visual Basic for Excel. The game is played with two players or a player and a computer. The student should implement the algorithms described below to generate computer moves when the game is played between a player and the computer. Below we describe how to play the game.

The playing board for this game consists of an equal number of vertical and horizontal arrays of dots. Players take turns in playing the game. On the player’s turn, the player joins two horizontally or vertically adjacent dots with a line. When the player creates a box by closing its fourth side, the player gets one point, the corresponding box is filled in with the player’s color, and the player gets another turn to play. If the same move completes two boxes, the player gets two points and both boxes are filled in with the player’s color, but the player takes only one extra turn. The game ends when all boxes have been colored. The player who closed the maximum number of boxes wins.

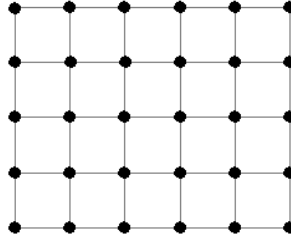
The following is a heuristic approach that the students can use to model the movements of the computer when this game is played between a player and a computer. Note that mathematical approaches have been used to model the movements of the computer when this game is played between a computer and a player. The branch of algebra and number theory called “Combinatorial Game Theory” provides relevant intelligence to program the “Dots” board game. If the students are interested in programming this game perfectly so that the computer almost always wins, we refer them to Berlekamp (2000).

Heuristic Approach

In order to program the computer’s moves while playing this game, follow these rules: Among all the possible legal moves of the computer, select the one that will allow the computer to complete the fourth side of a box (if possible) and/or will not allow the opponent to complete the fourth side of a box. If such a move is not possible, randomly choose to connect two dots. After every move, the program should check whether a box was created. If so, the box is colored and the player has another turn.

User Interface

1. Build the welcome form.
2. Build a form that includes the following controls:
 - a. Insert a text box that allows the player to type in the size of the play board. The size of the board could be anywhere from 5 by 5 to 25 by 25. The default size of the play board is 10 by 10. Upon the submission of this information, the play board appears on the Excel spreadsheet. A snapshot of a 6 by 6 (dots) play board is given below.



- b. Insert a combo box that allows the player to select a color for the board.
- c. Insert a combo box that allows the player to pick a color. This color will be used to fill in the box when the player successfully closes a box.
- d. Insert a combo box that allows the player to choose whether the game will be played between the player and the computer or between two players. Upon the submission of this information, text boxes appear for the player(s) to type in the name(s). In the case that the player chose to play with the computer, a combo box appears that allows the player to select the difficulty level of the game: beginner or advanced. For the beginner level, use the heuristic described above to model the computer's moves. For the advanced level, model the computer's moves using the approach described by Berlekamp (2000).
- e. Insert a combo box that allows the players to choose who will start the game.
- f. Insert text boxes that present the total number of scores for each player and the name of the player who should play next.
- g. Display a message box that presents the name of the winner and the total number of points when the game is over.
- h. Insert a command button that allows the player to choose whether to replay the game or close the program.

Design a logo for this game. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

Report the following information about the games played so far: the names of the players, the name of the winner, and the total number of points for each player.

Reference

Berlekamp, E., "The Game of Dots and Boxes – sophisticated child's play." *A. K. Peters Ltd.*, 2000.

<http://www.well.com/user/argv/java/dots.html>

61 *Board Game: Collector*

Problem Description

The aim of this project is to build a simple board game called “Collector.” Developing this game will enhance students’ ability to program using Visual Basic for Excel. The game is similar to chess, and it is played with two players or a player and a computer. The student should develop an algorithm, based on the game logic described below, to generate computer moves when the game is played between a player and the computer. Below we describe how to play the game.

The playing board for this game consists of 64 squares (eight squares wide and eight squares long). In the beginning of the game, each player has the following pieces: one king, one queen, two horses, two castles, and two bishops. Players select a color for their own pieces. There are 48 pawns in the game that, at the start of the game, do not belong to either player.

Rules of the Game

The following are the rules to play this game:

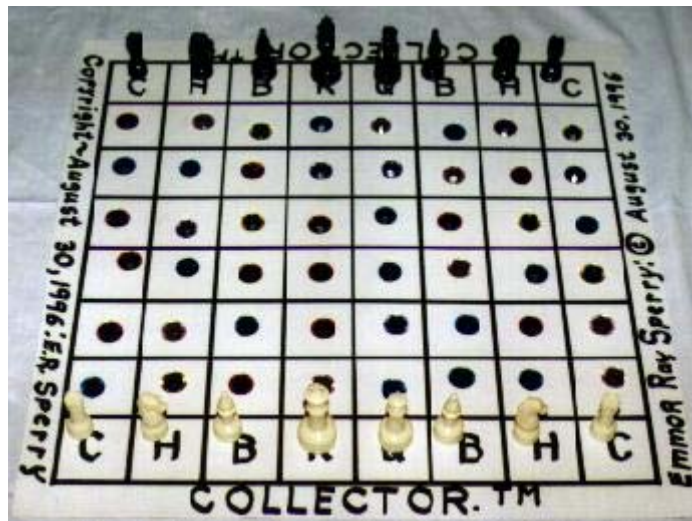
1. The game starts with the following set-up of the board: each player places the pieces in the first row of the playing board from each player’s own side, and the pawns are set one on each of the remaining 48 squares of the board. The pieces are placed in the following order from left to right: castle, horse, bishop, king, queen, bishop, horse, and castle. The setting is such that the kings of the two players face each other.
2. The players take turns in playing the game. On the player’s turn, the player moves one of the pieces. When a player lands the pieces on a square that is occupied by a pawn, the pawn is removed from the playing board and collected by the player. The player’s piece remains on the square. When a player lands one of the pieces on a square that is occupied by the opponent’s piece, the piece is removed from the playing board. The first player to collect 25 pawns wins the game. If both players collect 24 pawns, the game is a tie.
3. The maximum number of pieces that a player can collect from the opponent is seven. The king cannot be collected.
4. The king moves one square at a time horizontally, vertically, or diagonally.
5. The queen moves forward, backwards, horizontally, vertically, or diagonally up to the length or width of the playing board for as long as it is a straight run without any pawns or pieces between the start and the finish of the move.
6. The castle moves forward, backwards, horizontally, or vertically up to the length or width of the playing board for as long as it is a straight run without any pawns or pieces between the start and the finish of the move.
7. The bishop moves forward, backwards, or diagonally up to the length or width of the playing board for as long as it is a straight run without any pawns or pieces between the start and the finish of the move.
8. The horse moves in any “L” shape on the playing board. A horse can move one square up/down and two squares to the right/left; two squares up/down and one square to the right/left; one square to the right/left and two squares up/down; or two squares to the

right/left and one square up/down. A horse may jump over other pieces and pawns during a move in order to land on an empty square or to collect a piece/pawn.

Note the following: If the game is played between two players, the Visual Basic program that the student will build for this game should be such that it checks for illegal moves of players. If the game is played between a player and the computer, the Visual Basic program should not only check for illegal moves, but should also generate moves for the computer considering the rules stated above. In generating computer moves, prioritize moves that collect pieces/pawns, and avoid being collected by the opponent.

User Interface

1. Build the welcome form.
2. Build a form that includes the following items:
 - a. Build an 8 by 8 play board in an Excel spreadsheet. The slots/cells of the play board should be squared. A snapshot of the board is given below.



- b. Insert a combo box that allows the player to pick a color for the board.
- c. Insert a combo box that allows the player to pick a color for the player's own pieces. For example, in the play board presented above, one of the players has chosen white and the other black.
- d. Insert a combo box that allows the player(s) to choose symbols for the player's own pieces (such as king, queen, etc.).
- e. Insert a combo box that allows the player to choose whether the game will be played between the player and the computer or between two players. Upon submission of this information, text boxes appear for the player(s) to type in the name(s).
- f. Insert a combo box that enables the players to choose who will start the game.
- g. Insert text boxes that present the total number of pieces/pawns collected by each player and the name of the player who should play next.
- h. Display a message box in the following cases:

- i. If the game is over. The message box presents the name of the winner and the total number of pieces/pawns collected.
 - ii. If the game is a tie.
 - iii. If a player made an illegal move.
- i. Insert a command button that allows the player to choose whether to replay the game or close the program.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

Report the following information about the games played so far: the names of the players, the name of the winner, and the total number of pieces/pawns collected by each player.

Problem Description

The aim of this project is to build a board game called “TAG.” Developing this game will enhance students’ ability to program using Visual Basic for Excel. The game can be played with two to nine players. However, we will consider only the case when the game is played with two players or a player and a computer. The student should develop an algorithm, based on the game logic described below, to generate computer moves when the game is played between a player and the computer. Below we describe the rules of playing the game.

The playing board for this game consists of 81 squares (nine squares wide and nine squares long). The squares in the four corners of the playing board are colorless. The two squares adjacent to the two sides of each corner are labeled “Start for Chaser” (there are a total of eight such squares on the board). The center square of the playing board is labeled “The King.” The center square on each of the outside rows of the playing board is labeled “Home Base.”

To start the game, the players need two six-faced dice, three tokens, and a number of fence lines. The sides of the dice are numbered 1 through 6. At the start of the game, one of the tokens is set at the square labeled “The King,” and, therefore, we refer to this token as the king. The other two tokens are set at the squares labeled “Start for Chaser,” and, therefore, we refer to these tokens as the chasers. The player randomly picks one of the squares labeled “Start for Chaser” to initially land the player’s chaser. Players decide about the length and the number of the fence lines to be set in the play board. At the beginning of the game, each player locates the player’s own fence lines in the board. The fence lines are used as barrier for the tokens. The king or the chaser tokens cannot cross a fence line but, rather, must go around it.

The players take turns in playing the game. The player who starts the game is called the king player and the opponent is called the chaser player. The king player throws the two dice. Based on the total the player gets, the player moves the king token that many squares toward a home base. The objective of the king player is to touch all the four home bases without being touched/caught by the chaser (the opponent). In each move, the king player seeks the shortest path to the home base not touched yet. On the chaser player’s turn, the chaser player moves the token (that initially is set on one of the squares labeled “Start for Chaser”), trying to touch the king token. If the chaser reaches/touches the king, the chaser becomes the king player. The new king player places the king token at the king’s square (in the center of the board) and does not move the chaser token from the current position. The new chaser throws the dice and moves the chaser token (initially set at one of the “Start for Chaser” squares) in such a way to stop the king from reaching the home bases.

While programming the game, consider the following: the king token cannot jump over chaser tokens but, rather, must go around them; a chaser token cannot jump over another chaser token; a chaser token cannot land on a home base square; a player cannot count a square twice; the chaser token must move over as many squares as the number shown on the rolled dice; the king token moves over as many squares as the number shown on the rolled dice.

User Interface

1. Build a welcome form.
2. Build a form that includes the following items:
 - a. Create a 9 by 9 play board in an Excel spreadsheet. The slots/cells of the play board should be squared. Label the cells of the board as described above.
 - b. Insert a combo box that allows the players to select the total number of fence lines.
 - c. Insert a combo box that allows the player to pick a color for the board.
 - d. Insert a combo box that allows the player to pick a color for the token.
 - e. Insert a combo box that allows the player to choose whether the game will be played between the player and the computer or between two players. Upon submission of this information, text boxes appear for the player(s) to type in the name(s).
 - f. Insert a combo box that allows the players to choose who will start the game.
 - g. Insert text boxes that present the name of the player who should play next.
 - h. Display a message box in the following cases:
 - i. If the game is over. The message box presents the name of the winner.
 - ii. If a player made an illegal move.
 - i. Insert a command button that allows the player to choose whether to replay the game or close the program.

Design a logo for this game. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

Report the following information about the games played so far: the names of the players, the name of the winner, and the total number of moves made.

Problem Description

The Crystal Lake Printing Company prints a particular popular Christmas card once a year and distributes the card to stationery and gift shops throughout the country. It costs Crystal Lake 50 cents to print each card. The company receives 65 cents for each card sold.

The cards have the current year printed on them; therefore, the unsold cards are generally discarded. The managers want to decide how many cards to print. The dilemma is that if they produce too many cards, the company will have to dispose the unsold ones, but on the other hand, if they run out of cards, they lose the profits they would make otherwise. Based on past experience and forecasts of buying patterns, the probability distribution of the number of cards to be sold nationwide during the next Christmas season is estimated. Spreadsheet 1 keeps the data about the number of cards sold in past years.

The aim of this project is to build a decision support system that will help Crystal Lake Printing Company to decide about the number of cards to produce. This problem is known in the literature as the “News Vendor” problem. We describe below two methods to approach this problem: an optimization method and a simulation model. To learn more about the “News Vendor” problem, we refer the students to Nahmias (2000).

Optimization Model

The optimization approach suggests that Crystal Lake should produce quantity q^* that satisfies the following equation:

$$F(q^*) = \frac{c_u}{c_o + c_u}$$

Where, $F(q)$ is the demand distribution function, c_u is the unit under-stocking cost, and c_o is the unit overstocking cost. Note that, Crystal Lake will loose 50 cents (c_o) for each unsold Christmas card, and the company will loose potentially 15 cents/card (c_u) if they run out of cards.

Simulation Model

Below we describe the steps to build a simulation model for this problem. The managers want to experiment with demand distributions different from the discrete distribution presented in Spreadsheet 1. Therefore, randomly generate demands from normal, uniform, and exponential distributions with a mean and standard deviation as calculated using the data in Spreadsheet 1. In addition, randomly generate demands using normal, uniform, and exponential distributions with different means and standard deviations.

1. In each simulation run, do the following:
 - a. Randomly generate 20 demand points from distribution $F(q)$.
 - b. Calculate the total costs (costs from unmet demand and from extra stocking) for the 20 data points generated, assuming that the data from Spreadsheet 2 present the

actual demands and the generated data is the actual production during these 20 periods.

2. Select the distribution that gave the minimum total costs.

Create a summary table with information about the parameters used to generate each scenario and corresponding costs. This information will help the managers to understand the distribution of $F(q)$. This process will result in better decisions about the number of cards to produce.

Spreadsheets

1. The following spreadsheet presents estimates about the probability distribution of the demand for Christmas cards nationwide.

Quantity Sold	Probability
100,000-150,0000	0.10
150,000-200,0000	0.15
200,000-250,0000	0.25
250,000-300,0000	0.20
300,000-350,0000	0.15
350,000-400,0000	0.10
400,000-450,0000	0.05

2. Spreadsheet 2 presents the actual demand for cards during the last 20 years.

User Interface

1. Build a welcome form.
2. Build a data entry form. In this form, insert a frame titled "Problem Data." The frame includes two option buttons that allow the user to select whether to add or update data. Upon this selection, two option buttons appear that allow the user to choose whether to add/update the data in Spreadsheets 1 or 2. Include a command button that, when clicked on, opens the spreadsheet selected by the user.
3. Insert a data analysis form. In this form include the following controls:
 - a. Two text boxes that present the mean and standard deviation of the annual demand for Christmas cards.
 - b. A text box that presents production quantity calculated using the optimization model.
 - c. A frame titled "Simulation Analysis." The frame includes the following:
 - i. Two text boxes that allow the user to type in the seed for the simulation run and the total number of runs.
 - ii. A command button titled "Select a Distribution." This command button allows the user to choose a particular scenario for the simulation study. When the user clicks on this command button, a combo box appears. The combo box allows the user to select a distribution (such as the empirical distribution presented in Spreadsheet 1; or normal, uniform, or exponential

distributions). Upon selection of a distribution, text boxes appear where the user types in the corresponding parameters of the distribution.

- iii. A command button that, when clicked on, performs the simulation study, records the results, and prompts the user to select whether to perform another simulation run.
4. Create a form that allows the user to open the reports described below. Use option buttons.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the mean and standard deviation of demand.
2. Report the optimal production quantity as found from the optimization model.
3. Create a summary report with the results from each simulation run.
4. Graph the histogram of the probability distribution of demand for Christmas cards.
5. Graph the relationship between production quantity ($q^* = 0, \dots, \text{maximum demand}$) and the total expected costs (under-stocking and overstocking costs). This graph should help the managers (visually) to identify the production quantity that gives the minimum total cost.

Reference

Nahmias, S., "Production and Operations Analysis." *McGraw-Hill/Irwin*, 2000.

64 Warehouse Layout Problem

Problem Description

Warehouses usually have docks for loading and unloading goods and open areas for storing. The trucks that deliver/pick up goods arrive at one of the loading/unloading docks. The warehouse operators must collect or deliver the required goods from their storage area. In managing the warehouse, the operating staff must decide where to store the goods. The best location is close to any dock so that the cost of (a) accessing goods and transferring them to the dock for loading or (b) unloading the goods from the dock and transporting them to the storage area is minimized. As a result, goods compete for storage areas that are closest to the docks.

The aim of this project is to build a decision support system that will allow warehouse managers to decide about allocating the storage space available to goods in such a way that the material handling costs are minimized. To learn more about the warehouse layout problem, we refer the students to Francis et al. (1992). Below we present a mathematical model for this problem.

Mathematical Model

Suppose that there are p items to be stored, and the warehouse has r loading and unloading docks. Let w_{ik} be the total cost per foot incurred when item i is transported from its storage area to dock k . Typically, warehouses store items on pallets, and w_{ik} is proportional to the number of pallet loads of item i moving between dock k and the storage area of i .

This problem is discretized by subdividing the floor area into q square grids of equal size, numbered in any convenient manner from 1 to q . Let F_i be the total number of grids required

to store item i . Assume that $\sum_{i=1}^p F_i = q$. Let d_{ij} denote the distance between dock k and the center of grid j .

The decision variables are as follows:

$$x_{ij} = \begin{cases} 1 & \text{if we store item } i \text{ in grid square } j, \\ 0 & \text{if we do not store item } i \text{ in grid square } j, \end{cases} \quad i = 1, \dots, p; j = 1, \dots, q.$$

The integer programming formulation of this problem is as follows:

$$\min : \sum_{i=1}^p \sum_{j=1}^q c_{ij} x_{ij}$$

subject to :

$$\sum_{j=1}^q x_{ij} = F_i \quad \text{for } i = 1, \dots, p, \quad (1)$$

$$\sum_{i=1}^p x_{ij} = 1 \quad \text{for } j = 1, \dots, q, \quad (2)$$

$$x_{ij} \in \{0,1\} \quad \text{for } i = 1, \dots, p; j = 1, \dots, q. \quad (3)$$

Where, $c_{ij} = \frac{1}{F_i} \sum_{k=1}^r w_{ik} d_{kj}$ is the average cost for locating item i in grid j , assuming that

each item is equally likely to be loaded or unloaded from each dock. The objective is to minimize the average material handling costs at the warehouse. The first set of constraints shows that the space occupied by an item should be equal to the space required for storing the item. The second set of constraints shows that in a particular storage area we can store only one item. The last set of constraints is the binary constraints.

This problem is an instance of the transportation problem. The students can use Excel to solve the problem; however, we suggest that they develop a heuristic procedure for this problem. The following are some insights they can consider while developing their heuristic

approach: The problem assumes that $\sum_{i=1}^p F_i = q$; therefore, we know that all the space

available will be used. Sort the coefficients c_{ij} -s in ascending order. Start with the minimum c_{ij} , and assign item i in grid j . Continue assigning the items to grids. Note that if grid j^* is already assigned to an item, grid j^* will not be available for the next assignments. Also, after each assignment of item i^* the amount of that item available for the next assignments is reduced.

Excel Spreadsheet

1. Build a spreadsheet that presents the distance matrix d_{ij} for $k = 1, \dots, r, j = 1, \dots, q$.
2. Build a spreadsheet that presents the total cost per foot incurred when an item is transported from its storage area to loading/unloading areas (w_{ik} for $i = 1, \dots, p; k = 1, \dots, r$).

User Interface

1. Build a welcome form.
2. Build a data entry form. The following are suggestions to help you design this form. In this form include the following controls:
 - a. Three text boxes where the user types in the total number of storage areas (q), the total number of items (p), and the total number of loading/unloading docks (r).
 - b. Two option buttons that allow the user to choose whether to type in the data or read the data from the file. If the user chooses to type in the data, two tables appear

(one with dimensions r by q and the other p by r) where the user types in the distance matrix (d_{kj}) and material handling costs (w_{ik}). If the user chooses to read the data from a file, upon selection a text box appears where the user types in the name of the file (the location of Spreadsheets 1 and 2).

3. Build a form that allows the user to understand the warehouse layout problem by looking at an example. This form includes the following:
 - a. A problem statement.
 - b. A mathematical formulation of the stated problem.
 - c. The optimal objective function value and the corresponding optimal solution.
4. Build a form that does the following: allows the user to select a method for solving the problem, solves the problem, and presents the corresponding results. Use the following controls: check boxes, command buttons, etc. as needed.
5. The user may be interested to learn about the sensitivity of the optimal solution to changes in the distance between the storage area and loading/unloading docks, the total number of grids needed to store a particular item, unit handling costs, etc. Build a form that allows the user to perform a sensitivity analysis. Use a list box to allow the user to select a parameter for the analyses.
6. Build a form that allows the user to view any of the reports described below. Use option buttons.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the results from the sensitivity analysis.
2. Report the optimal assignment of the items to grids and the optimal material handling costs.
3. Give a graphical representation of the final solution.

Reference

Francis, R.L., McGinnis, L.F., White, J.A., "Facilities Layout and Location." 2nd Ed., *Prentice Hall*, 1992.

Problem Description

In different problem contexts, we wish to assign people to objects, such as employees to jobs, employees to rooms, etc. Each assignment has a value, and we wish to maximize the total value. Consider the case when a firm has hired N_1 graduates to fill N_2 vacant positions. Based on aptitude tests, college grades, and letters of recommendations, the firm calculates a proficiency index u_{ij} for placing candidate i in job j . The objective is to identify an assignment that maximizes the total proficiency score over all jobs.

The main objective of this project is to build a decision support system that would enable the user to assign employees to jobs, employees to rooms, etc. by solving an assignment problem. Below we give a mathematical formulation of the problem.

Mathematical Model

This problem can be formulated as an assignment problem in bipartite graphs. Let A be the set of arcs of the bipartite graph $((i, j) \in A, i \in N_1 \text{ and } j \in N_2)$. The decision variables for this model are as follows:

$$x_{ij} = \begin{cases} 1 & \text{if we assign employee } i \text{ to job } j, \\ 0 & \text{otherwise,} \end{cases} \quad i = 1, \dots, N_1; j = 1, \dots, N_2.$$

Where $|N_1| = |N_2|$.

Our problem states the following:

$$\max : \sum_{(i, j) \in A} c_{ij} x_{ij}$$

subject to :

$$\sum_{\{j: (i, j) \in A\}} x_{ij} = 1 \quad \text{for all } i \in N_1, \quad (1)$$

$$\sum_{\{j: (i, j) \in A\}} x_{ji} = 1 \quad \text{for all } i \in N_2, \quad (2)$$

$$x_{ij} \in \{0, 1\} \quad \text{for all } i \in N_1, j \in N_2. \quad (3)$$

The objective is to maximize the value of the assignment. The first set of constraints shows that an employee will perform a single job. The second set of constraints shows that a job will be performed by a single employee. The third set of constraints is the binary constraints.

The following algorithms can be used to solve this problem: successive shortest path algorithm, Hungarian algorithm, relaxation algorithm, and cost scaling algorithm. For more details about these algorithms and to learn more about the assignment problem and bipartite graphs, we refer the students to Ahuja *et al.* (1993).

Excel Spreadsheet

1. Build a spreadsheet that presents the proficiency index u_{ij} for $i = 1, \dots, N_1$ and $j = 1, \dots, N_2$.

2. Build a spreadsheet that presents node-arc incidence matrix for the bipartite graph.

User Interface

1. Build a welcome form.
2. Build a form that includes the following controls:
 - a. Insert a frame titled "Problem Data." The frame includes the following:
 - i. A text box where the user types in the total number of jobs/employees.
 - ii. Two option buttons that allow the user to choose whether to type in the data or read the data from a file. If the user chooses to type in the data, two tables appear (one with dimensions N_1 by N_2 and the other $N_1 \cdot N_2$ by $N_1 \cdot N_2$) where the user types in the proficiency indexes (u_{ij}) and node-arc incidence matrix. If the user chooses to read the data from a file, upon selection, a text box appears where the user types in the name of the file (the location of Spreadsheets 1 and 2).
 - b. Insert a command button titled "See an Example." When the user clicks on this button, a frame opens that includes the following:
 - i. The statement of an example of the personnel assignment problem.
 - ii. A mathematical formulation of the stated problem.
 - iii. The optimal assignment and corresponding optimal utility of the assignment.
 - c. Insert a frame titled "Solve the Problem." The frame includes a list box that allows the user to choose whether to use the Excel solver or an algorithm (from the list given above) developed by the students to solve the problem. Insert a command button that, when clicked on, solves the problem using the approach selected by the user.
 - d. Insert a frame titled "Sensitivity Analysis." In this frame include a combo box to allow the user to choose a parameter for the sensitivity analysis.
 - e. Insert a frame titled "Reports." Include a number of option buttons to allow the user to open one of the reports presented below.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the results from the sensitivity analysis.
2. Report the optimal assignment of employees to jobs and the optimal value of the assignment.
3. Give a graphical representation of the optimal solution.

Reference

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., "Network Flows: Theory, Algorithms, and Applications." *Prentice Hall*, 1993.

Problem Description

Finding the minimum spanning tree problem is a well-known problem in the area of network optimization. A spanning tree T of network $G = (N, A)$ is a connected acyclic sub-graph that spans all nodes. Given that with each arc of $(i, j) \in A$ is associated a cost c_{ij} , the objective is to find a tree that spans the network and has the minimum cost. The degree-constrained minimum spanning tree problem finds the minimum spanning tree of a network but with the added provision that the tree contains exactly k arcs incident to a given root node, say, node 1.

The degree-constrained minimum spanning tree problem has a number of applications. For example, in computer networking, the root node might be a central processor with a fixed number of ports, and the other nodes might be terminals that need to connect to the processor.

The aim of this project is to build a decision support system that will allow the user to solve the degree-constrained minimum spanning tree problem. Below we provide a network flow formulation of the problem.

Network Flow Formulation

The decision variables for this problem are as follows:

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is selected,} \\ 0 & \text{otherwise,} \end{cases} \quad \text{for all } (i, j) \in A.$$

The following is a network flow formulation of the degree-constrained minimum spanning tree problem.

$$\min: \sum_{(i,j) \in A} c_{ij} x_{ij}$$

subject to :

$$\sum_{j=2}^n x_{1j} = k, \quad (1)$$

$$\sum_{(i,j) \in A} x_{ij} = n - 1, \quad (2)$$

$$\sum_{(i,j) \in A(S)} x_{ij} \leq |S| - 1 \quad \text{for any set } S \text{ of nodes,} \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad \text{for all } (i, j) \in A. \quad (4)$$

Where, $|N| = n$. The objective is to find a tree of minimum cost. The first set of constraints shows that the degree of the root tree should be equal to k . The second constraint shows that the total number of arcs in the final solution should be equal to the total number of nodes in the network minus 1. In a network with n nodes, the corresponding spanning tree has $n-1$ arcs. The third set of constraints is the cycle elimination constraints. The final set of constraints is the binary constraints.

The degree-constrained minimum spanning tree problem can be solved using a Lagrangean relaxation algorithm. To learn more about this problem and Lagrangean relaxation, we refer the students to Ahuja *et al.* (1993).

Excel Spreadsheet

1. Build a spreadsheet that presents the cost c_{ij} for all $(i,j) \in A$.
2. Build a spreadsheet that presents the node-arc incidence matrix of the graph G .

User Interface

1. Build a welcome form.
2. Build a form that includes the following controls:
 - a. Insert a frame titled "Problem Data." The frame includes the following:
 - i. Three text boxes where the user types in the total number of nodes, the total number of arcs, and the degree of the root node (k).
 - ii. Two option buttons that allow the user to choose whether to type in the data or read the data from the file. If the user chooses to type in the data, two tables appear (one with dimensions A by 1 and the other n by A) where the user types in the cost coefficients (c_{ij}) and node-arc incidence matrix. If the user chooses to read the data from a file, upon selection, a text box appears where the user types in the name of the file (the location of Spreadsheets 1 and 2).
 - b. Insert a command button titled "See an Example." When the user clicks on this button, a frame opens that includes the following:
 - i. The statement of an example of the degree-constrained minimum spanning tree problem.
 - ii. A mathematical formulation of the stated problem.
 - iii. The optimal spanning tree and corresponding optimal cost.
 - iv. A graphical representation of the optimal degree-constrained spanning tree.
 - c. Insert a frame titled "Solve the Problem." The frame has two option buttons that allow the user to choose either Excel or the Lagrangean relaxation algorithm to solve the problem. Insert a command button that, when clicked on, solves the problem using the approach selected by the user.
 - d. Insert a frame titled "Sensitivity Analysis." In this frame include a combo box to allow the user to choose a parameter, such as the degree of the root node or arc costs, for the sensitivity analyses.
 - e. Insert a frame titled "Reports." Include a number of option buttons to allow the user to open one of the reports presented below.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the results from the sensitivity analysis.
2. Report the optimal spanning tree and the optimal objective function value.
3. Give a graphical representation of the optimal solution.

Reference

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., "Network Flows: Theory, Algorithms, and Applications." *Prentice Hall*, 1993.

Problem Description

The production planning problem is concerned with finding the best use of scarce resources (people, machinery, space, etc.) in order to meet customer demand at the least possible cost. Suppose that we are producing K items over a planning horizon that contains T periods. Suppose that we produce the items on the same machine and that we can produce at most one item in each period. We would like to find the least cost production plan that will satisfy demand d_{kt} for every item ($k = 1, \dots, K$) in each period ($t = 1, \dots, T$).

The aim of this project is to build a decision support system that will allow the user to find a production plan that satisfies demand at the minimum cost. We provide a mixed-integer programming formulation of this problem.

Integer Formulation

The following notation is used:

c_{kt}	the unit production cost
h_{kt}	the unit inventory holding cost
F_{kt}	set-up cost
d_{kt}	demand for product k in period t
P_{kt}	production capacity for product k in period t .

The decision variables for this problem are as follows:

x_{kt}	amount of product k produced in period t
h_{kt}	amount of product k in inventory in the end of period t

$$z_{kt} = \begin{cases} 1 & \text{if the machine is set-up in period } t \text{ to produce product } k, \\ 0 & \text{otherwise,} \end{cases}$$

The objective is to minimize the total production costs, inventory holding costs, and set-up costs during the planning horizon. The first set of constraints shows that in each time period, machines will be set up once and a single product will be produced. The second set of constraints is the flow conservation constraints and shows that production in a particular period plus the inventory from the last period should be equal to demand plus inventory in the end of the period. The third set of constraints shows that the amount of product k produced in period t should be less than or equal to production capacity for this product. The fourth set of constraints is the non-negativity constraints, and the final set of constraints is the binary constraints.

$$\min : \sum_{k=1}^K \sum_{t=1}^T c_{kt} x_{kt} + \sum_{k=1}^K \sum_{t=1}^T h_{kt} I_{kt} + \sum_{k=1}^K \sum_{t=1}^T F_{kt} z_{kt}$$

Subject to :

$$\sum_{k=1}^K z_{kt} \leq 1 \quad \text{for } t = 1, \dots, T, \quad (1)$$

$$x_{kt} + I_{k,t-1} - I_{kt} = d_{kt} \quad \text{for } k = 1, \dots, K; t = 1, \dots, T, \quad (2)$$

$$x_{kt} \leq P_{kt} z_{kt} \quad \text{for } k = 1, \dots, K; t = 1, \dots, T, \quad (3)$$

$$x_{kt}, I_{kt} \geq 0 \quad \text{for } k = 1, \dots, K; t = 1, \dots, T, \quad (4)$$

$$z_{kt} \in \{0,1\} \quad \text{for } k = 1, \dots, K; t = 1, \dots, T. \quad (5)$$

The students can use Excel to solve this problem or the Lagrangean relaxation technique. Note that constraints (1) are the only constraints in this model that link various items together. Relaxing these constraints via Lagrangean multipliers (λ_t) gives the following Lagrangean relaxation problem:

$$\min : \sum_{k=1}^K \sum_{t=1}^T c_{kt} x_{kt} + \sum_{k=1}^K \sum_{t=1}^T h_{kt} I_{kt} + \sum_{k=1}^K \sum_{t=1}^T (F_{kt} + \lambda_t) z_{kt} - \sum_{t=1}^T \lambda_t$$

Subject to :

(2), (3), (4) and (5)

The Lagrangean relaxation problem separates into K single-product production planning problems that are easier to solve. For the single-product problems, the production and inventory holding costs are the same as those of the original problem, and the fixed cost in the relaxation is λ_t units more than in the original problem. To learn more about the Lagrangean relaxation technique, we refer the students to Ahuja *et al.* (1993).

Excel Spreadsheet

Build Excel spreadsheets to present the following data:

1. Production unit cost c_{kt} for $k = 1, \dots, K, t = 1, \dots, T$.
2. Inventory holding cost h_{kt} for $k = 1, \dots, K, t = 1, \dots, T$.
3. Fixed cost F_{kt} for $k = 1, \dots, K, t = 1, \dots, T$.
4. Demand d_{kt} for $k = 1, \dots, K, t = 1, \dots, T$.
5. Production capacity P_{kt} for $k = 1, \dots, K, t = 1, \dots, T$.

User Interface

1. Build a welcome form.
2. Build a form that includes the following controls:
 - a. A frame titled "Problem Data." This frame includes the following:
 - i. Two text boxes where the user types in the following: the total number of products (K) and the total number of time periods (T).

- ii. Two option buttons that allow the user to choose whether to type in the data or read the data from a file. If the user chooses to type in the data, five tables appear (each with dimensions K by T) where the user types in the following: the production unit costs (C_{kt}), the unit inventory holding costs (h_{kt}), the set-up costs (F_{kt}), the demand (d_{kt}), and the production capacity (P_{kt}). If the user chooses to read the data from a file, upon the selection, a text box appears where the user types in the name of the file (the location of Spreadsheets 1 to 5).
- b. Insert a frame titled "Solve the Problem." The frame has two option buttons that allows the user to select either Excel or the Lagrangean relaxation algorithm to solve the problem. Insert a command button that, when clicked –on, solves the problem using the approach selected by the user.
- c. Insert a frame titled "Sensitivity Analysis." In this frame include a combo box to allow the user to choose a parameter for the sensitivity analyses.
- d. Insert a frame titled "Reports." Include a number of option buttons to allow the user to open one of the reports presented below.
- e. Insert a command button titled "See an Example." When the user clicks on this button, a frame opens that includes the following:
 - i. A statement of an example for the multi-item production planning problem.
 - ii. A mathematical formulation of the stated problem.
 - iii. The optimal production plan and corresponding optimal cost.
 - iv. The results from the iterations of the Lagrangean relaxation algorithm.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the results from the sensitivity analysis.
2. Report the optimal solution and the optimal objective function value.
3. Give a graphical representation of the total costs per period during the planning horizon.
4. Present the results from the Lagrangean relaxation algorithm.

Reference

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., "Network Flows: Theory, Algorithms, and Applications." *Prentice Hall*, 1993.

Problem Description

The U.S. Bureau of Land Management (BLM) manages millions of acres of public rangelands. A significant part of this land is used to grow vegetation that will be consumed by animals (both wild and domesticated). The management at BLM is concerned about determining the optimal number of animals that the land can support, given the land inventory and the dietary requirements for different types of livestock.

The aim of this project is to build a decision support system that enables BLM to manage public rangelands. Below we present a mathematical formulation of this problem.

Mathematical Formulation

The following notation is used:

I	total number of vegetation types
J	total number of livestock types
a_{ij}	amount of vegetation type i consumed per livestock type j
γ_j	total amount of vegetation to be consumed by livestock type j
b_i	an upper bound on the amount of vegetation type i produced
g_{ij}	the maximum requirement of vegetation i for a balanced diet for livestock type j
f_{ij}	presents the minimum requirement of vegetation type i for a balanced diet for livestock type j .

The decision variables are as follows:

x_j	the number of livestock type j supported by the public rangelands.
-------	--

The objective is to find the optimal number of livestock of different types that the public rangeland can support, given the vegetation inventory and the dietary requirements for each livestock type. The first set of constraints shows that the total consumption of each vegetation type should not be greater than the amount available in the inventory. The second set of constraints shows that the total consumption of vegetation type i consumed by livestock type j should not be greater than a predefined limit γ_j . The third and fourth sets of constraints relate to the dietary requirements for each animal type. These constraints show that the ratio of the intake of vegetation type i to the total intake, for each livestock type, should be between f_{ij} and g_{ij} .

$$\max : \sum_{j=1}^J x_j$$

Subject to :

$$\sum_{j=1}^J a_{ij} x_j \leq b_i \quad i = 1, \dots, I, \quad (1)$$

$$\sum_{i=1}^I a_{ij} x_j \leq \gamma_j \quad j = 1, \dots, J, \quad (2)$$

$$\frac{a_{ij}}{\gamma_j} x_j \leq g_{ij} \quad i = 1, \dots, I; j = 1, \dots, J, \quad (3)$$

$$\frac{a_{ij}}{\gamma_j} x_j \geq f_{ij} \quad i = 1, \dots, I; j = 1, \dots, J. \quad (4)$$

$$x_j \geq 0 \quad i = 1, \dots, I; j = 1, \dots, J. \quad (5)$$

Spreadsheets

1. Build a spreadsheet that presents coefficients a_{ij} , for $i = 1, \dots, I$ and $j = 1, \dots, J$.
2. Build a spreadsheet that presents coefficients γ_j , for $j = 1, \dots, J$.
3. Build a spreadsheet that presents coefficients b_i , for $i = 1, \dots, I$.
4. Build a spreadsheet that presents coefficients g_{ij} , for $i = 1, \dots, I$ and $j = 1, \dots, J$.
5. Build a spreadsheet that presents coefficients f_{ij} , for $i = 1, \dots, I$ and $j = 1, \dots, J$.

User Interface

1. Build a welcome form.
2. Build a form that includes the following controls:
 - a. A frame titled "Problem Data." The frame includes the following:
 - i. Two text boxes where the user types in the total number of livestock types (J) and the total number of vegetation types (I).
 - ii. Two option buttons that allow the user to choose whether to type in the data or read the data from a file. If the user chooses to type in the data, five tables appear (three with dimensions I by J , one with dimensions 1 by I , and one with dimensions 1 by J) where the user types in the coefficients a_{ij} , γ_j , b_i , g_{ij} , and f_{ij} . If the user chooses to read the data from a file, upon selection, a text box appears where the user types in the name of the data file(s) (the location of Spreadsheets 1 to 5).
 - b. Insert a command button titled "Solve the Problem." When this button is clicked on, the land management problem is solved using the Excel solver.
 - c. Insert a frame titled "Sensitivity Analysis." In this frame, include a combo box to allow the user to choose a parameter for the sensitivity analyses.

- d. Insert a frame titled “Reports.” Include a number of option buttons to allow the user to open one of the reports presented below.
- e. Insert a command button titled “See an Example.” When the user clicks on this button, a frame opens that includes the following:
 - i. The problem statement of an example.
 - ii. The mathematical formulation of the stated problem.
 - iii. The optimal objective function value and the corresponding optimal solution.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the results from the sensitivity analysis.
2. Report the optimal number of each type of livestock to be supported.

Problem Description

Companies are concerned about managing customer relationships, as it is vital for their wellbeing. Using the sales force, companies keep contact with the customers, inform the customers about new products, get feedbacks about existing products, etc. A number of studies have shown that there is a positive relationship between the revenues generated by a company and the efforts of the sales force. Companies use a number of decision support tools to decide about the size of the sales force. Syntex Labs (Winston and Albright, 2002) has estimated about a \$1 million increase in profitability per year since they started using a non-linear programming (NLP) model to allocate the sales force among the company's seven main drugs.

The main objective of this project is to build a decision support system that will enable companies to manage customer relationships. Below we present a mathematical model that can be used in making decisions about the size and allocation of the sales force to products.

Mathematical Model

The relationship between the size of the sales force and the sales revenues generated is described by the following equation

$$R = a + \frac{(b - a)S^c}{d + S^c} \quad (1)$$

Where, a , b , c , and d are constants, R presents the revenues generated from sales, and S presents the sales force efforts. This relationship has the property of diminishing returns; therefore, each extra unit of S contributes less and less to R . The curve that presents the relationship between S and R has an S-shape (starts out flat, gets steep, and then flattens out).

In order to estimate the sales response function, the managers estimate the revenues that would be generated under the following scenarios: no sales effort is assigned to the final product; sales efforts assigned are cut in half; sales efforts stay at the current level; sales efforts are increased by 50%; sales force efforts saturate the market; etc. For each scenario, the corresponding predicted sales revenues are calculated using equation (1). In order to calculate the predicted sales revenues, initially, we assign values to the constants, a , b , c , and d . We calculate the error (the difference) between the actual and predicted sales. The goal-seek feature of Excel is then used to find the values of constants a , b , c , and d that minimize the sum squared error. To learn more about the sales force allocation problem and S-shape curves, we refer the students to Winston and Albright (2002).

Sensitivity Analysis

The sales response function estimates revenues as a function of the size of the sales force. Increasing the sales force increases the total revenues generated at a diminishing rate. On the other hand, increasing the sales force increases the salary expenses. The profits are calculated by subtracting from the sales revenues the production costs and the annual salary expenses. The objective of the sensitivity analysis is to identify the sensitivity of the profits generated to the size of the sales force.

Spreadsheets

1. Build a spreadsheet that presents the annual salary and other related data about the sales persons.
2. Build a spreadsheet that presents the following data about each product: identification number, name, average annual revenues, price, production unit cost, descriptions, etc.
3. Build a spreadsheet that keeps record of sales. The spreadsheet presents the following data about each sale: the date of the sale, the product sold, and the revenues generated.
4. Build a spreadsheet that presents historical data about the contacts of sales persons with customers. This spreadsheet includes the following: the date of the contact, the identification number of the customer contacted, the name of the sales person, the length of the conversations, descriptions, etc.
5. Build a spreadsheet that keeps data about the customers. This data consists of the following: identification number, name, address, description of the business, etc.

User Interface

1. Build a welcome form.
2. Build a form titled “Add/Delete/Update Data.” The form includes five option buttons. The option buttons allow the user to open one of the spreadsheets described above. Upon selection, another frame opens that has three option buttons to allow the user to choose whether to add, delete or update the data in the selected spreadsheet.
 - a. If the user chose to add data, a form opens that consists of a number of text boxes and a command button. The user enters the data in the text boxes and clicks on the command button to submit the information.
 - b. If the user chose to delete data, a search form opens that allows the user to identify the data that should be deleted. The form has a combo box that allows the user to choose a field of the spreadsheet, a text box for the user to type in a key word, and a command button. A search for the key word on the selected field is performed and the results are presented to the user. The user clicks on the command button to delete the selected information from the spreadsheet.
 - c. If the user chose to update data, a search form opens that enables the user to identify the data that should be updated. The form has a combo box that allows the user to choose a field of the spreadsheet, a text box to type in a key word, and a command button. A search for the key word on the selected field is performed and the results are presented to the user. The user updates the information and clicks on the command button to submit the updated information.
3. Build a form titled “Analyze the Data.” The form includes a number of option buttons and a command button. The option buttons enable the user to choose to calculate one of the following statistics:
 - a. The average number of times a customer was contacted during the last month.
 - b. The total number of customer contacts per day during the last month.
 - c. The total number of customer contacts during the last month per sales person.
 - d. The average contact time per customer during the last month.

- e. The total number of customers who will be contacted next week. This number can be identified using historical data from the previous week/month/ year.
- f. The total revenues generated per day per product during the last month.
- g. The average cost per customer call.
- h. For each product, calculate the total number of calls, revenues generated, and sizes of the sales force assigned.

This frame includes a command button. When the user clicks on the command button, the statistic(s) selected are calculated and presented.

4. Build a form titled “Estimate the Sales Response Function.” The form includes n option buttons (n is the total number of products) that allow the user to select one or more products to estimate the corresponding sales response function. Include a command button that, when clicked on, uses the minimum sum of squared errors method to estimate the sales response function and displays the corresponding results. The user is prompted to type in the initial values for parameters a , b , c , and d . Note that historical data about the number of calls, revenues, and work force per product will help to identify the sales response function.
5. Build a form titled “Estimate Profits.” The form includes n text boxes and a command button. When the user clicks on the command button, the annual profits per product are presented in the text boxes. To calculate the profits per product, one should subtract from the total revenues generated the total production costs and salaries of sales persons assigned to that product.
6. Build a form titled “Sensitivity Analysis.” The form includes three text boxes, a combo box, and a command button. The user types in the text boxes the minimum and maximum size of the sales force and the step size. The combo box allows the user to select a product. When the user clicks on the command button, the profits from the selected product for different sizes of the sales force are calculated and presented.
7. Build a form titled “Reports.” This form allows the user to open one of the reports presented below.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the following:
 - a. The total number of times each customer was contacted during the last month.
 - b. The total number of customers contacted per day during the last month.
 - c. The total number of customer contacts per sales person during the last month.
 - d. The average contact time per customer during the last month.
 - e. The number of customers who should be contacted next week.
 - f. The total revenues generated per day per product during the last month.
 - g. The average cost per customer call.

- h. For each product, report the total number of calls, revenues generated, and sizes of the sales force assigned.
2. Graph the sales response function. Present the estimates for the constants a , b , c , and d .
3. Report the total annual revenues, production costs, sales force salary, and profits during the current year.
4. Report the results from the sensitivity analysis. These results allow the managers to identify the optimal size of the sales force.

Reference

Winston, W. L., Albright, S.C. "Practical Management Science: Spreadsheet Modeling and Applications Update." 2nd Ed., *Duxbury Press*, 2002.

Problem Description

Bin packing is a well-known problem in the area of combinatorial optimization. This problem considers packing a set of items into a number of bins such that the total weight of the bins, volume, etc. does not exceed some maximum value. This is an *NP*-hard problem. The following is a list of approximation algorithms that have been developed to solve this problem: first-fit, best-fit, first-fit decreasing algorithms, etc.

Manufacturing companies often face this problem. For instance, suppose we need a number of pipes of different, specific lengths to plumb a house and we can buy pipe stock that is 5 meters long each. In order to find how to cut these pipes to waste as little as possible (minimize the cost of pipe), one formulates this problem as a bin packing problem and solves it using one of the algorithms described below.

Mathematically the problem is formulated as follows: Given a finite set of elements $E = \{e_1, \dots, e_n\}$ with associated weights $W = \{w_1, \dots, w_n\}$ such that $0 \leq w_i \leq w(\text{bin})$; partition E into N subsets such that the sum of weights in each partition is at most $w(\text{bin})$ and that N is the minimum.

Approximation Algorithms

First Fit

- Label bins as 1, 2, 3, . . .
- Objects are considered for packing in the order 1, 2, 3, . . .
- Pack object i in bin j where j is the least index such that bin j can contain object i .

Best Fit

The same as FF, except that when object i is to be packed, find the bin which after accommodating object i will have the least amount of space left.

First Fit Decreasing

Reorder objects so that $w_i \geq w_{i+1}$, $1 \leq i \leq n$; then use FF.

Best Fit Decreasing

Reorder objects so that $w_i \geq w_{i+1}$, $1 \leq i \leq n$; then use the best fit algorithm.

Note the following: Packing generated by either first fit or best fit algorithms uses no more than $(17/10)OPT + 2$ bins. Packing generated by either first fit decreasing or best fit decreasing algorithms uses no more than $(11/9)OPT + 4$ bins.

To learn more about the bin packing problem and the approximation algorithms, we refer the students to Garey and Johnson (2000).

User Interface

1. Build a welcome form.
2. Build a form that includes the following controls:

- a. Insert a data entry form. The form consists of three text boxes and a command button. The user types in the text boxes the following information: the total number of bins available, the size of the bin, and the total number of objects to be packed. The user clicks on the command button to submit this data. Upon submission, a table appears (with dimensions 1 by *the total number of objects*) that allows the user to type in the weight of each object.
 - b. Insert a command button titled “See an Example.” When the user clicks on this button, a form opens that includes the following:
 - i. A statement of an example of the bin packing problem.
 - ii. A mathematical formulation of the stated problem.
 - iii. The solution to this problem found using the approximation algorithms described above. Describe each step of the algorithm.
 - c. Insert a command button titled “Solve the Problem.” When the user clicks on the command button, a frame appears that includes four option buttons and a command button. The option buttons allow the user to select one of the algorithms described above to solve the problem. When the user clicks on the command button, the problem is solved using the method chosen by the user, a report about the solution is created, and the user is prompted to choose whether to see the corresponding report.
3. Build a form titled “Sensitivity Analysis.” The form has a number of option buttons that allow the user to select one of the following parameters for the purpose of the sensitivity analysis: the size of the bin or the weight of an object. For example, the user might be interested to know how the total number of bins needed to pack the items changes when the size of the bin increases/decreases.
 4. Build a form titled “Reports.” The form has a number of option buttons that allow the user to open one of the reports presented below.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Present the following data about the solution to the bin packing problem using the approximation algorithms:
 - a. The total number of bins used.
 - b. The unused space for each bin.
2. Present the results from the sensitivity analysis.
3. Graph the relationship between the bin size and the total number of bins required to pack all objects.

Reference

Garey, M. R., Johnson, D. S., “Computers and Intractability: A Guide to the Theory of NP-Completeness.” *W.H. Freeman and Company*, 2000.

71 *Managing Inventories*

Problem Description

Managing inventories is a major concern for managers of all types of businesses. For companies that operate on relatively low profit margins, poor inventory management can seriously undermine the business. The objective of this project is to build a decision support system that would allow companies to manage their inventories. We provide a summary of mathematical models that can be used to identify the optimal amount of cycle, pipeline, and safety stock inventory to be carried in the system under different scenarios (for example, when demand is constant or not constant). To learn more about managing inventories, we refer the students to Nahmias (2000).

Mathematical Models

In the mathematical models presented below, we use the following notation:

Q	lot size
TC	total annual costs (inventory and ordering costs)
D	annual demand
DL	demand during lead-time
S	order/setup cost
h	unit inventory holding cost
z	service level
σ_P	standard deviation during protection interval.

The total cost function is calculated as follows: $TC = (Q/2)*h + (D/Q)*S$.

The following are some of the inventory control systems used by companies:

- In the case that the demand rate is constant and known with certainty, inventories are managed as follows: when the reorder point is reached, the economic order quantity (EOQ) is ordered. The EOQ is calculated using the following formula: $EOQ = \sqrt{2DS/h}$. The reorder point is set equal to demand during lead-time. These systems do not carry safety stock.
- In the case that demand rate is not constant, the following two inventory control systems could be used:
 - Continuous review system.* In this case, the inventories are continuously monitored; when the reorder point is reached, EOQ is ordered. The reorder point is equal to demand during lead-time plus safety stock.
 - Periodic review system.* In this case, the inventories are reviewed periodically rather than continuously; at the review time, an order is placed to bring the inventory position up to the target level T . The target level is set equal to demand during protection interval plus safety stock. The inventory position is equal to on-hand inventory, plus scheduled receipts, minus backorders.

Note the following: cycle inventory = $Q/2$; pipeline inventory = DL ; safety stock = $z\sigma_p$.

User Interface

1. Build a welcome form.
2. Build a form titled "Problem Data." The following are suggestions to help you design the form. The form has three text boxes, a combo box, and a command button. The user types in the text boxes the following data: order/setup cost, length of the lead-time, and unit inventory holding costs. The combo box allows the user to select whether demand is constant or not. Upon selection, in the case that demand is constant, a text box appears where the user types in the annual demand; otherwise, three text boxes appear where the user types in the average demand, corresponding standard deviation, and service level required. The user clicks on the command button to submit the data.
3. Build a form titled "Inventory Analyses." The form has three option buttons that allow the user to choose the type of system being analyzed: the constant demand system, the periodic review system, or the continuous review system.
 - a. In the case that the user chose the constant demand model, a frame appears that includes a number of option buttons and a command button. The option buttons allow the user to calculate the following: EOQ, reorder point, annual inventory holding costs, annual ordering costs, total costs, cycle inventory, pipeline inventory, time between orders, and number of orders in a year. When the user clicks on the command button, the calculations are performed and the corresponding results are presented to the user.
 - b. In case that the user chose the continuous review system, a frame appears that includes a number of option buttons and a command button. The option buttons allow the user to calculate the following: EOQ, reorder point, annual inventory holding costs, annual ordering costs, total costs, safety stock, cycle inventory, pipeline inventory, average time between orders, and average number of orders in a year. When the user clicks on the command button, the calculations are performed and the corresponding results are presented to the user.
 - c. In the case that the user chose the periodic review system, a frame appears that includes a number of option buttons and a command button. The option buttons allow the user to calculate the following: annual inventory holding costs, annual ordering costs, total costs, safety stock, cycle inventory, pipeline inventory, target inventory level, length of the order period, and number of orders in a year. When the user clicks on the command button, the calculations are performed and the corresponding results are presented to the user.
4. Build a form titled "Reports." The form has a number of option buttons that allow the user to open the reports described below.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the economic order quantity for this problem.
2. For each of the inventory management systems, report the following:
 - a. The annual inventory holding costs, annual ordering costs, and annual total costs.
 - b. The EOQ and reorder point.
 - c. The total number of orders placed in a year.
 - d. The average time between orders.
 - e. The amount of safety stock, cycle inventory, and pipeline inventory.
3. Graph the relationship between the following:
 - a. Annual inventory holding costs and order quantity.
 - b. Annual order costs and order quantity.
 - c. Annual total costs and order quantity.

Reference

Nahmias, S., "Production and Operations Analysis with Student CD-Rom." 4th Ed., *McGraw-Hill/Irwin*, 2000.

Problem Description

Material requirement planning (MRP) is a key element in managing resources in a manufacturing environment. MRP systems were developed to help companies manage dependent demand inventory and schedule replenishment orders. MRP systems have proven to be beneficial to many companies.

The aim of this project is to build a support system that would generate material requirements plans for a manufacturing company. This support system should be build using the principles of MRP systems. We use a simple example to show how MRP systems work; however, to learn more about these systems we refer the students to Krajewski and Ritzman (2002) and Nahmias (2000).

MRP System

An MRP system translates the master production schedule (MPS), bill of materials (BOM), and inventory records into a material requirement plan that specifies the replenishment schedules of all the subassemblies, components, and raw materials needed for the final product. We illustrate the inputs of the system and the final MRP plans using an example. We consider a manufacturing company that produces chairs. We present the BOM, MPS, and MRP plans for the ladder-back chairs produced by this company (Krajewski and Ritzman, 2002).

MPS presents the lot size and due date for the final products.

	April				May			
	1	2	3	4	5	6	7	8
Ladder-back chair	150					150		
Kitchen chair				120			120	
Desk chair		200	200					200
Aggregate production plan for chair family	670				670			

Figure 1: MPS for a Family of Chairs.

The BOM presents the components required to manufacture the final product, the parent-component relationships, and the usage quantities. Inventory records present an item's lot-size policy, lead-time, and other time-phased data. Below we present the BOM, MPS, inventory records, and the material requirements plan for a Ladder-Back Chair.

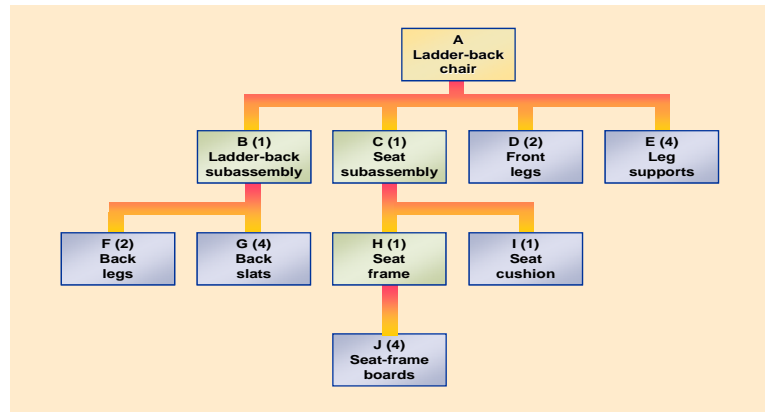


Figure 2: BOM for a Ladder-Back Chair.

Item: C Description: Seat subassembly		Lot Size: 230 units Lead Time: 2 weeks							
		Week							
		1	2	3	4	5	6	7	8
Gross requirements		150	0	0	120	0	150	120	0
Scheduled receipts		230	0	0	0	0	0	0	0
Projected on-hand inventory	37	117	117	117	-3	-3	-153	-273	-273
Planned receipts									
Planned order releases									

Figure 3: Inventory Records for the Seat Subassembly of the Ladder-Back Chair.

Item: Seat subassembly Lot size: 230 units		Week							
Lead time: 2 weeks		1	2	3	4	5	6	7	8
Gross requirements		150	0	0	120	0	150	120	0
Planned receipts					230			230	
Planned order releases			230			230			

Item: Seat frames Lot size: 300 units		Week							
Lead time: 1 week		1	2	3	4	5	6	7	8
Gross requirements		0	230	0	0	230	0	0	0
Scheduled receipts		0	300	0	0	0	0	0	0
Projected on-hand inventory	40	40	110	110	110	180	180	180	180
Planned receipts					300				
Planned order releases				300					

Item: Seat cushion Lot size: L4L		Week							
Lead time: 1 week		1	2	3	4	5	6	7	8
Gross requirements		0	230	0	0	230	0	0	0
Scheduled receipts		0	0	0	0	0	0	0	0
Projected on-hand inventory	0	0	0	0	0	0	0	0	0
Planned receipts			230			230			
Planned order releases		230			230				

Figure 4: Part of MRP Plans for the Ladder-Back Chair.

Excel Spreadsheets

1. Build a spreadsheet that presents for each final product the name and quantity of all the components used to produce the product.
2. Build a spreadsheet that presents for each item the amount on hand and scheduled receipts.
3. Build a spreadsheet that presents for each final product the amount and the due dates promised to customers.

User Interface

1. Build a welcome form.
2. Build a form titled "Bill of Materials." The following are suggestions to help you design this form. Insert a combo box titled "Choose a Product." The combo box allows the user to select one of the final products listed. Upon selection, a list of the components used to produce the product is displayed. The user should be able to select more than one component from the list as, more than one component may be used to produce the final product. Include two command buttons in this frame, one titled "Continue" and the other "Submit." If the user clicks on the "Continue" command button, list boxes appear, one for each component. The user selects for each item the corresponding components and submits the information. The same process continues until we have defined all parent-component relationships of the final product. If the user clicks on the "Submit" command button, then a form titled "Enter Usage Quantities" appears. This form presents all the components used by a product. Next to the name of each component, insert a text box for the user to type in the quantity used. Insert a command button that, when clicked on, submits the data and creates the bill of material for the selected product.
3. Build a form titled "Master Production Schedule." Insert a command button. When the user clicks on this button, the data about the amount of the final product due and the corresponding due date (Spreadsheet 3) are used to prepare the MPS. The MPS is presented to the user.
4. Build a form titled "Inventory Records." The frame consists of a combo box and a command button. The combo box lists the names of the final products and components. The user selects a product or a component and clicks on the command button to generate the inventory records for the selected item.
5. Build a form titled "See an Example." This form presents a simple example that demonstrates how the MRP systems work.
6. Build a form titled "Create the MRP Plans." The frame includes the following:
 - a. A text box where the user types in the length of the lead-time.
 - b. A combo box that allows the user to select a lot-sizing rule (FOQ, POQ, or L4L).
 - c. A command button that, when clicked on, uses the BOM, MPS, inventory records, lead-time, and lot-sizing rules to produce the MRP plans.
7. Build a form titled "Reports." The form has a number of option buttons and a command button. The option buttons allow the user to select one of the reports created. When the user clicks on the command button, the selected report is opened.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the BOM for the final product(s) produced by the company.
2. Report the MPS for the final product(s) produced by the company.
3. Report the MRP plans created using the three different lot-sizing rules.

Reference

Nahmias, S., "Production and Operations Analysis with Student CD-Rom." 4th Ed., *McGraw-Hill/Irwin*, 2000.

Krajewski, L.J., Ritzman, L.P. "Operations Management: Strategy and Analysis." 3rd Ed., *Prentice Hall*, 2000.

Problem Description

Material requirement planning (MRP) systems are widely used by companies to manage resources in a manufacturing environment. The output of an MRP system is a production plan that specifies the amount of each final product and subassemblies produced, the exact timing of the production lot sizes, and the final schedule of completion. The following are inputs for an MRP system: the bill of materials (BOM), master production schedule (MPS), and inventory records. In order to create MRP plans, different lot-sizing rules can be used.

In a previous case study we built a decision support system that creates MRP plans for manufacturing companies. The focus of this project is a bit different and narrower. In this case study, we will concentrate on building a support system that uses different lot-sizing techniques to plan order releases and deliveries given the net requirements for the product.

Below we present a sample example and describe different lot-sizing techniques used in MRP systems. To learn more about these techniques and MRP systems, we refer the students to Nahmias (2000).

Lot-sizing Techniques

We first present a simple example that illustrates how to use the EOQ (Economic Order Quantity) lot-sizing rule to plan order releases and deliveries given net requirements for the final product (Table 1). Next we will discuss some of the most popular lot-sizing rules used.

The Harmon Music Company produces a variety of wind instruments (Nahmias, 2000). One of the instruments produced is the model 185C trumpet. Valve casing subassembly is one of components used to produce the trumpet. It takes seven weeks to produce one trumpet. Production lead-time is four weeks. In Table 1, gross requirements present the demand for valve casing subassembly from weeks 8 to 17. Net requirements are calculated by subtracting from the gross requirements the on-hand inventory and scheduled receipts. Because of the lead-time, the net requirements are shifted four weeks in advance. The product is produced in lots of 139 units (EOQ = 139). The EOQ amount is calculated using the formula $EOQ = \sqrt{2DS/h}$, where D is the annual demand, S is the set-up cost and h is the unit inventory holding cost.

Setting the lot-size equal to EOQ is not the only method used by companies to manage their production schedule. For example, the periodic order quantity (POQ) method sets the size of each production lot equal to the requirements for a fixed number of periods; and the lot-for-lot (L4L) rule sets the lot size in such a way that no inventory is carried from one period to the other.

Silver-Meal is a heuristic used to determine lot sizes in MRP systems. Let $C(T)$ be the average holding and set-up cost per period if the current order spans T periods. Let (r_1, \dots, r_n) be the requirements over the n -period horizon. Note the following: $C(1) = S$; $C(2) = (S + hr_2)/2$; in general $C(j) = (S + hr_2 + 2hr_3 + \dots + (j-1)hr_j)/2$. When $C(j) > C(j-1)$, we stop, set $y_1 = r_1 + r_2 + \dots + r_{j-1}$, and begin the process starting at period j .

Least Unit Cost is a heuristic similar to the Silver-Meal method, except that instead of dividing the cost over j periods by the number of periods, j , we divide it by the total number of units demanded through period j , $r_1 + r_2 + \dots + r_j$.

Finally, the part period balancing method sets the order horizon equal to the number of periods that most closely matches the total holding cost with the setup cost over that period.

Table 1: MRP Plan for the Valve Casing Subassembly of the Model 185C Trumpet.

Week	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Gross Requirements					77	42	38	21	26	112	45	14	76	38
Scheduled Receipts					12		6	9						
On-Hand Inventory														
Net Requirements					42	42	32	12	26	112	45	14	76	38
Time-phased net requirements	42	42	32	12	26	112	45	14	76	38				
Planned order release (EOQ)	139	0	0	0	139	0	139	0	0	139				
Planned Deliveries					139	0	0	0	139	0	139	0	0	139
Ending inventory					97	55	23	11	124	12	106	92	16	117

User Spreadsheets

Build a spreadsheet that presents for each final product the gross requirements, scheduled receipts, and on-hand inventory.

User Interface

1. Build a welcome form.
2. Build a form that includes the following controls:
 - a. Insert a frame titled “Problem Data.” The frame includes a combo box, a text box, and two option buttons. The combo box allows the user to select a product. The text box allows the user to type in the length of the lead-time period. The option buttons allow the user to select whether to type in the data for the problem or read it from a file.
 - i. In the case that the user chose to read the data from a file, a text box appears where the user types in the location and name of the file.
 - ii. In the case that the user chose to type in the data, a form opens that consists of the followings controls: a text box that allows the user to type in the length (n) of the planning horizon. Upon submission of this information, a table similar to Table 1 with n columns and nine rows appears. The user types in the table the data about gross requirements, scheduled receipts, and on hand inventory; a command button that, when clicked –on, calculates and displays the net requirements and the time-phased net requirements for the selected product.
 - b. Insert a combo box titled “Choose a Lot-Size Method.” The combo box allows the user to choose a lot-size rule to identify the planned order releases.
 - c. Insert a command button titled “See an Example.” When the user clicks on this command button, a form opens that presents a simple example that demonstrates how different lot-sizing rules can be used to plan order releases in an MRP system.

- d. Insert a command button titled "Submit" that, when clicked –on, uses the information provided by the user about the lead-time, lot-sizing rule, and net requirements to create MRP plans.
- e. Insert a frame titled "Reports." The frame consists of a number of option buttons and a command button. The option buttons allow the user to select one of the reports described below. When the user clicks on the command button, the selected report is opened.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the MRP plans for a product using the following lot-sizing rules:
 - a. EOQ method.
 - b. POQ method.
 - c. L4L method.
 - d. Silver-Meal heuristic.
 - e. Least Unit Cost heuristic.
 - f. Part period balancing method.

Reference

Nahmias, S., "Production and Operations Analysis with Student CD-Rom." 4th Ed., *McGraw-Hill/Irwin*, 2000.

Problem Description

The aim of this project is to build a decision support system for a university library. The library carries a large number of books, journals, conference proceedings, reference books, and copies of some recorded lectures on CD. These items are loaned to members. Members of the library are mainly students, faculty, visiting scholars, and staff.

The manager of the library has a few concerns. The manager is concerned that a number of items are returned late. The manager would like to send out notices to members in the case that they have not returned the item on time. The manager is also interested in building an automated system that would facilitate the process of searching the database for books written by a particular author, books published by a certain publisher, books on a particular subject, etc.

When building the system, considers the following: the library has items that are to be used inside the library only, such as reference books; the maximum length of borrowing time depends on the item type (CDs can be borrowed for a maximum of three hours, journals for at most 2 days, and books for at most two months); the number of items borrowed at a time depends on the employment status of the member. Faculty and staff can borrow up to 5 items at a time; students can only borrow 2 items at a time. Each item has a unique access code. This code allows the librarian to identify the type of the item (book, journal, CD, etc.), the title, the author(s), and whether the item is on loan, overdue, or on shelf.

Excel Spreadsheets

1. Build a spreadsheet that presents the following data about books: identification number, title, author, publisher, publication year, and edition.
2. Build a spreadsheet that presents the following data about reference books: identification number, title, publisher, publication year, and edition.
3. Build a spreadsheet that presents the following data about journals: identification number, title, volume, number, and publication year.
4. Build a spreadsheet that presents the following data about CDs: identification number, title, course number, professor name, and date.
5. Build a spreadsheet that presents the following data about conference proceedings: identification number, title, volume, number, and publication year.
6. Build a spreadsheet that keeps the following data about members: identification number, name, address, and status (student, staff, etc.).
7. Build a spreadsheet that keeps historical data about the items borrowed by members, such as the borrowing date, the identification number of the item borrowed, and the due date.

User Interface

1. Build a welcome form.
2. Build a data search form that includes six command buttons. When the user clicks on the first command button, titled "Item Search," Form 3 opens. When the user clicks on

the second command button, titled “Member Search,” Form 4 opens. When the user clicks on the third command button, titled “Lend an Item,” Form 5 opens. When the user clicks on the fourth command button, titled “Members’ Information,” Form 6 opens. When the user clicks on the fifth command button, titled “Items’ Information,” Form 7 opens. When the user clicks on the sixth command button, titled “Other Operations,” Form 8 opens.

3. Build a form titled “Item Search” that consists of a frame and a command button. The frame, named “Select a Search Criteria,” includes a number of option buttons. The option buttons allow the user to select one of the following criteria when searching for an item: identification number, author name, title, publisher, publication year, subject area, etc. Upon selection, a text box appears where the user types in a key word. When the user clicks on the command button, the results from the search are presented.
4. Build a form titled “Member Search” that consists of a frame and a command button. The frame, titled “Select a Search Criteria,” includes a number of option buttons. The option buttons allow the user to select one of the following criteria when searching for a member: identification number, name, position, address, etc. Upon selection, a text box appears that allows the user to type in a key word. When the user clicks on the command button, the results from the search are presented.
5. Build a form titled “Lend an Item” that consists of two combo boxes, two text boxes, and a command button. The first combo box, titled “Select an Item,” allows the user to select an item based on the item’s identification number. The second combo box, titled “Select a Member,” allows the user to select a member based on the member’s identification number. The user types in the first text box the borrowing date (the default value is the current date) and in the second the due date. The user clicks on the command button to submit the information.
6. Build a form titled “Member Information” that consists of a frame and a command button. The frame has three option buttons that allow the user to choose whether to add/delete/update the information about a member.
 - a. If the user selected to add a new member, upon selection, a number of text boxes and a command button appear. The user types in the text boxes the information about the new member and clicks on the command button to submit the information.
 - b. If the user chose the delete option button, upon selection, a combo box and a command button appear. The combo box presents the identification numbers and names of the members. The user selects the name of a member from the combo box and clicks on the command button to delete the information about the member from the corresponding spreadsheet.
 - c. If the user chose the update option button, upon selection, a combo box and a command button appear. The combo box presents the identification number and name of the members. The user selects the name of a member from the combo box. Upon selection, another combo box and a text box appear. The combo box presents the names of the columns of Spreadsheet 7. The user selects the name of a column, types the updated information in the text box, and clicks on the command button to submit the updated information. For example, if the user wants to update the address of a member, the user selects the member using the first combo box. The second combo box allows the user to select the address column for this update, and, finally, the new address typed in the text box will replace the old address when the user clicks on the command button.

7. This form, titled “Item Information,” is similar to Form 6. Different from Form 6, this form allows the user to add/delete/update the information about items.
8. Build a form titled “Other Operations” that consists of a frame and a command button. The frame has of a number of command buttons that allow the user to select one of the following options: list the items that are already late; open the reports presented below; etc. When the user clicks on the command button, the results from the search are presented.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report a list with details about the items that are late.
2. Report the top ten most popular books of the year.
3. Report the top ten most popular authors of the year.
4. Report the five most read journals.
5. Create a summary report that presents the total number of journals, books, reference books, conference proceedings, and CDs that are present in the library and how many are lent to members.

Problem Description

McMartin Catalog Distributors distributes a wide range of specialty catalogs. The company accepts orders from customers by telephone using toll-free 1-800-numbers. The company has recently had several successful seasons of catalog sales. This increase in the business volume raised the question whether the company would be able to support the marketing and sales activities using its own telemarketing centers. The alternative is to buy call processing capacity from other call centers. However, managers believe that some of the call centers have excess capacity; therefore, a reassignment of clients to call centers will help to smooth the flow of activities.

The aim of this project is to build a decision support system that will allow the managers to decide about how to reallocate customers to call centers and, most importantly, identify whether the company can manage the increased volume of activities. Below we present an optimization model that can be used to assign clients to call centers.

The Optimization Model

The site selection problem can be modeled as a mixed-integer programming problem. Below we present the problem formulation. The following notation is used:

i	denotes the location of customers
j	denotes potential telemarketing center location
$C(i,j)$	cost per unit traffic from i to j
$F(j)$	fixed cost to open center j
$T(i)$	total usage hours originating at i
$U(j)$	upper limit on size of center j
$L(j)$	lower limit on size of center j
$MAXSITE$	upper bound on the number of sites to select
$MINSITE$	lower bound on the number of sites to select.

The following are the decision variables:

$X(i,j)$	fractional traffic originating at i and handled by site j
$y(j)$	takes the value 1 if we locate a center at site j , and 0 otherwise.

$$\min \sum_{j=1}^M F(j)y(j) + \sum_{i=1}^N \sum_{j=1}^M C(i, j)T(i)X(i, j)$$

Subject to:

$$\sum_{i=1}^N X(i, j)T(i) \leq U(j)y(j) \quad \text{for } j = 1, \dots, M,$$

$$\sum_{i=1}^N X(i, j)T(i) \geq L(j)y(j) \quad \text{for } j = 1, \dots, M,$$

$$\sum_{j=1}^M X(i, j) = 1 \quad \text{for } i = 1, \dots, N,$$

$$\sum_{j=1}^M y(j) \leq \text{MAXSITE}$$

$$\sum_{j=1}^M y(j) \geq \text{MINSITE}$$

$$0 \leq X(i, j) \leq 1 \quad \text{for } i = 1, \dots, N; j = 1, \dots, M,$$

$$y(j) \in \{0, 1\} \quad \text{for } j = 1, \dots, M.$$

Excel Spreadsheets

1. This spreadsheet presents the possible call center locations. For each site, we are given the annual fixed cost of operating along with the wage rate for operators. Note that the company is planning to operate two call centers, each handling approximately the same volume of calls.

Spreadsheet 1: Candidate Call Center Locations.

Area Code	Metropolitan Area	Annual Fixed Cost	Wage per hour
204	Selkirk, MB	\$542,950	\$8.77
204	Brandon, MB	\$407,140	\$9.23
514	Saint - Hyacinthe, PQ	\$610,710	\$8.87
514	Saint - Jean - sur -Richelieu, PQ	\$520,370	\$9.13
514	Saint - Jerome, PQ	\$452,310	\$8.95
613	Cornwall, ON	\$429,720	\$8.79
705	Sault Ste. Marie, ON	\$520,370	\$8.64
819	Shawinigan, PQ	\$565,540	\$7.86
819	Drummondville, PQ	\$452,310	\$8.31
902	Sydney, NS	\$678,460	\$7.33

2. This spreadsheet presents the expected number of incoming calls originated from a number of locations. For each location, we are given the area code and region. On average, each call lasts six minutes in length with an extra one minute of operator time required to process the order after disconnecting. Note that the expectations for the center are that employee utilization is about 87%.

Spreadsheet 2: Expected Volume of Incoming Calls from Each Calling Area.

Area Code	Region	Expected Number of Calls
204	MB	78,397
250	BC	112,746
306	SK	70,399
403	AB	193,330
416	ON	289,617
418	PQ	122,353
506	NB	51,854
514	PQ	238,019
519	ON	100,988
604	BC	131,861
613	ON	102,583
705	ON	41,983
709	NF	40,687
807	ON	17,591
819	PQ	141,881
902	NS	74,110
905	ON	191,599
Total Volume 1,999,998		

3. This spreadsheet summarizes the costs of processing a telephone call (per hour) from each customer location to the possible telemarketing sites.

Spreadsheet 3: The Cost of Processing Telephone Calls (per hour).

Area Code	204	514	613	705	819	902
204	\$9.20	\$12.36	\$11.75	\$11.75	\$12.36	\$12.36
250	12.36	12.36	12.36	12.36	12.36	12.36
306	11.75	12.36	12.36	12.36	12.36	12.36
403	12.36	12.36	12.36	12.36	12.36	12.36
416	11.75	11.75	10.93	10.93	11.75	12.36
418	12.36	10.93	11.75	11.75	10.93	12.36
506	12.36	11.75	12.36	12.36	11.75	11.75
514	12.36	7.80	11.75	11.75	10.93	12.36
519	11.75	11.75	10.93	10.93	11.75	12.36
604	12.36	12.36	12.36	12.36	12.36	12.36
613	11.75	11.75	10.64	10.93	11.75	12.36
705	11.75	11.75	10.93	11.30	11.75	12.36
709	12.36	12.36	12.36	12.36	12.36	11.75
807	11.75	11.75	10.93	10.93	11.75	12.36
819	12.36	10.93	11.75	11.75	10.64	12.36
902	12.36	12.36	12.36	12.36	12.36	12.79
905	11.75	11.75	10.93	10.93	11.75	12.36

User Interface

1. Build a welcome form.
2. Build a form that includes the following controls:
 - a. Insert a frame titled "Problem Data." The frame has three option buttons, a combo box, and a command button. The combo box allows the user to select a spreadsheet. The option buttons allow the user to select whether to add/delete/update information in the selected spreadsheet.
 - i. If the user chose to add information in a spreadsheet, a number of text boxes appear (one for each column of the selected spreadsheet) where the user can type in the new information. Depending on the spreadsheet, the number of the text boxes needs changes. The user clicks on the command button to submit the information.
 - ii. If the user chose to delete information from a spreadsheet, a text box appears where the user can type in the area code. When the user clicks on the command button, information about the selected area code is deleted from the selected spreadsheet.
 - iii. If the user chose to update information, a text box and two combo boxes appear. The first combo box allows the user to choose an area code, and the second one allows the user to choose the name of a field in the selected spreadsheet. The user types in the text box the updated information. When the user clicks on the command button, the information on the spreadsheet is updated.

- b. Insert a command button that, when clicked on, uses the optimization model to assign customers to call centers. The user is prompted to choose whether to open the corresponding reports.
- c. Insert a frame titled "Reports." This frame has a number of option buttons and a command button. The option buttons allow the user to open one of the reports presented below. When the user clicks on the command button, the selected report is opened.
- d. Insert a frame titled "Sensitivity Analysis." This frame has a number of option buttons and a command button. The option buttons allow the user to choose one of the following parameters for the purpose of the sensitivity analysis: the expected number of calls, the expected fixed annual cost, the wage per hour, etc. When the user clicks on the command button, the sensitivity analysis is performed and the results are presented.

Design a logo for this project. Insert this logo in the forms created above. Pick a background color and a font color for the forms created. Include the following in the forms created: record navigation command buttons, record operations command buttons, and form operations command buttons as needed.

Reports

1. Report the potential sites to locate the telemarketing centers identified by the optimization model. The report should include the area code and the corresponding annual fixed costs.
2. Report the location of the telemarketing center in the case that the company is to operate with a single call center. Report the corresponding fixed annual costs and operating costs.
3. Report the location of the telemarketing centers in the case that the company is to operate with two call centers. Report the corresponding fixed annual costs and operating costs.
4. Report the additional costs imposed (as compared to the results presented in the third report) if the managers insist on having one of the two call centers located in Ontario (either area code 613 or 705).
5. Report the results of the sensitivity analysis.

Reference

This case study was prepared based on the "McMartin" case study from the INFORMS website.