Problem Set 1

Due: 4:30PM, Friday February 15, 2002

Problem 1 [10%]

Send an e-mail to your tutorial TA that includes the information listed below. Feel free to add anything else you think he/she might be interested in.

- Name
- Athena username
- Course # and Year
- Computer experience, particularly experience with other programming languages
- What you want to learn from 1.00

Problem 2 [20%]

Write an application that asks the user to enter three positive integers as lengths of three line segments, and determine whether they can make a perpendicular triangle. Read inputs using a JOptionPane as shown on pages 69-71 of your text. You don’t need to check data validity in the input; assume the user correctly enters positive integer numbers. [Hint: Suppose a, b, c are three edges of a perpendicular triangle and c is longer than a and b. They must satisfy $a^2 + b^2 = c^2$]

Problem 3: Design of Logan Express Bus Service [70%]

Introduction

In this homework you will implement a simplified version of the model used by the Massachusetts Port Authority to design and plan the Logan Express bus service. The model is called Fracas (Fare and Route Analysis Computer Aided System) and was developed by one of us (Kocur) for general public transit route planning. It suggests optimal route structure, frequency of service and fares to achieve ridership, deficit and user benefit goals.

Your program will:

1. Define a set of parameters to be used in the model
2. Read a small number of inputs from the user to define the analyses to be done
3. Perform a set of intermediate calculations
4. Calculate the optimal service pattern (routes, frequencies, fares)
5. Calculate output performance statistics (ridership, costs, revenues, etc.)
6. Output results

Some general points about the assignment:

1. You do not need to check for data validity in the input or computations. This would be done in a production program, but not in this early homework
2. All your floating point variables should be double.
3. The equations in the assignment are simplified from the real model, so the approximations will be evident at times.

4. You can write the entire assignment in a single main() method. If you have prior programming experience and are familiar with methods (functions), it is better to partition the program into several methods.

Logan Express bus service operates from parking lots about 20 miles from Logan Airport, and there are now 3 routes (south, west and north of Boston). This analysis was done to establish their number, and the frequency of service and fares.

**Assignment**

1. **Parameters defined at start of program.**

Define the following variables at the beginning of your main program. All are constants except x, b and y, which will be read as input variables (see next section). You don’t need to declare the constants ‘final’. Typical values are given for x, b and y as examples.

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Value</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Boston</td>
<td>Analysis title</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>2</td>
<td>Avg trip duration</td>
<td>Days</td>
</tr>
<tr>
<td>L1</td>
<td>20</td>
<td>Inner bound of service area</td>
<td>Miles</td>
</tr>
<tr>
<td>L2</td>
<td>40</td>
<td>Outer bound of service area</td>
<td>Miles</td>
</tr>
<tr>
<td>W</td>
<td>81</td>
<td>Circumference of area at outer bound</td>
<td>Miles</td>
</tr>
<tr>
<td>p</td>
<td>0.021</td>
<td>Trip density</td>
<td>Trips/mi²/day</td>
</tr>
<tr>
<td>j</td>
<td>0.25</td>
<td>Auto access (local) travel speed</td>
<td>Miles/minute</td>
</tr>
<tr>
<td>b</td>
<td>(0)</td>
<td>Change in bus travel time</td>
<td>Minutes</td>
</tr>
<tr>
<td>k</td>
<td>0.4</td>
<td>Wait/headway ratio</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>130</td>
<td>Bus operating cost</td>
<td>Cents/minute</td>
</tr>
<tr>
<td>T</td>
<td>1050</td>
<td>Length of day</td>
<td>Minutes</td>
</tr>
<tr>
<td>x</td>
<td>(0)</td>
<td>Change in airport parking cost</td>
<td>Cents</td>
</tr>
<tr>
<td>s</td>
<td>47</td>
<td>Bus capacity</td>
<td>Passengers</td>
</tr>
<tr>
<td>v</td>
<td>0.5</td>
<td>Bus speed</td>
<td>Miles/minute</td>
</tr>
<tr>
<td>m</td>
<td>0.8</td>
<td>Fraction of trips in peak direction</td>
<td></td>
</tr>
<tr>
<td>a₀</td>
<td>0.68</td>
<td>Base bus market share if equal service as auto</td>
<td></td>
</tr>
</tbody>
</table>
a2 0.0125  Bus wait time coefficient
a3 0.005   Bus travel time coefficient
a4 0.0001  Bus fare coefficient
a5 0.01    Auto access time coefficient
y (2) Objective weight

A schematic of the service is:

The definitions of the variables are:

Analysis title. Name of the area being analyzed. This is a String.

Trip duration. Average number of days automobile is parked at the parking lot.

Inner bound of service area. Innermost point from which passengers use Logan Express; it is closer to Logan than the parking lot. (See schematic, variable L1)

Outer bound of service area. Point at which passenger trips to Logan drop to very low level; density approaches zero. (See schematic, variable L2)

Circumference of area at outer bound. Defines sector width (See schematic, W).
**Trip density.** This is the number of trips per square mile to and from Logan. This is actually a parameter to a distribution that has a high density near the airport and decreases as the distance to Logan increases.

**Auto access speed.** Travel speed by automobile for passengers driving from their homes to the Logan Express parking lot.

**Change in bus travel time.** If Logan Express buses use a reserved lane or get preferential treatment, this will lower their travel time. Faster bus times are entered as a positive number.

**Wait/headway ratio.** Average passenger wait time as fraction of bus headway (minutes between buses). Passengers know the schedule, so this parameter is usually less than 0.5.

**Bus operating cost.** Cost to Massport to operate a bus, in cents/minute, including labor and all operating expenses.

**Length of day.** Length of operating day for Logan Express, typically 5am to 11pm. In minutes.

**Change in airport parking.** If parking cost at Logan Airport itself is raised, this will increase ridership on Logan Express. Increases in parking costs are entered as positive numbers.

**Bus capacity.** Maximum number of passengers that can be accommodated on a bus. Typically the number of seats on the bus.

**Bus speed.** Average speed of bus from Logan Express parking lot to Logan Airport in miles/minute.

**Fraction in peak direction.** During peak periods, flows are imbalanced on buses. Peak flow is inbound to Logan in morning and outbound in afternoon. Used to compute bus capacity needed.

**Base bus market share.** If bus and auto service are equal (in some sense), this gives the bus market share. Because autos don’t have wait time and people don’t perceive auto operating cost in the same way as a bus fare, some statistical analyses are done to set this. The actual bus market share is computed from an equation; see below.

**Wait time coefficient.** All the coefficients are used in the market share model. As passenger wait time decreases, bus market share increases. This linear coefficient, estimated statistically, predicts the effect. See the market share equation below.

**Bus travel time coefficient.** As bus time decreases, this coefficient predicts the increase in bus market share.

**Bus fare coefficient.** As bus fare changes, up or down, this coefficient predicts the decrease or increase in bus ridership.
**Auto access time coefficient.** As a passenger must travel farther out of the way to the Logan Express parking lot, this coefficient predicts the decrease in market share by distance from the parking lot. This uses the perpendicular distance traveled (see “a” in schematic—this is ‘wasted’ distance traveled). The auto access time is in minutes; the average access time is computed from geometry and the coefficient is applied to that time.

**Objective weight.** Used to trade off deficit goals versus ridership goals. As objective weight increases, ridership and deficit decrease. As it decreases, ridership and deficit increase. This must be strictly greater than 1 and can increase without limit.

2. **Read inputs**

Read x, b using a JOptionPane as shown on pages 69-71 of your text. You don’t need to check the x and b inputs for validity.

Then read a minimum and maximum y to trace the tradeoff between ridership and deficit. Good values to use for testing are 1.1 and 2.1. Assume that y is incremented 0.1 each time the analysis is run; you’ll need a variable to store the increment, of course. No need to check validity of the inputs.

3. **Intermediate calculations.**

After defining and initializing the constants and the input values, perform the following calculations. You don’t need to derive or study these calculations (we provide a reference to a paper at the end if you’re interested). Note that A0 is different than a0

\[
a1 = a0 + a4x/2 + a3b \quad \text{[Revised bus base market share due to time, parking]}
\]

\[
v = \frac{(L1+5)}{(L1+5-bv)} \quad \text{[Revised bus speed due to change in bus time]}
\]

\[
A0 = \frac{(4 L2(L23 - L13) - 3 (L24 - L14))}{(6 L2(L22 - L12) - 4 (L23 - L13))}
\]

\[
L0 = \frac{6L2s}{(pa4m[3 L2(L22 - L12) - 2 (L23 - L13)])}
\]

\[
B0 = \frac{3L2L1 [4 L2(L23 - L13) - 3 (L24 - L14)]}{[3 L2(L22 - L12) - 2 (L23 - L13)]}
\]

\[
C0 = \frac{L2 L1 [3 L2(L22 - L12) - 2 (L23 - L13)]}{[4 L2(L23 - L13) - 3 (L24 - L14)]}
\]

\[
E0 = \frac{ca4a5(2y-1)}{2jk2vpa22a1y}
\]

\[
F0 = \frac{768ca4j2ka2(2y-1)}{(vpa1a52y)}
\]

\[
G0 = \frac{(y-1)}{[(2y-1)a4]}
\]

Example values of these intermediate values are given below as an aid to testing and debugging.
A0= 27.5
L0= 209821.4
B0= 4.125
C0= 0.000008
E0= 1092.4
F0= 3277.3
G0= 3333.3

Note that E0, F0 and G0 are functions of the variable y, which will vary from yMin to yMax that you read as input in step 2 above. You’ll need to begin a loop at this point for the calculation of these three intermediate values and the optimal service patterns in step 4 that depend on E0, F0, G0 and y.

4. Optimal service pattern

With the intermediate calculations complete, you now find the optimal service pattern.

- There are two sets of outputs. The first set of outputs is computed without any capacity constraint on the bus ridership. These analyses can be used to determine the optimal bus size.
- The second set of outputs is computed using the actual bus capacity. If the bus capacity is not a constraint on the operation (if the buses don’t fill to capacity), then these outputs will be the same as the first set.

The outputs are:

Headway (h). This is the optimal time between buses, in minutes, on each route.

Route spacing (g). This is the optimal spacing between routes, in radians.

Number of routes (n). This is the number of routes, simply derived by taking the angle of the service area in radians (see schematic above) and dividing by the route spacing. It will not be an even number in this simplified analysis. (A more sophisticated analysis is required to generate an integer number.)

Bus fare (f). This is the optimal fare. (It would be rounded to a more even number in practice; we don’t bother.)

Bus load factor (q). This is the fraction of bus capacity used in the peak direction; it is between 0.0 and 1.0 and is computed by dividing ridership by bus capacity.

If bus capacity is not a constraint:
h = (3E0B0)1/3

g = (3F0C0)1/3

n = 2W/(g L2)

f = G0 (a1 - a2kh - a5gA0/(4j))

q = gh (a1 - a2kh - a4f - a5gA0/(4j)) / (a4L0)

If q > 1.0, then bus capacity is a constraint, and the optimal service is given by:

hq = h q-1/2

gq = g q-1/2

nq = 2W/(gq L2)

fq = (1/a4) (a1 - a2khq - a5gqA0/(4j)) - L0/ (gqhq)

Recompute the load factor q using the same equation as above. It should be 1.0, but it may have some approximation error.

At this point, the optimal fare, routes and headways have been found, and the bus load factor has been calculated.

5. Performance statistics

You will now calculate the performance statistics, such as ridership, revenue, costs, etc. The measures are:

**Bus market share (Share).** Bus share of total passenger trips from service area to/from Logan.

**Bus ridership (Riders).** Ridership in both directions, per day.

**Bus revenue (Revenue).** Product of bus ridership and bus fare.

**Bus cost (Cost).** Bus operating cost per day.

**Deficit.** Bus cost minus bus revenue.

**Number of buses required (NumBus).** Minimum needed to meet the schedule. Fractional values must be rounded up.

The equations are below: Use the constrained or unconstrained value of service variables, as appropriate:
Share = (a1 - a2kh - a4f - a5gA0/(4j))

Riders = WT\( \text{Share} \) / (L2a4mL0)

Revenue = f (Riders) / 100

Cost = 4W(L1+5)cT / (100ghvL2)

Deficit = Cost - Revenue

\( \text{NumBus} = 100 \) (Cost) / (cT)     Must be rounded up to next integer; use Math.ceil().

6. Output results

Your program should produce output as follows. The numbers are an example only; they are not the actual numbers your program will calculate. Cast the ridership, deficit, headway, fare, and load per bus to int variables so they print compactly. Leave objective weight (y), the number of routes, market share and number of buses as doubles. (Buses should be an integral value.) Use Math.round() creatively to help you line up the columns for the doubles (e.g., round the value times 1000, divided by 1000, to give 3 decimal places). The columns should be:

Title: Boston  <use the Title variable in your program>

Riders  Deficit  Hdwy  Fare  Rtes  Share  Buses  Obj wgt

1000  22000  30  400  4.117  0.241  22.0  1.1

etc.

When the program has produced these outputs, display a JOptionPanel to ask the user if he or she wants to run another analysis. If they say "yes" or "y" (or whatever you tell them to type), prompt the user for x, b, minimum y and maximum y and re-run the analysis. Remember to use the equals() method (see page 65 in your text). When your program is complete, remember to call System.exit(0).

7. Testing

- Check that the results are reasonable; they should be close to the examples given in the homework.
- Run your program varying the b and x parameters. Make sure that the results change in the appropriate direction when you change these parameters. Note that the changes are relatively small.
- Change a few other parameters whose effect you can anticipate, and see if the model responds correctly.

**Turnin**

**Turnin Requirements**

- Problem 1: Email only. No hardcopy required.
- Problems 2 & 3: Hardcopy and electronic copy of ALL source code (all .java files).
- Place a comment with your name, username, section, TA's name, assignment number, and list of people with whom you have discussed the problem set on ALL files you submit.
- DO NOT turn in electronic or hardcopies of compiled byte code (.class files).

**Electronic Turnin**

To electronically turn in your problem sets, run Netscape

Then go to the 1.00 web page at:

[http://command.mit.edu/1.00Spring02](http://command.mit.edu/1.00Spring02)

Click on the "Submit Assignment" button. Be sure to set the Selection Bar to Problem Set 1 or your files may be lost. Finally, go back to the home page and click on the "View" section and be sure that your files were received. If you submit a file twice, the latest version will be graded.

**Penalties**

- Missing Hardcopy: -10% off problem score if missing hardcopy.
- Missing Electronic Copy: -30% off problem score if missing electronic copy.

Late Turnin: -20% off problem score if 1 day late. More than 1 day late = NO CREDIT.