1 Introduction

This note describes how to solve the facility location problem discussed in class. As we’ve seen, it is a simple variant of the Weber problem (and in fact can be transformed into a Weber problem, so strictly speaking, you don’t need this program at all). However, we provide one extension over the Weber problem computation: we allow you to solve the facility location problem using either a Euclidean or a City-Block metric (or both).

As with the Weber problem, we’ll use GAMS (installed in the Knowlton Hall labs) to solve the facility location problem.

2 The Facility Location Problem

Consider an urban area divided (exogenously) into $N$ zones. It is desired to locate some facility in the area, in such a way as to minimize users’ transport costs. We assume that zone $i$ is described by its user population $U_i$. This is the number of people in zone $i$ who use the facility, and may or may not be the actual population of the zone. We do not provide a way to determine the proportion of the population who are users: in the jargon, the $U_i$ are
exogenous. We assume that we may consider zone-$i$ users as concentrated at the point in space $(s_i, t_i)$; this will often be the centroid of zone $i$; but it need not be. Finally, we assume that each user in zone $i$ incurs the same transport cost of $k_i$ per mile.

One way to think about transport costs is that in each zone all travel is at speed $r_i$ (miles per hour); that each zone–$i$ user has a value-of-time of $v_i$ per hour; and that there is a vehicle cost (think of it as as gasoline cost) of $w_i$ per mile. Then the travel time per mile is $1/r_i$, time costs per mile are

$$\frac{v_i}{r_i}$$

and total cost per person-mile is the sum of time cost per mile plus vehicle cost per mile or

$$k_i = \frac{v_i}{r_i} + w_i$$

If all this seems complicated, it is; but note that you can finesse this, if it’s appropriate, by assuming that transport costs are the same for all zones. In that case, the optimal (transport-cost-minimizing) location is independent of the transport cost, just as in the Weber model.

Suppose the facility in question is located at coordinates $(s_f, t_f)$. Let $d(f, i)$ be the distance between zone $i$ and the facility. Then the total transport cost of getting zone-$i$ users to the facility is

$$U_i \times d(f, i) \times k_i$$

and our problem is to choose the facility location to minimize the sum of these costs over all zones $i$, that is, to minimize

$$\text{TTC} = \sum_{i=1}^{N} U_i \cdot d(f, i) \cdot k_i$$

which is our optimand.

We turn now to the question of metric. We consider two. The first is the familiar Euclidean (as-the-crow-flies) metric, written

$$d_E(f, i) = \sqrt{(s_f - s_i)^2 + (t_f - t_i)^2}$$

But given that we are typically concerned with facilities located in a city, it makes sense to consider a metric more geared to the way cities are laid out.
We thus also consider the city-block metric, in which

\[ d_C(f, i) = |s_f - s_i| + |t_f - t_i| \]

There is one small complication. The absolute-value function is not continuous (its graph is a V, with the point of the V at the origin; and you can see that there are many possible tangents at that point), and the GAMS routine we shall be using requires continuity. But we can easily finesse this by forcing each leg of the city-block distance to be positive, regardless of the order of the coordinates: we just square each part, and take the square root. Thus the form of the city-block metric we shall actually be using in the GAMS computations is

\[ d_C(f, i) = \sqrt{(s_f - s_i)^2} + \sqrt{(t_f - t_i)^2} \]

3 Run the model

These instructions will work on the KSOA network, where the GAMS system is already installed. You can download a free student-edition of GAMS [here](#). Though limited, it is more than capable of solving any of these problems on your own computer (as long as you’re running MS Windows or Unix/Linux). If you anticipate having to work with optimization models, it may be well worth learning the GAMS language: it is a very simple way of formulating large-scale (eg multi-region) models.

GAMS is a command-line program, so the first thing to do is to start a DOS session: click Start -> Programs -> Accessories -> Command Prompt. Switch to the directory containing your files.

You will also need the model setup file for the Weber problem, available as facility.zip. Unzip the files to some directory on the hard drive.

At this point you have two ways to run the model.

- Type

  ```
gams facility
  ```
You will see some on-screen output from GAMS, and when that's over you'll have a file *facility.lst* on your hard drive. The end of this file contains the results; it is a plain-text file, so you can view it using a text editor like NotePad: type `Notepad facility.lst` to load it in.

- GAMS produces a lot of output, most of which will probably not interest you, so I've provided a way to run the model and get “cleaner” output. Type
  
  `facrun`
  
  You'll see GAMS run as before. But this time when the run finishes you'll also have a file *facility.res* which contains only the useful output from the end of *facility.lst*. It is also a text file, so you can view it in a text editor like NotePad: type `Notepad facility.res` to do this. (This second batch file assumes that my program *gamsrd.exe*, which is included in *facility.zip*, is in the same folder as the *gms* file. If you want to know how to use this utility in your own GAMS-related work, see me).

Note that in either case, re-running the model will over-write the previous results files without warning, so if you need to keep them you should re-name *facility.lst* and/or *facility.res* before a second run.

### 4 Customizing the model

Here is how you can try out your own assumptions in this model. You do this by editing the beginning of *facility.gms*.

As distributed, the program runs *twice* for each set of assumptions: once using the Euclidean metric and once using the city-block metric. You can turn either of these off as follows:

- Open *facility.gms* and find the lines
  
  ```
  scalar euclid switch for euclidean metric /1/;
  scalar cblock switch for city block metric /1/;
  ```


To turn either of these off, change the /1/ to /0/. Be sure not to erase the trailing “/” or the semi-colon at the end of each line.

Otherwise the principal customization is to set up the characteristics of the urban area. You need to do two things. First, you must tell the model how many zones you will be working with. You do this in the statement

```
SET points /1*3/;
```

where you must change the 3 to be the total number of zones. So, for example, if you want to work with 10 zones, you’d change this to

```
SET points /1*10/;
```

Remember to retain the /; after your number, or else GAMS will complain.

Next you need to supply the data connected with each zone. You do this by adding rows to the GAMS Table called DATA. Each row begins with an identifying number representing the zone number; note that in contrast to the setup for the Weber problem, zone numbers begin with 1 and not with 0 (there is no concept of a “market” in this model). Zone numbers must be sequential, beginning with 1. For each zone, you enter four columns worth of data. The data are:

- **scoord**: the horizontal (x) coordinate of the zone’s centroid
- **tcoord**: the vertical (y) coordinate of the zone’s centroid
- **users**: the number of users in the zone
- **kcost**: the unit transport cost per person-mile.

**Notes on data entry**

- It is very important that you try to keep the data lined up under the column labels — GAMS can be finicky about this.
• You must not erase the semi-colon on the line following the last row of data — that’s how GAMS knows that the Table has ended.

• The comment character in GAMS is the asterisk (*): anything on a line beginning with an asterisk is ignored, so you can add your own comments as needed.

Note on the presentation of results GAMS has an odd way of displaying its answers: zero values appear as blanks, and columns of data consisting entirely of zeros may not be displayed at all. Suppose for example that you solve the facility location problem with just one metric, and suppose that (say) $s_f = 0$ (that is, the optimal facility location is somewhere along the vertical axis). Since you’ve solved only the one problem, all the optimal $s$-coordinates are zero; and the results will display values for $t_{coord}$ only. There will not even be a blank column labelled $s_{coord}$. The important thing is not to be fazed by this: just remember that if something’s missing (not displayed) then its value is zero.