The algorithm will be treated as a multicommodity traveling salesman problem. The source will be started from the centroid of the parcel (or cell), and calculated along the transportation network using driving and walking with public transportation. All costs of distances will be calculated by 1) actual money spent, i.e. gas, bus fare and 2) time spent, calculated in dollars at 10 dollars an hour. These will be summed and calculated for each time the person goes to another location.

The problem will be set up with a symmetrical travel matrix in the form of storeX-commodityX for each store commodity combination as the columns and rows. The value will be the summed cost of traveling to each combination plus the value of the cost of the item. If traveling to another commodity within the store, then the distance cost is 0 and the cost of travel is only the commodity price. The objective is to minimize the cost of traveling to each commodity type subject to having to visit each commodity type once and only once.

Although, options are available to find the perfect optimal solution, it was estimated that each of the 94,000 parcels would take roughly 10 minutes each to calculate. This is not done in a reasonable timeframe, nor do we have access to supercomputers. In order to find a near optimal solution in less than 5 seconds for each parcel, I used the minimum spanning tree algorithm with perfect matching heuristics. I then used the improvement technique of replacement over 5,000 (large enough?) iterations to calculate the final sum of all travel costs for all selected commodity types and store combinations.

Citation for commonly used practice of using CFCC for speed limits <http://faculty.unlv.edu/jensen/CEE_468/modules/NetworkAnalyst/pdf/ArcGISNetworkAnalystDataPrepTutorial.pdf>

Cost along the network was calculated with:

Time (hours) \* 10 + Distance (Miles) \* .585 (from the pretravel form of OARDC Centernet)

Filling Missing Data –

1. Split into categories of superstore, convenience store, and grocery store
2. For each product and each category, ran Krig interpolation from ArcGIS 10 with “Spherical” parameter
3. Did Get values from Rasters from ArcGIS 10 to take the interpolated data from the raster and load it back into the shapefile of each category. This represents that prices of products are generally similar in specific areas.
4. Calculated the percentage of the presence of product from each of the survey stores in each of the three categories
5. Randomly selected products were selected and removed up to the percentage of the product existing in surveyed stores.

Transportation Cost Multimodal –

1. Digitized the bus route as a separate shapefile and assigned a 15 mph speed.
   1. Could not figure out how to calculate the bus fares, so assuming that buses are free
      1. Potentially in the future could make if bus was used and add 2.50 to the total cost of transportation, but would not be factored into the optimized route
2. Routes that were determined as impassible were deleted from the walking shapefile. All remaining vectors on the walking shapefile were given an attribute of 3 mph.
3. All costs of transportation from all parcel points and stores to all stores was calculated using Network Analyst Origin-Destination Cost Matrix
4. The cost matrix was exported to a csv where it was loaded into a SQLite database using python. Used cvs2sqllite.py
   1. There were two stores on the shapefile that were not in the final price xls. Removed them with verifystorecsv2sqllite.py script

Parcels

1. Provided a multipart shapefile.
2. Converted to single part