Snow Plows in Iowa City

An Application of Graph Theory

Morgan Schiller

4/30/2009

Key Words: Chinese Postman Problem, Eulerian Circuit, Weighted graph, Adjacency matrix, Matching

Table of Contents

Paper outline:

I. Introduction
    a. Project I have chosen
        i. Definition of problem
    b. Why it interested me
        i. Personal Relevance
    c. Importance of Solution
        i. Fuel
            1. Reduce Gas Consumption of large city vehicles
                a. Money
                b. Environmental
        ii. Public
            1. Complaints
            2. Safety
        iii. System
            1. Current system in place
                a. Downtown/Bus routes
                b. Steep sloped areas
                c. Flat areas
            2. Effective systems will be useful for a long time
            3. Systematic→ little to no confusion every time system used
            4. Easier to teach to incoming drivers for the city
II. Math Concepts
    a. Seven Bridges of Konisburg
        i. Initial problem
        ii. Methods of Solving
        iii. Existence of Solutions
        iv. Theorems
    b. Chinese Postman problem
        i. Definition
        ii. Use here
    c. Weighted Graphs
        i. Definition
        ii. Use here
    d. Depth First Searching
        i. How this is done
        ii. Methods of finding "best" search
        iii. Relevance here
            1. One-way streets in the city
    e. How I have chosen to model
        i. Graphing problem
            1. Intersections are verticies
            2. Streets are edges

            3. Goal is to reduce the number of times edges are travelled
            4. Ideally each edge (road) would be travelled exactly twice
                a. Once in each direction
        ii. Graph or Digraph?
            1. Combination of the two
                a. Graph useful for streets going both ways
                b. Can then apply Eulerian paths
                c. Digraph useful for modeling one-way streets
                d. Apply Depth first searching where useful

III. Methods for solving
- a. Determine what will and will not be counted as part of the city
- b. Divide into sections
    - i. Why? (Example)
    - ii. Maps of sections in Appendix
- c. Identify one-way streets
- d. Determine best ways to handle one-way streets
    - i. Split among paths?
    - ii. Use one path?
- e. Find potential Eulerian paths

IV. Evaluation
- a. How “good” is my model?
- b. Comparison to current system
- c. Difficulty of Math?
- d. Complications within this problem

Introduction

I will examine ideal snow plow routes around Iowa City. This problem is based on the traditional problem of the Chinese Postman, but contains several other mathematical concepts which will be defined later. During the course of this project, I will consider the modeling of potential routes of snow plows (or garbage trucks, mailmen, etc) to ensure that each road in Iowa City is attended, but in the least amount of times so that the routes will be efficient. I will also consider what makes such a solution optimal, and how different definitions of optimization will affect a solution. This will become a graph theory problem which will be better defined later. This paper will discuss the importance of the problem for both the city government and public, the math involved in modeling the problem, historic methods for solving similar problems, and my final conclusions and recommendations for the city. Because a potential system for the city would be incredibly complex, and the city does not currently employ such a system, I will not deduce one universal solution, but will instead show how a solution might be accomplished.

This problem caught my interest for a personal reason. In the winter of 2007-2008, Iowa City (and the rest of the state for that matter) received record breaking amounts of snow. The snowfall combined with rain runoff to cause the flood of 2008, which would reach the level of a 500-year flood. Because of this disaster, I have paid very close attention to the weather, its effects, and how we as a society try to control and manipulate these effects to our advantage.

I live in a small no-outlet area in northwest Iowa City. During the winter, my neighborhood is not a first priority to be plowed. This is acceptable when considering the magnitude of travel on other streets when compared to ours. During the winter of ’07-’08, however, our neighborhood remained unplowed for weeks at a time. Eventually the area’s inhabitants travelled over the snow and packed it down until the streets were covered with a 6-8 inch layer of what was, effectively, ice. By the time the plows were able to reach our area, the ice was too thick, and the plow could not get through. Although this may seem to be a temporary problem for select few disgruntled citizens, other consequences followed. Because of the freeze/thaw nature of Iowa’s springs, some of the ice would melt into small cracks in the road. This water would freeze, expand, and cause cracks in the roads. There are now many noticeable cracks and potholes in the roads which did not exist in the fall of 2007. These roads will be fixed at the expense of the city. This money could have been saved if an appropriate system were established, which ensured that all roads were plowed in a timely fashion.

Importance of a Solution

Many reasons exist to promote a plowing system. The first is fuel. Currently the city vehicles (like snowplows, garbage trucks, etc.) run on gasoline. The further/longer each vehicle must drive, the more gasoline it uses. Because of this, it is desirable to drive as little as possible. One constraint for the system, however, is that each road must be travelled. Therefore, an ideal system of routes is one in which each road is travelled once and only once. This would save gasoline, which would save the city money. On a more sustainable note, less driving yields lower carbon monoxide emissions. As carbon monoxide emissions and other greenhouse gasses are a large component contributing to global climate change, lowering them is good for the environment.

Secondly, creating a system could solve many public relation problems. Currently the City of Iowa City website displays a page of frequently asked questions regarding snow plow routes. These questions range from "Why can't you plow my street now?" to "I'm having a party tonight. Can you be sure to plow in front of my house before my guests start arriving?" The nature of these questions implies that the public does not fully understand the current system of snow plow routes. Instead people believe that they can merely request a snow plow to travel their street, and one will become available. Obviously this cannot be the case. If a system were well-defined, it could be publicized and could describe where and approximately when the plows would be at a given time. Then, this type of question would not even need to be posed.

Another issue of public concern is that of safety. Driving on snow or ice covered streets is extremely dangerous due to reduced tire traction. Because of this loss of traction, drivers have difficulty stopping even if they travel slowly. Also, given a slope, reduced traction may prevent cars from moving upward, or may even cause them to slip in the opposite direction (downhill) out of the driver's control. One would think that these conditions would result in decreased traffic. Experience has shown, however, that this danger is not great enough to deter most citizens from travelling to where they "need" to go. Snow plows can help, however, by removing most of the snow. When combined with salt and sand spreading, the streets become far safer. For this reason, snow plowing needs to be done efficiently, and an effective system could contribute greatly to this cause.

A system could also be useful from the perspective of city officials. If completed correctly, an effective system will be useful for a long time. Little changes can be easily made to account for city expansions. A system could allow for efficient teaching to new and inexperienced city drivers. It could also reduce confusion among drivers during snowstorms. They could focus on one assigned route, and would therefore require less communication during an emergency situation.

The current snow plow system is not well-defined. The city maintains that no system is in place, but instead priorities are taken. First, downtown/high-traffic areas and bus routes are plowed. This is useful because more people are affected by traffic concerns on such roads. The second priority consists of roads with steep slopes. These roads will be the most dangerous as cars with low traction will have the least control on steep slopes. Finally, flat secondary roads are plowed. While this system is logical, it is not necessarily the most efficient, especially because each phase of the system may contain eulerian circuits or paths which are not being utilized. If such routes existed that would clear these streets in the least amount of time and the least amount of passes, that system may be ideal for all involved. Iowa City bus routes can be found in Appendix B.

Relevant Mathematical Concepts

Könisburg Bridge Problem

The first concept of importance is the problem of the seven bridges of Könisburg. Könisburg was a city in Prussia, which is now Kaliningrad, Russia. Figure 1 shows the layout of the city at the time the famous problem was posed. The question posed was this: Is it possible to cross each of the bridges exactly once and end in the same place from which you left? If so, what would such a path look like? Leonhard Euler examined this problem and it became the foundation of graph theory. Euler modeled Könisburg as a graph theory problem where land masses were represented by vertices and bridges were represented by edges. Figure 2 shows the graph corresponding to the city of Könisburg. Euler then proved that no path existed to cross each bridge exactly once and end in the starting point. Paths on a graph which traverse each edge exactly once are commonly called eulerian paths. If the starting point and the ending point of such a path are the same vertex, the path is called an eulerian circuit. (Roberts & Tesman, 2005)

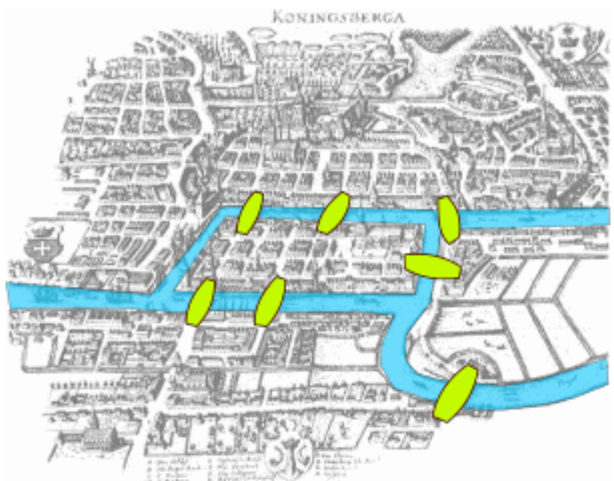


Figure 1. Könisburg, Prussia. (Seven Bridges of Konisburg, 2006)

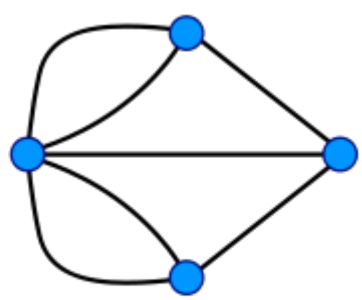


Figure 2. A graph corresponding to Könisburg. Vertices represent
Land masses, and edges represent bridges. (Seven Bridges of Konisburg, 2006)

Euler went on to describe conditions under which an eulerian path or circuit can exist. His theorem was as follows: a multigraph G has an eulerian chain if and only if G is connected up to isolated vertices and the number of vertices of odd degree is either zero or two. (Weisstein, 2009) In 1946, Irving John Good elaborated on this theorem, resulting in two other theorems:

- A multidigraph D contains an eulerian closed path if and only if D is weakly connected up to isolated vertices and for every vertex, indegree equals outdegree.
- A multidigraph D contains an eulerian path if and only if D is weakly connected up to isolated vertices and for all vertices with the possible exception of two, indegree equals outdegree, and for at most two vertices, indegree and outdegree differ by one. (Roberts & Tesman, 2005)

These theorems are very useful, as one can quickly examine a given graph, and determine if an eulerian path or an eulerian closed circuit exist. The question arises of what I will do if a section of the city does not contain an eulerian circuit. This is important because I will likely not encounter the most ideal circumstances. In this case, I intend to try to decompose the graph into multiple subgraphs such that these subgraphs have eulerian circuits. I will then have two options. I can assume one truck will complete each circuit one after another, or I can assume I will send more than one truck to that section. If I choose to send more than one truck to each area, I would prefer to find one eulerian circuit for each truck.

Chinese Postman Problem

The Könisburg bridge problem and eulerian paths have been used in solving another familiar problem: that of the Chinese Postman, also called a route control problem. Discovered in the 1960's by Chinese mathematician, Kwan Mei-Ko, the problem is closely linked to the bridge problem. The postman wishes to deliver mail to each street, but would like to travel the least distance. Put another way, the problem involves finding the shortest circuit (closed route) that touches each edge of the graph and results in the least distance. This is more general than finding an eulerian path, as it makes no distinction about the number of times an edge can be traversed. (Kann, 2000) If a graph contains an eulerian circuit, then that circuit would be the optimal solution to the route control problem as each edge would be touched once, which is the least amount of acceptable times. If an eulerian circuit does not exist, then, as mentioned, some vertices have odd degree. This implies that some of these edges must be visited more than once. The Chinese Postman problem has a solution algorithm. This solution will applied in the **Solving** section of this paper, but first, more mathematical concepts must be introduced. (Chinese Postman Problem)

Weighted Graphs

Because distance is important in the Chinese Postman problem, one must understand the concept of weighted graphs. A weighted graph is a graph with weights assigned to the edges. These weights can represent many different things like distance or cost, which can then signify one edge as "better" than another. In the Chinese postman problem, weights could be used to represent the distance of a street (edge in the graph). The postman must traverse each edge once (this would yield a known constant minimum distance), but, if he must traverse edges again, he would prefer to repeat edges of the shortest length, or weight. Figure 4 depicts an example of a weighted graph with a spanning tree. Trees and spanning trees will be defined later. (Roberts & Tesman, 2005)

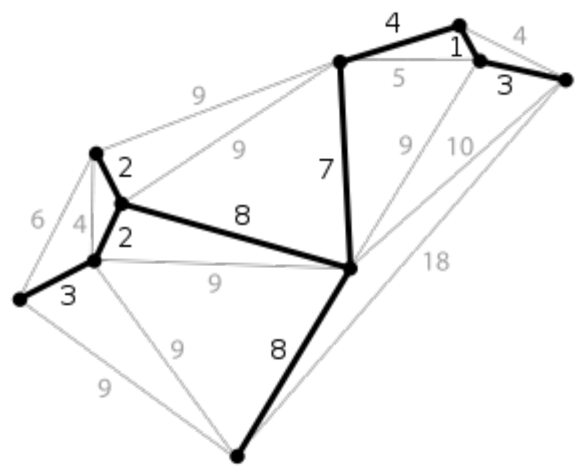


Figure 3. A weighted graph with its minimal spanning tree

Depth First Search Tree

Another useful concept concerning graph theory is that of the depth first search. Before this is introduced, some other words must be defined. First, a tree is a connected graph which has no circuits. An important property of trees is that the number of edges in a tree differs from the number of vertices by one. Given a graph with vertices and edges, a spanning tree is a tree which includes every vertex from the original graph. A spanning tree may or may not contain all the edges from the original graph. One algorithm to find a spanning tree yields what is called the depth-first search tree. This is found by choosing a vertex from the graph, and labeling it. Then an adjacent vertex is labeled (it is labeled such that the second vertex comes after the first) along with the edge joining the two. This is repeated until no unlabelled vertices remain, at which point the algorithm is backtracked until an unlabelled adjacent vertex is found. This algorithm yields a tree with the longest paths. Figure 3 shows an example of a depth-first search spanning tree. (Roberts & Tesman, 2005)

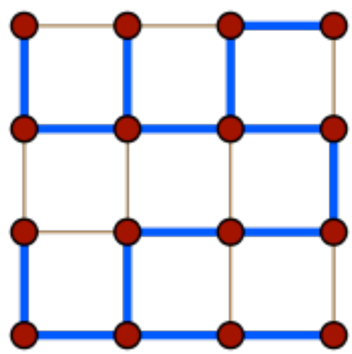


Figure 4. A depth-first search spanning tree.
The dark blue lines indicate the edges included in the tree.

Given a graph, G, and its depth-first search tree, G can become a digraph. First, each edge in G which is also in the tree is given the direction that points from the 'smaller' vertex to the 'larger' based on the depth-first search tree labeling system. Then, each edge in the initial graph which is not in the spanning tree is given a direction which points from the 'larger' to the 'smaller' vertex based on the labeling system. This will yield a digraph which is strongly connected. This means that each vertex can be reached by following the direction arcs from any other vertex. A depth-first search tree can therefore be used to create a system of one-way streets which can be travelled in a similar way to eulerian chains. (Roberts & Tesman, 2005)

Because this graph is strongly connected, any two vertices are connected by some path, which means each pair also has a shortest path. The shortest path between any two vertices can be used to judge which spanning trees are better than others. First, the length of the shortest path could be calculated for all pairs of vertices in the tree. One might wish to minimize the greatest of these lengths. This method would yield a length such that, for any pair of vertices in the tree, the shortest path between them is less than or equal to that length. One disadvantage to this method is that the lengths of all potential shortest paths could be relatively large. Another way to optimize the graph would be to find the average length of the shortest paths and try to minimize that value. A disadvantage to this method is that an average might poorly represent the values involved. A low average value would imply that shortest paths are all relatively short, but the average value can be warped by an outlier, a very long path between some pair of vertices, for example. (Roberts & Tesman, 2005)

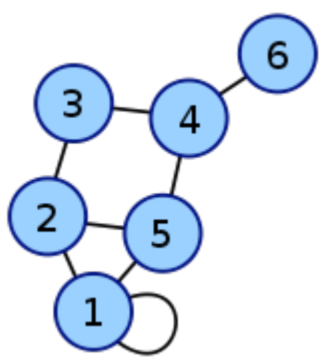
A depth-first search tree is important here because Iowa City has some one-way streets. Upon further investigation, however, it will not be applicable in this situation. Because the scope of my project does not include relabeling or creating more one-way streets in Iowa City, the likelihood of finding a depth-first search tree which coincides with the one-way streets in the city is very slim.

Important Matricies

Two matrices would be useful in applying this model. The first is an incidence matrix. This is an m x n matrix where m is the number of vertices in the multigraph and n is the number of edges. The rows of the matrix represent vertices, and the columns represent edges. The entry, $x_{i,j}$, of the matrix is 1 if vertex i is an endpoint of edge j. This matrix generally requires a lot of storage (at least n^2-n), but can be useful to fully describe the graph. (Roberts & Tesman, 2005)

Another useful matrix is an adjacency matrix. This is a square n x n matrix where n is the number of vertices. Each row and each column represent a vertex. The entry, x_{ij} is 1 if there is an edge from vertex i to vertex j, and zero otherwise. In a graph (containing no loops), the entries of the main diagonal will all be zero. This can be applied to a directed graph, where the entry x_{ij} is 1 if there is an arc from vertex i to vertex j. If G is a graph, then an edge from i to j will also be represented as an edge from j to i, which means the adjacency matrix used to represent G will be symmetric. Figure 5 shows a graph and its corresponding adjacency matrix. (Roberts & Tesman, 2005)

For the purposes of this project, the adjacency matrix will be used to calculate the degree of the vertices. The degree of each vertex is equal to the number of ones in the corresponding row or column. As you can see in Figure 5, this is not true for a graph which contains loops. The degree of vertex 1 in Figure 5 is 4, but the number of ones in the 1^{st} row and column is three. This is because a loop from vertex 1 to vertex 1 will increase the degree of the vertex by 2 (one to exit and another to reenter). On the other hand, the adjacency matrix only contains a one in the entry of the first column and first row. The storage required for an adjacency matrix is n^2. (Roberts & Tesman, 2005)



$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Figure 5: A graph and its corresponding adjacency matrix

Matching

Given a graph containing edges and vertices, a matching of G, M, is a set of edges such that each vertex of G is on at most one edge of M. It is often useful to find a matching with as many edges as possible. The number of edges in M is at most n/2 where n is the number of vertices in G. When applied to a weighted graph, matchings can be compared based on the weights of the edges in M. A perfect matching is the result of all vertices being on exactly one edge in M. (Roberts & Tesman, 2005)

My Model

As this problem involves a map of the city, it will easily be mapped as a graph theory problem. The graph which represents the city, however, can be described in different ways. First,let edges and arcs be defined by streets and vertices defined by every point where streets intersect. Because some streets are one way streets while others are bidirectional, the resulting graph will be a multigraph. Therefore, one-way streets will be modeled by directed arcs, while bi-way streets will be modeled by edges in the multigraph. My goal in this project will be to traverse each edge in the multigraph in the least amount of times. Ideally each edge would be traversed exactly once for each lane on the represented street. This is where eulerian paths will be most useful.

The multigraph will be weighted, but the weights can be used differently depending on the intended use of the model. If the city prefers its current system of plowing steep streets before flat ones, a weighted graph would be appropriate where weights can be used to represent the grade (slope) of a street. During route selection, then, I would choose a path which yielded the highest weights (implying the steepest streets) before a path with lower weights (implying a flatter street). This means I will choose a minimal path before a path of larger weight. Bringing in the element of time complicates things. To remedy this, once a path of maximal grade is found and traversed, it can be virtually removed from the graph, at which point the next path of maximal grade would be found.

If the city can be plowed completely in an acceptable amount of time, or if this model is used to represent a garbage truck or other city route, then the steepness priority will become irrelevant. At this point, the weights of the graph can represent distance, as in the original Chinese postman problem.

One other option would be to simply model each lane of each street as an arc, giving it a direction based on the orientation of the lane. Vertices would still represent intersections of streets.Because no city engineer would create a road going to a point from which there is no escape or one which cannot be reached, there is at least one arc emanating from and leading to each vertex.

The Chinese Postman problem can be solved in polynomial time if all streets are undirected. Because this is not the case in my model, complexity will increase. The Chinese Postman problem where the graph has both one-way and two-way streets is N-P complete.

Solving

Methods

A solution to the snow plow problem for the city of Iowa City would be very difficult to find. This is due to the vast number of vertices and edges, the nonhomogeneity of the street directions, and a lack of time in which to analyze the large system. The most appropriate method of solving this problem would be to examine the math concepts, create the model, and write a computer program to perform computations. Because of this, I will instead explain how one might go about solving the problem.

First, one must determine which streets fall under the responsibility of the city. Then, the city should be divided into sections. Because the city has several trucks (10 to 12), each one (or perhaps several in a group) can be dispatched to a different area. Therefore, each section can be analyzed separately. This will ease the problem solving process, and will not reduce the effectiveness of the final system.

Example: Importance of Dividing the City

To show that it would be more prudent to analyze city sections separately than to have each truck's route span the entire city, we will consider a counterexample (shown in Figures 6 and 7). Let's assume an absurd plowing system in which each truck is first assigned to an east/west streets, and then assigned to a north/south street, and finally the trucks plow the streets of unusual orientation (diagonals and curves). This system would ensure that each street was plowed, but would not be an optimal system. The layout of the city is complex: some streets run through the whole city and have regular intersections, while others end in cul-de-sacs, and still others curve and have very few intersections. Figure 6 shows a portion of the city. Figure 7 depicts the plowing routes of the counterexample. The red streets run east/west and would be plowed first. The blue streets run north/south and would be plowed second. The green streets are of unusual orientation, and would be plowed last. Consider a person driving from point A to point B during the point in the plowing process where only red streets had been plowed. This driver would drive across plowed intersections (would cross the red lines) where he would experience good traction and control over his car. He would mostly be travelling on unplowed roads, however, and may have difficulty stopping before reaching the intersection. If each truck was assigned a route spanning the city as above, they may also have trouble coordinating their efforts, which could mean that one truck would finish its route, while another, completing a parallel route, may have a lot of work/time remaining. To avoid these problems, the city will be divided into sections which will be analyzed separately. This is similar to the methods used by other cities to describe their snow plow routes. (Ann Arbor Street Plow Routes, 2009)

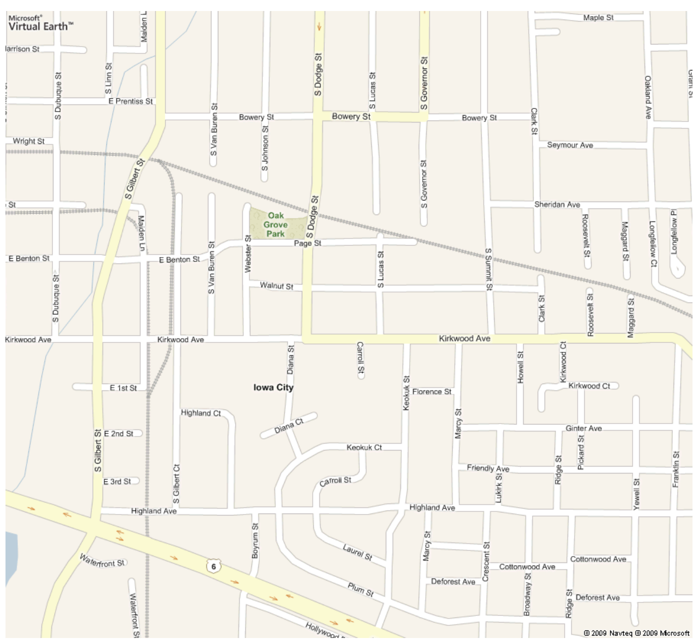


Figure 6. A section of Iowa City

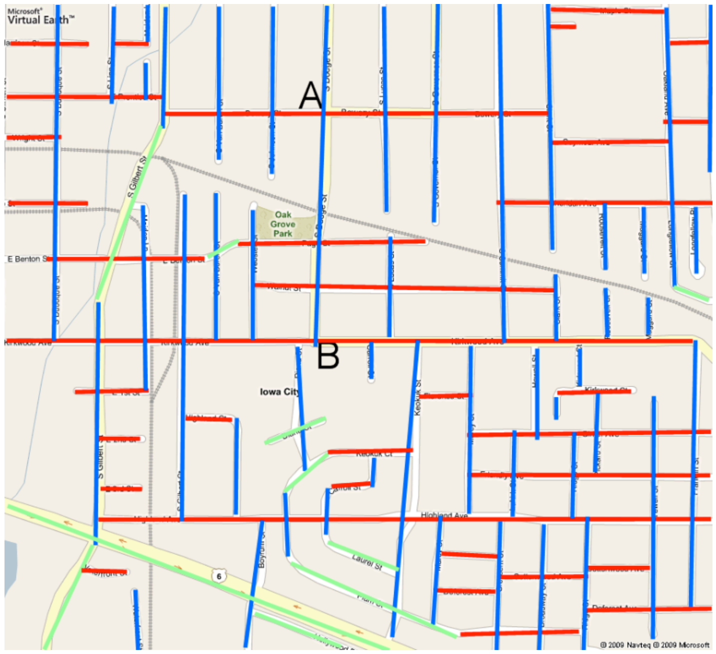


Figure 7. An example of a poorly structured plowing route. Plows are first assigned an east/west oriented street (shown here in red), then a north/south street (shown in blue). Finally, the streets of unusually orientations (shown in green) are plowed.

A map of Iowa City and maps of its sections can be found in Appendix A. The city currently employs 10 to 12 trucks with plowing capabilities.

Finding a Solution

At this point, a decision must be made by the person modeling. He could choose to follow the previous model by taking careful consideration regarding bus routes and busy streets, then slope. The plowing would occur in phases. The first phase would be bus routes. The routes to plow these would be simple, because they are, by nature, a circuit. The second phase would be to plow the steep streets. To accomplish this, the grade of each street would need to be calculated. Then, the streets should be divided into those of high grade and those of low grade. The streets of high grade must fall on the first chronological paths.

Should he choose not to follow the system currently being employed, the modeler would instead let the weights of the graph represent distance, and proceed to minimize the distance weighted path so that each edge was traversed.

For each section of the city, vertices should be labeled (subsequently each vertex will be counted). The edges in the graph should be recorded using an adjacency matrix. This matrix can be used to determine the degrees of each vertex. If each vertex has even degree, then an eulerian circuit exists and is the most appropriate path. If not, then an even number of vertices have odd degree. For each pair of these vertices, there exists a shortest path between them.

The vertices of odd degree will constitute a subgraph, H, of the section. H will be a complete graph, and, because the H contains an even number of vertices, it also has a perfect matching. Let the weight of the edge from vertex v to vertex u represent the distance of the shortest path from v to u. The minimal perfect matching will be used to determine the paths between these vertices. (Chinese Postman Problem)

For each edge in the shortest path from u to v, create a duplicate edge. Then, each vertex in H will have an even number of degrees, and should follow to duplicate path to go from u to v. This will create a graph with the minimum distance being repeated, which will constitute a minimal solution of the graph G. (Chinese Postman Problem)

When applied to each section of the city, and potentially sending more than one truck to a section, this system will allow each street to be plowed at least once, repeating the least distance.

Evaluation of Model

Because several model types and solutions have been suggested, it makes sense that the best model must be considered and selected for each situation. For example, when used for snow removal, it would be wise to look into the grade of the streets. In this case, a weighted graph where weights represent the grades of streets would be useful. A maximum weighted path would be plowed before a minimal path. Also, the lanes should be represented by directed arcs, as each lane must be individually plowed. On the other hand, when used for garbage truck routes, the most appropriate model would be a multigraph where one-way streets are modeled by directed arcs and two-way streets are modeled by edges. In this case, each edge or arc must be traversed exactly twice, once for either side of the street. In any model of the city, the presence of an eulerian circuit simplifies the problem, because it would be a path which traverses each edge in the least amount of times.

Some considerations have been neglected in the creation of this model. As snow plows are large vehicles, they may have trouble making sharp turns. So, although an eulerian circuit may designate a route with many sharp turns, the real situation may require that the plow travels straight for as long as possible.

My model contains no specification regarding whether snow plows are more effective on a slope driving uphill or downhill. If one proves to be easier than the other, the weights of the graph might be altered.

This model would require updates each time road work caused changes in city paths. Because winter is not the most appropriate time for road work, however, I believe this to be a minor problem. I also believe that a program which solves the model for the city could easily be altered to account for any changes.

Conclusions

The city of Iowa City currently has a pseudosystem in place which dictates the plowing of its streets. That system was logically created and is employed each year, but it could be improved. An optimal solution of this problem is important because it would be environmentally friendly, cost efficient for the city, safe and logical for the public, and simpler for the drivers. Graph theory can be used to model the city and some important mathematical concepts can help to find the optimal solution. After reviewing the concepts of eulerian paths, the Chinese Postman Problem, adjacency and incidence matrices, weighted graphs, and depth-first search trees, I found that many of the concepts can be directly applied and can be written into a computer program to solve the problem for the city.

I would recommend that city officials hire an expert to divide Iowa City into sections to be modeled by the aforementioned computer program. This person should maintain some elements of the current pseudosystem, in that the plowing should be completed in the following phases: bus routes and downtown streets, steep streets, and finally, flat secondary streets. Each phase should be evaluated separately so that an optimal route can be found. This will yield the safest and most appropriate snow plow routes.

Bibliography

Ann Arbor Street Plow Routes. (2009). Retrieved March 2009, from City of Ann Arbor: http://www.a2gov.org/government/publicservices/fieldoperations/Pages/StreetSnowPlowingStatus.aspx

Bus Schedules. (2009). Retrieved March 2009, from City of Iowa City: http://www.icgov.org/default/?id=1519

Chinese Postman Problem. (n.d.). Retrieved April 2008, from http://eie507.eie.polyu.edu.hk/ss-submission/B7a/

Iowa City, IA. (2009). Retrieved April 2009, from Google Maps: http://maps.google.com/

Iowa City, IA. (2009). Retrieved March 2009, from Maps.Live: http://maps.live.com/#JnE9eXAuSW93YStDaXR5JTdlc3N0LjAlN2VwZy4xJmJiPTU3LjMyNjUyMTIyNTIxNzElN2UtNjkuODczMDQ2ODc1JTdlMjEuODYxNDk4NzM0MzcyNSU3ZS0xMTMuNDY2Nzk2ODc1

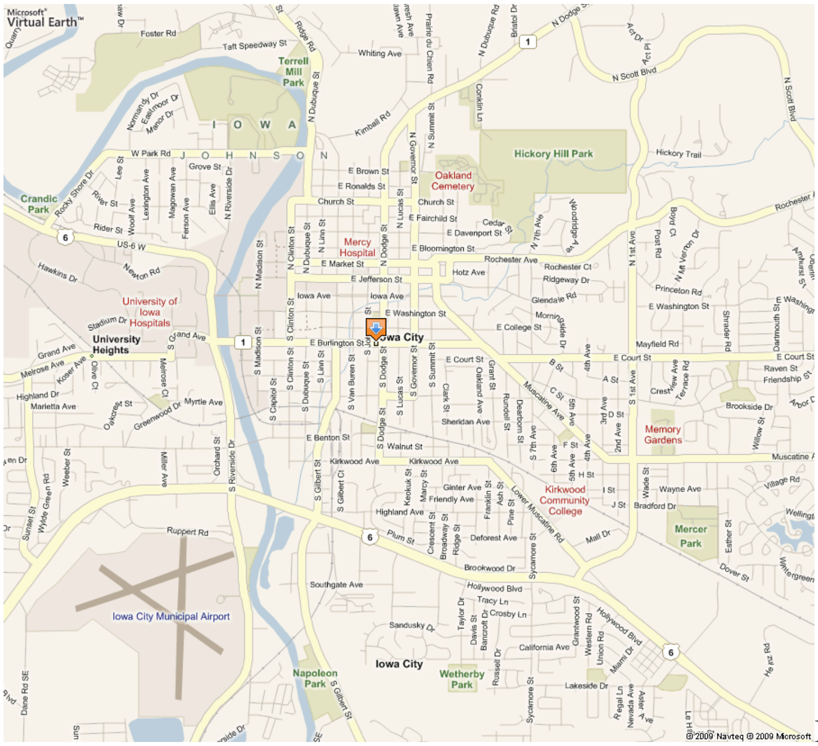
Kann, V. (2000, March 20). *Minimum K-Chinese Postman Problem.* Retrieved March 2009, from http://www.csc.kth.se/~viggo/wwwcompendium/node111.html

Roberts, F. S., & Tesman, B. (2005). *Applied Combinatorics.* Upper Saddle River, NJ: Pearson Education Inc.

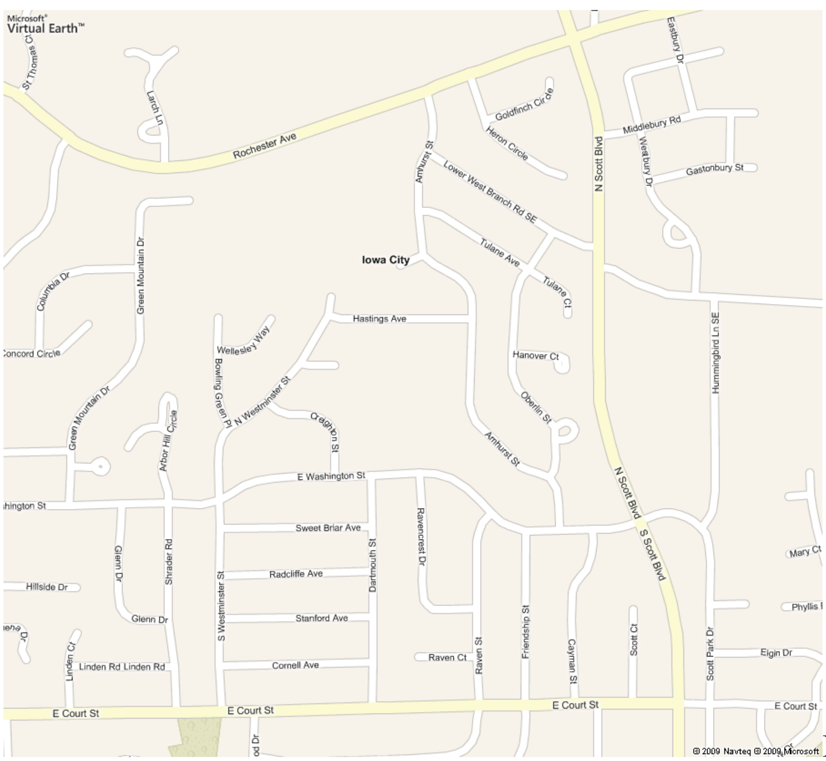
Seven Bridges of Konisburg. (2006, July). Retrieved April 2009, from Wikipedia: http://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg

Weisstein, E. W. (2009). *Eulerian Path.* Retrieved April 2009, from MathWorld: http://mathworld.wolfram.com/EulerianPath.html

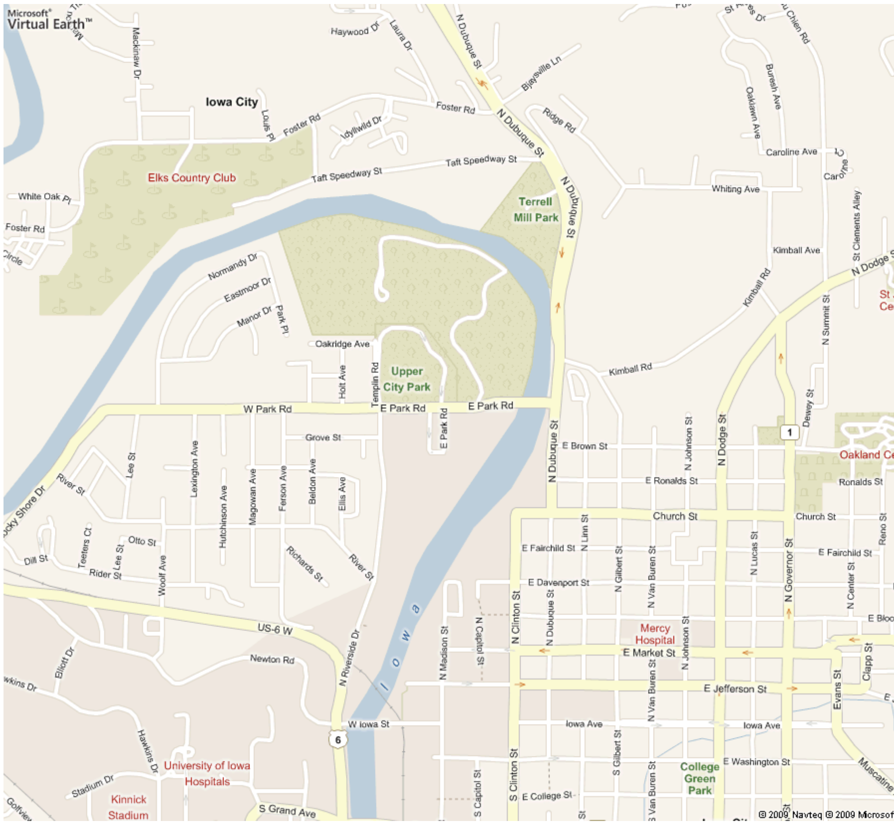
Appendix A: City Sections (Iowa City, IA, 2009)



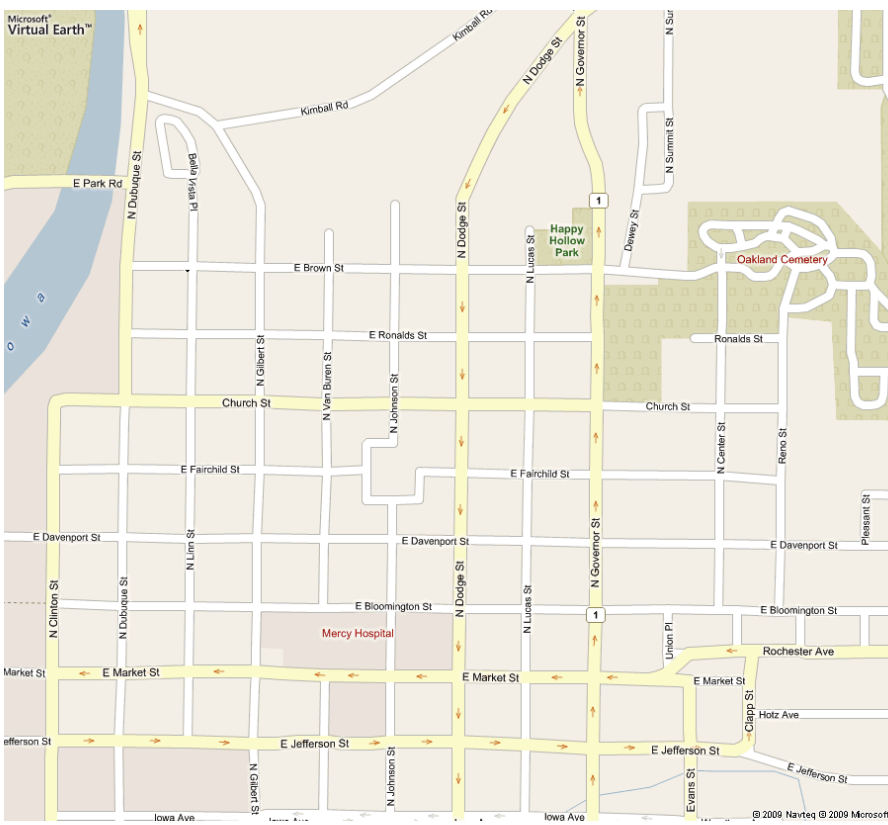
IowaCity



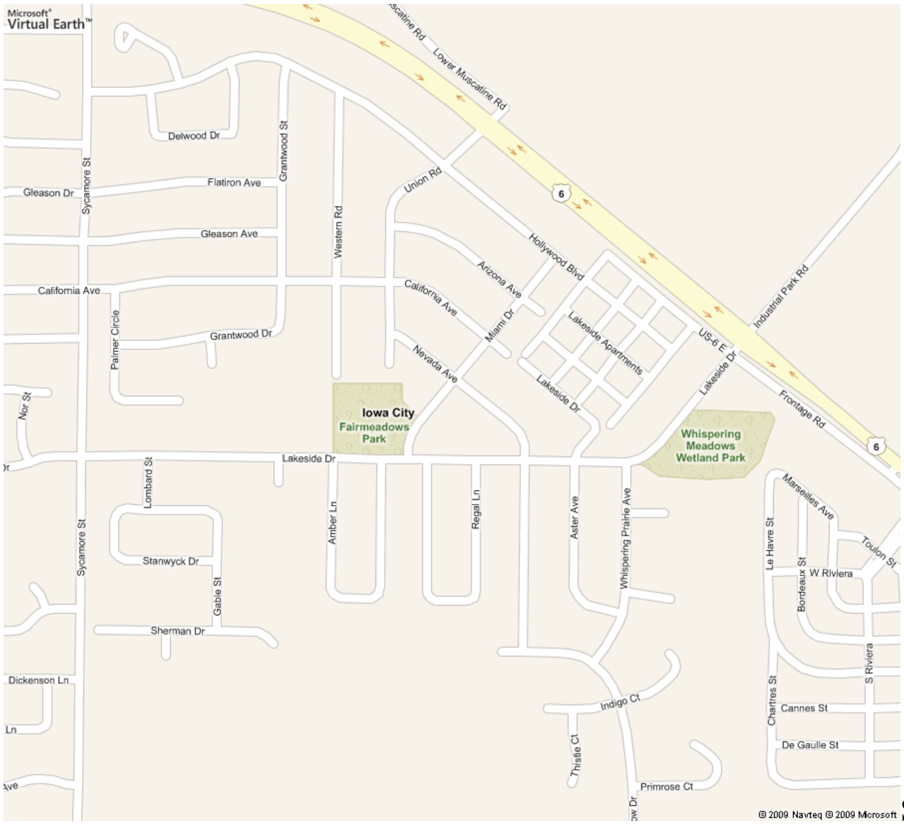
East Iowa City



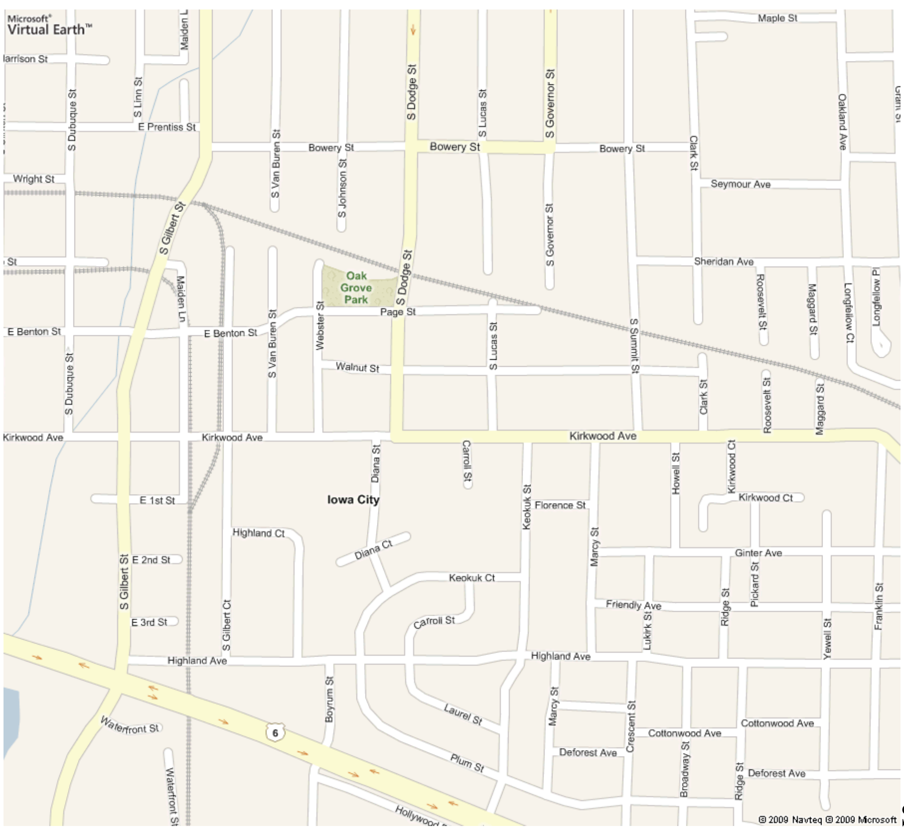
ManvilleHeights



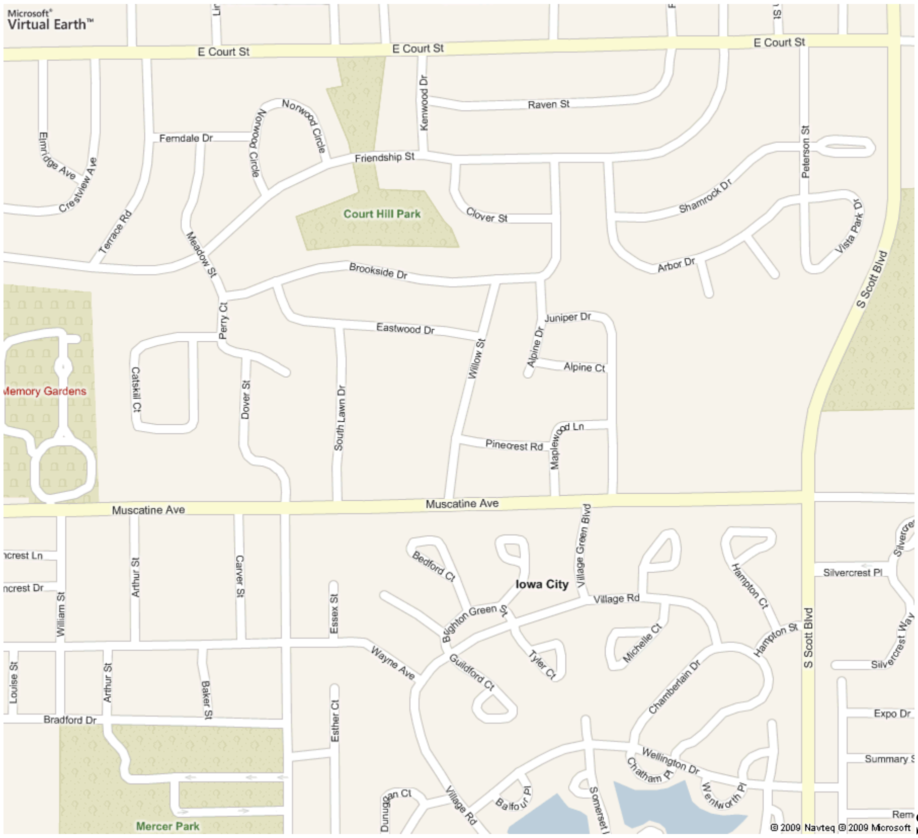
NorthDodge



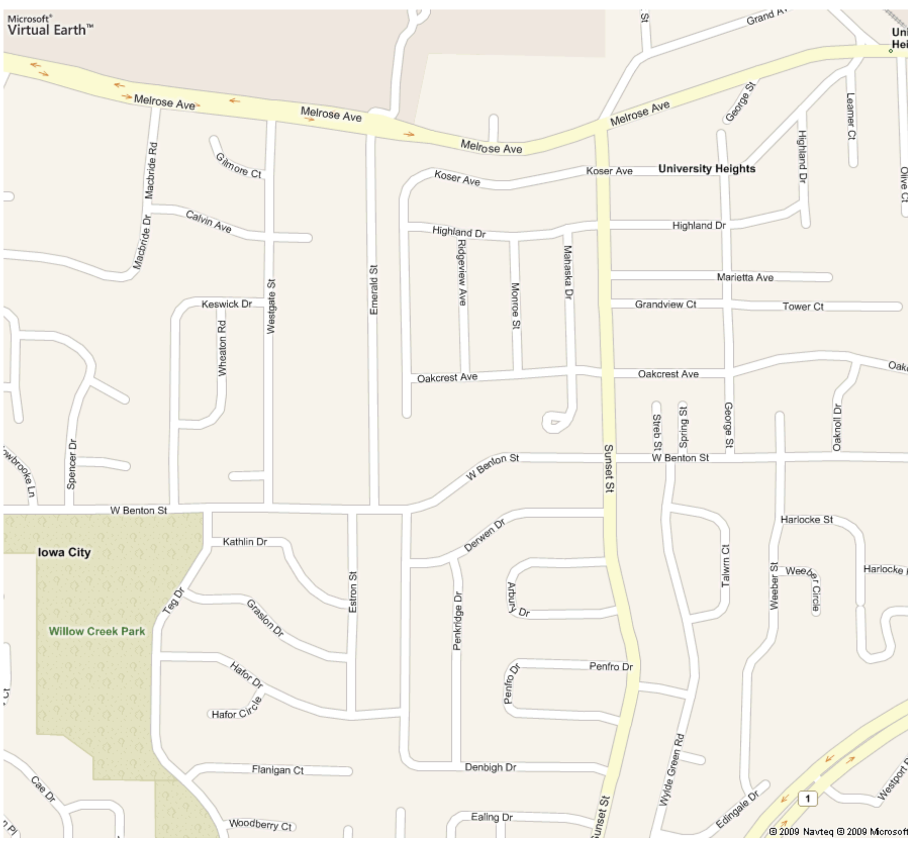
South IowaCity



South Campus



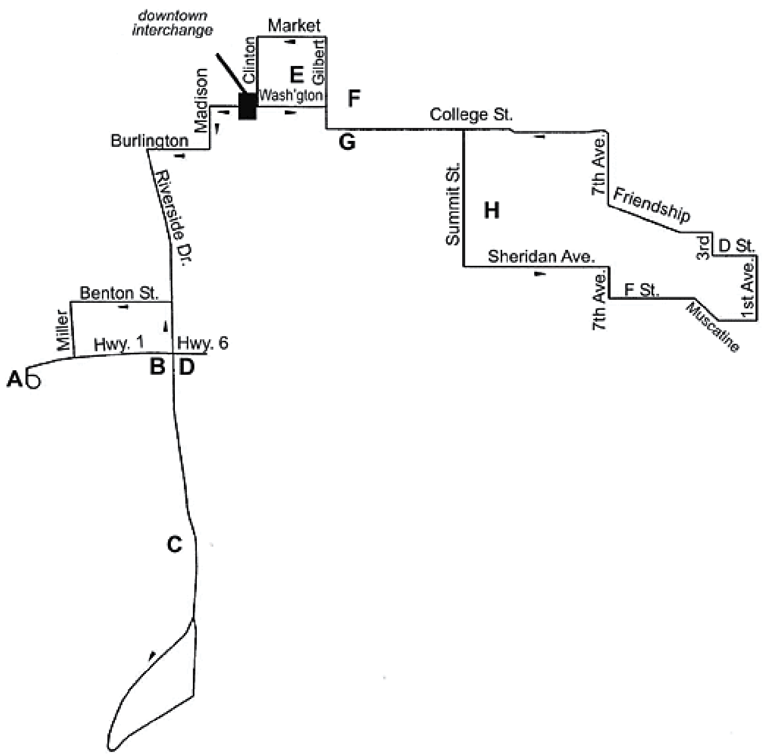
Southeast IowaCity

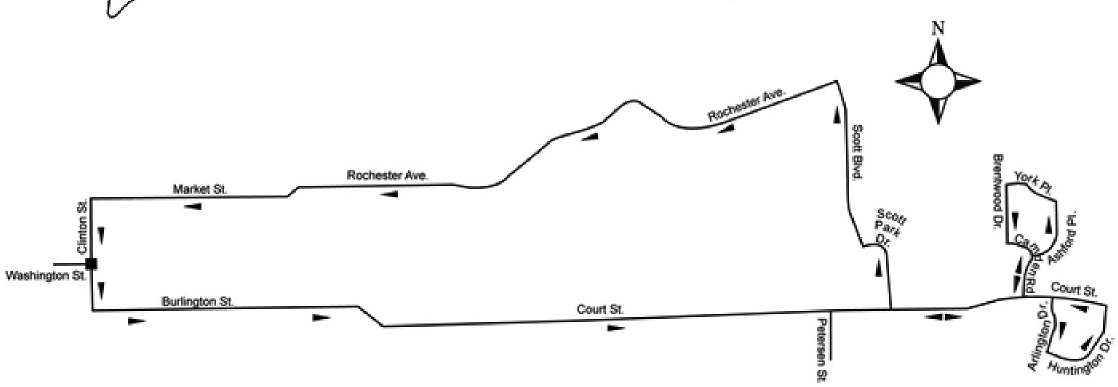


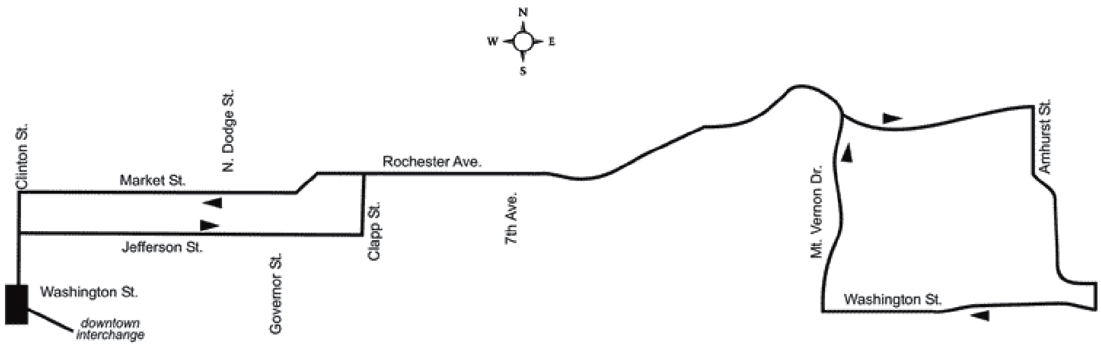
University Heights

Appendix B: City Bus Routes (Bus Schedules, 2009)

East Routes

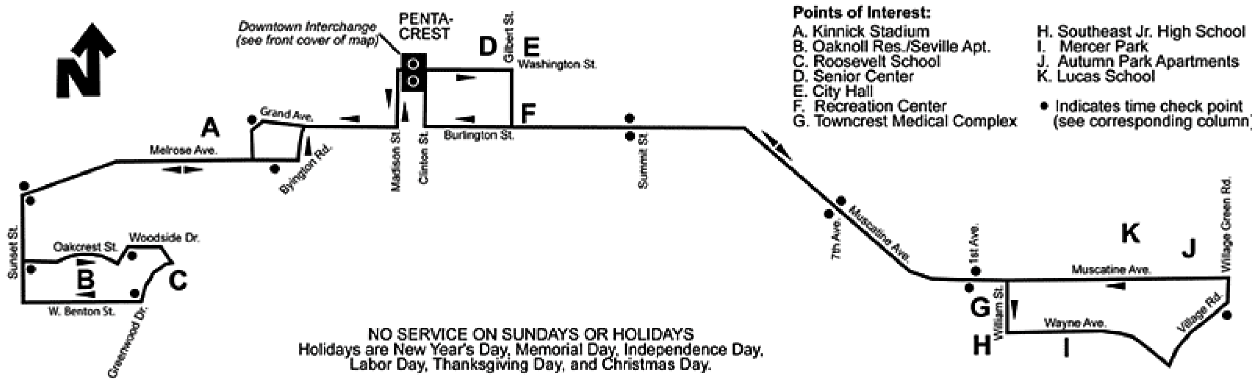




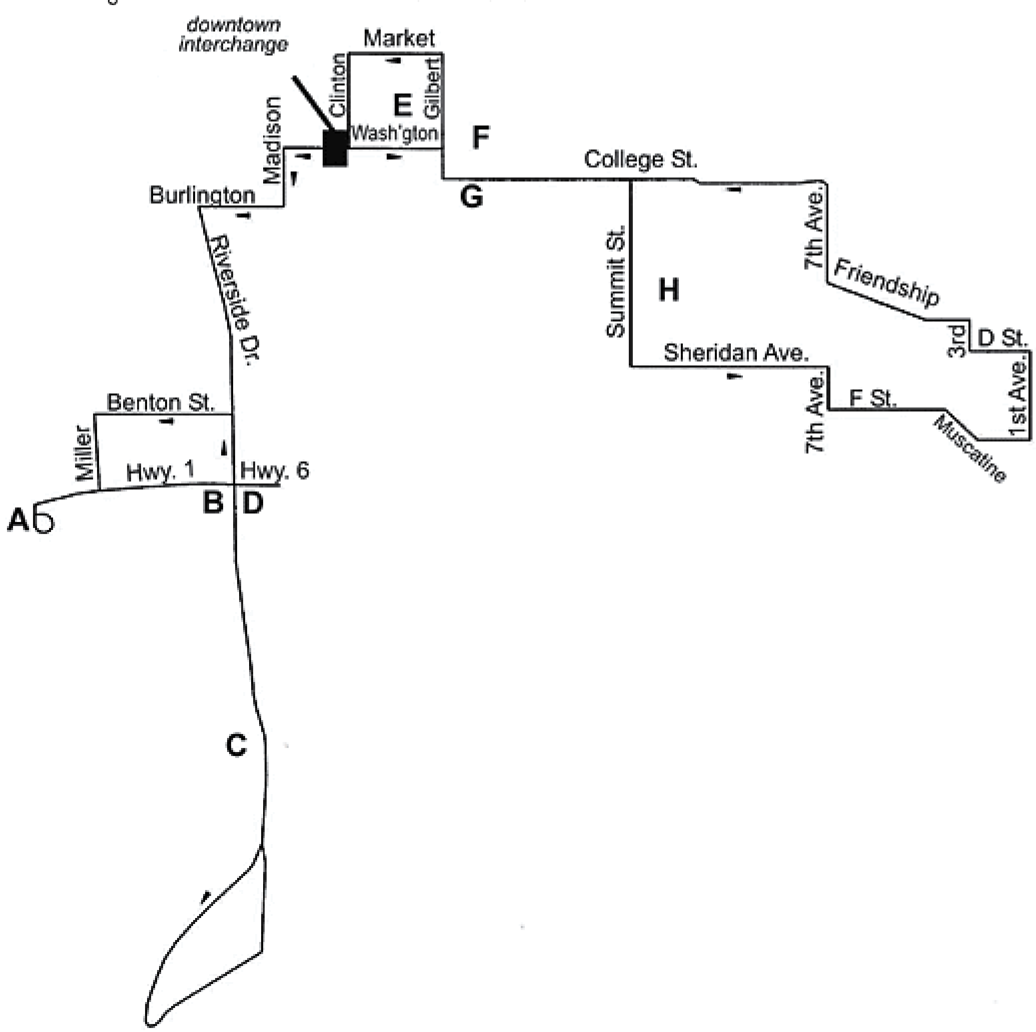

N
W
E
S
N. Dodge St.
Clinton St.
Market St.
Rochester Ave.
Jefferson St.
Governor St.
Clapp St.
7th Ave.
Washington St.
downtown interchange
Mt. Vernon Dr.
Amhurst St.
Washington St.

West Routes

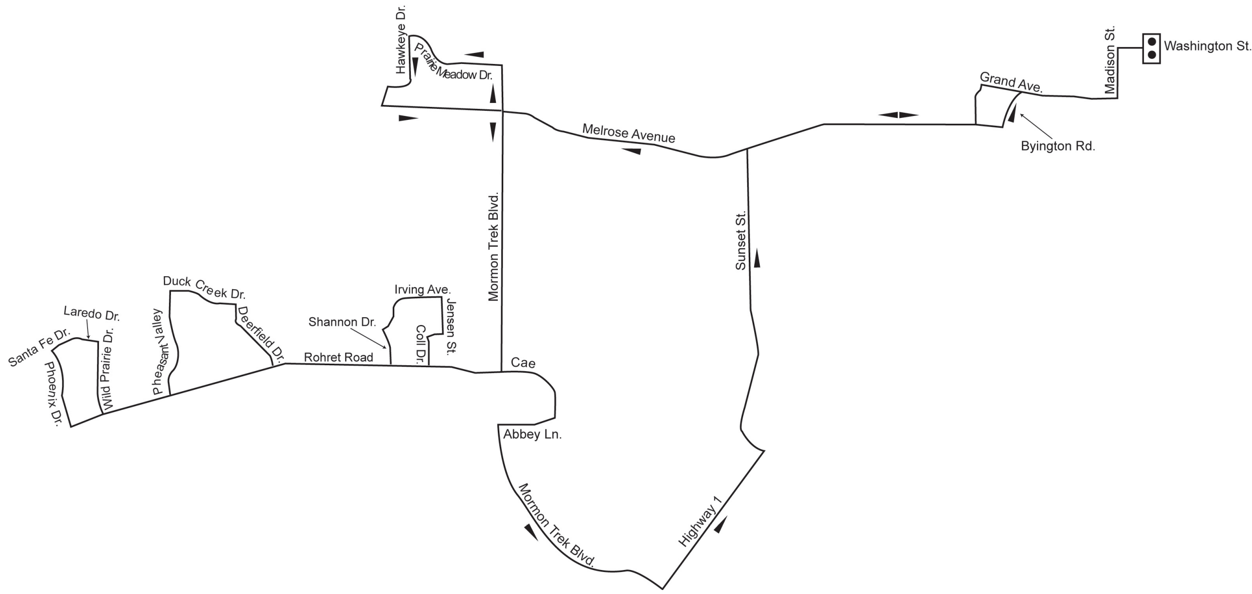
Newton Rd.
Elliot
Iowa Ave.
Washington
Clinton
Burlington
N
Hawkins Dr.
Melrose Ave.
Melrose Ave.
Melrose Ave.
Byington Rd.
Chatham Oaks



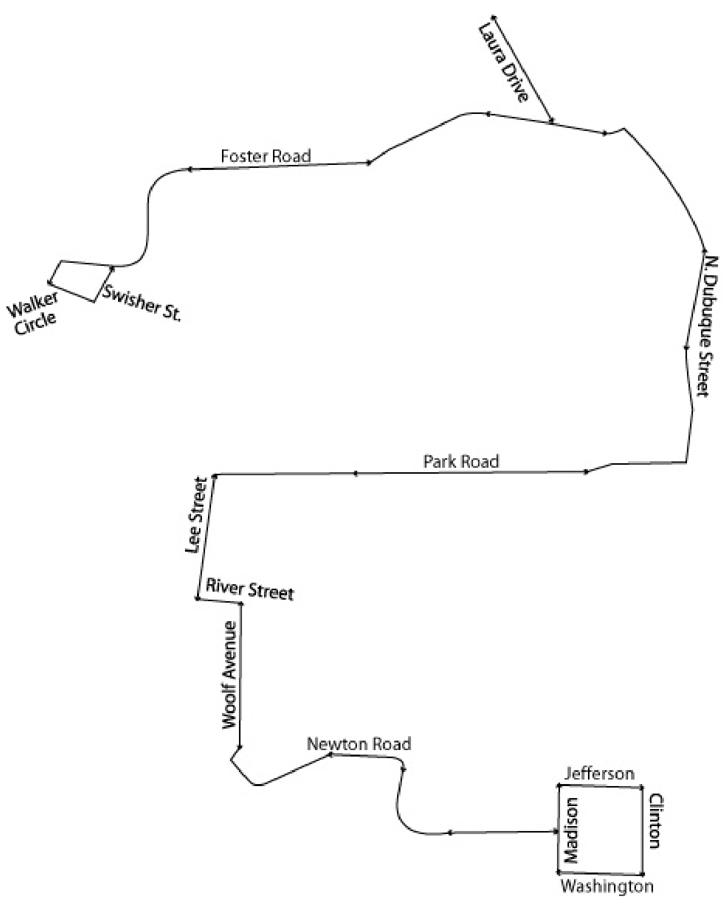
Downtown Interchange
(see front cover of map)
PENTA-
CREST
Gilbert St.
D E
Washington St.
F
A
Grand Ave.
Melrose Ave.
Byington Rd.
Madison St.
Clinton St.
Burlington St.
Summit St.
Points of Interest:
A. Kinnick Stadium
B. Oaknoll Res./Seville Apt.
C. Roosevelt School
D. Senior Center
E. City Hall
F. Recreation Center
G. Towncrest Medical Complex
H. Southeast Jr. High School
I. Mercer Park
J. Autumn Park Apartments
K. Lucas School
• Indicates time check point
(see corresponding column)
Sunset St.
Oakcrest St.
Woodside Dr.
B
C
W. Benton St.
Greenwood Dr.
7th Ave.
Muscatine Ave.
1st Ave.
K
J
Muscatine Ave.
Village Green Rd.
G
William St.
H
Wayne Ave.
I
Village Rd.
NO SERVICE ON SUNDAYS OR HOLIDAYS
Holidays are New Year's Day, Memorial Day, Independence Day,
Labor Day, Thanksgiving Day, and Christmas Day.

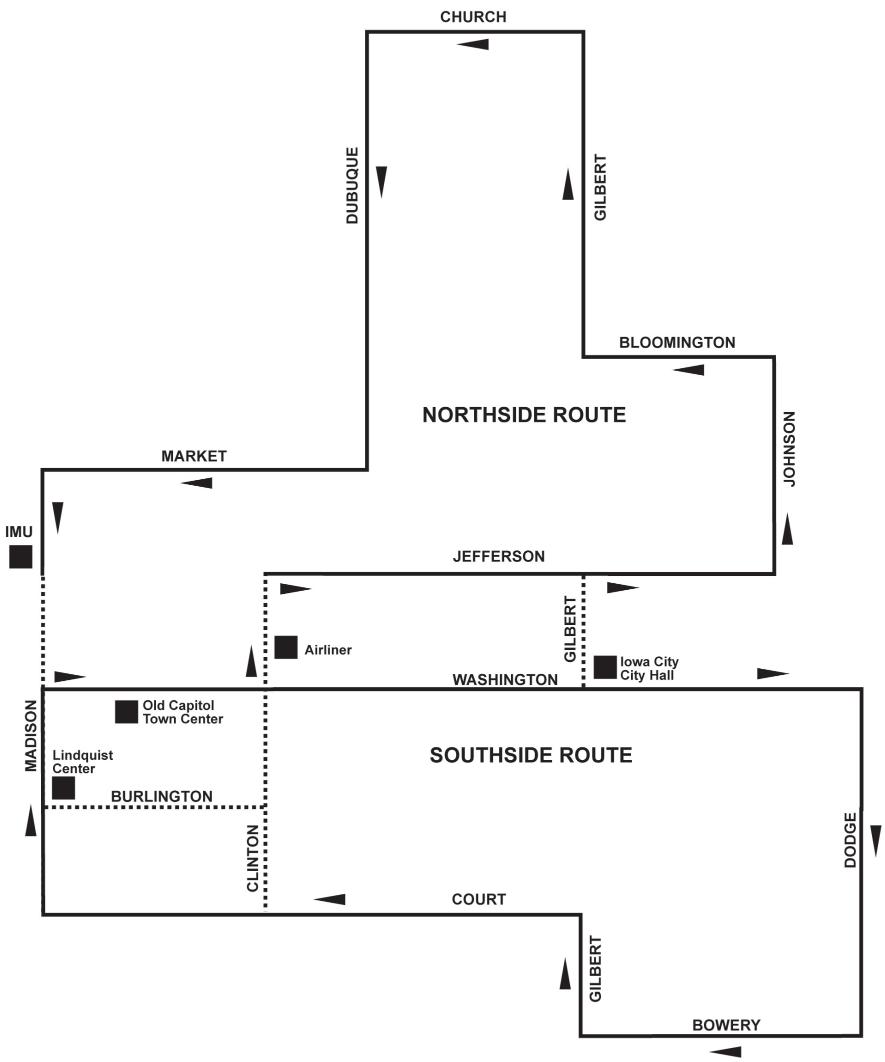


downtown
interchange
Market
Clinton
Gilbert
E
Madison
Wash'gton
F
College St.
Burlington
G
Riverside Dr.
Summit St.
H
7th Ave.
Friendship
3rd
D St.
Sheridan Ave.
7th Ave.
F St.
1st Ave.
Muscatine
Benton St.
Miller
Hwy. 1
Hwy. 6
A
B
D
C

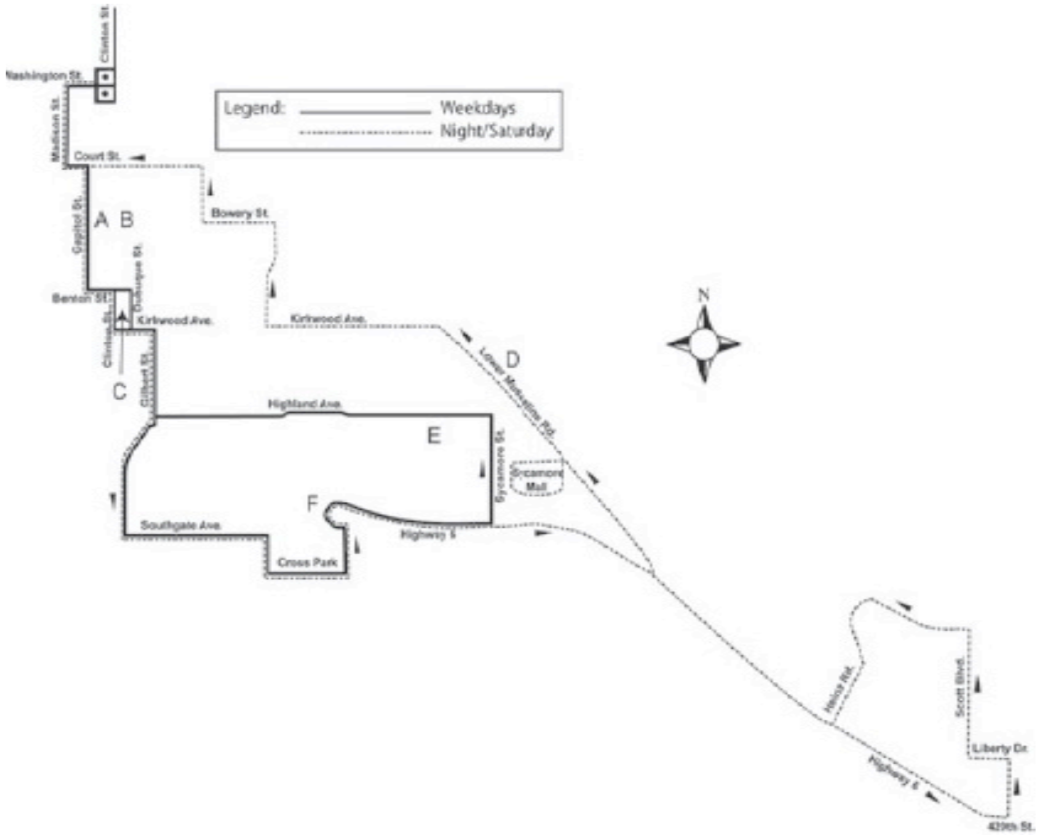


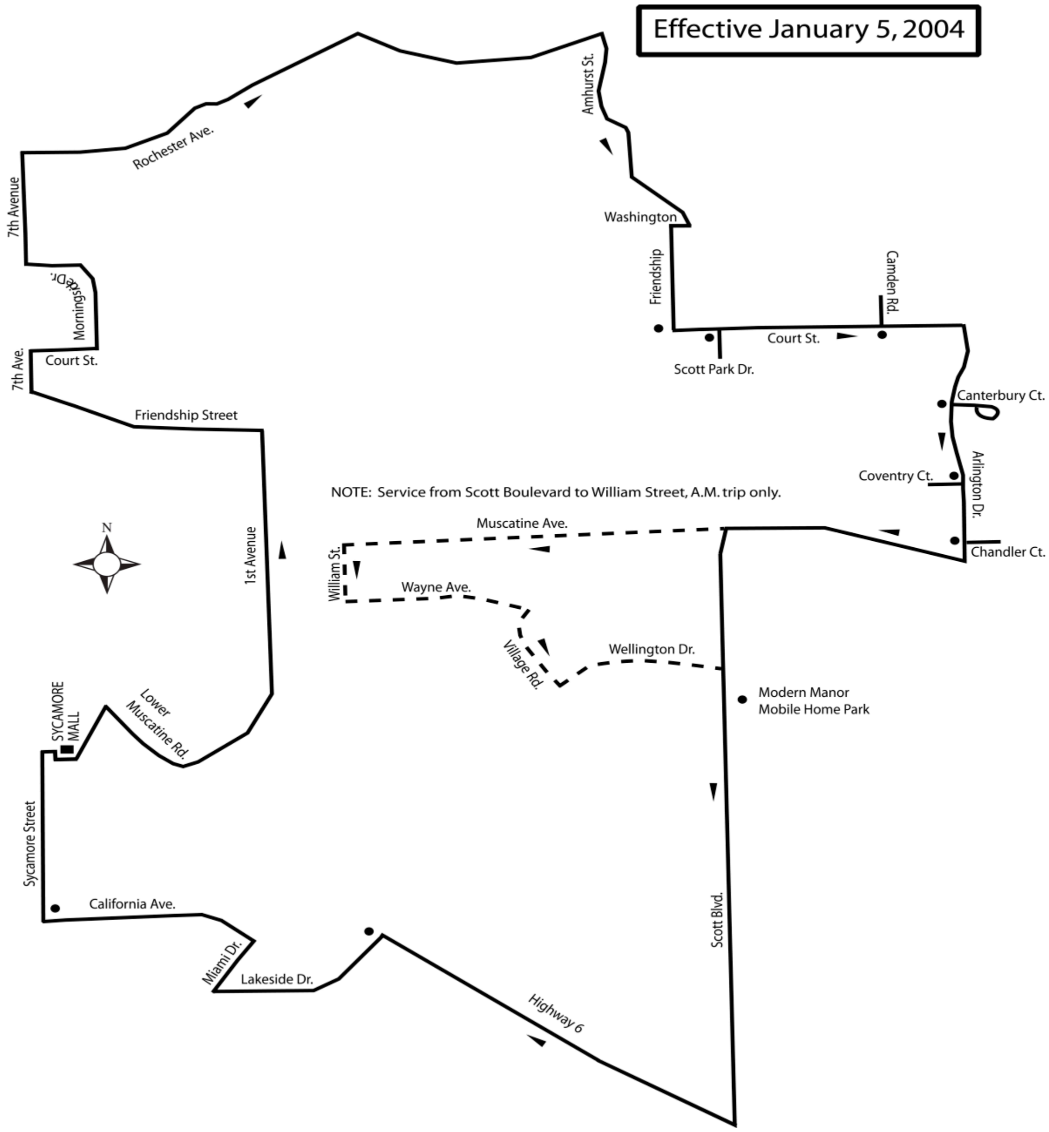
North Routes

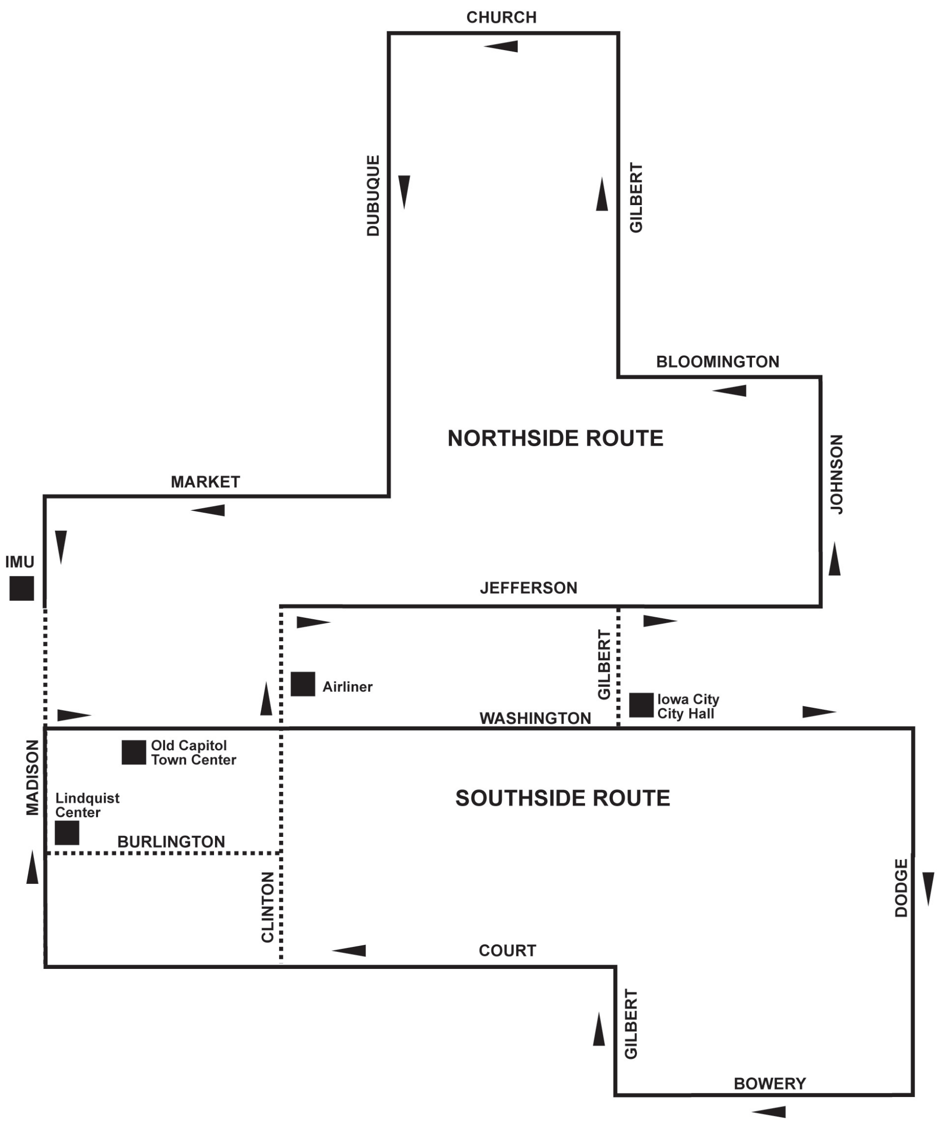



CHURCH
DUBUQUE
GILBERT
BLOOMINGTON
NORTHSIDE ROUTE
JOHNSON
MARKET
IMU
JEFFERSON
Airliner
GILBERT
Iowa City
City Hall
WASHINGTON
Old Capitol
Town Center
MADISON
Lindquist
Center
SOUTHSIDE ROUTE
BURLINGTON
CLINTON
DODGE
COURT
GILBERT
BOWERY

South Routes




Effective January 5, 2004
Rochester Ave.
Amhurst St.
Washington
7th Avenue
Morningside Dr.
Friendship
Camden Rd.
Court St.
Scott Park Dr.
7th Ave.
Court St.
Canterbury Ct.
Friendship Street
Coventry Ct.
Arlington Dr.
NOTE: Service from Scott Boulevard to William Street, A.M. trip only.
Muscatine Ave.
Chandler Ct.
N
1st Avenue
William St.
Wayne Ave.
Village Rd.
Wellington Dr.
Modern Manor
Mobile Home Park
Lower
Muscatine Rd.
SYCAMORE
MALL
Sycamore Street
California Ave.
Scott Blvd.
Miami Dr.
Lakeside Dr.
Highway 6


CHURCH
DUBUQUE
GILBERT
BLOOMINGTON
NORTHSIDE ROUTE
JOHNSON
MARKET
IMU
JEFFERSON
Airliner
GILBERT
Iowa City
City Hall
WASHINGTON
Old Capitol
Town Center
MADISON
SOUTHSIDE ROUTE
Lindquist
Center
BURLINGTON
DODGE
CLINTON
COURT
GILBERT
BOWERY