2014-12-12

Higher Wireless Connection Capacity Route Selection Algorithms for Automobiles Traveling Between Two Points

Brandon Clark Sato
University of Miami, brandonsato@gmail.com

UNIVERSITY OF MIAMI

HIGHER WIRELESS CONNECTION CAPACITY ROUTE SELECTION ALGORITHMS FOR AUTOMOBILES TRAVELING BETWEEN TWO POINTS

By

Brandon Sato

A THESIS

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Master of Science

Coral Gables, Florida

December 2014

UNIVERSITY OF MIAMI

A thesis submitted in partial fulfillment of
the requirements for the degree of
Master of Science

HIGHER WIRELESS CONNECTION CAPACITY ROUTE SELECTION ALGORITHMS FOR AUTOMOBILES TRAVELING BETWEEN TWO POINTS

Brandon Sato

Approved:

Dilip Sarkar, Ph.D. Associate Professor of Computer Science	Manohar Murthi, Ph.D. Associate Professor of Electrical and Computer Engineering
Mitsunori Ogihara, Ph.D. Professor of Computer Science	M. Brian Blake, Ph.D. Dean of the Graduate School

SATO, BRANDON (M.S., Computer Science)
Higher Wireless Connection Capacity Route (December 2014)
Selection Algorithms for Automobiles Traveling Between Two Points

Abstract of a thesis at the University of Miami.

Thesis supervised by Associate Professor Dilip Sarkar.
No. of pages in text. (37)

A GPS system selects routes between two points with minimum physical distance or minimum driving time. Here we address a different type of route selection problem. Given a road map with driving distance and wireless connectivity for each road segment, find a driving route that maximizes total wireless connectivity while its length is bounded by a predetermined value.

In this thesis, we present three heuristic algorithms. Initially they compute the shortest path for determining a distance bound. The first modifies the road map by replacing each road segment with the ratio of the distance and wireless communication capacity, and selects a route on this modified road map that satisfies the length bound. The second assigns a penalty value to intersections based on their distance from a shortest path. The closer the intersection to such a path, the higher the penalty value. It selects among unexplored intersections one that has the minimum penalty value. The final algorithm utilizes the first algorithm twice for selecting a route once to find distance and communication capacity of each intersection from the origin and then the same from the destination.

Through extensive simulation of grid road networks, we find that all three algorithms select routes that have higher communication capacity than any shortest

paths. More interesting is that the communication capacity gain is higher than the route length increase.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Advancement of wireless technologies, ease of its use, and desire of users to remain always connected to the Internet from everywhere forces service providers to increase availability from automobiles. Available connection speeds of 2G, 3G, and 4G networks varies widely. The same is true for WLAN, WiMax and WiFi. In some areas data rate is good enough for receiving streamed HD videos and in other areas there is barely any wireless connection for a voice over IP connection!

While traveling in an automobile, variability of connection speed poses an interesting problem from the connection capacity point of view. Is there a driving route between two points such that total wireless connection capacity is maximized, but total driving distance, or driving time is minimum or close to minimum? We define *wireless connection capacity* of a path as the amount of data that can be uploaded or downloaded while traveling on the path at a designated driving speed. By utilizing network coverage data from online sources, it may be possible to determine a solution to this problem. We can combine road maps with network coverage information to determine driving time and connection capacity on a driving route. Here we explore one possible application of doing so.

1.1 Related Work

To the best of our knowledge, the problem was first introduced in [1]. Some benefits for finding solutions to the problem are illustrated in [2] and [3]. One of the two algorithms presented in [4], finds a shortest route with maximum communication capacity. Being a modified version of Dijkstra's algorithm, it takes $O({I_n}^2)$ time for a grid-type road network work with I_n intersections. The other algorithm is for selecting a route whose length may be longer than the shortest route, but bounded by a multiplicative or additive constant; for instance, twice the length of the shortest route. The algorithm keeps the spirit of Dijkstras algorithm, but complexity of the algorithm is at least $O(I_n^4)$, if not higher[1]. Since bounded-length optimal path problem is NP-hard, it is unlikely to have any polynomial time algorithm. The most practical approach for solving this problem is to develop heuristic based polynomial time algorithms.

1.2 Main Contribution of Thesis

We present three heuristic-algorithms with $O(I_n^2)$ time complexity for grid-type road networks to find higher connection capacity routes within a given bound of the shortest path. Initially each computes the maximum connection-capacity shortest path for determining an upper bound for route length. The first algorithm (i) modifies

[1]Complexity analysis is not provided in [4]

the road map by replacing each road segment with the ratio of the distance of the road segment and its wireless communication capacity, and (*ii*) selects a route on this modified road map that satisfies the route-length bound. The second algorithm assigns a penalty value to intersections based on their distance from any shortest path. The closer an intersection is to a shortest path, the higher the penalty value. The algorithm selects among unexplored intersections one that has the minimum penalty value. The final algorithm utilizes the first algorithm twice for selecting a route once to find distance and communication capacity of each intersection from the origin and then to find the same from the destination. After distance and communication capacity of all intermediate intersections from source and destination are known, an intermediate node that meets the distance bound and has the highest connectivity is selected. Paths from source and destination to this intersection are stitched together to get the desired path.

1.3 Outline of the Thesis

The remainder of the thesis is organized as follows: Chapter 2 defines the problem and terms used in the thesis. Chapter 3 presents the heuristic algorithms introduced in the thesis in both pseudocode and written form Chapter 4 explores each algorithm's performance. To do this, we first present example failure cases. Following that, we use experimental simulations and analyze the average case performance on randomly generated road networks. Chapter 5 is a summary of the thesis and what further work may be done to continue the research presented here.

Chapter 2

Background Information

First we define a road network as a weighted graph. Every vertex of the graph is an intersection of two or more road segments. The weights of a road segment is its physical distance or time it takes to drive from one end to the other end of the road segment. Formally,

Definition 1 (Road Network) *A* road network $RN = (I, R)$ *is defined as a weighted undirected graph with a set of* n_I *intersections* $I = \{I_1, I_2, \cdots, I_{n_I}\}$*, and a set of* n_R *road segments* $R = \{R_1, R_2, \cdots, R_{n_R}\}$*. A road segment* $R_j = (I_k, I_l)$ *connects two distinct intersections* I_k *and* I_l*, and have a positive weight* w_j*, which is either the physical distance between them or the driving time from one end to the other end of the road segment.*

Definition 2 (Path) *A* path $P = (I_{k_1}, I_{k_{n+1}})$ *from a road intersection* I_{k_1} *to a road intersection* $I_{k_{n+1}}$*,* $k_1 \neq k_{n+1}$*, is a sequence of* n *road segments* $R_{k_1}, R_{k_2}, \cdots, R_{k_n}$ *such that* R_{k_i} *directly connects intersections* I_{k_i} *and* $I_{k_{i+1}}$*, for* $1 < i \leq n$*. Also, no* *intersection occurs more than once in a path, that is, no loop is allowed in a path.*

Definition 3 (Path Length) *The* path length LP *of a* path P *is the sum of the weights of the road segments on the path, that is,*

$$LP = \sum_{i=1}^{n} w_i \tag{2.1}$$

A *shortest path*, SP, from an intersection I_k to an intersection I_l, $k \neq l$, is a path whose path length is minimum among all the paths between the intersections I_k and I_l. It is assumed that wireless connections are available to *mobile terminals* (MTs) from some road segments. Thus, underlying a road network graph, there is a wireless capacity graph. Formally,

Definition 4 (Wireless Capacity Graph) *a* wireless capacity graph $WCG = (I, R, D)$ *is defined as an weighted undirected graph with the set of intersections I and the set of road segments R of the road network RN, and a set of n_R wireless connection capacity $D = \{D_1, D_2, \cdots, D_{n_R}\}$, where D_j is wireless connection capacity of R_j. The value of D_j is the maximum available data a mobile terminal can transfer while traveling at a normal driving speed on the road segment R_j while using only one transceiver.*

For simplicity of presentation, we assume that a channel has a fixed capacity, and once a channel is assigned to an user, its capacity does not change unless a new channel is assigned (and a hand off occurs).

Definition 5 (Wireless Connection Capacity) *Wireless connection capacity, WC, of a path $P =< R_{k_1},\ R_{k_2},\ \cdots,\ R_{k_n} >$ is the sum of the wireless connection capacity* *of the road segments on the path, that is,*

$$WC = \sum_{i=k_1}^{k_n} D_{k_i} \tag{2.2}$$

Definition 6 (Wireless Connectivity Ratio Graph) *For a road network RN and its corresponding wireless capacity graph WCG, the* wireless connectivity ratio graph *$WCRG = (I, R, D, C)$ is defined as a weighted undirected graph with the set of intersections I, the set of road segments R, the set of wireless connectivity capacity D, and the set of ratio $C = \{C_1, C_2, ..., C_{n_R}\}$, where $C_j = R_j/C_j$ for every road segment R_j and wireless connection capacity D_j.*

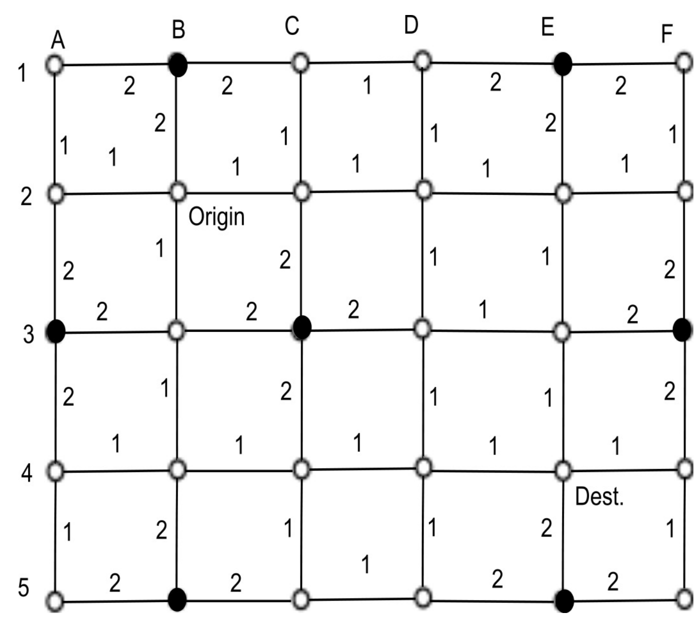

Figure 2.1: Example Wireless Access Network

Let us look at an example grid network, such as Manhattan streets, for illustrating the concepts we have defined, and for illustrating the concepts we will introduce next.

Example The road network shown in Fig. 2.1 consists of 30 road intersection points and 49 road segments. For ease of reference we have numbered the columns with letters, and rows with numbers. A road intersection is thus an ordered pair <row number, column letter>. While the physical distance, or driving speed, or driving time for each road segment can be different, to simplify discussion it is assumed that the distance, driving speed, and driving time for all road segments are identical (**for this example**). Let us assume that the physical distance of a road segment is 100 meters, driving speed is 36 kilometers per hour, which translates to a diving time of 10 seconds for each road segment. As will be clear in our subsequent discussions, this parameter values will simplify presentation without sacrifice to generality of the algorithms presented.

The connection speed for each of the road segments is labeled in the figure. For instance, four road segments around <3, C> have connection speed 2 Mbps and the road segments around <2, D> have connection speed 1 Mbps. As stated above, the driving time for each road segment is 10 seconds. Thus, wireless data capacity of a road segment is 10 or 20 Mb.

Let us consider paths from intersections $< 2, B >$ (`Org`) to $< 4, E >$ (`Dst`). There are 10 shortest paths of length 500 meters between them. For instance, the road segments connected by the intersections $< 2,\ B >$, $< 2,\ C >$, $< 2,\ D >$, $< 2,\ E >$, $< 3,\ E >$, and $< 4, E >$ forms one shortest path. A car driving at 36 KM per hour from $< 2, B >$ to $< 4, E >$ on any of these ten shortest paths take exactly 50 seconds. However, if we consider amount of wireless connection capacity of these paths, we can divide these 10 paths into two groups: four path have connection capacity 50 Mb and the other six paths have connection capacity 70 Mb. Indeed, any shortest path that include intersection $< 3, C >$ has connection capacity 70 Mb. This is so because of the fact that four road segments around this intersection have connection capacity of 2 Mb. Thus, any path that include the intersection $< 3, C >$ have wireless connection capacity 70 Mb.

Chapter 3

Approach

Building upon the highest connection capacity shortest paths found in the example in Chapter 2 and Figure 2.1, we examine heuristic algorithms for finding longer paths that have higher connection capacity without sacrificing computational complexity.

First a modified version of Dijkstra's shortest path algorithm from [5] is presented in Section 3.1. This algorithm finds a shortest path with maximum communication capacity and is used as a subroutine for further algorithms. Next, we alter Dijkstra's shortest path algorithm to utilize the Wireless Connectivity Ratio Graph to find higher connectivity paths. This is presented in Section 3.2. Further, in Section 3.3 we present an algorithm that prefers routes further from the shortest paths while utilizing the $WCRG$. The final algorithm is presented in Section 3.4, in which we explore even more paths by determining an intermediate intersection to pass through in order to improve connectivity.

For description of the algorithms, we adopt notation from [5]. The set of intersections of WCG is denoted by $WCG.I$. Similarly, distance and connection capacity of currently known path from the origin intersection (Org) to the intersection I_k are denoted as $I_k.dist$ and $I_k.cc$, respectively. The set of intersections adjacent to I_k is denoted by $WCG.Adj[I_k]$.

3.1 Max Connection-Capacity SP Selection Algorithm

Since the algorithms presented here utilize a maximum connection-capacity shortest path selection algorithm from [4], we present an outline of the algorithmin Fig. 3.2.

`Initialize-SP`(WCG, Org)
`for each` $I_k \in WCG.I$
    $I_k.dist = \infty;$
    $I_k.cc = 0;$
$Org.dist = 0;$

Figure 3.1: Initialization Procedure

The algorithm initializes the variables by calling the procedure `Initialize-SP`(WCG, Org). The inputs to initialization procedure (shown in Fig. 3.1) are the wireless connection capacity graph WCG and the origin intersection, Org. The set of intersections whose final value of distance and connection capacity are known is kept in S (see lines `02` and `06` in Fig. 3.2). As the algorithm starts, S is an empty set.

The *priority* queue, Q, keeps all the intersections of WCG that are not in S. The procedure `Mix-dist-Max-cc`(Q) on line `05` extracts from Q the intersection I_k that has minimum distance from Org. If more than one intersections have minimum distance, one of the intersection with maximum connection capacity is selected. The lines `07` through `11` update distances and connection capacities of all intersections adjacent to I_k. Lines `05` through `11` are repeated until Q is empty. Note that one could stop the algorithm once the intersection Dst is selected (line `05`).

Next we present a an algorithm for selecting a path that has higher connection capacity than a maximum connection-capacity shortest path, and the length of the path is bounded by a predetermined value.

```
Max-Connection-SP(WCG, Org, Dst)
01 Initialize-SP(WCG, Org);
02 S = φ;
03 Q = WCG.I;
04 while Q ≠ φ;
05    I_k = Min-dist-Max-cc(Q);
06    S = S ⋃ {I_k};
07    for each I_j ∈ WCG.Adj[I_k];
08       let R_i = (I_k, I_j);
09       if (I_j.dist > I_k.dist + w_i);
10          I_j.dist = I_k.dist + w_i;
11          I_j.cc = I_k.cc + D_i;
12 return Dst.cc;
```

Figure 3.2: Modified Dijkstra's Shortest Path Algorithm for Selecting Maximum Wireless Connection Capacity Shortest Path

3.2 Algorithm for Selecting Length-Bounded Route on $WCRG$

Before we present the algorithm, some additional notations are necessary. For an intersection I_k, the known ratio-distance from the origin intersection, `Org`, is denoted as $I_k.ratioDist$, which is computed using the ratio values in C. Also, $WCRG.I$ and $WCRG.Adj[I_k]$ denotes the set of intersections in $WCRG$ and the set of intersections adjacent to the intersection I_k, respectively.

The proposed algorithm works in two phases. In the first phase, the length and connection capacity of the shortest paths to all intersections from the origin is calculated by calling the Algorithm shown in Fig. 3.2. The shortest distance to the intersection Dst returned by the algorithm is used to compute distance bound GL as shown below.

$$GL = lb * Dst.dist; \tag{3.1}$$

where lb, for $1 \leq lb$, is a predetermined multiplicative factor for bounding the distance between Org and Dst. In the next phase, a modified version of the shortest path algorithm is used on the wireless connectivity ratio graph $WCRG$. The algorithm uses the ratio values in C for selection of paths. If $R_{i_1}, R_{i_2}, \cdots, R_{i_l}$ are the road segments on a path, the *length* of the ratio-path for this algorithm is $PL = \sum_{k=1}^{l} C_{i_k}$ (not the distance but the sum of ratios).

In addition to calculating and storing the total sum of ratios along a path, the total physical distance and total connection capacity are also calculated and stored. However, if the calculated distance to a neighboring intersection is greater than the distance bound GL, the road segment is disregarded. The algorithm is shown in Fig. 3.3.

The procedure `Min-ratioDist-Max-cc(`Q`)` on line `08` finds from Q the next intersection I_k that has minimum ratio distance from the origin intersection, `Org`. If

```
Dist-Bounded-Path(WCRG, Org, Dst, LG)
Initialize-SP(WCRG, Org);
for each I_k ∈ WCRG.I
    I_k.ratioDist = ∞;
Org.ratioDist = 0;
S = φ;
Q = WCRG.I;
while Q ≠ φ;
    I_k = Min-ratioDist-Max-cc(Q);
    if (I_k.dist < GL)
      S = S ⋃{I_k};
      for each I_j ∈ WCRG.Adj[I_k];
        let R_i = (I_k, I_j);
        if I_j.ratioDist > I_k.ratioDist + C_i;
          I_j.ratioDist = I_k.ratioDist + C_i;
          I_j.cc = I_k.cc + D_i;
          I_k.dist = I_k.dist + w_i;
return Dst.cc;
```

Figure 3.3: Modified Dijkstra's Algorithm for Selecting Distance-Bounded Path using Ratio of Distance of Road Segments and Their Communication Capacity.

more than one intersections have minimum ratio distance, an intersection with highest communication capacity is selected at random from them.

Presented in Fig. 3.4 is an example graph which we can step through running the algorithm. We can see that the origin is $< 2, A >$ and the destination is $< 1, D >$. We find the shortest path to the destination to be 4 units. With a 25% increase in distance allowed, we have an upper bound distance of 5 units. Next, we will run the shortest path algorithm with ratio distances. We will add all neighbors of the origin to the queue, checking to be sure that each is within the upper bound of 5 units. Each node passes this test, so the minimum ratio distance intersection in the queue is $< 2, B >$ with a value of 0.5. We then add unvisited neighbors of $< 2, B >$ to the queue, again checking for distance within the bound. From the queue, $< 1, A >$,

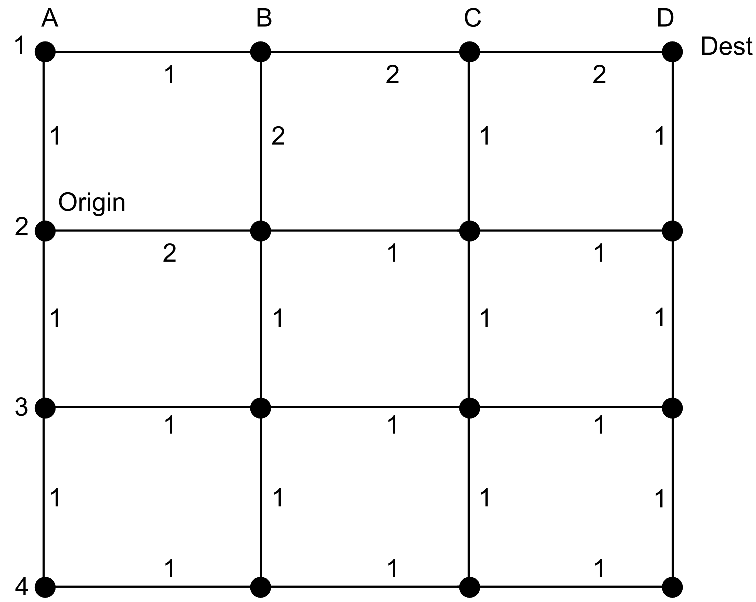

Figure 3.4: Example Network for Ratio Distance Algorithm

$< 3, A >$, and $< 1, B >$ then have a total ratio distance of 1.0. Visiting $< 1, A >$ will add no new intersections to the queue. Visiting $< 1, B >$ and $< 3, A >$ will add each intersection's unvisited neighbors after determining each to be within the distance bound. The minimum ratio distance intersection is now $< 1, C >$ with a total ratio distance of 1.5. Visiting $< 1, C >$ adds the destination, $< 1, D >$, to the queue which is tied for minimum ratio distance in the queue, with a value of 2.0. The minimum ratio distance to the destination within the distance bound is now known.

This algorithm finds some increased connectivity paths, but does not explore all paths within a given bound. To explore paths further from the shortest paths, an algorithm is presented next.

3.3 Algorithm for Selecting Length-Bounded Route on $WCRG$ with Penalty for Shorter Routes

The algorithm presented here is designed to explore first paths away from the shortest path. To achieve that objective, the algorithm penalizes intersections that are near to or on a shortest path to the destination in order to reach distances closer to the distance bound. The penalty value for each intersection is calculated by adding the current path distance and the distance to the destination and subtracting that sum from the distance bound. Similar to the algorithm presented in the previous section, it works in two phases. The shortest distance from each intersection to the destination is calculated in the first phase using the modified Dijkstra's algorithm shown in Fig. 3.2. The shortest distance obtained from this phase is utilized for calculation of distance bound using equation 3.1. An outline of the algorithm for the second phase is shown in Fig. 3.5. The next intersection, I_k, to be considered for exploration is obtained by calling `Min-penaltyDist-Max-cc(`Q`)` on line `10`. Among the intersections in Q, the intersection that has a minimum penalty is selected by `Min-penaltyDist-Max-cc(`Q`)`. After an intersection I_k is selected, the penalty value for each intersection, I_j, adjacent to I_k is calculated by adding the current path distance ($I_j.dist$) and the distance to the destination ($I_j.destDist$), and subtracting that sum from the distance bound (GL) as shown on lines `16` and `17`. Formally,

$$I_j.penaltyDist = GL - (I_j.dist + I_j.destDist). \tag{3.2}$$

```
Dist-Bounded-Path-with-Penalty(WCG, Org
                                Dst, LG)
for each I_k ∈ WCG.I
    I_k.destDist = I_k.dist;
    I_k.penaltyDist = ∞;
Org.penaltyDist = 0;
Initialize-SP(WCG, Org);
S = φ;
Q = WCG.I;
while Q ≠ φ;
    I_k = Min-penaltyDist-Max-cc(Q);
    if (I_k.dist < GL)
      S = S ⋃ {I_k};
      for each I_j ∈ WCG.Adj[I_k];
        let R_i = (I_k, I_j);
        if I_j.dist > I_k.dist + w_i;
          I_j.penaltyDist = GL − (I_j.dist
                                  + I_j.destDist);
          I_j.cc = I_k.cc + D_i;
          I_k.dist = I_k.dist + w_i;
return Dst.cc;
```

Figure 3.5: Modified Dijkstra's Algorithm for Selecting Distance-Bounded Path with Penalty for Shorter Paths

Let us examine the initial steps of the algorithm running on the example graph in Fig 3.6. First we determine the shortest distance from each node to the destination by reversing any direction in the graph and running the shortest distance algorithm using the destination as the origin. The shortest distance path is then 4, and we allow for a 50% increase which brings our upper bound to 6.

Next, we begin adding neighbors of the origin to the queue. To calculate the penalized distance, which is used for queuing purposes, we take the upper bound (6) and subtract the distance of the node from the origin. From the resulting value, we subtract the distance of the shortest path from the intersection to the destination.

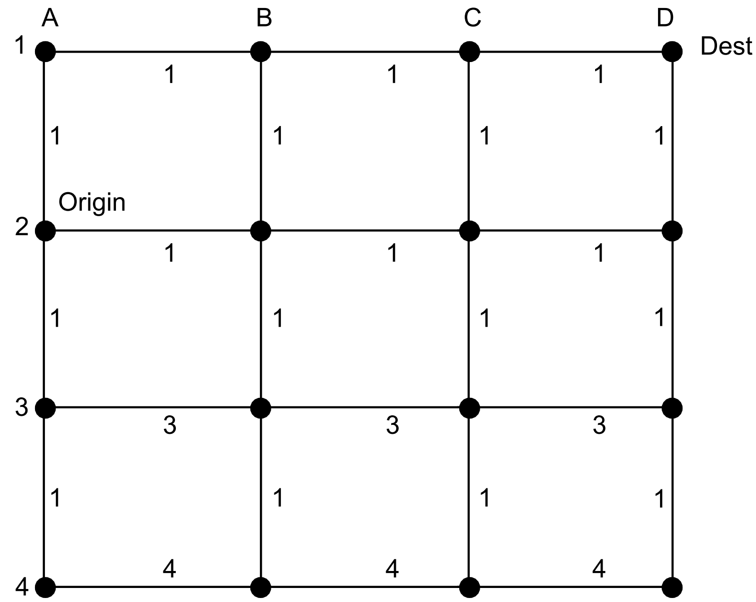

Figure 3.6: Example Network for Ratio Distance with Penalty Algorithm

We can then see for intersections $< 1, A >$ and $< 2, B >$, the penalized distance is $6 - (1 + 3) = 2$. For intersection $< 3, A >$, the penalized distance is $6 - (1 + 5) = 0$. Then, the next node to visit is $< 3, A >$. If the penalized distance ever becomes negative, we discard the road segment connecting to that intersection as any path using it will exceed our bound.

Because of differences in next intersection selection procedure, this algorithm discovers paths that are missed by the previous algorithm. Next we propose an algorithm that calls the algorithm described in Section 3.2 twice.

3.4 Two-Pass Algorithm for Selecting Length-Bounded Route on $WCRG$

The final algorithm we present here searches for a path by choosing an intermediate intersection and stitching together the shortest-ratio paths which pass through that intersection. It uses the ratio graph $WCRG$, instead of distance graph WCG. An outline of the algorithm is shown in Fig. 3.7. The algorithm works in two phases,

```
Two-Pass-Bounded-Dist-Path(WCRG, Org,
                                  Dst, LG)
Dist-Bounded-Path(WCRG, Org, Dst, LG);
for each I_k ∈ WCRG.I;
   I_k.OrgDist = I_k.dist;
   I_k.OrgCc = I_k.cc;
Dist-Bounded-Path(WCRG, Dst, Org, LG);
for each I_k ∈ WCRG.I;
   I_k.DstDist = I_k.dist;
   I_k.DstCc = I_k.cc;
Dst.cc = 0;
for each I_k ∈ WCRG.I;
   if I_k.OrgDist + I_k.DstDist ≤ LG
     if I_k.OrgCc + I.Dst.Cc > Dst.cc
        Dst.cc = I_k.OrgCc + I.Dst.Cc;
return Dst.cc;
```

Figure 3.7: Two-Pass Algorithm for Selecting Distance-Bounded Path

similar to the algorithms presented earlier. In the first phase, the algorithm for maximum connectivity shortest path is called to obtain length of the shortest path and then it is used in equation 3.1 to compute distance bound GL. In the second phase, first distance and communication capacity of each intersection from `Org` is computed by calling `Org` `Dist-Bounded-Path(`$WCRG, Org, Dst, LG$`)`; the results are stored in $I_k.OrgDist$, and $I_k.OrgCc$ for each intersection I_k (see lines `02` to `05`). Next, `Dist-Bounded-Path(`$WCRG, Dst, Org, LG$`)` is called switching the role of origin and destination. Thus, distances and communication capacity of each intersection from the destination intersection is obtained and stored in $I_k.DstDist$, and $I_k.DstCc$ for each intersection I_k. Computation steps are shown in lines `06` to `09`. Finally, the best communication capacity route is computed in lines `09` to `14`.

3.5 Implementation

For the implementation of the algorithms, we first must have Road Networks to test them on. The simplest way to do this is by utilizing two-dimensional arrays representing the adjacency matrix for the Road Networks. The same type of data structure is utilized for both the Wireless Capacity Graphs and the Wireless Connectivity Ratio Graphs.

For the simplicity's sake, we utilize grid-type Road Networks which are generated with the given number of rows and columns, with each intersection equidistant from those adjacent to it. We can then choose to pass the connectivity of road segments, pass the locations of hotspots with a given connectivity, or generate a given number of randomly located hotspots with a given connectivity. We can also choose to have entire rows of road segments have increased connectivity.

In the case of randomly generated hotspots, we generate random integers to designate hotspot placement until we have the given number of hotspots placed, making sure to discard any duplicates.

Chapter 4

Experiments & Results

In order to evaluate the performance of these algorithms, we have run them on many different road networks. Here we present those experiments and the results. In Section 4.1 we individually analyze each algorithm for successes and shortcomings of the desired outcome. That is, we present example cases where each algorithm does and does not find the optimal possible connection capacity within the desired bound. In Section 4.2, we generate multiple random road networks to determine the average improvement that each algorithm may give.

4.1 Initial Evaluation of Algorithms

We begin by determining cases in which each algorithm succeeds in finding higher connection capacity routes within a bound and cases where each fails to find a path that is available.

4.1.1 Algorithm for Selecting Length-Bounded Route on $WCRG$

First we evaluate the algorithm described in Section 3.2. Let us first look at a grid

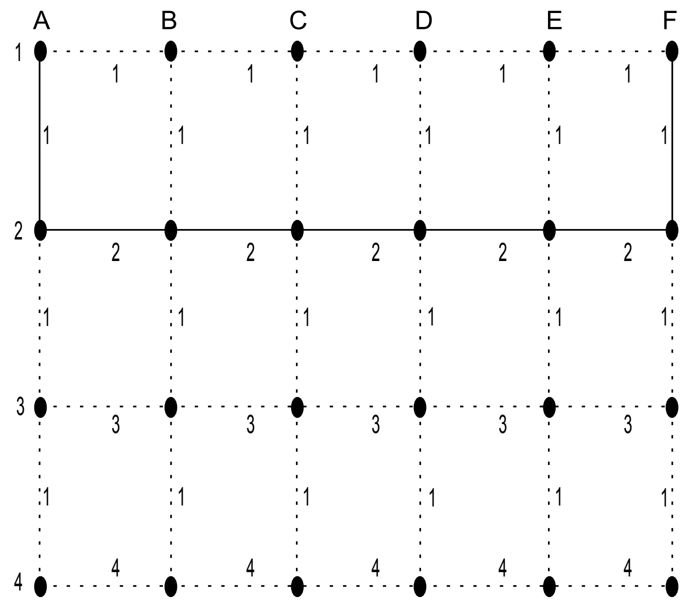

Figure 4.1: Example Success for Connectivity Ratio Algorithm. Road segments labeled with Wireless Connectivity value. Selected route is represented by solid lines.

network such as that in Figure 4.1 where the origin <1,A> and destination <1,F> have a direct path with low connectivity. Longer paths may utilize higher connectivity. This models a main road with relatively low connectivity and improved connectivity on side roads in an attempt to alleviate high traffic on the congested main road.

Using the algorithm shown in Figure 3.3, we can specify a desired overhead that will provide a path within a percent increase of the shortest path while maintaining a higher average connectivity. As shown in Figure 4.1, there is a clear shortest path of length 500 meters with connection capacity of 5. Given a 40% increase in distance from the shortest path, the algorithm will give a path of length 700 meters but an increase of connectivity from 5 to 12, or a 140% increase in connectivity overall and an 71% increase in capacity per unit distance.

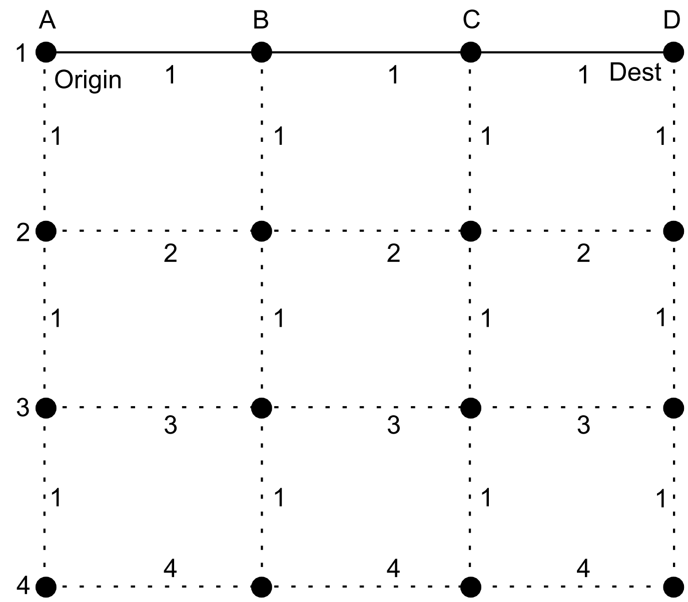

Figure 4.2: Connection Capacity Ratio Algorithm Failure. Desired length is twice the shortest path. Road segments labeled with Wireless Connectivity value. Selected route is represented by solid lines.

As seen in Figure 4.2, this algorithm will not choose a higher connectivity path even if one is available within the desired bound. This failure is illustrated showing that the path $< 1, A >$, $< 2, A >$, $< 2, B >$, $< 2, C >$, $< 2, D >$, $< 1, D >$ has length 5 with connectivity total of 8. However, the path illustrated with a solid line of length 3 and connectivity 3 is chosen, even when the desired bound is twice the length of the shortest path. The cause of this is that the path is built incrementally, edge by edge as a sum of ratios. This does not optimize the overall ratio. Ideally we would like to find the lowest ratio of sums.

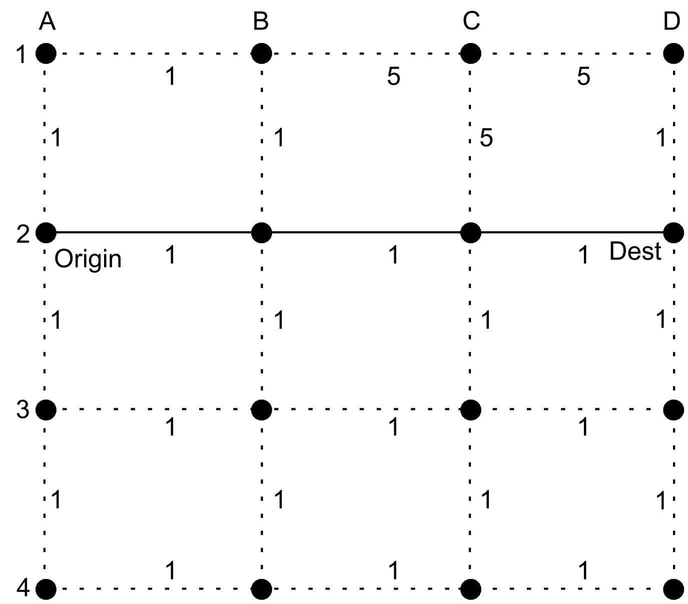

Figure 4.3: Penalty Failure. Road segments labeled with Wireless Connectivity value. Selected route is represented by solid lines.

4.1.2 Algorithm for Selecting Length-Bounded Route on $WCRG$ with Penalty for Shorter Routes

Next, we examine the algorithm introduced in Section 3.3. This algorithm allows exploration of nodes that end up further away from the shortest paths sooner, partially mitigating the issue of incremental path creation. This algorithm finds longer paths than the algorithm in Section 3.2, but still fails to find an ideal path in some cases. It succeeds by finding the mentioned 5 length path with connectivity 8 in Figure 4.2. However, it still fails to choose the ideal path in other cases. To illustrate such a situation, we present the Road Network in Figure 4.3. The algorithm still chooses a path with length 3 and connectivity 3 instead of a better connected path with ratio 13:5, available by traversing through $< 3, A >$.

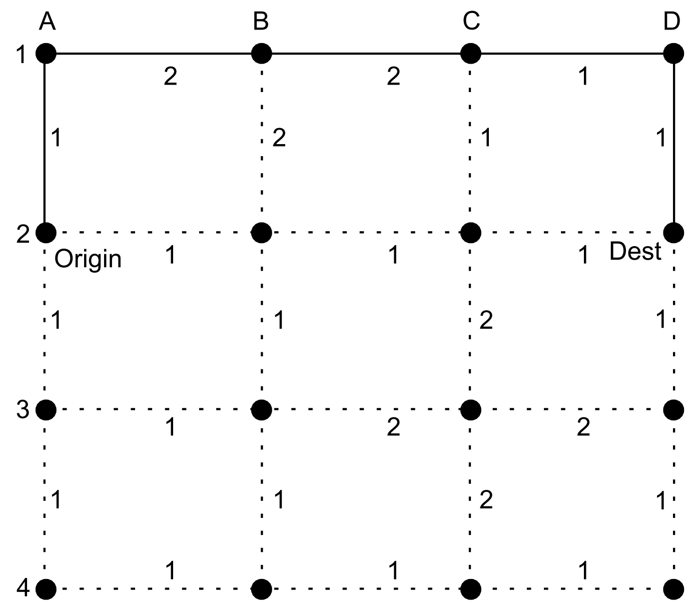

Figure 4.4: Two-Pass Connection Capacity Ratio Failure. Road segments labeled with Wireless Connectivity value. Selected route is represented by solid lines.

4.1.3 Two-Pass Algorithm for Selecting Length-Bounded Route on $WCRG$

Now, we analyze the algorithm from Section 3.4. This method can give very good results for finding a high capacity path and results are improved with the second pass. It can fail to find the optimal path in cases similar to the single pass Connection Capacity Ratio in that if a less than ideal intermediate path has the same properties as the overall path in the single pass example, it will be chosen over a more desired path. Such a failure is illustrated in figure 3 where the solid path with distance 5 and connectivity 7 path is chosen over the path $< 2, A >$, $< 1, A >$, $< 1, B >$, $< 1, C >$, $< 2, C >$, $< 3, C >$, $< 3, D >$, $< 2, D >$, which has a distance 7 and connectivity 11 path. Even in such a case, it will generally give the same or better results than the single pass Connection Capacity Ratio algorithm.

4.2 Experimental Evaluation

In order to analyze the presented algorithms, we utilize them on instances of randomized experimental road networks. That procedure is described in this section.

4.2.1 Evaluation Method

The results presented here are for a 20 by 20 grid with 400 intersections and 760 road segments. An origin and destination with a shortest path of 20 road segments was chosen. While the network size and distance between two adjacent intersections were kept constant, the communication capacity of the road-segments were varied by placing hotspots at randomly selected intersections. We present two sets of results: one for a set of 45 hotspots and the other for a set of 50 hotspots. It may be worth noting that 45 and 50 represent only 11.25% and 12.5% of all intersections, respectively. A hotspot allows for connectivity of 5 on each of the road segments connected to the selected intersections. All other road segments have connectivity of 1. Note that these are relative values, with 1 being the baseline. The value of 5 for hotspot accessible road segments is reasonable when comparing two widely available network types with differing speeds such as LTE and 3G data networks. LTE download speeds can be anywhere between 3 and 12 times faster than 3G. To evaluate the effects of altering the relative connection speed for hotspot accessible road segments, we vary the value for those segments. In addition to the value of 5, we also test relative values of 2, 3, and 4.

To eliminate any initialization bias, we generated 1000 random communication capacity networks for each set of intersections. Thus, presented results are averages of 1000 observations. On each network, a fixed shortest path was evaluated for connection capacity to simulate a major through-way that may or may not be ideal for connectivity. Next, the shortest path with highest connection capacity was found using the algorithm and its connection capacity was determined. Finally, all four path-selection algorithms computed their respective cumulative communication capacities for each random communication capacity network. The algorithms were allowed 10%, 20%, 30%, and 40% increases in path length which correspond to allowed distances of up to 22, 24, 26, and 28, respectively.

4.2.2 Overall Performance Metric

Before we present our results it is important to establish a performance metric. Since the algorithms find bounded-length paths, cumulative communication capacity may not be comparable. For instance, if path length is bounded by 24 distance units and two algorithms find the same cumulative communication capacity 48. Which algorithm is better? To answer this question one needs to know the length of paths selected by these algorithms. To avoid any confusion, we normalize communication capacity to per unit distance. An algorithm is better than another if on an average its normalized communication capacity is higher than that of the other algorithm.

Table 4.1: Performance Evaluation on 20x20 Road Network with 45 Hotspots and 20% path length increase allowed. Hotspot Connection Capacity of 5

Algorithm	Average Distance	Average Connectivity	Average Unit Distance Connectivity	Avg Connectivity Increase	Average Distance Increase
Fixed	20.0	36.9	1.85	0.0%	0.0%
Best Shortest	20.0	54.2	2.71	46.8%	0.0%
Ratio	22.3	71.8	3.22	94.5%	11.4%
Penalty	20.6	59.7	2.89	61.6%	3.2%
Two-Pass	22.2	71.5	3.22	93.6%	11.1%
Combination	22.4	72.6	3.24	96.6%	12.1%

4.2.3 Experimental Results and Observations

Table 4.1 shows the average results when increase of route length of is bounded by 20% of the length of a shortest route, that is, an increase of 4 road segments. Each row of the table shows results for one algorithm. We have included average distance and total connectivity of the found path, connectivity per unit distance, and the percentage increase over the shortest path's distance and connectivity. For instance, the connectivity ratio algorithm generated an average connectivity of 71.8 with distance 22.3, corresponding to an overall connectivity per unit distance of 3.22. This is an increase of 94.5% in total connectivity of the path with a distance increase of only 11.4%, both relative to the fixed shortest path that simulates the most popular through way. The Combination row is the result of taking the best path generated from all of the algorithms on a given instance of the grid and with the corresponding path length bound. It is clear that each algorithm produces increased connectivity not only proportional to the allowed extra distance, but far beyond it as well. On average, taking the best route allows for nearly double the connectivity for a mere

12.1% increase in path distance. The connectivity increase over the best shortest path is well beyond the path length increase as well.

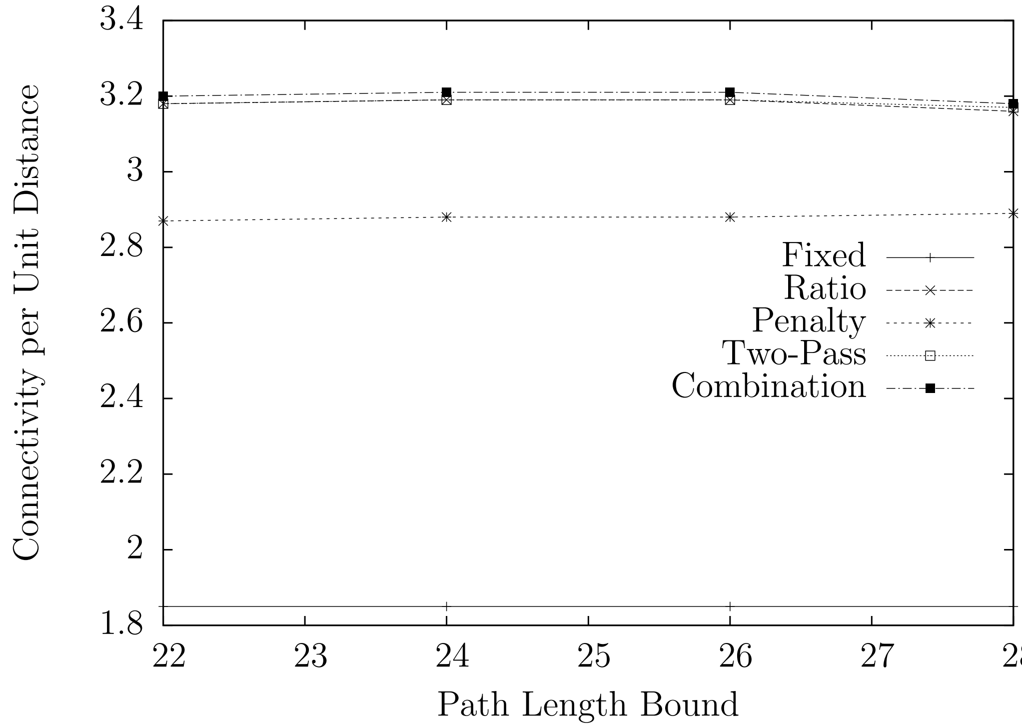

Figure 4.5: Connectivity per Unit Distance vs Path Length Allowed with 45 Hotspots and Hotspot Connection Capacity of 5

Instead of presenting our observations for other bounded path lengths in three tables we illustrate observed results as plots in Figures 4.5 and 4.6 for visual comparison. From the plots in Figure 4.5, we can see that per unit-distance communication-capacity for an algorithm is almost constant for all allowed path-length increases. The plots in Figure 4.6 show that on an average `penalty` algorithm gives smallest improvement, while `combination` algorithm gives the largest improvement (about 20%).

The results for the set of 50 hotspots are similar to that of 45 hotspots. We observed slight increase in normalized communication-capacity compared to the set

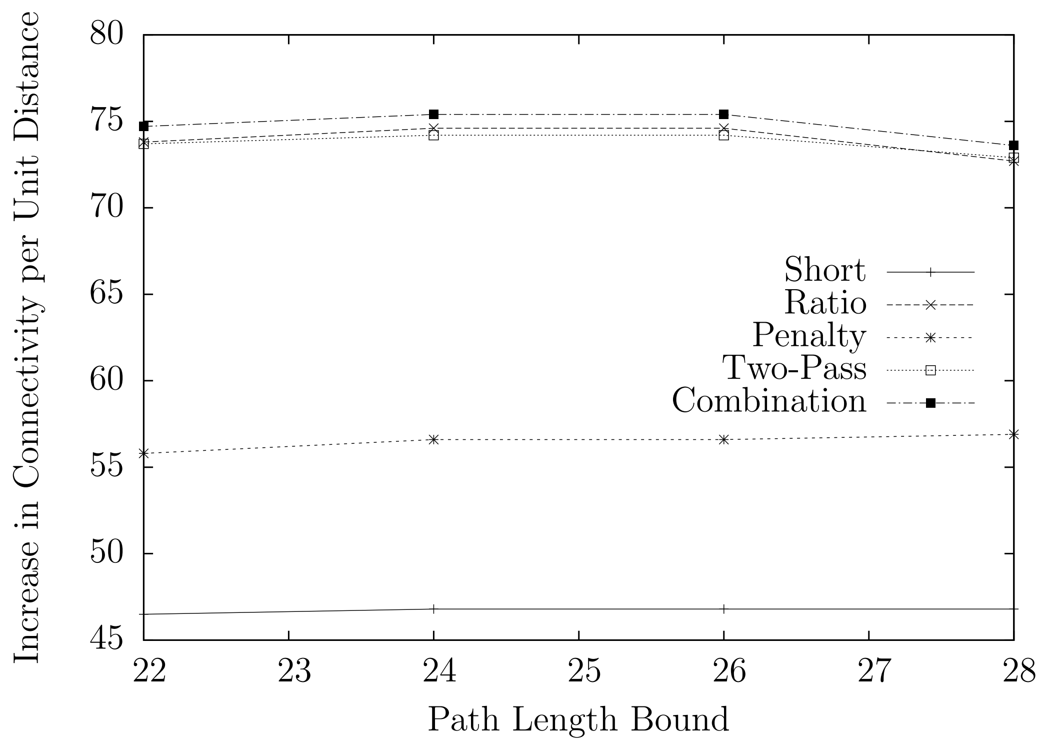

Figure 4.6: Percent Improvement in Connectivity per Unit Distance vs Path Length Allowed with 45 Hotspots and Hotspot Connection Capacity of 5

of 45 hotspots. Also, as can be seen from the plots in Figure 4.7, the improvements over the shortest paths' performance are slightly better. Figure 4.8 plots a sample of percent distance increases along with percent connectivity increases, and clearly shows that each algorithm gives consistent results where the connectivity increases outpace the distance increases. The path length increase is 12% while the connection capacity increases by 33%.

Running all of the algorithms simultaneously and choosing the best result allows us to utilize the strengths of each to offset the situations of failure for others. As seen in Figures 4.6 and 4.7, combination of algorithms improve performance by about 1% over the best algorithm.

In order to determine the impact of the relative connection capacity of road seg-

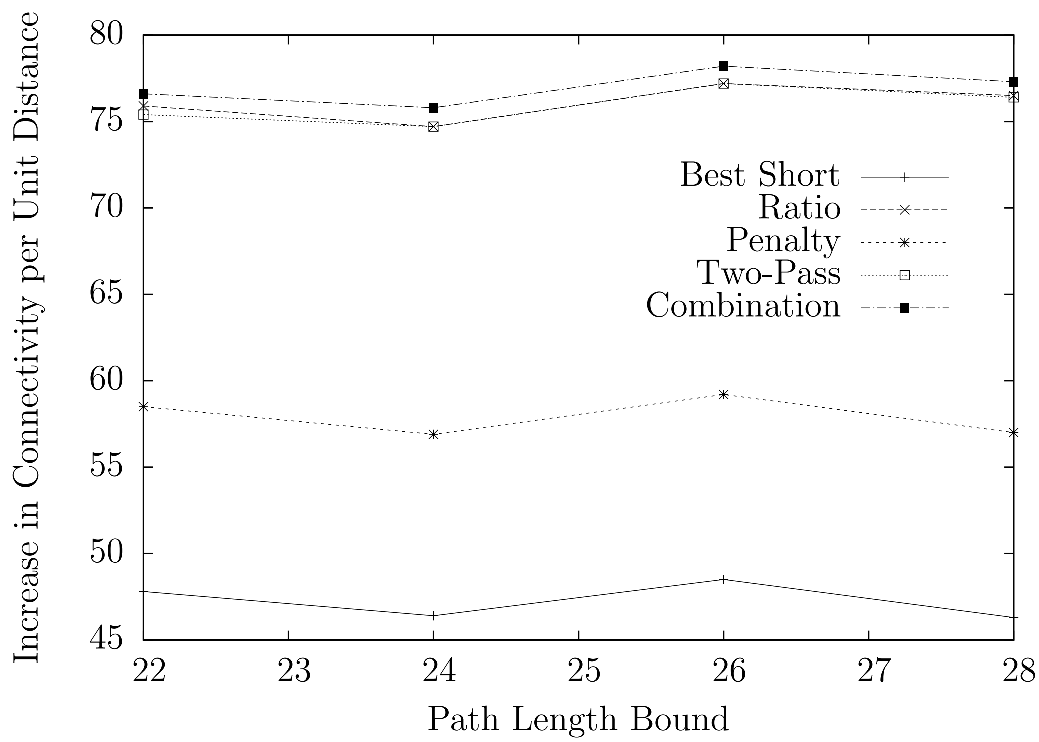

Figure 4.7: Percent Improvement in Connectivity per Distance vs Path Length Allowed with 50 Hotspots and Hotspot Connection Capacity of 5

ments with hotspot access, we then vary that value from 5 to the lower values of 2, 3, and 4. This has the obvious effect of lowering the overall connection capacity of any path passing through intersections with hotspot access, but lets us examine how it affects the improvement in connection capacity for algorithms compared to the fixed shortest path.

We can see in Figure 4.9 that there is a drop in the percent increase in overall connection capacity per distance from around 96% to only around 21%. This should be expected as the gain from utilizing a single extra hotspot accessible road segment is much lower. However, the improvement still exists. By utilizing the data from varying hotspot capacity, we plot the values for increase in distance and increase in connectivity against the hotspot capacity with fixed path bound and number of

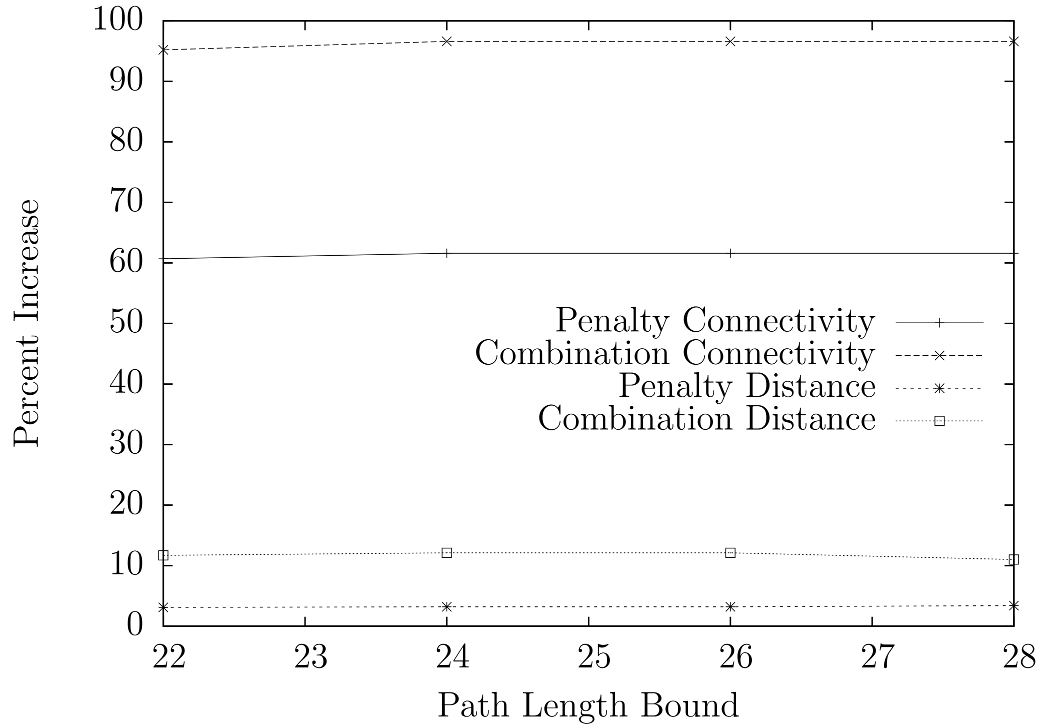

Figure 4.8: Percent Improvement in Connectivity and Percent Increase in Distance vs Path Length Allowed with 45 Hotspots for Two-Pass and Combination Algorithms with Hotspot Connection Capacity of 5

hotspots. Figure 4.10 is representative for trends in all of the data collected, including varying the algorithm used, number of hotspots, and path length bound. From the plot it is clear that as hotspot capacity increases, both distance and connectivity increase further beyond that of the shortest path. The increases in connectivity are faster than the increase in path length, meaning that a higher average connectivity increases the effectiveness of the algorithms.

In order to better determine the effects of hotspot capacity, we compare the increases in distance and connection capacity using a fixed path bound and number of hotspots. We can see from the plot that increasing the relative connection speed is multiplicative of the distance increase.

Table 4.2: Performance Evaluation on 20x20 Road Network with 45 Hotspots and 20% path length increase allowed. Hotspot Connection Capacity of 2

Algorithm	Average Distance	Average Connectivity	Average Unit Distance Connectivity	Avg Connectivity Increase	Average Distance Increase
Fixed	20.0	24.2	1.21	0.0%	0.0%
Best Shortest	20.0	28.6	1.43	17.9%	0.0%
Ratio	20.4	29.9	1.47	23.2%	1.9%
Penalty	20.2	29.3	1.45	20.9%	1.1%
Two-Pass	20.4	29.9	1.47	23.2%	1.9%
Combination	20.4	30.0	1.47	23.6%	2.0%

4.3 Algorithm Performance Comparison and Discussion

Now, we are able to evaluate the algorithms with respect to both experimental data and contrived failure situations from Section 6.1. While in real world applications, the distribution of hotspots or cell towers is far from random, we can still make valid deductions from the data presented. From the entirety of the experimental data set, we can see that the ratio and two-pass algorithms perform very similarly. This seems to be caused by the random distribution of high connection capacity road segments, as opposed to organized distribution that more likely may resemble situations such as that in Figure 4.2.

With a situation such as in Figure 4.2, where each road parallel to the ideal path has increasing connection capacity, the limits of the Ratio algorithm are much more apparent. Depending on the relative connection capacity of those side roads, the Ratio algorithm may be unable to utilize the entirety of the allowed distance increase. This

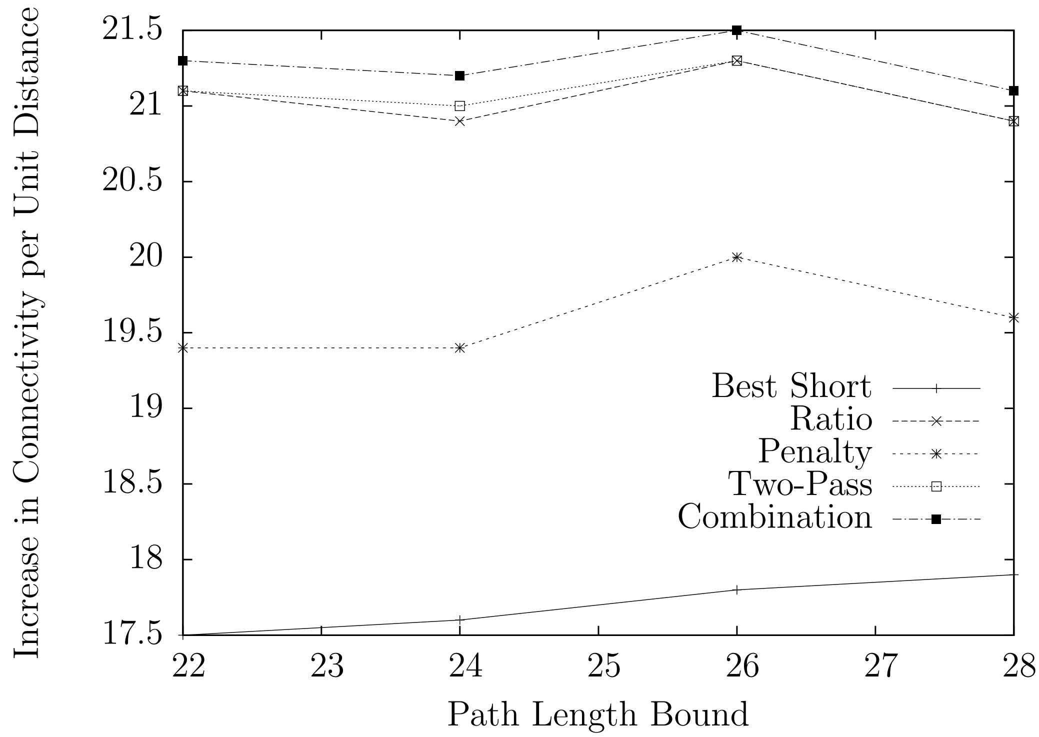

Figure 4.9: Percent Improvement in Connectivity per Unit Distance vs Path Length Allowed with 45 Hotspots and Hotspot Connection Capacity of 2

is precisely where the Two-Pass algorithm shines. There is ostensibly no upper limit to the allowed path length that the Two-Pass algorithm can utilize.

While it seems to be rare that the Penalty algorithm finds the ideal path, the algorithm covers a not insignificant case. This accounts for the results of the Combination algorithm over each individual algorithm.

Finally, it is clear from the data that each of the algorithms perform the intended function of adding connection capacity to a route at a rate higher than the distance increase. The actual value of the increase depends on the difference between low and high connectivity road segments, but in most cases a significant increase can be made.

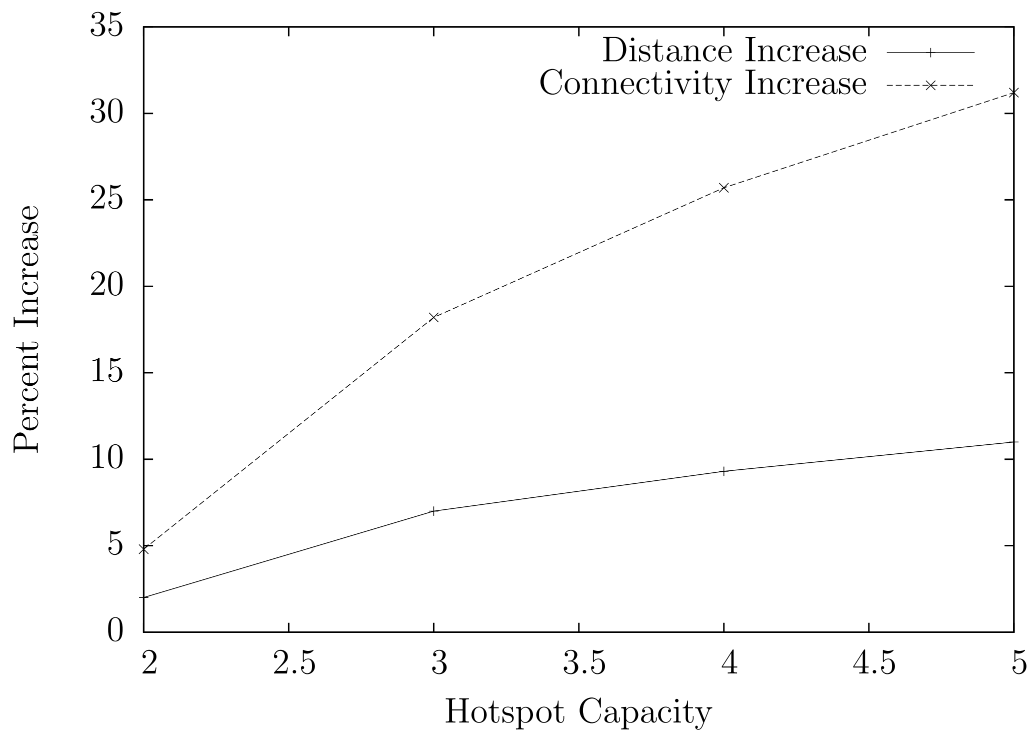

Figure 4.10: Percent Improvement in Connectivity and Percent Improvement in Distance vs Hotspot Capacity for Combination Algorithm with 45 Hotspots and Path Bound of 28

Chapter 5

Summary & Future Work

In this paper we have proposed and evaluated three algorithms for selecting driving routes between two points on a road network. Unlike standard GPS systems that select shortest or fastest driving routes, the proposed algorithms are for selecting routes that maximize wireless connection capacity while length of the selected routes are bounded by a given length. It is well known that bounded length path selection problem is NP-hard and hence a polynomial time algorithm is unlikely to exist.

Proposed algorithms may not find routes that have maximum wireless connection capacity, but results from extensive evaluations have shown that the communication capacity gain is higher than the route length increase. For instance, when distance increase was bounded by 20%, on an average path selected by one algorithm was 11.4% longer than the length of the shortest path but connection capacity was about 32.5% higher than that of all shortest paths.

Solutions to this problem have many practical applications. For instance, one can select a driving route between two points such that the total driving time is no more than a given bound but allows the complete upload of a video to YouTube. Another application would be distribution of automobile traffic in congested urban streets. Making higher speed broadband available on less traveled streets will divert traffic from most congested roads.

Application of the algorithms to real world data by connecting to sources such as opensignal.com to provide connection capacity would allow for practical utilization of the algorithms. It should be possible to connect this data together with road network data and embed the algorithms into GPS guidance systems.

References

[1] Y. Sudo, G. Motoyoshi, T. Murase, and T. Masuzawa, “Optimal longcut route selection for wireless mobile users (in japanese).” IEICE technical report, RCS2009269, 2010.

[2] Z. Wang and J. Crowcroft, “Quality of service routing for supporting multimedia applications,” *IEEE Journal of Selected Areas in Communications*, vol. 14, pp. 1228–1234, 1996.

[3] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, “Femtocell networks: A survey,” *IEEE Communications Magazine*, pp. 59–67, 2008.

[4] I.-T. Lin, D. Sarkar, T. Murase, and I. Sasase, “Dijkstra-based higher capacity route selection algorithm using bounded length and state change for automobiles,” in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, pp. 1–5, May 2012.

[5] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd ed., 2001.