Article

A parallel processing model for accelerating high-resolution geo-spatial accessibility analysis: A case study for poverty alleviation in Xiangxi

Mengyu Ma, Ye Wu*, Ning Guo, Luo Chen, Qi Gong and Jun Li

College of Electronic Science, National University of Defense Technology, Changsha 410073, China; mamengyu10@nudt.edu.cn(M.M.); guoning10@nudt.edu.cn (N.G.); luochen@nudt.edu.cn(L.C.); gongqi12@nudt.edu.cn(Q.G.); junli@nudt.edu.cn(J.L.)

* Correspondence: yewugfkd@nudt.edu.cn; Tel.: +86-137-5518-6456

Version May 16, 2018 submitted to MDPI

Abstract: Accessibility is an important issue in transport geography, land planning and many other related fields. Accessibility problems become computationally demanding when involving high-resolution requirements. Using conventional methods, providing high-resolution accessibility analysis for real time decision support remains a challenge. In this paper, we present a parallel processing model, named HiAccess, to solve the high-resolution accessibility analysis problems in real time. One feature of HiAccess is a fast road network construction method, in which the road network topology is determined by traversing the original road nodes only once. The parallel strategies of HiAccess are fully optimized through theoretical analysis and experimental comparisons. Moreover, map generalization techniques are adopted to reduce computation load without accuracy loss. The flexibility of HiAccess enables it to work well when applied to different accessibility analysis models. To further demonstrate the applicability of HiAccess, a case study of settlement sites selection for poverty alleviation in Xiangxi, Central China, is carried out. The accessibility of jobs, health care, educational resources and other public facilities are comprehensively analyzed for settlement sites selection. HiAccess demonstrates the striking performance of measuring high-resolution (using $100m \times 100m$ grids) accessibility of a city (in total over 250k grids, roads with 232k segments, 40 facilities) in 1 second without preprocessing, while ArcGIS takes nearly 1 hour to achieve a less satisfactory result. In additional experiments, HiAccess is tested on much larger datasets with excellent performance.

Keywords: geo-spatial accessibility; parallel computing; spatial indexes; road network construction; map generalization

---

1. Introduction

Since the 1950s, accessibility has been a core concept in transport geography, land planning and many other related fields[1–3]. It can be broadly defined as the ease with which activity locations or services can be reached from a particular location using the available transportation modes[4–6]. In recent years, accessibility is increasingly applied to numerous applications, such as evaluating transportation systems performance[7–9], investigating urban development[10–12], appraising access to jobs or health care[13–15], and estimating the sustainability of ecosystems[16]. In order to apply the concept of accessibility in these applications, it is necessary to introduce measures of accessibility.

With the widespread use of accessibility, different measurements have been developed for various analytical and evaluative purposes. Roughly, they can be classified into five categories [17–19]:

1. Distance-based measures use the distance from the original point to the desirable destinations. The distance can be average distance, weighted distance or closest distance[20].

2. Opportunity-based measures evaluate the accessibility based upon the amount of opportunities and destinations reachable within a specified travel distance, time or cost from the original point [14,21–25].
3. Gravity-type measures evaluate accessibility based on the potential interaction between locations[26,27].
4. Utility-based measures use the economic concepts of utility theory and consumer surplus to measure accessibility [28,29].
5. Space-time measures emphasize the range and frequency of the activities in which a person can participate within given spatio-temporal constraints[8,30–32].

With the development of geographic information technology, the computer's capability in spatial data management and spatial problem-solving is becoming more and more powerful. Geographic Information Systems (GIS) provides convenient and efficient accessibility analysis tools. The ArcGIS Network Analyst extension is one of the examples, which allows users to create a network dataset and perform accessibility analysis [7,12,33,34]. However, due to the inflexibility of existing GIS tools, it is difficult to apply complex accessibility analysis models to these tools. Thus, some researchers develop their own GIS extensions[15,17,23,35,36]. In the transportation industry, some professional GIS software, such as TRANSCAD or EMME, also provide tools for accessibility analysis. However, these tools are designed only for transportation planners and are inappropriate to be used for other purposes. Spatial indexes, used by spatial databases, can be used to optimize accessibility analysis. PostGIS is an open source software program that adds support for geographic objects to the PostgreSQL object-relational database. PgRouting extends PostgreSQL to provide geospatial routing functionality. A few studies implement accessibility analysis based on PostgreSQL and its extensions [37,38]. Still, it is inconvenient and cumbersome to install the huge GIS software or spatial databases to achieve accessibility analysis function. To eliminate the software, recent researchers have developed some standalone accessibility analysis tools: High-Resolution Transit Accessibility (HRTA)[8], UrbanAccess[39], AccessMod5[40].

Unfortunately, accessibility problems become computationally demanding when they involve high-resolution analysis requirements. All the methods mentioned above do not efficiently perform high-resolution accessibility analysis. Limited by the weak performance, most previous studies measure accessibility aggregately at a rather coarse resolution of municipalities[41], counties[42], neighborhoods[23], rough grids (Papa and Bertolini ($1km \times 1km$ grids), Ala-Hulkko *et al.* ($2km \times 2km$ grids)), and, most commonly, the traffic analysis zones (TAZs)[7,9,14,25,35,43]. Typically, centroids of TAZs are assumed as the origin and destination locations, leading to a common phenomenon of discontinuous estimates (or patchwork) when evaluating two adjacent zones, and representing a TAZ of large size with a single point may result in huge errors. In HRTA, Benenson *et al.* measured accessibility at the resolution of individual buildings and constructed high-resolution accessibility maps for an entire metropolitan area with its 200k buildings. However, the building data required by HRTA may be difficult to obtain. And worse still, HRTA is not adequate for building-scarce areas. Blanchard and Waddell presented a new open source tool, UrbanAccess, which used a generalized and scalable methodology to measure transit accessibility over large metropolitan extents. UrbanAccess crudely uses the nodes in road networks as the locations for analysis, which will certainly produce poor results when roads are sparse. The standalone product offered by the World Health Organization (WHO), AccessMod5, has been widely used to measure physical accessibility and geographic coverage in many low and middle income countries including Rwanda, Namibia and Sao Tomé and Principe. AccessMod5 avoids time-consuming network analysis through rasterizing the roads and deploying accessibility analysis on rasterized grids. AccessMod5 is efficient except in some complex accessibility analysis models (e.g., multimodal accessibility models, models considering one and two-way roads, etc).

Along with the rapid development of the surveying and mapping technology, spatial data with higher accuracy and larger scale is produced, which provides the possibility and necessity for high-resolution accessibility analysis. Developments in parallel computing technologies provides a prerequisite for high-performance accessibility analysis. Parallel computation has been extensively applied for solving complex spatial problems[44], but not for accessibility, particularly when facing the challenge of high-resolution analysis. Cheng *et al.* offered an approach using parallel processing techniques to measure job accessibility across Greater Manchester[45]. However, the parallel computing strategies in this approach are not fully optimized and spatial indexing techniques are not utilized.

In this study, we present a parallel processing model based on Message Passing Interface (MPI) , HiAccess, to solve the high-resolution accessibility analysis problems in real time. In HiAccess, spatial indexes are used to efficiently build and manage the road network. Map generalization techniques are adopted to reduce computation load without loss of accuracy. Through continuous optimization, HiAccess eventually demonstrates the striking performance of measuring high-resolution (using $100m \times 100m$ grids) accessibility of a city (in total over 250k grids, roads with 232k segments, 40 facilities) in one second without preprocessing, which takes nearly 1 hour to achieve a less satisfactory result in ArcGIS using Network Analyst. Moreover, a much larger dataset is tested. HiAccess is used to measure the medical facilities accessibility of Beijing, China (4m grids, 1m segments, 1481 facilities), requiring only 10 seconds to produce the final result. HiAccess is proved to be flexible, easily applied to complex models, and is independent of cumbersome software such as GIS or spatial databases. The results of HiAccess can be easily exported to any GIS software. Our approach has been successfully applied to settlement sites selection for poverty alleviation in Xiangxi, Central China, which uses HiAccess to comprehensively analyze the accessibility of jobs, health care, educational resources and other public facilities.

The remainder of this paper proceeds as follows. The next section analyzes the computational complexity of accessibility problems. In Section 3, the road network construction processes, parallel computing strategies and other optimization techniques of HiAccess are described in detail. A case study of the application of HiAccess is demonstrated in Section 4. The performance of HiAccess on much larger datasets is tested in Section 5, with the conclusions of this paper being reserved for Section 6.

2. Computational complexity of high-resolution accessibility analysis

Generally, the input of accessibility analysis are road layers, facility layers and a number of locations to be analyzed. In this paper, we utilize grids as the locations for analysis. Our output is the accessibility of grids to the facilities stored as the well-known GeoTIFF format. The computational complexity of high-resolution accessibility analysis mainly embodies the following:

- Building road network from road layer

  The original road layer cannot directly be used for analysis due to the absence of overt topology information. Instead, it should be converted to a road network first. Essentially, building road networks is to determine the vertices and edges, which is time-consuming especially when finding intersections of roads. Meanwhile, it should be considered that a good road network must be concise and contain as few unnecessary vertices as possible. Unnecessary vertex refers to the vertex with a degree of two (See Figure 1).

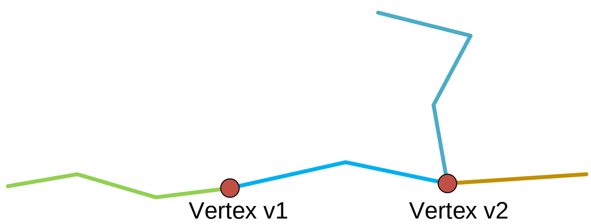


Figure 1. Unnecessary vertices (e.g. v1) and necessary vertices (e.g. v2) in a road network

- Measuring distance from grids to facilities

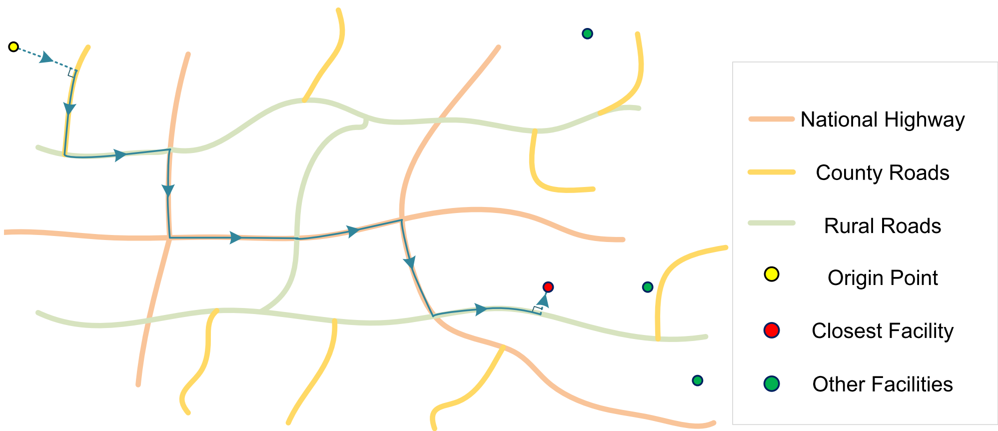
  Most points, representing grids or facilities, are not on the road. This results in the additional operations of finding points representing grids or facilities in road networks and measuring distance from the representative points to the original ones. The nearest points to grids or facilities in road networks should be chosen. However, the nearest point may not be the vertices of road networks, and in most cases, it is in the middle of a road segment, which will increase the complexity of road network analysis. After finding the representative points in road networks, the next step is a complicated shortest-path analysis in large road networks.
- Computational scale in high-resolution analysis

Obviously, the computational scale of accessibility analysis is proportional to the number of grids. When the resolution becomes n times, the number of grids changes to n^2 times the original. As a result, the computational scale becomes very large in the case of high-resolution accessibility analysis.

3. Methodology

This section focuses on the deployment of HiAccess using a distance-based measurement, namely the closest-distance measurement. Road network construction processes, parallel computing strategies and other optimization techniques of HiAccess are described in detail below. Closest-distance measurement, the most basic one, can be extended to other accessibility measurements. In the last part of this section, the closest-distance deployment of HiAccess is extended to opportunity-based and gravity-type measurements through minor changes, demonstrating the flexibility of HiAccess.

Broadly, the "distance" of closest-distance measurement can be time, money or any other indicators measuring the cost of a journey. Typically, HiAccess is used to measure accessibility by private cars. Compared with public transit, car-based accessibility requires a much larger computational scale due to the much larger size of road networks for private cars. Figure 2 illustrates the process of accessibility analysis for an original point. Different levels of roads are set with various speeds, and off-road travel is taken into account with the speed set to a relatively low value to enhance accuracy. The arrow lines show the quickest route to the closest facility with its cost as the analysis result of the original point.

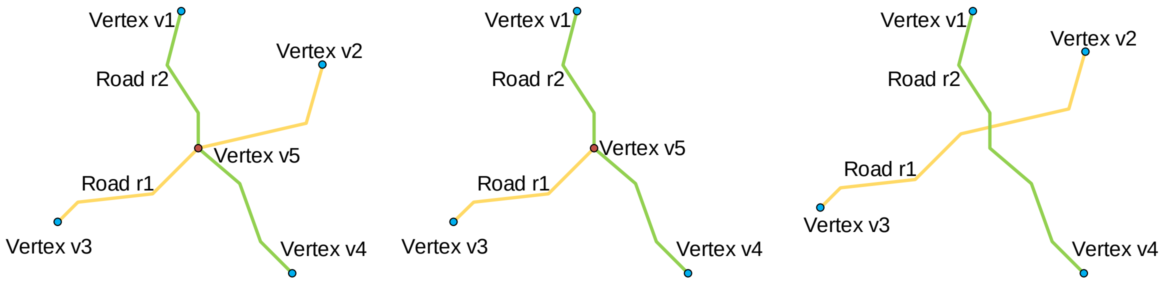


Road Level	National Highway	County Roads	Rural Roads	Off-road
Speed(km/h)	60	40	20	10

Figure 2. Accessibility analysis of the original point to destination facilities

3.1. Road network construction from road layers

A road layer is a dataset storing the road data. Roads in the layer are represented with linestrings, consisting of a series of two-point segments. A simple idea to build road networks is to use the ends of segments as vertices in networks (See UrbanAccess). However, there is one disadvantage to this method: most linestrings are made up of dozens or even hundreds of segments; therefore, using the ends of segments as vertices will cause a large number of unnecessary vertices (See Figure 1). A good road network should be concise and contain as few unnecessary vertices as possible. Another intuitive method is to use the two ends of each linestring as vertices and the linestrings as edges connecting the vertices. This method can effectively avoid unnecessary vertices. Even so, some necessary ones will also be omitted and it may lead to serious errors in road network construction. Figure 3 (a) and (b) show the situations where errors occur. In the situation of crossroads (See Figure 3 (a)), the key vertex connecting r1 and r2 (v5) will be omitted in error. In the T-intersection situation (See Figure 3 (b)), the vertices can be determined correctly, but still, regarding r2 as the edge connecting v1 and v4 will be a mistake (r2 is divided into two parts by v5, v1 and v4 are not connected directly in the road network). What needs to be further explained is that intersecting roads with no common node at the intersection (See Figure 3 (c)) are assumed to be disconnected in our approach. The PostgreSQL extension, pgRouting provides functionality to build road networks from road layers. pgRouting assumes that all intersecting roads are connected, but it is unsuitable in some complex cases (e.g. viaducts). In addition, in the process of normal GIS data production, road connections mostly bring about common nodes in road linestrings. Accordingly, our assumption is reasonable.



(a) Crossroads (b) T-intersection (c) Disjoint roads

Figure 3. Different situations of road intersections

Good road network construction methods should consider both speed and quality. To address the accessibility problems, we present an extended road network structure in which spatial indexes are introduced to improve analysis speed. Based on the structure, we propose a fast road network construction method named traverse-once-building (TOB). In TOB, the road network topology is efficiently determined by traversing the original road nodes only once. The topology built by TOB is succinct and accurate.

3.1.1. Road network structure for accessibility analysis

The extended road network structure in our approach consists of three parts (Table 1): network topology, edge attributes and spatial indexes. In graph theory and computer science, network topology is normally represented as an adjacency list or adjacency matrix[46]. As a typical kind of network, a road network has its own characteristic: the degree of each vertex is generally from 1 to 4, which is far less than the number of all vertices. For this reason, if the topology of a road network is stored as an adjacency matrix, the matrix will be sparse, leading to waste in storage and performance degradation in network analysis. An adjacency list is a collection of unordered lists. Each list describes the set of neighbors of a vertex in the network. An adjacency list is suitable for a sparse network; thus, we choose an adjacency list to store the network topology of roads. Edge attributes store the detailed information of edges needed by accessibility analysis. In order to reduce repeated computations, some redundant information (e.g. length of segments) is also included. Before road network analysis, it is necessary to map the off-road grids or facilities to the road networks. For this purpose, we build spatial indexes for road networks to find the nearest points to the off-road points in road networks. In reality, road networks may be disconnected and consist of multiple connected subgraphs. For such cases, we find the nearest points in each subgraph and implement a shortest-path analysis separately. R-tree is an efficient tree data structure widely used for spatial searching[47,48]. The C++ library Boost.Geometry provides high performance implementations of R-tree. Using the packing algorithm, it can achieve the performance of about 1.5m values inserts or 1.4m spatial queries per second [49]. In HiAccess, we use Boost.Geometry to build R-trees for road networks.

Table 1. The extended road network structure.

Network Topology	
NeighborCount[1]	Number of neighbor vertices. NeighborCount_i represents the neighbor number of Vertex_i (the ith vertex).
AdjacencyVertex[2]	AdjacencyVertex_i is the list of neighbor vertices of Vertex_i. $\text{AdjacencyVertex}_{i,j}$ represents the jth neighbor of Vertex_i.
AdjacencyDistance[2]	The distances to neighbor vertices. $\text{AdjacencyDistance}_{i,j}$ represents the distance from Vertex_i to $\text{AdjacencyVertex}_{i,j}$.
Edge Attributes	
EdgeHeadVertex[1]	EdgeHeadVertex_i represents the head vertex of Edge_i (the ith edge).
EdgeEndVertex[1]	EdgeEndVertex_i represents the end vertex of Edge_i.
EdgeLevel[1]	Road levels of edges.
EdgeNode[2]	Nodes (express as point(X,Y)) in edges. $\text{EdgeNode}_{i,j}$ represents the jth node in Edge_i.
EdgeSegmentLength[2]	Length of segments in edges. $\text{EdgeSegmentLength}_{i,j}$ represents the length of jth segment in Edge_i.
EdgeSegmentCount[1]	Number of segments in edges. $\text{EdgeSegmentCount}_i$ represents the segments number of Edge_i.
Spatial Indexes	
RtreeSubgraph[1]	Spatial indexes for road network. RtreeSubgraph_i is the index of the ith connected subgraph in road network. The value type of object stored in spatial indexes is tuple(segment(point(X_1,Y_1),point(X_2,Y_2)),EdgeID,SegmentID)

[1] One-dimensional array.
[2] Two-dimensional array.

3.1.2. Traverse-once-building road network construction method

The general idea of TOB method is as follows. The first step is determining the preliminary topology by roughly using the two ends of each road as vertices and the roads as edges connecting the vertices. In this process, we only need to traverse the first and last nodes of each road. However, the preliminary topology may contain serious errors (see Figure 3) and further processing is needed. Next, we traverse the interior nodes of roads and correct the preliminary topology using operations such as breaking edges and adding vertices. In TOB, spatial indexes are used to efficiently manage nodes and generate vertices in topology, and a concise and accurate road network topology is determined by traversing all the road nodes only once. Then, we fill in the attributes of the extended road network structure (Step 3). This step is simple and executes quickly. For the sake of brevity, roads are regarded as two-way roads by default. TOB can also be used to process one-way roads. The last step is constructing spatial indexes. As the road networks may be disconnected and consist of multiple connected subgraphs, we use the depth-first-search (DFS) method to determine the subgraphs and then construct spatial indexes for segments in each subgraph separately.

Algorithm: TOB road network construction

Input:

Roadlayers: road layers.

Tolerance: snapping tolerance of disconnected edges (in projection unit).

Output:

Network Topology, Edge Attributes and Spatial Indexes of road networks.

Step 1 (Initialisation) Initialize the output extended road network structure and process variables: spatial indexes RtreeVertex (value type pair(point(X,Y),VertexID)) and RtreeNode (value type tuple(point(X,Y),RoadID,NodeID)); arrays storing corresponding vertices of first and last nodes in original roads (RoadHead and RoadEnd); vertex and edge numbers (Vcount and Ecount); a list of priority queues storing the break nodes in roads (RoadBreak with item type pair(NodeID,VertexID) and item with lower NodeID has higher priority); flag array identifying the subgraphs for vertices (SubgraphFlag).

Step 2 (Determine network topology) First, traverse the two ends of each road, and then traverse the interior nodes. Through spatial index technology, the road network topology is determined by traversing all the nodes in roads once. The result topology information is stored in RoadHead, RoadEnd and RoadBreak.

```
Vcount ← 0
for i = 0 → ROADCOUNT(Roadlayers)−1 do
    for j in set(0,NODECOUNT(road_i)−1) do                     ▷ Traverse first and last nodes
        buffer ← BUFFER(node_{i,j}, Tolerance)
        vertexpair ← satisfying RtreeVertex.INTERSECT(buffer)
        if vertexpair is not null then
            tmp ← vertexpair.VertexID
            j < 1 ? RoadHead.APPEND(tmp) : RoadEnd.APPEND(tmp)
        else
            RtreeVertex.INSERT(pair(node_{i,j}, Vcount))
            j < 1 ? RoadHead.APPEND(Vcount) : RoadEnd.APPEND(Vcount)
            Vcount++
for i = 0 → ROADCOUNT(Roadlayers)−1 do
    for j = 1 → NODECOUNT(road_i)−2 do                          ▷ Traverse interior nodes
        buffer ← BUFFER(node_{i,j}, Tolerance)
        vertexpair ← satisfying RtreeVertex.INTERSECT(buffer)
        if vertexpair is not null then
            tmp ← vertexpair.VertexID
            RoadBreak_i.PUSH(pair(j, tmp))
        else
            nodetuple ← satisfying RtreeNode.INTERSECT(buffer)
            if nodetuple is not null then
                RtreeVertex.INSERT(pair(node_{i,j}, Vcount))
                RoadBreak_i.PUSH(pair(j, Vcount))
                tmp1 ← nodetuple.RoadID , tmp2 ← nodetuple.NodeID
                RoadBreak_{tmp1}.PUSH(pair(tmp2, Vcount))
                Vcount++
            else
                RtreeNode.INSERT(tuple(node_{i,j},i, j))
```

Step 3 (Fill in attributes) According to the road network topology and the original road data, fill in attributes of the extended road network structure. Roads are regarded as two-way roads by default.

```
Ecount ← 0
for i = 0 → ROADCOUNT(Roadlayers)−1 do
    hvid ← RoadHead_i , evid ← RoadEnd_i
    nc ← NODECOUNT(road_i)
    RoadBreak_i.PUSH(pair(nc − 1, evid))
    head ← pair(0, hvid)
    for j = 0 → PAIRCOUNT(RoadBreak_i)−1 do
        end ← RoadBreak_j.POP                                   ▷ Items with lower NodeID pop first
        hvid ← head.VertexID , evid ← end.VertexID
        hnid ← head.NodeID , enid ← end.NodeID
        EdgeHeadVertex_{Ecount} ← hvid , EdgeEndVertex_{Ecount} ← evid
        EdgeLevel_{Ecount} ← LEVEL(road_i)
        AdjacencyVertex_{hvid}.APPEND(evid) , NeighborCount_{hvid}++
        AdjacencyVertex_{evid}.APPEND(hvid) , NeighborCount_{evid}++
        dist ← 0 , EdgeNode_{Ecount}.APPEND(node_{hnid})
        for k = 1 → enid − hnid + 1 do
            EdgeNode_{Ecount}.APPEND(node_{k+hnid})
            tmp ← DISTANCE(EdgeNode_{Ecount,k−1},EdgeNode_{Ecount,k})
            EdgeSegmentLength_{Ecount}.APPEND(tmp) , dist ← dist + tmp
            EdgeSegmentCount_{Ecount}++
        AdjacencyDistance_{hvid} ← dist , AdjacencyDistance_{evid} ← dist
        Ecount++ , head ← end
```

Step 4 (Construct spatial indexes) Determine the connected subgraphs in road network based on the depth-first-search method and construct spatial indexes for segments in road network.

```
SubgraphFlag ← DFSSEARCH(AdjacencyVertex)
for i = 0 → Ecount − 1 do
    tmp ← EdgeHeadVertex_i
    gid ← SubgraphFlag_{tmp}
    for j = 0 → EdgeSegmentCount_i−1 do
        segment ← pair(EdgeNode_{i,j}, EdgeNode_{i,j+1})
        RtreeSubgraph_{gid}.INSERT(tuple(segment, i, j))
return
```

3.2. Parallel computing strategies for accessibility analysis

In this paper, we utilize grids as the locations for accessibility analysis. Following road network construction is the complicated operation of measuring the accessibility from grids to facilities. The scale of computation is larger in high-resolution cases, as the computational scale is proportional to the square of resolution. In order to solve high-resolution accessibility problems in real time, it is necessary to accelerate analysis by parallel computing technologies. A simple parallel computing strategy for accessibility analysis is to regard each grid as an independent task and assign all tasks to the processes equally with the tasks executed one by one in each process. However, the strategy is not effective because neighboring grids may share same route to facilities (see Figure 4) and processing each grid separately causes repeated computations, which can affect the capability of parallel processing. However, it does not mean that repeated computations are not recommended; sometimes the cost of communication between processes is higher than the computation cost. Good parallel strategies should consider both computation and communication[50]. In HiAccess, we designed the strategies through theoretical analysis and experimental comparison. The final parallel computing strategies are shown in Figure 5.

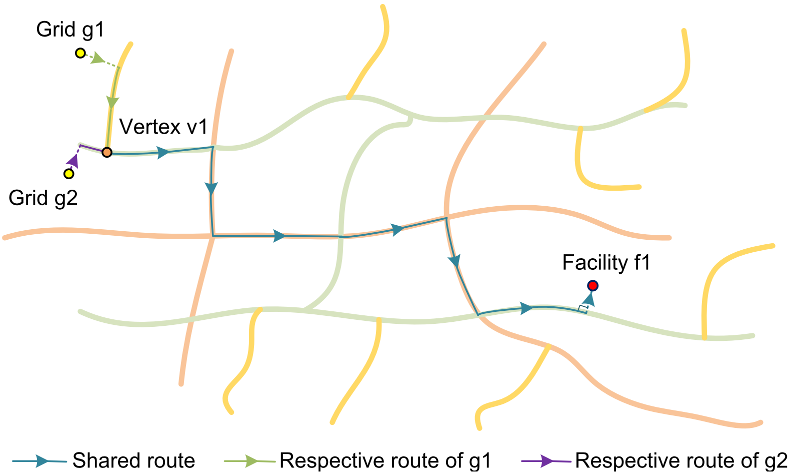


Figure 4. Example of neighboring grids sharing same route to facilities

Step 1 (Build road network) In HiAccess, the road network is built in each process separately. It means that the road network is neither built in parallel nor built by only one master process (other processes access the road network through communication). The reasons for using this strategy are as follows: a) It is difficult to separate the road network construction into parallel tasks due to the massive dependencies; b) Using TOB, the extended road network can be constructed rapidly, and there is no urgent need to parallel; c) The competition for system resources among processes is negligible. Consequently, building the road network in multi processes simultaneously is not slower than in a single process; on the contrary, it is even faster. This is because multi processes accelerate the speed of reading road layer files from the disk.

Step 2 (Measure accessibility of vertices in network) As shown in Figure 4, v1 is the first vertex passed in the route from g1 (or g2) to f1; after v1, g1 and g2 share the same route to f1. Measuring the accessibility of g1 (or g2) to f1 can be divided into measuring the shared cost from v1 to f1, which can be calculated together, and measuring the respective cost from g1 (or g2) to v1. Actually, in high-resolution accessibility problems, each vertex in the network is the first vertex passed for many more than just two grids to the closest facilities. In order to avoid repeated computations, we first measure accessibility of all vertices in the network and then measure the cost from grids to the corresponding vertices, greatly reducing the amount of calculation. Measuring accessibility of all vertices in the road network needs to measure the cost from vertices to each facility. This is a series of single-source shortest-path problems (regard each facility as a source). Dijkstra algorithm provides an efficient solution for these problems. The implementation of it based on a min-priority queue implemented by a Fibonacci heap and running in $O((|E| + |V|) \log |V|)$(where $|V|$ is the number of nodes and $|E|$ is the number of edges) is asymptotically the fastest known single-source shortest-path algorithm for arbitrary directed graphs with unbounded non-negative weights [46]. In HiAccess, we regard measuring the cost from all vertices to each facility as an independent task and process the tasks in parallel. As the road network may be disconnected, in order to measure the accessibility for all vertices in the road network, we need to analyze accessibility in each subgraph respectively. The process of measuring the cost from vertices in a subgraph to a specific facility is shown in Figure 6.

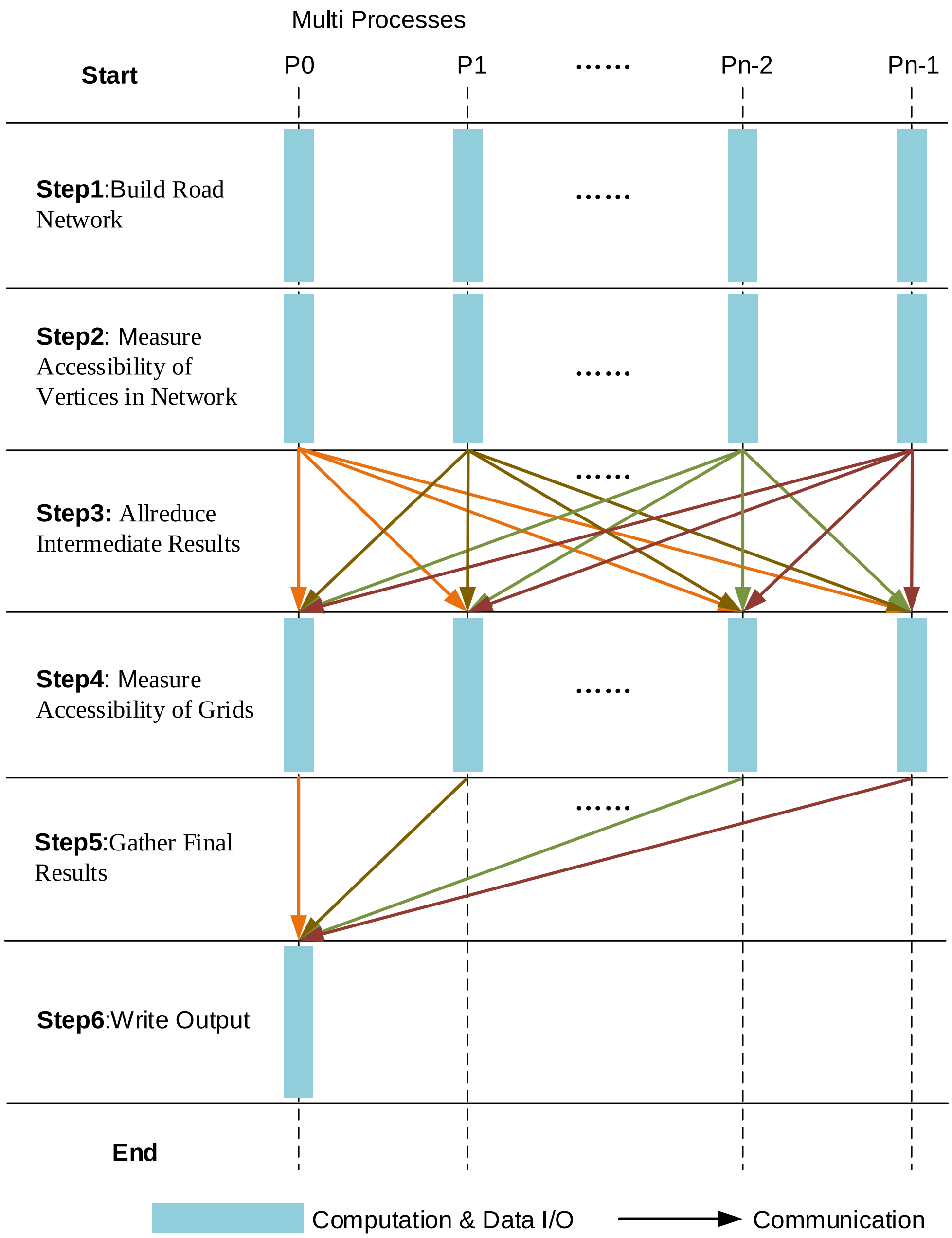


Figure 5. Parallel accessibility analysis flow of HiAccess based on MPI

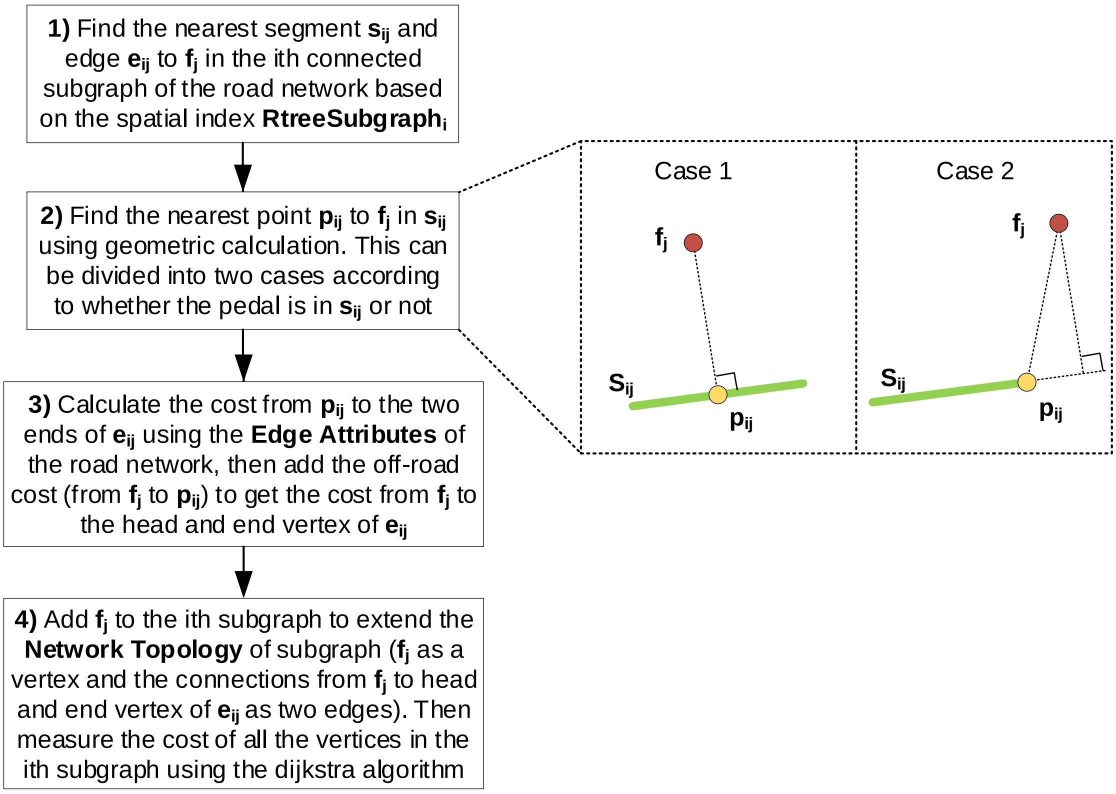


Figure 6. Measure the cost from vertices in the ith subgraph to facility f_j

Step 3 (Allreduce intermediate results) In the previous step, we regard measuring the cost from all vertices to a facility as an independent task and assign all the facilities to the processes; namely, each process only measures the cost from all vertices to the assigned facilities. Using the closest-distance measurement, it is necessary to find the minimum cost to facilities for each vertex. In each process, the results of minimum cost are stored in the array $\mathsf{VertexCosttmp}$ with the length equal to the number of vertices ($\mathsf{VertexCosttmp}_i$ represents the minimum cost to the assigned facilities from Vertex_i). Merge the results from each process to get the final minimum cost to all the facilities ($\mathsf{VertexCost}$).This is implemented using MPI_Allreduce with the MPI_MIN reduction operation (see Figure 7). As a collective communication method in MPI, MPI_Allreduce is more efficient compared with the point-to-point communication[50].

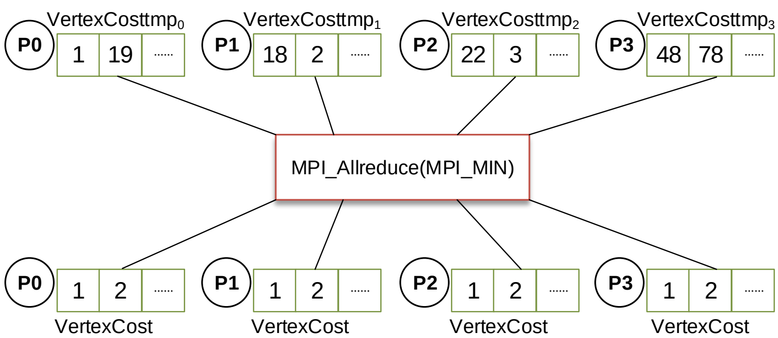
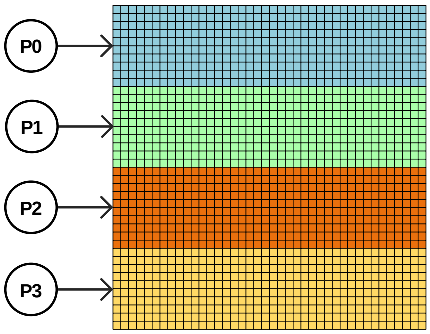


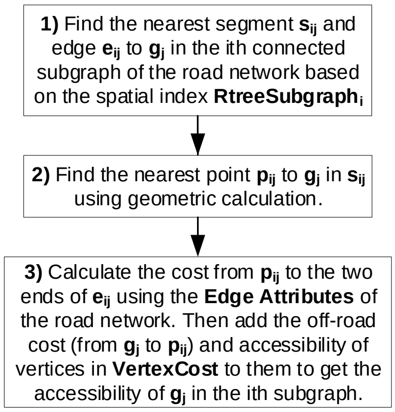
Figure 7. Allreduce intermediate results in HiAccess (4 processes)

Step 4 (Measure accessibility of grids) In this step, we measure the cost from grids to the corresponding vertices, then determine the accessibility of grids to facilities by adding the accessibility of vertices in $\mathsf{VertexCost}$.

As for parallel strategy, we divide all the grids by row and assign them to the processes equally (Figure 8 (a)). As the road network may be disconnected, it is necessary to analyze accessibility in each subgraph respectively to get the minimum cost from grids to all the facilities. The process of a grid in a subgraph is shown in Figure 8 (b).



(a) Parallel strategy with 4 processes



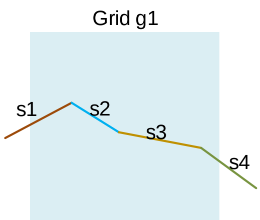
(b) Measure the accessibility of grid g_j in the ith subgraph

Figure 8. Measure accessibility of grids

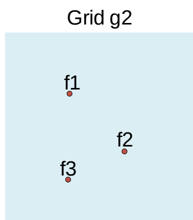
Step 5 and Step 6 (Gather and write final results) In step 5, the master process P0 gathers results from other processes using MPI_Gather. In step 6, P0 writes the final results to disk. Typically, the well-known GeoTIFF is chosen as the output format. Another possible strategy is that each process writes results to disk in parallel without gathering data from other processes. However, experiments show that writing results in parallel has better performance only when the output is large (over 1 GB) and has poor performance when the output is small. Normally, the output GeoTIFF of high-resolution accessibility is not larger than 100MB. So in HiAccess, we abandon the strategy of writing in parallel.

3.3. Computation reduction using map generalization techniques

Using the methods mentioned above, we are able to rapidly solve high-resolution accessibility problems. In the deployment of HiAccess, we find that some grids overlap with more than one segment or facility. As Figure 9 shows, g1 overlaps with four segments while g2 overlaps with three facilities. For such cases, we will build four segments for s1, s2, s3, s4 in the road network and measure the cost to f1, f2, f3 respectively. Given that the precision of HiAccess cannot be higher than the size of a grid, measuring accessibility with all details from the original data will not acquire higher accuracy. This means that we can simplify the original data to reduce computation without sacrificing measurement accuracy.



(a) Segments s1, s2, s3 and s4



(b) facilities f1, f2 and f3

Figure 9. Example of a grid overlaping with multiple segments or facilities

Map generalization is the name of the process that simplifies the representation of geographical data[51]. Usually, the purpose of map generalization is to produce a map at a certain scale with a defined and readable legend. In HiAccess, we use map generalization techniques to reduce accessibility analysis computation by simplifying the segments in road networks and facilities. Map generalization has been developed for decades, and researchers have presented many mature algorithms, such as K-means[52], Settle-spacing ratio algorithm[53], Douglas-Peucker algorithm[54] and Li-Openshaw algorithm[55]. Algorithms with better simplification results are usually with higher computational complexity. Thus, it is necessary to select proper simplification methods based on the specific application scenarios.

In HiAccess, the purpose of map generalization is to reduce computation without sacrificing measurement accuracy. To this end, the number of segments and points after simplification should be as few as possible. Meanwhile, the simplification process should not be computing expensive. In addition, the simplification of segments should not change the topology framework of road networks. The methods we used in HiAccess are as follows:

1. (Segments in road networks) For each edge in the road network, traverse the nodes, select the node of which the distance to the previous selected node is longer than the resolution of grids. Specifically, we adjust Step 3 of TOB algorithm (line 15-20) to the following segments simplification process. The simplification operations are efficient (only adding some condition judgments), and it will not affect the topology of road networks since the topology is determined in Step 1 of TOB.
2. (Facilities expressed as points) Cluster the facilities intersecting with the same grids into groups, and use the center point of gravity to represent the facilities in each group.

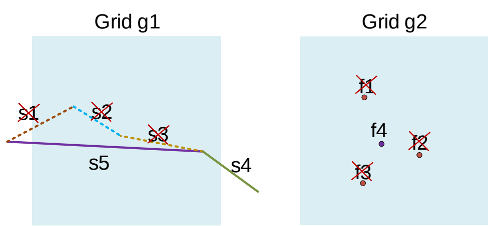
Process: Segments simplification

......

Step 3 (Fill in attributes)

......
$dist \leftarrow 0$, $EdgeNode_{Ecount}.\text{APPEND}(node_{hnid})$
$ktmp \leftarrow 1$
for $k = 1 \rightarrow enid - hnid + 1$ **do**
    $tmp \leftarrow \text{DISTANCE}(EdgeNode_{Ecount,ktmp-1}, node_{k+hnid})$
    if $tmp >= GridSize$ or $k == enid - hnid + 1$ **then** ▷ Added condition judgments
        $EdgeNode_{Ecount}.\text{APPEND}(node_{k+hnid})$
        $EdgeSegmentLength_{Ecount}.\text{APPEND}(tmp)$, $dist \leftarrow dist + tmp$
        $EdgeSegmentCount_{Ecount}{+}{+}$
        $ktmp{+}{+}$
$AdjacencyDistance_{hvid} \leftarrow dist$, $AdjacencyDistance_{evid} \leftarrow dist$
......

Figure 10 shows the simplification of segments or facilities intersecting with a grid. The simplification operations in HiAccess are simple, efficient and highly effective. In the process of measuring the medical facilities accessibility of Beijing, China, the problem scale is changed from 1.13m segments, 1481 facilities to 0.37m segments, 1022 facilities with grids at the size of 100 meters. The computing time is reduced to two-thirds of the original, with the accuracy unchanged.



(a) Replace s1, s2, s3 with s5 (b) Replace f1, f2, f3 with f4

Figure 10. Example of segments and facilities simplification

3.4. Scalability and flexibility of HiAccess

In this section, we extend the closest-distance deployment of HiAccess to opportunity-based and gravity-type measurements through minor changes, demonstrating the high degree of flexibility of HiAccess. Opportunity-based measures evaluate accessibility based on the amount of opportunities and destinations reachable within a distance from a point of origin. It can be expressed as:

$$A_i = \begin{cases} \sum_j M_j & D_{ij} \leq L \\ 0 & D_{ij} > L \end{cases}$$

where A_i is the accessibility of origin i, M_j is the attractiveness at destination j for a given set of opportunities, D_{ij} is the distance from i to j, and L is the given distance limit. Gravity-type measures evaluate accessibility based on the potential interaction between locations and can be written as follows:

$$A_i = \sum_j M_j \mathrm{f}\left(D_{ij}\right)$$

$f(D_{ij})$ is the interaction function. As shown in the formulas, to measure the opportunity-based or gravity-type accessibility of a grid, it is necessary to synthesize the distances from the grid to all the destination facilities, not only to the closest facility in closest-distance model.

To adapt to the new models, we make slight adjustments to the parallel accessibility analysis of HiAccess in steps 3 and 4. For step 3, in each process we replace VertexCosttmp with a two-dimensional array VertexCostListtmp. $\text{VertexCostListtmp}_{ij}$ represents the cost to the ith assigned facility from Vertex_j. As shown in Figure 11, we use the collective communication method MPI_Allgather to gather the cost from all vertices to all facilities (VertexCostList). $\text{VertexCostList}_{ij}$ represents the cost from Vertex_j to Facility_i. Then we can use VertexCostList to calculate the cost from a grid to all the destination facilities and measure the opportunity-based or gravity-type accessibility of the grid in step 4.

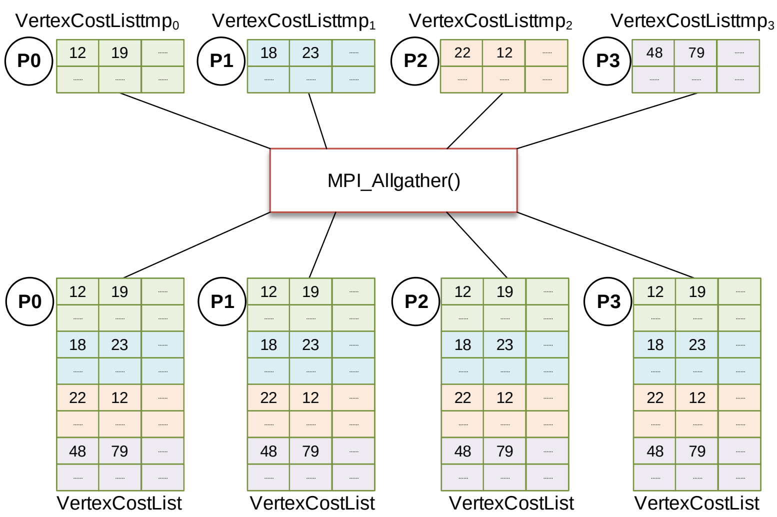


Figure 11. All gather intermediate results in HiAccess (4 processes)

Additionally, although it is difficult to adapt HiAccess to some more complex accessibility models, e.g. utility-base or space-time models, our methods give some useful and meaningful references.

4. Case study: settlement sites selection for poverty alleviation in Xiangxi, Central China

4.1. Background

In China, after the implementation of reformation and opening, great progress has been made in the construction of the economy and the development of society. But at the same time, the income gap has continued to widen and there are still millions of people living in poverty. The Chinese government is developing a sound national poverty alleviation system to help lift poor people out of poverty. In November 3, 2013, Xi Jinping, the President of China, visited Xiangxi and took it as the key area for poverty alleviation. Xiangxi is short for Tujiazu Miaozu Autonomous Prefecture of Xiangxi, which is located $109.10° \sim 110.28°$ E, $27.44° \sim 29.38°$ N in Hunan Province in China. It consists of 8 counties which are all under the poverty line. Most of the poor people in Xiangxi are living in mountainous areas with inconvenient traffic conditions, a high rate of unemployment, out-of-date techniques, and information block, etc.

Poverty alleviation in Xiangxi begins with selecting settlement sites for the poor. Settlement sites selection comprehensively analyzes the accessibility of jobs, health care, educational resources and other public facilities. It is a series of accessibility analysis problems. To achieve a better result, high-resolution analysis is required. As it takes nearly 1 hour to measure the high-resolution ($100m \times 100m$ grids) accessibility to hospitals for Huayuan county in Xiangxi using ArcGIS Network Analyst, it may take a week for a skilled ArcGIS operator to get high-resolution accessibility to all kinds of facilities for all the counties in Xiangxi at once, which is obviously infeasible. In addition, the accessibility results are also used for planning of new facilities and roads, leading to more accessibility analysis requests. Our approach has been successfully applied to settlement sites selection for poverty alleviation in Xiangxi. Using HiAccess, we are able to solve the problems of high-resolution accessibility analysis of a county in real time.

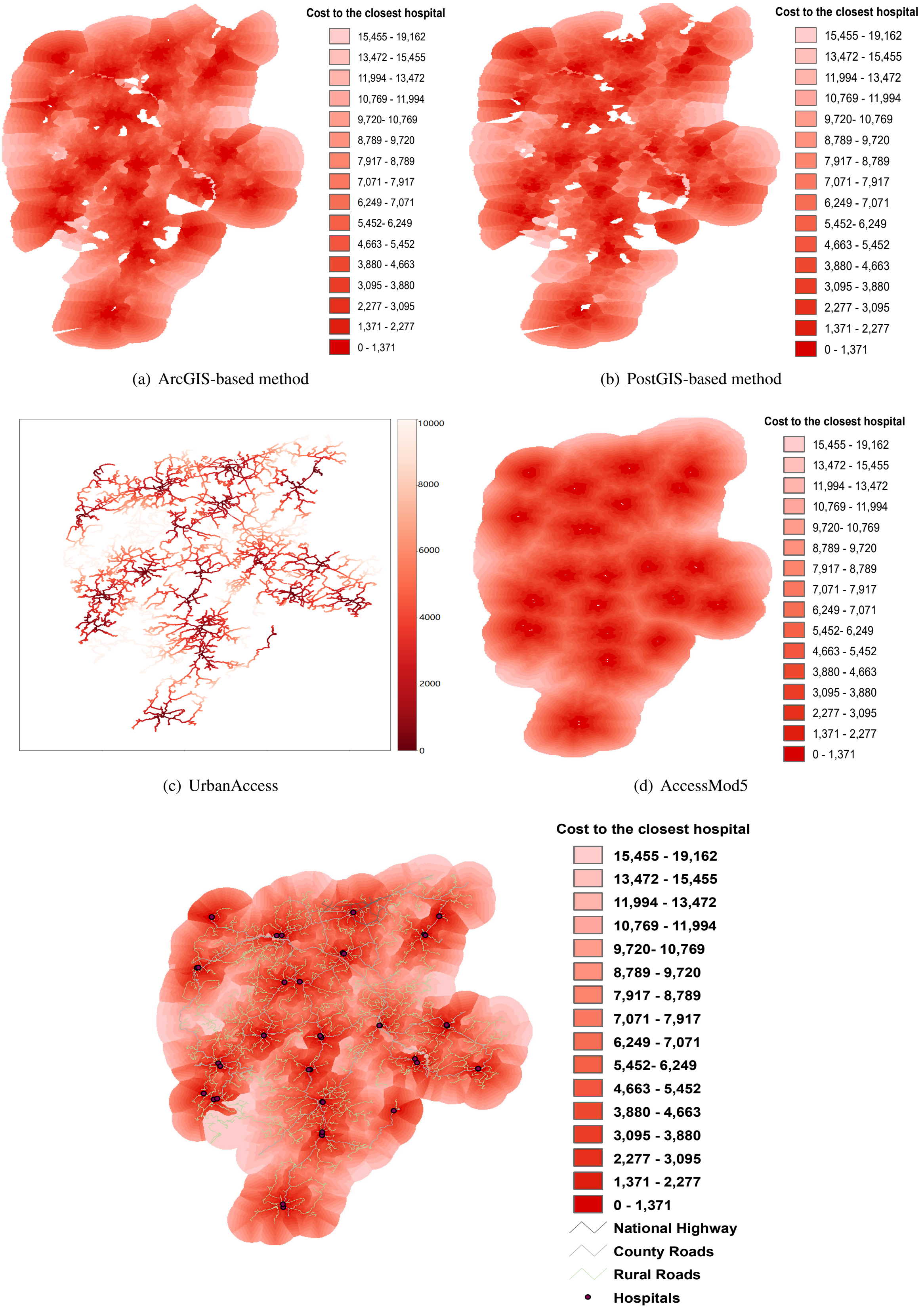
4.2. Comparing with other accessibility analysis methods

In this section, we compare HiAccess with four representative accessibility analysis methods, namely, the GIS-based method (using Network Analyst extension in ArcGIS), the method based on spatial databases (using PostGIS with pgRouting), and two standalone accessibility analysis tools (UrbanAccess and AccessMod5). We choose the roads and hospitals data of Huayuan county as test dataset (see Table 2). In the contrastive experiments, we use the closest-distance model for accessibility measurement. The experimental environment is shown in Table 3. The computation time and accessibility results of the methods are shown in Figure 13 and Figure 12.

Table 2. Roads and hospitals data of Huayuan County

Name	Type	Size
National Highway	Road layer	23646 segments
County Road	Road layer	632 segments
Rural Road	Road layer	205473 segments
Hospital	Facility layer	40 points

The Network Analyst extension in ArcGIS allows users to build a network dataset from road layers. The closest facility solver of Network Analyst can be used to measure the cost of traveling between incidents and facilities and determine which are nearest to one another. In ArcGIS-based method, we import the grids within the distance of 3km to the roads (in total 105235 grids) as incidents and the 40 hospitals as facilities. The import process takes about 22 minutes. It is time-consuming because this process contains matching the grids or hospitals with the nodes in the road network. After an additional 19 minutes, the closest facility solver displays the best routes between incidents and facilities, and reports their travel costs. However, the result only measures



(a) ArcGIS-based method

(b) PostGIS-based method

(c) UrbanAccess

(d) AccessMod5

(e) HiAccess

Figure 12. Accessibility analysis results ($100m \times 100m$ grids) of hospitals in Huayuan county using different methods

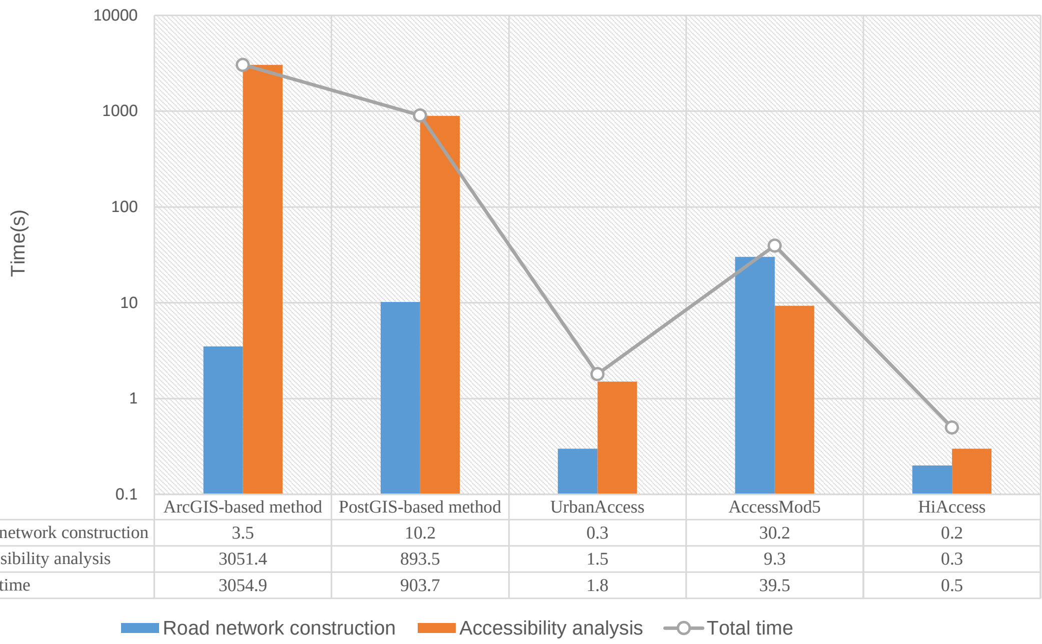
Table 3. Experimental environment

Item	Description
CPU	4core*2, Intel(R) Core(TM)i7-4700MQ@2.40GHz
Memory	16 GB
Software	ArcGIS 10.2, PostgreSQL 9.5, PostGIS 2.2, pgRouting 2.3.2, UrbanAccess 0.1.0, AccessMod_shiny 1.1.1
Operating System	Windows10, X86_64 for ArcGIS-based method. Ubuntu 16.04, X86_64 for PostGIS-based method, UrbanAccess and AccessMod5.

the cost in the road network; to get the final result, we need to add the off-road travel cost. In all, it takes 3054.9 seconds (51 minutes) to get the accessibility analysis result (see Figure 12 (a)).

In PostGIS-based method, we use the pgRouting functions pgr_nodeNetwork and pgr_createTopology to build a network topology from road tables. In order to speed up the process of matching the grids or hospitals with the nodes in road network, we build a spatial index for nodes in the network. We use pgr_dijkstraCost to analyze the cost from grids to the nearest hospitals. The analysis result is shown in Figure 12 (b). It takes 903.7 seconds (15 minutes) to get the final results.

UrbanAccess crudely uses the nodes in road networks as the locations for analysis. It takes less time but produces an unsatisfactory result (Figure 12 (c)). AccessMod5 avoids time-consuming network analysis through rasterizing the roads and deploying accessibility analysis on rasterized grids. It takes only 39.5 seconds to get the analysis result (see Figure 12 (d)).



	ArcGIS-based method	PostGIS-based method	UrbanAccess	AccessMod5	HiAccess
Road network construction	3.5	10.2	0.3	30.2	0.2
Accessibility analysis	3051.4	893.5	1.5	9.3	0.3
Total time	3054.9	903.7	1.8	39.5	0.5

Figure 13. Computation time of the methods

As illustrated in Figure 13, HiAccess shows high performance both in road network construction and accessibility analysis process. Using HiAccess, we can successfully achieve the final result in real time, precisely, 0.5 seconds. Comparatively, it takes 3054.9 seconds using ArcGIS-based method, more than 6000 times compared with HiAccess. In PostGIS-based method, using efficient spatial indexes, we achieve the result in 903.7 seconds. This is much faster than the ArcGIS-based method, though it is still very slow. In addition, there are many "holes" both in the results of ArcGIS-based method and PostGIS-based method. The road network is disconnected and consists of multiple subgraphs, but both ArcGIS-based and PostGIS-based methods only search routes in the main subgraph. HiAccess searches routes in all the subgraphs and produces improved results without "holes" (Figure 12(e)). The UrbanAccess is fast, because it only analyzes the accessibility for nodes in the road network. However, the result is poor, since the roads in Huayuan are sparse. AccessMod5 produces the result with tolerable time cost, 39.5 seconds. The time for road network construction of AccessMod5 ranks first, because in AccessMod5 the road network is rasterized. AccessMod5 is incapable of solving multimodal accessibility models and models considering one and two-way roads. Besides, the rasterized road network is not accurate because disjointed roads overlaping with the same grid are treated as intersecting roads. All in all, HiAccess has the best performance with the best accessibility analysis result among the methods.

4.3. Settlement sites selection for Huayuan County in Xiangxi

This section provides a description of the entire process of applying HiAccess to settlement sites selection for Huayuan County in Xiangxi. Firstly, we analyze the accessibility of jobs, health care, educational resources and all other public facilities. Then we score the accessibility analysis results according to the score rules. Finally, we merge the scores of different accessibility analysis results to get the final scores. Table 4 shows all the facilities for accessibility analysis and their score rules. The whole process takes only 7 seconds. The final scores for settlement sites selection in Huayuan county is shown in Figure 14. Figure 15 displays the distribution of final scores. According to the distribution, we can select the settlement sites with the given area (Figure 16), providing useful analysis for the government.

Table 4. Facilities of Huayuan County

Facility	**Size**	**Model**	**Scoring Rules**[1]
Kindergarten	76 points	Closest-distance	0-500:6,500-1000:4,1000-3000:3,3000-5000:1,5000-Inf:0
Primary school	140 points	Closest-distance	0-500:6,500-1000:4,1000-3000:3,3000-5000:1,5000-Inf:0
Middle school	16 points	Closest-distance	0-500:6,500-1000:4,1000-3000:3,3000-5000:1,5000-Inf:0
Training school	4 points	Closest-distance	0-5000:2,5000-Inf:0
Health-center	22 points	Closest-distance	0-500:4,500-1000:3,1000-3000:2,3000-5000:1,5000-Inf:0
Hospital	40 points	Closest-distance	0-500:4,500-1000:3,1000-3000:2,3000-5000:1,5000-Inf:0
Nursing home	18 points	Closest-distance	0-5000:4,5000-Inf:1
Factory&Enterprise	4417 points	Opportunity-based	0-1:0,1-2:3,2-6:4,6-Inf:5
Tourist attraction	1 points	Closest-distance	0-10000:10,10000-Inf:0
Bank&Credit union	75 points	Closest-distance	0-500:2,500-1000:1,1000-3000:0.5,3000-Inf:0
Town market	15 points	Closest-distance	0-1000:3,1000-3000:2,3000-Inf:1
Supermarket	85 points	Closest-distance	0-1000:3,1000-3000:2,3000-Inf:1
Government facility	253 points	Opportunity-based	0-1:0,1-2:3,2-Inf:5

[1] l-u:s represents accessibility results in range [l,u) are scored with s.

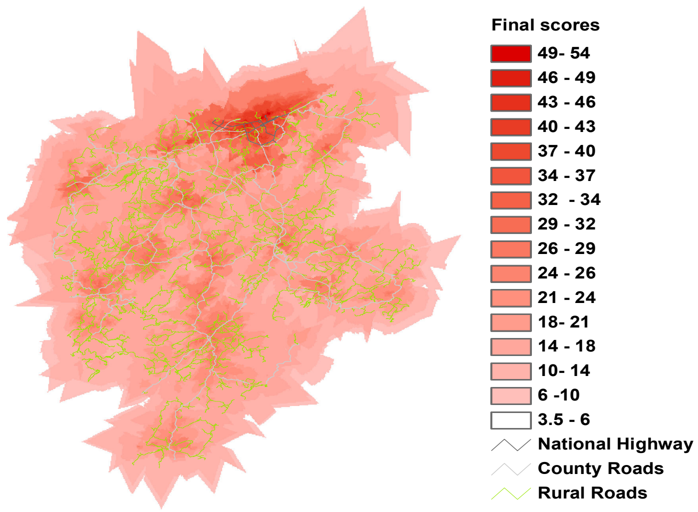


Figure 14. Final scores for settlement sites selection in Huayuan county ($100m \times 100m$ grids)

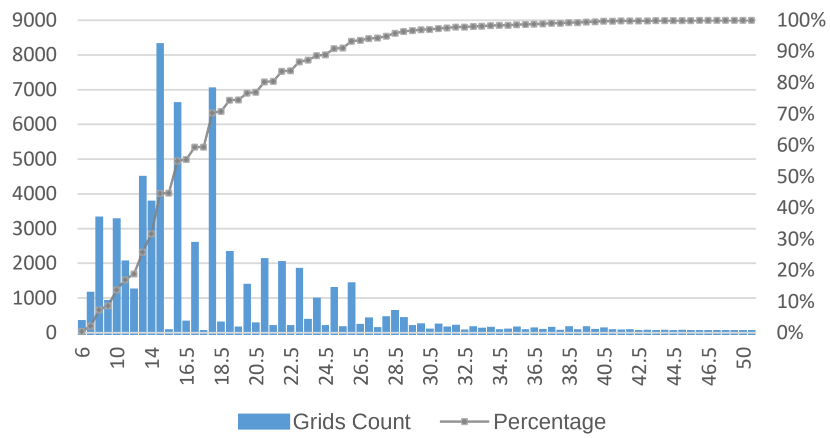


Figure 15. Histogram of the final scores

(a) Score over 18 ($189km^2$) (b) Score over 20 ($149km^2$) (c) Score over 22 ($104km^2$) (d) Score over 24 ($72km^2$)

Figure 16. Settlement sites selection according to the scores

5. Performance on large datasets

HiAccess can be used to solve high-resolution accessibility problems on large datasets. In this section, we apply HiAccess to the medical facilities accessibility analysis in Beijing, China, of which the dataset scale is much larger than that of Huayuan County (Table 5). Accordingly, we use a Symmetrical Multi-Processing(SMP) server for experiments (Table 6). We use the closest-distance model; the accessibility analysis results of medical facilities in Beijing is shown in Figure 17. As Figure 18 shows, we conduct a series of experiments to evaluate the parallel performance of HiAccess. With the increase of process numbers, HiAccess achieves high performance of parallel acceleration, which is approximate to linearity when the process number is below 8. The final results are produced in only 10 seconds using HiAccess with 32 processes. Additionally, the performance of HiAccess not using map generalization is also tested as a contrast. The results show that map generalization greatly reduces the computation time without sacrificing measurement accuracy.

Table 5. Roads and medical facilities data of Beijing

Name	**Type**	**Size**
Beijing Road	Road layer	1126123 segments
Medical facility	Facility layer	1481 points

Table 6. Experimental environment (SMP Server)

Item	**Description**
CPU	32core*2, Intel(R) Xeon(R)E5-4620@2.20GHz
Memory	512 GB
Operating System	CentOS 6.5

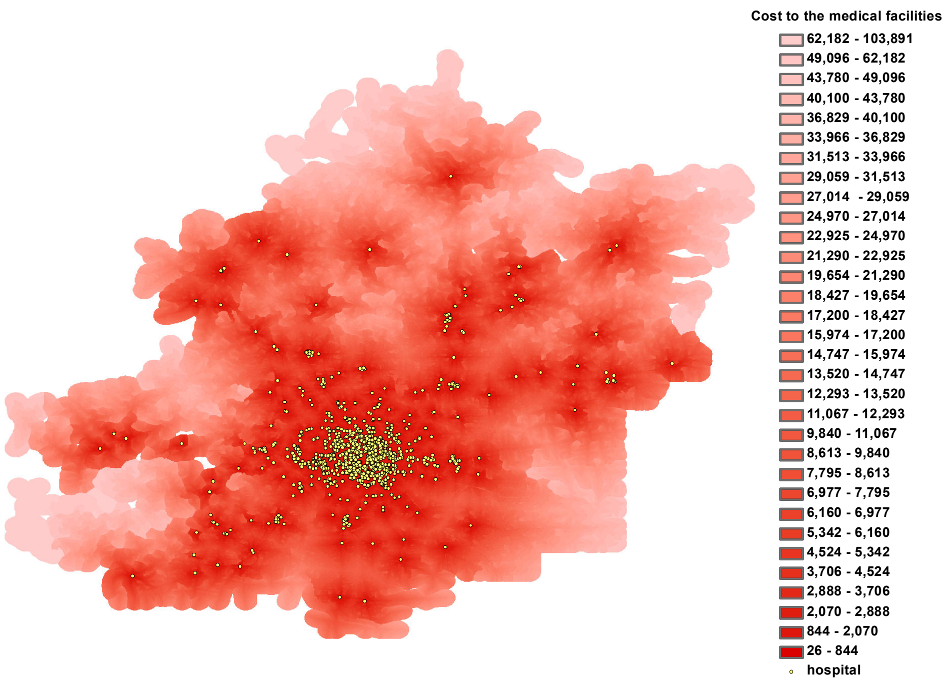


Figure 17. Accessibility analysis results ($100m \times 100m$ grids) of medical facilities in Beijing

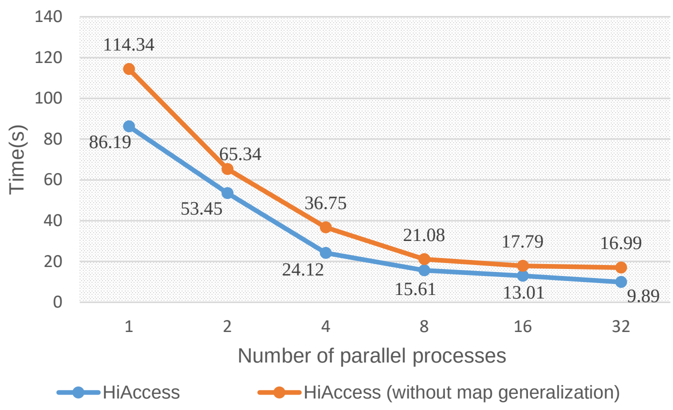


Figure 18. Parallel performance of HiAccess

6. Conclusions and future work

This paper puts forward a parallel processing model, HiAccess, to solve the problems of high-resolution accessibility analysis in real time. In HiAccess, we present an extended road network structure and a corresponding fast construction method named TOB. Spatial indexes are introduced to the structure to improve accessibility analysis speed and the road network topology is efficiently determined by traversing the original road nodes only once. The road network built by TOB is succinct and accurate. Parallel computing technologies based on MPI are used to accelerate analysis. Both repeated computations and the cost of communication between processes are considered in order to improve the capability of parallel processing. The parallel strategies of HiAccess are designed through theoretical analysis and experimental comparison. In the deployment of HiAccess, we find that some grids overlap with more than one segment or facilities. We use map generalization techniques to simplify the segments and facilities without sacrificing measurement accuracy. Using map generalization, the problem scale of medical facilities accessibility analysis in Beijing is reduced from 1.13m segments, 1481 facilities to 0.37m segments, 1022 facilities with grids at the size of 100 meters, and the computing time is reduced to two-thirds of the original. HiAccess is extremely flexible and can be applied to different models with minor changes.

To test the applicability of HiAccess, a case study of settlement sites selection for poverty alleviation in Xiangxi using HiAccess is implemented. Experiments show that HiAccess has the most efficient performance compared with the four representative accessibility analysis methods (ArcGIS-based method, PostGIS-based method, UrbanAccess and AccessMod5). Using HiAccess, we are able to solve high-resolution accessibility analysis problems of a county in real time, precisely, 0.5 seconds. The whole process of settlement sites selection for Huayuan County in Xiangxi (comprehensively analyze the accessibility of jobs, health care, educational resources and other public facilities) takes only 7 seconds. The medical facilities accessibility analysis in Beijing takes 10 seconds, demonstrating that HiAccess can be applied to much larger datasets.

HiAccess does have limitations and some directions for future research are worth noting:

1. More accurate road network construction method remains to be developed. In HiAccess, with the assumption that road connections bring about common nodes in roads, we determine the connectivity of intersecting roads based on whether there is a common node at the intersection (Figure 3). To improve the accuracy of road networks, more attributes of the roads (e.g., height, level, one or two-way, etc) should be taken into consideration to determine the road connectivity.
2. HiAccess needs to be applied to more complex accessibility models. In our research, we deploy HiAccess to distance-based, opportunity-based and gravity-type models. However, HiAccess cannot be directly applied to some more complex accessibility problems, for example, the accessibility analysis based on public transport which takes into account the transport timetable.

Acknowledgments: This work was supported in part by the National High Technology Development Plan of China under Grant No. 2015AA123901 and National Natural Science Foundation of China under Grant No. 41471321.

Author Contributions: Mengyu Ma and Ye Wu designed and implemented the algorithm; Luo Chen and Ning Guo implemented the ArcGIS-based method and PostGIS-based method; Mengyu Ma performed the experiments and analyzed the data; Jun Li contributed to the construction of experiment environment; Mengyu Ma wrote the paper and Qi Gong offered English language advice.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hansen, W.G. How Accessibility Shapes Land Use. *Journal of the American Institute of Planners* **1959**, *25*, 73–76.
2. Ingram, D. The concept of accessibility: A search for an operational form. *Regional Studies the Journal of the Regional Studies Association* **1971**, *5*, 101–107.
3. Hanson, S.; Giuliano, G. *The geography of urban transportation*; Guilford Press,, 1986.
4. Handy, S.L.; Niemeier, D.A. Measuring accessibility: an exploration of issues and alternatives. *Environment and planning A* **1997**, *29*, 1175–1194.
5. Levine, J.; Garb, Y. Congestion pricing's conditional promise: promotion of accessibility or mobility. *Transport Policy* **2002**, *9*, 179–188.
6. Kwana, M.P.; Weberb, J. Scale and accessibility: Implications for the analysis of land use–travel interaction. *Applied Geography* **2008**, *28*, 110–123.
7. Ford, A.C.; Barr, S.L.; Dawson, R.J.; James, P. Transport accessibility analysis using GIS: Assessing sustainable transport in London. *ISPRS International Journal of Geo-Information* **2015**, *4*, 124–149.
8. Benenson, I.; Ben-Elia, E.; Rofé, Y.; Geyzersky, D. The benefits of a high-resolution analysis of transit accessibility. *International Journal of Geographical Information Science* **2017**, *31*, 213–236.
9. Bocarejo, J.P.; Escobar, D.; Hernandez, D.O.; Galarza, D. Accessibility analysis of the integrated transit system of Bogotá. *International Journal of Sustainable Transportation* **2016**, *10*, 308–320.
10. Bhandari, K.; Kato, H.; Hayashi, Y. Economic and equity evaluation of Delhi metro. *International Journal of Urban Sciences* **2009**, *13*, 187–203.
11. Gonzalez-Feliu, J.; Salanova Grau, J.M.; Beziat, A. A location-based accessibility analysis to estimate the suitability of urban consolidation facilities. *International Journal of Urban Sciences* **2014**, *18*, 166–185.
12. Papa, E.; Bertolini, L. Accessibility and transit-oriented development in European metropolitan areas. *Journal of Transport Geography* **2015**, *47*, 70–83.
13. Brondeel, R.; Weill, A.; Thomas, F.; Chaix, B. Use of healthcare services in the residence and workplace neighbourhood: the effect of spatial accessibility to healthcare services. *Health & place* **2014**, *30*, 127–133.
14. El-Geneidy, A.; Levinson, D.; Diab, E.; Boisjoly, G.; Verbich, D.; Loong, C. The cost of equity: Assessing transit accessibility and social disparity using total travel cost. *Transportation Research Part A: Policy and Practice* **2016**, *91*, 302–316.
15. Chen, B.Y.; Yuan, H.; Li, Q.; Wang, D.; Shaw, S.L.; Chen, H.P.; Lam, W.H. Measuring place-based accessibility under travel time uncertainty. *International Journal of Geographical Information Science* **2017**, *31*, 783–804.
16. Ala-Hulkko, T.; Kotavaara, O.; Alahuhta, J.; Helle, P.; Hjort, J. Introducing accessibility analysis in mapping cultural ecosystem services. *Ecological Indicators* **2016**, *66*, 416–427.
17. Liu, S.; Zhu, X. Accessibility analyst: an integrated GIS tool for accessibility analysis in urban transportation planning. *Environment and Planning B: Planning and Design* **2004**, *31*, 105–124.
18. Geurs, K.T.; Van Wee, B. Accessibility evaluation of land-use and transport strategies: review and research directions. *Journal of Transport geography* **2004**, *12*, 127–140.
19. Vandenbulcke, G.; Steenberghen, T.; Thomas, I. Mapping accessibility in Belgium: a tool for land-use and transport planning. *Journal of Transport Geography* **2009**, *17*, 39–53.
20. Makri, M.C.; Folkesson, C. Accessibility measures for analyses of land use and travelling with geographical information systems. *Department of Technology and Society, Lund Institute of Technology, Sweden* **1999**, pp. 1–17.
21. Vickerman, R.W. Accessibility, attraction, and potential: a review of some concepts and their use in determining mobility. *Environment and Planning A* **1974**, *6*, 675–691.
22. Elgeneidy, A.M.; Levinson, D.M. Access to Destinations : Development of Accessibility Measures. *Land Use* **2006**.

23. Witten, K.; Pearce, J.; Day, P. Neighbourhood Destination Accessibility Index: a GIS tool for measuring infrastructure support for neighbourhood physical activity. *Environment and planning A* **2011**, *43*, 205–223.
24. Mavoa, S.; Witten, K.; McCreanor, T.; O'sullivan, D. GIS based destination accessibility via public transit and walking in Auckland, New Zealand. *Journal of Transport Geography* **2012**, *20*, 15–22.
25. Owen, A.; Levinson, D.M. Modeling the commute mode share of transit using continuous accessibility to jobs. *Transportation Research Part A: Policy and Practice* **2015**, *74*, 110–122.
26. Minocha, I.; Sriraj, P.; Metaxatos, P.; Thakuriah, P. Analysis of transit quality of service and employment accessibility for the Greater Chicago, Illinois, Region. *Transportation Research Record: Journal of the Transportation Research Board* **2008**, pp. 20–29.
27. Grengs, J.; Levine, J.; Shen, Q.; Shen, Q. Intermetropolitan comparison of transportation accessibility: Sorting out mobility and proximity in San Francisco and Washington, DC. *Journal of Planning Education and Research* **2010**, *29*, 427–443.
28. Ben-Akiva, M.E.; Lerman, S.R. Disaggregate travel and mobility choice models and measures of accessibility **1979**.
29. Small, K.A.; Rosen, H.S. Applied welfare economics with discrete choice models. *Econometrica: Journal of the Econometric Society* **1981**, pp. 105–130.
30. Hägerstraand, T. What about people in regional science. *Papers in regional science* **1970**, *24*, 7–24.
31. Miller, H.J. Measuring space-time accessibility benefits within transportation networks: basic theory and computational procedures. *Geographical analysis* **1999**, *31*, 1–26.
32. Tong, L.; Zhou, X.; Miller, H.J. Transportation network design for maximizing space–time accessibility. *Transportation Research Part B: Methodological* **2015**, *81*, 555–576.
33. Kibambe Lubamba, J.P.; Radoux, J.; Defourny, P. Multimodal accessibility modeling from coarse transportation networks in Africa. *International Journal of Geographical Information Science* **2013**, *27*, 1005–1022.
34. Wang, K.; Chen, N.; Tong, D.; Wang, K.; Wang, W.; Gong, J. Optimizing precipitation station location: a case study of the Jinsha River Basin. *International Journal of Geographical Information Science* **2016**, *30*, 1207–1227.
35. Benenson, I.; Martens, K.; Rofé, Y.; Kwartler, A. Public transport versus private car GIS-based estimation of accessibility applied to the Tel Aviv metropolitan area. *The Annals of Regional Science* **2011**, *47*, 499–515.
36. Jaakkola, T. GIS-based method for modeling drive and travel times – example of the Helsinki Metropolitan Area. Master's thesis, University of Helsinki, 2013. An optional note.
37. Demontis, R.; Lorrai, E.; Muscas, L.; DeMarr, M.; Estabrook, S. Cost function for route accessibility with a focus on disabled people confined to a wheelchair: the TOURRENIA Project. *Urban and Regional Data Management: UDMS Annual 2013* **2013**, p. 7.
38. Shukla, N.; Wickramasuriya, R.; Miller, A.; Perez, P. Population accessibility to radiotherapy services in New South Wales Region of Australia: a methodological contribution. Journal of Physics: Conference Series. IOP Publishing, 2015, Vol. 616, p. 012004.
39. Blanchard, S.D.; Waddell, P. UrbanAccess: generalized methodology for measuring regional accessibility with an integrated pedestrian and transit network. *Transportation Research Record: Journal of the Transportation Research Board* **2017**, pp. 35–44.
40. Chen, Y.N.; Schmitz, M.M.; Serbanescu, F.; Dynes, M.M.; Maro, G.; Kramer, M.R. Geographic access modeling of emergency obstetric and neonatal care in Kigoma Region, Tanzania: transportation schemes and programmatic implications. *Global Health: Science and Practice* **2017**, *5*, 430–445.
41. Ivan, I.; Horak, J.; Fojtik, D.; Inspektor, T. Evaluation of public transport accessibility at municipality level in the Czech Republic. *International Multidisciplinary Scientific GeoConference: SGEM: Surveying Geology & mining Ecology Management* **2013**, *1*, 527.
42. Karner, A.; Niemeier, D. Civil rights guidance and equity analysis methods for regional transportation plans: a critical review of literature and practice. *Journal of Transport Geography* **2013**, *33*, 126–134.
43. Yang, W.; Chen, B.Y.; Cao, X.; Li, T.; Li, P. The spatial characteristics and influencing factors of modal accessibility gaps: A case study for Guangzhou, China. *Journal of Transport Geography* **2017**, *60*, 21–32.
44. Aji, A.; Wang, F.; Vo, H.; Lee, R.; Liu, Q.; Zhang, X.; Saltz, J. Hadoop gis: a high performance spatial data warehousing system over mapreduce. *Proceedings of the VLDB Endowment* **2013**, *6*, 1009–1020.
45. Cheng, J.; Tu, J.; Han, L. Parallel computation for accessibility based planning support. Conference Paper, 2015.
46. Bondy, B.A.; Murty, U.S.R. *Graph Theory*; Springer London, 2008; pp. 359–372.
47. Guttman, A. *R-trees: A dynamic index structure for spatial searching*; Vol. 14, ACM, 1984.

48. Leutenegger, S.T.; Lopez, M.A.; Edgington, J. STR: A simple and efficient algorithm for R-tree packing. Data Engineering, 1997. Proceedings. 13th international conference on. IEEE, 1997, pp. 497–506.
49. Fernández, F. Boost Geometry library, 2017. Boost Geometry library Available from: http://www.boost.org/doc/libs/1651/libs/geometry/doc/html/index.html.
50. Pacheco, P.S. *Parallel programming with MPI*; Morgan Kaufmann, 1997.
51. Brassel, K.E.; Weibel, R. A review and conceptual framework of automated map generalization. *International Journal of Geographical Information System* **1988**, *2*, 229–244.
52. MacQueen, J.; others. Some methods for classification and analysis of multivariate observations. Oakland, CA, USA., 1967, Vol. 1, pp. 281–297.
53. Langran, G.E.; Poiker, T.K. Integration of name selection and name placement. 1986, pp. 50–64.
54. Douglas, D.H.; Peucker, T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* **1973**, *10*, 112–122.
55. Li, Z.; Openshaw, S. Algorithms for automated line generalization1 based on a natural principle of objective generalization. *International Journal of Geographical Information Systems* **1992**, *6*, 373–389.

Sample Availability: Samples of the compounds ...... are available from the authors.