

A proximity analysis application for large source datasets.

Anders Dahlgren

STU 03/2005-001

Swedish National Rural Development Agency, Samuel Permans Gata 2,
831 30 Östersund, Sweden

Anders.Dahlgren@glesbygdsverket.se

Abstract. Doing detailed proximity analyses in a national perspective of the rural areas of Sweden means dealing with large source datasets. This paper presents the research that is the first phase of the development of a new proximity analysis tool. The hypothesis of this research is that it is possible to develop an effective proximity analysis tool by letting a single functionality, namely proximity analysis of large source datasets, influence the whole design of an application. The main focus for the research project is to improve the performance on an existing application used in production. To reach that goal, new ways of using and combining discoveries within the research areas GIS architecture and Computer science are used.

1. Introduction

Proximity analyses are of interest within many areas. Used in case studies they make an important foundation in describing accessibility relations. In temporal studies they can be used to monitor the change in accessibility over time. Some of these analyses only becomes of real interest when large datasets are used. The datasets can become large both by doing analysis of large geographic areas or by using high-resolution data. A tool that works fine with a small dataset can show unrealistic calculation times or don't work at all when it tries to analyze large datasets. This project deals with a proximity analysis application that has problems with performance when datasets become large.

The hypothesis for this research project is that is possible to develop an effective tool for proximity analysis by:

- Staying focused on the one purpose functionality of the tool, namely proximity analysis in large datasets, letting this influence the design from top to bottom.
- Using direct or in new ways combining discoveries in the scientific frontline within the research areas of GIS architecture and Computer science.

The objective with this paper is to present the project and the work being done so far.

The paper is structured as follows. First a short background to the project. Secondly the requirements on the forthcoming application are listed. Thirdly a test of the existing application that acts as the platform for the development is presented. Then with the test in mind, probable performance improving research areas are listed. Finally some concluding remarks.

2. Background

“Rural areas make up more than half of Sweden. Distances from workplaces and various service outlets are long. Altogether the rural population of Sweden numbers some one and a half million people: an average of three and a half inhabitants per square kilometer. Of these, nearly half live in particularly sparsely populated areas. They are the main concern and focus of Glesbygdsverket, the Swedish National Rural Development Agency” [1]

One of Glesbygdsverkets (the NRDA's) tasks is to define these rural areas and analyze the living conditions for the people living there. NRDA has partly done this by using GIS. Two methods have been used in this analysis. The first, fairly simple, is to look at the density of the population and the second one, a bit more complex, is to use proximity relations between the population and points of interest for the population e.g. food stores, train stations or hospitals. The NRDA has a national responsibility for rural areas all over the country. This means dealing with datasets covering the whole country and because this datasets need to be detailed to make the rural areas visible they become large. If the datasets are generalized and/or lumped together, it is often on the expense of showing detailed conditions in the rural areas.

The GIS platform for doing these analyses has evolved during a 10-year period. One important application in this platform is an application for proximity analysis. The first generation of that application was developed in the early nineties. In time the datasets became more detailed. The performance of the calculations became a growing problem, despite the increased computer power. In the years 2002 - 2003 the second generation of the tool was developed. Referenced in the paper as MapProx (*Map* for the mapping capabilities and *Prox* because it is a proximity tool). Examples of the output from MapProx of such proximity analysis are shown in Appendix 1. When this second generation of the tool was taken into production, the work with developing the tool in a third generation started. This time it was decided to take a more scientific approach to the development by starting it out with a research project.

3. Requirements on the tool

The design of the tool should have a scalable approach. The source datasets used today within NRDA will with all certainty become larger, both expanding into new geographic areas (i.e. Europe) and become more detailed. One example of a more detailed dataset is to use the NVDB from the Swedish Road Agency. This road network is, looking at the number of polylines, twice as dense as the road network used earlier (Blå kartans vägar). Another, more detailed dataset, is using 250m squares instead of 1 Km squares for describing where the population lives. There are about 120 000 populated Km-squares and about 400 000 populated 250m-squares in Sweden.

At a first glance the process of doing proximity analysis seems to be a fairly static task; “once it's done it's done”. But considering changes in the source datasets and simulation functions dealing with iterations, means that a lot of calculations are done. This makes high performance an important requirement on the application.

The development of the application should be done in such a way that dependencies to operating systems and GIS-platforms are held to a minimum. This allows the application to evolve into heterogeneous computer environments.

The design of the application shall consider the future trends in the computer hardware development aiming at e.g. computers with multiple processors and 64-bits processors.

A requirement that of course is of interest is the applications usability. When doing time-consuming calculations it is important for the user to get information on the progress. This also has an impact on the performance while updating the user interface uses computer power.

When new source dataset is used in the application, there are often small changes from the datasets used in earlier calculations. In MapProx today a total recalculation is made regardless of the size or type of the changes. If some incremental update routines could be implemented they will with all certainty have a positive effect on performance.

The application should have the possibility to use a categorization of the target points. This function can then be used to calculate the shortest distance to more than one type of target points in the same calculation.

The MapProx tool has two implementations. Firstly it will work as a desktop application for the employees of NRDA, doing straightforward proximity analyses. Secondly, it will work as a module in larger, more complex systems that have proximity analyses as a base function. An example of the latter is a system for calculating tax adjustments between municipalities¹. This means that the interface to the functions in the tool could be accessed of both a human and another computer process.

When a new system is developed or bought by an organization the transparency of the system should be considered. Buying a ready made system in most cases tend to become a “black box” with few or no possibilities changing the system according to the requirements of the organization, whereas a system developed inside the organization becomes a “transparent box”. It seems to be a growing requirement from organizations to get access to the source code for the applications they use. This is a point often brought forward in discussions in favor of Open Source concepts [2]. Doing proximity analysis is a core function within the NRDA, which makes it important to have a clear insight into the details of the application. The NRDA wants to have access to the source code to have this insight. Of course this discussion of trying to rank, developing a new system with buying a pre-built system, have more angles to it. In this project the transparency of the application was an important factor when deciding to develop an own system because of a legacy of failed projects with pre-built systems.

4. A test of the existing application

The application that was set into production in 2004, the MapProx tool of the second generation, acts as the platform to start out from for this research project. In this part a test of the platform is presented. The aim for the test is to pinpoint the bottlenecks in performance in the application. This was done by analyzing performance in calculations with different setups in a series of tests. All calculations was performed on a 3GHz, 2,0 Gb Ram computer.

4.1 The effect on performance when the maximum search distance where changed

A first test was conducted with large datasets with a variable maximum search distance, calculating the distance to the closest target point. The result is shown in table 1.

- A road network from the National Land survey (Blå kartans vägar). Containing 766894 polylines.
- A point file containing the whole Swedish population tiled into 250 m squares. Containing 400885 squares (The central points of the squares were used in the calculations).
- A point file of food stores. Containing 1767 points.

The MapProx application has a function where it is possible to set the maximum search distance. In the table 1 the maximum search distance is varied to study what impact this has on calculation time. If the maximum search distance is set to low not all starting points reach a target point. In this case no one in the population have more than 200 Km to its closest food store. Not surprisingly, the time for connecting points and generalizing the network stays stable if source datasets are the same.

¹ The NRDA assists the Ministry of Finance to calculate tax adjustments dependent on structural differences between municipalities.

Maximum search distance (Km)	Connecting points to network.	Generalizing the network	Calculating the shortest distance	Total time for calculation	Gives complete results.
50	55 min	16 min	7 min	1h 18min	no
100	55 min	16 min	21 min	1h 32min	no
150	55 min	16 min	38 min	1h 49min	no
200	55 min	16 min	1h 25min	2h 36min	yes
300	55 min	16 min	3h 17min	4h 28min	yes
500	55 min	16 min	8h 23min	9h 34min	yes

Table 1. A test showing that calculation time varies with different maximum search distances.

4.2 The effect on performance when two different road networks were used.

A second test was done using two different road networks with the same start and target points, calculating the closest target point. The maximum search distance was set to 200 Km. The results are shown in table 2.

- Two road networks:
 - A road network from the National Land survey (Blå kartans vägar). Containing 766894 polylines. The size in MapInfo format is 141 Mb
 - A road network from the Swedish Road Agency (NVDB). Containing 1876472 polylines. The size in MapInfo format is 515 Mb
- A point file containing the whole Swedish population tiled into 250m squares. Containing 400885 squares (The central point is used in the calculations).
- A point file of food stores. Containing 1767 points.

Road network	Connecting points to network.	Generalizing the network	Calculating the shortest distance	Total time for calculation
Blå kartans vägar	55 min	16 min	1h 25min	2h 36min
NVDB	4h 49min	39 min	1h 27min	6h 55min

Table 2. A test with two different road networks and many target points.

A variation of the third test was to use a single target point and higher maximum search distance. The results are shown in table 3. The changes in the input parameters from the test before were:

- One single target point was used (A central point of Stockholm)
- The maximum search distance was set to 2000Km.

Road network	Connecting points to network.	Generalizing the network	Calculating the shortest distance	Total time for calculation
Blå kartans vägar	55 min	27 min	34 min	1h 56 min
NVDB	4h 45min	38 min	50 min	6h 13min

Table 3. A test with two different road networks and a single target point.

4.3 Calculating a large distance matrix.

A test to calculate a large distance matrix was defined:

- A road network from the National Land survey (Blå kartan). Containing 766894 polylines.
- A point file of all Swedish population centers with more than 200 inhabitants. Containing 1936 points.

The test calculated distances between all population centers to all the other population centers ending up with a distance matrix with theoretically 3748096 calculated distances (taking in count the zero distance if the start and target points are the same). The actual number of calculated distances was 3648400 because the island of Gotland does not have a connection to the mainland in this road network. The calculation took 57 minutes in MapProx. Breaking down the calculation into sub processes:

- 10 minutes of connecting points to the road network.
- 7 minutes of generalizing the road network
- 40 minutes of calculating the distances

5. Identifying research areas.

Looking at the result of the test, four candidates for performance improving research areas are selected. Three of areas of can be directly related to the measurable sub processes in the test:

- Create a spatial index and connect the points to the road network.
- Generalize the road network with the connected points.
- Calculate the shortest distance between start and target points.

The overall interest area of the internal data structure can be added to this list. This area is affected of all the three above but should because of it's importance be handled separately. The points identified above delimit this work. There are of course other areas that could affect the performance of the application and should be treated in the development project but they are not the concern of this paper.

5.1 Create a spatial index and connect the points to the road network

When Start and Target nodes are imported they are connected to the closest point on the road network. This can be a node or at right angles on a link. If the later method is used the link is broken up in two and a new node is introduced. To make this procedure effective a spatial index must be used. In MapProx of the second generation a Kd-tree is used, e.g. [3]. The test in the previous chapter show that the time for connecting the points to the road networks is a significant part of the overall process. This is especially true when a short maximum search distance is used. When a 50 Km maximum search distance was used in the test connecting the points was 67% of the total calculation time, this can be compared with 10% of the total calculation time using a 500 Km maximum search distance. A number of spatial index methods should be studied to se if they can improve performance of this step in the calculation. An important fact that should be used as a starting point is that the spatial index in this case only has one purpose, namely connecting points to the closest line.

5.3 Generalization of the road network with the connected points.

The MapProx tool is a one-purpose tool, namely performing proximity calculations. Tools with similar functions often have a multipurpose approach. They often tend to broaden their field of operation to please a waste amount of users. However, such designs often lead to compromises in the core function of the application. An example of this is the mixing of proximity analysis and navigation. In a proximity analysis tool you are not interested in a description of the path from start point to target point. You just want to know the travel distance or travel time. This fact has impact on the generalization process where a pure

proximity calculation can use a more effective approach. Removing relations between the complete geographic representation of the road network and the generalized road network is no problem.

Start points and target points must be connected to the road network before the generalization process can begin. The goal with the process is to simplify the road network to make the calculations of the distances as smooth as possible without tempering with the correctness of the result.

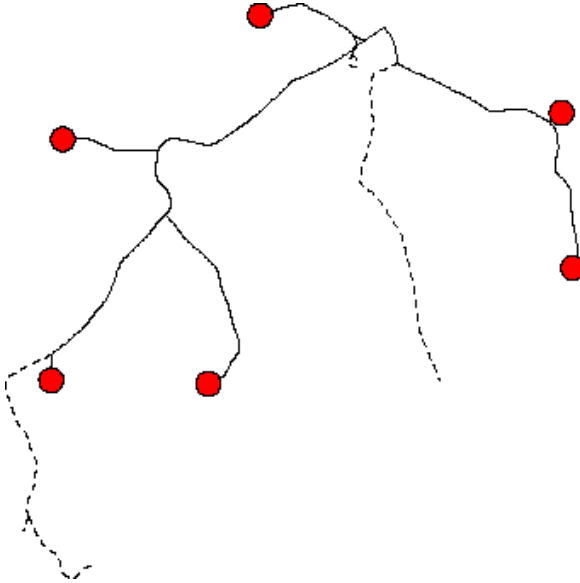


Figure 1. The figure describes removable links (dotted) in a road network. The dots are start or target nodes.

One step is to remove all dangling polylines that doesn't have of a start or target point. Figure 1 shows as dotted lines the polylines that can be removed without introducing errors in the final result.

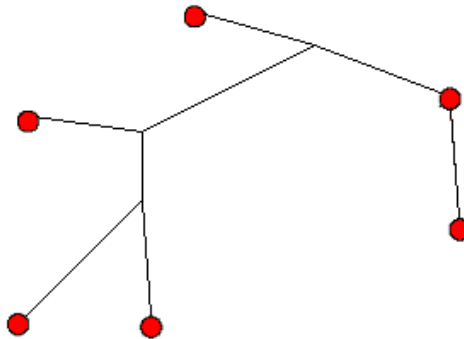


Figure 2. The figure shows how straight lines can replace complex polylines.

Another generalizing step is to remove nodes in polylines that are not start or target nodes, or have more than two links connected to it. When this is done it is important to recognize that the geometry of the link is destroyed by and the actual distances (or time-distance) are saved as attributes to the links. This is shown in figure 2 where the road network from

figure 1 is generalized in the described way. The generalization process affects the development of the topological format. The test shown no significant bottlenecks in the generalization process when the datasets become large. This gives the generalization process low priority in the research project.

5.4 Calculate the shortest distances between start and target points.

Finding the shortest way in a network graph is a classic problem in computational and mathematic sciences. The algorithm used in the tested application is the Dijkstra algorithm [4] later described and discussed in many sources e.g.[3-5]. There have also been a number of suggestions of improvements and modifications of the Dijkstra algorithm e.g. [6, 7]. The Dijkstra has been proven to be the fastest exact search algorithm but suggestions in [6, 7] could be of interest for the further development of MapProx. The test showed an increased calculation time with increased search distances. The recommendation is still to prioritize the internal data structure before going into details of the search algorithm.

5.5 The internal data structure

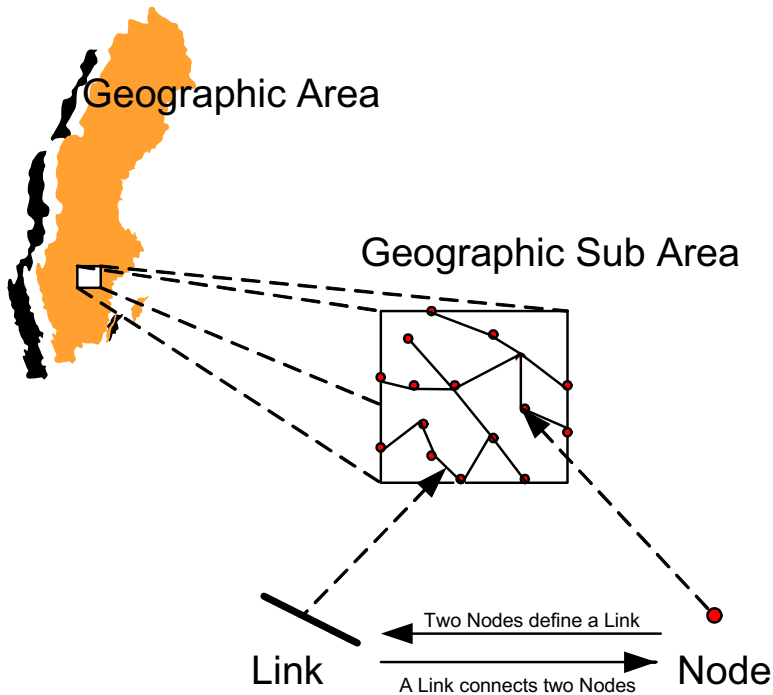


Figure 3. The figure describes a simplified schema of the internal format used.

Figure 3 shows the internal format the existing application uses in a simplified manner. There is a geographic area tiled into of sub areas. One sub area tile holds the reference to the corresponding links and the nodes that lay within the area. The advantage with the approach of tiling the geographic area into sub areas is that the sub areas become “independent” from each other, which make them easier to handle. The Sub areas are serialized into individual files in the internal format. The sub areas can be handled independently in their own calculating threads, suggesting parallel calculations as a possible performance enhancing measure. The disadvantage with the tiling is that a new type of nodes must be introduced on the borderlines between the tiles. These border-nodes will appear in two adjacent sub tiles. The intimate connections between links and nodes (in the implementation realized with pointers) are the core of the internal topology of the

internal format. As long as the road network stays intact the same imported network can be reused in forthcoming calculations.

A suggestion is to let the geographic sub area have a polymorphic behavior. The area should have three different interfaces, illustrated in figure 4, used as described below. It remains to be tested if this way of structure of the data has an effect of performance or only works as a structural clarification.

The processes that connect start and target points to the road network use the first interface. In this interface it is important that the right geographic representation of the road network is used. When the start and target points are connected the generalization process can start presenting its result, a simplified road network, through the second interface. This is used for searching within a Geographic sub area. When a sub area only has the function as transportation between two other sub areas, i.e. the area does not have any start or target point of interest for the calculation connected, the third interface is used. This interface shows the sub area as a matrix of distances between the sub areas border nodes. An interesting fact is that the matrix can be calculated before the connection of the start and target nodes in a precalculation.

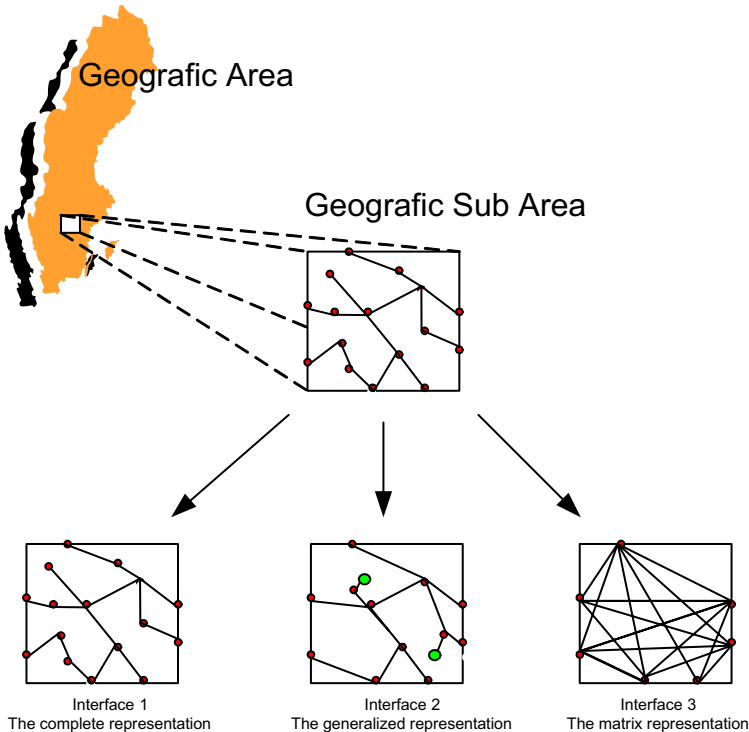


Figure 4 Three interfaces of the Geographic Sub Area component (The big nodes in Interface 2 represents start or target nodes)

In the design it should be considered what processes that could use parallel calculations. One overall calculation process can use both a single thread approach in a sub calculation and a multithread approach in another (figure 4). Examples of this can be the import of a single in-data file that is a single thread procedure. When the dataset is imported and tiled into subsets, a multithread approach can be used. The test could not directly pinpoint any bottlenecks in the internal data structure because it affects the whole calculation process. When the size of the datasets increases a hieratic approach [8] and the suggested interface structure should be of interest of study further.

6. Concluding remarks

One conclusion that could be drawn from the test is that when the road network becomes large, the time for connecting points to the network becomes unnaturally large. Comparing the connection step of the process with the other calculation steps e.g. in test 4.2 (one target point and the larger network) the time for the connection step is 76 % of the total calculation time. This indicates that the spatial indexation in the application was not built for large road networks and should have the first priority in the continuing research work.

As the second priority the internal data structure should be considered. An approach with a hierarchical structure should give result in improving performance. It seems like the one level tiling used in MapProx is not optimal solution for very large datasets.

The third priority should be the calculation step where the actual distances between start and target points are calculated. Especially when the search distances are large. In test 4.1 it is shown that when the search distances grows, the time for the calculation increase rapidly. If a better method for this calculation step is found, gains in performance can be won.

As told before this research project is the start of a larger project of developing a new generation of the MapProx tool. To limit the research project the recommendation for the continuing work is to focus on the three prioritized points above.

The design of MapProx the second generation is implemented in a functional application, as partly shown in test, but that work can also be looked at as a first step. The future development can now be done with the confidence that it is possible to go back one step and still have a working system. The object-oriented development methods allow the developers to look at the different parts of the application separate. A good design is a design that can be implemented and maintained in a cost effective way. It is important for the designer to make himself understood by the developers. This is a limiting factor for the designer. It is often better to design an algorithm that is easy to understand than a complex algorithm of scientific excellence. With modern object oriented development methods it is also a good standard to start out with a simple algorithm and then change it to a more complex when you have a functioning overall system. This method of iterative development is supported by e.g. the Rational Unified Process [9]. Because this approach was used to develop MapProx it should be well suited for any changes that could be suggested.

7. Acknowledgements

I would like to thank Lars Harrie at Lund University and Professor Anders Östman at the University of Gävle for valuable point of view on the work and the article. I would also like to thank the NRDA for financing the studies.

References:

1. NRDA, www.glesbygdsverket.se, The National Rural Development Agency's Homepage. 2003.
2. Raymond, E.S., *The cathedral and the bazaar : musings on Linux and open source by an accidental revolutionary*. Rev. ed. 2001, Sebastopol, Calif.: O'Reilly. 241.
3. Berg, M.d., *Computational geometry : algorithms and applications*. 2., rev. ed. 2000, Berlin: Springer. xii, 367.
4. Dijkstra, E.W., A note on two problems in connection with graphs. *Nummer. Math.*, 1959. 1: p. pp. 215-248.
5. Ahuja, R.K., J.B. Orlin, and T.L. Magnanti, *Network flows : theory, algorithms and applications*. 1993, Englewood Cliffs, N.J.: Prentice Hall. xv, 846.
6. Lauther, U., An Extremely fast, exact algorithm for finding shortest paths in static networks with geographical background. *Geoinformation und Mobilität*, 2004. 22: p. 219-230.
7. Ertl, G., Shortest path calculation in large road networks. *OR Spektrum* 20, 1998: p. 15-20.
8. Car, A., and A. U. Frank, . and General Principles of Hierarchical Spatial Reasoning - The Case of Wayfinding. *Proceedings of the 6th Spatial Data Handling Symposium*, Edinburgh, 1994. Vol. 2: p. pp. 646-664.
9. Kruchten, P., *The rational unified process : an introduction*. 2. ed. Addison-Wesley object technology series,. 2000, Reading, Mass.: Addison-Wesley. xviii, 298.

Appendix: 1

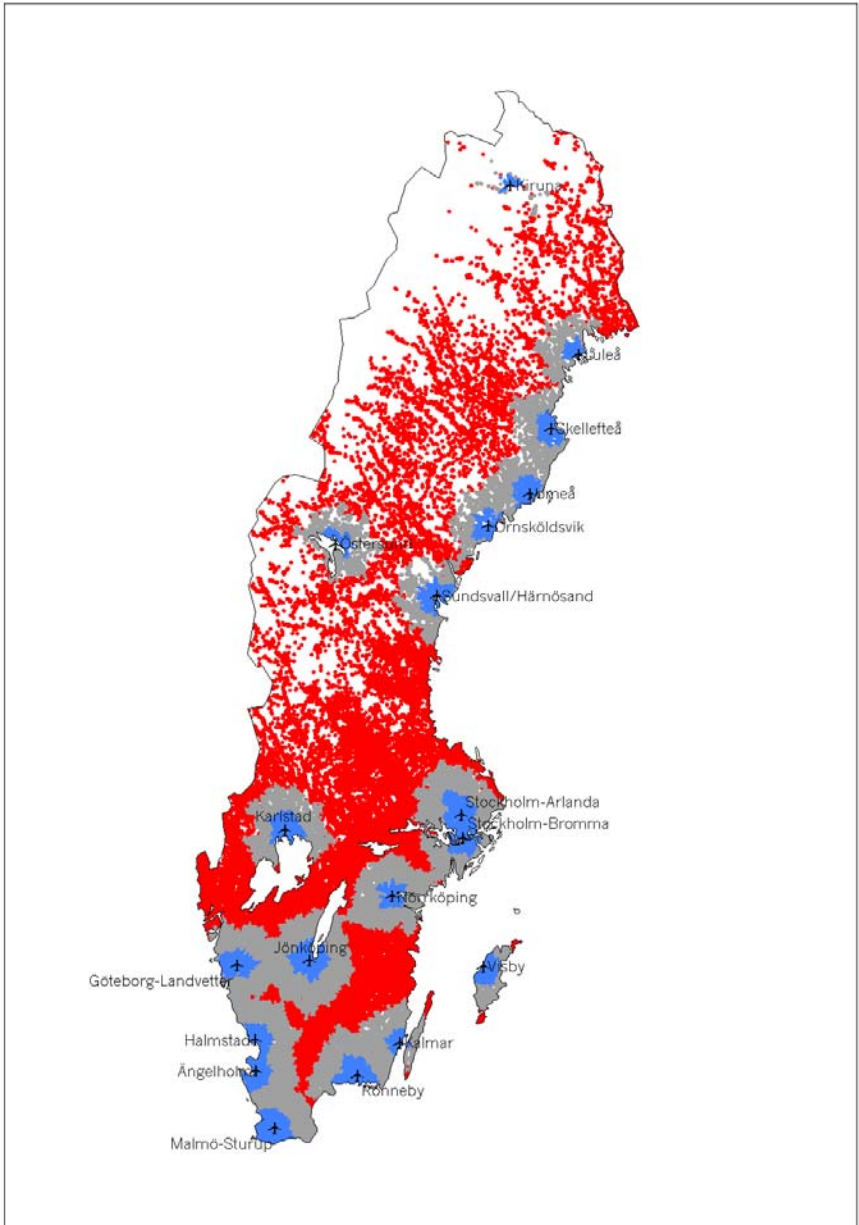


Figure 5 Accessibility to governmentally owned Swedish Airports.

Traveltime by car to closest airport

- More than 60 minutes
- 30 to 60 minutes
- 0 to 30 minutes
- Areas without population

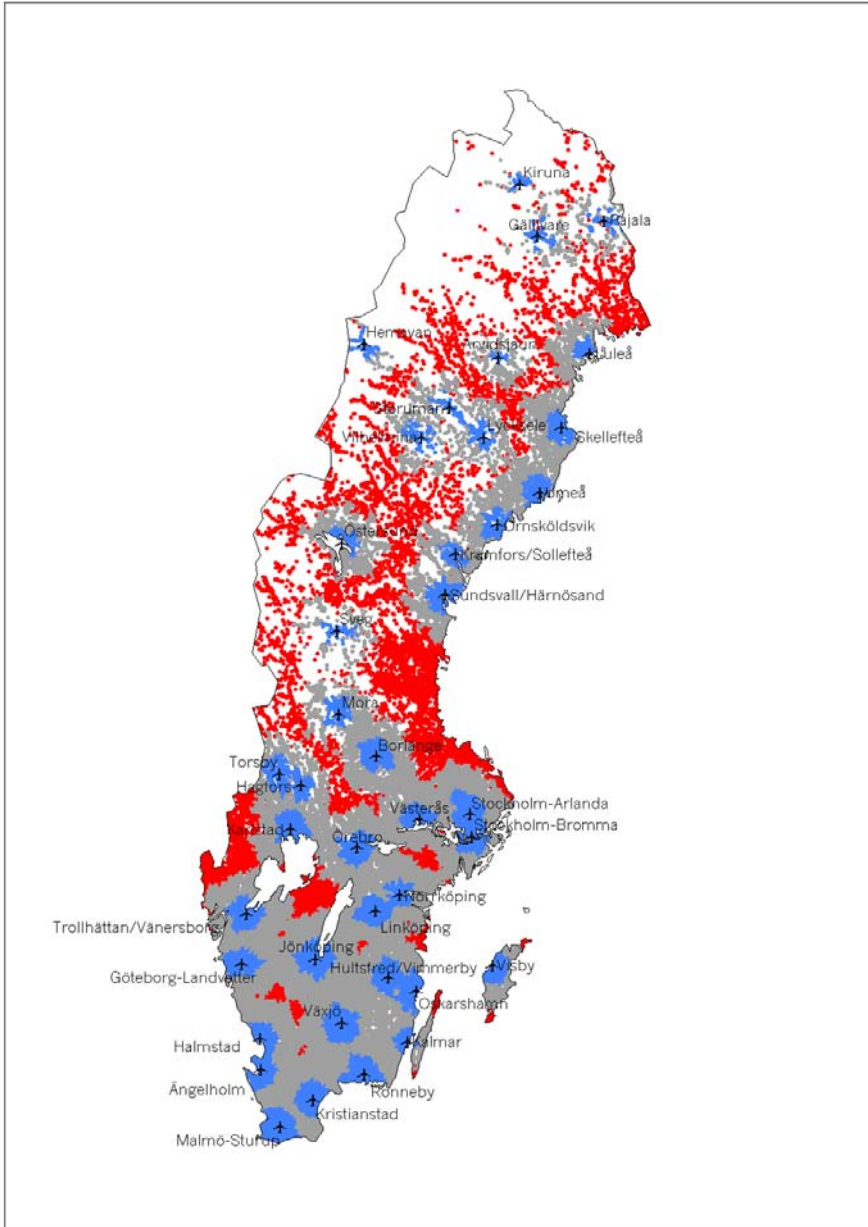


Figure 6 Accessibility to all Swedish Airports with regular domestic traffic.

Traveltime by car to closest airport

- More than 60 minutes
- 30 to 60 minutes
- 0 to 30 minutes
- Areas without population