

Dr. Gürdal Ertek
gurdalertek.org
Working Papers
research.sabanciuniv.edu

Sabancı
Universitesi

Yamaner, Y., Yamaner, Y., Akçay, A. E., Ertek, G. (2008) “An open source Java code for visualizing supply chain problems” . CELS 2008, Jönköping, Sweden. (presented by Yekta Yamaner, Yalçın Yamaner and Alp Eren Akçay).

Note: This is the final draft version of this paper. Please cite this paper (or this final draft) as above. You can download this final draft from <http://research.sabanciuniv.edu>.

An Open Source Java Code For Visualizing Supply Chain Problems

Yekta Yamaner, Yalcin Yamaner, Alp Eren Akcay, Gurdal Ertek

Faculty of Engineering and Natural Sciences,
Sabanci University, Orhanli, Tuzla, 34956, Istanbul, Turkey

Abstract – In this paper, we describe an open source Java class library for visualizing supply chain problems within a geographical context. The highly competitive markets and recent technological advances make the use of such supply chain network visualizations critical in both strategic and tactical levels. The most important characteristic of our work is its easy integration with any Java

application. Our software differs from any other commercial and open source supply chain visualization tool by its simple structure, easy adoption and implementation and high compatibility. The main motivation of our study was to develop a simple – yet effective – library that would not require to learn and apply complicated visualization tools and data structures such as Geographical Information Systems (GIS). In this study, we illustrate the use of our visualization tool through maps of Turkey, Europe, North and South America, the United States and the NAFTA. We believe that ease of visualization offered by our open source tool will contribute to a multitude of projects in supply chain design, as well as increasing productive communication among practitioners, especially involved in strategic level decision making processes. We foresee that our supply chain visualization tool will fill a gap in this area with its simple but effective structure.

Keywords – Supply Chain Visualization, Information Visualization in Logistics, Open Source Network Visualization Library

Introduction

The visualization of the supply chain networks plays an effective role in the interpretation of the supply chain activities. Such visualizations facilitate the perception of both the extend and the volume of the transportation within the network. They virtually increase the level of communication between various stages of decision making in a company as well. The boom in the number of commercial and open source software tools and libraries for supply chain visualization indicates the significance of the subject.

In this study, we initially posed the following design question: Is it possible to represent and visualize supply chain problem solutions in a simple manner, focusing on supply chain context and without getting lost in the details of geographical features? The open source Java class library we present in this paper for visualizing supply chain problems is the outcome of this question. The fundamental characteristic of our visualization tool is its easy integration with any

Java application. It differs from any other commercial and open source visualization tools by this simple utilization. In addition, high compatibility and easy implementation distinguishes our visualization tool from its counterparts. In our study, we constructed a text database of only essential geographical information for supply chain planning rather than storing every type of geographical information. Specifically, we collected data on regions, city names and city coordinates. The Java class library automatically reads the data for a specified geographical region and plots the map on which the supply chain is visualized. The database, including the geographical information can be easily extended to cover any region of the world. We believe that our current library satisfies the initial design goals: It is lightweight, crossplatform, simple and extremely easy to use, and effective for the specific domain that we are interested in.

The Problems to be Visualized

In this study, we propose a simple but effective visualization of the given network flows within a supply chain. Thus, we believe that our work can be a complementary part of innumerable studies which are conducted to solve the widely known *Supply Chain Network Design* problems and the combinatorial optimization problems of *Traveling Salesman Problems (TSP)* and *Vehicle Routing Problems (VRP)*.

In TSP, each node of the network has to be visited exactly once on a single tour, with the objective of minimizing the tour's length. In VRP, the goal is to construct a number of vehicle routes which start and end at the depot such that the total travel distance is minimum. The number of routes can be set by the user apriori or can be a decision variable. Again, each node in the network has to be visited once and additional constraints may be present in the problem. Vehicle capacity, route length restriction and time windows in demands are some of these important constraints which make VRP more complicated. A thorough review about VRP can be found in La Porte et al. (2000). In addition, a detailed analysis of TSP and VRP is available in many classic operations research books such as Aarts and Lenstra (1997), Bramel and Simchi-Levi (1997), Korte and Vygen (2005), Gutin and Punnen (2002), Toth and Vigo (2001). Historical notes, various formulations and solution algorithms for such combinatorial optimization problems can be found in these resources.

TSP and VRP are heavily studied in the literature but the visualization has not been the primary focus of existing software libraries. An effective visualization of the solutions proposed by any algorithm can be very helpful to understand the mechanics of that algorithm, to compare the algorithm with benchmark instances and to grasp the practical implementation of the solution. We believe that the following characteristics are vital in an effective visualization software:

- The visualization software should be lightweight, yet sufficient for the selected domain. (ex: the domain of supply chain planning)
- The software should be simple and easy to use.
- The software should effectively visualize the domain – specific data.
- The background on which the supply chain design is visualized should be similar to the real network region. Thus, a real geographical map of the region is highly useful.
- The use of the visualization tool should be simple so that it is compatible with other applications developed in the same language and other developers can easily alter the tool for their own use.
- The amount of flow between each pair of nodes should be illustrated visually for an effective visualization, and should be scalable with respect to the range of values.
- Especially in the strategic level, the visualization should focus on the representation of the information related with the supply chain planning and the details in the geographical structures (i.e. highways, mountains, city borders etc.) can be omitted to keep the attention on domain specific data (ex: supply chain entities and flows).

Related Work

Integration of the geographical data with advanced visualization techniques urges supply chain planners to show their work in a descriptive and visual manner. In that sense, Geographical Information Systems (GIS) are widely used in the state of the art software tools that visualize the attributes of various regions on the world. In this section, firstly, we will explain GIS briefly and discuss some of the important open source GIS software. Next, the integration of supply chain design with geographical information and some commercial and open source supply chain visualization tools will be examined.

Geographical Information Systems (GIS)

“Geographic Information Systems (GIS) are an organized collection of computer hardware, software, geographic data, and personal design to efficiently capture, store, update, manipulate, analyze, and display all forms of geographic and geologic information.” (ESRI, 1995) A thorough history and evolution of GIS can be found in Foresman (1998). Applications of GIS range from environmental protection, meteorology, transportation and military operations to business planning and marketing. In other words, it is possible to benefit from GIS applications to better understand the earth and the interactions of the human with the earth. For example, for transportation research and management late 1980s are the golden ages in terms of widespread use of the GIS (Thill, 2000). Web-based traffic information systems and trip planning engines, in-vehicle navigation systems and real-time congestion management and accident detection are some of them.

The development and widespread use of open source GIS software in recent years has brought cost-effective and customized alternatives. Some of the renowned open source GIS software need to be highlighted. Geographic Resources Analysis Support System (GRASS) is the most widely known open source GIS software, and operates on various platforms through a graphical user interface (GUI). It is possible to state that GRASS is the largest, most powerful and reliable open source GIS project. GRASS has a detailed documentation and the users can easily write their own modules even with a basic C programming language skill (Smotrisky, 2004). Another

popular open source GIS application is Quantum GIS (QuantumGIS), which is often abbreviated as QGIS. It is a multi-platform application and is compatible with different operating systems while supporting vector, raster and database formats. Figure 1 gives a routing visualization example using QNav software, which creates the visualizations in QGIS. MapServer (MapServer) is one of the widely used web-based mapping editors. Spatially-enabled web mapping applications and services can be developed in this development environment, which was originally created by the University of Minnesota. Some other important open source GIS software are MapWindow GIS, GeoTools, NRDB Pro, OpenGeoDB (listed under the References section). There are also some open source tools to generate GIS data according to the International Organization for Standardization (ISO) standards. Some of the important ones are MIG Editor and GeoCatalogo.

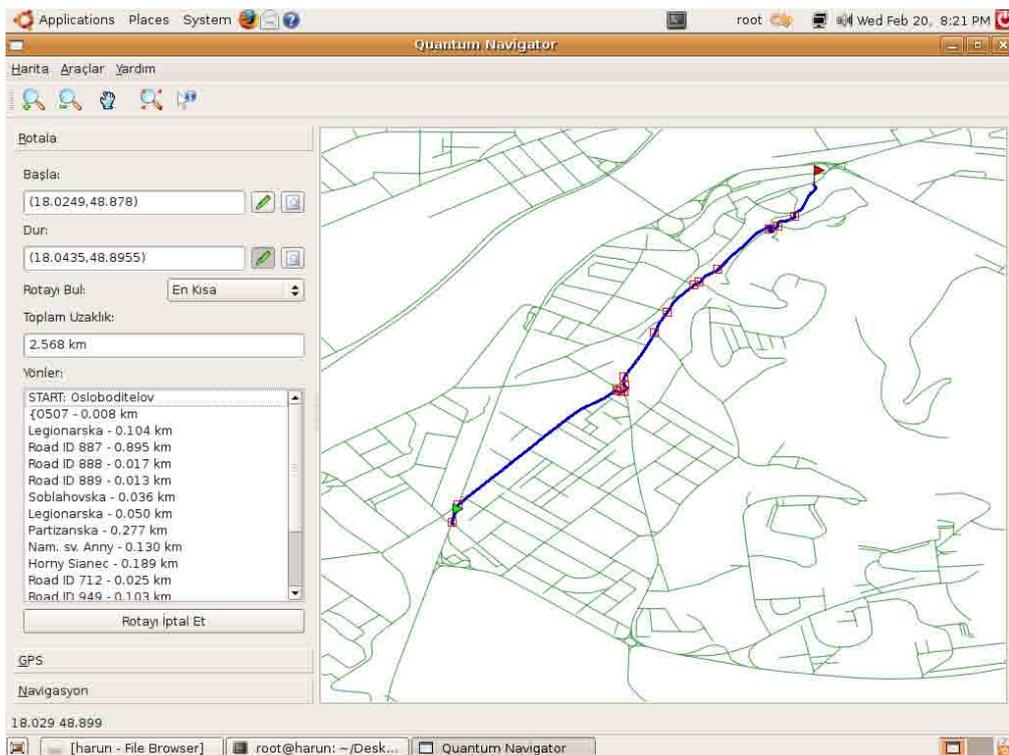


Figure 1. Routing in supply chain using QNav software

Ramsey (2007) evaluates the current state of open source GIS software in detail. The author examines almost all open source GIS software in accordance with their implementation languages and platforms such as C, Java and .NET. Shared libraries and applications for each

class are exemplified. In addition, a survey of open source GIS web projects is provided. Nasr (2007) also investigates the use of open source GIS software and its impact on organizations. It involves a comparative analysis between a leading open source web GIS tool, MapServer, and three of the leading commercial web GIS software, namely ESRI's ArcIMS, Intergraph's GeoMedia WebMap and MapInfo's MapXtreme. It concludes that the open source MapServer is technically equivalent to its commercial counterparts and lack of awareness in the open source concepts is the main reason behind the poor adoption of open source GIS software in business in contrast to academia. Camara et al. (2007) investigates the publicly available open source GIS software in terms of their maturity, support and functionality as well as the developers' organization structure (i.e. individual based, network team and corporation-based).

The Integration of Supply Chain Design with Geographical Information

The visualization of the solution of a supply chain design optimization problem significantly improves the effectiveness of the solution by conveying the results in a more descriptive and explanatory way. There are a number of visualization tools integrated with an optimization engine available in the literature. LogicTools is a proprietary division of ILOG, world's leading optimization software developer. Network visualization is available in the LogicTools software in addition to other supply chain planning and scheduling activities. Transport Powerops is another proprietary software developed by ILOG. It also provides supply chain design visualization for professionals and researchers working on supply chain design. Llamasoft is another proprietary software for supply chain optimization. Network simulation is also available in this software in addition to an effective visualization. A screenshot from this commercial GIS software is given in Figure 2.

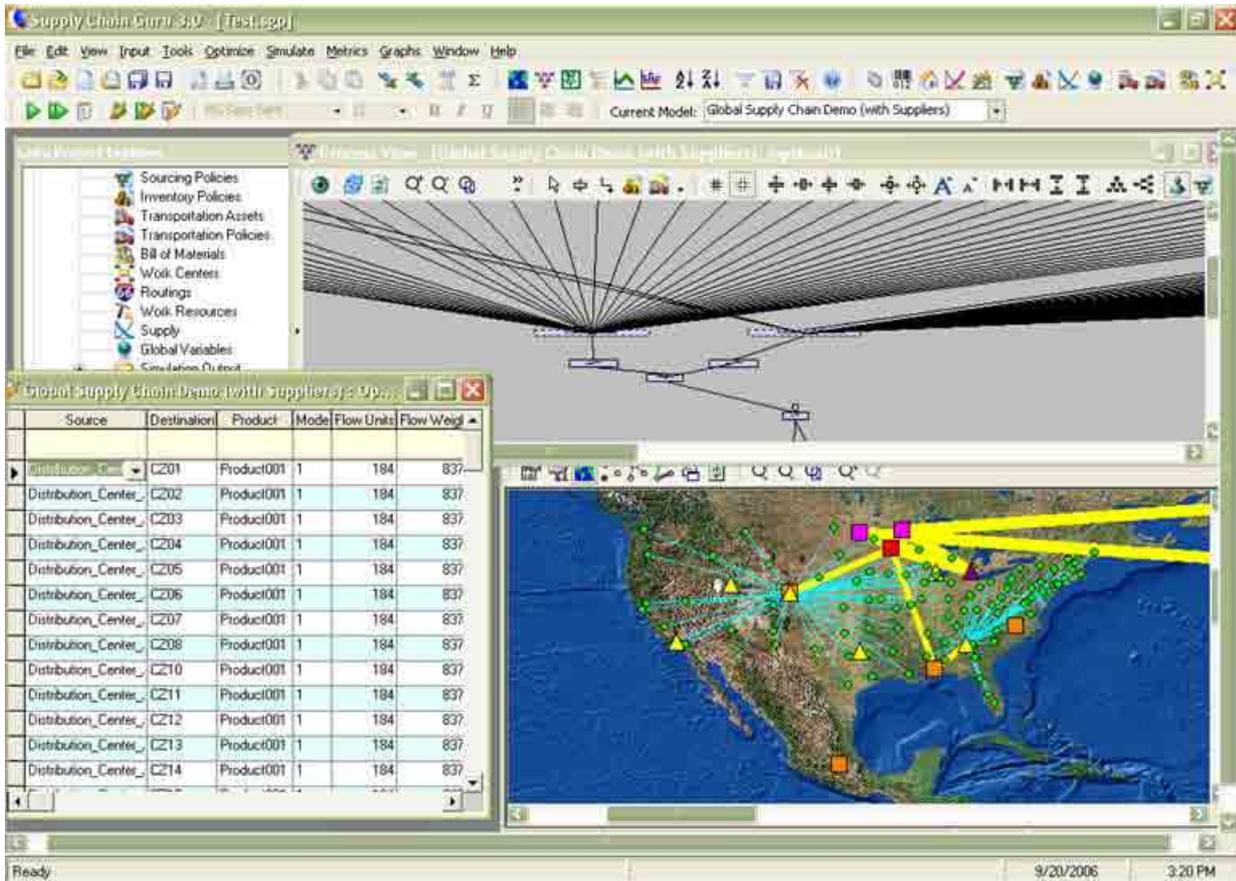


Figure 2. A supply chain design with llamasoft software
(used with permission from LlamaSoft)

ESRI, one of the leading commercial GIS software, also provides a tool integrating the benefits of GIS with logistics and supply chain problems. A network-based spatial analysis can be fulfilled by the extension, ArcGIS Network Analyst in ESRI. Nardi et al. (2007) discusses the development and implementation of a GIS-based constrained linear programming model about the minimization of transportation and storage costs for soybeans and associated products. General Algebraic Modeling System (GAMS) is used to solve for the model and ArcMap, a component of ESRI's ArcGIS, is used to map geographical data before and after the linear programming optimization in GAMS. A supply chain planning system which is enhanced with visualization modules is discussed by Camm et al. (1997). The system described in their study

visualizes the results of an optimization algorithm that eventually saved the giant consumer products firm Procter & Gamble more than \$200 million. A detailed list of the proprietary software which can be used in supply chain and logistics problems can be found in the web site <http://www.logisticsworld.com/software.asp>.

Different features of the various supply chain visualization software play a significant role as selection criteria of practitioners or supply chain design researchers. They include:

- Pricing of the software license and the license terms,
- Ease of the installation,
- Stability,
- Compatibility with any other already existing tools such as Enterprise Resource Planning (ERP) or Customer Relations Management (CRP) software,
- Service quality of the provider,
- The ease and convenience in learning and practicing the software even by a person without deep technical knowledge.

In the further parts of the paper, a detailed analysis of our proposed visualization tool is given in terms of these criteria.

The Developed Software

We devised and developed a Java code library for the visualization of the flows within a supply chain residing in a bounded geographic region. We used the Eclipse as the IDE (Integrated Development Environment) for developing our Java code. There are a number of reasons behind our decision to choose Java as the programming language for our supply chain visualization tool:

1. The cross-platform character of the Eclipse allows us to build and deploy our software across multiple platforms. For instance, our cross-platform application may be run on Microsoft Windows, Linux and Mac OS X.
2. Open source characteristic of Java appeals to us in terms of its availability, accessibility of detailed documentation free of charge and large developer communities sharing common goals and interests.
3. Parallel to this large Java community, we also consider the extensive libraries and packages that can boost the development process of any application written in Java.
4. A prospective integration of our visualization tool into professional ERP software is another reason for the selection of Java. As a result of a close relationship with a leading ERP software developer of Turkey, the project team plans to embed the proposed visualization tool into the distribution module of a widely-known ERP software in Turkey.
5. The existence of very effective, high quality and freely available IDEs to develop applications in Java (i.e. Eclipse, Netbeans...etc.).
6. The expertise of the project team in Java is the last important factor.

The structure of the geographic data related to any geographic region is very simple to gather and use. Within the framework of a previously completed project by some of the freshmen students at Sabanci University, we collected x - y coordinates for the region border and cities for Europe and Turkey in the form of a text file in the UTF-8 format (which allows characters from the Turkish alphabet). For storing the supply chain specific data, another text file is necessary with three data fields for each flow: the city from where the flow emanates, the city into where the flow enters and the amount of flow.

We constructed four Java classes (i.e. **MainClass**, **MapFrame**, **MyFileReader**, **SolutionFileReader**) to build the structure of our software. Figure 3 shows the UML class diagram of our software library, representing the collection of classes and the detailed system design. **MainClass** is the main class of the program. It gets two input text files as the inputs and creates an object from the **MapFrame** class. The **MapFrame** class uses **MyFileReader** and **SolutionFileReader** classes to read the text files and get the necessary geographical and supply chain design information, respectively. **MyFileReader** and **SolutionFileReader** classes are both constructed to read text file content line by line. They include two split methods, namely **splitMethod** and **splitMethod2**, in order to split the data in each line. The data is then saved into one dimensional array. The **MyFileReader** reads the map information in terms of border line coordinates, city names and city coordinates. The **SolutionFileReader** reads the solution text file that includes the specific city names where through which transportation takes place and the flow rate information. Examples of the supply chain design text files are provided in the Appendix A1 and A2 for two different scenarios.

After obtaining the arrays that include the information stored in the text files, **MapFrame** class calls the **scale** method. This method detects the maximum and minimum coordinates of the given map and scales all coordinates up to a numeric value as defined by the user. An algorithm is developed to make the appropriate scaling. The algorithm works by finding the ratio between the maximum x and y coordinates. So two different scale factors are calculated to scale x and y coordinates of the overall map independently. We design and implement the code of our

program in such a way that all information is accessible and the user can change some specific parameters such as object colors, line thicknesses, window sizes, and scale factors conveniently.

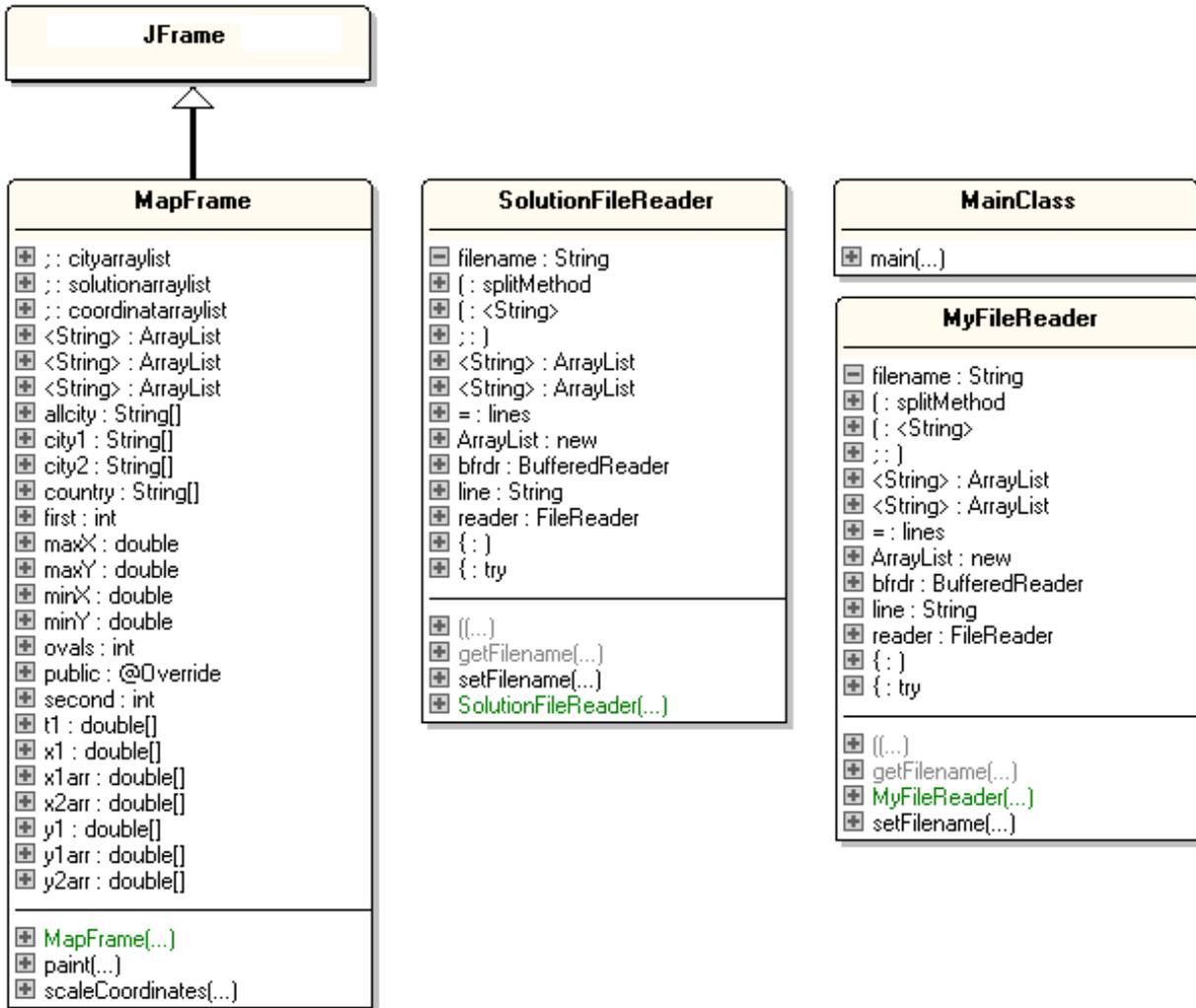


Figure 3. UML Class Diagram of the Software

The UML sequence diagram illustrating how objects interact with each other during a program run is given in Figure 4. The UML sequence diagram facilitates the understanding of how our system actually works and how we design the relationship of objects with one another. The rectangular boxes represent the object instances. The dashed lines, called *life lines*, indicate the object's life during interaction. An interaction between two objects is performed when a message

is sent from one object to another. A message is represented by an arrow between the life lines of two objects. The flow of objects' operation calls in Figure 4 gives the flow of object interaction. For instance, **MyFileReader** object is created by the **MapFrame** object. **MyFileReader** object interacts with the database (i.e. text file) and it returns a **mfr** object of its class. Similarly, **SolutionFileReader** object is created and it returns **sfr** object after the text file is read. These two objects, **mfr** and **sfr** are used by the **MapFrame** object and the map of the region and the specified supply chain are drawn by using the data obtained from these objects.

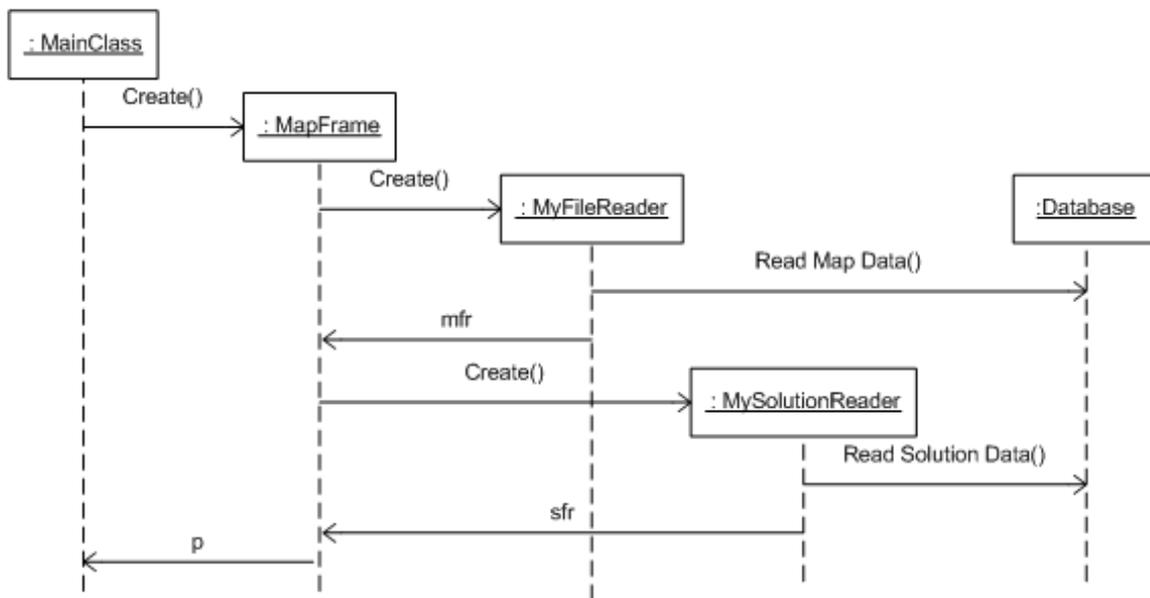


Figure 4. UML Sequence Diagram

It is also possible to follow the order of behavior shown in the system by using a UML activity diagram (Figure 5), which is similar to the flow chart. We give each step in the visualization of a supply chain network as an action in the activity diagram. For instance, the user has to collect map data or get the supply chain design solution at the first place. These two actions can be handled simultaneously and they can be completed independently. However, the subsequent

action (i.e. forming the text files storing map and solution data) cannot be started before these two actions are completed. In a similar manner, the flow of all actions necessary to visualize a supply chain design problem is given in the activity diagram.

In order to illustrate the contributions of our software to supply chain visualization, we describe the use and the output graphs of three different supply chain scenarios.

Visualization 1: Tire cord exports from Turkey

The first visualization shows the export of tire cord from Istanbul (where the Sabanci Group headquarters are located) to some European countries and is given in Figure 6. In this case, we were motivated by an earlier study completed in Sabanci University, where a spreadsheet optimization model was developed to find the optimal material handling policy at the KordSA production facility. KordSA, founded in 1972 by the Sabanci Group of Turkey, as a manufacturer of yarn and tire cord, currently operates in seven countries with nine manufacturing facilities.



Figure 5. UML Activity Diagram

The most of the finished tire cord products export from Turkey to European countries is achieved by KordSA and the deep understanding of extent and volume of these exports is important for strategic level supply chain design and planning in the company. That is why we decided to visualize the tire cord exports from Turkey to the European countries, which is a significant part of Turkey's global tire cord exports. The dataset is compiled from the Comtrade web portal of United Nations (Comtrade), which lists trade data between countries of the world. The visualized data is given in Appendix A1. In this visualization scheme, each country is represented by its capital city (except Turkey) and the flow between two countries is represented with solid lines with the thickness varying according to the amount of the associated flow. For example, the supply chain visualization in Figure 6 clearly illustrates that the largest amount of tire cord export from Turkey to Europe is realized to Germany and then to Italy, while the exports to the other countries are at similar levels.

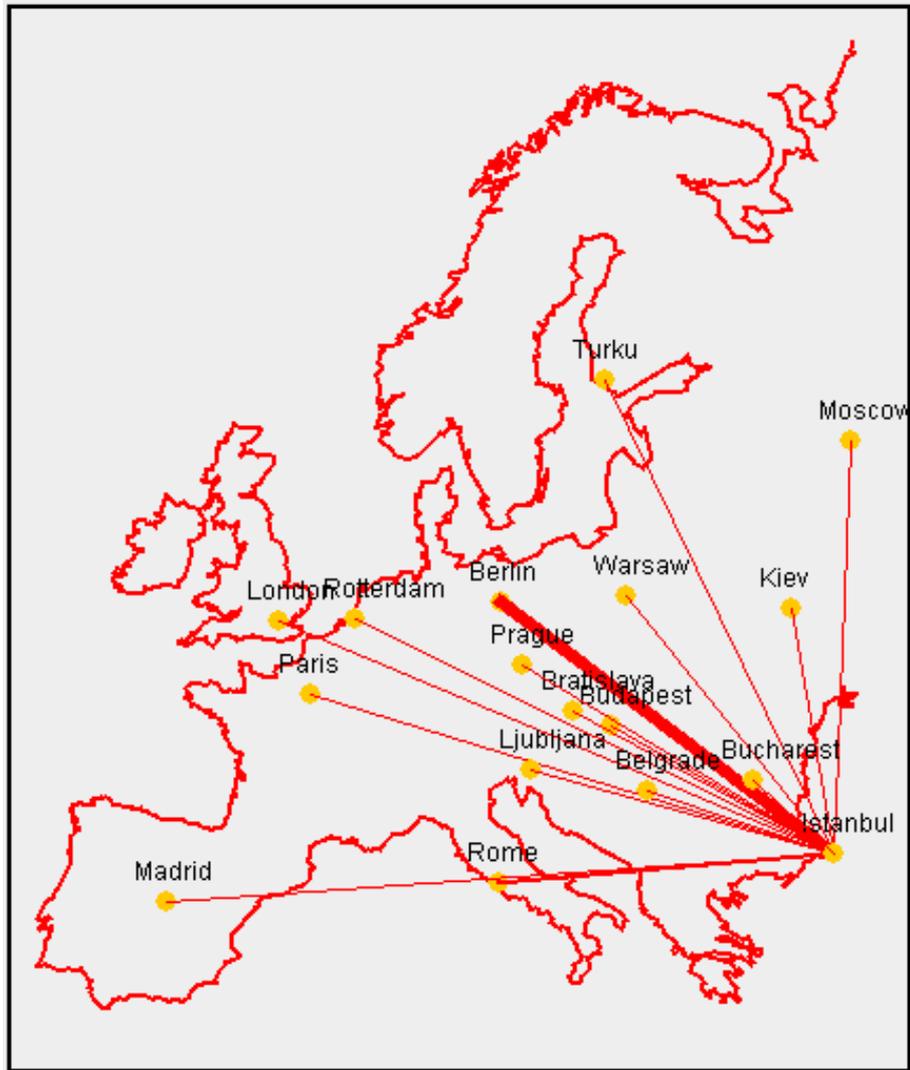


Figure 6. Tire cord exports of Turkey to some European countries

Borders of the European map are created depending on an origin point; distances between each point in the border lines to the origin point are measured and included in the “europe_map.txt”. City coordinates are also measured according to that origin point. In the algorithm the user can also scale the map by setting a scaling constant a priori. That is, the user can decide size of the map by choosing an appropriate value for this coefficient.

Visualization 2: Delivery routes of vehicles

In addition to the visualization of a supply chain network on the European map shown in Figure 6, we also illustrate the output of our software by visualizing delivery routes in Turkey. The geographical data related to Turkey were collected in a similar manner with the European map. The border lines were examined in very small segments such that these short straight lines would constitute the whole borderline for Turkey. The x and y coordinates for the start and end points of the each line segment are again stored in a text file. Different from the European map data, the x and y coordinates for all provinces and administrative districts were retrieved and are also stored for Turkey. Thus, more extensive information can be visualized for any region within Turkey. The visualization of the solution of a fictitious traveling salesman problem in Turkey is given in Figure 7. In this solution, the delivery tour begins and ends at the Southeastern megacity of Diyarbakir. The corresponding data is available in Appendix A2. Consistent with the requirements imposed by a traveling salesman problem, each of the ten cities in the model is visited exactly once and the tour ends at the initial node. The varying thickness for the flow values implies that the amount of flow entering and exiting a city is not the same. This is consistent with each flow value given in the input file, which is given in Appendix A2. In this sample solution, a large percent of the truck's load is delivered at Istanbul and the truck travels the return trip from Istanbul to Diyarbakir almost empty. A new visualization can be easily generated by simply changing the names of cities in the supply chain design solution text file.

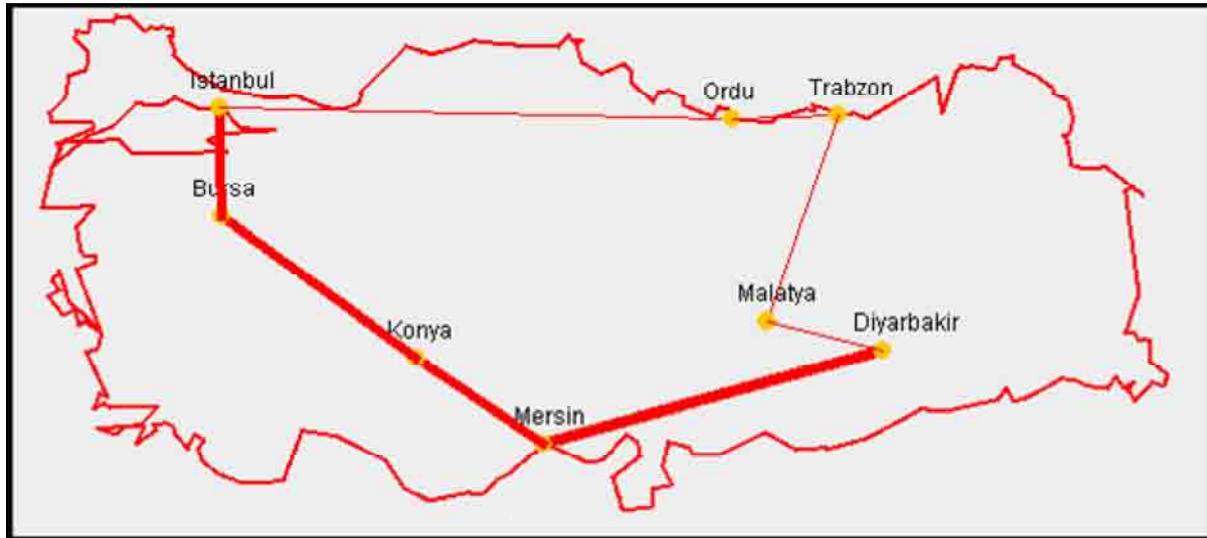


Figure 7. Visualization of a fictitious traveling salesman problem solution in Turkey.

Figure 8 gives the visualization of the solution of a fictitious vehicle routing problem, again referencing certain Turkish cities. The corresponding data is available in Appendix A3. Consistent with the requirements imposed by a vehicle routing problem, each of the cities in the model is visited exactly once and each tour ends at the same node it starts (i.e. the depot located in Istanbul). In this specific scenario, the graph presents two different routes from Istanbul passing through several cities and going back to Istanbul with the objective of minimizing total travel distance subject to constraints. The same thickness for all flow values implies that the amount of flow entering and exiting a city is the same. This tells us that for this VRP, the amount of demand at each city was either zero, or was not visualized.



Figure 8. Visualization of a fictitious vehicle routing problem solution in Turkey.

Visualization 3: Supply chain networks in America

In order to extend the applicability of the visualization tool, geographical data for South and North America are also collected. The visualization of a fictitious supply chain network is shown for the whole continent in Figure 9. Similar to the previous visualizations, the geographical data includes city names for each state. Supply chain networks can be illustrated efficiently using the software facilitating strategic and tactical level decision making.



Figure 9. Visualization of a supply chain design problem in North and South America

In Figure 10, the supply chain visualization is performed for the United States. Considering the trade volume of the United States with nearby countries, the visualization of the Canada, Mexico, and the United States is provided to represent the North American Free Trade Agreement (NAFTA) countries (Figure 11). Since NAFTA eliminates the majority of tariffs in product flows between United States, Canada and Mexico, a huge volume of trade is observed within the region. It is apparent that focus on supply chain design for this trade zone can significantly improve the efficiency of the supply chain operations, which is the motivation for our focus on the visualization of the geographical data for NAFTA.

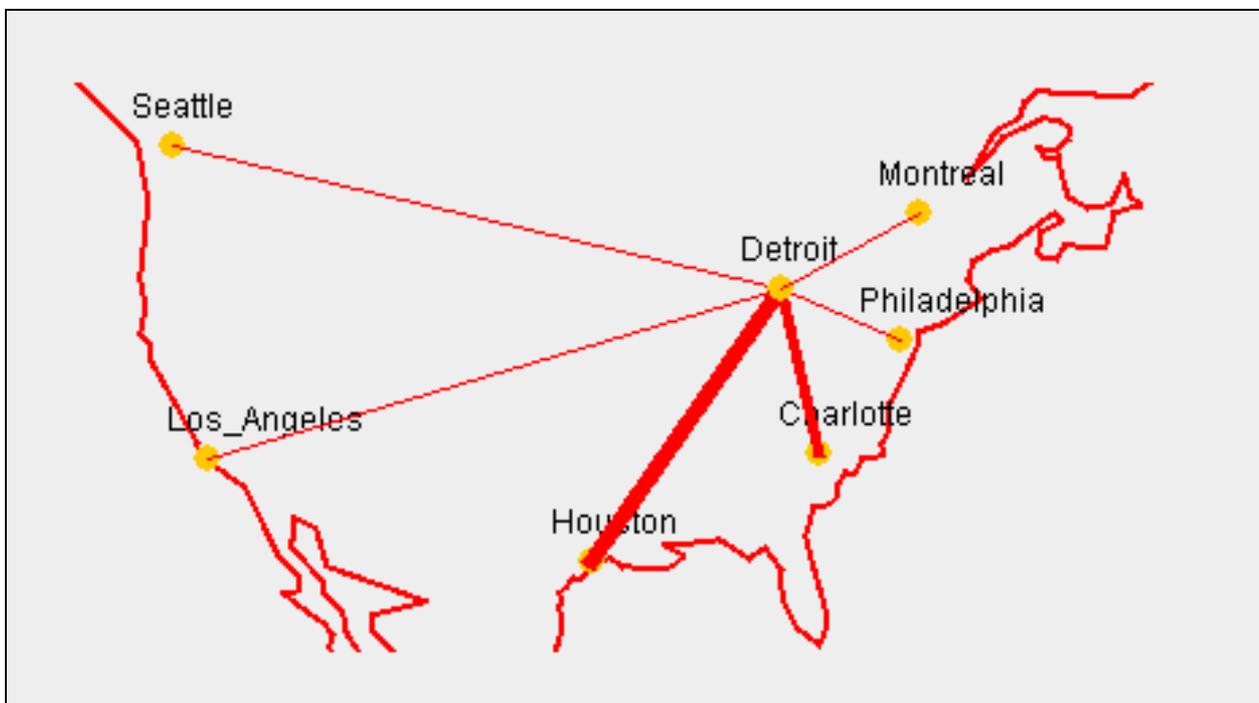


Figure 10. Visualization of a supply chain design problem in the United States

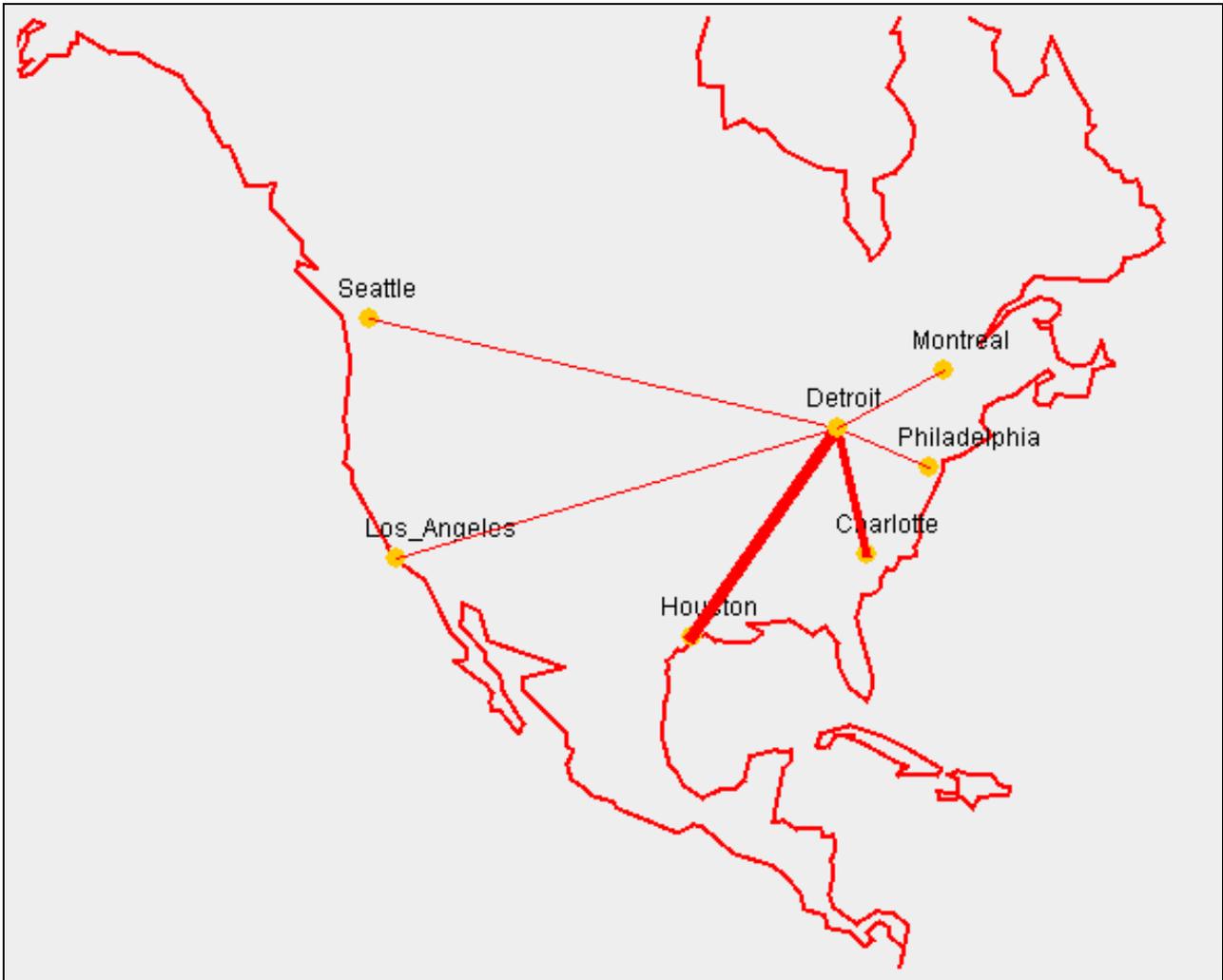


Figure 11. Visualization of a supply chain design problem within NAFTA

Conclusion and Future Research

In this study, we present a simple yet effective software library for supply chain visualization mapped to geographical regions of Europe and North and South America. Turkey is visualized separately providing more detail in the geographical data such as all city and district names. In other words, the visualization tool addresses the users who are involved with supply chain design within America and Europe regions. Considering the economic development level, the

demographic structure of the region (i.e. cultural interactions, historical background) and current level of product flows within these regions, it is possible to conclude that our visualization software can be an important tool for supply chain researchers and practitioners. Conveying the ideas in logistics, sharing the solutions of supply chain design problems and interpreting these solutions will be more efficient in terms of quality and time. There is no doubt that a supply chain network solution will have more impact compared to the one which has only a list of numbers.

In the implementation of our supply chain visualization tool, the main idea was to devise a system that is both useful and easy-to-learn. We investigated a significant number of open source and commercial software tools in this domain and find out that many of them require command of fairly complex geographical data structures to learn and implement. In order to devise and develop a simple and easy-to-understand geographical visualization tool, border lines for a region are divided into small pieces and the starting and end points of each of these small pieces are determined. Having stored this relevant yet extremely simple geographical data for the solutions of supply chain problems is visualized on a geographical map.

Although our visualization tool combines simplicity and ease-of-use in a single framework, there is still significant room for improvement without scarifying these two goals. Future research can focus on the addition of further characteristics to the software. The further contributions to this study can evolve around the topics given below:

- It is possible to extend the visual performance and appearance of the software by adding interactive zooming, panning and selection.
- The test of the visualization tool with the real world data can provide valuable insights in the strategic and tactical levels for a company.
- The integration of the visualization tool into a well known ERP software in Turkey is an important goal of the project team. The visualization tool is fully compatible with the considered ERP software since both are written in the Java programming language.

- An extensive documentation is essential for the wide-spread adoption of the visualization library by large audiences. In order to reach a large group of professionals and researchers in supply chain area, authors plan to post the software and its extended documentation in several languages on the project web site www.opendeclstions.org.
- Information visualization is a growing research area with advances in information systems and hardware technology. It is possible to find studies in the information visualization literature, combining the temporal and spatial data. For example, Kapler and Wright et al. (2004) analyzes observations over time and geography. The spatial interconnectedness of information over time and geography is achieved with an interactive and three-dimensional view. In this aspect, our software is open to any improvement in terms of information visualization.

Acknowledgments

An earlier report on our study has been awarded with the 3rd place from among approximately 25 groups in the Industrial Engineering Student Symposium (İTÜ EMÖS) organized at Istanbul Technical University (2007). The project is based on the PROJ 102 projects of undergraduate students Yener Kızılın, Talha Boz, Emre Adnan Işık, Hüseyin Çağlar Su and Funda Şentürk who created the geographical datasets. Authors would like to thank Dr. Burcin Bozkaya for his contributions in the supply chain visualization literature survey and to Ahmet Demirelli for his support in coding the Java program.

References

Aarts, E., Lenstra, J.K, 1997. Local Search in Combinatorial Optimization, E. H. L. John Wiley and Sons, 1st edition.

ArcIMS: <http://www.esri.com/software/arcgis/arcims/index.html>

Bramel, J., Simchi-Levi, D., 1997. The logic of logistics: theory, algorithms, and applications for logistics management, Springer, 1st edition.

Câmara, G., & Onsrud, H., 2004. Open-source geographic information systems software: Myths and realities. In J.M. Esanu & P.F. Uhler (Eds.), Open access and the public domain in digital data and information for science: Proceedings of an International Symposium (pp. 127-133).

Camm, J. D., Chorman, T. E., Dill, F. A., Evans, J. R., Sweeney, D. J., Wegryn, G. W. 1997. Blending OR/MS, judgment, and GIS: Restructuring P&G's supply chain. *Interfaces*, 27: 128-142.

Comtrade: <http://comtrade.un.org>

ESRI (1995) Environmental Systems Research Institute: Understanding GIS-The ARC/INFO Method, GeoInformation International, United Kingdom, i, 1-10.

Foresman, T.W., (1998) The History of Geographic Information Systems: Perspectives from the Pioneers. Prentice-Hall, Upper Saddle River, NJ.

GeoCatalogo : <http://sourceforge.net/projects/geocatalogo/>

GeoMedia WebMap : <http://www.intergraph.com/gmwm/default.asp>

GeoTools: <http://sourceforge.net/projects/geotools/>

G. Laporte, M. Gendreau, J.Y. Potvin and F. Semet, Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research* 7 (2000), 285 – 300.

Gutin, G., Punnen, A.P., 2002. The traveling salesman problem and its variations, Springer, 1st edition.

GRASS: <http://grass.itc.it/>

Kapler, T., Wright, W., 2005. GeoTime Information Visualization, Information Visualization Journal, Palgrave Macmillan, 4,(2), 136-146.

Korte, B., Vygen, J., 2005. Combinatorial optimization: theory and algorithms, Springer, 3rd edition.

Java: <http://java.sun.com>

MapXtreme: <http://extranet.mapinfo.com/products/overview.cfm?productid=1849>

MapServer: <http://mapserver.gis.umn.edu/>

MapWindow GIS: <http://www.mapwindow.org/index.php>

MIG Editor: http://sourceforge.net/project/showfiles.php?group_id=208145

Nasr, M. R., 2007. Open Source Software: The Use of Open Source GIS Software and its Impact on Organizations. Middlesex University / MIT.

Nardi, M. G., Sperry, S. E., Davis, T. D., 2007. Grain Supply Chain Management Optimization Using ArcGIS in Argentina, 2007 ESRI User Conference Proceedings.

NRDB Pro: <http://www.nrdb.co.uk/nrdbpro/>

OpenGeoDB: <http://sourceforge.net/projects/opengeodb/>

QuantumGIS: <http://www.qgis.org/>

Ramsey, P., 2007. The State of Open Source GIS, Refrations Research Inc.

Smotrisky, Y., 2004. Running GIS on Open Source, Connect: Information Technology at NYU, Fall/Winter 2004, 17-20.

Thill, J.C., 2000: Geographic information systems for transportation in perspective. Transportation Research C 8: 3-12.

Toth, P.,Vigo, D., 2001. The vehicle routing problem Society for Industrial & Applied Mathematics ,1st edition.

APPENDIX

The structure of used solution files were shown in this section. All .txt files are generated from MS Excel by using tab eliminated save method.

A.1 Supply_KordSA.txt

City1	City2	Flow Rate
Istanbul	Berlin	42,499,402
Istanbul	Paris	12,816,371
Istanbul	Moscow	10,988,897
Istanbul	Rome	18,237,952
Istanbul	Madrid	13,675,921
Istanbul	London	3,357,640
Istanbul	Warsaw	1,085,154
Istanbul	Budapest	198,039
Istanbul	Turku	8,073,750
Istanbul	Ljubljana	990,522
Istanbul	Bratislava	1,399,283
Istanbul	Rotterdam	3,725,216
Istanbul	Prague	6,193,788
Istanbul	Belgrade	6,348,360
Istanbul	Kiev	236,918
Istanbul	Bucharest	3,554,569

A.2 supply_turkey_tsp.txt

City1	City2	Flow Rate
Diyarbakır	Mersin	5000
Mersin	Konya	4800
Konya	Bursa	4600
Istanbul	Ordu	4400
Ordu	Trabzon	500
Trabzon	Malatya	150
Malatya	Diyarbakır	50