Vehicle routing problems with the use of multi-agent system

Lukasz Chomatek, Aneta Poniszewska-Maranda

Institute of Information Technology, Technical University of Lodz, Poland,
lukasz.chomatek@p.lodz.pl, anetap@ics.p.lodz.pl

Abstract. Increasing number of vehicles on the roads caused the increase of the popularity of the GPS devices that the drivers can install in their cars. Internet sites, where it is possible to compute efficient route from one point to another are also popular, because they are not only maps but also can fulfill some needs of the user.
The paper describes the proposition of solving the path finding problem with the use of multi-agent system. The idea of multi-agent system includes cooperation between autonomous software agents to complete a certain task.

1 Introduction

Increasing number of vehicles on the roads caused the increase of the popularity of the GPS devices that the drivers can install in their cars. Internet sites, where it is possible to compute efficient route from one point to another are also popular, because they are not only maps but also can fulfill some needs of the user. The user's needs can be divided into some groups:

– saving of time - user of the system only needs to know the destination address and in some cases he needs to enter the start point as well because the path is computed automatically by the system,
– finding some information - system can show the way to the nearest restaurant, gas station or shop,
– informing about the situation on the road, e.g. traffic jams, speed cameras, road works, accidents.

Both mentioned websites and GPS devices have to execute large numbers of queries about the route between points on the map. The website is a service dedicated for the large number of users and GPS device has to reflect dynamically changing road conditions (i.e. driver was supposed to turn left on the crossroads but went straight and now the system must compute a detour). Large number of queries can only be handled when either users' requests can be processed in a longer time or by use of very efficient path finding algorithms.

The most efficient algorithms for solving Single Source Shortest Path (SSSP) are hierarchical approaches [1]. They are usually based on the fact that some road segments can be marked as having higher importance than others. What is more, road network can be preprocessed by removing some nodes and introduce some shortcuts instead (i.e. there is only one connection from one node to another, so all nodes between them can be substituted by a direct link for this nodes).

The paper describes the proposition of solving the path finding problem with the use of multi-agent system. The idea of multi-agent system includes cooperation between autonomous software agents to complete a certain task. For solution of SSSP problem based on road network hierarchy,the agents can be divided into some groups: graph constructing agents, agents interacting with the system user and miscellaneous agents. The second significant term in the domain of multi-agent systems is the environment, in which the agents are located. In the case of road traffic it is very well defined and contains hardly any subjective factors. It includes vehicles, roads, road signs and signals and some important places which are usually named points of interest (POI). Road environment is obviously dynamic, due to the fact that hardly any part of it remains unchanged for a long time.

2 Approaches of hierarchical single source shortest path

Typical algorithms designed for solving SSSP problems do not use any preprocessing of the graph. Preprocessing phase can take a long time, so that such algorithms can be easily applied, when there is a little number of queries about the shortest path. Most popular SSSP solving algorithms are Dijkstras algorithm, Bellman-Ford algorithm and A* algorithm.

Hierarchical algorithms include some kind of preprocessing of the graph in order to shorten the time required to process a single query. It is notably important when number of queries is very high and sometime can be expended before deployment of the system.

The algorithm of Hierarchical Path Views proposed in the literature [4, 5] was based on the following ideas:

- base road network was split into some fragments - places of split were chosen by its geographical coordinates,
- connections which are outside generated fragments belong to higher hierarchy level,
- division and level transfer is an iterative process.

The result of such a division are the matrices containing the shortest path lengths for each segment and each level. After the division phase, to perform a query, A* algorithm was used.

Base for other branch of hierarchical algorithms for solving SSSP problem was Highway Hierarchies algorithm proposed in [1, 2]. Dijkstras algorithm is used in the preprocessing phase to calculate the neighborhood for each vertex. Next, the vertices that fulfill some criteria are moved to the higher hierarchy level. When this phase is done, the higher hierarchy level is preprocessed that allows to generate shortcuts between certain vertices. Number of hierarchy levels and size of the neighborhood are parameters of the algorithm. Proper choose of them influences on the amount of time needed to process a single query.

Some parts of the construction phase of Highway Hierarchies algorithm can be performed concurrently. We decided to try performing division of road network graph, so that Highway Hierarchies algorithm can be performed on a single part of this graph. After completion of the algorithm on each part, all subgraphs should be merged to obtain a final Highway Hierarchies graph.

3 Multi-agent system for road network hierarchization problem

The standards of architecture for multi-agent systems were described by FIPA organization [7]. Due to this specification, multi-agent system consists of some number of Agent Platforms that were as a parts of the system and they can be used to host the agents. Each Agent Platform consists of three parts to handle the management of agents:

– Message Transport System (MTS) that is supposed to be used by the agents for communication,
– Agent Management System (AMS) that represents a catalogue of existing agents,
– Directory Facilitator (DF) that stores the information about the services provided by the agents.

The analysis of modern programming techniques shows that some practices can be applied in newly designed multi-agents systems:

– use Service Oriented Architecture (SOA) to simplify and improve the possibilities of agent communication,
– make Directory Facilitator the mandatory part of multi-agent system,
– try to apply the enterprise design patterns such as dependency injection to coordinate the communication of agents on a single machine,
– simplify the architecture using the Windows Communication Foundation (WCF) [6, 7].

The introduction of web services allowed the developers to connect the applications based on different software and hardware platforms, for example Java and .NET Framework. The Web Services use a specific protocol to expose a schema of transferred data and allow the clients to make the synchronous calls of exposed methods [3].

The service oriented architecture is used in the process of creation of the multi-agent systems. It gives the system more elastic and moreover the interaction between the agents is simpler and its safety is on the upper level. Figure 1 presents the skeleton of proposed architecture of the multi-agent system based on the service oriented architecture. An agent uses the service directories: *Directory Facilitator* and *Agent Management System*, he makes accessible certain services and he profits by the services offered by other agents [3].

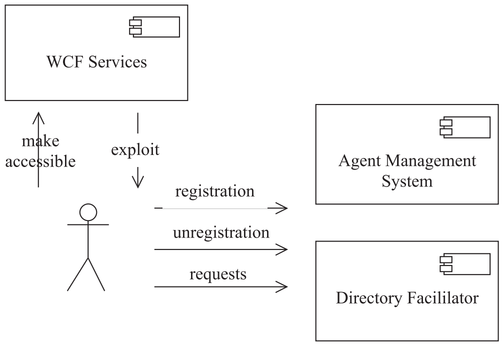

Fig. 1. Architecture skeleton of multi-agent system using SOA

Two main assumptions were made for proposed application of multi-agent system for building Highway Hierarchies graph:

- system must be able to take into account the user's preferences (i.e. route should be the shortest, traveling time should be lowest) and environmental conditions (i.e. weather, time of a day),
- computations should be done concurrently, where it is possible to be done.

To complete the first of these assumptions, weights of the road segments must be assigned using different criteria, such as length, average traveling time, speed limits, etc. It was decided to introduce some number of reactive agents that collect the data from different road segments. This type of agents can work in two different ways, depending on the data structure which is used to store the road network technology. The first way is associated with the nodes as it is easy to get information about edges connected to the node. Second way is related to edges. If list of edges in the graph is directly provided, it can be divided into some parts and each part can be analyzed by a single agent.

In our system both graph nodes and edges are kept in the separate lists. However references are duplicated and it simplifies the way of access to the needed data and allows the simple and complex weight assignment rules.

Regardless of the chosen solution, this process can be performed in parallel, what means sharing work for several agents. Depending on the selection criteria by which individual weights are calculated, work on each road section may perform one or more agents (each can calculate the weight using different method). If the weight of the segment is calculated on the basis of several criteria, use of a coordinating agent for the weights assignment process can be considered. The coordinating agent can calculate weight in accordance with certain rules (e.g. use the weighted average of the values calculated by the agents). Coordinating agent may have some adaptive abilities, depending on the application of the system.

Concurrent computation can be also applied in the other parts of Highway Hierarchies graphs creation process. Obviously, calculation of neighborhood for each vertex is independent of each other. The only nuisance is that for each vertex, different queue of vertices intended to be visited must be kept. Any number of agents can be used to calculate such a neighborhood. Depending on the developer choice, these agents cooperate directly with agents responsible for assigning weights to graph edges or with the coordinating agent.

The responses to user's queries for the system should take into account his preferences regarding the itinerary and the current conditions on the road. It might be necessary to create several Highway Hierarchies graphs, which will be used to obtain a system response depending on certain factors. Different graphs can be prepared for example for the city center during peak hours and at night. To implement this assumption, the introduction of a special type of agent can be considered. Such an agent will redirect the user query to the appropriate Highway Hierarchies graph.

Proposed architecture of multi-agent system described above is shown on figure 2. The tests revealed that for diverse criteria the calculated hierarchies differ very much. The results obtained for three proposed criteria: speed limits, traveling time and road length, shown that these hierarchies graphs have at each level only a few common edges with other hierarchies graphs. Moreover, expected convergence between dominating user's preference and number of common edges with the hierarchies graph for this criterion was observed.

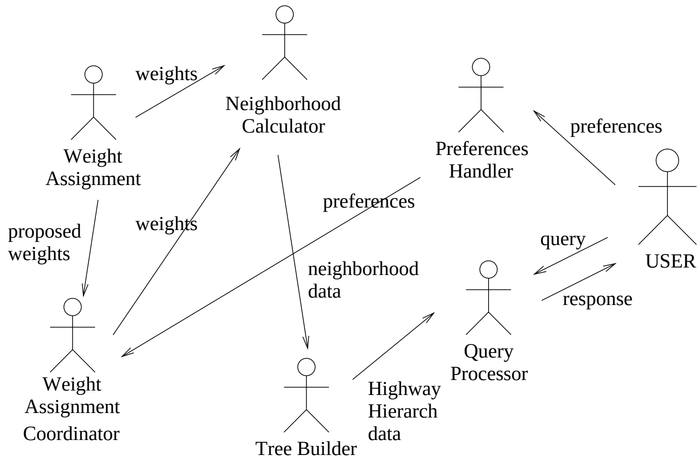

Fig. 2. Types of agents in Highway Hierarchies algorithm

Algorithm described above was implemented using C# 4.0 language in Windows environment. Tests were run on different maps both for single and split road network graph. In general, road segments on the top hierarchy levels in the full graph are on high hierarchy level in a part of the graph too, overall number on the highest hierarchy level is smaller for the subgraphs. It is caused by the fact that when the number of edges is smaller, promotion to the higher hierarchy level is harder. The tests performed for both maps shown that the time needed to execute the calculations depends in slight degree on the number of edges in the subgraph.

4 Conclusions

The presented paper focuses on the problems of efficient vehicle routing. Nowadays these problems are very important because of increasing number of vehicles on the roads. More and more drivers have the abilities of use the devices to support the planning of their itinerary and it is important to find the new solutions, new algorithms to improve the applications for finding the best routs taking into consideration different criteria, static and dynamic. The use of agent concept and use of multi-agent approach seems to be the interesting solution for solving the problems with vehicle routing and finding the optimal itinerary.

References

1. P. Sanders and D. Schulters, Engineering highway hierarchies. Lecture Notes in Computer Science , 4168, pages 804-816, 2006
2. P. Sanders and D. Schulters, Highway hierarchies hasten exact shortest path queries. Lecture Notes in Computer Science , 3669, pages 568-579, 2005
3. L. Chomatek and A. Poniszewska-Maranda, Modern Approach for building of Multi-Agent Systems. Lecture Notes in Computer Science , 5722, pages 351-360, 2009
4. G. Nannicini, P. Baptiste, G. Barbier, D. Krob and L. Liberti, Fast paths in large-scale dynamic road networks. Computational Optimization and Applications , 45 (1), pages 143-158, 2008
5. D. Eppstein, M. Goodrich and L. Trott, Going off-road: transversal complexity in road networks. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 23-32, 2009
6. C. Vasters, Introduction to Building WCF Services, MSDN Library, 2005
7. FIPA, Abstract Architecture Specification, 2002