Optimization of Taxi Cabs Assignment in Geographical Location-based Systems

Abilio A. M. de Oliveira[1], Matheus P. Souza[1], Marconi de A. Pereira[1], Felipe A. L. Reis[2], Paulo E. M. Almeida[2], Eder J. Silva[1], Daniel S. Crepalde[1]

[1]Departamento de Tec. em Eng. Civil, Computação e Humanidades
Universidade Federal de São João Del-Rei (UFSJ) - Ouro Branco, MG - Brazil

[2]Departamento de Computação
Centro Federal de Educação Tecnológica de MG (CEFET-MG) - Belo Horizonte, MG - Brazil

Abstract. *In this paper, different approaches are evaluated to assign taxi cabs to customers in geographical location-based systems. The main purpose of this work is to identify the solution in which all current customers are met in an acceptable time, however minimizing the distance traveled by existing free taxi cabs. Two aspects are considered: 1) the method to calculate the distance between vehicles and customers; and 2) a vehicle assignment strategy. The methods to calculate the distance between vehicles and customers are: a GPS-based routing (a shortest path algorithm) and the Euclidean distance. On the other hand, as vehicle assignment approaches, the considered strategies are: a greedy algorithm, which assigns each vehicle to the closest customer, and an optimization algorithm, which assigns vehicles considering the whole scenario, minimizing the global distance traveled by taxi cabs to meet the customers. This last strategy considers an optimization model in such a way that the calls are not readily answered. In this case, a short waiting window is implemented, when the calls are stored and then the optimization algorithm is executed, in order to minimize the required distance and to meet all current customers. The combination of the two methods of distance calculation and the two vehicle assignment strategies formed four possible approaches, which are evaluated in a realistic simulator. Results show that the approach which uses the shortest path algorithm and an optimization algorithm reduces the average service time by up to 27.59%, and the average distance traveled by up to 45.79%.*

1. Introduction

Taxi services are an alternative urban transportation mode which offers convenience and speed in relation to public transport. It is noticed that taxi service is not efficient because much of the time, about 50%, they are vague [Reis et al. 2011]. On the other hand, some passengers complain that some calls are missed. In Belo Horizonte, for example, about 10% to 15% of customers, who prefer the convenience of ordering a taxi by phone, give up because of the long wait for the service [Castelo-Branco 2012].

To make this service more efficient, several studies are being conducted and new attribution methods based on geographical location are being adopted [Reis et al. 2011]. These methods use GPS (Global Positioning System) to locate taxis and customers and propose some way to better assign each vehicle to attend the existing calls. Usually, two aspects are explored by these approaches:

1. The procedure used to estimate the distance traveled by a taxi to achieve a customer. Two very common possibilities are the Euclidean distance, and the shortest path between points. It is important to note that the evaluation of the distance between the taxi and the customer can be treated in a more elaborate way, taking also into account the route that the taxi will go, considering factors that complicate the taxi displacement, such as speed limits, traffic jams, among others;
2. The vehicle assignment strategy. A common vehicle assignment method is based on a greedy approach, where the taxi cab which is closest to a customer is designated to take the call. The intent is to minimize the passenger waiting time and to minimize the idle taxi time. Moreover, as all greedy heuristic, this solution can be a good one in the local space of solutions but can be a poor choice when considering global solution space. Thus, the problem can be modeled as an optimization problem, where it is wished to minimize the idle taxi time, treating the passenger waiting time as a restriction of this problem.

This paper aims to explore these two aspects, in such a way to measure the effectiveness of the most common ways currently adopted to estimate distance and to assign taxi cabs to customers. Also, this work intends to propose an optimization approach to solve this problem, to objectively evaluate its performance by means of micro-simulation and to draw some practical conclusion about the experiments undertaken.

The remaining of this article is organized as follows: in Section 2 a historical background is revised, and some study cases are briefly presented. In Section 3, the tools and methods used to simulate and evaluate the discussed approaches are detailed. A practical experiment and some simulation approaches are presented in Section 4. Results and discussions about the experiments are shown in Section 5. Finally, in Section 6, some conclusions are presented along with some future work suggestions.

2. Related Work

The use of mobile applications is increasing the productivity of resources, especially in the context of smart cities [Steenbruggen et al. 2015]. A constantly growing demand, combined with an increasing traffic complexity, made the task to identify the best taxi to attend a call to become very hard. This problem has been studied for some time around the world, showing that the problem is not exclusive to big cities, emphasizing the need for a solution to improve this picture.

In this sense, [Xu et al. 2005] describe a scenario in Shanghai, where taxis which joined Dazhong Company were modified and received GPS trackers, to indicate their location in real time. Customer requests arrive at a call center, which uses the trackers to know the nearest free taxi to answer each call, since the customer's location is also known. Then, the taxi drivers, within a certain range, respond to the call accepting it or not, via a button located on the equipment installed in the vehicles. Each consumer is then informed about how long it will take for his/her demand be met. The control of occupied and free taxi cabs and their distribution throughout the city is made in the dispatch center of the company itself (DZDC - *Dazhong Dispatching Center*), which provides a higher speed in attendance.

In the city of Singapore, the system that handles the taxi service is the Automatic Vehicle Location and Dispatch System (AVLDS). This system, like the one used in Shanghai, consists of taxi cabs sending their positions to a central station. Customers calls arrive at this central, which congregates all taxi companies in the city. The system assigns the closest vehicle to each customer and waits around 10 seconds for the taxi driver to accept or decline the call. If he/she declines, then the system performs a new search for taxis using a new request for that service [Liao 2009][Yang and Wong 1998].

In both cities, the adoption of the mentioned systems provided a reduction of 16% to 32% on traveled distances without a passenger, and a reduction of 15 to 30 minutes in the customer average waiting time. These numbers were obtained considering the scenario with the traditional method, where the call center sends a broadcasting message, and a taxi driver who believes he/she is closer to the passenger offers to take the call.

[Liu et al. 2015] directed their work to highlight the importance of communication between various taxi drivers, to predict where and when a customer will need a taxi. The past route information can be used in a machine learning process, combined to a optimization process in order to reduce the cabs idle time.

Conversely, [Santos and Xavier 2015] and [Jung et al. 2015] present a different practice that helps to reduce the idle traveled distance, using a cab sharing method (ride sharing). Each vehicle has a known location and all cabs are connected to a wireless network, so that the customers can find out if a vehicle will pass through the desired route. Then, they can share the same taxi cab with existing passengers. Therefore, this practice consists in helping both the driver and the customers, reducing both the global cost of travel and the overall traveled distances.

3. Tools and Methods

3.1. Tools

MATLAB (acronym for MATrix LABoratory) is a multi-purpose tool focused on numerical modeling, having matrices as its fundamental data structure. SUMO (Simulation of Urban MObility) is a simulation platform for discrete event traffic, microscopic, using continuous space and presenting inter- and multi-modal capabilities [Krajzewicz et al. 2012]. SUMO was chosen because it is open source software, very popular in works focused on traffic simulations.

The relationship between SUMO and MATLAB is based on TraCI4Matlab [Acosta et al. 2015], which is an API (Application Programming Interface) developed in MATLAB. TraCI4Matlab is an implementation of TRACI (Traffic Control Interface) protocol, which allows for interactions between MATLAB and SUMO in a client-server scenario. TraCI4Matlab can be used to implement vehicles control, traffic lights timing, intersections monitoring, among other applications of traffic for SUMO simulations.

3.2. Assignment Problem

Linear Programming Problems (LP) are those which present a linear objective function and linear constraints, regarding the existing design variables [Taha 2008]. Certain special cases of LP, such as those involving network flow and multi-commodity flow, are considered so important that they have generated several researches and specialized algorithms in order to solve them.

There is a special case of LP which is called the "Transportation Problem" [Taha 2008]. A common mathematical modeling approach to this case can be seen in equation (1), where m indicates the origins, n destinations, c_{ij} the cost per transport unit and x_{ij} the amount sent from i to j.

Minimize:

$$z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} * x_{ij}. \tag{1}$$

The solution of a Transportation Problem consists of determining x_{ij} values which will minimize the total cost of transportation and satisfy all constraints of demand and supply.

There is a special case of transportation problems, called "Designation Problem" (DP) [Taha 2008], which can be used in this context. The objective here is to systematically assign an available taxi cab to attend a customer, meeting some criteria z. One mathematical model to solve this problem can be seen in equation (2) [Goldbarg and Luna 2005].

Minimize:

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} * x_{ij}, \tag{2}$$

subject to:

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, ..., m; \tag{3}$$

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, ..., m; \tag{4}$$

for:

$$x_{ij} \geq 0, i = 1, ..., n; j = 1, ..., n. \tag{5}$$

DP can be understood as the problem of allocating n producing cells to n tasks at a cost c_{ij}. Each variable x_{ij} is binary, that is, its value is equal to 1 when the cell i receives the task j to be fulfilled; otherwise its value is set to 0 [Goldbarg and Luna 2005].

The solution of a DP can be obtained using the Hungarian Algorithm (HA) [Jonker and Volgenant 1986]. It works by analyzing a matrix consisting of j lines, indicating the number of calls to be answered, and i columns, representing each taxi cab available for service at any given cycle time.

In the context handled in this paper, the Assignment Problem is used to identify which vehicle will met each customer call. To do this, it is necessary to calculate the distance (or cost) of each taxi to meet each customer. It can be done by a classic shortest path algorithm, like the Dijkstra algorithm [Dijkstra 1959]. For all free taxi cabs, the shortest path to met the passenger is calculated; then, HA considers the costs of these paths to identify which is the best solution, considering the whole scenario and trying to minimize the criteria z.

In order to provide realistic calculations, Traci4Matlab library, coupled with SUMO, is used. It is important to note that these calculations go beyond to simply calculating the distance between sources and destinations. SUMO allows for a realistic simulation of the scene, where traffic signals, different speed limits, as well as the existence of other vehicles traveling on the streets, are simultaneously considered. This effect is similar to those GPS applications for smartphones, such as Waze[1] and Tomtom[2], which use traffic information to calculate the best route to reach a given destination.

3.3. Methods

It is possible to classify a taxi service in four ways, according to the presence or absence of information about geographical locations of taxis and passengers [Reis et al. 2011]. The first one, a "Random Searching" mode, occurs when a customer waits in a random location for taxis moving around on the way. This situation is characterized by the ignorance of positions of both, taxis and customers.

A second mode, "Fixed Stop", occurs when customers go after a taxi stop point. It is characterized by the knowledge of taxi cab positions, however the customer's initial position is unknown. The third mode is the "Broadcasting", which uses the geographical location of customers, but the positions of taxi cabs is unknown by the central control system. In an example of this mode, a customer calls the central service and the customer's location information is passed on by radio to the taxi drivers. The first driver to accept the request is confirmed by the central to pick up the customer. Since the position information of the available taxi cabs is unknown, this method does not guarantee that the closest driver or the most suitable vehicle is assigned to the customer.

The last mode is represented by "GPS-based models", which can identify the geographical position of both, customers and taxis. These GPS methods are distinguished from each other through the algorithms used to define the taxi driver which will be responsible for picking up each passenger. The geographical location of the actors involved underlies the criteria and costs used by the algorithms to choose the attendant to each request. Cost is a factor that determines the difficulty of a taxi driver to answer a call. Basically, two methods can be used: the euclidean distance and the routing distance between each customer and the available vehicles (the distance that each taxi driver must travel through streets and avenues to reach the customer). The following sections explain some features of GPS-based algorithms, exploring both the distance calculation methods between actors, as the vehicle assignment strategy.

3.3.1. Greedy Algorithm based on Euclidean Distance

This allocation method has the advantage of low computational cost required, compared to the next methods, due to the simplicity of the calculations involved. The operating principle is to calculate the distance between two points, and then finding the taxi cab which has the lowest linear distance from the customer, calculated through Equation (6):

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \tag{6}$$

[1] https://www.waze.com/
[2] http://www.tomtom.com/

where x_1 and y_1 are the coordinates of the customer and x_2 and y_2 are the coordinates of an available taxi cab.

For every customer call, the system checks what is the best candidate, using therefore a greedy approach. As can be seen in the Figure 1, not always the result of this method is the actual best. Due to the traffic structure of the cities, the euclidean distance is not equal to the distance to be traveled to reach the destination. However, this is a solution to be analyzed, since it has a very large computational gain when compared to other methods based on shortest path algorithms.

In the case presented in Figure 1, Taxi 1 is very close to Customer 1, considering the linear (Euclidean) distance. However, it must go through four different streets to meet the customer. On the other hand, Taxi 2 is more distant from the customer, considering the linear distance. Although, only three streets separate the taxi cab and the customer. This scenario is a small sample of the complexity that the context (traffic jams, traffic flow, speed limits) may transform the calculation of the cost in a non-trivial scenario.

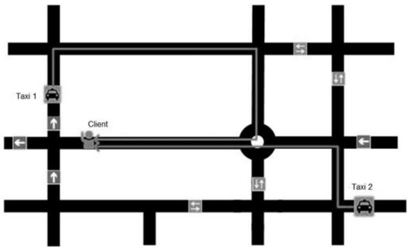

Figure 1. Problem of the Euclidean distance in the assignment of taxi cabs [Reis et al. 2011].

3.3.2. Greedy Algorithm based on Shortest Path

This variation of the previous greedy algorithm aims to partially repair the deficiency pointed by Figure 1. The operating principle is to calculate the actual distance, finding the taxi cab which has the lowest driving distance to the customer.

This method still uses a greedy approach, since, for each customer call, a taxi cab which has the shortest driving distance to the customer is selected to do the service. The following methods to be detailed, on the other hand, try to consider a global view of the scenario. In some way, they effectively try to minimize the total distance that vehicles have to travel through the streets, without a passenger, before reaching the customers.

3.3.3. Hungarian Algorithm based on Euclidean Distance

The Hungarian Algorithm (HA) is a popular procedure for solving assignment problems. In this work, it analyzes the matrix formed by customer calls collected in a time window of 50 seconds and the taxi cabs available to attend, so that the total cost of the chosen solution pairs is minimized. So, this is a very different approach from Greedy Algorithm. The cost factor used, as this model's name implies, is the Euclidean distance between customers and taxi cabs.

If the number of calls is less than the number of available vehicles, some taxis are not included in the search, but all calls are answered in such a way that the total cost is the lowest possible. The same goes for the opposite case, when the number of taxis is lower. Some calls are not answered on that iteration, always looking to find a solution which brings the lowest overall cost.

The evidence that HA determines the best assignment among taxi drivers and customers is given by the Great Allocation Theorem and the König Theorem [Konig 1931]. The Great Allocation Theorem says that if a real number is added or subtracted from all entries of a row or column in a cost matrix, then an optimal assignment for the resulting cost matrix is also an optimal assignment for the original cost matrix [Kuhn 1955].

3.3.4. Hungarian Algorithm based on Shortest Path

In this approach, the cost calculation used in HA is based on the actual distance to be traveled by the taxi cab to meet a customer. The calculation consists on the analysis of a matrix formed by available taxis and customer calls, on a time window of 50 seconds. Thus, the algorithm minimizes the total cost of taxi assignments. This optimization approach provides an improvement in the previous algorithm, since now the actual distance to be traveled by each driver is considered.

The use of HA does not imply a significant rise in processing time when comparing this approach with greedy algorithms. Its major issue is the need for a set of calls to improve the results, which demands the time window. The value of 50 seconds was chosen with the intention to avoid leading to excessive idleness of the vehicles, and conversely, a big amount of calls to be answered, keeping a trade-off duration during the processing of customer calls.

4. Experiment and Simulations

The four algorithms presented in Section 3.3 were implemented in practice: Greedy Algorithm based on Euclidean Distance (GED), Greedy Algorithm based on Shortest Path (GSP), Hungarian Algorithm based on Euclidean Distance (HED), Hungarian Algorithm based on Shortest Path (HSP). Simulations for each algorithm were carried out under the same conditions, considering a same number and an equivalent geographical distribution of taxi cabs and other vehicles. Likewise, customer requests are located in the same places and amount for all simulated algorithm. The choice of each taxi to meet the requests followed the criteria for each algorithm. Spent time and distances traveled by taxi cabs to service the requests are then made available by SUMO to MATLAB, allowing fair comparison among the methods.

The simulation environment tried to be as realistic as possible, considering two-way and one-way streets, traffic lights and other vehicles traveling freely. The streets and avenues cover an area of 54 km^2 (see Figure 2). Border roads were designated as rapid traffic routes, having a speed limit of 80 km/h. Vertical roads were considered arterial roads with a limit of 60 km/h and horizontal roads were considered collector roads with speed limit of 40 km/h.

The amount and types of vehicles which formed the traffic flow were set in propor-

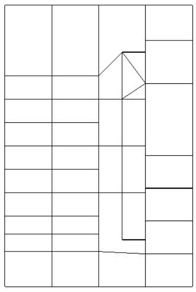

Figure 2. Simulation environment in SUMO.

tion to the environmental area of the city of Belo Horizonte [Castro 2014] [IBGE 2014]. The routes for all private (non taxi) vehicles were randomly determined. The simulation included: 9,333 private cars, 3,443 motorcycles, 155 buses and 450 taxi cabs, among which 225 were green (available) and 225 red (busy). The number of taxi cabs was defined according to the average value of cabs into each 0.12 km^2 in the city of Belo Horizonte [Reis et al. 2011].

Calls and taxi cabs were distributed uniformly and randomly throughout the network. The moments of occurrence of calls were also random and those are kept the same for all simulations. In order to better represent the dynamics of the system, taxi cabs have a specific area of initial actuation which is altered according to assignments to answer calls. Routes were created by vehicle, by Dijkstra least-cost algorithm, already implemented in an internal function of the simulator, after taxi-customer assignments were determined.

The relationship between the amount of available and occupied taxis is kept constant throughout the simulation, which keeps the tests with no tendency of increase or decrease of the taxi supply relative to the number of customer calls. This steady state ensures that waiting times for services keep up uniform, directly related to the amount of time the taxis run free, without customers [Yang and Wong 1998]. However, in actual life, for each day of the week, time, region, events, weather, etc., there are variations in the number of requests. So, for this set of simulations, it is assumed to be a normal day, with a constant number of requests, giving a better scenario for comparisons.

Another feature, not included on the simulations, were the concurrency requests. There is the possibility, in an actual environment, of a taxi driver having its status changed at the same time a call processing is initiated, making this driver unavailable for those particular calls being processed.

To statistically validate the results, the number of simulations conducted for each scenario was equal to 30, with a duration time of 4 hours, totaling 120 hours (or 5 days), to comply with Central Limit Theorem. On average, a taxi cab serves 15 calls per day, working for 13 hours (two or more drivers can use the same vehicle, in complementary time [Lopes 2014]). Therefore, for each simulation of 4 hours, a number of approximately 4 calls per taxi was randomly generated and distributed for both, instant of time and occurrence position, totaling 54,000 requests simulated for each algorithm.

The requests were generated until a predetermined final time was achieved, and the same conditions were repeated for each simulation and for each algorithm, considering place, calling time and traffic, so ensuring that the same factors were present under the same conditions, for each algorithm. Therefore, the algorithms defined the best taxi cab for each request, and the values of the waiting time and the distance were extracted and evaluated.

In our experiments, each simulation can be objectively described as follows:

- Vehicles are randomly distributed on the map: private cars, motorcycles and taxis. In each second, vehicles can enter to and exit from the network, simulating the traffic flow. However, the number of taxi cabs remains constant throughout the simulation;
- Half of the taxi cabs remain free and the other half occupied;
- Requests are made and the simulation is independently carried out in 4 scenarios, one for each algorithm. Each algorithm determines the best taxis, according to its own criteria, to meet each customer;
- A taxi cab state changes from free to busy and it is classified as *ongoing service*, avoiding to be assigned to other calls. A minimum cost algorithm determines the best route taken by the driver to the customer;
- To keep the ratio of free and busy taxis, when a taxi is assigned, another taxi currently classified as *ongoing service* becomes free;
- Once a taxi meets a customer, the distance and the traveled time spent to arrive to the passenger's location are stored.

5. Results and Discussions

The four algorithms were tested considering the same conditions for the simulation in SUMO, as presented before. The results could then be analyzed with assurance that comparisons are fair. As can be seen in Table 1, average and standard deviations of both attendance **time** and the total **distance** traveled were calculated for each algorithm, considering all 54,000 simulated calls divided into a total of 30 simulations performed for each algorithm.

The first result was the average waiting time for each call, which was measured from the time that the request was made by the client until the instant that a taxi arrived. For algorithms based on the Euclidean distance, it was observed that instantaneous distribution algorithm (greedy) was approximately 7.98% more efficient in attendance time, but the Hungarian Algorithm was 1.09% more efficient at average traveled distance to meet the customer. Variation of the average values for both, time and traveled distance, were very close. It was observed also that, when using the Euclidean distance as a cost criterion in the algorithms, the implementation of the Hungarian Algorithm does not become more efficient, not generating significant results.

Table 1. Distance and Attendance Time (Average $\pm$ Standard Deviation).

	Time	Distance(Km)
GED	4min36sec $\pm$ 5min48sec	3.3718 $\pm$ 3.0467
HED	5min $\pm$ 5min51sec	3.3352 $\pm$ 3.0531
GSP	2min28sec $\pm$ 2min11sec	2.0459 $\pm$ 1.9868
HSP	1min47sec $\pm$ 1min38sec	1.1090 $\pm$ 1.3777

The highlight is due to the algorithms based on the actual distance to be traveled between customers and taxi cabs. The Greedy Algorithm, the worst based on the shortest path, showed up a gain of 46.48% in the average attendance time, when compared to GED, and a gain of 38.66% in average distance traveled, when compared to HED.

Now, the proposed algorithm to optimize the taxi service process, using Hungarian Algorithm (HA), presented an average attendance time of 27.59% lower than Greedy Algorithm (GA). As for the total distance traveled in attendance, HA presented an even better result, approximately 45.79% below. Another point to note is the stability of responses observed through standard deviation estimations. HA showed up a variance around 25.10% lower than GA in service time and a variance 30.66% lower than AG, when comparing the average distances traveled.

Considering the algorithms based on shortest path, these results mean an average gain of 0.9369 Km for HA in relation to GA in a single request of attendance. As 54,000 calls were simulated, the total gain was around 50,620 kilometers relative to GA. The result becomes even more significant considering, for example, the city of Belo Horizonte, where around 96,000 calls happen in a day [Reis et al. 2011]. In this scenario, HA would imply a gain in relation to GA by approximately 89,991 kilometers daily.

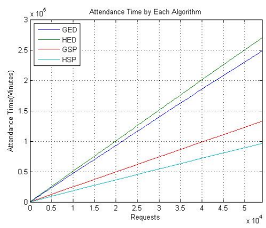

Figure 3. Sum of the attendance time by each algorithm.

Figure 3 shows another view of the obtained results, detailing the sum of the attendance times for calls. The algorithms based on the Euclidean distance presented a big difference when compared to algorithms that use actual shortest path. GED, however, showed up a total gain of 21,400 minutes compared to HED. In the graphic of distance traveled, shown in Figure 4, the discrepancy between the lines of the algorithms based on the Euclidean distance is smaller, with a lead of 2,000 kilometers for Hungarian algorithm (HED).

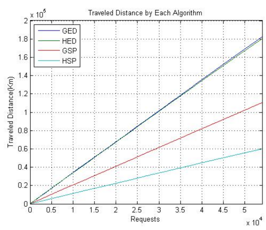

Figure 4. Sum of the traveled distance by each algorithm.

The algorithms based on the shortest path stood out. The total attendance time of GSP is similar to the attendance time of 29 thousand requests of GED. GSP exceeded HSP in 36,680 minutes. The total time of HSP service (to 54 thousand calls) is similar to the attendance time of 39 thousand requests of GSP.

Even more significant are the results found for the traveled distance (Figure 4). The total traveled distance by taxi cabs to all calls of GSP amounts to the same distance of HED in approximately 32,860 calls. HSP again achieved the best result. It was smaller than GSP in 50,620 kilometers. Its total distance is similar to the distance of GSP in 29,300 calls. So, HSP would save 50,620 kilometers of travel, which is equivalent to 24,700 requests.

6. Conclusion

The experiments included the choice of taxis from the Greedy Algorithm based on the Euclidean Distance (GED), the Greedy Algorithm based on Shortest Path (GSP), the Hungarian Algorithm based on Euclidean Distance (HED), and the Hungarian Algorithm based on Shortest Path (HSP). GED and GSP can be considered the most popular algorithms, because they resemble the most used form of order a taxi: either via radio or via smartphone, the vehicle assignment systems try to identify an available taxi cab which is closest to the customer, so that it can answer the call. The use of an optimization algorithm, like HA, for the taxi cab assignment process achieved positive results (when using actual distance) when compared to instantaneous selection algorithm, based on greedy strategies.

Simulations were based on a map with some complexity presented in a small and a medium city (different speed limits, traffic lights, two-way and one-way streets, and so on). Additionally, the scenery simulation represented normal hours of service, with usually lower demand than the number of available taxis. Also, the time window used was arbitrarily chosen to be 50 seconds, and in non ordinary days, such as rainy days or special events (concerts, sports), this interval could have a different value. It is expected that in more complex maps, the difference between the algorithms evaluated become even more significant.

Current results are promising, since the possibility of involving the Designation Problem with different quantity and a more complex distribution of calls. This work directed the research focus on taxis but those algorithms can have applications in different situations, as fleets of private firms and even on autonomous cars research field.

Acknowledgments

The Authors would like to thank CAPES Foundation, under Grant 10224-12-2, for the financial support.

References

Acosta, A., Espinosa, J., and Espinosa, J. (2015). Traci4matlab: Enabling the integration of the sumo road traffic simulator and matlab® through a software re-engineering process. In Behrisch, M. and Weber, M., editors, *Modeling Mobility with Open Data*, Lecture Notes in Mobility, pages 155–170. Springer International Publishing.

Castelo-Branco, A. (2012). Demora no atendimento de táxi em bh leva 15% dos passageiros a cancelar pedido. [Online; Access Date: 5 julho 2014].

Castro, C. M. d. (2014). Viaduto da floresta tem 40mil carros por dia, diz pesquisa origem destino. Website. Disponível em: http://g1.globo.com/minas-gerais/noticia/2012/10/viaduto-da-floresta-tem-40-mil-carros-por-dia-diz-pesquisa-origem-destino.html.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Goldbarg, M. C. and Luna, H. P. L. (2005). *Otimização combinatória e programação linear: modelos e algoritmos*. Elsevier, Rio de Janeiro, RJ, Brasil.

IBGE (2014). Área territorial de bh. Website. Disponível em: http://www.ibge.gov.br/home/geociencias/areaterritorial/area.php?nome=Belo

Jonker, R. and Volgenant, T. (1986). Improving the hungarian assignment algorithm. *Operations Research Letters*, 5(4):171–175.

Jung, J., Jayakrishnan, R., and Park, J. (2015). Dynamic shared-taxi dispatch algorithm with hybrid-simulated annealing. *Computer-Aided Civil and Infrastructure Engineering*.

Konig, D. (1931). Gráfok és mátrixok. matematikai és fizikai lapok, 38.

Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. (2012). Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Liao, Z. (2009). Real-time taxi dispatching using global positioning systems. *Communications of ACM*, 46(5):81–83.

Liu, S., Wang, S., Liu, C., and Krishnan, R. (2015). Understanding taxi drivers' routing choices from spatial and social traces. *Frontiers of Computer Science*, 9(2):200–209.

Lopes, V. (2014). Bhtrans fecha o cerco ao táxi ocioso. Website. Disponível em: http://www.em.com.br/app/noticia/gerais/2013/02/15/interna_gerais,350571/bhtrans-fecha-o-cerco-ao-taxi-ocioso.shtml.

Reis, F. A. L., de Arruda Pereira, M., and Almeida, P. E. M. (2011). Location-based dispatch to reduce the waiting time for taxi services.

Santos, D. and Xavier, E. (2015). Taxi and ride sharing: A dynamic dial-a-ride problem with money as an incentive. *Expert Systems with Applications*, 42(19):6728–6737.

Steenbruggen, J., Tranos, E., and Nijkamp, P. (2015). Data from mobile phone operators: A tool for smarter cities? *Telecommunications Policy*, 39(3-4):335–346. New empirical approaches to telecommunications economics: Opportunities and challenges Mobile phone data and geographic modelling.

Taha, H. A. (2008). *Pesquisa Operacional: uma visão geral*. Person Prentice Hall, São Paulo, SP, Brasil.

Xu, Z., Yuan, Y., Jin, H., and Ling, H. (2005). Investigating the value of location information in taxi dispatching services: A case study of dazhong taxi. *Pacific Asia Conference on Information Systems*.

Yang, H. and Wong, S. (1998). A network model of urban taxi services. *Transportation Research Part B: Methodological*, 32(4):235–246.