CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 95 (2016) 40 – 47
Procedia Computer Science

Complex Adaptive Systems, Publication 6
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2016 - Los Angeles, CA

Applied Graph Theory to Real Smart City Logistic Problems

Jose M. Gutierrez[a,b]*, Michael Jensen[a,c], Tahir Riaz[b]

[a]*Aalborg University, Fredrik Bajers vej 7, 9220 Aalborg Ø, Denmark*
[b]*RadioAnalyzer ApS, Nytorv 11 3, 9000 Aalborg, Denmark*
[c]*Netplan ApS, Hedeager 3, 8200 Aarhus N, Denmark*

Abstract

In a recent project in the Region of North Jutland in Denmark, an empirical study of the mobile broadband conditions was carried out by installing measurement equipment on two cars driving around everywhere in the Region. This paper provides an overview of the system designed for carrying out the project, focusing on the solution for the logistics problem presenting well-defined challenges. We present a feasible solution for the non-trivial problem of planning routes, guaranteeing 100% coverage of the 19.000 km of roads in the Region and having hard computational time constrains. The solution is a combination of applied graph theory and Geographic Information Systems (GIS), framed into a realistic context to be able to carry out the project. One of the future foreseen application fields of cellular networks is to provide efficient network connectivity to cyber physical systems for the required data transactions between the devices and data processing units. Hence, the results derived from similar measurement projects, among many others, is the evaluation of whether the broadband infrastructure is currently ready for the deployment of Smart City outdoor cyber physical systems and implicitly, which steps should be taken towards the deployment of these systems.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of scientific committee of Missouri University of Science and Technology

Keywords: Smart Logistics, Route Planning, Smart City, Graph Optimization

* Corresponding author. Tel.: +45 9940 8616
E-mail address: jgl@es.aau.dk

1877-0509 © 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of scientific committee of Missouri University of Science and Technology
doi:10.1016/j.procs.2016.09.291

1. Introduction

Sustainable cities, machine to machine communications, broadband connectivity anywhere any time, or big data analytics are terms commonly associated with the digital development towards Smart Societies. As part of this trend, mobile broadband access is an important piece of the puzzle, not only for the connectivity of people but also of *things* [1]. Therefore, the mapping of this type of technologies is not only convenient for understanding how the traditional mobile communications (voice or data from/to mobile devices) perform by measuring its real-life performance but also to evaluate the potential use of such technologies for cyber physical systems connectivity in concrete geographic scenarios.

This paper describes part of the work in a project carried out by the region of North Jutland in Denmark, with the objective of mapping the coverage and quality of the mobile broadband networks of four operators. To gather the measurements, two cars carrying specialized equipment drove around the region's area for six months. The main theme of this paper is the description of how the driving campaigns were designed. In addition, a data analytics system was implemented and the resulting data was visualized in a publicly accessible GIS based website [2].

One of the biggest challenges was how to design the driving routes in order to guarantee that the mobile coverage was measured on 100% of the roads in the studied area (one of the requirements in the project) due to its combinatorial nature. Moreover, the time and computational resources available added an extra challenge when solving the problem, not being able to apply the theoretical complex methodologies commonly studied in academic combinatorial optimization research [3].

Thus, there was significant mismatch between the hard constrains in the project definition and time/computational resources available, providing a good illustrative case where theoretical formulations and solving methods cannot be applied to the practical realization of a real project. Several well-known problems in graph theory were faced, and having the theoretical solving methods behind these as a reference, we were forced to reformulate our solving approach to an *ad-hoc* practical methodology to comply with the demands, requirements, and objectives.

In a nutshell, the idea behind this work and the framework of this paper can be summarized in one simple question: "*How do we move from our theoretical formulations to a feasible applicable solution when ideal solving methods cannot be applied to real-life problems and having limited resources?*".

2. Background

One of the most convenient ways of working with road paths and routing is to treat the road network as a graph. Usually, roads and streets are presented in Geographic Information Systems (GIS) as polylines (sets of straight segments defined each by two points) [4]. These segment points can be considered as the vertices of a graph while the segments would be the edges. By using this approach, it is fairly simple to solve very well-known problems in logistics optimization such as calculating the minimum length route connecting several waypoints, also known as "*The travelling salesman problem*" [5].

The nature of this type of logistics problems is of combinatorial optimization, where the objective is to optimize routes, fulfilling a set of demands and/or constraints. Very well-known methods have been successfully used when solving this type of problems. Examples of these methods are evolutionary/meta-heuristics algorithms used in [6] where Genetic Algorithm was used for designing the network flow for businesses relaying on logistics, in [7] where Ant Colony algorithm was applied for logistics optimization for delivering goods and help to areas critically affected by disasters, or in [8] where Simulated Annealing Algorithm was applied for scheduling optimization of container loading tasks in harbours. Additionally, in other some cases, more advanced methods were applied, such as Integer Linear Programing in [9].

Normally, most of these route optimization problems are based on location points, where the design of routes relays on covering these concrete locations. However, in this project the problem is different; the objective is to create minimum length routes traversing all roads in a given area, and implicitly minimizing the overlapping factor (final route length/sum of the roads' length traversed).

The size of the area covered and length of the roads to be traversed (19.000 km) provides an additional challenge when solving the problem. And combined with the previously mentioned practical constraints of the project, the solving methodology applied is based on a "*divide and conquer*" approach which has also been used in similar cases, such as in [10], where a single global solving method became unfeasible due to the problem size. More specifically, our method consisted of two steps, presented more in depth in Section 4:

A. Divide the roads into subsets to be covered by car and day, solved by clustering.
B. Solve each of the subsets individually as isolated problems which when combined, fulfil the objective of driving 100% of the roads in the region. This was solved by the proposed applied graph theory method in this paper.

3. The system architecture

This Section presents an overview of the whole system designed for carrying out the project, without describing in depth the processes and tasks involved in each of the parts. Although this work is focused on the graph theory algorithms used for driving route planning, it is important to present the real context in which they were applied. Fig. 1a presents the diagram of the system and its interfaces. There are four main parts forming the system:

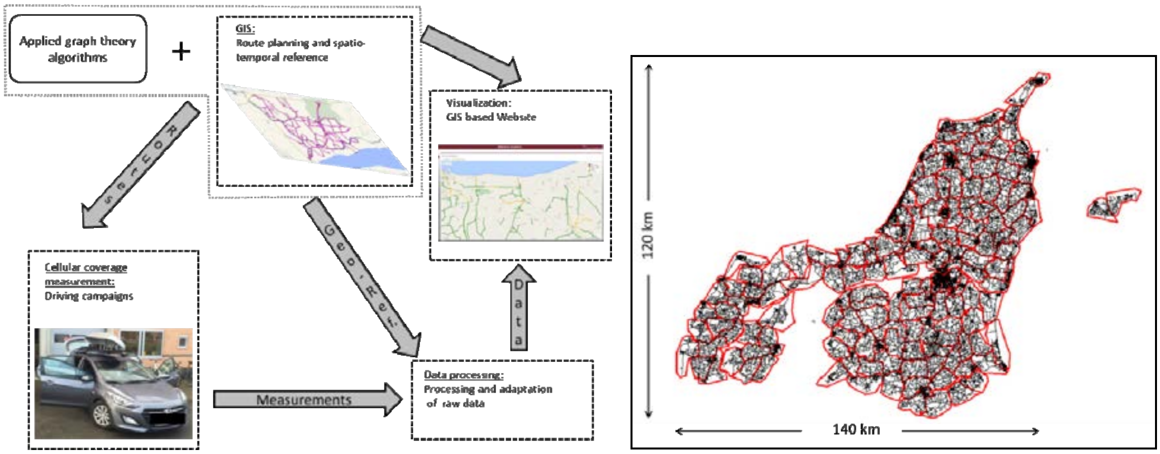

Fig. 1 a) Overview of the Project’s System. b) Region Under Study and Clustering Representation.

1. Geographic Information Systems (GIS): The whole project (i.e. Route planning, data processing and visualization) is supported by the use of GIS in order to link the mobile signal measurements into a spatio-temporal reference. The complete process of route planning, described in depth in Section 4, relies on the processing of GIS road data combined with the proposed applied graph theory algorithms.
2. Mobile coverage/quality measurement: The accurate measurement of mobile signals required rather special equipment to be mounted on top of a car (two in this project). The equipment consisted of a scanner collecting the relevant data and in addition, several mobile phones were connected to the equipment to gather additional measurements for each of the four mobile operators investigated. Every morning, the routes for the day were loaded into custom made navigation software running in tablets for the drivers to be able to follow the routes precisely. At the end of each driving day, the data collected was uploaded to a FTP server.
3. Data Processing: Three types of measurements were collected, mobile signal strength, mobile signal quality and down/upstream data throughput. Regardless of the type of measurement, each data point was associated to GPS coordinates and a timestamp. However, the required granularity of the visualization was not points but road segments. Hence, each data point was linked to the closest road segment and the proper calculations were performed, in order to aggregate its value to the rest of the measurement points belonging to the same segment. This approach implies that the data visualized is relative to segments and not to the initial measurement points. In this way, it ensures a smoother visualization of the measured parameters as each presented value is the result of the average of many samples over the same concrete road segment rather than scattered points in a map.

4. Visualization: The results obtained after processing the data was presented in a GIS based webpage [2]. The concrete numerical values of the measurements were disguised into rather broad intervals represented by a palette of colors, in order to preserve the market competition conditions among mobile operators.

4. The Route Planning

The goal of the driving campaigns was to measure the mobile broadband coverage in every single road in the region of North Jutland, Denmark. This accounts for around 19.000 km of roads (12.000 miles) to be covered by two cars. The most restrictive constrain/objective was to guarantee 100 % road coverage by the planned routes. In addition, the problem presented the following practical constraints given by the framework of the project:

- Two driving routes to be delivered per working day, giving a maximum search time constraint of 12 hours/route assuming full resource dedication to the calculation of each route.
- Each route must be *humanly* drivable, no more than 8 hours of driving per car and day.
- No time to pre-calculate routes before driving campaigns started.
- No special computational equipment available for route planning, just standard PC (Intel i7 @ 3.4 GHz and 4 GB Ram)

4.1 Preparation

As introduced in Section 2, the "*divide and conquer"* approach was used to solve the route planning problem. The 19.000 km of roads were divided into 160 clusters with the following properties:

- Each cluster must form a connected graph on its own. In this way, a continuous route can be planned using only the roads belonging to one cluster (isolating each cluster to be an independent problem to be solved).
- The sum of the roads within each cluster should not exceed 150 km. This value is given in relation to the maximum time it would take to complete a driving route within a working day.

Essentially, clustering in this context implies the grouping of the road segment points. The formation of the clusters was performed by distributing a number of centroids placed strategically across the target area. The density of centroids varied in relation to the density of the roads (higher density of roads implies higher concentration of centroids). Using these centroids as the roots, shortest path spanning trees (*SPST*) were calculated to form the clusters. Each cluster was defined as the set of road segment points (vertexes in the graph) sharing the same closest centroid, in other words, they share the same shortest path root. In this way, the division of the road network in connected sub-graphs is guaranteed (one connected graph per cluster). The cluster formation procedure required a refinement phase by manually reallocating some of the centroids in order to further balance the length of the roads covered by some of the clusters.

This problem of distributing centroids in order to achieve certain pre-defined objectives and fulfilling concrete constraints is well-known as "*The Facility Location Problem*" [11]. Once again, this is another good example of how classic methods for solving this type of problems were not suited due to the mismatch between their complexity and the available resources, having to find an alternative feasible method. However, the objective of dividing the problem into smaller solvable problems and maintaining the initial requirements was achieved by applying this alternative clustering approach. Fig. 1b presents the area covered, the roads involved and the resulting clusters.

4.2 The Route Planning

<u>The problem:</u>

Regardless of the size of the area covered or the length of the road network considered, creating a route is essentially equivalent to sorting a number of vertexes or edges to be visited or traversed. Moreover, when an objective function and constrains are added to the sorting, it becomes a combinatorial optimization problem. A good example of this type of problems is "*The Minimum Linear Arrangement Problem*" which is categorized as NP-hard [12]. The route optimization problem faced in this project can be formally defined as follows:

Let *G (V, E)* be a connected road graph in a cluster *C* where a route $R_C = \{r_0, r_1, r_2 \ldots r_{n+m}\}$ traversing all edges in $E_C = \{e_0, e_1, e_2 \ldots e_n\}$ has to be calculated minimizing its total length, $L(R_C) = min(\sum l(r_i)) \ \forall r_i \in R_C)$, being $l(r_i)$ the length of a segment r_i. R_C must fulfil two conditions to be considered a valid solution:

- $R_C \supseteq E_C$ → All edges in the road graph must be part of the resulting route.
- r_i *and* r_{i+1} *are adjacent* $\forall r_i \in R_C$ → The resulting route is a connected.
- r_0 *and* r_{n+m} *are adjacent* → The starting and ending point of the route is the same vertex.

<u>The methods:</u>

Among all the possibilities for solving the route problem, two sets of methods were considered from a practical perspective: A) Spanning tree based greedy algorithm and B) Evolutionary based algorithm.

A) **Spanning tree based greedy algorithms** combined with a graph navigation algorithm provided a robust method characterized by the following three steps for each cluster:

1. Calculation of traditional *SPST* having as a root the closest vertex to a garage where the cars were parked.
2. *SPST* modification to also include the edges not belonging to the initial *SPST*, referred to as of ghost edges and vertexes for the rest of the paper.
3. Graph Navigation algorithm applied over the modified *SPST* for sorting the vertexes in visiting order to create the route (as a sequence of waypoints).

This method provides the following features:

- The algorithm can only be applied on two way streets.
- Guaranteed 100 % road coverage.
- Runtime at the order of seconds.
- Guaranteed to traverse each road segment twice and only twice. This feature is convenient for two reasons, 1) route verification, if a road segment is not traversed exactly twice some sort of error/inconsistency happened during the calculations and 2) measurement consolidation, having redundant measurement points at the same locations but at different timestamps (one "*going*" and the other "*coming back*") allow correcting inconsistences in the data due to, for example, equipment instabilities.

Method description, the three steps:

1. *SPST* (*T*) connecting all vertexes in a cluster: This part consists of an implementation based on Dijkstra's algorithm [14] and it must run from a selected root for each cluster. The result when running the algorithm is the division of the edges in two groups. Let E_C be the set of edges in cluster *C*, E_C is then divided into E_T as the set containing all edges belonging to the calculated *T* and $\overline{E_T}$ containing the rest of the edges in a cluster.
2. *SPST* modification (*T'*): In order to traverse all roads, all edges in *C* should be part of the *T'*. For this purpose, all edges in $\overline{E_T}$ need to be modified and added to *T* to form *T'* by following this procedure:
   Let *e* be an edge in $\overline{E_T}$, characterized by its two vertexes v_1 and v_2,which needs to be modified by associating it to a ghost node of one of its two vertexes. Let v_1 with coordinates x_1 and y_1 be the chosen vertex to be duplicated and modified to v_1'with exactly the same coordinates ($v_1' = (x_1, y_1)$). *e'* is the result of modifying *e* associating it to vertexes v_1' and v_2. Then, *e'* is added to *T'*, *implying* that the degree of v_1'=1.
3. Graph navigation algorithm: It is used to determine the visiting order of the vertexes having as starting and ending points the root vertex for each cluster. The fundament of the algorithm is an iterative process, having the root vertex of *T'* as starting navigation point, where the next vertex to be visited is decided from a current vertex at each iteration. The basic rule followed is to move to a further away neighbor vertex (further away from the

root) as long as it is possible. Otherwise, the only option left is to return to a previous vertex. Each time a vertex is visited, it is inserted into a list characterized by a visiting order field (auto-increment). For implementing this algorithm, each vertex in *T'* has two attributes: Degree and distance to the root. It is worth highlighting that the degree of each node is calculated based on *T'* and not on the original road graph. Let v_i be a generic vertex of *T'*, characterized by its degree and distance to the root, $deg(v_i)$ and $dist(v_i)$ respectively. Fig.2a presents the pseudo-code of the algorithm applied.

B) **Evolutionary algorithms** such a Genetic Algorithms or Simulated Annealing [13] are good options for solving combinatorial optimization problems in logistics, obtaining near optimal solutions in a relatively short time. The proposed method is a modification and adaptation of a Genetic Algorithm used to solve "*The traveling salesman problem*" in [15]. The proposed procedure is as follows:

1. Calculation of shortest route traversing all vertices within a cluster (and returning to the source vertex) using Genetic Algorithms. The solution results in a cycle graph.
2. Modification of the cycle graph to include the edges not traversed by the original route as branches of the cycle. The exact same procedure as for the modified *SPST* method was used by adding ghost vertices and edges as the branches of the cycle graph.
3. Graph Navigation algorithm to create the route as a sequence of vertices to be visited in order.

This method provides the following features:

- Possibility of improving the solution provided by the SPST method by obtaining an overlapping factor lower than 2. However, it is not guaranteed to find such a solution within 12 hours (limitation given the requirement of delivering two routes per day).
- Guaranteed 100 % road coverage.
- Runtime at the order of hours.

Method description, the three steps:

1. Shortest path traversing all vertices in *C* (*P*): Route *P* is found when applying the Genetic Algorithms for solving the visiting order for the travelling salesman problem for all vertices in *C*, done exactly as in [15]. Similarly to the *SPST* method, the set of edges in cluster *C* (E_C) is divided into E_P as the set containing all edges belonging to the calculated *P* and $\overline{E_P}$ containing the rest of the edges in a *C*.
2. Modified route (*P'*): *P* is then modified to include all edges not traversed by *P* into *P'* by following the exact same procedure as Step 2 for the *SPTS* (omitted to avoid redundancy). *P'* results in the cycle graph (*P*) with the addition of branches, being these branches edges with one vertex having degree>2 (vertex connecting the branch to the cycle) and the other vertex (ghost) having degree=1. The rest of the vertices in *P'* have degree 2.
3. Graph navigation algorithm: The fundament of the algorithm is the opposite as for the *SPST* method. For each vertex v_i in *P* visited in the navigation process, all branches connected to v_i must be visited before the next vertex in the *P* can be visited. This algorithm can be easily implemented using the degree of the vertices in *P'* and the visiting order in the cycle graph *P*. The pseudo-code for this algorithm is presented in Fig 2b.

```
//Initialization
vc → Root
ind →0
 While ∑∀ vi in T'  deg (di) > 0 do
   Insert vc in order list as
         Order → ind
         Label → vc
         ind++
//Find next vertex
   For each vj adjacent to vc do
        if deg(vj)>0 and dist(vj)> dist(vc)
            Next vertex → vc= vj
            Exit
    if Next vertex is null // Only one option left, to move backwards
            Next vertex → vc= vj where deg(vj)>0
     deg(vj) - -
Repeat
```

```
//Initialization
ind →0
i→ 0   //visiting order in the cycle graph solution (P)
 While i< size (P) do
   vi →vc
   Insert vi in order list as
         Order → ind
         Label → vc
         ind++
 //check for branches
   if deg(vc)>2 do
      For each vj adjacent  to vc  and deg(vj)=1 do
         Insert vj and vc in order list as
         Order → ind
         Label → vj
         ind++

         Order → ind
         Label → vc
         ind++
i++
Repeat
```

Fig. 2 a) Pseudo-code for method A. b) Pseudo-code for method B.

Algorithms' performance:

The two proposed algorithms were applied to three clusters with different edge density (edges/km), quantifying their performance in terms of run time and overlapping factor of the resulting route. The density is used as a normalized reference of the size of the problem to be solved for each of the clusters (which have similar road length but very different number of edges to be sorted).

As previously mentioned, the *SPST* based solution is deterministic in the sense that a solution can be found within seconds with a guaranteed overlapping factor of 2. On the other hand, the Genetic Algorithm based method is not deterministic and for each cluster, it was executed for 12 hours in order to obtain the best solution within the run time limitations. Table 1 presents the comparison results.

Table 1. Performance Results.

Density (edges/km)	Run time (SPST)	Run time (GA)	Over. Factor (SPST)	Over. Factor (GA)
24.3	32 (s)	12 (h)	2	1.62
39.3	37 (s)	12 (h)	2	2.57
55.5	45 (s)	12 (h)	2	2.45

After 12 hours of run time, it was observed that the Genetic Algorithm was still converging towards better solutions for all the clusters evaluated. A solution with a lower overlapping factor than 2 was found only for the lowest density cluster. The conclusion after this evaluation is that the *SPST* solution can be used for establishing an upper bound solution within seconds. As an extension, the Genetic Algorithm method may improve this upper bound in some cases with the run time limitations.

5. Conclusion

Logistics optimization problems have always been in the scope of graph theory and combinatorial optimization research. In addition, due to the explosion in the use of digital data and data communication networks, a whole new set of logistics problems to be solved are arising, like the one presented in this work.

Basically, the presented problem was to calculate driving routes in North Jutland, Denmark covering 100% of the roads in the region. This problem was part of a project focused on measuring mobile networks' coverage and quality, by driving around with special equipment mounted on the roof of two cars. Moreover, the limited time and computational resources available were not suited for taking advantage of complex solving techniques commonly used in academic environments. Instead, having applied graph theory as a foundation, a set of practical approaches allowed us to succeed in the project, by implementing algorithms which could fulfil the requirements and could be executed within the given time and computational constraints.

This paper presents the methodology used to solve the route planning problem but more importantly, it illustrates an example of how to move from theory to a real-world practical application of graph theory and combinatorial optimization. There is no doubt that graph theory is one of the pillars for innovative solutions in Smart Logistics, however other elements must be taken into consideration such as the practical framework, including the available resources under which the solutions must be obtained.

References

1. Y. Liu and G. Zhou, Key Technologies and Applications of Internet of Things, Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference on, Zhangjiajie, Hunan, 2012, pp. 197-200.
2. Region Nord Mobile Broadband coverage website: http://bredbaand.rn.dk/pages/default.aspx
3. William J. Cook, László Lovász, Jens Vygen. Research Trends in Combinatorial Optimization, 2009, Springer-Verlag Berlin Heidelberg
4. D. Wilkie, J. Sewall and M. C. Lin, Transforming GIS Data into Functional Road Models for Large-Scale Traffic Simulation, in IEEE Transactions on Visualization and Computer Graphics, vol. 18, no. 6, pp. 890-901, June 2012.
5. Gilbert Laporte, The traveling salesman problem: An overview of exact and approximate algorithms, European Journal of Operational Research, Volume 59, Issue 2, 1992, Pages 231-247.
6. Hyun Jeung Ko, Gerald W. Evans, A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs, Computers & Operations Research, Volume 34, Issue 2, February 2007, Pages 346-366, ISSN 0305-0548.
7. Wei Yi, Arun Kumar, Ant colony optimization for disaster relief operations, Transportation Research Part E: Logistics and Transportation Review, Volume 43, Issue 6, November 2007, Pages 660-672.
8. Der-Horng Lee, Zhi Cao, Qiang Meng, Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm, International Journal of Production Economics, Volume 107, Issue 1, May 2007, Pages 115-124,
9. Laporte, G., Marín, A., Mesa, J. A. and Perea, F. (2011), Designing robust rapid transit networks with alternative routes. J. Adv. Transp., 45: 54–65. doi: 10.1002/atr.132.
10. Marc Reimann, Karl Doerner, Richard F Hartl, D-Ants: Savings Based Ants divide and conquer the vehicle routing problem, Computers & Operations Research, Volume 31, Issue 4, April 2004, Pages 563-591, ISSN 0305-0548.
11. Gilbert Laporte, The traveling salesman problem: An overview of exact and approximate algorithms, European Journal of Operational Research, Volume 59, Issue 2, 1992, Pages 231-247.
12. J. Petit, Experiments on the minimum linear arrangement problem,. Sistemes Informatics, 2001., vol. 8, pp. 112–128, 2001.
13. D. T. Pham and D. Karaboga. 1998. Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks (1st ed.). Springer-Verlag New York, Inc., Secaucus, NJ, USA.
14. Yue Yang, Jian-ya. Gong, An efficient implementation of shortest path algorithm based on Dijkstra algorithm, Journal of Wuhan Technical University of Surveying and Mapping, 24 (3) (1999), pp. 209–212.
15. Jose M. Gutierrez, Michael Jensen, Morten Henius, Tahir Riaz, Smart Waste Collection System Based on Location Intelligence, Complex Adaptive Systems 2015. Procedia Computer Science, Volume 61, 2015, Pages 120-127, ISSN 1877-0509, http://dx.doi.org/10.1016/j.procs.2015.09.170.