Location Selection for Ambulance Stations: A Data-Driven Approach

Yuhong Li[†#1], Yu Zheng[#2], Shenggong Ji[#3], Wenjun Wang[*4], Leong Hou U[†5], Zhiguo Gong[†6]

[#]Microsoft Research, Beijing, China

[†]Department of CIS, University of Macau, Macau

[*]Tianjin University, Tianjin, China

{[1]yb27407,[5]ryanlhu,[6]fstzgg}@umac.mo, {[2]yuzheng,[3]v-sheji}@microsoft.com, [4]wenjunwang@263.net

ABSTRACT

Emergency medical service provides a variety of services for those in need of emergency care. One of the major challenges encountered by emergency service providers is selecting the appropriate locations for ambulance stations. Prior works measure spatial proximity under Euclidean space or static road network. In this paper, we focus on locating the ambulance stations by using the real traffic information so as to minimize the average travel-time to reach the emergency requests. To this end, we estimate the travel-time of road segments using real GPS trajectories and propose an efficient PAM-based refinement for the location problem. We conduct extensive experimental evaluations using real emergency requests collected from Tianjin, and the result shows that the proposed solution can reduce the travel-time to reach the emergency requests by 29.9% when compared to the original locations of ambulance stations.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining, Spatial database and GIS*

General Terms

Algorithm

Keywords

City Planning, Emergency Medical Service

1. INTRODUCTION

Emergency medical service, a.k.a., EMS, is a system that provides not only a variety of medical services, e.g., prehospital medical and trauma care, but also the transportation for those in need of emergency care. One of the major challenges encountered by the emergency service providers is selecting the appropriate locations for ambulance stations. Regarding to the construction of an urban EMS system, it is crucial that the emergency requests can be reached in a time-efficient manner as saving time is saving lives. However, most of ambulance stations are not well located, and currently they are located in or near the hospitals which do not consider the distribution of real emergency requests.

A good criteria to measure the quality of the selected locations is computing the average response time for real emergency requests. Through this data-driven approach, we aim at optimizing the average travel-time to reach the locations of real emergency requests. Thus the objective function is the MinSum model and it is a NP-hard problem [2]. Previous works tackle this problem by using some approximate solutions while they measure the spatial proximity, e.g., travel-time between the ambulance station and real emergency request, under either Euclidean space or static road network. In this work, we propose to select the locations under more realistic spatial setting by taking the real traffic information into consideration.

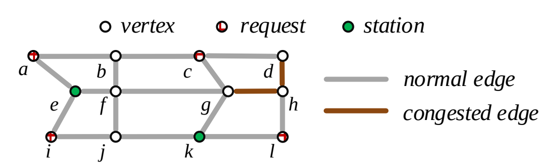


Figure 1: Problem demonstration

Effect of Real Traffic Information. Figure 1 shows the topology of a road network and four emergency requests which are marked by red grid, i.e., $R = \{a, c, i, l\}$. We are allowed to set up two ambulance stations. By using the distance of road segments, the optimal locations are $F = \{e, h\}$ [1]. However, the distance of a road segment can not well capture its travel-time. For example, it costs a lot of time to traverse road segments $e(d, h)$ and $e(g, h)$ even their distance is short due to the traffic congestion. By considering the real traffic information, the optimal locations should be changed to $F = \{e, k\}$.

Obviously, the real-time traffic condition will affect the selection of locations. In this work, we estimate the travel-time of each road segment by using the real trajectories and propose an effect solution based on fast PAM-based refinement

[1]It is sufficient to locate the stations only on the vertices.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SIGSPATIAL'15, November 03-06, 2015, Bellevue, WA, USA.

© 2015 ACM ISBN 978-1-4503-3967-4/15/11 ... $15.00.

DOI: http://dx.doi.org/10.1145/2820783.2820876 .

[3] to solve the location problem. The proposed solution is not only applicable in the construction of urban EMS, but can also be adopted to other urban planning tasks, e.g., fire stations, transit stations for city express and polling stations for public election, etc.

The contribution of this paper can be summarized as:

I We propose to locate the ambulance stations based on the MinSum model by well measuring the travel-time of road segments.

II We propose an efficient PAM-based refinement method to solve the location problem effectively.

III Extensive experimental results on real emergency requests collected from Tianjin demonstrate the effectiveness and efficiency of our proposed solution.

The rest of this paper is organized as follows. In Section 2, we formally define the problem and give an overview of our proposed framework. We present our solutions for travel-time estimation and location selection in Section 3 and Section 4 respectively. In Section 5, we experimentally evaluates the efficiency and effectiveness of the proposed solutions. Finally, we conclude our work with a discussion about future work in Section 6.

2. PRELIMINARY

Location Problem. In this work, the emergency requests are mapped to the vertices and we only consider to locate the ambulance stations on the vertices.

DEFINITION 1 (LOCATION PROBLEM). *Given a road network $G(V, E)$, a set of emergency requests R and the number of ambulance stations to be established k, the location problem aims at finding a set of locations F such that $F \subseteq V$, $|F| = k$ and $\sum_{r \in R} t_F(r) \leq \sum_{r \in R} t_{F'}(r)$ for all $F' \subseteq V$, $|F'| = k$.*

where $t_F(r)$ is the minimum time to reach the emergency request r by dispatching the ambulance from stations in F at the time when r issued. As a remark, two requests located at the same location but in different time slots will require different travel-time to be reached from the same station.

Figure 2 presents the framework of our proposed solution. We first estimate the travel-time of each road segment using real trajectories, then pre-compute the travel-time to reach requests by all the vertices. Finally we propose an efficient PAM-based refinement to find the appropriate k locations.

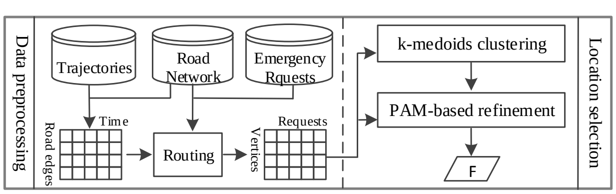


Figure 2: Framework of the proposed solution

3. TRAVEL-TIME ESTIMATION

To estimate the travel-time of road segments, we first map the trajectories of taxicab onto the road network using a map-matching algorithm [4], then we estimate the travel-time of road segments covered by trajectories. We divide a day into 24 hours and the average travel-time of road segments in each hour are calculated. Due to the data sparsity, i.e., the trajectories of taxicab cannot cover all roads at all time slots, we follow a matrix-factorization approach in [6, 8] to compute the travel-time of road segments that are absent of trajectory data.

During the selection of locations, the travel-time of each vertex in V to all requests R need to be used several times. Computing these travel-time on demand will introduce a lot of shortest path computations. We are aware of the methods for efficient shortest path computation in the road network [7]. Nervelessness, we pre-compute the travel-time matrix M, e.g., vertices to requests, and store it in memory.

4. PAM-BASED REFINEMENT

Among many algorithms for the location problem, partitioning around medoids (PAM) is known to be most powerful [5]. Even though, PAM is a local search heuristic, it can provide the approximation guarantee with a factor of at most 5 from the global optimum [1, 2]. In practice, the gap is usually much smaller [2]. Alg. 1 shows its pseudo codes. Given the initial stations F_{ini}, the PAM-based refinement tentatively replaces each current medoids $p \in F_{ini}$ by every vertex $p' \in \{V - F_{ini}\}$, and the overall travel-time of this replacement is estimated, cf. Line 3 - 7. Among all these tentative replacements, the one with the maximum travel-time reduction is carried out, cf. Line 9. The PAM-based refinement repeats this kind of replacements until the overall travel-time cannot be further reduced, cf. Line 11.

Algorithm 1 PAM-based refinement algorithm

Input Road network G, Emergency requests R, Travel-time matrix M, Initial k ambulance stations F_{ini}
Output k ambulance stations F
1: $bsf := \sum_{r \in R} t_{F_{ini}}(r), F_{bsf} := F_{ini}$
2: **while** *true* **do**
3:   **for** each $p \in F_{ini}$ **do** $\triangleright\ O(k^2|R||V|)$
4:     **for** each $p' \in V$ **do** $\triangleright\ O(k|R||V|)$
5:       $F := F_{ini} - p + p'$
6:       **if** $\sum_{r \in R} t_F(r) < bsf$ **then** $\triangleright\ O(k|R|)$
7:         $bsf := \sum_{r \in R} t_F(r); F_{bsf} := F$
8:   **if** $F_{ini}! = F_{bsf}$ **then**
9:     $F_{ini} := F_{bsf}$
10:   **else**
11:     return F_{ini}

The PAM-based refinement is effective for the location problem, but it works inefficiently for large datasets due to its time complexity. The time complexity of PAM-based refinement in one iteration is $O(k^2|R||V|)$. More specifically, it need to carry out $k \cdot |V|$ replacements in each iteration. For each replacement, it takes $k \cdot |R|$ to estimate the overall travel-time. To reduce the number of iterations, we apply a fast heuristic method in selecting the k initial stations, then we propose two optimizations to boost up the PAM-based refinement in each iteration.

4.1 Selecting k initial stations

One can arbitrary select k vertices as F_{ini}. However, if the selected F_{ini} can produce a smaller overall travel-time, fewer swaps are required to reach the final solution in PAM. [2] suggests to employ the k-means clustering algorithm on R in selecting F_{ini}. In this work, we suggest to apply k-medoids clustering [5] in selecting F_{ini} as it can provide more better initial stations. The k-medoids clustering algorithm performs similar to k-means clustering algorithm. It refines F_{ini} by iteratively assigning and updating the medoids for the emergency requests. The major difference between these two clustering algorithms is the medoids (i.e., centers) updating of clusters. Please refer [5] for more detail.

4.2 Fast PAM-based Refinement

Even by properly selecting the initial stations, we can reduce the iteration of replacements in PAM-based refinement. However, it still takes quite a few iterations to reach the final solution according to the experimental evaluation. To boost up the PAM-based refinement, we propose two optimizations which aim at pruning unpromising replacements by batch and reduce the estimation cost of each replacement.

Prune unpromising replacements: To prune unpromising replacements by batch, we first group the vertices according to their spatial proximity. Instead of checking the vertex replacement directly, we first estimate the travel-time lower bound for each group replacement. The travel-time lower bound of a group replacement can be easily derived by precomputing minimum travel-time to reach each request by all vertices in a group. If the travel-time lower bound of this group replacement is already larger than bsf, then vertex replacements in this group can be safely pruned.

Efficient estimation of (group) replacement: According to the aforementioned method, unpromising vertex replacements can be pruned by batch. However, it still takes $O(k|R|)$ to estimate the travel-time lower bound for a group replacement. If a group replacement is survived from the pruning phase, it takes the same cost, i.e., $O(k|R|)$, to estimate the overall travel-time of a vertex replacement.

To reduce the estimation cost for group replacement, we propose to divide the requests into m groups. We denote the travel-time lower bound of replacing a station p in F by a vertex group V_i as $LB(F, p, V_i)$, and it can be calculated according to the following equation.

$$LB(F,p,V_i) = \sum_{i=1}^{m} \begin{cases} (|R_i| \cdot LB_{F-p}(R_i) & \text{if } LB_{F-p}(R_i) < LB_{V_i}(R_i) \\ |R_i| \cdot LB_{V_i}(R_i) & \text{if } LB_{F-p}(R_i) \geq LB_{V_i}(R_i) \end{cases}$$

R_i is the i-th request groups. $LB_{F-p}(R_i)$ is the travel-time lower bound for a station in $\{F - p\}$ to reach a request in R_i. It is formally defined as below.

$$LB_{F-p}(R_i) = \min_{p' \in \{F-p\}, r \in R_i} t_{p'}(r)$$

Similarity, the $LB_{V_i}(R_i)$ is the travel-time lower bound of a vertex in V_i to reach a request in R_i. $LB_{V_i}(R_i)$ for all V_i and R_i combinations can be pre-computed with the time complexity of $O(|R||V|)$. In order to estimate the lower bound of group replacement, it still need to compute $LB_{F-p}(R_i)$. In one iteration of PAM-based refinement, F is fixed. The $LB_{F-p}(R_i)$ for all station $p \in F$ and all request partitions can be pre-computed in $O(k^2|R|)$. As a consequence, these precomputations can reduce the estimation cost of a group replacement from $O(k|R|)$ to $O(m)$. As a remark, the cost of these two precomputations are trivial as they can be amortized by $\xi \cdot k \cdot n$ and $k \cdot n$ group replacements respectively, ξ is the number of iterations in PAM-based refinement.

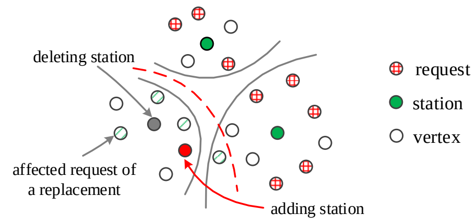


Figure 3: Affected requests

For a group replacement which is survived, each vertex replacement in this group need further refinement. A naive method takes $O(k|R|)$ to estimate each vertex replacement. Actually, the overall travel-time of a vertex replacement can be measured by only checking *the affected requests*. More specifically, the affected requests of a vertex replacement are the requests whose nearest station changed after this replacement. Figure 3 shows the affected requests of a vertex replacement. It is obvious that the affected requests of a vertex replacement is the subset of its corresponding group replacement. For an un-pruned group replacement, we first calculate the affected requests of this group replacement with the cost of $O(k|R|)$, then for each vertex replacement in this group, we only need to estimate the overall travel-time by using these affected requests. As a remark, the early abandon technique can be applied to further improve the performance as we only need to carry out the best vertex replacement in one iteration of PAM-based refinement.

5. EXPERIMENT

In this section, we evaluate the efficiency and effectiveness of our proposed methods. All methods are implemented in C# and performed on a windows server 2012 with 2.4GHz Intel Core E5-2665 CPU and 128GB main memory.

5.1 Dataset

Road networks: We use the road network of Tianjin, which is comprised of 99,007 vertices and 133,726 road segments.

Emergency requests: The real emergency requests were collected from Tianjin during three weeks. This dataset contains 9913 emergency requests.

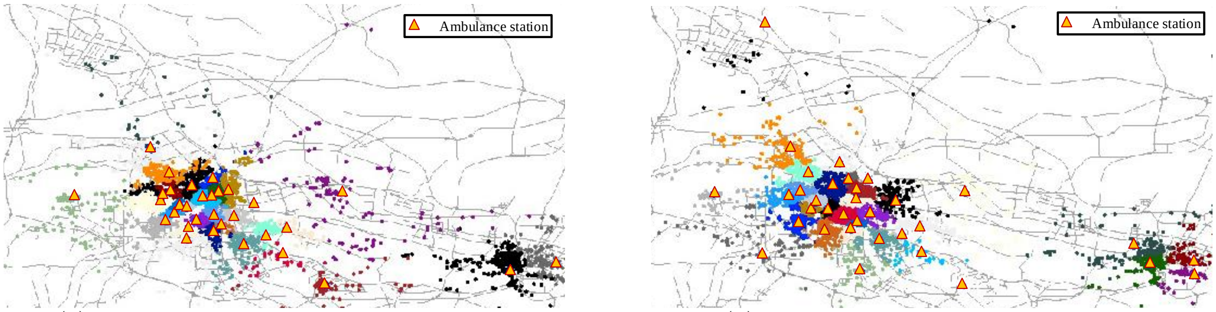
Taxi trajectories: We use a GPS trajectory dataset which is generated by 3,501 taxicabs from Tianjin in 61 days. The number of GPS points reaches 753,059,212, and the total length of the trajectories is over 46,028,698km.

5.2 Experiment Results

5.2.1 Overview

Figure 4(a) shows the original locations of 34 ambulance stations in Tianjin [2]. The stations are marked by yellow tri-

[2]We only show the ambulance stations in the given region.



(a) Original ambulance stations in Tianjin (b) New ambulance stations in Tianjin

Figure 4: Ambulance stations in Tianjin, $k = 34$

angles and the emergency requests which are served by the same station are marked by the same color. It takes 534.5 seconds on average to reach the emergency requests by original stations. However, our solution as shown in Figure 4(b) only takes 374.6 seconds on average to reach the emergency requests and it can saves 29.9% travel-time.

Regarding to the travel-time distribution of requests, only 85.2% emergency requests can be reached in 15 minutes by original locations. However, our solution can reach 90.6% emergency requests in 15 minutes. For the time efficiency of our proposed solution, it only takes 2960 seconds to find these locations by utilizing the proposed optimizations which enables the possibility to further explore real big request data to improve the quality of the selected locations.

5.2.2 *Effectiveness Study*

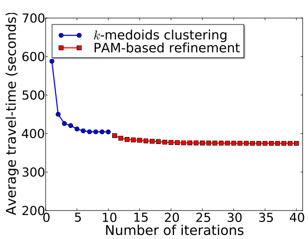


Figure 5: Travel-time by iterations, $k = 34$

Exp-1. Effect of refinement methods. In Figure 5, we study the effect of refinement methods, i.e., k-medoids clustering and PAM-based refinement. According to the result, the k-medoids clustering algorithm can reduce the average travel-time significantly in a few iterations, e.g., it reduces the average travel-time from 587.9 seconds to 404.2 seconds in 10 iterations. The PAM-based refinements further reduce the travel-time by 7.4% in 31 iterations.

Exp-2. Impact of real-time traffic information. To study the impact of real-time traffic information, we compare the locations selected by distance of road segments using the same algorithm. To make a fair comparison, once the locations are selected by distance of road segments, we measure the average travel-time of these locations by using the real travel-time matrix M. According to the experimental result, the locations selected by real travel-time performances better than distance by 8.4%.

5.2.3 *Efficiency Study*

The efficiency of PAM-based refinement comes from two methods, (i) batch pruning, (ii) efficient estimation of replacements. According to our experimental evaluations, our methods can prune 97% unpromising replacements. To estimate a vertex replacement, our methods only need to check 497 (among 9913) emergency requests on average.

6. CONCLUSION

In this paper, we propose to locating the ambulance stations by using the data-driven approach. We take the real emergency requests and traffic information into consideration. The experimental evaluation verified the efficiency and effectiveness of our proposed solution. As a future work, we plan to study the prediction of emergency requests and dynamic ambulances dispatching.

Acknowledgement

This project was supported by grant MYRG105(Y1-L3)-FST13-GZG and MYRG109(Y1-L3)-FST12-ULH from UMAC Research Committee, grant FDCT/106/2012/A3, FDCT/116/2013/A3 from Macau FDCT, the National Social Science Fund (14ZDB153), the National Natural Science Foundation (91224009).

7. REFERENCES

[1] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.

[2] K. Deng, S. W. Sadiq, X. Zhou, H. Xu, G. P. C. Fung, and Y. Lu. On group nearest group query processing. *IEEE Trans. Knowl. Data Eng.*, 24(2):295–308, 2012.

[3] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[4] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang. Map-matching for low-sampling-rate gps trajectories. In *Proc. of GIS'09*, pages 352–361, 2009.

[5] H. Park and C. Jun. A simple and fast algorithm for k-medoids clustering. *Expert Syst. Appl.*, 36(2):3336–3341, 2009.

[6] J. Shang, Y. Zheng, W. Tong, E. Chang, and Y. Yu. Inferring gas consumption and pollution emission of vehicles throughout a city. In *Proc. of KDD'14*, pages 1027–1036, 2014.

[7] L. H. U, H. J. Zhao, M. L. Yiu, Y. Li, and Z. Gong. Towards online shortest path computation. *IEEE Trans. Knowl. Data Eng.*, 26(4):1012–1025, 2014.

[8] Y. Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):29, 2015.