2018 IEEE 34th International Conference on Data Engineering

Node Selection in Large Networks

Alvis Logins
Expected Graduation Date: 15.11.2019
Supervised by Christian S. Jensen and Panagiotis Karras
Computer Science Department, Aalborg University
alvis@cs.aau.dk

***Abstract*—A node selection query returns a set of network nodes that optimize some objective function. The problem of enabling fast and accurate node selection is of high importance in fields such as logistics, service planning, and advertising. Due to the computational complexity, it is often impossible to provide an optimal solution to particular node selection queries. We study new approximation methods for node selection in million-node networks, such as social and road networks. We extend existing models to a more realistic scenarios by introducing time and uncertainty into the problem domain, and we apply the proposed solutions to real-world datasets.**

I. Introduction

Network models are used widely in business analytics to gain insight into geo-social customer interactions and to improve these. Due to the amounts of information and computational restrictions, network models are often simplified by ignoring node and edge labels, by ommiting stochastic and time-dependent components in the models, or by applying graph sampling. We study possible improvements for a particular family of problems, namely *Node Selection* in large-scale networks (NS). Finding points of interest in networks enriched with quantitative geo-social proximities of people and places can significantly improve urban life, can reveal patterns of human behavior, and can provide a foundation for addressing the needs of people.

Important NS functionality falls into the class of *Facility Location* (FL). According to a recent survey [1], the general formulation of FL entails allocation of *facilities* (resourses) in order to minimize the cost of satisfying a set of *customers* (demands) with respect to a set of *constraints*. The simpliest models assume deterministic graphs with customers assigned to nodes. The objective function can be either the sum of distrances to the nearest facility or the maximum distance to the closest facility across all customers. Variations of the problem involve Hubs, Facility Hierarchies, Supply Chains, and Multi-Criteria for user preferences. Most of these problems are known to be NP-hard.

Advanced contemporary models consider the temporal and stochastic nature of customers. Examples of time-dependency include periodic demands [2] and real-time location updates [3]. Historical data about moving customers can be represented either as trajectories [4], i.e., as sequences of timestamped locations, or as source-destination tuples [5]. Facilities can also be represented as static points with temporal availability [2], paths [6], or trajectories. The latter case usually occurs in the scope of routing algorithms.

Stochatic models may embody uncertainty in both node and edge labels. An example of the former occurs when a customer has a probabilitic demand [7]. If a customer is represented as a trajectory, each edge has a probabilistic distribution that shows the *flow* of customers through the edge. The objective function is expressed as an *expected value* of some cumulative cost. For instance, Markovic et al. [2] consider the problem of placing vehicle inspection stations. They analyze the flow of overloaded trucks that produce damage to the environment unless it is regulated. The objective is to minimize the expected value of the damage. The uncertainty is associated with a size of flow and the willingness of drivers to increase travel distance to avoid inspection stations. In this setting, there is no deterministic assignment of which facility serves which customer, as it depends on a particular *scenario*, i.e., the outcome from a probability space. In order to minimize the objective, we need to consider *probabilities* of possible interactions between facilities and customers.

One special case of edge probabilities has the form $\mathbb{P}[w = 1] = p$, $\mathbb{P}[w = \infty] = 1 - p$, where w is the weight of an edge, and $0 < p \leq 1$ is a probability of passing the edge. As some customers might not be connected to any facility, the objective is to maximize the expected number of served customers. This model is used in the *Influence Maximization* problem (IM) that is usually not considered as being within the scope of FL problems due to its higher computational complexity and different application domain. The IM problem aims to identify the most influential people in a social network. It simulates a process of information diffusion, categorizing all network nodes as influenced (active) or not influenced (inactive). Specifically, starting with initial set of active nodes (*seeds*), information iteratively propagates through edges with probability p. A seed is the analogue of a facility, and other nodes (or subset of nodes) are customers with demand 1, that it is not obligatory to satisfy. Two models of node activation are possible. In the Independent Cascade model, a node becomes active if at least one adjacent node was activated in the previous iteration. In the Linear Threshold model, activation happens when there are at least θ active adjacent nodes, where θ is a parameter.

Recently, much attention has been given to two variations of the IM problem, namely Dynamic Seeding [8] and Influence Minimization. Dynamic Seeding represents a way of combining time-dependency and probabilities into one model. Tong et al. [8] propose that new seeds can be added to a

2375-026X/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDE.2018.00216

graph sequentially during the diffusion process. Two patterns of adding seeds are considered: one node per iteration and one node after an ongoing propagation is completed. Influence Minimization is the dual problem of minimizing the number of active nodes given preallocated seeds, by either removing nodes [9] (node immunization), removing edges [10], or by placing seeds of "opposing" influence [11].

The rest of the paper is organized as follows. In Section II, we discuss existing generic and domain-specific approaches that apply to different NS objectives and graph models, including a solution to p-median and IM problems for linear graphs. In Sections III and IV, we consider two examples of the NS problem, namely Multicapacity Facility Selection (MFS) and Influence Minimization by Node Immunization (IMNI). We cover the related literature and describe our proposed solutions to these problems. Finally, Section V concludes.

II. Solution Techniques

A. Common Ground

As NS is an optimization problem, common optimization techniques are applicable. Here, we cover several of them and describe their applicability to FL and IM problems, supported by an example.

The simplest class of optimization techniques consists of greedy heuristics, where top-k nodes are selected according to some ranking function. This kind of approach is often applied in cases of high computational complexity [1], [10]. Submodularity of the objective function allows to derive an approximation guarantee of $(1 - 1/e) \cdot opt$ [1].

Other solutions include Linear Programming (LP), Mixed Integer Programming, and various LP-relaxation based heuristics. They are able to contend with graphs of up to several thousand nodes and provide flexibility in constraint and objective formulation. For instance, consider a linear graph with nodes $a_1, a_2, ..., a_n$. The complexity of an IM problem on a linear graph is smaller than on a graph with cycles, since it is relatively easy to calculate the probability of a path between any two nodes. Pairwise probabilities form a matrix $d_{ij} = \prod_{k=i+1..j} p_{k-1,k}$, where $0 < p_{i,j} \leq 1$ is the probability that edge (a_i, a_j) survives. For FL problems, d_{ij} denotes a matrix of geodesic distances.

Consider a canonical case of FL, namely the k-center problem, where we minimize the distance between nodes and the *closest* facility by selecting k nodes for new facilities. The objective function is shown in Eq. 1. Variable x_i is an indicator that a_i is selected for a new facility. Variable y_{ij} is another Boolean that captures whether node a_i is assigned to facility a_j. A set of constraints in Eq. 2 assures that a_i can be asigned only if a_j has a facility and the total amount of new facilities is k. Due to Eq. 2 and the monotonically increasing $d(i, j)$, the optimal assignment will always correspond to the assignment to the closest facility.

$$\min_{y_{ij}} \sum_i \sum_j d_{ij} y_{ij}, \quad x_j, y_{ij} \in \{0, 1\} \tag{1}$$

$$y_{ij} \leq x_j, \quad \sum_j x_j = k, \quad \sum_j y_{ij} = 1 \tag{2}$$

On the other hand, in the IM model, the probability of a node i being active in a linear graph is equal to

$$S(i) = 1 - (1 - d_{li})(1 - d_{ir}), \tag{3}$$

where l and r are indexes of the left and right closest seeds, respectively. The expected number of active nodes is the sum of $S(i)$ over all nodes. We can express d_{li} as $\sum_{j=1}^{n} d_{ji} \cdot y_{ji}$ with the constraint that $\sum_{j=i+1}^{n} y_{ji} = 0$, meaning that we assign only to seeds with a smaller index. One new variable $z_{ij} \in \{0, 1\}$ is used for assignment to a seed with a larger index in a similar manner. We can linearize the objective by taking into consideration that probabilities of information propagation in real-world graphs are small [12]. The resulting LP objective is presented in Eq. 4 with the corresponding constraints occurring in Eqs. 5,6,7.

$$\max_{y_{ij}, z_{ij}} \sum_i \sum_j d_{ij} y_{ij} + \sum_i \sum_j d_{ij} z_{ij} \tag{4}$$

$$y_{ij} \leq x_j, \quad z_{ij} \leq x_j, \quad \sum_j x_j = k \tag{5}$$

$$\forall k \quad \sum_{i=0}^{k-1} y_{ik} \leq 1, \quad \sum_{i=k+1}^{n} z_{ik} \leq 1 \tag{6}$$

$$\forall k \quad \sum_{i=k+1}^{n} y_{ik} = 0, \quad \sum_{i=0}^{k-1} z_{ik} = 0 \tag{7}$$

We can also find an optimal solution in a linear graph in polynomial time using Dynamic Programming. Eqs. 8 and 9 show the Bellman equations for the FL and IM problems. Here, $f_{\text{FL}}(m, k)$ says that the cost of the assignment of the k-th facility (state variable) to node m (control variable) is equal to the minimum cost of the assignment of the $(k-1)$ facility to node j, plus the cost of adding a new facility to the m-th node (payoff function). The minimization occurs over control variable j. The cost of the new facility is equal to the sum of distances to the closest facility over all nodes in the range $j..m$. Next, $f_{\text{IM}}(m, k)$ has similar meaning, with the payoff function shown in Eq. 10.

$$f_{\text{FL}}(m, k) = \min_{1 \leq j < m} \{f(j, k-1) + \sum_{j < i < m} \min_{x \in \{m, j\}} d_{xi}\} \tag{8}$$

$$f_{\text{IM}}(m, k) = \max_{1 \leq j < m} \{f(j, k-1) + S(j, m)\} \tag{9}$$

$$S(j, m) = \sum_{x=j}^{m} (d_{jx} + d_{xm} - d_{jx} d_{xm}) \tag{10}$$

However, in real-world scenarios, graphs have complex topologies, and LP is a much more powerful tool. LP is used widely for solving FL problems [1] and enables efficient solutions to IM problems for arbitrary graph topologies [13]. Sing and Dinh [13] presents a mixed LP and randomized algorithm. Their solution can handle a network with 1.5K nodes and 2.7K edges within a time limit of 1h.

B. Scalable Solutions

Once the scale of the problem increases, NS solutions exploit more domain-specific and data-driven approaches to achieve efficiency. For example, geometric graphs allow embedding of clustering, spatial indexing, and pruning based on spatial location [4]. Another approach is to derive assumptions about the edge weight distribution in a graph, which allows to simplify the model [12] and to predict the stochastic behaviour of customers [9]. Stochasticity is also handled well by randomized algorithms. Deriving a proper estimator for an objective leads to an accurate result within a reasonable time even for large graphs [14].

Recent scalable solutions in the context of FL were introduced for placing a single facility at graph edges. The problem is called Optimal Location Query (OLQ) [3], [15]. While these works focus on choosing a point on an edge, they also consider the question of selecting an edge, which relates to our work. In their solution, multiple facilities are selected in a greedy fasion, by repeated application of the single facility procedure. This approach may have worse accuracy in comparison with algorithms where the decision about facility location is mutual for all facilities at once, by means of cancellation of previous placement decisions (e.g.. the Ford–Fulkerson maxflow algorithm [16]) or iterative location refinement (e.g., k-means clustering [17]).

Based on existing best practices in FL and IM, we consider a combination of traditional LP, randomized algorithms, efficient shortest path calculation, and data pruning, and we derive new scalable and more accurate methods for probabilistic and time-dependent NS models. In the following sections, we present two real-world NS problems that show the efficiency of our approach.

III. Exploration-based heuristic and pruning on dual variables

A. Related Work

In the setting of large road networks, pruning of the search space can have a dramatic effect on the running time of an NS algorithm. One particular bottleneck for spatialy-related problems is efficient shortest path (SP) computation. While efficient SP solutions exist, e.g., contraction hierarchies [18], a potentially more efficient approach is to prune SP. Pruning bounds can be maintained dynamically based on sorted customer costs [15]. Another option is to introduce a relaxation method on primal or dual variables, which is widely used in flow optimization problems. U et al. [19] show that utilizing a potential function on nodes yields a tight pruning bound and allows to significantly descrease SP computations in the assignment problem. We improve on that result, showing that we can apply similar technique in the FL problem, thus prining the majority of SPs in the underlying spatial network.

B. Multicapacity Facility Selection

In real-world scenarios, the capacities of facilities can be limited due to constraints in time, space, and human resources. For example, restaurants have a limit on available tables and warehouses are limited by the storage volume. In both examples, the contraints are defined by the location. Motivated by that, we introduce the *Multicapacity Facility Selection* problem (MFS), where capacities are assigned to each *potential* facility location. The objective of the problem is to select k nodes in a graph, such that the sum of distances from each customer to the nearest available facility is minimized. A selected node can serve up to c_i customers, where i is the index of the node. MFS is closely related to the Capacitated k-median problem, where a fixed set of capacities can be distributed among the facilities placed in a graph.

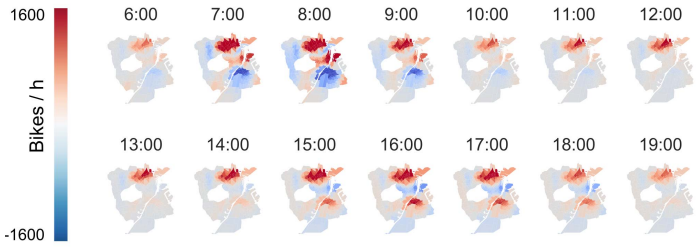
We consider the MFS problem in the context of dockless bike sharing, where a bike can be left at any place after use. The rapid growth of companies such as Mobike[1], oBike[2], and Ofo[3] shows the potential of this new model. However, these companies suggest using "preferable" bike docking stations. Periodically, a service gathers bikes and distributes them to such stations in order to enable more convenient access to the bikes. In case a company wants to place new docking stations, it should consider the available space at each potential new location. We consider the case of building a dockless bike service in Copenhagen Municipality, a city with some of the densest bike traffic in the world. Using data from the Open Data København Portal[4], we determine the locations of docking stations with available bike slots (Figure 1b). We assume that new stations can be placed near existing ones with similar capacity. We distribute bikes around the city according to daily bike traffic data and then find the k best locations for new docking stations according to existing capacities and traffic data.

More specifically, we generate bikes as follows. Bike traffic is calculated by road-side counters. A counter has a known location and accumulates the number of bikes that pass by in each street direction on an hourly basis. Given this information, we can perform interpolation to obtain a vector function $\vec{g}$ of "bike flow" in the city. Magnitude $|\vec{g}|$ is presented in Fig. 1a, where the sign indicates the direction of the flow. Then we calculate the divergence $\nabla\vec{g} = \frac{\partial g_x}{\partial x} + \frac{\partial g_y}{\partial y}$ in each node of a road network, that is proportional to the expected number of bikes parked at that node per hour. Finally, we repeat this for each hour of the day and calculate the variance $\sigma\nabla\vec{g}$. After normalization, we consider the resulting scalar values as an approximation of the probabilistic distribution of parked bikes.

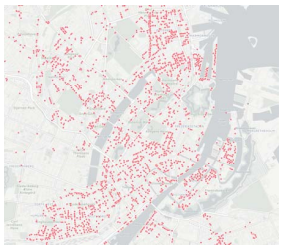
C. Wide Matching Algorithm

To solve the MFS problem, we propose a so-called *Wide Matching Algorithm* (WMA) that iteratively assigns customers to candidate facilities until there is a subset S containing k facilities such that each customer is assigned to at least one facility in S. The novelty of the approach is that each potential facility has a capacity that limits the number of customers that can be assigned to it. As a result, each iteration contains a

[1]https://mobike.com/
[2]https://www.o.bike/
[3]http://www.ofo.so/
[4]http://data.kk.dk/



(a) Interpolated bike traffic



(b) Existing bike docking stations

Fig. 1: Data about bike traffic in Copenhagen Municipality.

reassignment phase, where capacitated facilities may exchange customers in order to improve the objective function. This procedure is known as Bipartite Matching, and WMA exploits an efficient data-driven pruning technique [19], which was not considered previously for matching in spatial networks. Moreover, WMA performs SP calculation and matching as one efficient procedure so that it avoids calculating many unnecessary paths.

We compare WMA to a baseline solution and a commercial optimizer while considering quality and runtime. The *Gurobi Optimizer* [20] is a solver that uses LP techniques and provides an optimal solution to the MFS problem. As the baseline, we implemented an algorithm that divides the input customer set into k buckets based on spatial proximity and then selects the centroid of each bucket as a facility. We create such buckets using a *Hilbert curve* [21].

We study the performance on a synthetic dataset and the bike docking problem with 10^3 bikes. Our results show that WMA significantly outperforms Gurobi in runtime and has close to optimal accuracy (Fig. 2). For the bike docking problem, Gurobi fails to deliver a result within a time limit of 6h with an out-of-the-box solution for SP calculation. Next, we consider the benefit of taking into account the multicapacities. *Uniform WMA* (UF WMA) captures the accuracy of WMA when facility locations are calculated by assigning average capacities to all facilities. We see that the accuracy drops. Also, we see that the inconvenience of customers expressed by the objective drops as the number of facilities increases.

In future work, we aim to extend WMA to being able to accommodate customer trajectories by embedding an efficient Dijkstra-based incremental method to calculate SP between trajectories and potential facilities.

IV. Probabilitic Networks and Influence Minimization

We proceed to discuss the IM and IMNI problems introduced in Section I. We assume a social network $G = (E, V)$ with edge probabilities $p(v_1, v_2), \forall (v_1, v_2) \in E$. The *inverse graph* G^{-1} is the graph G with all edges inversed. The state-of-the-art solutions to the IM problem are based on building *Reverse Reachable Sets* (RR) [22], [14]. Let us consider a random node $n \in V$ and a scenario $g \in G^{-1}$. All nodes reachable from n in g form n's RR set. After sampling n and g, the best seeds are selected as the nodes that belong to the most RR sets. Intuitively, the RR sets induced by n indicate which nodes are likely to influence n if selected.

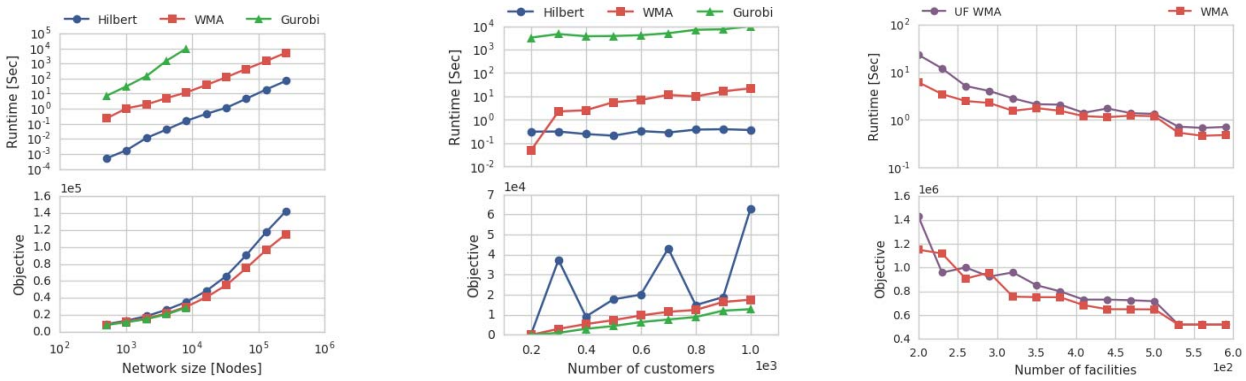
The RR approach is not directly applicable to IMNI, where a seed set S is given and a set of nodes for immunization R is to be defined. Consider a scenario $g \in G^{-1}$ with nodes u,v reachable from S and an existing path from u to v. If v is selected for R and removed, then u may still be reachable from S, unless v is an *articulation node* that disconnects g. Thus, the probability of u being activated by S depends on the number of scenarios where R is a *vertex cut*, i.e., a set of vertices that separate u and S into two connected components if removed.

Previously, the Influence Minimization problem was solved using greedy heuristics. The simpliest approach is the *degree heuristic*, when R contains the nodes with the highest degree [9]. A more efficient greedy solution is presented for the edge blocking version of the problem [10], where the influence of a node is predicted with a help of the Bond Percolation Method. Another greedy solution is presented by Wang et al. [9] for time-dependent IMNI. Their model implies probabilities per each pair of nodes with a particular time-decaying factor. They estimate the global propagation based on rumor general popularity and individual edge probabilities.

Inspired by the ideas of node sampling and RR, we propose a novel randomization algorithm for time-independent IMNI. First, a new virtual node u is added to G with edges $(u, s), \forall s \in S : p_{(u,s)} = 1$. Then, we sample graph scenarios $g \in G$ using Monte-Carlo simulation. For each g, we randomly pick a subset of nodes $V' \in V$ and calculate all minimal vertex separators between $v \in V'$ and u. A *vertex separator* for vertices u and v is a set of vertices Q such that removal of Q from g separates u and v into two disconnected components of g. Set Q is *minimal* if any subset of Q is not a vertex separator. We build sets of minimal separators using an existing algorithm [23]. Finally, R is selected by the maximization of the cumulative frequences of separators that contain nodes from R, under the constraint $|R| = k$.

We test our approach on generated graphs with powerlaw degree distribution, and the VK social network[5]. Preliminary results show that our approach outperforms the naive heuristic (selection of k nodes with the highest degree), and provides significantly higher accuracy than random node selection.

[5]http://vk.com/



(a) Uniform point distribution (b) Clustered point distribution (c) Bike docking problem

Fig. 2: WMA performance on synthetic and real-world datasets.

V. Conclusions

In this work, we discuss the general Node Selection (NS) problem and its applications. We provide a brief overview of the related literature. We discuss Facility Location and Influence Maximization as special cases of the NS problem. More specifically, we consider the problems of Multicapacity Facility Selection (MFS) and Influence Minimization by Node Immunization (IMNI) and provide solutions for them. We report on an experimental evaluation of the MFS solution that offers evidence of scalability and accuracy. The proposed solution to the IMNI problem is based on node sampling and Reverse Reachable Sets. In future work, it is of interest to develop solution for time-dependent FL and IMNI problems, and it is of interest to provide a unified NS computation framework for graph databases.

References

[1] R. Farahani and M. Hekmatfar, "Facility location: Concepts," *Models, Algorithms and Case Studies, Heidelberg: Physica-Verlag Heidelberg*, 2009.

[2] N. Markovi, I. O. Ryzhov, and P. Schonfeld, "Evasive flow capture: A multi-period stochastic facility location problem with independent demand," *European Journal of Operational Research*, vol. 257, no. 2, pp. 687–703, 3/1 2017.

[3] B. Yao, X. Xiao, F. Li, and Y. Wu, "Dynamic monitoring of optimal locations in road network databases," *The VLDB Journal*, vol. 23, no. 5, pp. 697–720, 2014.

[4] S. Mitra, "Identifying top-k optimal locations for placement of large-scale trajectory-aware services," *VLDB 2016 PhD Workshop*, 2016.

[5] A. Zockaie, H. Z. Aashtiani, and M. Ghamami, "Solving detourbased fuel stations location problems," *ComputerAided Civil and Infrastructure Engineering*, vol. 31, no. 2, pp. 132–144, 2016.

[6] M. C. S. Felice, D. P. Williamson, and O. Lee, "A randomized mathrm O(log n)-competitive algorithm for the online connected facility location problem," *Algorithmica*, pp. 1–19, 2014.

[7] L. V. Snyder, M. S. Daskin, and C.-P. Teo, "The stochastic location model with risk pooling," *European Journal of Operational Research*, vol. 179, no. 3, pp. 1221–1238, 2007.

[8] G. Tong, W. Wu, S. Tang, and D.-Z. Du, "Adaptive influence maximization in dynamic social networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 1, pp. 112–125, 2017.

[9] B. Wang, G. Chen, L. Fu, L. Song, and X. Wang, "Drimux: Dynamic rumor influence minimization with user experience in social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2168–2181, 2017.

[10] M. Kimura, K. Saito, and H. Motoda, "Blocking links to minimize contamination spread in a social network," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 2, p. 9, 2009.

[11] D. Yang, X. Liao, H. Shen, X. Cheng, and G. Chen, "Relative influence maximization in competitive social networks," *Science China Information Sciences*, vol. 60, no. 10, p. 108101, 2017.

[12] Y. Yang, E. Chen, Q. Liu, B. Xiang, T. Xu, and S. A. Shad, "On approximation of real-world influence spread," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 548–564.

[13] Y. Song and T. N. Dinh, "Optimal containment of misinformation in social media: A scenario-based approach," in *International Conference on Combinatorial Optimization and Applications*. Springer, 2014, pp. 547–556.

[14] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 1539–1554.

[15] Z. Chen, Y. Liu, R. C.-W. Wong, J. Xiong, G. Mai, and C. Long, "Optimal location queries in road networks," *ACM Trans. Database Syst.*, vol. 40, no. 3, pp. 17:1–17:41, Oct. 2015.

[16] E. L. Johnson, "Networks and basic solutions," *Operations Research*, vol. 14, no. 4, pp. 619–623, 1966.

[17] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[18] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, "Contraction hierarchies: Faster and simpler hierarchical routing in road networks," *Experimental Algorithms*, pp. 319–333, 2008.

[19] L. H. U, K. Mouratidis, M. L. Yiu, and N. Mamoulis, "Optimal matching between spatial datasets under capacity constraints," *ACM Trans. Database Syst.*, vol. 35, no. 2, pp. 9:1–9:44, May 2010.

[20] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2016. [Online]. Available: http://www.gurobi.com

[21] I. Kamel and C. Faloutsos, "Hilbert r-tree: An improved r-tree using fractals," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 500–509.

[22] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2014, pp. 946–957.

[23] T. Kloks and D. Kratsch, "Listing all minimal separators of a graph," *SIAM Journal on Computing*, vol. 27, no. 3, pp. 605–613, 1998.