Traversal Optimizations and Analysis for Large Graph Clustering

Waqas Nawaz*
Supervisor: Young-Koo Lee†, Track: Towards Graduation
Department of Computer Engineering,
Kyung Hee University, Republic of Korea.
Email: {*wicky786, †yklee}@khu.ac.kr

***Abstract*—Graph is an extremely versatile data structure in terms of its expressiveness and flexibility to model a range of real life phenomenon, such as social, biological, sensor, and computer networks. Finding groups of vertices based on their similarity is the fundamental graph mining task to get useful insights. The existing methods suffer from scalability issues due to enormous computations of an exact similarity estimation. Therefore, we introduce Collaborative Similarity Measure (CSM) based on shortest path strategy, instead of all paths, to define structural and semantic relevance among vertices efficiently. We evaluate this measure for personalized email community detection as an application scenario. However, an abundance of structural information has resulted in non-trivial graph traversals. Shortcut construction is among the utilized techniques implemented for efficient shortest path (SP) traversals on graphs. The shortcut construction, being a computationally intensive task, required to be exclusive and offline, often produces unnecessary auxiliary data. To overcome this issue, we present Shortest Path Overlapped Region (SPORE), a performance-based initiative that improves the shortcut construction performance by exploiting SP overlapped regions. Path overlapping with empirical analysis has been overlooked by shortcut construction systems. SPORE avails this opportunity and provides a solution by constructing auxiliary shortcuts incrementally, using SP trees during traversals, instead of an exclusive step. SPORE is exposed to a graph clustering task, which requires extensive graph traversals to group similar vertices together, for realistic implications. We further suggest an optimization strategy to accelerate the performance of the clustering process using confined subgraph traversals. Leveraging the SPORE with multiple SP computations consistently reduces the latency of the entire clustering process. A parameter-free graph clustering with scalable graph traversal strategy for a billion scale graph remain an open issue.**

I. Introduction

Graph clustering is one of the fundamental mining operations for analyzing and identifying strongly related groups of vertices in an entire graph [1][2][3][4][5]. The estimation of the exact relevance among vertices is an expensive operation even with a linear clustering framework, *e.g.,* K-means. Generally, pair-wise vertex relevance measures are categorized into either local or global methods, based on their search space of the entire graph [6]. Local strategies [7][8] use the direct neighborhood information of vertices to define an approximate closeness measure. However, a global measure [9] requires an entire scan of a graph to estimate accurate similarity among vertices. Intuitively, it is a trade-off between the time efficiency and effectiveness of the clustering results using either approach. We are interested in finding the clusters of vertices in multi-attributed weighted graph using the global relevance measure, without compromising the quality of the results.

Many global measures have been proposed in literature to define the degree of closeness between an arbitrary pair of vertices [10][11][12][13][14]. Among them, the shortest path (SP) strategy is inherently simple and robust [6]. Mainstream work is either focused on the topological structure or homogenous characteristics; unfortunately, very few recent approaches lead us towards the conjunction of both aspects at the expense of computation or quality of clustering results [15] [16] [17] [18]. To overcome this problem, *we introduce an efficient global similarity measure based on SP to cluster multi-attributed graph* **(section II)**. Dijkstra's algorithm [19] is a classical solution for computing the SP. It can take several seconds to compute a single pair shortest path (SPSP) in $O(|V|log|V| + |E|)$, where $|V|$ and $|E|$ are the total number of vertices and edges in a graph, respectively. The naïve clustering process, *i.e.,* K-means, requires more than $|V|$ single-source shortest path (SSSP) computations which makes it even more complicated and impractical for large graphs.

A plethora of techniques have been presented to improve the time performance of Dijkstra's algorithm [20][21]. These methods either focus on reducing the total number of expansions or latency cost by introducing the auxiliary edges, *i.e.,* shortcuts, and an efficient indexing structure on a disk, respectively, at the pre-processing stage [22][23][24][25][26]. It has been empirically and theoretically proven that these auxiliary data can enhance the performance of the algorithm by providing sufficient speed-up on the runtime traversal queries [27]. Nevertheless, the pre-processing time and index size are expected to be quadratic and unrealistically large for massive graphs [25]. Therefore, *we empirically analyze SP traversals in real graphs to anticipate effective regions for auxiliary data* **(section III)**. The key concerns of our extended study [28] are as follows. (i) *Avoid the exclusive pre-processing step and still gain the similar speed-up*, **(section IV)**. (ii) *Reduce the computation overhead of repeated SP traversals over the entire graph by a restrictive search space as a subgraph*, **(section V)**. (iv) *parallel approach to compute a set of SPs by expanding multiple vertices at the same time to reduce the CPU latency*, **(section VI)**. However, *few challenging issues*,**(section VII)**, still need to be addressed in near future.

[0]*Corresponding Author: Young-Koo Lee (yklee@khu.ac.kr)*

II. Efficient Global Similarity Measure

We propose an alternate similarity measure, i.e., Collaborative Similarity Measure (CSM), for intra graph clustering problem [29][30]. The similarity measure determines the strength of relationship or connectivity among pair of vertices. CSM leverages the graph clustering solution with structural and contextual, i.e. semantic, traits to achieve comparatively similar quality at less computational cost. It belongs to the category of graph vertex clustering, an un-supervised method. It requires the number of clusters as an input parameter prior to find the clusters. It also provides an opportunity to control the connectivity or similarity strength among the clustered nodes through single parameter. The proposed method deviates from existing state-of-the-art approaches in terms of the following facets: (a) an efficient pair-wise similarity measure based on shortest path strategy rather than all possible paths, (b) simple and an effective strategy by considering both structural and contextual aspects concurrently, (c) a linear iterative partitioning strategy for clustering, (d) captures three basic scenarios for vertex pair connectivity, i.e. connected, indirectly connected, or disconnected.

The collaborative similarity among an arbitrary pair of vertices, source to destination in a graph, is computed through structural and contextual similarity inspired by Jaccard similarity coefficient [31]. The structural similarity between two vertices is defined as the weighted ratio of common neighbors to all the neighbors of both vertices. One of the key aspects of our approach is to consider contextual similarity to attain structural cohesiveness among nodes. Its importance is evident from the applications where the nodes emerge in different contexts. For instance, in social network, the users are represented by nodes and edges reflect their relationships. Each user can have different roles or contexts like occupation as student, doctor, engineer, or designer. The combine effect of structure and context relevance among vertices is defined in Eq. (1). The parameter α is introduced to control the influence of both similarity aspects.

$$CSIM_{(v_a,v_b)} = \alpha * SIM^{struct}_{(v_a,v_b)} + (1-\alpha) * SIM^{context}_{(v_a,v_b)} \quad (1)$$

$$DIST_{(v_a,v_b)} = \begin{cases} \frac{1}{CSIM_{(v_a,v_b)}}, \ e(v_a,v_b) \in \mathrm{E} \\ \sum_{i=0}^{len(p)} \frac{1}{CSIM_{(v_{pi},v_{pi+1})}}, \\ \ p(v_a \dots v_b) \ and \ v_{pi} \in \mathrm{V} \\ \infty, \ otherwise \end{cases} \quad (2)$$

The collaborative similarity value is calculated for indirectly connected vertices by following a path. However, a pair of vertices can have multiple paths. We choose the shortest path as a candidate path to estimate the similarity value. In order to extend the similarity using shortest path approach, we must utilize the desirable property, i.e., similarity (distance) value decreases (increases) as we move far from the source vertex. We achieve this by taking the reciprocal of similarity measure by defining a distance function in Eq. 2. The distance value in close proximity is expected to be low due to transitivity property. In a weighted graph, shortest path between two vertices may not be unique. We pick the one with least distance value at initial expansions. The time complexity for estimating the collaborative similarity or distance among all pair of vertices is non-linear. However, the symmetric property and shortest path strategy, which is done efficiently in order of $O(|V|^2 log|V|)$ instead $O(|V|^3)$, where V and E are the set of vertices and edges, respectively, in a graph G.

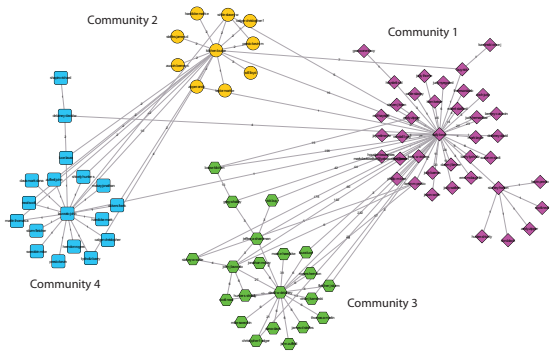


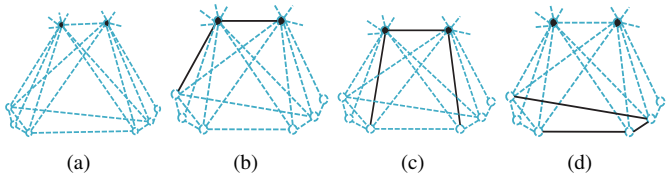
Fig. 1. [32] Sally-Beck's Four Personalized Communities.

Application To elaborate the effectiveness of the proposed similarity measure, we present a personalized community detection method over an email network [32], which is solely based on emails extracted from email account. Personal emails of a user describe social activities that are transformed to an undirected weighted graph for structural and semantic analysis. Each user, i.e., either sender or receiver, is represented by a node and an edge reflects shared emails, where frequency is associated as an edge weight. The first phase extracts the communication patterns of interest (CPI) using emails as informative features to describe the communication behavior of each user. Subsequently, the second phase detects user communities via an intra-graph clustering method [30] by contemplating structural and semantic aspects together. We validate the effectiveness of the proposed technique on real email dataset in terms of various performance measures, i.e., density, entropy, and f-score. The personalized communities are visualized in Fig. 1.

III. Shortest Path Traversal Analysis

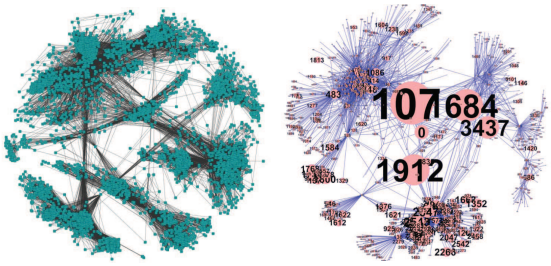
Our intention is to empirically validate the traversal patterns shown in Fig. 2, i.e., high degree vertices have higher probability to be encountered during majority shortest path traversals. Consequently, we have the following concerns towards graph traversals; (a) Identifying frequently occurring vertices (b) the role of high degree vertices, i.e., hub-nodes (c) speculating effective regions for auxiliary data using social graph properties, e.g. degree distribution (d) participation of set of vertices together in SPs (e) repetitive exploration of graph regions.

We provide an empirical analysis on continuous overlapped regions (COREs) through SP traversals in social networks [34][33]. The term CORE refers to the portion of the graph which is traversed through multiple shortest paths. In other words, it is a sequence of adjacent vertices or edges. First, we compute the shortest paths between set of vertices. Each



(a) (b) (c) (d)

Fig. 2. [33] Shortest Path (SP) Traversal Patterns (a) Toy graph with two high degree nodes and (b) SP from low to degree region vertices (c) SP passing through high degree vertices (d) SP followed by low degree vertices.



(a) (b)

Fig. 3. [34] Visualization of the Facebook Network (a) Original graph, and (b) Shortest path traversals.

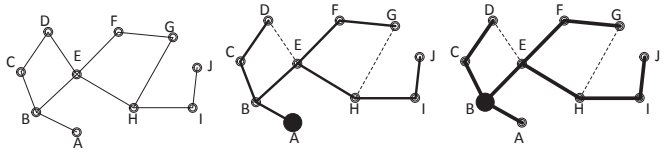
shortest path is considered as one transaction. Second, we utilize the pattern mining approach to identify the frequency of occurrences of the vertices in all transactions. We also provide statistical analysis in terms of network properties, e.g. degree distribution, average shortest path, and clustering coefficient. The CORE provides an opportunity for constructing space efficient auxiliary data to speedup SP queries. The contributions of this study are as follows:

- Empirically prove that the significant amount of shortest paths are overlapped.
- The behavior of the overlapped regions in diverse networks, e.g. Scale free networks.
- The impact of hub-nodes on the shortest paths, e.g. What portion of the shortest paths are pass through the hub nodes or across dense regions?
- Visual analysis on the coverage of the entire graph through shortest paths.

After extensive experiments on real datasets, we observed that the degree distribution of the vertices is preserved in terms of frequency of occurrences of vertices during shortest path traversal, as shown in Fig. 3. It shows that high degree vertices are encountered in majority shortest paths. We have also noticed that the average degree vertices are less likely to appear in SPs. The visual description of the shortest path trajectories have preserved the original structure of the graph.

IV. Shortest Path Overlap Regions for Pre-Computation

We introduce a novel concept based on SP overlapped region (SPORE) [28], i.e., the portion of the graph which is traversed through multiple SPs. The process of answering



(a) (b) (c)

Fig. 4. Occurrence of vertices (or edges) through SPs (a) Original graph with unit edge weights, *i.e.,* 1.0 (b) SSSP traversal from vertex A (c) SSSP traversal from vertex B.

SP queries, i.e., global search, over a large graph requires traversing the graph repeatedly. During this repetitive traversal process, we observe that a set of edges is visited frequently. This reveals that a set of SPs shares a sequence of vertices (or edges) among them. We illustrate the concept of SP overlap by visiting edges multiple times in Fig. 4. We have an original undirected graph with unit edge weights. We perform an SSSP search from vertices A and B successively. The solid line of an edge represents the pair of vertices of an edge being visited during SP traversal and the strength reflects the frequency of occurrence. Therefore, the vertices are expected to be visited again and again.

The SP computation for mining large graphs, *e.g.,* graph clustering, is a computationally intensive task. For instance, the naïve approach for graph clustering using the K-means framework requires all pairs of SP computations in each cluster to update the centroids, *i.e.,* $O(|V|^2)$, where $|V|$ is the number of vertices in a graph. Shortest path searching generally adopts a BFS to traverse a graph. A large-scale sparse graph exhibits longer SPs. Therefore, a BFS requires a large number of iterative expansions for huge graphs. In order to overcome this problem, several algorithms [22][35][36][37][38][39] use a two-stage framework. In the first stage, it computes auxiliary data, such as additional edges (shortcuts) and labels or values associated with vertices or edges. In the second stage, the auxiliary data is then used to accelerate point-to-point (from the source to destination vertex) SP queries, typically by pruning or directing Dijkstra's algorithm [19] with fewer expansions. The pre-processing is practical and produces a modest amount of auxiliary data [27].

We propose an effective strategy to construct the potential shortcuts, using the SPORE idea. The shortcuts constructed through the proposed idea are called SPOREs. We use the recent computation for the SPORE construction, instead of a blind or exhaustive approach. Our aim is to add SP segments, *i.e.,* SPOREs, in an incremental way based on few recent SP traversals. We believe that recently visited edges are expected to be visited in successive iterations, which we call potential edges.

For instance, we have a distance graph prior to clustering as shown in the Fig. 5. We compute the SSSP from vertex A then we get an SP-tree rooted at A which has four levels. We traverse the SP-tree to construct the shortcuts. At each level of the SP-tree, from top to bottom, we add a shortcut to its children except for the direct descendants. In this way, we are not required to explicitly compute SPs from intermediate nodes in the SP-tree to their child nodes using the original graph, *e.g.,* from C to E and F are SPOREs.

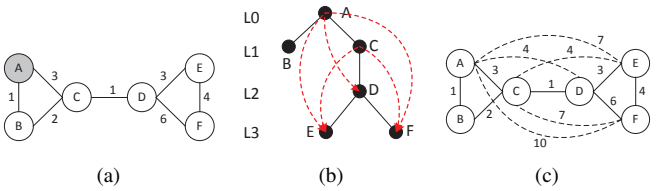


Fig. 5. Augmenting graph with shortcuts using SPORE through SP-tree. (a) Transmuted distance graph, (b) Potential shortcuts from SP-Tree rooted at A are presented as dotted lines, and (c) Augmented graph.

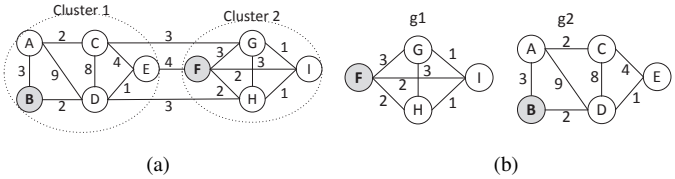


Fig. 6. Restricted subgraph computation, (a) Graph clusters after centroid initialization, and (b) Subgraphs from each cluster without inter-cluster edges for cluster updates.

V. Restricted Traversals and Graph Clustering

The notion of subgraph computation helps us to reduce the search space for SP computation. We propose a heuristic approach to compute all SP pairs on a restricted graph [28]. It has marginal impact on both the overall distance computation and convergence of the clustering algorithm. Assume that after good centroid initialization we get clusters, as shown in Fig. 6(a). We partition the graph into subgraphs, *e.g.,* g1 and g2 in Fig. 6(b), where each cluster is represented as a subgraph by excluding the inter-cluster edges. The search space for SP traversal in a subgraph is much smaller than the entire graph. In this way, we reduce the overall computation time for APSP to update the centroids.

Intuitively, an SP between a pair of vertices can follow inter-cluster edges. This means that an SP can have a transition from one cluster to another along the way. However, this phenomenon rarely happens in real life graphs due to hub nodes, *i.e.,* high degree nodes which connect dense regions. The transition probability from hub nodes to the other nodes of the graph is too low. The distance is estimated as the linear addition of edge weights on a transmuted graph. Therefore, a longer path passing through inter-cluster edges following a hub node leads to a high distance value.

VI. Parallel Shortest Path Computation

In this section, we explain our strategy to speed-up the pair-wise SP computations during the clustering process. We achieve parallelism by computing the set of SPs concurrently [28]. In practice, each SP can be computed independently with the static graph.

The SPs are either computed in a sequential or parallel manner. The former approach computes multiple SSSPs from each source one after another, *i.e.,* the element approach. The traversal of an entire graph for each SSSP in sequential order incurs an enormous amount of latency for the entire clustering procedure. We reduce the overall latency through parallelism,

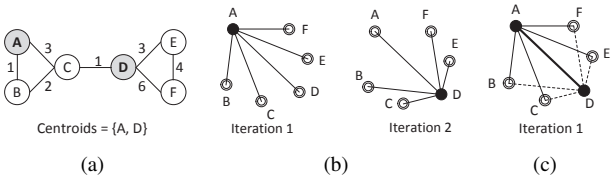


Fig. 7. SSSP computation from two vertices, (a) Original graph, (b) Element approach with sequential order, and (c) Set approach with parallel strategy.

by traversing the graph for multiple SSSPs at the same time, *i.e.,* the set approach. For instance, we need to compute the SSSP from two source vertices, A and D, as shown in Fig. 7(a). The SSSPs for A and D require two iterations where two exclusive scans of the entire graph are required for traversal; this is presented in Fig. 7(b). We parallelize it by unifying the set of SSSP traversals to reduce the total number of iterations. However, we do not explicitly process it with a multi-threaded program for simplicity.

VII. Conclusion and Open Challenging Issues

In this study, we improved the time complexity of pair wise similarity computation by introducing collaborative similarity measure using shortest path for intra-graph clustering problem. We have provided an extensive empirical analysis for shortest path traversals in real life graphs to identify useful patterns. We also presented SPORE, an effective, scalable, and performance efficient shortcut construction strategy for shortest path computation. The restricted and set of shortest path traversals further reduce the overall computation overhead. These optimizations show the significant improvement in graph clustering process.

However, We have few open issues which need to be solved in near future. To predict an appropriate number of clusters in a graph is essential with linear clustering frameworks. The set approach has a trivial issue for the APSP computation through multiple SSSPs on billion scale graphs. It requires significant storage for maintaining the intermediate path expansions for each SP. It becomes non-trivial when we deal with scale-free graphs, where vertices follow a power law distribution. An empirical justification for shortest path traversals and overlapped regions on directed and weighted graphs are yet to be made.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No.2014R1A2A1A05043734). We thank anonymous reviewers who provided comments that greatly improved the manuscript.

References

[1] R. Xu and I. Wunsch, D., "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, May 2005.

[2] S. E. Schaeffer, "Survey: Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, aug 2007. [Online]. Available: http://dx.doi.org/10.1016/j.cosrev.2007.05.001

[3] H.-P. Kriegel, P. Kroger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Trans. Knowl. Discov. Data*, vol. 3, no. 1, pp. 1:1–1:58, Mar. 2009. [Online]. Available: http://doi.acm.org/10.1145/1497577.1497578

[4] V. M. Satuluri, "Scalable clustering of modern networks," Ph.D. dissertation, The Ohio State University, 2012.

[5] B. Boden, "Combined clustering of graph and attribute data," in *Dissertation, Fakultt fr Mathematik, Informatik und Naturwissenschaften, RWTH Aachen University*. Aachen: Apprimus-Verlag, 2014.

[6] S. Cohen, B. Kimelfeld, and G. Koutrika, "A survey on proximity measures for social networks," in *Search Computing*, ser. Lecture Notes in Computer Science, S. Ceri and M. Brambilla, Eds. Springer Berlin Heidelberg, 2012, vol. 7538, pp. 191–206. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-34213-4_13

[7] A. Lada and A. Eytan, "How to search a social network," *Social Networks*, vol. 27, no. 3, pp. 187–203, 2005.

[8] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007. [Online]. Available: http://dx.doi.org/10.1002/asi.20591

[9] E. Cohen, D. Delling, F. Fuchs, A. V. Goldberg, M. Goldszmidt, and R. F. Werneck, "Scalable similarity estimation in social networks: Closeness, node labels, and random edge lengths," in *Proceedings of the First ACM Conference on Online Social Networks*, ser. COSN '13. New York, NY, USA: ACM, 2013, pp. 131–142. [Online]. Available: http://doi.acm.org/10.1145/2512938.2512944

[10] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Proceedings of the Sixth International Conference on Data Mining*, ser. ICDM '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 613–622. [Online]. Available: http://dx.doi.org/10.1109/ICDM.2006.70

[11] G. Jeh and J. Widom, "Simrank: A measure of structural-context similarity," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 538–543. [Online]. Available: http://doi.acm.org/10.1145/775047.775126

[12] B. Bollobas, *Modern Graph Theory*. Springer, jul 1998. [Online]. Available: http://www.worldcat.org/isbn/0387984887

[13] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953. [Online]. Available: http://dx.doi.org/10.1007/BF02289026

[14] H.-H. Chen and C. L. Giles, "Ascos: an asymmetric network structure context similarity measure." in *ASONAM*, J. G. Rokne and C. Faloutsos, Eds. ACM, 2013, pp. 442–449.

[15] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, p. 026113, Feb 2004. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.69.026113

[16] Y. Tian, R. A. Hankins, and J. M. Patel, "Efficient aggregation for graph summarization," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '08. New York, NY, USA: ACM, 2008, pp. 567–580. [Online]. Available: http://doi.acm.org/10.1145/1376616.1376675

[17] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," *Proc. VLDB Endow.*, vol. 2, no. 1, pp. 718–729, Aug. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1687627.1687709

[18] H. Cheng, Y. Zhou, and J. X. Yu, "Clustering large attributed graphs: A balance between structural and attribute similarities," *ACM Trans. Knowl. Discov. Data*, vol. 5, no. 2, pp. 12:1–12:33, feb 2011. [Online]. Available: http://doi.acm.org/10.1145/1921632.1921638

[19] E. W. Dijkstra, "A note on two problems in connexion with graphs," *NUMERISCHE MATHEMATIK*, vol. 1, no. 1, pp. 269–271, 1959.

[20] D. Delling, P. Sanders, D. Schultes, and D. Wagner, "Engineering route planning algorithms," in *Algorithmics of Large and Complex Networks*, J. Lerner, D. Wagner, and K. A. Zweig, Eds. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 117–139. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02094-0_7

[21] C. Sommer, "Shortest-path queries in static networks," *Submitted to ACM Computing Surveys*, 2012.

[22] J. Gao, R. Jin, J. Zhou, J. X. Yu, X. Jiang, and T. Wang, "Relational approach for shortest path discovery over large graphs," *Proc. VLDB Endow.*, vol. 5, no. 4, pp. 358–369, dec 2011. [Online]. Available: http://dx.doi.org/10.14778/2095686.2095694

[23] A. D. Zhu, H. Ma, X. Xiao, S. Luo, Y. Tang, and S. Zhou, "Shortest path and distance queries on road networks: Towards bridging theory and practice," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '13. New York, NY, USA: ACM, 2013, pp. 857–868. [Online]. Available: http://doi.acm.org/10.1145/2463676.2465277

[24] A. D. Zhu, X. Xiao, S. Wang, and W. Lin, "Efficient single-source shortest path and distance queries on large graphs," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: ACM, 2013, pp. 998–1006. [Online]. Available: http://doi.acm.org/10.1145/2487575.2487665

[25] T. Akiba, Y. Iwata, and Y. Yoshida, "Fast exact shortest-path distance queries on large networks by pruned landmark labeling," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '13. New York, NY, USA: ACM, 2013, pp. 349–360. [Online]. Available: http://doi.acm.org/10.1145/2463676.2465315

[26] D. Delling, A. V. Goldberg, T. Pajor, and R. F. Werneck, "Robust exact distance queries on massive networks," Microsoft Research, USA, Tech. Rep., 02 2014.

[27] I. Abraham, D. Delling, A. Fiat, A. V. Goldberg, and R. F. Werneck, "Highway dimension and provably efficient shortest path algorithms," Microsoft Research, USA, Tech. Rep., 09 2013.

[28] W. Nawaz, K.-U. Khan, and Y.-K. Lee, "Spore: Shortest path overlapped regions and confined traversals towards graph clustering," *Applied Intelligence (Accepted)*, Dec. 2014.

[29] W. Nawaz, Y.-K. Lee, and S. Lee, "Collaborative similarity measure for intra graph clustering," in *DASFAA Workshops*, 2012, pp. 204–215.

[30] W. Nawaz, K.-U. Khan, Y.-K. Lee, and S. Lee, "Intra graph clustering using collaborative similarity measure," *Distributed and Parallel Databases (Accepted)*, Dec. 2014.

[31] P. Jaccard, "Étude comparative de la distribution florale dans une portion des alpes et des jura," *Bulletin del la Société Vaudoise des Sciences Naturelles*, vol. 37, pp. 547–579, 1901.

[32] W. Nawaz, Y. Han, K.-U. Khan, and Y.-K. Lee, "Personalized email community detection using collaborative similarity measure," *The 5th International Conference on Data Mining and Intelligent Information Technology Applications (ICMIA-2013)*, vol. abs/1306.1300, 2013.

[33] W. Nawaz, K.-U. Khan, and Y.-K. Lee, "Core analysis for efficient shortest path traversal queries in social graphs," *The 7th IEEE International Conference on Social Computing and Networking - SocialCom-2014, BDCloud-2014 (Accepted)*, vol. 0, 2014.

[34] W. Nawaz, K. U. Khan, and Y.-K. Lee, "Shortest path analysis for efficient traversal queries in large networks," *Proc. Contemporary Engineering Sciences*, vol. 7, no. 16, pp. 811–816, jul 2014. [Online]. Available: http://dx.doi.org/10.12988/ces.2014.4696

[35] P. Sanders and D. Schultes, "Engineering highway hierarchies," *J. Exp. Algorithmics*, vol. 17, pp. 1.6:1.1–1.6:1.40, sep 2012. [Online]. Available: http://doi.acm.org/10.1145/2133803.2330080

[36] A. V. Goldberg, H. Kaplan, and R. F. Werneck, "Reach for a*: Efficient point-to-point shortest path algorithms," in *The Shortest Path Problem: Ninth DIMACS Implementation Challenge*. American Mathematical Society, USA, 2009, pp. 93–139.

[37] H. Bast, S. Funke, P. Sanders, and D. Schultes, "Fast routing in road networks with transit nodes," *Science*, vol. 316, no. 5824, p. 566, apr 2007. [Online]. Available: http://www.mpi-inf.mpg.de/ funke/Papers/SCIENCE07/SCIENCE07.pdf

[38] R. Geisberger, P. Sanders, D. Schultes, and C. Vetter, "Exact routing in large road networks using contraction hierarchies," *Transportation Science*, vol. 46, no. 3, pp. 388–404, aug 2012. [Online]. Available: http://dx.doi.org/10.1287/trsc.1110.0401

[39] I. Abraham, D. Delling, A. V. Goldberg, and R. F. F. Werneck, "Hierarchical hub labelings for shortest paths." in *ESA*, ser. Lecture Notes in Computer Science, L. Epstein and P. Ferragina, Eds., vol. 7501. Springer, 2012, pp. 24–35. [Online]. Available: http://dblp.uni-trier.de/db/conf/esa/esa2012.html#AbrahamDGW12