Exact and Heuristic Algorithms for Some Spatial-aware Interest Group Query Problems

Chih-Yang Huang[1], Po-Chuan Chien[2], Yen Hung Chen[3]

[1] Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
[2] Research Center for Information Technology Innovation, Academia Sinica, Taiwan
[3] Department of Computer Science, University of Taipei, Taiwan
user@hcy.idv.tw, dibery@citi.sinica.edu.tw, yhchen@utaipei.edu.tw

Abstract

Location-based social networks are important issues in the recent decade. In modern social networks, such as Twitter, Facebook, and Plurk, attempt to get the accurate address positions from their users, and try to reduce the gap between virtuality and reality. This paper mainly aims at both the interests of Internet users and their real positions. This issue is called the spatial-aware interest group query problem (SIGQP). Given a user set U with n users, a keyword set W with m words, and a spatial object set S with s items, each of which contains one or multiple keywords. If a user checks in a certain spatial object, it means the user could be interested in that part of keywords, which is countable to clarify the interests of the user. The SIGQP then tries to find a k-user set U_k, $k \leq n$, such that the union of keywords of these k users will equal to W, and additionally, the diameter (longest Euclidean distance of two arbitrary users in U_k) should be as small as possible. The SIGQP has been proved as NP-Complete, and two heuristic algorithms have been proposed. Extended from SIGQP, another problem is in finding the smallest k for U_k to cover all the keywords, with the users' distance as the secondary criterion, called as "minimum user spatial-aware interest group query problem" (MUSIGQP). This paper designs a branch & bound method and a measure & conquer method to solve SIGQP and MUSIGQP respectively, and a performance analysis is given.

Keywords: Spatial-aware interest group query problem, Exact algorithms, Branch & bound, Measure & conquer, Location-based service

1 Introduction

Mobile phone positioning and social network have been seen as significant and innovative issues in the recent decade [1]. The goal of group queries in location-based social networks is to find a group of users where members are close to each other and have the same interests [2-5].

Li et al. proposed the spatial-aware interest group query problem (SIGQP) in order to combine the issues of positioning and group queries in location-based social networks [6]. Let $U = \{u_1, u_2, ..., u_n\}$ be the user set with n users, $S = \{p_1, p_2, ..., p_s\}$ be the spatial object set of s spatial objects, and $W = \{w_1, w_2, ..., w_m\}$ be the keyword set of m query keywords. Each spatial object contains one or more keywords. It is feasible to count the value of interests of a user if it checks in some spatial objects. The normalized interest value of user u over keyword w is indicated as $I(u, w)$, which is illustrated in Equation 1, where $count(u, p)$ is the total time of the user u checking in the spatial object p, and the collection of i spatial objects that user u has been is denoted as $S_u = \{p'_1, p'_2, ..., p'_i\}$.

$$I(u,W) = \frac{count(u,w)}{\Sigma_{p \in S_u} count(u,p)} \quad \textbf{(1)}$$

We then furthermore consider a more general case where a user is interested in multiple keywords, collected as a set W'. The interest score $I(u, W')$ over that keyword group is evaluated as the aggregation of the interest values of all keywords in W' (i.e., $\sum_{w \in W'} I(u,w)$). A spatial-aware interest group (SIG) query is defined to find a user set U_k with k users, $k \leq n$, to cover all m keywords. Furthermore, an adjustment α is introduced so that we can adjust the weight between the interest and the distance of the users, and the maximization objective function (where the maximization target is the $Rank$ of the selected U_k) is defined in Equation 2 [6].

$$Rank_\alpha(U_k)= \alpha \times \min\{I(u, W) | u \in U_k\} + (1-\alpha)(D(U)-D(U_k)) \quad (2).$$

$D(U)$ and $D(U_k)$ are the diameters of the user sets U and U_k, which is the farthest Euclidean distance between any two users in that set, and can be calculated by

$$\max_{i,j \in U_k} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

*Corresponding Author: Yen Hung Chen, E-mail: yhchen@utaipei.edu.tw
DOI: 10.3966/160792642020072104028

When $\alpha = 1$, the distance of users would be irrelevant and using the fewest group of users to cover all the keywords will be the main issue. We call this problem as a minimum user spatial-aware interest group query problem (MUSIGQP) and highlight on the keywords coverage over the users' distance. In this case, the minimum set cover (MSC) method, an exact algorithm, can be applied, and we can derive an exact answer toward MUSIGQP. Because MSC uses the least number of sets to cover all keywords, the result of MUSIGQP will consist of as least groups as possible. However, this would totally ignore the factor of the distances, so in later sections, we will show how to deal with the distances between users by integrating the concept of distance into the MSC algorithm.

Our contribution is as follows: We provide a branch & bound method to solve SIGQP and analyze it. To our knowledge, there is no MUSIGQP research for the design of non-trivial brute force exact algorithms, and hence we provide a precise solution to the problem.

In real-world applications, every check-in place could be seen as a spatial object p in SIGQP (MUSIGQP), and there are some features or keywords for those places. For example, Starbucks is representative of the words: "food", "coffee", or "drink", while VieShow Cinemas is the symbol of the words: "movie" and "popcorn". Also, every word could be seen as a keyword in SIGQP (MUSIGQP). If a user checks in a Starbucks coffee shop, it means the user would be interested in words (keywords) like "coffee" or "drink". Our algorithms on SIGQP (MUSIGQP) consider not only the keywords but also the users' distances. It finds users having common interests and staying in near spots. Broadcasting the advertisements and marketing are possible applications for SIGQP (MUSIGQP).

2 Related Work

With the development of location-aware devices, ubiquitous Internet, and the techniques of social computing, access to the position as well as social information of a user are more easily available. A lot of researches [2-9] are aiming at finding a group of users who are close and are interested in the common issues, based on group queries in location-based social networks. For example, the issue of expert collaboration queries is to find a group of experts where members know all the skills, and their distances are not far.

If we consider spatial query processing, R-tree and R* tree are available. In the recent 30 years, there are a lot of queries we can use, such as k-nearest-neighbor queries [10-14], range queries [15-16], and closest-pair queries [17-19]. There are also some researches [20-25] combining spatial query processing and keywords. In addition, some researches propose group and team query on social network [2-3, 7-8, 26-27]. Li and the others identify $Rank_\alpha(U_k)$ and SIGQP [6, 28-30] and develop two heuristic algorithms, based on the technique of IR-tree [20]. They reduce the set cover problem into a SIGQP and prove it as NP-Complete [31]. However, SIGQP is similar to maximum coverage problem [32-33]. It can be solved by $(1 - 1/e)$-approximation algorithm [32] ($e \approx 2.718$) through Greedy skills. Its lower bound of approximation rate is $(1 - 1/e) + o(1) \approx 0.632$ [32-34],

3 Problem Formulation & Solution Approach

In a SIGQP, we consider a situation where a user set $U = \{u_1, u_2, ..., u_{|U|}\}$, a spatial object set $S = \{p_1, p_2, ..., p_{|S|}\}$, and a query keyword set $W = \{w_1, w_2, ..., w_{|W|}\}$ exist, and satisfy $\forall\ u \in U$, $\exists p \in S \wedge p \in S_u$ (S_u is places that user u has been to, and S_u is not empty, see Table 1.) and $\forall\ p \in S$, $\exists\ w \in W \wedge w \in W_p$, from which we can infer that an element $u \in U$ is factually a set of keywords $w \in W$, and that every user in the user set will be inherently associated with a group of keywords using the transitive law.

Table 1. Symbols shown in this paper

Symbol	Description
U	A user set with n users with a keyword set and a spatial object set attributes, which is $\{u_1, u_2, ..., u_n\}$.
U_u	The union of all keywords that are bound to any spatial object in S_u for user u (see below).
W	A keyword set with m words, which is $\{w_1, w_2, ..., w_m\}$.
W_p	The collection of words that spatial object p has, which is $\{W_{p1}, W_{p2}, W_{p3}, ..., W_{p\|Wp\|}\}$
S	A spatial object set with s items, which is $\{p_1, p_2, ..., p_s\}$.
S_u	The collection of all i spatial objects that user u has been, which is $\{p'_{u1}, p'_{u2}, ..., p'_{ui}\}$.
$D(U)$	The farthest Euclidean distance between two arbitrary users in the user set U.
$I(u, w)$	Interest score of user u over the keyword (set) w.
α	Weight adjustment coefficient between the distance and the weight.

Since the goal of a MUSIGQP is to select a user set $U' \subseteq U$ such that all keywords in W are covered, we can, without loss of generality, eliminate the spatial object group S from the problem and construct the direct relationship between a user and a keyword as a bipartite graph. Because our goal is still to select a U' such that the union of keywords from all users in U' will be equal to W, we can restate the MUSIGQP as a minimum set cover (MSC) problem, by regarding W as the universe set, and U as the collection of sets. The ***user_set*** is denoted as a group of users and each of user has its position as well as part of keywords.

In the following subsections, we will be concentrated on the branch & bound method for SIGQP as well as the exact solution approach of the minimum set cover problem on MUSIGQP. The symbols used in our problems definition are organized in Table 1.

3.1 Branch & Bound Method

We design a heuristic algorithm to solve SIGQP. The $Rank_{\alpha}$ in Figure 1 is derived from Equation 2. We modify Equation 2 and present it as below for the branch & bound method.

```
user_set BranchAndBound (user_set U) {
Initial:
user_set the_ goal_ set // the goal set
double ratio // set the bound to stop iteration
int   k // find k users and put them into the goal set

for each user set in U {
 Find a user with the most keywords and put it in the_
 goal_ set .
}

if (|U| = 0)   return   ϕ.   // base case

while (|the_ goal_ set|≤ k × ratio) {
  Find a user with high rank and add it to the_goal_set
}
if (|the_goal_set|> k × ratio) {
 Find the rank of the remaining users.
 Put them into the goal set based on the rank until
 |the_goal_set |= k.
}
return the_goal_set.
}
```

Figure 1. Branch and bound method for SIGQP

$$Rank_{\alpha}(u_x,u_y)=\alpha\times\min\{|u_x|,|u_y|\}+(1-\alpha)\times(D(U)-D(u_x,u_y)) \quad \textbf{(3)}$$

There is a coefficient, α, set as 0.5. $D(U)$ means the farthest distance between any two users in the set U as Figure 5. $D(u_x,u_y)$ means the distance between the users u_x and u_y. Based on Figure 1, we can find the approximation set for SIGQP. The performance will be discussed later.

3.2 Measure & Conquer Exact Algorithm for MSC

Fomin, Grandoni, and Kratsch proposed an algorithm to exactly solve the MSC problem by finding the minimum number of sets required to cover all the keywords [35]. However, such number does not help when we wish to find out which sets are chosen. Therefore, we modify this algorithm, which is illustrated in Figure 2, so that the algorithm returns the actually selected sets.

For a user set U, when each user u in U covers at most 2 keywords in W, finding the minimum user to cover all keywords, denoted by $2\_msc(U)$, can be

```
user_set msc (user_set U) {
if (|U | = 0)
 return ϕ. // base case (empty set)
if (∃ u_A, u_B ∈ U and u_A ⊆ u_B) // u_B has all u_A's keyword
 return msc (U \{u_A}). // u_A is useless as u_B is better
if (only user u ∈ U has a specific keyword)
 return u ∪ msc (del (u, U)). // must select u
take u ∈ U of maximum cardinality.
if (|u| = 2) // this user has only 2 keywords
return 2_msc (U). // 2_msc (U) is a kind of Graph
 Matching
return min{msc(U\{u}), u ∪ msc (del (u, U))}.
 //recursion, try with u either selected or not selected
}
```

Figure 2. Algorithm msc for the minimum set cover problem on MUSIGQP

solved in polynomial time by the graph matching algorithm [31, 35-36]. Additionally, during the recursion process, the notation of $U\backslash\{u\}$ means that we remove user u from the user universe set U to the next recursion step as if it never exists (i.e., we do not choose u in the final result), and the notation of $del(u, U)$ means that we select user u in the final result, and therefore all keywords of u are marked as found (and u is eliminated) when the recursion steps down.

3.3 Measure & Conquer Exact Algorithm for MSC with Consideration of Distance

When there are multiple combinations of groups that can satisfy the criteria that every keyword is selected, we would show an inclination to choose the group whose diameter, calculated by the max distance between user (i, j) in the group, is smaller. The standard method described in the above subsection does not, however, take the Euclidean distance between users into consideration; consequently, such a method may not be able to derive the desired solution.

To handle such situations, we will need to take the distance between users into account. The modified version, denoted as exact_MUSIGQP later in this paper, is illustrated in Figure 3.

```
user_set exact_ MUSIGQP (user_set U) {
if (|U| = 0)
 return ϕ. // base case
if (∃ u_A, u_B ∈ U and u_A ⊆ u_B) // u_B has all u_A's
 keyword
 return msc (U \{u_A}). // u_A is useless as u_B is better
if (only user u ∈ U has a specific keyword)
 return u ∪ msc (del (u, U)). // must select u
 take u ∈ U of maximum cardinality.
 if (|u| = 2) // this user has only 2 keywords
return 2_msc (U). //2_msc (U) is a kind of Graph
 Matching
return min_dist (msc (U \{u}), u ∪ msc (del (u, U))).
}
```

Figure 3. Algorithm exact_MUSIGQP with consideration of distance on MUSIGQP

In the last line, we change the min to *min_dist* (), which will evaluate the set size first, and when the size of the two sets are equal, the group who has a smaller diameter will be returned. The algorithm of *min_dist* () is presented in Figure 4, description of the *diameter* () function is given in Figure 5.

```
user_set min_dis( user_set grpA, user_set grpB) {
if | grpA | < | grpB |
return grpA.
else if | grpA | > | grpB |
return grpB.
else if diameter (grpA) < diameter (grpB) // size equal
return grpA.
return grpB. // the last possibility
}
```

Figure 4. Choose the set with a smaller size and a smaller diameter

```
diameter ( user_set grp) {
d = 0 // the diameter
for each element pair (A, B) in grp
if Euclidean_distance (A, B) > d
d = Euclidean_distance (A, B).
return d.
}
```

Figure 5. Measure the diameter of a group

Regarding the correct solution set as *Opt*, we can say that a user set u may or may not be in *Opt*. If $u \in Opt$, then the solution is then further found by selecting u and remove all keywords in u from other existent users; otherwise, the solution is found by eliminating u from the user set. In this way, we are capable of traversing all possible combination of groups, and because in each step of the recursion, we choose the one with a smaller diameter, in the global scope, we can achieve the optimal solution.

In the next section, we will mainly utilize this method to solve MUSIGQP.

4 Performance Evaluation

All algorithms are implemented in Java programming language. The models of the CPU and RAM are Intel Core i7 2.2 GHz and 16 GB 1600 MHz DDR3, respectively.

4.1 Branch & Bound Method

To test the performance of the branch & bound method, we test it on a situation where 100 users, from which we will select 20 as the solution, exist in the square within (0, 0) and (10, 10) and calculate the similarity with the greedy solution. In addition, we set the size of keywords to 50, and α to 0.5. The experimental result is plotted in Figure 6, with each ratio conducted 100 times and take the average.

It can be easily inferred that by the increase in the branch ratio, we can get better results, as the trend shown in the figure. In addition, we may observe that the similarity has reached 0.769 when the branch ratio is 0.1, and that there is an outstanding breakthrough when the branch ratio has reached 0.5, both of which imply that we can reduce the calculation load while still obtaining favorable results, and one can choose the most suitable ratio, depending on the trade-off between accuracy and time. We also plot the execution time of the branch & bound method in logarithm base 2 as Figure 7.

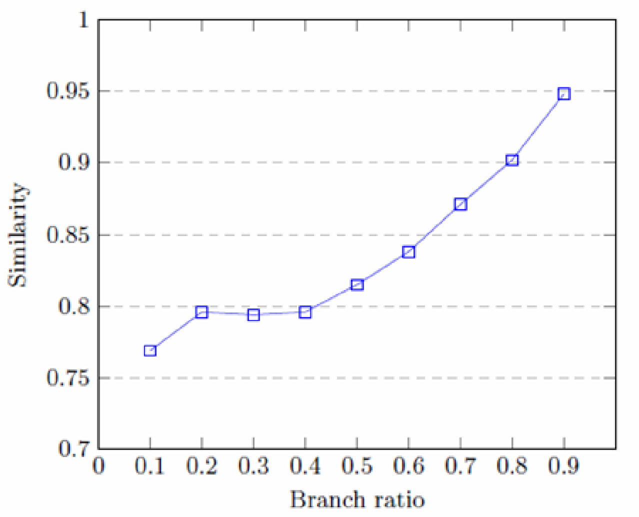

Figure 6. The similarity trend between the branch & bound method and the greedy algorithm

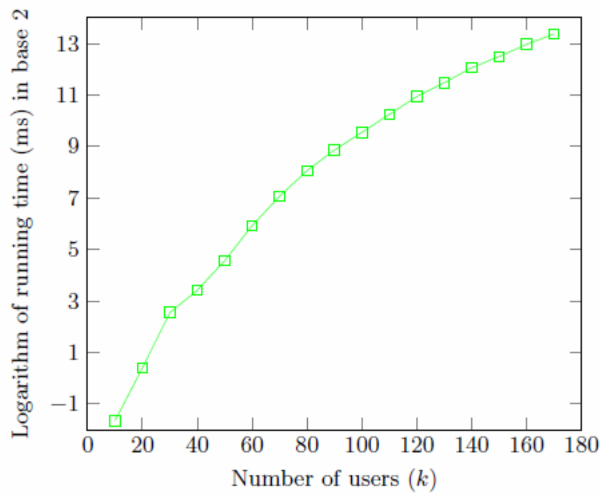

Figure 7. Running time versus the value of k users in heuristic algorithm

4.2 Measure & Conquer Exact Algorithm for MSC with Consideration of Distance

To each of our algorithms, an experiment is conducted. We test them on a situation where k users, whose value within the interval [10, 170], $k/2$ keywords(interests), and all k users will stand in the position where $0 \le x \le k$, $0 \le y \le k$ in Euclidean space.

Each user has a random group of keywords and a random position. The experimental result is plotted in Figure 8, with every k users conducted 100 times and take the average. Owing to the fact that the time complexity of this algorithm is in exponential time, we represent the logarithm of running time in base 2 as Figure 8.

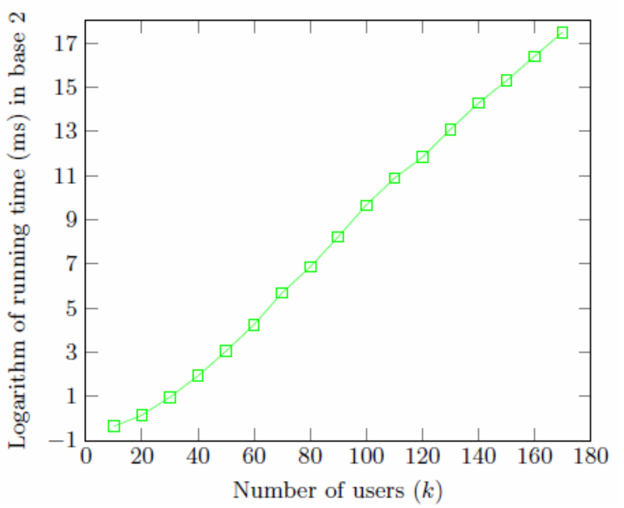

Figure 8. Running time versus the value of k users in exact algorithm

It is inferred that the transformation of MUSIGQP into MSC algorithm is feasible. The time complexity of the designed algorithm would be $O(2^{0.465n})$ upper bound. On the other hand, $\Omega(2^{0.396n})$ and $\Omega(2^{0.142n})$ would be the possible lower bounds. This theorem is on the ground of a measure & conquer approach for the analysis of exact algorithms described in [35]. Thus, we could decrease the time complexity of MUSIGQP and transform MUSIGQP into MSC.

5 Conclusion

In this paper, we investigated the spatial-aware interest group query problem and provide two algorithms on SIGQP and MUSIGQP, with their performance being analyzed. One is the branch & bound method and the other is the measure & conquer exact algorithm.

The branch & bound method is a heuristic algorithm whose time complexity is $O(n^3)$. Although this solution provides, rather than the correct answer, only an approximation, it is capable of reducing the time complexity to polynomial time, which is much affordable even when the size scales out.

The other algorithm is based on the MSC algorithm proposed in [35]. We transform SIGQP into MSC. The first trial exact algorithm (Measure & Conquer Exact Algorithm for MSC) doesn't take the Euclidean space into consideration, while the second one (Measure & Conquer Exact Algorithm for MSC with Consideration of Distance) is concerned about the distance. Both of them take $O(2^{0.465n})$ upper bounds, which is lower than $O(2^n)$, the time complexity of a brute force approach.

All in all, we provide these algorithms and try to decrease the time complexity when SIGQP is proved as NP-Complete.

Acknowledgments

This research was supported by the Ministry of Science and Technology in Taiwan under the grants MOST 107-2813-C-845-025-E and 109 new-type information security practice demonstration course program by Ministry of Education, Taiwan. Corresponding author: Yen Hung Chen, Email: yhchen@utaipei.edu.tw.

References

[1] P. E. Ross, Top 11 Technologies of the Decade, *IEEE Spectrum*, Vol. 48, No. 1, pp. 27-63, January, 2011.

[2] W. Liu, W. Sun, C. Chen, Y. Huang, Y. Jing, K. Chen, Circle of Friend Query in Geo-social Networks, *International Conference on Database Systems for Advanced Applications*, Busan, Korea, 2012, pp. 126-137.

[3] D. N. Yang, C. Y. Shen, W. C. Lee, M. S. Chen, On Socio-Spatial Group Query for Location-Based Social Networks, *18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, China, 2012,pp. 949-957.

[4] Z. Zhang, F. Shao, Reliability Analysis with Diameter Constraint in Social Networks, *Journal of Internet Technology*, Vol. 19, No. 7, pp. 2047-2055, December, 2018.

[5] Z. Zhang, Y. Liu, A List-wise Matrix Factorization Based POI Recommendation by Fusing Multi-Tag, Social and Geographical Influences, *Journal of Internet Technology*, Vol. 19, No. 1, pp. 127-136, January, 2018.

[6] Y. Li, D. Wu, J. Xu, B. Choi, W. Su, Spatial-Aware Interest Group Queries in Location-Based Social Networks, *Data & Knowledge Engineering*, Vol. 92, pp. 20-38, July, 2014.

[7] T. Lappas, K. Liu, E. Terzi, Finding a Team of Experts in Social Networks, *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, 2009, pp. 467-476.

[8] C. T. Li, M. K. Shan, Team Formation for Generalized Tasks in Expertise Social Networks, *2010 IEEE Second International Conference on Social Computing*, Minneapolis, MN, USA, 2010, pp.9-16.

[9] Y. Yao, J. Zhu, J. Liu, N. N. Xiong, Verifiable and Privacy-preserving Check-ins for Geo-social Networks, *Journal of Internet Technology*, Vol. 19, No. 4, pp. 969-980, July, 2018.

[10] G. R. Hjaltason, H. Samet, Distance Browsing in Spatial Databases, *ACM Transactions on Database Systems*, Vol. 24, No. 2, pp. 265-318, June, 1999.

[11] N. Katayama, S. Satoh, The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries, *ACM SIGMOD Record*, Vol. 26, No. 2, pp. 369-380, June, 1997.

[12] M. Kolahdouzan, C. Shahabi, Voronoi-Based k Nearest

Neighbor Search for Spatial Network Databases, *30th International Conference on Very Large Data Bases*, Vol. 30, Toronto, Ontario, Canada, 2004, pp. 840-851.

[13] D. Papadias, Q. Shen, Y. Tao, K. Mouratidis, Group Nearest Neighbor Queries, *20th International Conference on Data Engineering*, Boston, MA, USA, 2004, pp. 301-312.

[14] N. Roussopoulos, S. Kelley, F. Vincent, Nearest Neighbor Queries, *1995 ACM SIGMOD International Conference on Management of Data*, San Jose, California, USA, 1995, pp 71-79.

[15] B. U. Pagel, H. W. Six, H. Toben, P. Widmayer, Towards an Analysis of Range Query Performance in Spatial Data Structures, *12th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Washington D.C. USA, 1993, pp. 214-221. ACM.

[16] Y. Tao, X. Xiao, R. Cheng, Range Search on Multidimensional Uncertain Data, *ACM Transactions on Database Systems*, Vol. 32, No. 3, pp. 15, August, 2007.

[17] A. Corral, Y. Manolopoulos, Y. Theodoridis, M. Vassilakopoulos, Closest Pair Queries in Spatial Databases, *2000 ACM SIGMOD International Conference on Management of Data*, Dallas, Texas, USA, 2000, pp. 189-200.

[18] G. R. Hjaltason, H. Samet, Incremental Distance Join Algorithms for Spatial Databases, *ACM SIGMOD Record*, Vol. 27, No. 2, pp. 237-248, June, 1998.

[19] H. Shin, B. Moon, S. Lee, Adaptive Multi-stage Distance Join Processing, *ACM SIGMOD Record*, Vol. 29, No. 2, pp. 343-354, May, 2000.

[20] G. Cong, C. S. Jensen, D. Wu, Efficient Retrieval of the Top-k Most Relevant Spatial Web Objects, *VLDB Endowment*, Vol. 2, No. 1, pp. 337-348, August, 2009.

[21] I. De Felipe, V. Hristidis, N. Rishe, Keyword Search on Spatial Databases, *IEEE 24th International Conference on Data Engineering (ICDE)*, Cancun, Mexico, 2008, pp. 656-665.

[22] C. Long, R. C. W. Wong, K. Wang, A. W. C. Fu, Collective Spatial Keyword Queries: A Distance Owner-Driven Approach, *2013 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 2013, pp. 689-700.

[23] J. B. Rocha-Junior, O. Gkorgkas, S. Jonassen, K. Nørvåg, Efficient Processing of Top-k Spatial Keyword Queries, *International Symposium on Spatial and Temporal Databases*, Minneapolis, MN, USA, 2011, pp. 205-222.

[24] D. Wu, M. L. Yiu, C. S. Jensen, G. Cong, Efficient Continuously Moving Top-k Spatial Keyword Query Processing, *2011 IEEE 27th International Conference on Data Engineering*, Hannover, Germany, 2011, pp. 541-552.

[25] Y. Zhou, X. Xie, C. Wang, Y. Gong, W. Y. Ma, Hybrid Index Structures for Location-Based Web Search, *14th ACM international conference on Information and Knowledge Management*, Bremen, Germany, 2005, pp. 155-162.

[26] S. J. Chen, L. Lin, Modeling Team Member Characteristics for the Formation of a Multifunctional Team in Concurrent Engineering, *IEEE Transactions on Engineering Management*, Vol. 51, No. 2, pp. 111-124, May, 2004.

[27] N. Garg, G. Konjevod, R. Ravi, A Polylogarithmic Approximation Algorithm for the Group Steiner Tree Problem, *Journal of Algorithms*, Vol. 37, No. 1, pp. 66-84, October, 2000.

[28] L. Chen, G. Cong, C. S. Jensen, D. Wu, Spatial Keyword Query Processing: An Experimental Evaluation, *VLDB Endowment*, Vol. 6, No. 3, pp. 217-228, January, 2013.

[29] B. Martins, M. J. Silva, L. Andrade, Indexing and Ranking in Geo-ir Systems, *2005 Workshop on Geographic Information Retrieval*, Bremen, Germany, 2005, pp. 31-34.

[30] D. Wu, G. Cong, C. S. Jensen, A Framework for Efficient Spatial Web Object Retrieval, *The VLDB Journal*, Vol. 21, No. 6, pp. 797-822, December, 2012.

[31] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, San Francisco, 1979.

[32] D. S. Hochbaum, Approximating Covering and Packing Problems: Set Cover, Vertex cover, Independent Set, and Related Problems, in: D. S. Hochbaum (Ed.), *Approximation Algorithms for NP-Hard Problem*, PWS Publishing Company, 1996, pp. 94-143.

[33] G. L. Nemhauser, L. A. Wolsey, M. L. Fisher, An Analysis of Approximations for Maximizing Submodular Set Functions—I, *Mathematical Programming*, Vol. 14, No. 1, pp. 265-294, December, 1978.

[34] U. Feige, A Threshold of ln n for Approximating Set Cover, *Journal of the ACM*, Vol. 45, No. 4, pp. 634-652, July, 1998.

[35] F. V. Fomin, F. Grandoni, D. Kratsch, A Measure & Conquer Approach for the Analysis of Exact Algorithms, *Journal of the ACM*, Vol. 56, No. 5, pp. 25, August, 2009.

[36] M. Karpiński, W. Rytter, *Fast Parallel Algorithms for Graph Matching Problems*, Oxford Lecture Series in Mathematics and Its Applications Vol. 9, Oxford Science Publications, May, 1998.

Biographies

Chih-Yang Huang received the B.S. degree on Department of Computer Science, University of Taipei in 2019. He is studying for his master's degree on Department of Computer Science and Information Engineering, National Taiwan University. His research interests include Algorithms, Natural Language Processing, Computational Linguistics, Social Network Analysis and Digital Humanities.

Po-Chuan Chien received the B.S. degree on Department of Accounting and Department of Computer Science, National Chengchi University in 2015 and the M.S. degree on Department of Information Management, National Taiwan University in 2017. His research interests include financial text analysis and algorithm design.

Yen Hung Chen received his Ph.D. degree on Department of Computer Science from National Tsing Hua University in 2006. He is currently serving as an Associate Professor of Department of Computer Science from University of Taipei. His research interests include algorithms, computer theory, graph theory and operations research.