Protecting Privacy Using K-Anonymity with a Hybrid Search Scheme

Cui Run, Hyoung Joong Kim, Dal-Ho Lee, Cheong Ghil Kim, and Kuinam J. Kim

***Abstract*—In this paper, a new search algorithm to achieve k-anonymity for protecting privacy is introduced. For this purpose, two algorithms, Tabu Search and Genetic Algorithm, are combined. The simulation results show that the proposed algorithm is superior to the individual search algorithm in average.**

***Index Terms*—K-anonymity, database, privacy protection, heuristic algorithm.**

I. Introduction

In some case, organizations have to show micro-data to the public for special usages such as statistical analysis and health condition research. In order to protect individual privacy, known identifiers (e.g., name or social security number) must be removed. In addition, this process must consider the possibility of combining other attributes with external data to uniquely identify the individuals. Such kinds of attribute combination, called quasi-identifiers, can locate the individual using a unique mark for each individual. For example, an individual might be re-identified by joining the released data with another (public) database on age, sex, and salary. The *k*-anonymity model is one of the most popular ways to solve the privacy protection problem. It provides modification to the tuples in the database to remove the quasi-identifiers. It makes sure that each record is indistinguishable from at least $k-1$ other records. The idea is simple, but it is difficult to get *k*-anonymity property in the database.

Up to now, there have been many excellent researches on *k*-anonymity [1-3]. Most of them are based on the analysis of the data to achieve k-anonymity fast. But for the complexity of the data, there is no "common" model in most cases, which means that information loss is brought in with the pattern achieved higher after the modification applied to the database. To solve the problem above, some self-adoptive methods are invented. One of the efficient examples is heuristic algorithm [4]. As the *k*-anonymity is a NP hard problem, heuristic algorithm is quite applicable for such cases. Unfortunately, most of the heuristic methods such as Genetic Algorithm pay attention to the data modification in record level rather than full domain. This provides more chances for the analysts to get useful information from the modified records. In such kind of method, 2 similar records can have a high probability to be modified to different internals and such situation introduces higher distinguishable ability for the two processing records. Full domain consideration is important in k-anonymity problem.

The concept of lattice is introduced into the field in [5] in order to enhance the full domain modification property. Each node in a lattice in k-anonymity represents a way of modification. With a lattice, how to find a suitable solution has been changed into how to search in the lattice node space to get a suitable node. And in [6], the author provides an efficiency binary search algorithm (OLA) to find the optimal node in lattice space. But it focuses on lattice space with monotones property. If the nodes are not monotones, it can also give good solutions but no support in theory.

Based on the previous work mentioned above, we provide a new heuristic search method in lattice solution space; this approach is a combination of traditional Tabu Search method and Genetic Algorithm. It inherits the strong "climbing" ability of Tabu Search and the multiple start point property of Genetic Algorithm. We compare the performance of this new approach separately with the Tabu Search and Genetic Algorithm, our method performs better in most of cases.

In Section 2, we explain how to construct a lattice solution space for the database and introduce some necessary preparations such as information loss. In Section 3, the details of the hybrid algorithm are shown, and in Section 4, experiment results are given to compare the new method with other traditional heuristic algorithms such as genetic and Tabu method. Finally, the conclusion is in the last section.

II. Lattice Generation and Evaluation Matrix

A. Lattice Generation

Lattice in the k-anonymity problem is based on the concept of value generalization hierarchies; it is a tree structure and its internal nodes are intervals and its leaves are values appeared in the corresponding attributes. Each value generalization hierarchy is corresponding with one of the attributes in the database. The leaves are really values appeared in this column and the upper node is interval which can contain all the values or intervals in its sub-node. All the interval nodes located in the same height of the tree are disjoint.

Samarati and Sweeney [7], [8] have formulated mechanisms for k-anonymity property using the ideas of generalization and suppression. In their work, they showed

Manuscript received March 12, 2012; revised May 5, 2012.

Cui Run is with Korea University, Seoul, Korea (e-mail: tetons915@gmail.com).

Hyoung Joong Kim is with Gachon University, Sungnam, Kyeonggido, Korea

Cheong Ghil Kim is with Namseoul University, Cheonan, Choongnam, Korea

Kuinam J. Kim is with Kyonggi University, Suwon, Kyeonggido, Korea

the basic knowledge about the construction with a small scale example. But in heuristic method, we can process much more attributes at the same time by increasing the size of candidate solutions easily. The variables we use for the construction of lattice solution space are as follows:

TABLE 1: SOME DEFINITIONS IN THE PAPER

N: the amount of attributes we use in the database.
***A_i*:** the *i*th attribute in the database. $1 \le i \le N$.
H_i: the height of the value generalization hierarchy for A_i.
L: the lattice set for solution nodes.
T: temporary set for storing nodes.
Mark: variable to mark the levels of the nodes in lattice
NeighborSet: this set is defined as follows: for a node $[x_1,x_2,\ldots,x_n]$, we check the node $[x_1\text{-}1,x_2,\ldots,x_n]$, $[x_1,x_2\text{-}1,\ldots,x_n]$, $\ldots[x_1,x_2,\ldots,x_n\text{-}1]$. Among these candidate nodes, the one with negative values inside will be deleted. All the elements remained after checking will form the *NeighborSet*.

TABLE 2: ALGORITHM OF LATTICE-CONSTRUCTION

1. $L = \{[H_1, H_2, \ldots, H_N]\}$, *Mark* = 1, assign the *Mark* value to all the nodes in L, $T = \Phi$.
2. Find all the nodes with highest level value in L. For each node, calculate its *NeighborSet* and add the set to T.
3. If T is not all 0
   i. *Mark* = *Mark* + 1,
   ii. For each Node x in T, find its *NeighborSet*, add the set to L, assign the value of *Mark* to their level value.
   iii. Go to 2.
4. Else, add [0 0 0,…,0] to L
5. End algorithm.

Here we show how to get the lattice of solution space in k-anonymity problem.

After applying the algorithm above to the attributes, we can get a solution space in the form of lattice.

B. Evaluation Method

There are many different kinds of Information Loss Evaluation Method but until now there is no common standard rule for that. So users can choose any kind of traditional Information Loss Matrix or design their own method based on their desire and usages.

C. Discern-ability Metric

In this paper, firstly, to achieve a non-monotone property, we adopt the Discern-ability Metric (DM) as in Eq. 1, where f_i is the size of equivalence classes:

$$DM\_value = \sum_{f_i \ge k} (f_i)^2 + \sum_{f_i < k} (n \times f_i) \quad (1)$$

DM value means the distinguishability of the records. Bigger DM value represents that we can distinguish data more easily and less information loss occurs. In such a case, only few data modification is required.

D. Information Loss Metric

Another method we adopt in this paper is information loss metric, which focuses on the exactly information loss for the data. The Information Loss Matric (*ILM*) is consisted by two parts: interval information loss rate and generalization hierarchy loss rate.

Some definitions we use here are defined as follows

TABLE 3: SOME DEFINITIONS FOR ILM

DGH_i: domain generalization hierarchy for attribute *i*.
T: original dataset to be processed.
x : a tuple in the original database**.** ***x*(*i*)** : value for attribute i of *x*. ***M_i*(*x*)** : the middle value of the interval that ***x*(*i*)** located in the DGH_i.
MAX_i: the *maximal value in* attribute *i*.
MIN_i: the *minimal value in* attribute *i*.
***H_i*(*x*)**: for tuple *x*, the height of the value generalization hierarchy for A_i.
H_i: the height of the value generalization hierarchy for A_i.

So the *ILM* is defined as follows:

$$ILM = \sum_{x \in T} \sum_{i \in N} \left(\frac{M_i(x)}{MAX_i - MIN_i} + \frac{H_i(x)}{H_i} \right) \quad (2)$$

Here the higher ILM value represents higher information loss rate. This value can show us with a more precise information loss level.

With the two evaluation values mentioned above, we continue this work.

III. PROPOSED HYBRID METHOD

Now we will introduce our new search method in the lattice space. This method is a combination of Tabu Search method and Genetic Algorithm.

The traditional Tabu Search has a strong ability of "climbing". It focuses on the connection among the "neighbours". It can jump out of the local optimal solution and has a possibility to achieve best solution point. But this climbing ability is highly limited by the start point of Tabu Search. "Climbing" costs most of the time and the final results are always limited by the connection level between the solutions. For some problems of Tabu Search which get the start point with greedy method, it always shows that the start point is optimal. Such problems may be mainly caused by the long distance between optimal solution and the start point. As we use a good start point search strategy, it may

limit the chance to achieve the really optimal solution at the same time. For genetic method, it owns a good multiple start point properties; random strategy brings about such kind of advantages. With the large size of populations, it can also find a very good solution but with high level of randomness. If you are lucky enough, you will find a much better solution than you imagine. It cannot produce a stable best or nearly best solution for the problem processing.

In this new method, a Tabu Search is embedded into a traditional Genetic Algorithm to implement local search from multiple start points getting from genetic method. Here Tabu Search performs the role of mutation in Genetic Algorithm.

In Genetic Algorithm part, the setting for each part is described in sequence as follows:

- ***Initial solution***: an initial population will be randomly generated which performs as multiple start points of the whole algorithm within lattice space. A uniform distribution is used in this random procedure.
- ***Stop condition***: a value of 100 is appointed as the limitation of the cycle time. If it arrives at the limitation, the algorithm will stop and output the best solution node that can be achieved.
- ***Fitness value***: each node in lattice space corresponds with a strategy of database modification. In this paper, *DM* value becomes larger with less information loss; and *ILM* value become smaller with less information loss. We can use the inverse of DM value multiply with *ILM* value to evaluate the information loss, which also performs the role of fitness value. The fitness is defined as follows:

$$fitness\_value = \frac{ILM}{DM\_Value} \tag{3}$$

- ***Roulette wheel selection***: this part is a traditional strategy to choose the candidates with the fitness value. Assuming each node, x, f_x is the corresponding fitness value and *IPS* is initial population set. Then in the selection wheel, it will own a chosen probability P_x defined as:

$$P_x = f_x \Big/ \sum_{i \in IPS} f_i \tag{4}$$

  Then we can choose the number of candidates with random chosen according to the possibility as above.
- ***Crossover***: in this part, we will randomly choose pairs from the candidates. Cut the pairs from a random point between two attributes and exchange the second halves of the solutions to get a new pair of solutions. The procedure will be repeated until we get enough new candidates.
- ***Update population***: after the Tabu part, which performs the role of Mutations in the Genetic Algorithm, we will get new group of solutions. The original population will be replaced by a new group of solution in this step.

The procedures in the Tabu Search part are listed as follows:

- ***Initial point***: in this step, for each node passed by crossover part, Tabu Search will deal with it as a start point and search the local area around it.
- ***Generation limitation check***: the circle time will be check in this procedure. If the limitation is arrived, the Tabu part will be ended.
- ***Neighbor Search***: the concept of neighbor is defined as follows: For any pair of nodes *x* and *y*, we subtract the corresponding attributes value from *x* to *y*. If the results are all 0s but only 1 non-zero integer, we say *x* and *y* are neighbors.

  For example, *x* = [1 3 3], y = [1 3 4], *x* – *y* = [0 0 -1], then *x* and *y* are neighbors; if *x* = [2 3 3], y = [1 3 2], *x* – *y* = [1 0 1], then *x* and *y* are not neighbors.

  In this step, we will produce all the possible neighbors for the nodes we are checking and calculate their fitness values at the same time. The union of neighbor sets from different attributes will be sent to the next step to process.
- ***Candidates Chosen***: this procedure will choose the candidates for the circle of Tabu Search. There are two kinds of nodes: k-anonymity node and non-k-anonymity node. All the k-anonymity will share a possibility 0.7 and all non-k-anonymity nodes will share the left 0.3. But among each subclass of nodes, the candidates are equally chosen. In this step we always remember the best node solution for *k*-anonymity. Candidates in the Tabu list will not be considered until they get out of the Tabu list.
- ***Tabu list Update***: after the candidate chose, the Tabu list will be updated. The new candidates will enter into Tabu list. Some nodes in the Tabu list will be removed if their living time is arrived.
- ***Output***: the best solution for Tabu Search will be output.

IV. Simulation Results

This section evaluates the performance of our new search method for different *k* values (3 to 15). At the same time we also compute the performance of traditional Genetic Algorithm and Tabu Search method separately. The test database is Pima Indians Diabetes Database which is consisted by 768 records, 9 attibutes.

In genetic part of our method, we set the population size as 20 and the circle time as 20. In Tabu part of our algorithm, we set the number of candidates is 20; size of Tabu list 6; living time 7; circle time 20. In traditional Tabu Search and Genetic Algorithm, we adopt same setting except the circle time is 300.

The algorithm is repeated for 50 times and the average results are shown in Fig. 1 and Fig. 2.

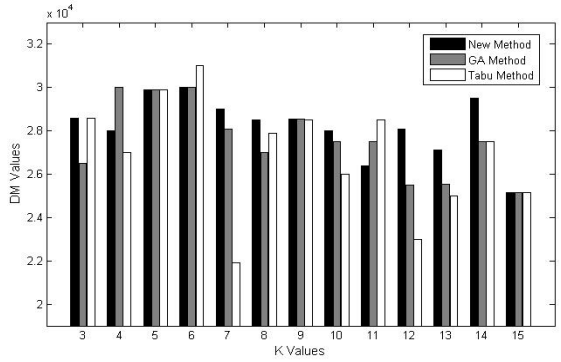

Fig. 1. Average performance results of DM values for Pima Indians Diabetes Database

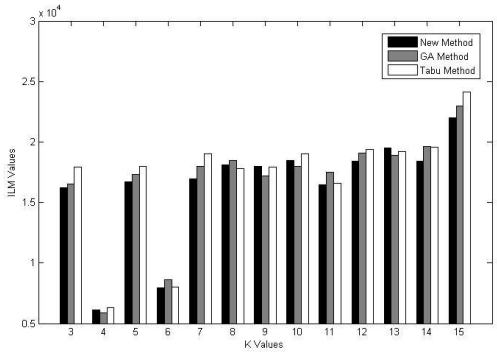

Fig. 2. Average performance results of ILM values for Pima Indians Diabetes Database

From the analysis of experiment results, we can see that in most the case, our new approach is better than single Tabu Search or Genetic Algorithm. It has a stable performance than the two methods. For small values of k, the differences among the three methods are small, but as the increase of *k* the differences increase at the beginning and decrease later.

V. Conclusion

This paper presents a hybrid search method composed of Tabu Search and Genetic Algorithm. The experimental results show that the proposed heuristic approach is a good method to search for the solutions in lattice space. Even though the method is simple, it achieves better perform than the other two traditional heuristic method.

There are still many things to do in the future work. We can try to find some efficient aspiration criterion in the Tabu part. Also in the crossover part, we can replace it with many other kinds of method such as two point crossover.DM values is not the only way to evaluate the effort of the algorithm, we can design other kinds of special evaluating matrix to satisfy specific command.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (KRF 2011-0027264).

Reference

[1] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *Proceedings of the 22nd International Conference on Data Engineering, IEEE*, pp. 25-25, 2006.

[2] Z. H. Wang, W. Wang, B. Shi, "Clustering-Based Approach for Data Anonymization," *Journal of Software*, pp. 680-693, 2010.

[3] H. Park and K. Shim, "Approximation algorithms fork-anonymity," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, ACM, pp. 67-68, 2005.

[4] R. Chaytor, "A Better Problem Representation for k –Anonymity," in *Proc. 1st ACM SIGKDD Int'l Work.on Privacy, Security, and Trust in KDD*, ACM, pp 52-61, 2007.

[5] R. J. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in *ICDE '05 Proceedings of the 21st International Conference on Data Engineering*, IEEE Computer Society, pp 217-228, 2005.

[6] K. El Eman, F. K. Dankar, R. Issa *et al.*,"A Globally Optimal k-anonymity Method for the De-Identification of Health Data," *Journal of the American Medical Informatics Association : JAMIA*, pp. 670-682, 2009.

[7] P. Samarati. "Protecting respondents' identities in micro data release," in *IEEE Transactions on Knowledge and Data Engineering, IEEE Educational Activities Department Piscataway*, pp. 1010-1027, 2001.

[8] L. Sweeney, "K-anonymity: a model for protecting privacy," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, Citeseer, pp. 557–570, 2002.

[9] A. Friedman, "Providing k-Anonymity in Data Mining," *The International Journal on Very Large Data Bases*, Springer, pp. 789-804, 2008.