Using Multi-objective Optimization to Analyze Data Utility and Privacy Tradeoffs in Anonymization Techniques

Rinku Dewri, Indrakshi Ray[*], Indrajit Ray, and Darrell Whitley

Department of Computer Science

Colorado State University, Fort Collins, CO 80523, USA

Abstract

Data anonymization techniques have received extensive attention in the privacy research community over the past several years. Various models of privacy preservation have been proposed: k–anonymity, ℓ–diversity and t–closeness, to name a few. A typical drawback of these models is that there is considerable loss in data utility arising from the use of generalization and suppression techniques. Optimization attempts in this context have so far focused on maximizing the data utility for a pre-specified level of privacy. To determine if better privacy levels are obtainable with the same level of data utility, most of the existing formulations require exhaustive analysis. Further, the data publisher's perspective is often missed in the process. The publisher wishes to maintain a given level of data utility and then maximize the level of privacy within acceptable limits. In this paper, we explore multi-objective formulations to provide substantial information to a data publisher about the trade-offs available between the privacy level and the information content of an anonymized data set. Results from our analysis on a benchmark data set indicate that often typical characteristics in the solution set can provide beneficial information to a data publisher about the interaction between privacy and utility generated by the usage of a particular anonymization technique.

Keywords: Privacy, Data utility, Multi-objective optimization.

[*] Corresponding Author: 211 University Services Center, Fort Collins, CO 80523-1873, USA. Tel. No. +1(970)491-7986 Fax No. +1(970)491-2466 Email: iray@cs.colostate.edu

1 Introduction

Various scientific studies, business processes and legal procedures depend on quality data from large data sources. However, such data sources often contain sensitive personal data, dissemination of which is governed by various privacy requirements. Data in such cases need to be sanitized of personally identifying attributes before it can be shared. Anonymizing data is challenging because re-identifying the values in sanitized attributes is not impossible when other publicly available information or an adversary's background knowledge can be linked with the shared data.

Database researchers have worked hard over the past several years to address such privacy concerns. Sweeney proposed the concept of k–anonymity that reduces the chances of a *linking attack* being successful [9]. Anonymization of data sets involve transforming the actual data set into a form unrecognizable in terms of the exact data values, by using data generalization and suppression techniques. Generalization of data is performed by grouping together data attribute values into a more general one. An example of this is replacing a specific age by an age range. Data suppression on the other hand removes entire tuples making them no longer existent in the data set. A transformed data set of this nature is said to be k–anonymous if each record in it is same as at least $k-1$ other records. The higher the value of k, the stronger the privacy that the model offers.

An unavoidable consequence of performing such anonymization is a loss in the quality of the data set. Researchers have therefore looked at different methods to obtain an optimal anonymization that results in a minimal loss of information [13, 1, 14, 12, 8, 15]. However, the privacy level reported by such algorithms may not be sufficient from the perspective of a human subject whose privacy is under concern. This may require a higher k value with a pre-specified acceptable level of information loss. Optimization attempts of the above nature are not intended to solve such a problem.

Moreover, as research in this front progressed, other types of attacks have been identified such as *homogeneity attack, background knowledge attack, skewness attack* and *similarity attack* [11, 3]. Models beyond k–anonymity have been proposed to counter these new forms of attacks on anonymized data sets and the hidden sensitive attributes. Two of the more well known models in this class are the ℓ–diversity model [3] and the t–closeness model [11]. While these models enable one to better guarantee the preservation of privacy in the disseminated data, they still come at a cost of reduced quality of the information.

The (possibly) unavoidable loss in data quality due to anonymizing techniques presents a dilemma to the data publisher. Consider the case of a data source such as Microsoft HealthVault^{TM}. It is committed to the protection of the privacy of individuals who have trusted the organization with their health record. At the same time, Microsoft also aims to earn revenue by sharing this data appropriately. Thus, although the organization needs to use some anonymization technique when disseminating the data, it needs to maintain some pre-determined level of utility in the published data so as to earn revenue from it. In such a situation, it is imperative that the data publisher understands the implications of setting a parameter in a privacy model (for example, k in k–anonymity or ℓ in ℓ–diversity) to a particular value. There is clearly a trade-off involved. Setting the parameter to a very low value impacts the privacy of individuals in the database. Picking a "very high" value disrupts the inference of any significant statistical information from the anonymized data set.

We believe that in order to understand the impact of setting the relevant parameters, a data publisher needs to answer questions similar to the following.

1. What level of privacy can one assure given that one may not suppress any record in the data set and can only tolerate an information loss of 25% (say)?

2. What is a good value for k (assuming the k–anonymity model) when one may suppress 10% (say) and be able to tolerate an information loss of (maybe) 25%?

3. Under the "linking attacks" threat model and assuming that the users of the published data sets are likely to have background knowledge about some of the individuals represented in the data set, is it possible to combine the k–anonymity and the ℓ–diversity models to obtain a generalization that protects against the privacy problems one is worried about?

4. Is there a generalization that gives a high k and a high ℓ value if one is ready to suppress (maybe) 10% of the records and tolerate (say) 20% of information loss?

Unfortunately, answering these questions using existing techniques will require us to try out different k (or ℓ) values to determine what is suitable. Further, such a methodology does not guarantee that better privacy results cannot be obtained without incurring any or an acceptable increase in the information loss. Although recent studies have looked into the development of fast algorithms to minimize the information loss for a particular anonymization technique with a given value for the corresponding parameter, we are not aware of any study that explores the data publisher's dilemma. We believe that the problem of privacy needs to be addressed from the opposite direction as well – given an acceptable level of information loss determine a k and ℓ value that satisfies the privacy requirements of the data set.

Our approach is primarily based on the formulation of a series of multi-objective optimization problems, the solutions to which provide a preliminary understanding of the trade-off present between the level of privacy (k or ℓ value) and the quality of the anonymized data set. We provide an analytical discussion on the formulated problems to show how information on this trade-off behavior can be utilized to adequately answer the data publisher's questions. We employ a popular evolutionary algorithm to solve the multi-objective optimization problems relevant to this study.

The remainder of the paper is organized as follows: Section 2 reviews some of the existing optimization algorithms for k–anonymization. The required background on multi-objective optimization is presented in Section 3. We introduce the terminology used in the paper in Section 4. Section 5 provides a description of the four multi-objective problems we formulate and the underlying motivation behind them. The specifics of the solution methodology as particular to solving the problems using the evolutionary algorithm is given in Section 6, and a discussion of the results so obtained is presented in Section 7. Finally, Section 8 summarizes and concludes the paper.

2 Related Work

Several algorithms have been proposed to find effective k–anonymization. The μ-argus algorithm is based on the greedy generalization of infrequently occurring combinations of *quasi-identifiers* and suppresses outliers to meet the k–anonymity requirement [1]. μ-argus suffers from the shortcoming that larger combinations of quasi-identifiers are not checked for k–anonymity and hence the property is not always guaranteed [8]. Modifying the algorithm to include such combinations usually makes the algorithm computationally inefficient.

Sweeney's Datafly approach uses a heuristic method to generalize the attribute containing the most distinct sequence of values for a provided subset of quasi-identifiers [8]. Sequences occurring less than k times are suppressed. Although the algorithm is efficient, the solutions obtained from the approach may be suboptimal. In the same work, Sweeney proposes a theoretical algorithm that can exhaustively search all potential generalizations to find the one that minimally distorts the data during anonymization.

Samarati's algorithm [12] can identify all k–minimal generalizations, out of which an optimal generalization can be chosen based on certain preference information provided by the data recipient.

Iyengar proposes a flexible generalization scheme and uses a genetic algorithm to perform k–anonymization on the larger search space that resulted from it [14]. Although the method can maintain a good solution quality, it has been criticized for being a slow iterative process. In this context, Lunacek et al. introduce a new crossover operator that can be used with a genetic algorithm for constrained attribute generalization, and effectively show that Iyengar's approach can be made faster [10]. As another stochastic approach, Winkler propose using simulated annealing to do the optimization [15].

In order to obtain a guaranteed optimal solution, Bayardo and Agrawal propose a complete search method that iteratively constructs less generalized solutions starting from a completely generalized data set [13]. The method employs a dynamic search strategy to prune away subspaces of possible generalizations and avoid the enumeration of all possible generalizations.

The drawbacks of using k–anonymity are first described by Machanavajjhala et al. [3]. They identify that k–anonymized data sets are susceptible to privacy violations when there is little diversity in the sensitive attributes of a k–anonymous equivalence class. Further, in the presence of background knowledge, an adversary can conclude with near certainty what value a sensitive attribute holds for a particular individual. In order to alleviate such privacy breaches, they propose the model of ℓ–diversity which obtains anonymizations with an emphasis on the diversity of values on a k–anonymous equivalence class. They also show that current k–anonymization algorithms can be adapted to ℓ–diversity with minor modifications. Note that obtaining an anonymization with a particular value of ℓ may not be possible for a given value of k.

Further work presented by Li et al. show that the ℓ–diversity model is also susceptible to certain types of attacks [11]. To this effect, they emphasize having the t–closeness property that maintains the same distribution of sensitive attribute values in an equivalence class as is present in the entire data set, with a tolerance level of t. They realize that t–closeness does not deal with identity disclosure scenarios and propose that it should be used in conjunction with k–anonymity. The conjunctive use of different privacy models is promising in their approach, although it is not clear whether the loss in data utility will be acceptable or not to a data publisher when using a specified value of k and t.

The potential problem in using the above approaches is that they are targeted towards obtaining an optimal generalization for a fixed value of k, ℓ, or t, sometimes in conjunction. Besides running the algorithms multiple times with different values of this parameter, no attempt is known to have been made to understand how the generalizations and the related cost metrics change with changes in the parameter values. Our work seeks to fill this gap.

3 Multi-objective Optimization

In real world scenarios, often a problem is formulated to cater to several criteria or design objectives, and a decision choice to optimize these objectives is sought for. An optimum design problem must then be solved with multiple objectives and constraints taken into consideration. This type of decision making problems falls under the broad category of multi-criteria, multi-objective, or vector optimization problem.

Multi-objective optimization differs from single-objective ones in the cardinality of the optimal set of solutions. Single-objective optimization techniques are aimed towards finding the global optima. In case of multi-objective optimization, there is no such concept of a single optimum solution. This is due to the fact that a solution that optimizes one of the objectives may not have the desired effect on the others. As a result, it is not always possible to determine an optimum that corresponds in the same way to all the objectives under consideration. Decision making under such situations thus require some domain expertise to choose from multiple trade-off solutions depending on the feasibility of implementation.

Formally, we can state the multi-objective optimization problem (MOOP) as follows:

General MOOP: *Find the vector* $\overrightarrow{x^*} = [x_1^*, x_2^*, \ldots, x_n^*]^T$ *which optimizes the M-dimensional vector function*

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_M(\vec{x})]^T$$

satisfying p inequality and q equality constraints

$$\begin{aligned} g_i(\vec{x}) &\geq 0 \qquad i = 1, \ldots, p \\ h_i(\vec{x}) &= 0 \qquad i = 1, \ldots, q \end{aligned}$$

where $\overrightarrow{x} = [x_1, x_2, \ldots, x_n]^T$ *is the vector of decision variables and M is the number of objectives in the problem.*

Due to the conflicting nature of the objective functions, a simple objective value comparison cannot be performed to compare two feasible solutions to this problem. Most multi-objective algorithms thus use the concept of dominance to compare feasible solutions.

Dominance and Pareto-optimal set: *In a minimization problem with M objectives, a feasible solution vector $\vec{x}$ is said to dominate another feasible solution vector $\vec{y}$ if*

1. $\forall i \in \{1, 2, \ldots, M\} \qquad f_i(\vec{x}) \leq f_i(\vec{y})$ *and*

2. $\exists j \in \{1, 2, \ldots, M\} \qquad f_j(\vec{x}) < f_j(\vec{y})$

$\vec{y}$ is then said to be dominated by $\vec{x}$, denoted by $x \preceq y$. If the two conditions do not hold, $\vec{x}$ and $\vec{y}$ are said to be non-dominated w.r.t. each other, denoted by the $\npreceq$ symbol. Further, the set of all non-dominated solutions obtained over the entire feasible region constitutes the Pareto-optimal set.

In other words, a Pareto-optimal solution is as good as other solutions in the Pareto-optimal set, and better than other feasible solutions outside the set. The surface generated by these solutions in the objective space is called the Pareto-front or Pareto-surface. Fig. 1 shows the Pareto-front for a hypothetical two-objective problem, with the dominance relationships between three feasible solutions.

In the context of the k–anonymity problem, the Pareto-front for the two objectives – maximize k and minimize loss – provides the decision maker an understanding of the changes in the information loss when k is varied. Consider two solutions s_1 and s_2 with corresponding k and loss as $(k_1, loss_1)$ and $(k_2, loss_2)$ respectively. Let us assume that $k_1 < k_2$ and $loss_1 = loss_2$. A decision maker using s_1, and unaware of s_2, misses on the fact that a higher k value is possible without incurring any increase in the loss. A multi-objective algorithm using the dominance concept can expose this relationship between s_1 and s_2, namely $s_2 \preceq s_1$. As another example, consider the case with $loss_2 - loss_1 = \epsilon > 0$. s_1 and s_2 are then non-dominated solutions, meaning that one objective cannot be improved without degrading the other. However, if ϵ is a relatively small quantity acceptable to the decision maker, s_2 might be preferable over s_1. Such trade-off characteristics are not visible to the decision maker until a multi-objective analysis is carried out.

The classical way to solve a multi-objective optimization problem is to follow the preference-based approach. A relative weight vector for the objectives can help reduce the problem to a single-objective instance, or impose orderings over the preference given to different objectives. However, such methods fail to provide a global picture of the choices available to the decision maker. In fact, the decision of preference has to be made before starting the optimization process. Relatively newer methods have been proposed to make the decision process more interactive.

Evolutionary algorithms for multi-objective optimization (EMO) have been extensively studied and applied to a wide spectrum of real-world problems. One of the major advantages of using evolutionary algorithms is their ability to scan through the global search space simultaneously, instead of restricting to localized regions of gradient shifts. An EMO works with a population of trial solutions trying to converge on to the Pareto-optimal set by filtering out the infeasible or dominated ones. Having multiple solutions from a single run of an EMO is not only an efficient approach but also helps a decision maker obtain an intuitive understanding of the different trade-off options available at hand. The effectiveness of an EMO is thus characterized by its ability to converge to the true Pareto-front and maintain a good distribution of solutions on the front.

A number of algorithms have been proposed in this context [4, 7]. We employ the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [6] for the multi-objective optimization in this study. NSGA-II has gained a wide popularity in the multi-objective optimization community, partly because of its efficiency in terms of the convergence and diversity of solutions obtained, and partly due to its extensive application to solve real-world problems.

4 Preliminaries

A data set D can be visualized as a tabular representation of a multi-set of tuples $r_1, r_2, \ldots, r_{n_{row}}$ where n_{row} is the number of rows in the table. Each tuple (row) r_i comprises of n_{col} values $\langle c_1, c_2, \ldots, c_{n_{col}} \rangle$ where n_{col} is the number of columns in the table. The values in column j correspond to an *attribute* a_j, the domain of which is represented by the ordered set $\Sigma_j = \{\sigma_1, \sigma_2, \ldots, \sigma_{n_j}\}$. The ordering of elements in the set can be implicit by nature of the data. For example, if the attribute is "age", the ordering can be done in increasing order of the values. For categorical data, obtaining an ordering requires the user to explicitly specify a hierarchy on the values. A hierarchy can be imposed based on how the values for the attribute can be grouped together. Fig. 2 shows an example hierarchy tree for the attribute "marital status". The leaf nodes in this example constitute the actual values that the attribute can take. The ordering for these values can be assigned based on the order in which the leaf nodes are reached in a preorder traversal of the hierarchy tree [14].

A *generalization* G_j for an attribute a_j is a partitioning of the set Σ_j into ordered subsets $\langle \Sigma_{j_1}, \Sigma_{j_2}, \ldots, \Sigma_{j_K} \rangle$ which preserves the ordering in Σ_j, i.e. if σ_a appears before σ_b in Σ_j then, for $\sigma_a \in \Sigma_{j_l}$ and $\sigma_b \in \Sigma_{j_m}$, $l \leq m$. Further, every element in Σ_j must appear in exactly one subset. The elements in the subsets maintain the same ordering as in Σ_j. For the age attribute having values in the range of $[10, 90]$, a possible generalization can be $\langle [10, 30], (30, 50], (50, 70], (70,\ 90] \rangle$. A possible generalization for the marital status attribute can be $\langle$*Not Married, civ-spouse, AF-spouse, spouse-absent*$\rangle$. It is important to note that generalizations for categorical data is dependent on how the hierarchy is specified for it. Further, generalizations are restricted to only those which respect the hierarchy. The generalization is said to be *constrained* in such a case. For example, the generalization $\langle\{$*Never Married, Divorced*$\}$, $\{$*Widowed, Separated*$\}$, *Married*$\}\rangle$ is not valid for marital status since the hierarchy tree specifies that the values $\{$*Divorced, Widowed, Separated*$\}$ can only be generalized as *Once-Married*, if at all.

Given the generalizations $G_1, G_2, \ldots, G_{n_{col}}$, the data set D can be transformed to the *anonymized* data set D' by replacing each value v at row i and column j in D by $G_j(v)$ where $G_j(v)$ gives the index of the subset to which v belongs to in the generalization G_j. Note that if a particular generalization G_j is equal to the domain of values Σ_j, all values of the corresponding attribute will be transformed to the same subset index 1, in which case all information in that attribute is lost and the *cell is suppressed.*

4.1 k–Anonymity

Equivalent tuples in D' (subset indices are equal in every column) can be grouped together into equivalence classes. In other words, an equivalence class groups all tuples in D that got transformed to the same tuple in D' under some generalization. The *k–anonymity* problem is then defined as follows.

k–Anonymity problem: *Given a data set D, find a set of generalizations for the attributes in D such that the equivalence classes induced by anonymizing D using the generalizations are all of size at least k.*

The problem can also be explained as obtaining the generalizations under which every tuple in D' is same as at least $k - 1$ other tuples. Thus, a higher value of k evaluates to a lower chance of privacy breach.

4.2 ℓ–Diversity

The set of attributes can be divided into *sensitive* and *non-sensitive* classes. A sensitive attribute is one whose value must not be revealed (or get revealed) for any tuple in the data set. All other attributes are considered non-sensitive.

Let a_s be a sensitive attribute in a data set with the domain of values $\Sigma_s = \{\sigma_1, \sigma_2, \ldots, \sigma_{n_s}\}$. Further, let $Q_1, \ldots, Q_p$ be the equivalence classes induced by a generalization. If $c(\sigma)_j$, where $\sigma \in \Sigma_s$, denotes the count of the number of tuples with the sensitive attribute value σ in Q_j, then the *ℓ–diversity* problem can be stated as follows.

ℓ–Diversity problem: *Given a data set D, find a set of generalizations for the attributes in D such that for each equivalence class induced by anonymizing D using the generalizations, the relation*

$$\frac{c(\sigma)_j}{|Q_j|} \leq \frac{1}{\ell} \tag{1}$$

holds for all $\sigma \in \Sigma_s$ and $j = 1, \ldots, p$.

In other words, the ℓ–diversity property guarantees that a sensitive attribute value cannot be associated with a particular tuple with a probability more than $1/\ell$. The higher the value of ℓ, the better is the privacy.

The trivial generalization $G_j = \langle \Sigma_j \rangle$, where $j = 1, \ldots, n_{col}$, can provide the highest k and the highest ℓ value. However, such a generalization results in an anonymized data set with little or no statistically significant information. It is often desired that the anonymized data set be still applicable to some level of statistical analysis. In such a case, the decision on the value of k (or ℓ) is subjected to a loss measurement of the information content in the anonymized data set, usually done using some metric. An optimization problem defined for a given value of k (or ℓ) tries to find the generalizations that result in a minimal loss given by the metric.

Depending on the distribution of data values in a data set, obtaining a generalization with an acceptable loss for a given value of k (or ℓ) may or may not be possible. This happens when the data set has outliers that cannot be anonymized without overly generalizing the remaining data points. It therefore becomes a requirement that such outliers be *suppressed* completely in order to avoid an over generalization. A suppressed tuple is usually considered nonexistent in the data set. The loss metric can account for this suppression in its loss measurement.

When suppression is allowed, an anonymized data set can be made k–anonymous by suppressing all tuples that belong to equivalence classes of size less than k. Similar suppression methods can be used to enforce the ℓ–diversity property. The case without suppression can be modeled into the earlier scenario (with suppression) by assigning an infinite loss when suppression is performed [13]. However, it should be noted that the presence of outliers will always force the requirement for suppression, in which case the loss measurement will always become infinite. Furthermore, even though suppression is not allowed, such an approach enforces the k–anonymity (or ℓ–diversity) property by suppressing outliers. If all the data points in the data set has to stay in the anonymized data set as well, the desired privacy properties cannot be ascertained even after adopting such modeling.

Furthermore, we define a (k, ℓ)–*safe* data set to incorporate the benefits of k–anonymity and ℓ–diversity in an anonymization.

(k, ℓ)–**Safe:** *An anonymized data set is* (k, ℓ)*–safe if it is* k*–anonymous and* ℓ*–diverse.*

From a data publisher's perspective, a (k, ℓ)–safe anonymization is more meaningful since it allows one to ascertain the presence of the privacy guards obtainable from both k–anonymity and ℓ–diversity. A high k value in this case prohibits the likelihood of linking attacks, while a high ℓ value prohibits the likelihood of homogeneity and background knowledge attacks. When multiple types of attacks are possible on a data set, a data publisher would most certainly want to safeguard the published data against as many of them as possible. However, the question of information loss still remains, which then has to be optimized to be minimal.

5 Problem Formulation

As stated in the previous section, an optimization algorithm requires a numeric representation of the information loss associated with a particular generalization. A quantified loss value enables the optimization algorithm to compare two generalizations for their relative effectiveness. Loss (cost) metrics assign some notion of penalty to each tuple whose data values get generalized or suppressed, thereby reflecting the total information lost in the anonymization process. In this paper, we use the *general loss metric* proposed by Iyengar [14]. The general loss metric computes a normalized information loss for each of the data values in an anonymized data set. The assumption here is that information in every column is potentially important and hence a flexible scheme to compute the loss for both numeric and categorical data is required.

5.1 Generalization loss

Consider the data value $v_{i,j}$ at row i and column j in the data set D. The general loss metric assigns a penalty to this data value based on the extent to which it gets generalized during anonymization. Let $g_{i,j} = G_j(v_{i,j})$ be the index of the subset to which $v_{i,j}$ belongs to in the generalization G_j, i.e. $v_{i,j} \in \Sigma_{j_{g_{i,j}}}$. The penalty for information loss associated with $v_{i,j}$ is then given as follows:

$$loss(v_{i,j}) = \frac{|\Sigma_{j_{g_{i,j}}}| - 1}{|\Sigma_j| - 1} \tag{2}$$

For categorical data, the loss for a cell is proportional to the number of leaf nodes rooted at an internal node (the generalized node) of the hierarchy tree. The loss attains a maximum value of one when the cell is suppressed ($G_j = \langle \Sigma_j \rangle$), or in other words, when the root of the tree is the generalized node. Subtracting one ensures that a non-generalized value incurs zero loss since the cardinality of the subset to which it belongs would be one. The generalization loss is then obtained as the total loss over all the data values in the data set.

$$GL = \sum_{i=1}^{n_{row}} \sum_{j=1}^{n_{col}} loss(v_{i,j}) \tag{3}$$

5.2 Suppression loss

Although the loss due to suppression can be incorporated into the generalization loss, we decided to separate it out for the purpose of our study. When a row is suppressed, all cells in the row are suppressed irrespective of the generalization. Each cell thereby incurs a loss of one. Let n_{sup} be the number of rows to be suppressed in the data set. The suppression loss for the data set is then given as,

$$SL = n_{col} \times n_{sup} \tag{4}$$

5.3 The multi-objective problems

The multi-objective problems we formulate in this paper are intended to analyze and understand the trade-off nature of the generalization and suppression loss when k (or ℓ) is varied. A single objective optimization problem to minimize the generalization loss with a fixed k (or ℓ) will require multiple runs of the algorithm to understand this trade-off. By adopting a multi-objective approach, we can generate a fairly good approximation of the Pareto-front in a single run of the algorithm, which in turn provides us with the requisite information to make a better decision on the choice of k (or ℓ) for anonymization. In this context, we formulate a series of multi-objective problems for our analysis. Although, Problems 1, 2, and 3 are described for the k–anonymity problem, similar analysis can be carried out for the ℓ–diversity problem as well. Problem 4 caters to the (k, ℓ)–safe problem.

The problems under study are not intended to provide the data publisher a "best" value for the parameter(s) involved in the anonymization technique. Rather, we put forward a methodology to understand the implications of choosing a particular value for the parameter(s) in terms of the resulting privacy and the data utility. Hence, we shall often find that one or more solutions (equivalence classes) returned by the optimization process are trivially not acceptable either in terms of privacy or utility, or in some cases, both. It is not our objective to consider such solutions as degenerate and prohibit them from appearing in the solution set. After all, they are also a manifestation of the privacy-utility trade-off, which would likely be never selected as a choice by the data publisher but still possible. For e.g., an extreme solution will correspond to a situation where every tuple in the data set belongs to its own equivalence class, thereby resulting in no privacy and maximum utility. Another extremity is the case where all tuples are grouped together in a single equivalence class resulting in maximum privacy but no utility. One cannot deny the fact that in the case of privacy versus utility, both of these are possible solutions. The multi-objective optimization do not incorporate the required domain knowledge to identify these extremities (or other such solutions) as non-practical. Only the data publisher has the requisite knowledge to make such identification and disregard such solutions. This is often a post-optimization process. Hence, focus in the solution set should be concentrated on the practical solutions reported by the method. Quite often there will be more than one, and the methodology provides the data publisher a distinctive picture of the differences arising between privacy and utility when it makes a decision to choose one solution over another.

5.3.1 Problem 1

The presence of outliers in a data set makes it difficult to find a suitable value of k when suppression of data is not allowed. In this formulation, we strictly adhere to the requirement that no tuple in the data set can be deleted. Intuitively, such a strict requirement makes the k–anonymity problem insensible to solve for a given k, as the optimization algorithm will be forced to overly generalize the data set in its effort to ensure

k–anonymity. The outliers usually belong to very small equivalence classes and the only way to merge them into a bigger one is by having more generalization. This results in more information loss which is often not acceptable to an user.

Although solving the k–anonymity problem is not possible in terms of its strict definition, it is worth noting that a generalization can still affect the distribution of the equivalence classes even when suppression is not allowed. An equivalence class E_k in this description groups all tuples that are similar to exactly $k-1$ other tuples in the anonymized data set. An ideal generalization would then maintain an acceptable level of loss by keeping the number of rows in smaller equivalence classes (small k) relatively lower than in the bigger equivalence classes. Although this does not guarantee complete k–anonymity, the issue of privacy breach can be solved to a limited extent by reducing the probability that a randomly chosen row would belong to a small equivalence class.

With this motivation, we define the *weighted-k–anonymity* multi-objective problem to find generalizations that produce a high weighted-k value and low generalization loss. Each equivalence class E_k defines a k value, $k \leq n_{row}$, for its member tuples – every tuple in the equivalence class is same as exactly $k-1$ other tuples in the same class.

Note that this notion of an equivalence class is different from the one stated in the k–anonymity problem. Two rows in the original data set belong to the same equivalence class in the k–anonymity definition if the generalization transforms them into the same tuple. However, in this formulation, two rows belong to the same equivalence class E_i if a generalization makes them i–anonymous.

The weighted-k for a particular generalization inducing the equivalence classes $E_1, E_2, \ldots, E_{n_{row}}$ on the anonymized data set is then obtained as follows:

$$k_{weighted} = \frac{\sum_{i=1}^{n_{row}} (i \cdot |E_i|)}{\sum_{i=1}^{n_{row}} |E_i|} \tag{5}$$

Recall the concept of local recoding [2] in this context. A local recoding scheme produces a k–anonymization by using an individual generalization function (instead of a global one) for each tuple in the data set. This is a more powerful scheme compared to having a single generalization function since outliers can be easily suppressed without the drawbacks of an over generalization, hence data utility can be maintained. The weighted-k–anonymity based generalization is orthogonal to this concept in certain ways. Local recoding explores the domain of generalization functions and uses multiple points in this domain to recode different subsets of the data set differently. This puts outliers in their own subset(s), thereby making it easy to enforce a given minimum equivalence class size (k). Weighted-k–anonymity, on the other hand, works with a single generalization function and, instead of trying to enforce a fixed minimum equivalence class size, it flexibly creates equivalence classes of different sizes with no minimum size constraint. The outliers then must lie on smaller equivalence classes in order to maximize data utility. The common criteria in both the methods is that the outliers gets treated differently than the rest of the data set.

The weighted-k is an estimation of the equivalence class distribution, and hence, although the chances are very rare, a high value need not always indicate that there exists no tuple in the anonymized data set appearing in its original form. Since the method results in an average case analysis, rather than worst case, such generalizations can appear. However, in the context of multi-objective optimization, we cannot exclude such solutions as being faulty since they just represent an aspect of the privacy-utility tradeoff. Ideally, if another generalization with the same (or better) level of utility but without the isolated tuples exists, i.e. the tuples are embedded in equivalence classes of size more than one, then it would result in a higher value of weighted-k. Moreover, such a generalization will dominate the previous one and remove it from the solution set.

Note that, in most cases, not all equivalence classes with all possible k values will be generated. The weighted-k value provides a sufficiently good estimate of the distribution of the equivalence classes. A high weighted-k value implies that the size of the equivalence classes with higher k is relatively more than the size of the lower k ones. The multi-objective problem is then formulated as *finding the generalization that maximizes the weighted-k and minimizes the generalization loss for a given data set.*

5.3.2 Problem 2

In this problem, we enable suppression and allow the user to specify an acceptable fraction η of the maximum suppression loss possible ($n_{row} \cdot n_{col}$). Such an approach imposes a hard limit on the number of suppressions allowed [13]. However, unlike earlier approaches, by allowing the user to specify a suppression loss limit independent of k, the optimization procedure can be made to explore the trade-off properties of k and generalization loss within the constraint of the imposed suppression loss limitation.

When suppression is allowed within a user specified limit, all tuples belonging to the equivalence classes

$E_1, \ldots, E_d$ can be suppressed, where d satisfies the relation

$$\sum_{i=1}^{d}(|E_i| \cdot n_{col}) \leq \eta \cdot n_{row} \cdot n_{col} < \sum_{i=1}^{d+1}(|E_i| \cdot n_{col}) \tag{6}$$

Thereafter, the k value induced by the generalization is equal to $d+1$, which also satisfies the suppression loss constraint. We can now define our optimization problem as *finding the generalization that maximizes d and minimizes the generalization loss.* The problem can also be viewed as the maximization of k and minimization of GL satisfying the constraint $SL \leq \eta \cdot n_{row} \cdot n_{col}$. Note that the problem formulation allows the optimization procedure to find generalizations that create equivalence classes with lower k's of smaller size and thereby increase d.

5.3.3 Problem 3

The third problem is formulated as an extension of the second one where the user does not provide a maximum limit on the suppression loss. The challenge here is the computation of k, GL and SL for a generalization without having a baseline to start with. Since the three quantities are dependent on each other for their computation, it is important that we have some base k value to proceed. We adopt the weighted-k value at this point. Although not very precise, the weighted-k value provides a good estimate of the distribution of the equivalence classes. If a very high weighted-k value is obtained for a generalization, then the number of tuples with low k's is sufficiently small, in which case we can suppress them. If the weighted-k value is low, then most of the tuples belong to equivalence classes with low k. In this case, a higher amount of suppression is required to achieve an acceptable k for the anonymized data set. Also, high weighted-k generally implies a high generalization loss. Such trade-off characteristics are the point of analysis in this problem.

To start with, a particular generalization's weighted-k value is first computed. Thereafter, all tuples belonging to an equivalence class of $k < k_{weighted}$ are suppressed, enabling the computation of SL. This makes the k for the anonymized data set equal to at least $k_{weighted}$. The generalization loss GL is then computed from the remaining data set. The multi-objective problem is defined as *finding the generalization that maximizes $k_{weighted}$, and, minimizes GL and SL.*

5.3.4 Problem 4

This problem is motivated by the requirement that a data publisher may impose on obtaining an anonymized data set that is (k, ℓ)–safe. To formulate the problem, we define the equivalence class $E_{i,j}$. A tuple belongs to this equivalence class if it belongs to an i–anonymous and j–diverse class. Next, an order of importance is imposed on the k and ℓ properties. Such an order specifies which property is more desired by the data publisher and enables us to define a total ordering on the equivalence classes $E_{i,j}$. The ordering is obtained by first arranging the equivalence classes w.r.t. an increasing value in the least desired property, and then for a given value in this property, the equivalence classes are rearranged w.r.t. an increasing value in the most desired property. For example, if the ℓ property is more desirable (denoted by $k \ll \ell$), then an example total ordering could be $E_{1,1} < E_{1,2} < \ldots < E_{2,1} < E_{2,2} < \ldots$. Otherwise, the ordering would be $E_{1,1} < E_{2,1} < \ldots < E_{1,2} < E_{2,2} < \ldots$. The objective behind such an ordering is to find the first equivalence class E_{d_1,d_2} in that order such that, for a given acceptable fraction of suppression loss η,

$$k \ll \ell : n_{col} \cdot (\sum_{i=1}^{d_1-1} \sum_{j=1}^{\mathcal{L}} |E_{i,j}| + \sum_{j=1}^{d_2} |E_{d_1,j}|) > \eta \cdot n_{row} \cdot n_{col} \tag{7}$$

$$\ell \ll k : n_{col} \cdot (\sum_{i=1}^{\mathcal{K}} \sum_{j=1}^{d_2-1} |E_{i,j}| + \sum_{i=1}^{d_1} |E_{i,d_2}|) > \eta \cdot n_{row} \cdot n_{col} \tag{8}$$

where, $\mathcal{L}$ and $\mathcal{K}$ are the maximum obtainable values for ℓ and k respectively for the given data set. In other words, all tuples belonging to equivalence classes prior to E_{d_1,d_2} in the order can be suppressed without violating the suppression loss constraint. For a given generalization, the d_1 and d_2 values then signify the k and ℓ values obtainable within the suppression loss constraint. The multi-objective optimization problem is then defined as *finding the generalization that maximizes d_1, maximizes d_2, and minimizes GL.*

6 Solution Methodology

Classical approaches developed to handle multiple objectives concentrated on transforming the multi-objective problem into a special form of a single objective problem formulated using certain user-based preferences. However, because of the trade-off nature of multi-objective solutions, the quality of a solution obtained from a transformed single objective problem is contingent on the user-defined parameters. Evolutionary algorithms for multi-objective optimization are *multi-point methods* usually working with a population of solutions and concentrate on obtaining multiple optimal solutions in a single run. We thus employ the NSGA-II [6] algorithm to solve the multi-objective problems defined in the previous section.

6.1 Solution encoding

Before NSGA-II can be applied, a viable representation of the generalization has to be designed for the algorithm to work with. Here we adopt the encoding suggested by Iyengar in [14]. Consider the numeric attribute "age" with values in the domain $[10, 90]$. Since this domain can have infinite values, the first task is to granularize the domain into a finite number of intervals. For example, a granularity level of 5 shall discretize the domain to $\{[10, 15], (15, 20], \ldots, (85, 90]\}$. Note that this is not the generalization used to anonymize the dataset. The discretized domain can then be numbered as $1 : [10, 15], 2 : (15, 20], \ldots, 16 : (85, 90]$. The discretized domain still maintains the same ordering as in the continuous domain. A binary string of 15 bits can now be used to represent all possible generalizations for the attribute. The i^{th} bit in this string is 0 if the i^{th} and $(i + 1)^{th}$ intervals are supposed to be combined, otherwise 1. For attributes with a small domain and a defined ordering of the values, the granularization step can be skipped. For categorical data, a similar encoding can be obtained once an ordering on the domain values is imposed as discussed in Section 4. Fig. 3 shows an example generalization encoding for a "workclass" attribute. The individual encoding for each attribute are concatenated to create the overall encoding for the generalizations for all attributes.

6.2 NSGA-II

Similar to a simple genetic algorithm [5], NSGA-II starts with a population P_0 of N random generalizations. A generation index $t = 0, 1, \ldots, Gen_{MAX}$ keeps track of the number of iterations of the algorithm. Each trial generalization is used to create the anonymized dataset and the corresponding values of the quantities to be optimized are calculated. Each generation of NSGA-II then proceeds as follows. An offspring population Q_t is first created from the parent population P_t by applying the usual genetic operations of selection, crossover and mutation [5]. For constrained attributes, a special crossover operator is used as discussed in the next subsection. The offspring population also gets evaluated. The parent and offspring populations are then combined to form a population $R_t = P_t \cup Q_t$ of size $2N$. A non-dominated sorting is applied to R_t to rank each solution based on the number of solutions that dominate it. Rank 1 solutions are all non-dominated solutions in the population. A rank r solution is only dominated by solutions of lower ranks.

The population P_{t+1} is generated by selecting N solutions from R_t. The preference of a solution is decided based on its rank; lower the rank, higher the preference. By combining the parent and offspring population, and selecting from them using a non-dominance ranking, NSGA-II implements an elite-preservation strategy where the best solutions obtained are always passed on to the next generation. However, since not all solutions from R_t can be accommodated in P_{t+1}, a choice is likely to be made when the number of solutions of the currently considered rank is more than the remaining positions in P_{t+1}. Instead of making an arbitrary choice, NSGA-II uses an explicit diversity-preservation mechanism. The mechanism, based on a *crowding distance metric* [6], gives more preference to a solution with a lesser density of solutions surrounding it, thereby enforcing diversity in the population. The NSGA-II crowding distance metric for a solution is the sum of the average side-lengths of the cuboid generated by its neighboring solutions. Fig. 4 depicts a single generation of the algorithm. For a problem with M objectives, the overall complexity of NSGA-II is $O(MN^2)$.

6.3 Crossover for constrained attributes

The usual single point crossover operator in a genetic algorithm randomly chooses a crossover point and creates two offspring by combining parts of the bit string before and after the crossover point from two different parents. As shown in Fig. 5 (left), such an operation can result in an invalid generalization for constrained attributes. Iyengar proposed modifying such invalid generalizations to the nearest valid generalization [14]. However, finding the nearest valid generalization can be time consuming, besides destroying the properties on which the crossover operator is based on. In this regard, Lunacek et al. proposed a special crossover operator that always create valid offspring for constrained attributes [10]. Instead of randomly choosing a crossover point, their operator forces the crossover point to be chosen at a location where the bit value is one for both parents. By doing so, both parts (before and after the crossover point) of both parents can be guaranteed to be valid generalizations individually, which can then be combined without destroying the hierarchy requirement. Fig. 5 (right) shows an instance of this operator.

6.4 Population initialization

In order to be able to use Lunacek et al.'s crossover operator, the validity of the parent solutions must be guaranteed. This implies that the initial population that NSGA-II starts with must contain all valid generalizations for the constrained attributes. For a given hierarchy tree, we use the following algorithm to generate valid generalizations for the constrained attributes in the initial population.

Starting from the root node, a node randomly decides if it would allow its subtrees to be distinguishable. If it decides not to then all nodes in its subtrees are assigned the same identifier. Otherwise the root of each subtree receives an unique identifier. The decision is then translated to the root nodes of its subtrees and the process is repeated recursively. Once all leaf nodes are assigned an identifier, two adjacent leaf nodes in the imposed ordering are combined only if they have the same identifier. Since a parent node always make the decision if child nodes will be combined or not, all generalizations so produced will always be valid.

6.5 Experimental setup

We applied our methodology to the "adult.data" benchmark dataset available from the UCI machine learning repository[1]. The data was extracted from a census bureau database and has been extensively used in studies related to k–anonymization. We prepared the dataset as described in [13, 14]. All rows with missing values are removed from the dataset to finally have a total of 30162 rows. The attributes "age", "education", "race", "gender" and "salary class" are kept unconstrained, while the attributes "workclass", "marital status", "occupation" and "native country" are constrained by defining a hierarchy tree on them. The hierarchy trees are not shown in this paper due to space restrictions. The remaining attributes in the dataset are ignored. For Problem 4, the occupation attribute is considered sensitive.

For NSGA-II, we set the population size as 200 for Problem 1 and 2, and 500 for Problem 3 and 4. The maximum number of iterations is set as 250. A single point crossover is used for unconstrained attributes while Lunacek et al.'s crossover operator is used for constrained attributes. Also, mutation is only performed on the unconstrained attributes. The remaining parameters of the algorithm are set as follow: crossover rate = 0.9, mutation rate = 0.1 with binary tournament selection. We ran the algorithm with different initial populations but did not notice any significant difference in the solutions obtained. The results reported here are from one such run.

[1] ftp://ftp.ics.uci.edu/pub/machine-learning-databases/adult/

7 Results and Discussion

Before presenting our results and the analysis, we would like to emphasize that the rationale behind doing the multi-objective analysis is not to come up with a way of determining the best possible value of a model parameter. Our intention is focused at providing a global perspective of what values of the parameter are possible at different levels of data utility. The final choice of a solution depends on other feasibility criteria as well, for example, if the parameter value found at a particular utility level is acceptable to the human subjects involved or not. An inherent human factor (the data publisher or the human subjects) is thus involved in the selection of a final solution. Further, the use of NSGA-II may raise questions on whether the obtained solutions are optimal. It is possible that another algorithm, or a different metric, provides better solutions. However, our problem formulation neither has any dependency on the methodology chosen to solve them nor is particular to the loss metric used. Further, we want to emphasize that the solutions generated by the NSGA-II implementation are *valid* at each iteration of the algorithm owing to the approach we undertake in formulating the problems. For example, a solution always gives a generalization resulting in d–anonymity in Problem 2, and, d_1–anonymous and d_2–diverse in Problem 4. Of course the objective is to maximize these quantities along with the minimization of the information loss.

The parameters associated with NSGA-II did not have any significant effect on the quality of the solutions obtained. We believe that the special crossover operator provides a much faster rate of convergence as compared to the genetic algorithm implementation by Iyengar [14]. The following results are obtained from the standard settings as mentioned in the previous section.

The term *loss* in the following discussion signify the total information loss as a result of generalization and suppression, i.e. $loss = GL+SL$. The differentiation between GL and SL is made wherever appropriate.

Fig. 6 shows the different trade-off solutions obtained by NSGA-II. A point in the plot corresponds to a solution that induces a particular distribution of k values on the anonymized data set. As expected, the generalization loss increases as the distribution of equivalence classes gets more inclined towards higher k values. In the absence of suppression, a single k value is often hard to enforce for all tuples in the data set. Thus, a solution here results in different k values for different tuples. A higher k-weighted value signifies that most tuples have a high k value associated with them, in which case, the generalization loss is higher. A solution with low k-weighted value results in a generalization with low k values for its tuples.

The inset figures in the plot depict the cumulative distribution of the number of tuples belonging to equivalence classes (y-axis) with different k values (x-axis). The distributions of two extreme solutions corroborate the speculation that a higher generalization loss must be incurred to assure a greater level of privacy (higher k values) for a larger section of the dataset. Low generalization losses are only possible when most tuples belong to equivalence classes of lower k value.

However, it should be noted that the distributions for the two example solutions are not complementary in nature. For the solution with lower generalization loss, the distribution has a continuously increasing trend, implying that equivalence classes of different k values exist for the solution. The other solution shows an abrupt increase signifying that the tuples either belong to equivalence classes with very small k or ones with very large k. The sought balance in the distribution can therefore exist with an acceptable level of generalization loss.

Fig. 7 shows the trade-off between k and *loss* in Problem 2 when a maximum of 10% suppression loss is allowed. The top-leftmost plot shows all the solutions obtained for the problem. Each subsequent plot (follow arrows) is a magnification of the steepest part in the previous plot. Each plot shows the presence of locally flat regions where a substantial increase in the k value does not have a comparatively high increase in the *loss*. These regions can be of interest to a data publisher since it allows one to provide higher levels of data privacy without compromising much on the information content. Also, since the solutions corresponding to these flat regions evaluate to distantly separated k values, an analysis based on a single objective formulation with a fixed k shall require a much higher number of runs of the algorithm to identify such trade-off characteristics.

Interestingly, the trend of the solutions is similar in each plot. The existence of such repeated characteristics on the non-dominated front suggests that a data publisher's choice of a specific k, no matter how big or small, can have avenues for improvement, specially when the choice falls in the locally flat regions. A choice of k made on the rising parts of the front is seemingly not a good choice since the user would then be paying a high cost in degraded data quality without getting much improvement on the privacy factor. The rational decision choice in such a case would be to lower the k value to a flat region of the front. We observed similar trends in the solutions when the suppression loss was reduced to a low 1%.

The trade-off characteristics in Problem 3 are depicted in Fig. 8. Preliminary observations from the plot indicate that an increase in generalization loss results in a decrease in the suppression loss. A similar trend is observed when the k value increases. Since the k values in this case are computed directly from the weighted-k, an explanation for these observations is possible. A high generalization loss signifies that most tuples in the dataset belong to equivalence classes with high k values, thereby inducing a high weighted-k. This implies a low accumulation in suppression loss resulting from the deletion of tuples in equivalence classes with $k < k_{weighted}$. Also, as $k_{weighted}$ increases, the equivalence class distribution incline more towards the ones with high k values resulting in lesser number of tuples available for suppression.

The benefit of solving this problem comes in the form of an approximate solution set available for first-level analysis. For example, Fig. 9 (left) shows the solutions from the set when the suppression loss is set at a maximum allowable limit of 20%. Although GL and SL are conflicting objectives here, the analysis is intended to see if an acceptable level of balance can be obtained between the two with a reasonably good value of k. The encircled region in the plot show that three solutions around the point $(k = 5000, GL = 35\%, SL = 17\%)$ are available in this case, and hence a more specific analysis can be performed. A similar solution is found when the analysis is performed by setting the generalization loss limit to 30% (Fig. 9 (right)).

Fig. 10 depicts a subset of the solutions obtained for Problem 4. The solutions correspond to a suppression loss limit set as $\eta = 0.1$. Further, the plots only show solutions for which the $loss$ is less than 30%. The existence of multiple solutions for a fixed value of ℓ (or k) signifies that there is a trade-off involved in the amount of information loss and the value of k (or ℓ). An observation to make in here is the number of solutions obtained for varying values of k (or ℓ). When the preference is inclined towards the k–anonymity property, the solutions obtained give more choices for the parameter k than ℓ (Fig. 10 (left)). Similarly, when preference ordering is changed to $k \ll \ell$ (Fig. 10 (right)), more choices are available for the ℓ parameter. More importantly, any solution in either of the the two plots satisfy the 30% constraint on $loss$ and hence is a viable solution to a data publisher's request with the same constraints. To choose a single solution, we can follow the same preference ordering as was used while defining the optimization problem. For example, if the ℓ–diversity property is more desirable, we can choose the solution with the highest value of k from the set of solutions with the highest value of ℓ (a $(15, 7)$–safe solution in the plot).

We can extend our analysis to see if improvements are obtainable without much increase in the information loss. Often, the data publisher's constraint on the information loss is specified without an understanding of the trade-offs possible. A multi-objective analysis reveals the nature of these trade-offs among different objectives and can provide suggestions on how one might be improved. For example, Fig. 11 (left) shows solutions when the data publisher has given a 20% constraint on the information $loss$. For a good balance between the two desired privacy properties, say the data publisher chooses the $(20, 4)$–safe solution. Solutions with $\ell = 3, 5$, or 6 are avoided at this point because of the low value of k associated with these solutions. Fig. 11 (right) shows the solutions to the same problem but with a slightly higher *loss* limit of 21%. The encircled solutions depict the new choices that are now revealed to the data publisher. For a small amount of increase in the *loss* limit, the data publisher now has a choice – $(20, 5)$–safe – which offers the same value of k as in the old choice, but with a higher value of ℓ. Further, the earlier reluctance to choose a solution with $\ell = 3$ is reduced after the revelation of the $(22, 3)$–safe solution. In fact, there is even a candidate solution for $\ell = 6$ and a much better value in k. With this information, the data publisher now has some idea about the trade-offs possible between privacy and information loss. In fact, the trade-off analysis in this case may motivate the data publisher to relax the loss constraint, which is not a big relaxation in itself, and reap the benefits of better privacy.

It is possible to analyze the trade-off surface generated for a particular data set and provide the data publisher with an analytical form for it. Given that an approximation of the Pareto-front is known, an analytical form can be derived through a polynomial curve fitting approach. However, it must be noted that analyzing the privacy-utility tradeoff in this manner is rather problem specific. It cannot be ascertained that the Pareto-front always has a definitive structure for all data sets. Any theoretical analysis motivated from Pareto behavior in a data set is limited to that particular data set, and is not directly extensible to another data set. We cannot say for sure if the Pareto front will be similar for different data sets with similar distributions. Understanding the trade-off is rather empirical in this work, represented directly in terms of the parameter values and the resulting utility (represented by the value of the utility metric of choice). Nonetheless, it is not always true that empirical results are just tuples of ⟨privacy,utility⟩ values without providing much insight into the trade-off involved. Observing the Pareto-front on a graphical plot can reveal underlying traits. A classic case of this is seen in Fig. 7. Our observations indicate here that a major part of the Pareto-front is flat, or steep, in this data set. This signify that the data publisher has more flexibility in choosing a k value for a given level of utility. The steep jumps in utility signify that there exist points for which utility can often be improved significantly with a slight deterioration in privacy.

To summarize the above discussion, we go back to the questions asked by the data publisher in Section 1 and try to provide answers to them w.r.t. the benchmark data set.

1. We can generalize the data set in such a way that more number of tuples have a low probability of

being identified by a linking attack. There is a generalization that results in 22% loss and attains a weighted average k value of 2528 (from Fig. 6). The inset figure shows that a substantial fraction of the tuples belong to high k equivalence classes.

2. For the constraints given, a generalization with $k = 14$ is known (from Fig. 7). However, if the information loss constraint can be relaxed to 26%, a solution with $k = 36$ is known. Note that analysis of the nature performed in Problem 3 can be used to provide further suggestions on the trade-offs available for suppression loss.

3. A (k, ℓ)–safe solution can provide the benefits of both k–anonymity and ℓ–diversity. A generalization with a high value of k and ℓ can be an answer. However, it is required that the more desired property be specified for better analysis.

4. For the given constraints, and assuming that ℓ–diversity is more desired, a solution with $k = 20$ and $\ell = 4$ offers a good balance between the two privacy measures (from Fig. 11). There are other solutions with trade-offs in the k and ℓ values. However, if the information loss constraint is relaxed to 21%, the ℓ value can be increased to 5. Besides, this will also allow two additional solutions: $(22, 3)$–safe and $(18, 6)$–safe.

8 Conclusions

In this paper, we present an analytical approach to demonstrate that the choice of the parameter value in the k–anonymity problem can be made in a much informed manner rather than arbitrarily. The multi-objective problems are formulated to cater to differing requirements of a decision maker, primarily focused on the maximization of the k value and minimization of the losses.

For generalizations without suppression, an unique k may not be available. However, the analysis indicates that generalizations are possible that provide a higher level of privacy for a higher fraction of the dataset without compromising much on its information content. When suppression is allowed up to a hard limit, the user's choice of k should be based on an analysis similar to that performed in Problem 2. Typically, the nature of the non-dominated solution set provides invaluable information on whether an anonymization exists to improve a particular value of the model parameter without much degradation in quality of the data. First-level explorations in this context can begin with gaining an overall understanding of the trade-off characteristics in the search space. Our results also indicate that different privacy models can be combined and optimized to result in minimal information loss. However, the trade-off picture is better portrayed in cases when the model parameters are kept separated and formulated as multiple objectives.

The formulations presented in the paper also address the data publisher's dilemma. They provide a methodology to analyze the problem of data anonymization in manners that appeal to the actual entity that disseminates the data. We believe that such an analysis not only reinstates the data publisher's confidence in its choice of a particular privacy model parameter, but also identifies ways of examining if the level of privacy requested by a human subject is achievable within the acceptable limits of perturbing data quality.

Future work in this direction can start with examination of the framework with other models of privacy preservation. Real valued parametrization of t makes the t–closeness model an interesting subsequent candidate. Hybrid models catering to different forms of attacks are also required. Work on this can begin with an exploration on what trade-offs are generated when looking for the existence of two, or more, privacy properties simultaneously. We believe that transitioning these different models into the real world requires us to synchronize our perspective of the problem with those that actually deal with it.

References

[1] A. Hundepool and L. Willenborg. Mu and Tau Argus: Software for Statistical Disclosure Control. In *Proceedings of the Third International Seminar on Statistical Confidentiality*, 1996.

[2] A. Takemura. Local Recoding by Maximum Weight Matching for Disclosure Control of Microdata Sets. CIRJE F-Series CIRJE-F-40, CIRJE, Faculty of Economics, University of Tokyo, 1999.

[3] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ–Diversity: Privacy Beyond k–Anonymity. In *ICDE 2006: Proceedings of the 22nd International Conference on Data Engineering*, page 24, Atlanta, GA, USA, 2006.

[4] C. A. C. Coello. An Updated Survey of GA-Based Multiobjective Optimization Techniques. *ACM Computing Surveys*, 32(2):109–143, 2000.

[5] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[7] K. Deb. *Multi-objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons Inc., 2001.

[8] L. Sweeney. Achieving k–Anonymity Privacy Protection Using Generalization and Suppression. *International Journal on Uncertainity, Fuzziness and Knowledge-based Systems*, 10(5):571–588, 2002.

[9] L. Sweeney. k–Anonymity: A Model for Protecting Privacy. *International Journal on Uncertainity, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[10] M. Lunacek, D. Whitley, and I. Ray. A Crossover Operator for the k-Anonymity Problem. In *GECCO 2006: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 1713–1720, Seattle, Washington, USA, 2006.

[11] N. Li, T. Li, and S. Venkatasubramanian. t–Closeness: Privacy Beyond k–Anonymity and ℓ–Diversity. In *ICDE 2007: Proceedings of the 23rd International Conference on Data Engineering*, pages 106–115, Atlanta, GA, USA, 2007.

[12] P. Samarati. Protecting Respondents' Identities in Microdata Release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[13] R. J. Bayardo and R. Agrawal. Data Privacy Through Optimal k-Anonymization. In *ICDE 2005: Proceedings of the 21st International Conference on Data Engineering*, pages 217–228, Tokyo, Japan, 2005.

[14] V. S. Iyengar. Transforming Data to Satisfy Privacy Constraints. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 279–288, Alberta, Canada, 2002.

[15] W. Winkler. Using Simulated Annealing for k–Anonymity. Technical report, US Census Bureau Statistical Research Division, Washington, DC, USA, 2002.

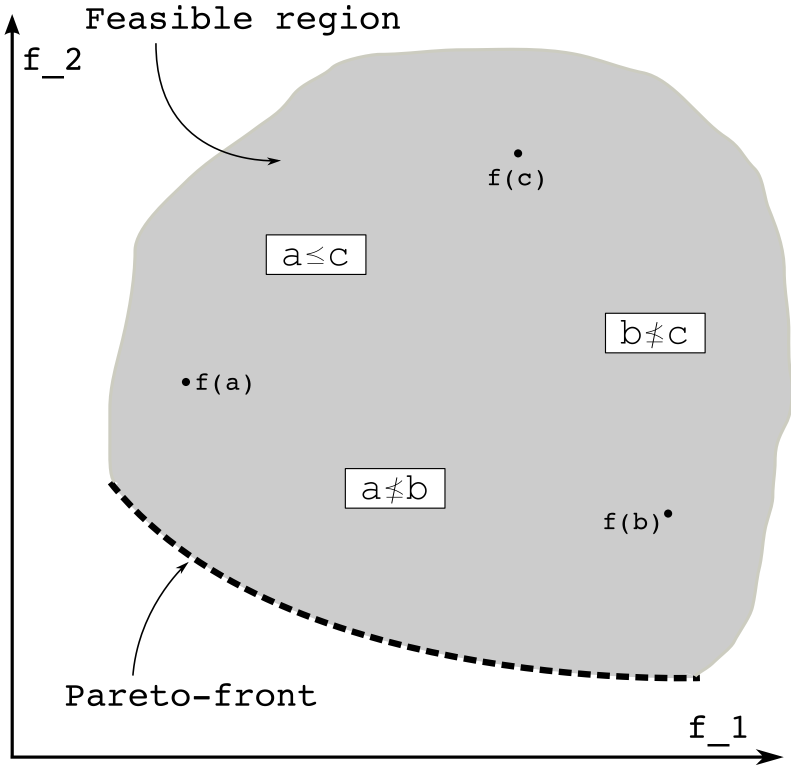

Figure 1: Pareto-front for a hypothetical two-objective problem.

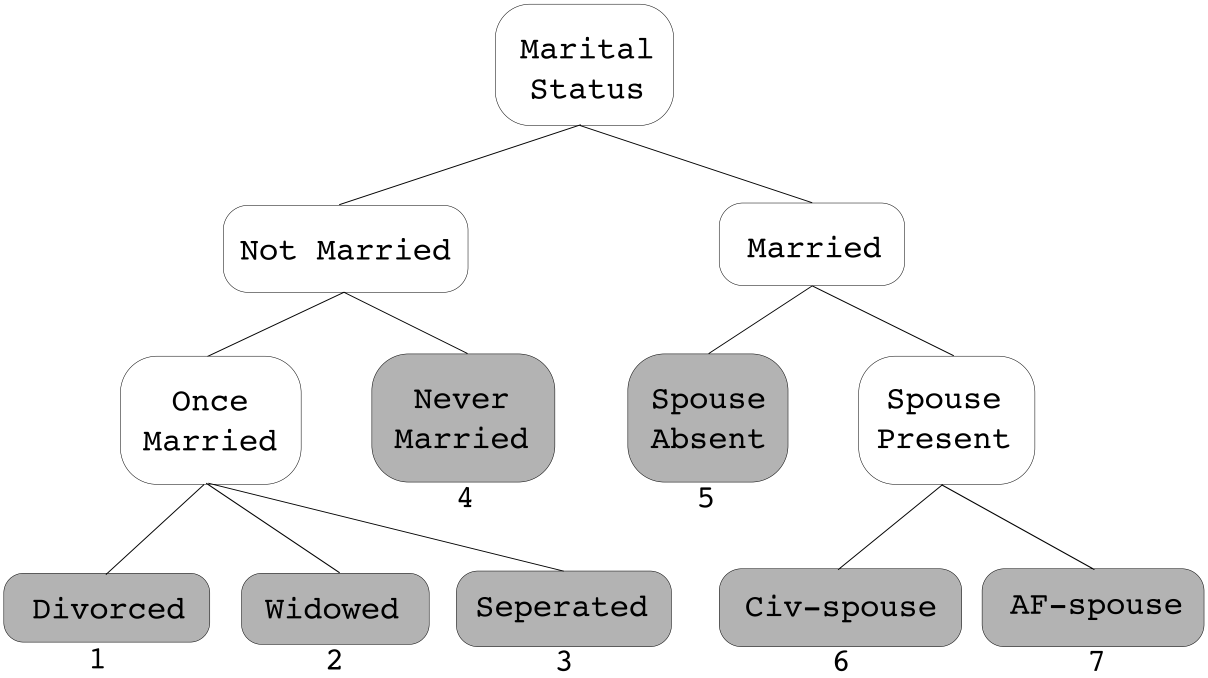

Figure 2: Hierarchy tree for the *marital status* attribute. Numbering on the leaf nodes indicate their ordering in $\Sigma_{marital\ status}$.

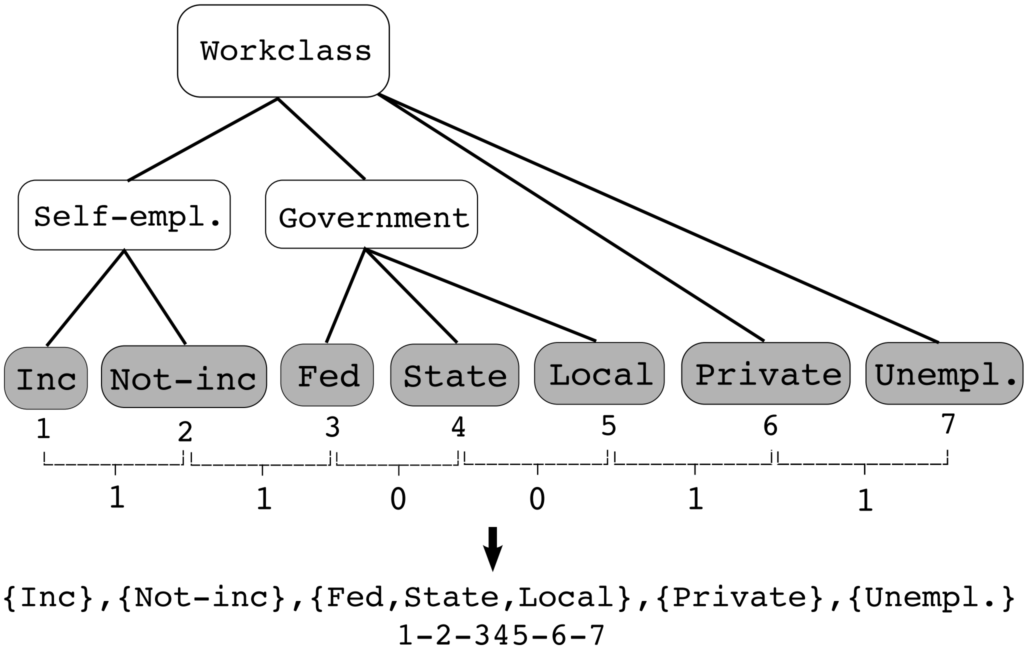

Figure 3: Example generalization encoding for the *workclass* constrained attribute.

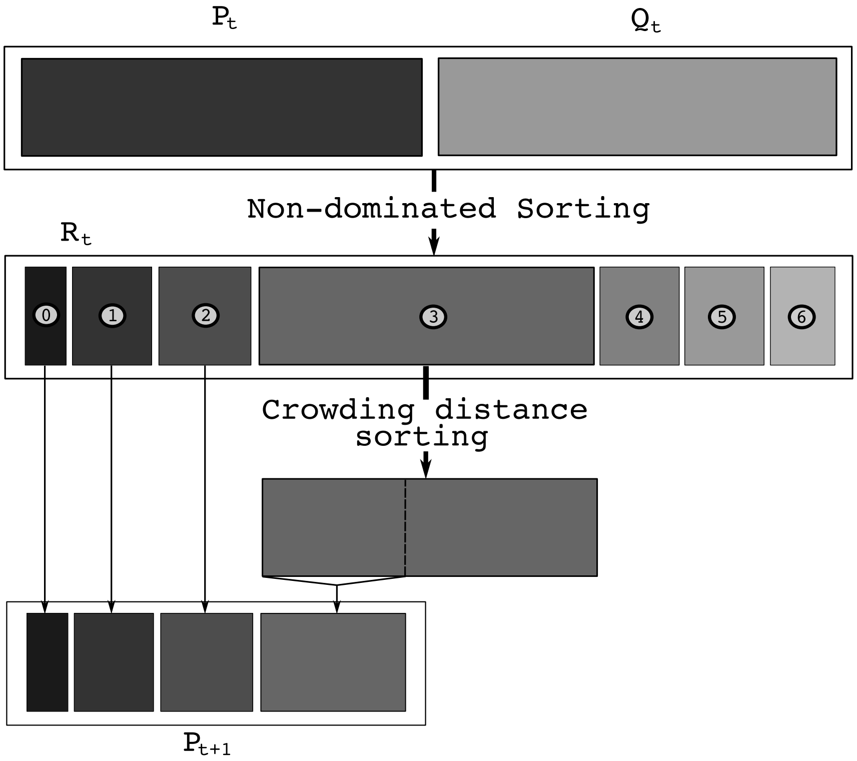

Figure 4: One generation of NSGA-II.

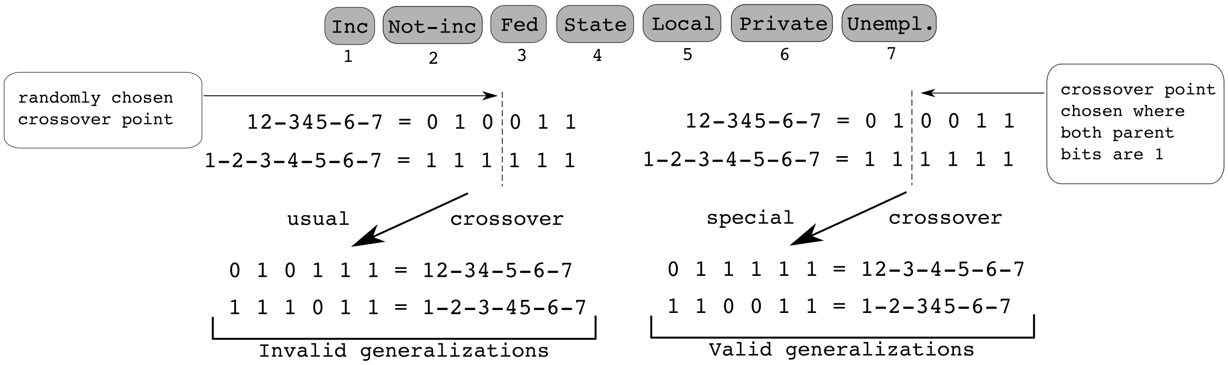

Figure 5: Usual single point crossover (left) and special crossover for constrained attributes (right).

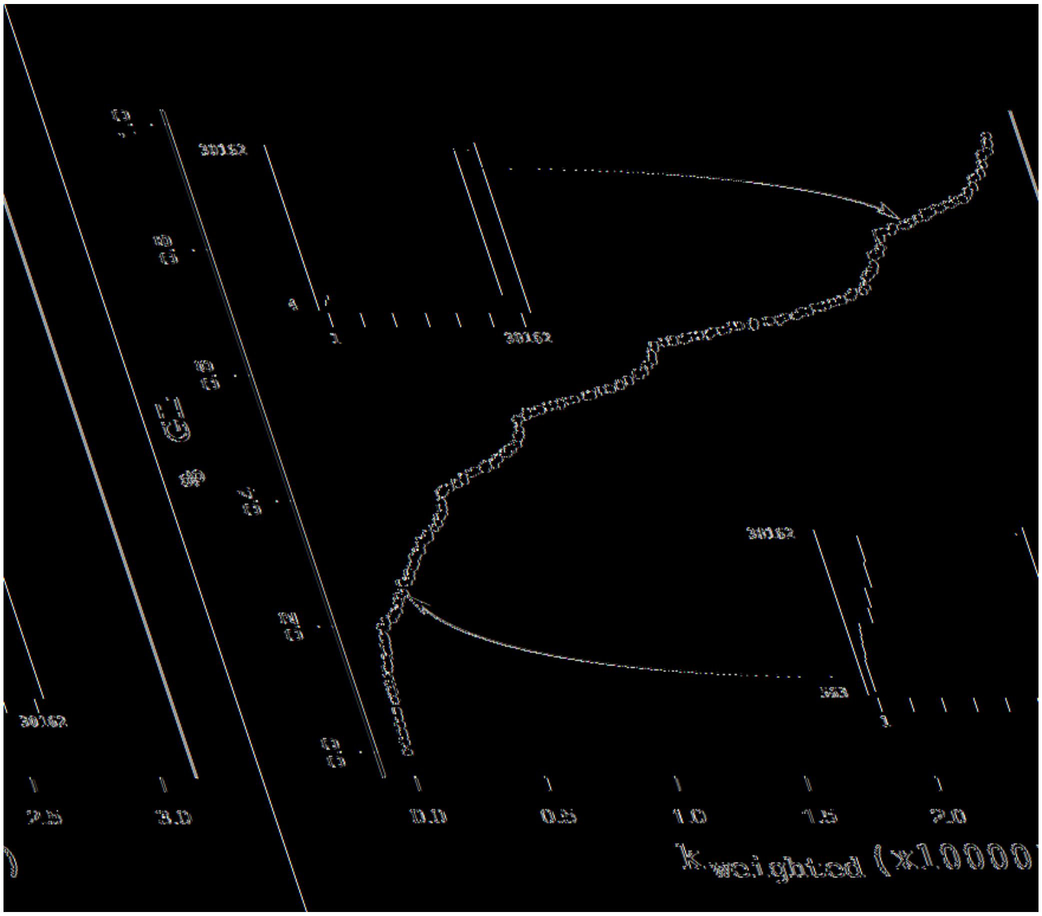

Figure 6: Solutions to Problem 1 found by NSGA-II. Inset figures show cumulative distribution of $|E_k|$ as k increases.

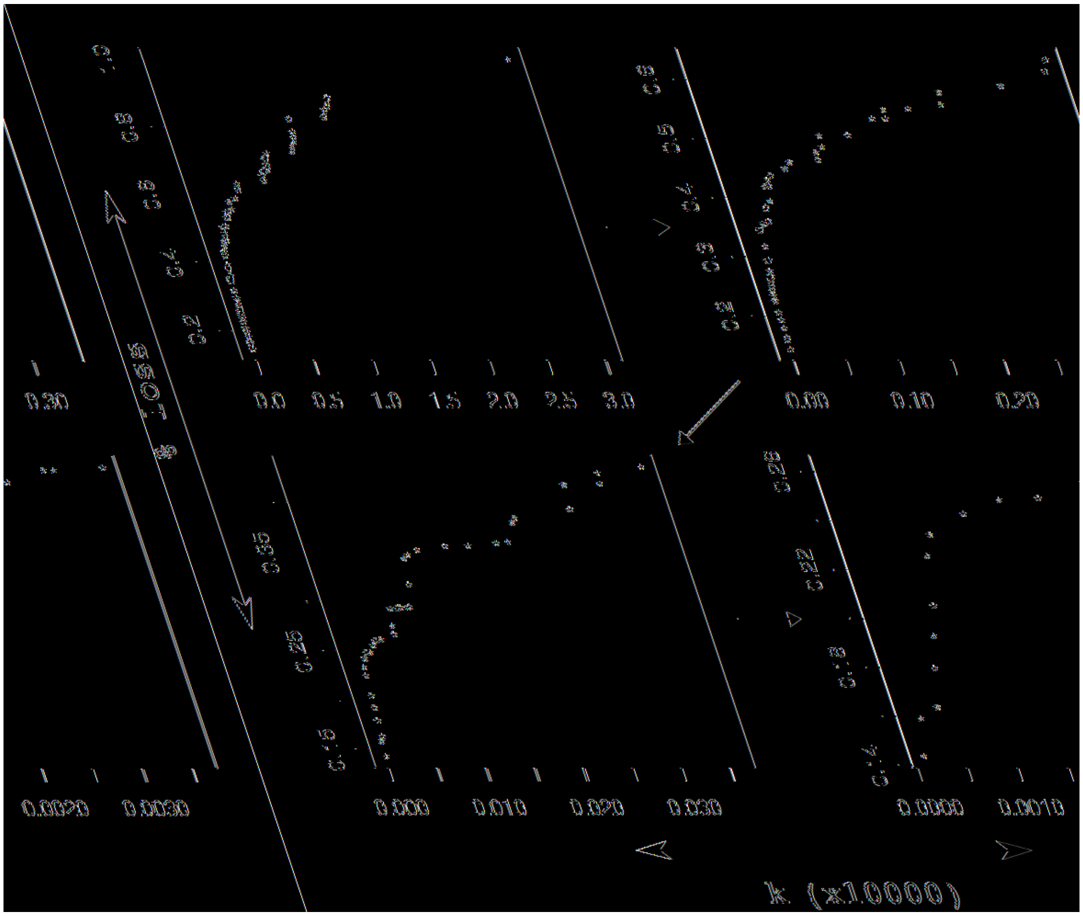

Figure 7: Solutions to Problem 2 (η = 10%) found by NSGA-II. Top-leftmost plot shows all obtained solutions. Each subsequent plot (follow arrows) is a magnification of a region of the previous plot.

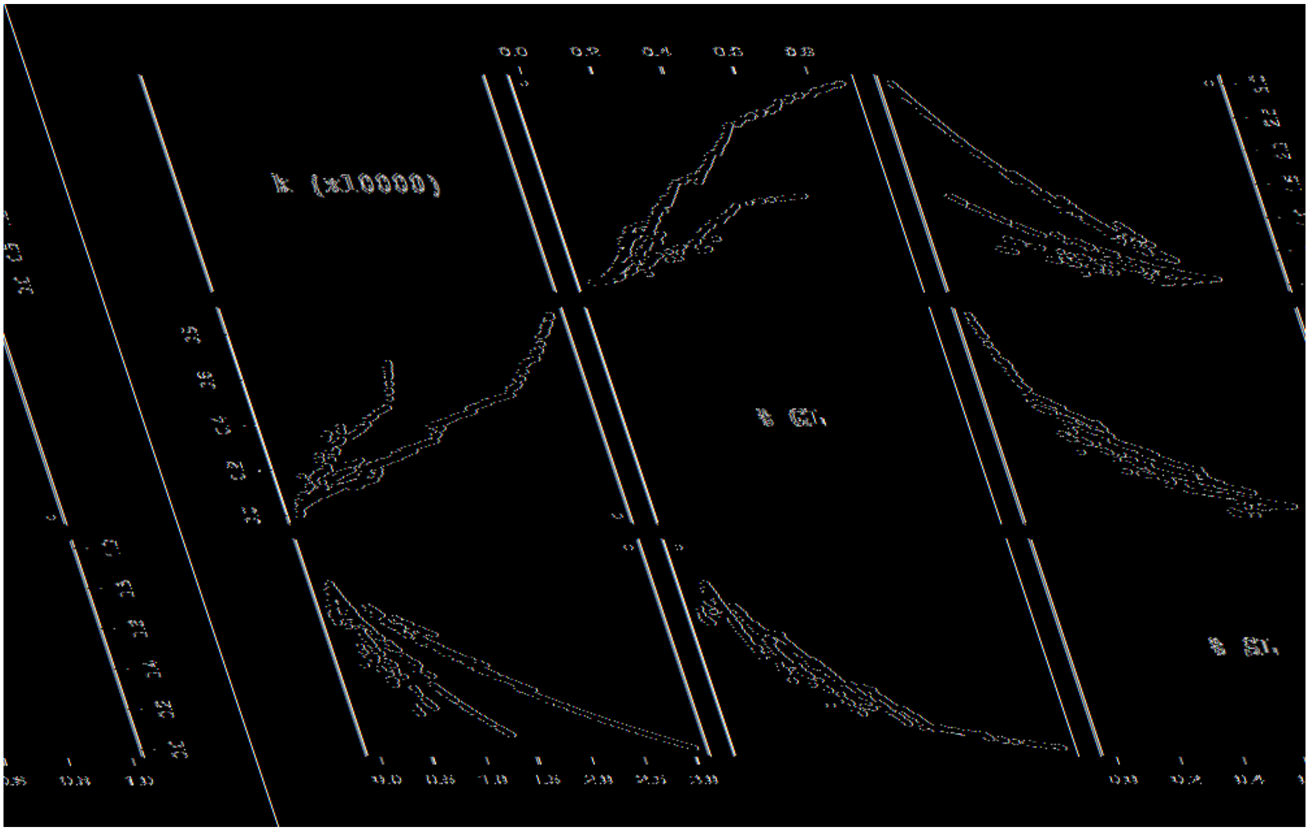

Figure 8: Solutions to Problem 3 found by NSGA-II. Read x-axis label for a plot from the text-box along the same column and y-axis label from the text-box along the same row.Trade-off characteristics are visible across different pairs of the objective functions.

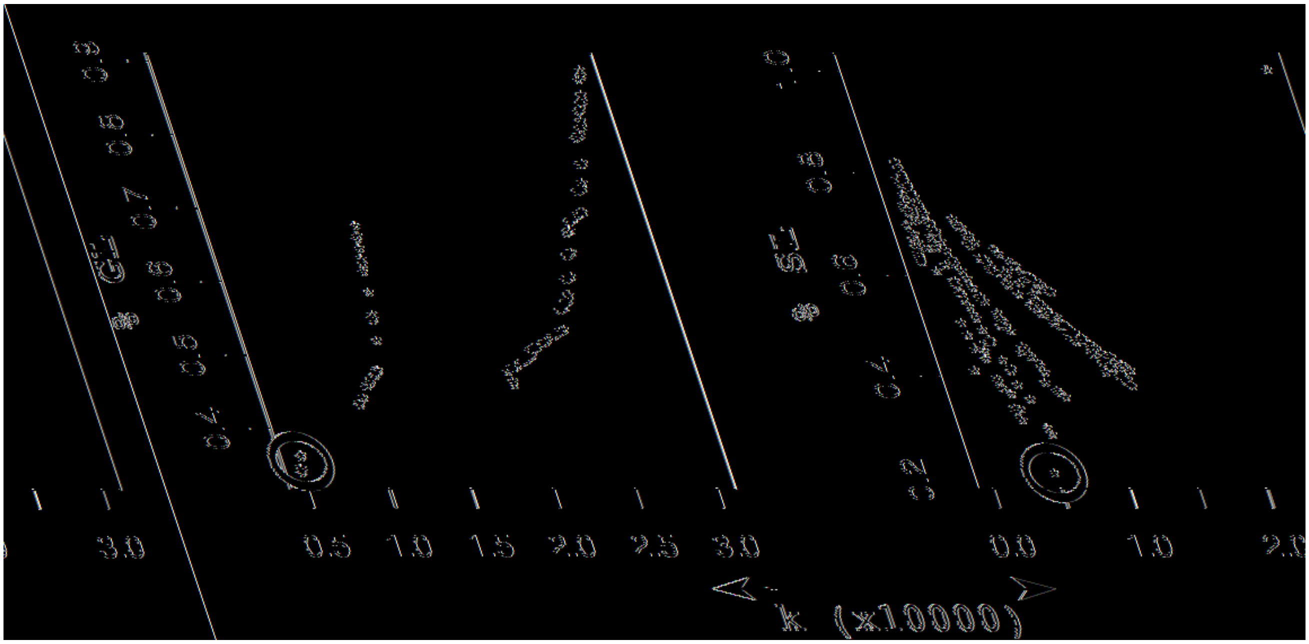

Figure 9: Problem 3 solutions for $\%SL < 0.2$ (left) and $\%GL < 0.3$ (right). Encircled solutions can be of interest to an user.

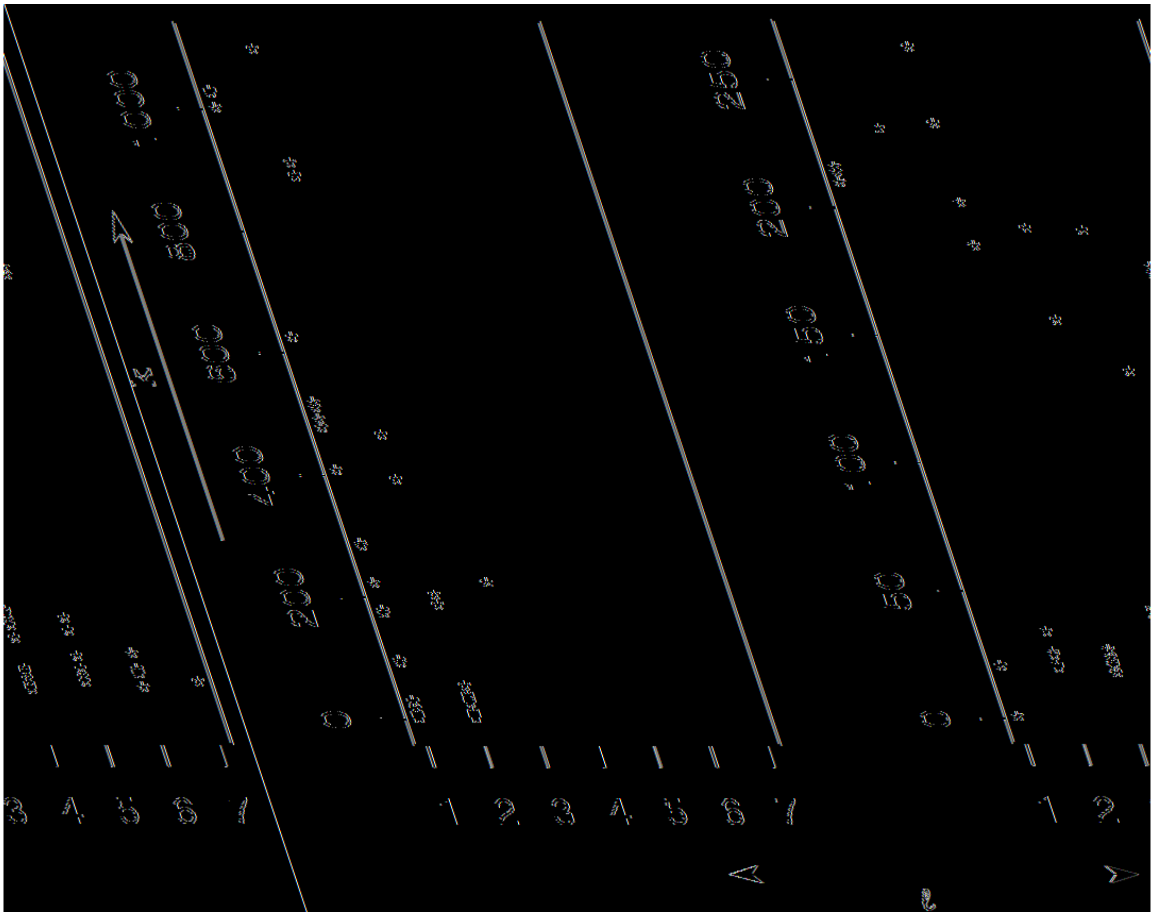

Figure 10: Problem 4 solutions for $\eta = 0.1$ and $\%loss < 0.3$. Left plot shows solutions when k–anonymity is more desired and right plot shows solutions when ℓ–diversity is more desired.

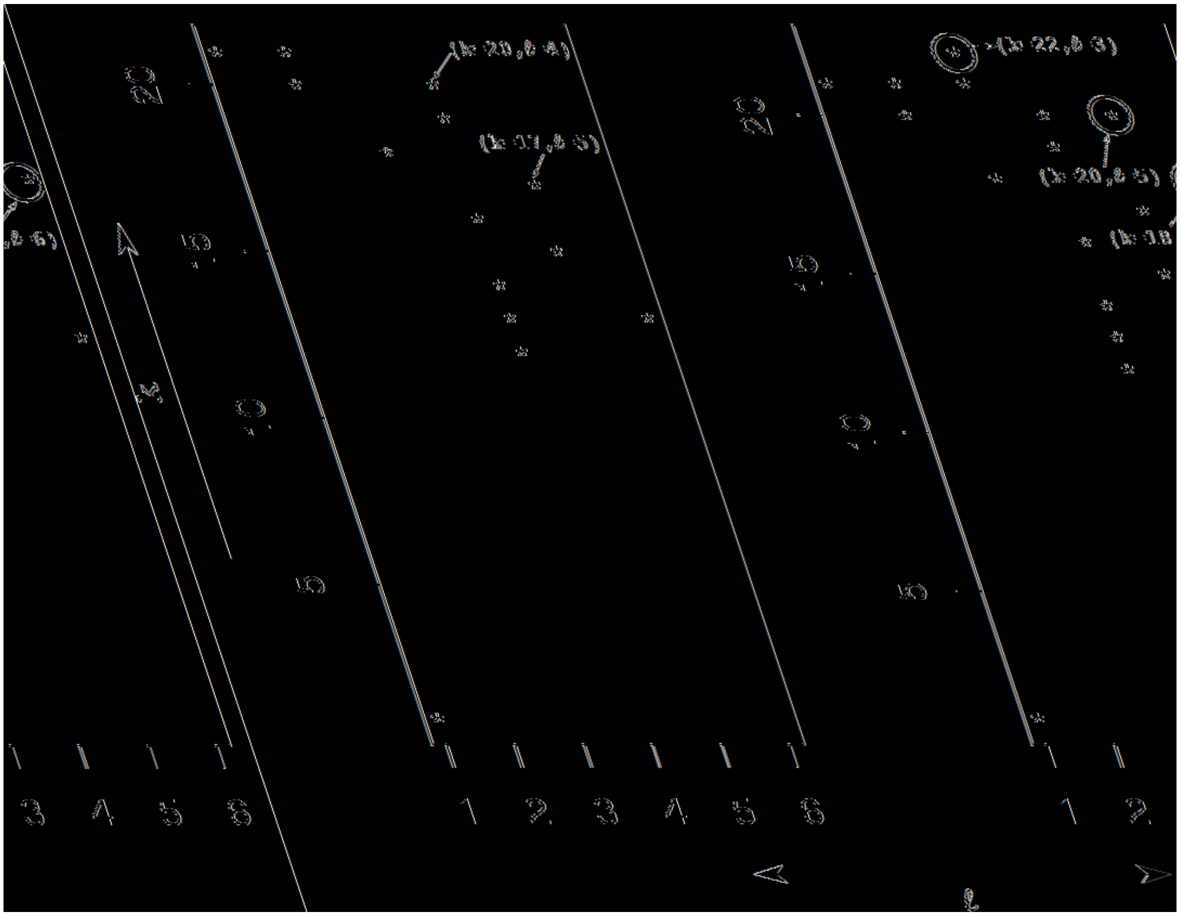

Figure 11: Problem 4 solutions for $\eta = 0.1$ and $k \ll \ell$. Left plot shows solutions when maximum *loss* allowed is 20%. Right plot shows solutions when maximum *loss* is increased to 21%. The quality of solutions as well as the number of choices available increase considerably for a slight increase in the *loss* limit.