$k-$Anonymization in the Presence of Publisher Preferences

Rinku Dewri, Indrajit Ray, Indrakshi Ray, and Darrell Whitley

Abstract—Privacy constraints are typically enforced on shared data that contain sensitive personal attributes. However, owing to its adverse effect on the utility of the data, information loss must be minimized while sanitizing data. Existing methods for this purpose modify the data only to the extent necessary to satisfy the privacy constraints, thereby asserting that the information loss has been minimized. However, given the subjective nature of information loss, it is often difficult to justify such an assertion. In this paper, we propose an interactive procedure to generate a data generalization scheme that optimally meets the preferences of the data publisher. A data publisher guides the sanitization process by specifying aspirations in terms of desired achievement levels in the objectives. A reference direction based methodology is used to investigate neighborhood solutions if the generated scheme is not acceptable. This approach draws its power from the constructive input received from the publisher about the suitability of a solution before finding a new one.

Index Terms—k-anonymity, reference point optimization

---

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 information, improper disclosure of which can lead to serious privacy violations. Dissemination of the data is therefore controlled by various privacy requirements. However, merely removing personally identifiable information, such as name and social security number, is often not sufficient to ensure privacy.

To address such privacy concerns, Samarati and Sweeney proposed that *data generalization* be used to satisfy a property called *k–anonymity* [2], [3]. Generalization is performed by grouping together data attribute values into a more general one, for example, replacing the attribute age by an age range. A transformed data set of this nature is then said to be k–anonymous if each record in it is same as at least $k-1$ other records.

A consequential drawback of performing generalization is a loss in information content of the data set. A number of algorithms have therefore been proposed to generalize a data set to meet the k–anonymity property while resulting in minimum information loss [4], [5], [6], [7], [8], [9], [10], [11], [12]. The standard approach is to progressively generalize the data until it is k–anonymous. However, such an approach cannot guarantee optimality if different attributes carry different levels of significance. For example, in a medical data set, attributes such as age and disease are more important than the ZIP code of the underlying patient. This opens

- R. Dewri is with the Department of Computer Science, University of Denver, Denver, CO 80208.

- I. Ray, I. Ray and D. Whitley are with the Department of Computer Science, Colorado State University, Fort Collins, CO 80523.

up the possibility that a minimum information loss can be sustained even for higher values of k, thereby providing better privacy than specified. Searching for higher privacy generalizations is also fruitful if the data publisher can tolerate an information loss higher than the minimum possible. Existing optimization attempts do not embrace such preference criteria.

Further, k–anonymity is only a minimalistic measure of the privacy level. The actual privacy levels of two individuals in a k–anonymous data set can be very different. For example, consider a 3–anonymous data set. If record A is same as 2 other records while record B is same as 9 other records, the privacy level of individual B is much higher (9/10) than that of individual A (2/3). This characteristic, which we call the *anonymization bias* [13], is induced by the nature of the k–anonymity model since it only helps to identify the worst case privacy level.

As the first contribution in this paper, we propose an adaptation of a goal programming based interactive procedure to resolve the problem of choosing a generalization scheme that meets a privacy property along with minimum bias and information loss. First, we build on the idea of anonymization bias to provide a quantitative measurement of the feature. This enables us to define a precise vector optimization problem for minimizing the privacy bias and information loss. An interactive procedure is discussed to help explore the set of optimal solutions to this problem based on feedback received from the data publisher. The procedure employs a *reference direction approach* in order to generate multiple solutions in the neighborhood of the data publisher's preferences.

The second contribution is an approach to obtain data generalizations satisfying the k–anonymity property given preference values on the information loss and

privacy bias. As part of the reference direction approach, the scalarizing function formulated in the first contribution is subjected to a constrained minimization. We show how the proposed evolutionary multi-objective approach solves the minimization problem and helps resolve the issue of finding better privacy levels than specified (by the parameter k) in the presence of varying data attribute significance and data publisher preferences.

The remainder of the paper is organized as follows. Section 2 reviews some of the existing works in k–anonymization. Section 3 provides a preliminary background on the problem. Section 4 provides a formal definition of an efficient solution to the problem. A scalarization of the vector probem is discussed in Section 5. Section 6 discusses the interactive reference direction approach designed on top of a minimax problem. Section 7 presents our multi-objective algorithm to solve the minimax problem. Empirical results on a benchmark data set are presented in Section 8. Finally, Section 9 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 [6]. Sweeney's *Datafly* approach uses heuristics to generalize the attribute containing the most distinct sequence of values for a specified subset of quasi-identifiers [3]. Samarati's algorithm [11] 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. A similar full-domain generalization is also proposed in *Incognito* [8]. Iyengar proposes a flexible generalization scheme and uses a genetic algorithm to perform k–anonymization on the larger search space that resulted from it [7].

Meyerson and Williams have recently proposed an approximation algorithm that achieves an anonymization within O($k \log k$) of the optimal one [14]. However, the method is not suitable when larger values of k are desired.

Bayardo and Agrawal propose a complete search method that iteratively constructs less generalized solutions starting from a completely generalized data set [4], stopping at the one that is minimally k–anonymous. The idea of a *solution cut* is presented by Fung et al. in their approach to top down specialization [5]. LeFevre et al. extend the notion of generalizations on attributes to generalization on tuples in the data set [9].

The drawbacks of using k–anonymity are first described by Machanavajjhala et al. [15]. They propose the ℓ–diversity model that obtains anonymizations with an emphasis on the diversity of sensitive attribute values on a k–anonymous equivalence class. Further work presented by Li et al. show that the ℓ–diversity model is also susceptible to certain types of attacks [16], and they proposed having the t–closeness property to prevent such attacks. The notion of *personalized anonymity* has also been proposed to find generalizations that satisfy personal privacy requirements [17].

Quantification of data utility has been approached from different perspectives by researchers. Early notion of information loss is based on the number of generalization steps one has to perform to achieve a given privacy requirement [11]. Information loss is also measured in terms of the amount of distortion in a generalized table [3], [18]. The *general loss metric* [7] computes a normalized information loss for each data value in the generalized data set. A widely used loss metric, called the *discernibility metric*, assigns a penalty to each tuple based on the number of tuples in the anonymized data set that are indistinguishable from each other [4]. Preliminary metrics to evaluate the effectiveness of anonymized data in answering aggregate queries have also been proposed [9], [19].

Dewri et al. focus on a multi-objective optimization formulation of the privacy-utility problem based on a model called *weighted-k anonymity* [20]. A similar trade-off analysis is presented by Huang and Du in the problem of optimizing randomized response schemes for privacy protection [21]. Li and Li reinforced the requirement for privacy-utility trade-off analysis stressing on the fact that privacy is typically a specific characteristic of a data set while utility is an aggregate characteristic [22]. However, these approaches are limited in terms of incorporating preference criteria on the data quality or anonymization bias.

3 DISCLOSURE CONTROL

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. Categorical data are usually associated with a taxonomy tree. The leaf nodes in this tree 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 tree [7].

A *generalization* G_j for an attribute a_j is a partitioning of the set Σ_j into ordered subsets $\langle \Sigma_{j_1}, \Sigma_{j_2}, \ldots, \Sigma_{j_P} \rangle$ that 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 with the

TABLE 1
Example data set and its 2–anonymous generalized version.

Employee Code (*emp*)	Salary Class (*sal*)	*emp*	*sal*
81521	C1	8152*	C1
81522	C1	8152*	C1
81523	C1	8152*	C1
82635	C2	8263*	C2
82636	C2	8263*	C2
82647	C2	8264*	C2
82648	C2	8264*	C2
81634	C3	8163*	C3
81631	C3	8163*	C3
81632	C3	8163*	C3
81639	C3	8163*	C3
81630	C3	8163*	C3

domain $\{10, \ldots, 90\}$, a possible generalization can be $\langle \{[10, 30]\}, \{(30, 50]\}, \{(50, 70]\}, \{(70, 90]\} \rangle$. For categorical attributes, a generalization is typically required to respect the taxonomy tree.

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_{ij} at row i and column j in D by $G_j(v_{ij})$ where $G_j(v_{ij})$ gives the subset index to which v_{ij} belongs in the generalization G_j. Note that the number of partitions (or groups) of an attribute domain, i.e. P, signifies the extent of generalization that will be performed for the attribute. If $P = 1$ then all values of the attribute will be transformed to the same subset index 1, in which case all information in that attribute is lost. On the other extreme, if $P = |\Sigma_j|$ for attribute a_j then every value will map to its own unique index (no generalization) and all information in the attribute will be maintained in the original form.

A consequence of performing generalization is the appearance of equivalent tuples. Two tuples in D are equivalent if their subset indices are equal in every column of D′. Such equivalent tuples can then be grouped together into equivalence classes. We associate a value ec_i to each tuple in D′ signifying the size of the equivalence class to which it belongs. *k–anonymity* is then defined as follows.

Definition 1 (k–anonymity). *An anonymized data set* D′ *is said to be k–anonymous if* $min(EC_{D'}) \geq k$, *where* $EC_{D'}$ *is the vector* $(ec_1, \ldots, ec_{n_{row}})$ *for* D′.

Example 1 *Consider the data set in Table 1 (left). The data set has 12 entries of 5-digit employee codes and the corresponding salary class. The right table is a generalized version of this data set where the last digit of the employee code is removed. As a result, the entries can be grouped together into equivalence classes and the data set becomes 2–anonymous.*

In other words, every tuple in a k–anonymous data set is same as at least $k - 1$ other tuples. Higher the value of the parameter k, better is the privacy guarantee. We can say that the probability of privacy breach is at most $1/k$ in a k–anonymous data set. k–anonymity satisfies the *monotonicity* property, i.e. a k–anonymous data set

is also $(k - 1)$–anonymous. We shall thus refer to the parameter k in k–anonymity as k_{pref} and $min(EC_{D'})$ as the effective k resulting from the generalizations.

3.1 Normalized Weighted Penalty

Consider the data value v_{ij} at row i and column j in the data set D. Let $g_{ij} = G_j(v_{ij})$ be the index of the subset to which v_{ij} belongs in the generalization G_j, i.e. $v_{ij} \in \Sigma_{j_{g_{ij}}}$. Further, let $(w_1, \ldots, w_{n_{col}})$ be a vector of weights where weight $0 \leq w_i \leq 1$ reflects the importance of the attribute a_i. The sum of weights is fixed at 1.0. The penalty for information loss associated with the value v_{ij} is then given as follows.

$$penalty(v_{ij}) = \frac{w_j(|\Sigma_{j_{g_{ij}}}| - 1)}{(|\Sigma_j| - 1)}$$

The loss is thus proportional to the size of the partition to which a data value belongs to. It attains a maximum value (equal to the weight of the attribute) when $P = 1$. The *normalized weighted penalty* in D′ is then obtained as the fractional penalty over all tuples in the data set.

$$NWP(D') = \frac{\sum_{i=1}^{n_{row}} \sum_{j=1}^{n_{col}} penalty(v_{ij})}{n_{row}}$$

3.2 Normalized Equivalence Class Dispersion

The k–anonymity model is only representative of the worst case privacy measurement. As a result, it is possible that two anonymized versions of a data set, both satisfying k–anonymity, result in very different equivalence class sizes for the tuples. The privacy level of a tuple is directly related to its ec_i value – the higher the value, lower is the probability of privacy breach. Since the k–anonymity definition does not enforce any requirement on how ec_i values should be distributed, it is often possible that an anonymization is biased towards a set of tuples ($ec_i \gg k_{pref}$) while providing minimalistic privacy ($ec_i = k_{pref}$) for others. Our attempt here is to control the occurrence of such biased privacy within acceptable limits.

The value of ec_i for a tuple can range from 1 to the number of tuples in the data set, i.e. n_{row}. This range reflects the maximum bias that can be present in the anonymized data set. The *normalized equivalence class dispersion* measures the bias as the maximum dispersion present in the ec_i values relative to the maximum possible dispersion.

$$NECD(D') = \frac{\max(EC_{D'}) - \min(EC_{D'})}{n_{row} - 1}$$

Note that tighter privacy constraints can implicitly satisfy relaxed ones owing to the monotonicity property. Hence, if the privacy constraint is 2–anonymity, then any generalization that achieves k–anonymity with $k \geq 2$ is a privacy preserving generalization. In this case, a 2–anonymous and a 3–anonymous table are both privacy preserving but with the possibility that the latter induces lower bias than the former.

4 OBJECTIVE SCALARIZATION

A typical vector optimization problem involves decision making under the presence of multiple conflicting objectives. The most important characteristic of these problems is the non-existence of a single optima, but rather a set of "incomparable" solutions with respect to the objectives. In the context of data privacy, these solutions embody the trade-off characteristics in the two objectives – minimum bias (for e.g., minimum NECD) and minimum information loss (for e.g., minimum NWP) – and are the points for analysis by a data publisher. Further, while multiple such trade-off solutions may exist, a data publisher is only interested in those that induce NECD and NWP values close to some preference levels. We ask the following questions in this regard.

1) What is an efficient solution in the multi-objective minimization of NECD and NWP?
2) Can such a solution be obtained by minimizing a scalar function that also incorporates preferences on NECD and NWP?
3) What guarantees that a minimum of the scalar function will be an efficient solution of the multi-objective problem?
4) Will it be possible to generate different efficient solutions by minimizing the scalar function?
5) Can we find a scalar function whose minimum is an efficient solution in the neighborhood of the data publisher's preferred NECD and NWP values?

The answer to the first question is grounded in the dominance based comparison of points in a multi-objective space. We provide the definition of a trade-off solution in this space using the principle of dominance, also called an *efficient point*. The second question is answered by introducing the concept of *scalar achievement functions* that combine the two objectives into one and take the data publisher preferences as one of its parameters. The issues raised in the third and fourth questions are resolved by enforcing the *strictly order preserving* and *strictly order representing* properties in the scalar function. Finally, such a function will be formulated in the next section as an answer to the fifth question.

Let $\mathcal{F}$ be the set of privacy preserving generalizations given the privacy constraint $\mathcal{P}_{\mathcal{CON}}$. In other words, all generalizations in $\mathcal{F}$ satisfy the privacy constraint $\mathcal{P}_{\mathcal{CON}}$. A generic privacy constraint is considered in order to emphasize that this approach is not limited to k–anonymity alone. Other models such as ℓ–diversity or t–closeness may as well be used to specify $\mathcal{P}_{\mathcal{CON}}$. Further, the following discussion is free from any intrinsic characteristic of the privacy constraint, other than the fact that the set of generalizations considered (the set $\mathcal{F}$) satisfy the constraint. We shall later see in Section 7 how the search algorithm stays focused on this set. Due to the same reason, the privacy guarantees provided by a resulting generalization will also be same as that provided by the underlying privacy model.

Let $\Delta : \mathcal{F} \to \mathbb{R}$ be a privacy bias function that assigns a privacy preserving generalization a real number signifying the privacy bias induced by it. Similarly, let $\Pi : \mathcal{F} \to \mathbb{R}$ be an information loss function signifying the amount of information lost due to a privacy preserving generalization. NECD and NWP are examples of Δ and Π respectively.

Consider the set of points in $Q = \{(\delta, \pi) | \delta = \Delta(F), \pi = \Pi(F), F \in \mathcal{F}\}$. The set Q contains the points signifying the bias and information loss for each possible privacy preserving generalization and shall be called the *efficiency space*. Hence, each point in Q can be associated with a privacy preserving generalization in $\mathcal{F}$. A partial order can be imposed on the points in Q as follows.

Definition 2 (Dominance). $q_1 = (\delta_1, \pi_1)$ *weakly dominates* $q_2 = (\delta_2, \pi_2)$, *denoted by* $q_1 \prec_W q_2$, *iff* $\delta_1 < \delta_2$ *and* $\pi_1 < \pi_2$. *Further,* $q_1 = (\delta_1, \pi_1)$ *strongly dominates* $q_2 = (\delta_2, \pi_2) \neq q_1$, *denoted by* $q_1 \prec_S q_2$, *iff* $\delta_1 \leq \delta_2$ *and* $\pi_1 \leq \pi_2$.

A privacy preserving generalization can then be characterized in terms of its ability to dominate other generalizations.

Definition 3 (Efficient Point). $q^* \in Q$ *is called a strongly efficient point of* Q *iff* $\nexists q_0 \neq q^* \in Q$ *such that* $q_0 \prec_S q^*$. q^* *is called weakly efficient iff* $\nexists q_0 \in Q$ *such that* $q_0 \prec_W q^*$.

The set of all strongly and weakly efficient points of Q is denoted by $\mathcal{E}_S$ and $\mathcal{E}_W$ respectively. In the context of the optimization problem at hand, efficient points correspond to privacy preserving generalizations that induce a level of bias and information loss that cannot be reduced simultaneously by another generalization. Using weak efficiency can result in points that are equal in at least one objective compared to other weakly efficient points. Strongly efficient points demonstrate a trade-off in both objectives. Strong efficiency implicitly implies weak efficiency.

Example 2 *Consider the data set shown in Table 1 (left). Assume that the employee code (denoted as 'nnnnn') can be generalized progressively by removing the last four digits from right to left one at a time. The removed digits are denoted by an asterisk. The only generalization allowed for the salary class is to merge class 1 and 2 (denoted by 'C12/3'), otherwise it must stay in an ungeneralized form (denoted by 'C1/2/3'). Let* $\mathcal{P}_{\mathcal{CON}}$ *be 2–anonymity. Table 2 shows the NECD and NWP values corresponding to the eight possible generalizations satisfying 2–anonymity. These eight generalizations form the set* $\mathcal{F}$ *and the corresponding NECD and NWP pairs form the set* Q. $\mathcal{G}_1$ *and* $\mathcal{G}_2$ *are the two strongly efficient points in this case.* $\mathcal{G}_3$, $\mathcal{G}_4$, $\mathcal{G}_6$, $\mathcal{G}_7$ *and* $\mathcal{G}_8$ *are weakly efficient.* $\mathcal{G}_5$ *is neither weakly nor strongly efficient since* $\mathcal{G}_2$ *weakly dominates and* $\mathcal{G}_1$ *strongly dominates* $\mathcal{G}_5$.

Multiple techniques exist to solve a vector optimization problem. These techniques may focus on approximating the set of efficient points using ϵ–*dominance* [23],

TABLE 2
Efficiency and **ach** values for different generalizations of the data set in Table 1.
$\overline{q_1} = (0.3, 0.08); \overline{q_2} = (0.2, 0.15); \overline{q_3} = (0.15, 0.05); \overline{q_4} = (0.25, 0.05); \overline{q_5} = (0.1, 0.5)$. $w_{emp} = 0.3; w_{sal} = 0.7$.

	Generalization	$NECD$	NWP	k	Efficiency	$s(\cdot, \overline{q_1})$	$s(\cdot, \overline{q_2})$	$s(\cdot, \overline{q_3})$	$s(\cdot, \overline{q_4})$	$s(\cdot, \overline{q_5})$	$pref_{dev}$ from $\overline{q_5}$
$\mathcal{G}_1$	nnnn* + C1/2/3	0.27	0.07	2	strong	**0.06**	0.12	0.07	**0.06**	0.23	-0.26
$\mathcal{G}_2$	nnn** + C1/2/3	0.18	0.09	3	strong	0.07	**0.08**	**0.06**	0.07	**0.15**	**-0.33**
$\mathcal{G}_3$	nn*** + C1/2/3	0.18	0.15	3	weak	0.12	0.09	0.12	0.13	**0.15**	-0.26
$\mathcal{G}_4$	n**** + C1/2/3	0.18	0.30	3	weak	0.24	0.17	0.23	0.25	**0.15**	-0.12
$\mathcal{G}_5$	nnnn* + C12/3	0.27	0.27	2	-	0.22	0.16	0.20	0.23	0.23	-0.05
$\mathcal{G}_6$	nnn** + C12/3	0.18	0.29	3	weak	0.23	0.17	0.22	0.24	**0.15**	-0.13
$\mathcal{G}_7$	nn*** + C12/3	0.18	0.36	3	weak	0.28	0.20	0.27	0.30	**0.15**	-0.06
$\mathcal{G}_8$	n**** + C12/3	0.18	0.50	5	weak	0.40	0.29	0.38	0.42	**0.15**	*0.09*

[24], [25] or concentrate on parts of the set using *reference points* [26], [27], [28]. A standard approach in the latter category is to associate a scalar problem to the original vector problem and generate efficient points by a single objective optimization of the scalar function [29], [30]. A typical scalarizing function $s : \mathbb{R}^2 \to \mathbb{R}$ defined over $\mathbb{R}^2$ depends on an arbitrary parameter $\overline{q} \in \mathbb{R}^2$, the functional form being denoted as $s(\cdot, \overline{q})$. Such functions must possess certain properties so that a minimum of the function can imply an efficient point of Q and vice versa.

Definition 4 (Strictly Order Preserving). *Let $s : \mathbb{R}^2 \to \mathbb{R}$ be a scalarizing function with respect to $\overline{q} \in \mathbb{R}^2$. s is strictly order preserving in Q iff*

P1. $\forall q_1, q_2 \in Q, q_1 \prec_S q_2 \Rightarrow s(q_1, \overline{q}) \leq s(q_2, \overline{q})$
P2. $\forall q_1, q_2 \in Q, q_1 \prec_W q_2 \Rightarrow s(q_1, \overline{q}) < s(q_2, \overline{q})$.

These two properties enforce the monotonicity requirements on s so that a minimum of the function is an efficient point of Q. This provides a sufficient optimality condition. The function s must satisfy another property so that any efficient solution of Q can be obtained by minimization of s parametrized by $\overline{q}$.

Definition 5 (Strictly Order Representing). *Let $s : \mathbb{R}^2 \to \mathbb{R}$ be a scalarizing function with respect to $\overline{q} \in \mathbb{R}^2$. s is strictly order representing in Q iff*

R1. $\forall q_0 \in S_0 = \{q \in Q | s(q, \overline{q}) \leq w_c\}$, we also have $q_0 \prec_S \overline{q}$, where $w_c = s(\overline{q}, \overline{q})$
R2. $\forall q_0 \in S_0' = \{q \in Q | s(q, \overline{q}) < w_c\}$, we also have $q_0 \prec_W \overline{q}$.

The strictly order representing properties help us prove that each efficient solution of Q has a corresponding value of the parameter $\overline{q}$ so that the minimum of s is the efficient solution. The scalarization thus provides the necessary optimality conditions. The following propositions provide a characterization of the efficient points of Q in terms of the solutions to a strictly order preserving and strictly order representing function s. The proofs are not presented for their overall simplicity.

Proposition 1. If q^* is a global minimum point of $s(q, \overline{q})$ on Q, then $q^* \in \mathcal{E}_W$.

Proposition 2. If q^* is a unique global minimum point of $s(q, \overline{q})$ on Q, then $q^* \in \mathcal{E}_S$.

Proposition 3. If $q^* \in \mathcal{E}_W$, then q^* is a global minimum of $s(q, \overline{q})$ on Q for $\overline{q} = q^*$.

Proposition 4. If $q^* \in \mathcal{E}_S$, then q^* is a unique global minimum of $s(q, \overline{q})$ on Q for $\overline{q} = q^*$.

A global minimum of the scalar achievement function should not be confused as a global minimum in the NWP vs. NECD objective space. In fact, a global minimum is not defined in a multi-objective case. Following Propositions 1 and 2, a global minimum of $s(\cdot, \overline{q})$ (a single objective function) only implies that the resulting solution is an efficient point of the two-objective space. Further, by Propositions 3 and 4, one can arrive at different efficient points by modifying the parameter $\overline{q}$ in the achievement function.

Example 3 *With reference to Table 2, Propositions 1 and 2 guarantee that the global minimum of a strictly order preserving and strictly order representing function will never appear at generalization $\mathcal{G}_5$. Further, if the parameter $\overline{q}$ of the function is set to the corresponding NECD and NWP values of a generalization, then the global minimum is guaranteed to appear at that generalization (Propositions 3 and 4). Hence, every efficient point is reachable by the function.*

Efficient points are popularly known as skylines in database query processing [31]. Skyline queries return data points such that no point in the result set is dominated by any other point in the database in the context of the queried attributes. For instance, in a HOTELS database with PRICE and DISTANCE attributes, searching for hotels with both minimum price and minimum distance may result in an empty result set. Skyline queries here retrieve data points whose price and distance values cannot be improved simultaneously by any other existing point in the database. An alternative to such queries is the use of utility functions where individual scores with respect to the queried attributes are combined to determine the most relevant results to the query. However, a user may never see certain results when using certain forms of utility functions (say, a weighted aggregation of individual scores). The scalar achievement function discussed here does not have this

drawback by virtue of the strictly order preserving and strictly order representing properties.

Note that minimal points of a scalarizing function should not be considered as the final solution of choice by the data publisher. They are rather used to locally approximate the preferences of the data publisher with the guarantee that the generated solutions are efficient. Data publisher preferences are typically embodied in the parameter $\bar{q}$ of the function (also called a *preference point*, *reference objective* or *aspiration level*) with the idea that an efficient solution minimizing some sort of distance from $\bar{q}$ is sought. This is followed by an interaction with the data publisher to inquire if the reported solution is satisfactory. Therefore, in this framework, an optimal solution from the perspective of the data publisher is not the generalization that results in minimum information loss (as assumed in existing works), but the generalization that meets the preference levels in the best possible way. The next section dwells into the formulation of a scalar achievement function that integrates these requirements.

5 SCALARIZING BIAS AND LOSS

A typical constrained optimization problem explored in disclosure control is to find an anonymized version of a data set, or effectively a set of generalizations resulting in the anonymized version, that induce minimum information loss subject to the constraint that the anonymized data set is k–anonymous. Given the NP-hard nature of the problem [14], heuristic based approaches in this context progressively increase the amount of generalization for the attributes until the k–anonymity constraint is satisfied [4], [5], [12]. The anonymized data set at this point is assumed to incur minimum information loss. These approaches have two major drawbacks.

First, the information loss metric is assumed to have a monotonic relationship with the amount of generalization. In other words, as more generalization is performed (no matter for which attribute), the information loss increases. Only under this assumption can one claim that by performing generalization only to the extent necessary to satisfy the k–anonymity constraint, we shall also be minimizing the information loss. However, the assumption is not valid when all attributes do not carry the same significance.

Second, existing approaches do not take into account any preference specified on information loss. There are some successful attempts to obtain all possible k–anonymized versions of a data set [8], [11], out of which the optimal one can be chosen based on preference criteria. Nonetheless, the set of solutions obtained with such an approach still remains exponentially large, making the search for an optimal choice equally difficult to perform. The issue of privacy bias remains unexplored in all these attempts.

The objective behind the scalarization of bias and utility is to arrive at privacy preserving generalizations that correspond to efficient solutions that are close to a reference objective. The reference objective is a point that signifies a tolerable level of bias and information loss to the data publisher. Depending on whether the reference objective lies inside or outside the efficiency space, the specified bias and information loss constraint may or may not be satisfied. If solutions better than the reference objective exist, then an efficient solution as far as possible from the reference point will provide the best possible improvements beyond the aspirations of the data publisher. On the other hand, if the reference objective is unachievable, then the data publisher can at best have a solution which is efficient and closest to the reference point.

We seek a scalarizing function that embeds these two requirements. In addition, the strictly order preserving and strictly order representing properties must be satisfied so that efficient points can be generated by a minimization of the scalar function. We present here one possible formulation for such a function.

Let $q_{ideal} = (\delta_{ideal}, \pi_{ideal})$ denote the ideal point in $\mathbb{R}^2$, the components of which are obtained by individual minimization of the bias and loss functions, i.e. $\delta_{ideal} = \min \Delta(\cdot)$ and $\pi_{ideal} = \min \Pi(\cdot)$. If a unique $F \in \mathcal{F}$ minimizes both the bias and loss functions, then q_{ideal} is the optimal solution. However, under the presence of trade-off behavior in the two functions, such a generalization will not exist. Hence, the ideal point is of theoretical importance only. For most cases in data privacy, the ideal point is the point $(0,0)$. Next, let $q_{utp} = (\delta_{utp}, \pi_{utp})$ be an utopian point computed as $\delta_{utp} = \delta_{ideal} - \epsilon_\delta$ and $\pi_{utp} = \pi_{ideal} - \epsilon_\pi$ where ϵ_δ and ϵ_π are small positive numbers. The scalarizing function **ach** $:= s(q, \bar{q})$ for $q = (\delta, \pi) \in \mathbb{R}^2$ and $\bar{q} = (\bar{\delta}, \bar{\pi}) \in \mathbb{R}^2$ is then formulated as

$$s(q, \bar{q}) = \max \ [w(\delta - \delta_{utp}), (1 - w)(\pi - \pi_{utp})]$$

$$\text{where } w = \left[\frac{1}{\bar{\delta} - \delta_{utp}} \right] \bigg/ \left[\frac{1}{\bar{\delta} - \delta_{utp}} + \frac{1}{\bar{\pi} - \pi_{utp}} \right].$$

This scalarization of bias and loss provides a maximal over-achievement of the objectives if the reference point is feasible. Otherwise, the function provides a minimal underachievement. The parameter w allows us to vary the weights on the two objectives, thereby providing a mechanism to explore the neighborhood of a solution. The precise impact of w is discussed in Section 6.

Example 4 *Consider the preference point $\bar{q_1} = (0.3, 0.08)$ in Table 2. The minimum of the **ach** function in this case appears at $\mathcal{G}_1$ since the corresponding NECD and NWP values of 0.27 and 0.07 can over-achieve the preferred ones. However, $\mathcal{G}_1$ cannot provide the same improvements for the preference point $\bar{q_2} = (0.2, 0.15)$. The minimum in this case appears at $\mathcal{G}_2$ since it can provide the best over-achievement in the two objectives. $\bar{q_3} = (0.15, 0.05)$ and $\bar{q_4} = (0.25, 0.05)$ are infeasible preferences. The minima here are obtained at the*

generalizations that produce minimal underachievement, i.e. $\mathcal{G}_2$ *and* $\mathcal{G}_1$ *respectively. Further, the minimum point in all cases is also an efficient point in the NECD vs. NWP objective space. This is due to the strictly order preserving property of* **ach**.

Theorem 1 *The scalarizing function* **ach** *is strictly order preserving.*

Proof: To prove property P1, consider the distinct points $q_1 = (\delta_1, \pi_1), q_2 = (\delta_2, \pi_2) \in Q$ such that $q_1 \prec_S q_2$.

Hence, we have $\delta_1 \leq \delta_2$ and $\pi_1 \leq \pi_2$. Assuming that $\bar{q} > q_{utp}$ (a valid assumption since the reference point will at best be q_{ideal}), we have $0 < w < 1$, giving us $(1 - w) > 0$. We can thus obtain the following two relations.

$$w(\delta_1 - \delta_{utp}) \leq w(\delta_2 - \delta_{utp})$$
$$(1 - w)(\pi_1 - \pi_{utp}) \leq (1 - w)(\pi_2 - \pi_{utp})$$

Using the observation that $a \leq b$ and $c \leq d$ implies $\max(a, c) \leq \max(b, d)$, we obtain $s(q_1, \bar{q}) \leq s(q_2, \bar{q})$, thus proving property P1.

To prove property P2, let $q_1 \prec_W q_2$. We then have $\delta_1 < \delta_2$ and $\pi_1 < \pi_2$. The remainder of the proof follows in a manner similar to above, giving us $s(q_1, \bar{q}) < s(q_2, \bar{q})$. Thus **ach** is strictly order preserving. $\square$

Theorem 2 *The scalarizing function* **ach** *is strictly order representing for*

$$w_c = 1 / \left[\frac{1}{\bar{\delta} - \delta_{utp}} + \frac{1}{\bar{\pi} - \pi_{utp}} \right].$$

Proof: To prove property R1, let $q_1 = (\delta_1, \pi_1) \in S_0$. Hence $s(q_1, \bar{q}) \leq w_c = s(\bar{q}, \bar{q})$, i.e.

$$\max \left[w(\delta_1 - \delta_{utp}), (1 - w)(\pi_1 - \pi_{utp}) \right] \leq w_c.$$

Rewriting the expression in terms of w_c we get

$$\max \left[w_c \left(\frac{\delta_1 - \delta_{utp}}{\bar{\delta} - \delta_{utp}} \right), w_c \left(\frac{\pi_1 - \pi_{utp}}{\bar{\pi} - \pi_{utp}} \right) \right] \leq w_c$$

i.e. $w_c \left(\dfrac{\delta_1 - \delta_{utp}}{\bar{\delta} - \delta_{utp}} \right) \leq w_c$ and $w_c \left(\dfrac{\pi_1 - \pi_{utp}}{\bar{\pi} - \pi_{utp}} \right) \leq w_c$.

After simplification we get, $\bar{\delta} - \delta_1 \geq 0$ and $\bar{\pi} - \pi_1 \geq 0$. Hence, $q_1 \prec_S \bar{q}$. This proves property R1.

To prove property R2, let $q_1 \in S_0'$. Hence $s(q_1, \bar{q}) < w_c$. By proceeding in a manner as above we get $\bar{\delta} - \delta_1 > 0$ and $\bar{\pi} - \pi_1 > 0$. Therefore, $q_1 \prec_W \bar{q}$. Thus **ach** is strictly order representing. $\square$

Based on Propositions 1 and 2, minimization of **ach** (a minimax problem) over Q will therefore result in an efficient privacy preserving generalization. However, solutions to the minimax problem can return either weakly or strongly efficient solutions. Ideally, a strongly efficient solution is more desirable since weakly efficient solutions cannot guarantee that one of the objective values cannot be reduced by keeping the other constant. Strongly efficient solutions are indicated by a unique global minimum of **ach**. For the case when multiple global minima exists, we choose a solution that provides the maximum over-achievement or minimum under-achievement with respect to the reference objective, i.e. choose the solution $q = (\delta, \pi) \in Q_g$ with minimum *preference deviation*, $pref_{dev} = (\delta + \pi - \bar{\delta} - \bar{\pi})$, where $Q_g \subseteq Q$ is the set of global minima points of **ach**.

Example 5 *Consider the preference point* $\bar{q_5} = (0.1, 0.5)$ *in Table 2. Multiple minima points exist for* **ach** *in this case. These points provide the least underachievement in NECD while satisfying the NWP preference of 0.5. The minimum preference deviation is obtained in* $\mathcal{G}_2$ *as it best over-achieves the NWP preference.* $\mathcal{G}_2$ *is also the strongly efficient point in the set of minima.*

Introduction of user preferences distinguish our handling of efficient points compared to as in a skyline computation. A query such as SELECT * FROM HOTELS WHERE PRICE¡$100 AND DISTANCE¡1mi would essentially return an empty result set if no record matches the WHERE clause. However, the achievement function formulated here does not try to "strictly" meet the preferences, but attempts to "best" meet it. Hence, the result returned by the achievement function would be analogous to returning the skyline closest to the PRICE=500 and DISTANCE=1 record in the database. Being able to perform maximal over-achievement or minimal under-achievement is what distinguishes the achievement function used here from the typical notion of utility functions in databases.

Incorporating data publisher preferences in the optimization procedure can also potentially hinder minimality attacks [32]. Minimality attacks exploit the knowledge that most data generalization algorithms attempt to enforce the privacy requirement with as less generalization as possible (minimum information loss). Hence, most existing methods are prone to this attack. However, the **ach** function subjected to minimization in this work is parametrized by the data publisher preferences. Therefore, a minimum of this function is not necessarily the generalization that induces the minimal information loss. In other words, the extent of generalization performed is not guided by the minimality principle (as assumed in minimality attacks), but is rather dependent on what the data publisher specifies as tolerable information loss. It is possible that an adversary possesses the domain knowledge required to infer some of the data publisher preferences. A more extensive analysis is required to determine the efficacy of minimality attacks in this scenario, and how knowledge about data publisher preferences can be used by an adversary to launch such attacks.
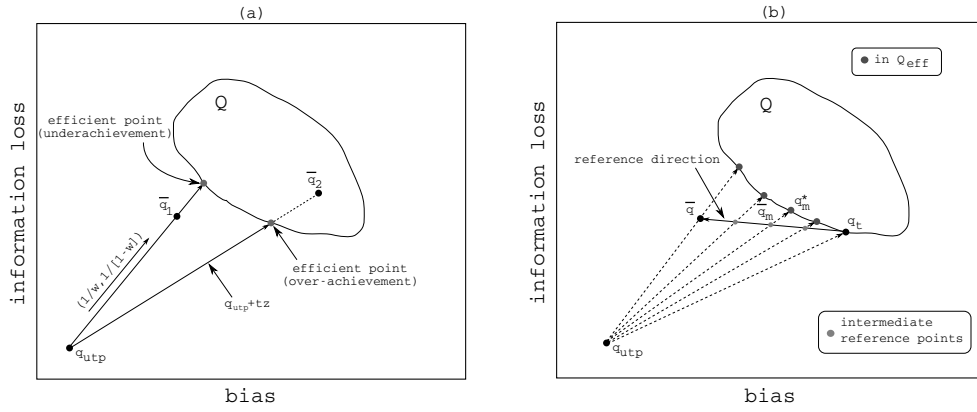
Fig. 1. (a) Geometrical interpretation of solution procedure to Chebyshev min-max problem. (b) Geometrical interpretation of reference direction approach.

6 REFERENCE DIRECTION APPROACH

While a solution obtained by minimizing the **ach** function is efficient, a data publisher may not find the reported solution satisfactory enough with respect to the reference objectives in mind. This is because the initial knowledge of the data publisher on the feasibility of a solution inducing the aspired bias and loss is limited. Once a solution is generated, the data publisher obtains additional knowledge on what levels of bias and loss are possible around the neighborhood of the preference point. This prompts for an interactive method that allows the data publisher to progressively explore efficient solutions until a satisfactory one is found. We adapt a method based on *reference directions* to facilitate this exploration [33], [34].

Definition 6 (Reference Direction). *Let q be a point in $\mathbb{R}^2$ and $\bar{q} \in \mathbb{R}^2$ be a preference point. The reference direction d is then defined as the vector $d = \bar{q} - q$.*

The basic approach here is to generate a number of efficient points along the projection of a reference direction on the efficiency space. This gives the data publisher an idea of the trade-off characteristics of solutions in the neighborhood where the data publisher's interest lies in the first place. If a satisfactory solution is found, then the process stops. Otherwise, one of the generated solutions is chosen to define a new reference direction for the next iteration of the process. This procedure of specifying a new search direction enables a data publisher to control how much deviation from the aspiration levels is tolerable in a certain objective. The reference direction approach computes what parameter value needs to be passed to the scalar achievement function to generate solutions in this new direction of interest. A new set of solutions is then generated and the process continues. By doing so, the data publisher can extensively explore the neighborhood surrounding the preference point until a solution is in agreement with the publisher's requirements. We provide below the basic steps of the method.

Step 1: Choose an initial point $q_1 \in \mathbb{R}^2$ and a preference point $\bar{q} > q_{utp}$. Set iteration counter $t = 1$.

Step 2: Compute reference direction $d_t = \bar{q} - q_t$. Choose a positive integer n as the number of efficient solutions to generate along the reference direction.

Step 3: Obtain the set of points $Q_{eff} = \{q_1^*, \ldots, q_n^*\}$ where q_m^* is the solution to the problem of minimizing **ach** $:= s(q, \bar{q_m})$ with $\bar{q_m} = q_t + (m/n)d_t$, for $m = 1, \ldots, n$.

Step 4: If a solution in Q_{eff} is satisfactory, stop; otherwise choose a point in the set as q_{t+1} to define a new reference direction.

Step 5: Set $t = t + 1$ and repeat from Step2.

Example 6 *Consider the data set in Table 1. Assume that the data publisher starts with a preference point $\bar{q} = (0.1, 0.1)$. Based on the extent of NECD and NWP values to explore, the publisher sets q_1 to $(1.0, 0.2)$ and $n = 10$. Multiple points will then be generated on the line segment joining $\bar{q}$ and q_1 at intervals of $(0.1, 0.01)$. Each such point is used as the parameter while minimizing the* **ach** *function, resulting in $Q_{eff} = \{\mathcal{G}_1, \mathcal{G}_2\}$. The data publisher at this point can choose either of $\mathcal{G}_1$ or $\mathcal{G}_2$ as the solution, or move ahead to the second iteration by selecting one of these points as q_2. However, no new solution will be found in this case.*

In order to understand the method from a geometrical perspective, we rewrite the minimax problem in the so-called *Chebyshev min-max* form. In this notation, $\min_{q \in Q}$ **ach** is written as a constrained minimization problem, given as

$$\text{minimize } \lambda \text{ subject to } \begin{cases} w(\delta - \delta_{utp}) & \leq \lambda \\ (1-w)(\pi - \pi_{utp}) & \leq \lambda \\ (\delta, \pi) & \in Q \end{cases}$$

The Chebyshev problem has the geometrical interpretation of a directional search starting at q_{utp} and progressing along the direction $z = (\frac{1}{w}, \frac{1}{1-w})$, i.e. the

search takes place on the straight line $q_{utp} + tz$ where t is a real positive parameter. Note that the reference point $\bar{q}$ lies on this straight line, as given by the point when $t = w_c$. Since a point on this line moves away from q_{utp} as t increases, thereby increasing λ, minimum value of λ is achieved with the lowest value of t that gives a point in Q. Refer to Fig. 1a. For the unattainable reference objective $\bar{q_1}$, this gives the point where the shifted reference point along the search direction first touches the efficiency space. On the other hand, for the reference objective $\bar{q_2}$, the search along the direction encounters a point in the efficiency space before encountering the reference point, i.e. $t < w_c$, thereby providing an over-achievement.

Note that the direction of search is decided by w, which in turn is parametrized by the reference objective. It is therefore possible to change the direction of search by providing different reference objectives. The interactive procedure does so by generating intermediate reference objectives $\bar{q_m}$ on the reference direction. Refer to Fig. 1b. At iteration t, a current solution q_t and the reference objective $\bar{q}$ defines the reference direction. Intermediate reference points are then generated along this direction. For each such point, a search is performed along the straight line joining q_{utp} and the reference point. In other words, solutions to the Chebyshev problem is found taking different search directions, each returning an efficient solution in the neighborhood of q_t and $\bar{q}$. For example, the solution q_m^* is found as the shifted intersection of the reference direction and straight line defined by the vector $(\frac{1}{w}, \frac{1}{1-w})$ with w being computed using the reference objective $\bar{q_m}$.

Theorem 3 *Let $q^* \in Q$ be an efficient solution. There exists a value of w such that $0 < w < 1$ and q^* is a unique global minimum of* **ach**.

Proof: Let $q^* = (\delta^*, \pi^*) \in Q$ be an efficient solution (weakly or strongly). To the contrary, let us assume that there exists no positive value of w such than q^* is a unique global minimum of **ach**.

Let us set $\bar{q} = q^*$. This gives us

$$w = \left[\frac{1}{\delta^* - \delta_{utp}} \right] \Big/ \left[\frac{1}{\delta^* - \delta_{utp}} + \frac{1}{\pi^* - \pi_{utp}} \right].$$

Note that w is greater than zero (since for any $q \in Q$, $q > q_{utp}$) and less than one. In the Chebyshev formulation of **ach**, q^* will be a feasible solution. In other words, we can find a λ^* such that $w(\delta^* - \delta_{utp}) \leq \lambda^*$ and $(1 - w)(\pi^* - \pi_{utp}) \leq \lambda^*$. After simplification, the least value of such a λ^* is found to be w_c.

However, since q^* is not a unique global minimum, there must exist another point $q_0 = (\delta_0, \pi_0) \in Q$ with a corresponding $\lambda = \lambda_0$ such that $\lambda^* \geq \lambda_0 \geq w(\delta_0 - \delta_{utp})$ and $\lambda^* \geq \lambda_0 \geq (1-w)(\pi_0 - \pi_{utp})$. By substituting λ^* with w_c, and the value of w in the two relations, we arrive at

the following two expressions: $\delta^* \geq \delta_0$ and $\pi^* \geq \pi_0$, or in other words, $q_0 \prec_S q^*$. Hence, q^* cannot be an efficient solution unless $q_0 = q^*$. Hence, q^* is a unique global minimum of the Chebyshev problem. Therefore, there exists a value $0 < w < 1$ such that q^* is a unique global minimum of **ach**. $\square$

Thus, **ach** can be used to generate any efficient point by varying its parameters. However, unlike a typical weighted sum approach, parameter specification in this approach is more intuitive to the data publisher, namely the aspiration levels of the publisher. Further, convergence in the interactive procedure is completely guided by the data publisher.

7 MINIMIZING THE ACHIEVEMENT FUNCTION

A crucial step in the reference direction approach is finding a privacy preserving generalization that minimizes **ach** (a solution q_m^* in Step 3). This optimization problem is repeatedly solved as part of the approach. With respect to $k-$anonymity and the metrics NWP and NECD, the problem can be stated in the following manner. Here, $\bar{\pi} = NWP_{pref}$ and $\bar{\delta} = NECD_{pref}$.

Optimization Problem (OP): *Given a data set* D, k_{pref}, NWP_{pref} *and* $NECD_{pref}$, *find the anonymized data set* D′ *that minimizes the achievement function* **ach** *subject to the constraint* $k_{pref} - \min(EC_{D'}) \leq 0$.

The optimization problem at hand is a constrained single objective problem. In this section we propose an approach based on evolutionary multi-objective optimization to find a solution to the problem. The method involves transforming the constraint into a separate objective giving us a bi-objective vector optimization problem [35]. The multi-objective variant of OP is formulated as follows.

Multi-Objective Optimization Problem (MOOP): *Given a data set* D, k_{pref}, NWP_{pref} *and* $NECD_{pref}$, *find the anonymized data set* D′ *that minimizes the achievement function* $f_1(D')$: **ach**(D′) *and the function* $f_2(D')$: $k_{pref} - \min(EC_{D'})$.

Solutions to the MOOP are characterized by the Pareto-dominance concept. Under such a characterization, an anonymized data set D′ found by the solution methodology is a non-dominated solution to the MOOP if it cannot find another solution D″ such that

- $f_1(D'') \leq f_1(D')$ and $f_2(D'') < f_2(D')$, or
- $f_1(D'') < f_1(D')$ and $f_2(D'') \leq f_2(D')$.

A direct and positive consequence of using this formulation is the exposure of higher effective k solutions, if any. Note that a solution to OP only needs to satisfy the constraint $k_{pref} - \min(EC_{D'}) \leq 0$. In the multi-objective formulation, the solutions undergo further filtering based on non-dominance – for two solutions with
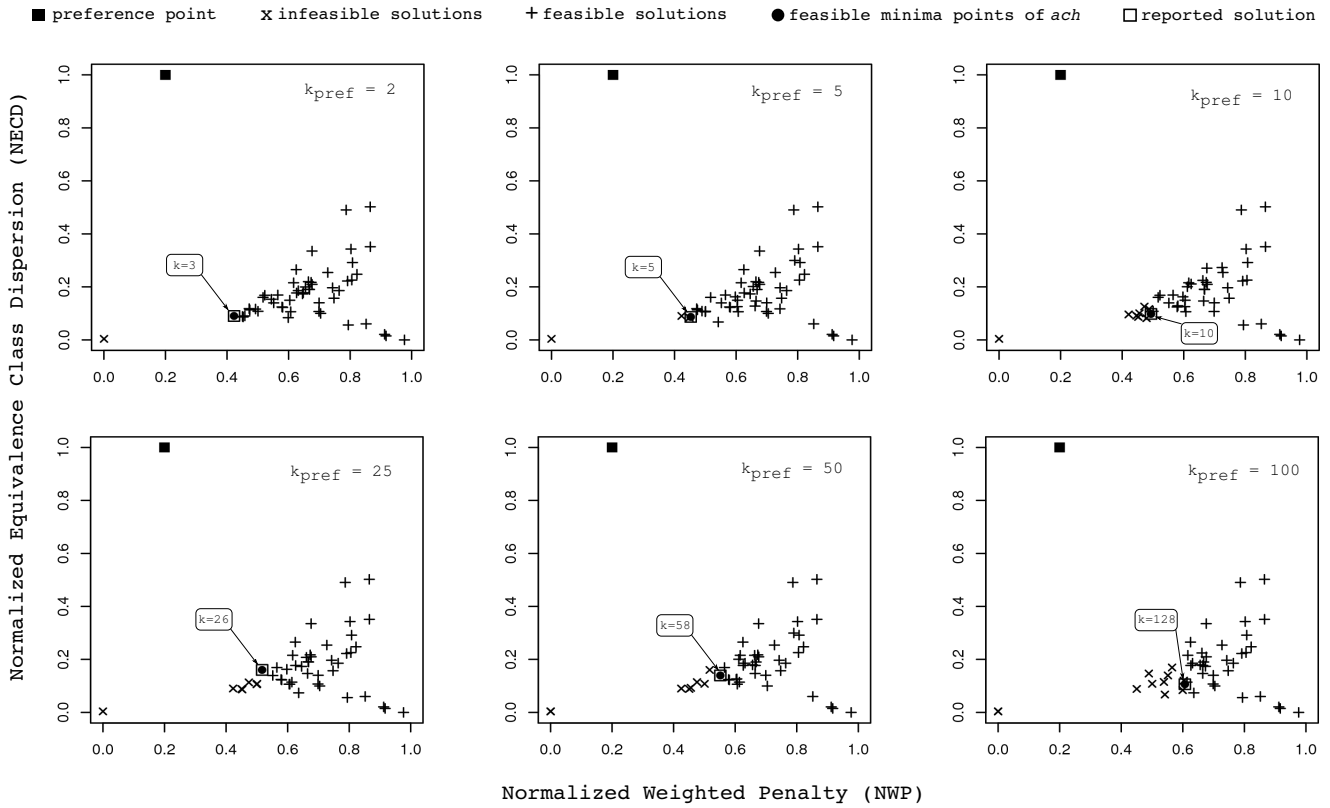
Fig. 2. NWP and NECD values of non-dominated solutions returned by NSGA-II for different k_{pref} (effective k value obtained is highlighted).

equal value of **ach**, the one with higher effective k (lower f_2) gets preference.

Example 7 *Recall the case when multiple minima are obtained with $\bar{q}_5 = (0.1, 0.5)$ as the preference point in Table 2. All points except $\mathcal{G}_8$ will be filtered out by the multi-objective optimizer since it provides a comparatively higher value for k. Given that a NWP value of 0.5 can be tolerated, $\mathcal{G}_8$ leverages it to improve the privacy level. Note that if the NWP preference is changed to say 0.2, then $\mathcal{G}_8$ will no longer be a minimum point of **ach**. In that case, there will be multiple solutions with same k and **ach** values. The preference deviation metric will then be used to choose a solution.*

We use the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) to obtain solutions to the MOOP. Readers are requested to refer to [36] for details on the algorithm.

7.1 Solution representation

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 [7]. 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 applied on the age attribute. 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 is now used to represent all possible generalizations for the attribute. The ith bit in this string is 0 if the ith and $(i + 1)^{th}$ intervals are supposed to be combined, otherwise 1. The granularization step can be skipped for attributes with a small domain and a defined ordering of the values. The individual encodings for each attribute are concatenated to create the overall encoding for the generalizations for all attributes.

7.2 Selection operator

While most components of NSGA-II are retained in the original form, a modification is proposed for the selection procedure in order to direct solutions towards the feasible region of OP [38]. NSGA-II employs a *crowded comparison operator* as part of its binary tournament selection scheme. This operator gives preference to solutions with lower ranks, assigned in accordance with a nondominated ranking procedure. Ties in ranks are broken using a *crowded distance* metric [36]. Our modification involves distinguishing between feasible ($f_2(D') \leq 0$) and infeasible ($f_2(D') > 0$) solutions of OP during the

TABLE 3
Attributes and domain size from the *adult census* data set.

Attribute	Domain Size
Age (*age*)	20 (granularity=5)
Work Class (*wkc*)	7
Education (*edc*)	16
Marital Status (*mst*)	7
Occupation (*occ*)	14
Race (*rac*)	5
Gender (*gen*)	2
Native Country (*ncy*)	41
Salary Class (*slc*)	2

selection procedure. The procedure is outlined as follows for two solutions x and y.

1) If both x and y are feasible, select based on crowded comparison operator.
2) If x is feasible and y is not, or vice versa, select the feasible one.
3) If both x and y are infeasible:
   a) select one with minimum f_2.
   b) if f_2 is equal, select one with minimum f_1.
   c) if f_1 is also equal, use crowding distance metric.
4) Use arbitrary selection for any unresolved case.

Using this selection procedure, we can initially direct the search towards the feasible region of OP and thereafter concentrate on exploring the trade-off characteristics. Note that the feasibility check here involves determining the size of the smallest equivalence class and whether it is at least k_{pref}. This check must be appropriately modified when considering a privacy model different from k–anonymity. This involves determining the value of the privacy parameter (e.g. k in k–anonymity or ℓ in ℓ–diversity) resulting from using a generalization, and checking whether it satisfies the minimum threshold set by the data publisher.

7.3 Solution to OP

Once the final non-dominated solution set $\mathcal{ND}$ to MOOP is obtained, the solution to OP is chosen as the point D′ such that D′ $= \underset{D''\in\mathcal{ND}_f}{\operatorname{argmin}} f_1(D'')$, where $\mathcal{ND}_f = \{D_i \in \mathcal{ND} | f_2(D_i) \leq 0\}$. The case of multiple such solutions is resolved using the preference deviation metric. Since the minimum of **ach** obtained in this manner is only justifiable w.r.t. ND_f, we shall say that D′ is an efficient solution only w.r.t. the non-dominated solutions generated by NSGA-II. Hence, although a global minimum of **ach** is guaranteed to be an efficient solution (w.r.t. the entire efficiency space) in theory, the NP-hard nature of the problem prevents us from claiming that the solution found is indeed a true minimum.

8 EMPIRICAL RESULTS

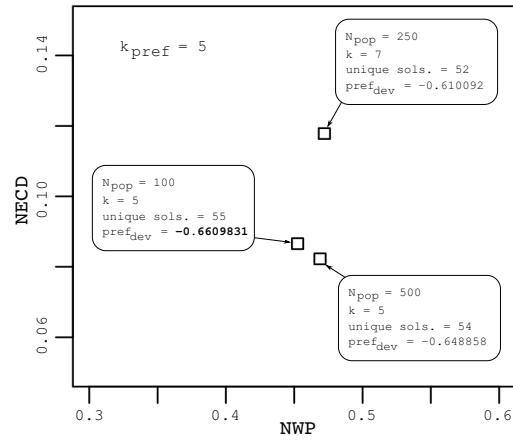We applied the NSGA-II methodology to the "adult.data" benchmark data set

Fig. 3. Impact of population size (N_{pop}) on solution quality for $k_{pref} = 5$ and preference point $NWP_{pref} = 0.2$; $NECD_{pref} = 1.0$.

(*ftp://ftp.ics.uci.edu/pub/machine-learning-databases/adult/*). All rows with missing values are removed to finally have a total of 30162 rows. The attributes used in this study along with their domain size are listed in Table 3. Recall that each attribute requires (*domain size - 1*) bits to represent all possible generalizations. This gives us a chromosome of length 105 representing a solution.

For NSGA-II, the population size N_{pop} is set at 100 with a maximum of 50,000 function evaluations. Binary crossover is performed on the entire chromosome with rate 0.8. Mutation is performed on the individual encodings of every attribute with a rate of 0.001. The modified selection operator is used for binary tournament selection. Weights on the attributes are assigned equally (1/9), unless otherwise stated.

8.1 Solution efficiency

Fig. 2 illustrates the NWP and NECD values of the non-dominated solutions returned by NSGA-II for different values of k_{pref}. The preference point of $NWP_{pref} = 0.2$ and $NECD_{pref} = 1.0$ is used in these experiments. Choosing a NECD preference of 1.0 effectively allows NSGA-II to look for low NWP solutions irrespective of the privacy bias they induce. As higher values of k_{pref} are used, the number of feasible solutions obtained decreases. This is likely to happen since the search space is known to be very dense for low values of k_{pref}, while solutions become rare as higher privacy requirements are enforced. Consequently, while reported solutions for $k_{pref} = 2, 5$ and 10 have an effective k close to k_{pref}, higher values are obtained for $k_{pref} = 25, 50$ and 100. However, higher information loss has to be sustained for stronger privacy requirements. A unique feasible minimum of **ach** is obtained in all the cases. In confirmation to our theoretical observation, the minimum point is a non-dominated point in the NWP vs. NECD objective space w.r.t other feasible solutions returned by NSGA-II. Further, the existence of solutions at effective k values
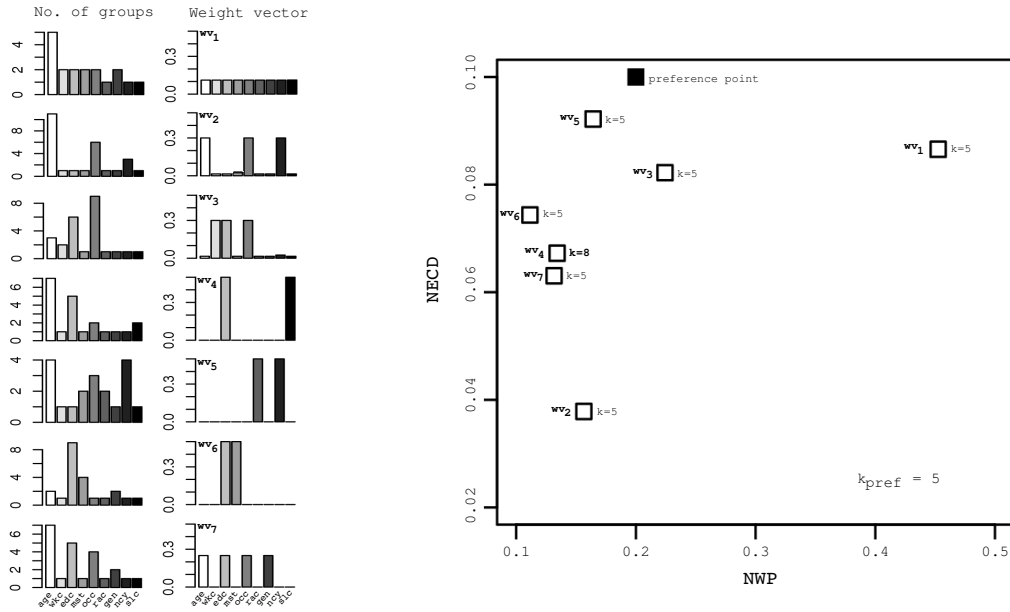
Fig. 4. Effect of the weight vector on attainable NWP and NECD. Also shown are the corresponding number of groupings in the attribute domains.

higher than k_{pref} (for example $k = 3$ for $k_{pref} = 2$) strengthens our claim that the optimal solution need not always have effective k value equal to k_{pref}. Once again, the concept of Pareto-dominance helps here in discovering these solutions.

8.2 Impact of population size

Fig. 3 shows the reported solutions for three different settings of population size, $N_{pop} = 100, 250$ and 500. Notice that increasing the population size, while keeping the number of function evaluations fixed, seem to have only marginal impact on the overall solution quality. Solutions are slightly less effective in terms of the preference deviation metric for larger population size, albeit there is no logical pattern in the behavior. Larger populations typically have the potential to explore more parts of the search space. However, the absence of an uniform distribution of solutions in the search space makes this exploration difficult. We also notice that the number of unique solutions obtained is similar irrespective of the population size used. Large populations have more duplicates that affect the convergence rate of the population. This is primarily due to the higher selective pressure of duplicate solutions, which limits the exploratory capabilities of the population. Small populations and higher number of iterations is a key element in solving this problem.

8.3 Effect of weight vector

Fig. 4 illustrates the solutions obtained for different assignment of weights to the attributes. As is evident from the solutions, the assignment of equal weights (wv_1) in this problem results in a much higher NWP and NECD.

Weight assignments impact the amount of generalization that may be performed for an attribute, which in turn influence the information content of the anonymized data set. Even when all attributes are equally important, higher weights can be assigned to attributes with larger domain sizes to retain as much information as possible. For example, while most solutions in the figure completely suppress (number of partitions=1) the "Native Country" attribute, assigning a higher weight to the attribute (as in wv_2 and wv_5) return solutions with more number of partitions. In general, NSGA-II is seemingly effective in generating solutions with higher number of partitions in accordance with the weight assignments.

8.4 Impact of bias preference

Fig. 5 illustrates the impact of setting the NECD preference value. A typical preference of 1.0 effectively means that any level of bias is acceptable. As a result, a solution only needs to perform as much generalization as is necessary to meet the feasibility constraint, assuming that the minimum value of NWP is attained at $k = k_{pref}$. Such a case happens with the weight vector wv_2. However, when the bias preference is dropped below 0.1, solutions are generated with higher NWP (although within the preference value of 0.2) and higher effective k. This happens because the method is now forced to explore solutions with more generalization in order to better meet the low bias preference. More generalization typically yield higher effective k. Notice that as the bias preference is lowered, the effective k increases. It is imperative to ask at this point why a bias preference of 1.0 should not be set for this problem since the best solution (with $k = 5$) is obtained with this setting. The answer lies in the trade-off characteristic of the solutions
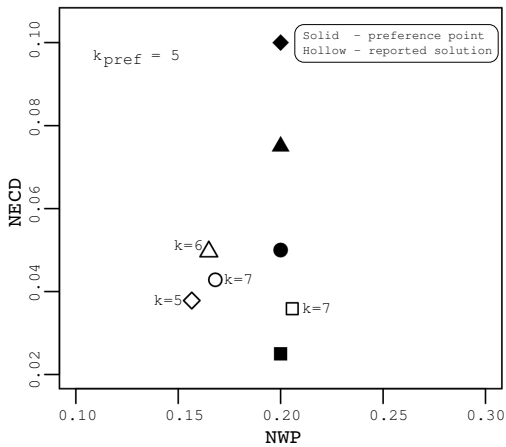
Fig. 5. Impact of bias preference on solutions obtained with weights wv_2.

between the level of privacy and NWP. Note that the $k = 6$ solution has higher NWP at the expense of slightly higher privacy level than the $k = 5$ solution. Since both solutions meet the NWP preference, the $k = 6$ solution is more preferable. In fact, given the four solutions in the figure and the NWP preference of 0.2, the $k = 7$ solution (one marked with a circle) is the solution of choice. This solution overachieves the preference criteria and provides better privacy than the $k = 5$ and $k = 6$ solutions.

8.5 Efficiency

Evolutionary algorithms often receive criticism for their high running time. NSGA-II takes around 15 minutes (on an Intel Core 2 Duo 2x2.83GHz machine with 2GB RAM) to complete the **ach** minimization problem on the test dataset. This can be reduced by temporarily storing evaluated points. The reference direction procedure could require multiple calls to this optimization routine depending on how well a solution meets the data publisher preferences, or how extensively the data publisher explores the neighborhood solutions. However, this is typically an offline problem. Further, evolutionary algorithms are inherently parallel and can easily be adapted to utilize the processing power of today's massively parallel systems [39], thereby significantly improving the run time.

9 CONCLUSIONS

In this paper, we explore the problem of privacy bias minimization along with data utility maximization in the space of data generalizations that satisfy a particular privacy property. In addition, we also emphasize that data publisher preferences are an important component in this optimization problem. As a possible solution methodology, we propose using scalarizing functions based on preferences of the data publisher to transform the vector optimization problem to a scalar one.

Minimization of the scalar function is performed by transforming the privacy constraint into a second objective, and then applying an evolutionary multi-objective algorithm. Moreover, a reference direction based interactive procedure iteratively uses this algorithm to help a data publisher explore efficient solutions until a satisfactory one is found. Results on a benchmark data set demonstrate the effectiveness of the evolutionary algorithm in finding solutions that best achieve the preferences of the data publisher. The method is also able to find higher effective k values depending on the weights assigned to different attributes.

It would be interesting to see how different notions of homogeneity in privacy levels can be used to define bias metrics, and what impact they have on the information preservation efficiency of a generalization. The efficacy of minimality attacks when generalizations are based on preferences is also a direction worth exploring.

REFERENCES

[1] P. Golle, "Revisiting the Uniqueness of Simple Demographics in the US Population," in *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, 2006, pp. 77–80.
[2] P. Samarati and L. Sweeney, "Generalizing Data to Provide Anonymity when Disclosing Information," in *Proceedings of the 17th ACM Symposium on Principles of Database Systems*, 1998, p. 188.
[3] L. Sweeney, "Achieving k–Anonymity Privacy Protection Using Generalization and Suppression," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 571–588, 2002.
[4] R. J. Bayardo and R. Agrawal, "Data Privacy Through Optimal k-Anonymization," in *Proceedings of the 21st International Conference on Data Engineering*, 2005, pp. 217–228.
[5] B. C. M. Fung, K. Wang, and P. S. Yu, "Top-Down Specialization for Information and Privacy Preservation," in *Proceedings of the 21st International Conference in Data Engineering*, 2005, pp. 205–216.
[6] 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.
[7] V. S. Iyengar, "Transforming Data to Satisfy Privacy Constraints," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 279–288.
[8] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-Domain k-Anonymity," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, 2005, pp. 49–60.
[9] ——, "Mondrian Multidimensional K-Anonymity," in *Proceedings of the 22nd International Conference in Data Engineering*, 2006, p. 25.
[10] G. Loukides and J. Shao, "Capturing Data Usefulness and Privacy Protection in k-Anonymisation," in *Proceedings of the 2007 ACM Symposium on Applied Computing*, 2007, pp. 370–374.
[11] P. Samarati, "Protecting Respondents' Identities in Microdata Release," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.
[12] K. Wang, P. Yu, and S. Chakraborty, "Bottom-Up Generalization: A Data Mining Solution to Privacy Protection," in *Proceedings of the 4th IEEE International Conference on Data Mining*, 2004, pp. 249–256.
[13] R. Dewri, I. Ray, I. Ray, and D. Whitley, "On the Comparison of Microdata Disclosure Control Algorithms," in *12th International Conference on Extending Database Technology*, 2009, pp. 240–251.

[14] A. Meyerson and R. Williams, "On the Complexity of Optimal k-Anonymity," in *Proceedings of the 23rd ACM Symposium on the Principles of Database Systems*, 2004, pp. 223–228.

[15] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "ℓ–Diversity: Privacy Beyond k–Anonymity," in *Proceedings of the 22nd International Conference on Data Engineering*, 2006, p. 24.

[16] N. Li, T. Li, and S. Venkatasubramanian, "t–Closeness: Privacy Beyond k–Anonymity and ℓ–Diversity," in *Proceedings of the 23rd International Conference on Data Engineering*, 2007, pp. 106–115.

[17] X. Xiao and Y. Tao, "Anatomy: Simple and Effective Privacy Preservation," in *Proceedings of the 32nd International Conference on Very Large Data Bases*, 2006, pp. 139–150.

[18] J. Li, R. C. Wong, A. W. Fu, and J. Pei, "Achieving k-Anonymity by Clustering in Attribute Hierarchical Structures," in *Proceesing of 8th International Conference on Data Warehousing and Knowledge Discovery*, 2006, pp. 405–416.

[19] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu, "Aggregate Query Answering on Anonymized Tables," in *Proceedings of the 23rd International Conference on Data Engineering*, 2007, pp. 116–125.

[20] R. Dewri, I. Ray, I. Ray, and D. Whitley, "On the Optimal Selection of k in the k-Anonymity Problem," in *Proceedings of the 24th International Conference on Data Engineering*, 2008, pp. 1364–1366.

[21] Z. Huang and W. Du, "OptRR: Optimizing Randomized Response Schemes for Privacy-Preserving Data Mining," in *Proceedings of the 24th International Conference on Data Engineering*, 2008, pp. 705–714.

[22] T. Li and N. Li, "On the Tradeoff Between Privacy and Utility in Data Publishing," in *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2009, pp. 517–526.

[23] Vassilvitskii, S. and Yannakakis, M., "Efficiently Computing Succinct Trade-off Curves," *Theoretical Computer Science*, vol. 348, no. 2, pp. 334–356, 2005.

[24] I. Diakonikolas and M. Yannakakis, "Small Approximate Pareto Sets for Bi-objective Shortest Paths and Other Problems," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2007, pp. 74–88.

[25] H. Ackermann, A. Newman, H. Röglin, and B. Vöcking, "Decision-making Based on Approximate and Smoothed Pareto Curve," *Theoretical Computer Science*, vol. 378, no. 3, pp. 253–270, 2007.

[26] A. P. Wierzbicki, "The Use of Reference Objectives in Multiobjective Optimization," in *Multiple Criteria Decision Making Theory and Applications*, 1980, pp. 468–486.

[27] Y.Yun, H. Nakayama, and M. Yoon, "Sequential Approximation Method in Multi-objective Optimization Using Aspiration Level Approach," *Evolutionary Multi-Criterion Optimization*, vol. 4403, pp. 317–329, 2007.

[28] M. Luque, K. Miettinen, P. Eskelinen, and F. Ruiz, "Incorporating Preference Information in Interactive Reference Point Methods for Multiobjective Optimization," *Omega*, vol. 37, no. 2, pp. 450–462, 2009.

[29] K. Miettinen and M. M. Mäkelä, "On Scalarizing Functions in Multiobjective Optimization," *OR Spectrum*, vol. 24, no. 2, pp. 193–213, 2002.

[30] E. Miglierina and E. Molho, "Scalarization and Stability in Vector Optimization," *Journal of Optimization Theory and Applications*, vol. 114, no. 3, pp. 657–670, 2002.

[31] S. Börzsönyi, D. Kossmann, and K. Stocker, "The Skyline Operator," in *Proceedings of the 17th International Conference on Data Engineering*, 2001, pp. 421–430.

[32] R. C. Wong, A. W. Fu, K. Wang, and J. Pei, "Minimality Attack in Privacy Preserving Data Publishing," in *Proceedings of the 33rd International Conference on Very Large Data Bases*, 2007, pp. 543–554.

[33] K. Miettinen and L. Kirilov, "Interactive Reference Direction Approach Using Implicit Parametrization for Nonlinear Multiobjective Optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 13, no. 2-3, pp. 115–123, 2005.

[34] K. Deb and A. Kumar, "Interactive Evolutionary Multi-objective Optimization and Decision-Making Using Reference Direction Method," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2007, pp. 781–788.

[35] C. A. C. Coello, "An Updated Survey of GA-Based Multiobjective Optimization Techniques," *ACM Computing Surveys*, vol. 32, no. 2, pp. 109–143, 2000.

[36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

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

[38] R. Dewri, I. Ray, I. Ray, and D. Whitley, "A Multi-Objective Approach to Data Sharing with Privacy Constraints and Preference Based Objectives," in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, 2009, pp. 1499–1506.

[39] E. Alba and M. Tomassini, "Parallelism and Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 443–462, 2002.

Rinku Dewri is an Assistant Professor in the Computer Science Department at University of Denver. He obtained his Ph.D. in Computer Science from Colorado State University. His research interests are in the area of information security and privacy, risk management, data management and multi-criteria decision making. He is a member of the IEEE and the ACM.

Indrajit Ray is an Associate Professor in the Computer Science Department at Colorado State University. Prior to that, he worked as an Assistant Professor in the Computer and Information Science Department at the University of Michigan-Dearborn. His main research interests are in the areas of computer and network security, database security, security and trust models, privacy and computer forensics. He is on the editorial board of several journals, and has served or is serving on the program committees of a number of international conferences. He is a member of IEEE, IEEE CS, ACM, ACM SIGSAC, IFIP WG 11.3 and IFIP WG 11.9.

Indrakshi Ray is an Associate Professor in the Computer Science Department at Colorado State University. Prior to joining Colorado State, she was a faculty at the University of Michigan-Dearborn. She obtained her Ph.D. from George Mason University. Her research interests include security and privacy, database systems, e-commerce and formal methods in software engineering. She served as the Program Chair for SACMAT 2006 and IFIP WG 11.3 DBSEC 2003. She has also been a member of several program committees such as for EDBT, SACMAT, ACM CCS and EC-Web. She is a member of the IEEE and the ACM.

Darrell Whitley is Professor and Chair of the Computer Science Department at Colorado State University. He also currently serves as Chair of ACM SIGEVO. He was chair of the governing board for International Society for Genetic Algorithm from 1993 to 1997, and was Editor-in-Chief of the MIT Press journal Evolutionary Computation from 1997 to 2002.