Information Sciences 500 (2019) 203–216

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Solving bin-packing problems under privacy preservation: Possibilities and trade-offs

Rowan Hoogervorst[a,*], Yingqian Zhang[b], Gamze Tillem[c], Zekeriya Erkin[c], Sicco Verwer[c]

[a] *Erasmus School of Economics, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, the Netherlands*
[b] *Department of Industrial Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, the Netherlands*
[c] *Cyber Security Group, Department of Intelligent Systems, Delft University of Technology, 2628 CD Delft, the Netherlands*

ARTICLE INFO

Article history:
Received 16 April 2018
Revised 16 March 2019
Accepted 5 May 2019
Available online 22 May 2019

Keywords:
Data anonymization
Bin-packing
k-anonymity
Stochastic programming
Robust optimization

ABSTRACT

We investigate the trade-off between privacy and solution quality that occurs when a *k*-anonymized database is used as input to the bin-packing optimization problem. To investigate the impact of the chosen anonymization method on this trade-off, we consider two recoding methods for *k*-anonymity: full-domain generalization and partition-based single-dimensional recoding. To deal with the uncertainty created by anonymization in the bin-packing problem, we utilize stochastic programming and robust optimization methods. Our computational results show that the trade-off is strongly dependent on both the anonymization and optimization method. On the anonymization side, we see that using single dimensional recoding leads to significantly better solution quality than using full domain generalization. On the optimization side, we see that using stochastic programming, where we use the multiset of values in an equivalence class, considerably improves the solutions. While publishing these multisets makes the database more vulnerable to a table linkage attack, we argue that it is up to the data publisher to reason if such a loss of anonymization weighs up to the increase in optimization performance.

1. Introduction

In the last decades, many different methods have been proposed to preserve privacy in published databases. The two most common frameworks are that of recoding the data through generalization and/or suppression, and that of adding random noise to the data [28]. In both cases, the original data is published in a perturbed format to achieve anonymity. Hence, maintaining the utility of the anonymized data for further processing is a challenge.

Fields that already deal extensively with this challenge are those of data mining, machine learning, and statistics. In [11] for instance, specialized algorithms have been developed that aim to minimize the effects of the perturbations on query counts. Another example is [16], where a privacy preserving transformation is proposed that maintains distances between data rows. Typically, studies are limited to the effect of data perturbation on data mining measures such as frequency counts, statistics, distances, and predictive accuracy. Surprisingly, the effect of data perturbation on subsequent decision making has

[*] Corresponding author.
E-mail addresses: hoogervorst@ese.eur.nl (R. Hoogervorst), yqzhang@tue.nl (Y. Zhang), G.Tillem@tudelft.nl (G. Tillem), Z.Erkin@tudelft.nl (Z. Erkin), S.E.Verwer@tudelft.nl (S. Verwer).

https://doi.org/10.1016/j.ins.2019.05.011
0020-0255/© 2019 Published by Elsevier Inc.

been largely ignored in the current literature. In this paper, we fill this gap by presenting a first study on the effect that data transformations for *k*-anonymity have on the decision quality of optimization problems.

The topic of privacy preservation is by itself not new to the field of optimization, where different works have focused on finding optimization methods that provide solutions which are privacy preserving, see, e.g., [23,43]. Our approach differs substantially from the path taken in these earlier works. Instead of assuming that the input data to the optimization constitutes the true data and that we want to obtain a privacy preserving solution, we assume that the input data of the optimization problem already satisfies a formal privacy criterion. The challenge in optimization is then to deal with the uncertainty about the true input data that is created by the anonymization. This setting of using privacy preserving data as input to optimization offers interesting challenges for both optimization and data publishing parties.

We study the effect of data perturbation on an optimization problem that is commonly encountered in operations research: bin-packing. This problem for example models loading containers onto the different available transport modes such as ships, rail, or trucks. Crucial in such cargo-loading problems is that no transport vehicle should be overloaded (the bin size). A container's weight is thus very important information for minimizing the number of required vehicles. However, it is also a very identifying attribute for attackers that seek to find a specific container. Hence, this example illustrates the privacy-utility trade-off between the anonymity of the data and the use of the data in optimization that we investigate in this paper.

The introduced data perturbations correspond to the familiar case of data uncertainty in optimization, for which many specialized frameworks exist. Prime examples are the multitude of robust [5] and stochastic [10] optimization methods, which often rely on the implementation of chance constraints that limit the probability of obtaining an infeasible solution. An interesting new perspective when considering data anonymization is that the level of data uncertainty is strongly influenced by the chosen method of anonymization and the corresponding output statistic that is provided with the anonymization. We show how this knowledge can be utilized by stochastic and robust optimization methods in order to significantly increase the solution quality.

The contributions of our work are twofold. First of all, we study the extend to which the chosen method of recoding for *k*-anonymity affects the solution quality when this *k*-anonymized database is used as input to a bin-packing optimization model. Second, we show how the chosen optimization method affects the final solution quality when handling *k*-anonymized input data and propose two novel optimization methods for this setting. Although we study only one optimization problem, and only a single anonymization framework, our study brings to light several important observations:

- We show that the chosen method of recoding in case of *k*-anonymity has a significant effect on solution quality, motivating the choice for (near-)exact methods to achieve privacy preservation.
- We show that the main challenge in the use of *k*-anonymized datasets in an optimization settings lies in achieving feasibility, where the penalty to the solution quality to achieve feasibility is substantial.
- We show that a novel anonymization-aware stochastic optimization method, that uses the exact multiset of perturbed values over the aggregation ranges to reduce the effect of data uncertainty, is better able to balance feasibility with the obtained objective value.

With our work, we hope to open up a new line of research investigating this trade-off between optimization and anonymization and the development of advanced methods for combining anonymization and optimization. Using the current state-of-the-art methods, it is very hard to reach acceptable levels of privacy preservation and optimization performance simultaneously when using *k*-anonymity as a privacy criterion.

2. Literature review

Our work on the use of anonymized data in optimization is strongly influenced by the advances of other researchers that have worked on integrating privacy preservation in the field of data mining. The idea of privacy-preserving data mining was introduced by Agarwal and Srikant [1] and Lindell and Pinkas [39]. In their work, the aim is to extract information from users' private data without having to reveal individual data items. Since then, a large number of privacy protection mechanisms have been proposed in [12,17,29,30,45,46]. These works can generally be divided into two categories: the ones using cryptographic tools [2,3,9,12,17,20,29,30,34,50] and the ones using perturbation based techniques [45,46].

The works using cryptographic tools mostly have a multi-party communication setting, either using secret sharing [29,30,39] or homomorphic encryption [2,3,9,17,20,50] to preserve privacy. While such methods achieve solutions of similar utility to those in the non-private setting, they do so at the cost of increased computation time. In particular, such methods generally operate on ciphertexts of 2048 bits opposed to the typical values of 32 or 64 bits considered in non-private applications [35]. It is thus not straightforward to use these methods efficiently. This already holds for simple greedy algorithms such as the K-means clustering algorithm and it will thus be very challenging if not impossible to apply them for solving the typically NP-hard problems considered in optimization. We are aware of preliminary work on using such methods for optimization, where [14] develop such methods for solving simple linear inequalities using an adapted simplex algorithm, but the sizes of problems these methods can handle are still very small.

The perturbation based privacy mechanisms are mostly based on the idea of transforming the data (e.g. by generalization) or on adding a random perturbation to the data. Opposed to the cryptographic tools, these methods do lead to a loss of utility of the found solution when compared to the non-private setting, as any subsequent algorithms will only be able to use the perturbed data. However, they do generally allow for fast solution algorithms which are close to the algorithms used in the non-privacy preserving setting, explaining the widespread use of these methods in practice. As a result, a wide variety of such methods have been developed to output a privacy preserving database, including the popular methods of k-anonymity [48,52] and differential privacy [19].

Considering the loss of utility that is implied by such methods, numerous papers have focused on finding anonymization and data mining methods that minimize this utility loss [11,16,45,46]. The randomization approach taken by Oliviera and Zaiane [45,46] is very related to our study in the sense that they care about the performance of (clustering) algorithms after the data has been anonymized. Using geometric data transformations they obfuscate the data in such a manner that the distances between the individual data items remain intact, ensuring it is still useful for analysis using distance-based data mining methods. Similarly, we study when data anonymization results in data sets that are still useful for optimization.

Most recently, some work has been done on incorporating differential privacy into distributed allocation problems, where the focus is on how to randomly perturb coordination signals, in order to avoid revealing private information of the users [24–27]. In [27], the authors assume the private user information is encoded in its cost functions and study the problem of differentially private distributed convex optimization. In [25,26], the authors assume that the private user information is encoded in the individual constraints. For example, Hsu et al. [26] investigate how to hide the presence or absence of a single constraint in linear programs. Different from these existing work where only partial data are privacy preserved, our work assumes all inputs to the optimization problem are anonymized. Moreover, a different privacy preserving method, i.e., k-anonymity, is studied in this paper.

To solve the bin-packing problem with anonymized weights, we make use of techniques from stochastic programming [10] and robust optimization [5,6]. These methods take into account uncertainty in optimization problems based on respectively a probability distribution on the uncertainty and an uncertainty set. We show that we can choose the information that remains after anonymization in such a way that we are able to derive such distributions and sets. Alternatively to these mathematical programming techniques, one might use heuristic solution methods such as simulated annealing [31], sampling [42], and Monte Carlo based techniques [22]. However, it is non-trivial to effectively deal with this knowledge about the uncertainty using these heuristics. It could be an interesting topic for future work.

To the best knowledge of the authors, this is the first paper to consider the effect of applying k-anonymity to the input data of a combinatorial optimization problem. In this way, this paper aims to quantify the effect of k-anonymity preserving data transformations on solution quality and suggests novel optimization methods to limit the effect that these data transformations have on the final solution quality.

3. Problem formulation

In this section, we introduce the bin-packing optimization problem and the concept of k-anonymity. Moreover, we introduce the general framework that we consider for achieving anonymity and for optimizing the bin-packing problem with anonymized data.

3.1. Bin-packing

In the bin-packing optimization problem, we need to find an allocation of n items over n bins such that the minimum number of bins is used. Here, each item $j \in N$ has weight w_j and each bin $i \in N$ has capacity c, where $N = \{1, \dots, n\}$. If we introduce the decision variables

$$y_i = \begin{cases} 1 & \text{if bin } i \text{ is used,} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

$$x_{i,j} = \begin{cases} 1 & \text{if item } j \text{ is packed into bin } i, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

the formulation of the bin-packing problem as proposed by [41] is given by

$$\min \sum_{i \in N} y_i \tag{3}$$

$$\text{s.t.} \sum_{j \in N} w_j x_{i,j} \le c y_i \qquad \forall i \in N, \tag{4}$$

$$\sum_{i \in N} x_{i,j} = 1 \qquad \forall j \in N, \tag{5}$$

$$y_i, x_{i,j} \in \{0, 1\} \qquad \forall i, j \in N. \tag{6}$$

Table 1
An example of a possible dataset with the item weights as sole quasi-identifier.

id	1	2	3	4	5
w_j	81	81	83	87	88

The objective (3) is to minimize the number of used bins. Constraints (4) ensure that the capacity of each bin is respected, where items can only be placed in a bin if it is used. Constraints (5) ensure that each item is placed in a bin. Constraints (6) ensure that the decision variables are binary. To improve this formulation, which tends to perform relatively poor due to symmetry considerations, we add the valid inequalities and symmetry-breaking constraints as suggested by [18]. It will be this joint model that we will refer to as the bin-packing optimization model.

3.2. Anonymization

Optimization methods often operate on datasets that contain privacy-sensitive information. The input data may, for example, consist of medical data, where some of the variables provide information that may lead to the identification of individuals in the data. More generally, let us assume that the data is given by a table of rows and columns, where the rows represent the data entries and the columns (i.e., attributes) the different variables of interest in the table. The table can then be represented as $T(A_1, A_2, \dots, A_m)$, where A_i denotes the ith attribute. In privacy-preserving data publishing one often separates these attributes into sensitive attributes and quasi-identifiers [21]. The sensitive attribute defines the sensitive information in the dataset, while quasi-identifiers may be used to identify the individual to whom the sensitive attribute belongs.

In our application, we will assume that the bin weights are a quasi-identifier and may be used to identify the item they belong to. This is, for example, an issue in ports, where the weight of a certain container may be important information for smugglers to identify the location and means of transport of a container. Another application is present in hospitals, where for example the duration of an operation may lead to the identification of the individual that is operated. A small example of how our input data looks like is given by the dataset in Table 1, which contains the item weights as the sole quasi-identifier and an additional id that represents the sensitive attribute.

A crucial step for the data publisher in privately publishing a database is to pick a privacy criterion that should reflect on the sensitivity and further use of the data. As achieving an absolute definition of privacy, which protects in all cases, is impossible in the presence of background information on behalf of the adversary [19], the field of Privacy Preserving Data Publishing (PPDP) has focused on finding privacy criteria for situations where the adversary has only limited background information. In this setting, one can then identify possible attacks that an adversary could employ to compromise privacy in a published database.

Determining the exact type of attack to focus on for operations research problems is difficult, as the relevant type of attack depends on the privacy implications of the used data. For this reason, we will focus in this paper on one type of attack that has been given significant attention in the PPDP literature: the linkage attack. This type of attack refers to the situation in which an adversary is able to link an individual to either a certain entry in the table, a given attribute of the table or even to the table as a whole [21]. A privacy measure that has been linked to this type of attack is that of k-anonymity:

Definition 1 (k-Anonymity [40])**.** A table satisfies k-anonymity if every record in the table is indistinguishable from at least $k-1$ other records with respect to every set of quasi-identifier attributes.

One can achieve k-anonymity in a database through various ways of recoding the data, where the two most common techniques are that of generalization and suppression [4,32]. Here, generalization refers to making the given entry more general, while suppression accounts to replacing the entry by a predefined suppressing token. Clearly, the exact way in which this anonymization is performed impacts the performance of the complete optimization process, as it determines the information available to the optimization process. In this paper we will investigate optimization performance under two common recoding methods used for achieving k-anonymity to test the dependence of the final solution quality on the chosen privacy preservation method.

3.3. The framework

The framework that we have proposed in this section is illustrated in Fig. 1. The first step in the process is for the data publishing party to anonymize the data. In this paper, we assume that the publishing party uses k-anonymity as a privacy concept. The resulting anonymized database is then made available by the data publishing party.

This published database, which thus adheres to k-anonymity, now acts as input to the bin-packing optimization problem. Hence, the optimizer has to deal with the uncertainty that is created regarding the true inputs for the bin-packing problem. The result of the optimization procedure is a solution to the bin-packing problem. Considering that the input to the bin-

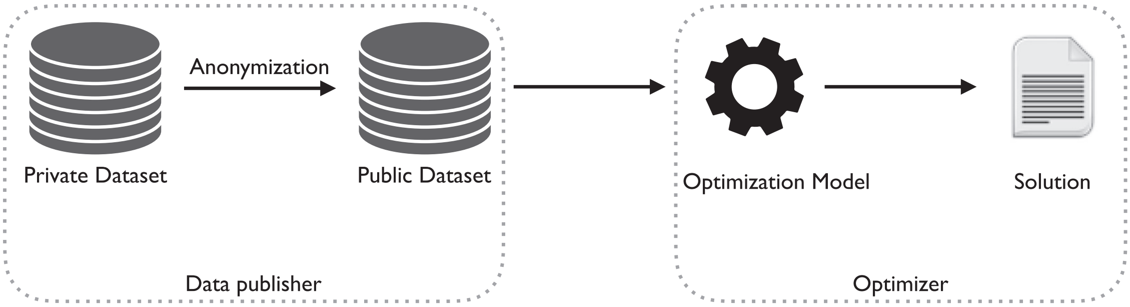


Fig. 1. Schematic overview of the proposed framework.

packing problem was anonymized, this solution to the problem does not leak any privacy sensitive information as well. The solution can then be published by the optimizer or implemented in some business process.

The aim in this paper is to perform anonymization and optimization in such a way that the quality of the solution is as good as possible. In particular, this implies that we try to find a solution that is in quality as close as possible to the optimal solution that would be obtained in case no anonymization is applied. For this reason, we will investigate in the next two sections how to perform the anonymization of the data and how to perform the optimization of the bin-packing problem in such a way that the loss of solution quality due to anonymization is as small as possible.

4. Recoding methods for *k*-anonymity

As finding an optimal *k*-anonymous generalization is, in general, an NP-hard problem [36,44], different methodologies and algorithms have been developed to perform the recoding. A general distinction to be made is between global and local recoding. A global recoding is a recoding in which equal data values are mapped to identical generalizations. Local recoding relaxes this condition. As global recoding has been developed considerably more than local recoding, we will focus on global recoding methods in this paper.

For our further analysis, we adopt the domain generalization relation $<_D$ as proposed by [37]. Here, a domain relates to the possible values that can be taken for a certain attribute. Now, $D_i \leq {}_D D_j$ implies that the domain D_i is identical or generalized by D_j. A many-to-one value generation function γ: $D_i \to D_j$ is thus associated to each domain generalization, where one domain generalizes many other domains. Extending this, we can define a domain generalization hierarchy to be a set of domains that is totally ordered by the relation $<_D$. These generalizations can be represented by nodes and edges. Here an edge implies a direct generalization as given by γ, whereas a path gives an implied generalization as denoted by γ^+. If we then construct the tree implied by the functions γ, we obtain a value generalization hierarchy, for which an example is shown in Fig. 2. We will now distinguish between the global recoding techniques of full-domain generalization and partition-based single-dimensional recoding.

4.1. Full-domain generalization

The most often used technique of global recoding is that of full-domain generalization [49]. The main idea behind this method is that the level of generalization is determined at the attribute level, implying that all values belonging to an attribute are generalized to an equivalent level in the value generalization hierarchy. More formally, if we let D_{A_i} indicate the domain of attribute A_i:

Definition 2 (Full-domain generalization [37]). Let $T(A_1, \ldots, A_m)$ be a table with quasi-identifier attributes $Q_1, \ldots, Q_r$. A full-domain generalization is defined by a set of functions $\phi_1, \ldots, \phi_r$, each of the form $\phi_i : D_{Q_i} \to D_{B_i}$, where $D_{Q_i} \leq_D D_{B_i}$. ϕ_i maps each value $q \in D_{Q_i}$ to some $b \in D_{B_i}$, such that $b = q$ or $b \in \gamma^+(q)$. A full-domain generalization V of T is obtained by replacing the value q of attribute in Q_i in each entry of T with the value $\phi_i(q)$.

An important reason for the popularity of full-domain generalization lies in the reduction of search space it implies, as the level of generalization is decided at the attribute level instead of at the value level. This comes at the cost of generalizing values further than strictly necessary to obtain *k*-anonymity, meaning a loss of precision and increased data uncertainty in optimization.

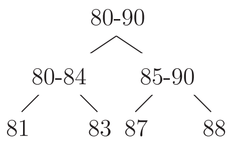


Fig. 2. Possible value generalization hierarchy for the weights in Table 1.

To investigate the effect of this loss of precision on the final solution quality of the optimization we will consider the Flash implementation of full-domain generalization as suggested by [33] as one of our anonymization methods. To guide the search over the search lattice constructed by this method, we make use of a variant of the loss metric as suggested by [28] for the case of numeric data. This metric provides a good candidate for optimization as it minimizes the total spread in the data and is thus likely to reduce the uncertainty in the data. Here we define the set of equivalence classes E, where an equivalence class consists of all data entries that have become generalized to the same generalization. Let the interval to which the values in this equivalence class with respect to attribute i are generalized be given by L_i^e and U_i^e and let L_i and U_i indicate respectively the largest and smallest value in the domain of Q_i. The metric is then given by

$$C_{LM} = \sum_{i=1}^{r} \beta_i \left(\sum_{\forall e \in E : |e| \geq k} |e| \frac{U_i^e - L_i^e}{U_i - L_i} + \alpha \sum_{\forall e \in E : |e| < k} |e| \right). \tag{7}$$

The first inner sum represents the sum of (standardized) costs for values that are generalized, while the second inner sum gives the cost of values that are suppressed instead. The scaling constant α allows to change the degree of suppression.

4.2. Partition-based single-dimensional recoding

The second type of recoding we consider is that of partition-based single-dimensional recoding, where the assumption of equal equivalence levels is relaxed and where, instead of assuming a value generalization hierarchy, we assume that the interval can be partitioned into a set of disjoint intervals:

Definition 3 (Single-dimensional partitioning [38])**.** Let $T(A_1, \ldots, A_m)$ be a table with quasi-identifier attributes $Q_1, \ldots, Q_r$. Assume the domain of each quasi-identifier Q_i can be represented by a totally ordered set. Let ϕ_i map $D_{Q_i} \to Z$, such that Z is a set of disjoint intervals that cover D_{Q_i}. Then if we map every $q \in D_{Q_i}$ to $\phi_i(q)$ this defines a single-dimensional partitioning.

The aim is thus to find some partitioning $(I_1, I_2, \ldots, I_o)$ such that these domains cover D_{Q_i}, fulfill k-anonymity and where this partitioning gives us the highest information utility possible. The algorithm we employ is that of K-Optimize [4], which according to [21] is one of the few algorithms that can perform single-dimensional partitioning effectively. K-Optimize first formulates a set representation of anonymizations, where every element in the set indicates the start of a new interval. The empty set is thus the most general anonymization, while if we let ν_{l_i} indicate the smallest value in D_{Q_i}, $\Sigma_{\text{all}} = \bigcup_{i=1}^{r} (D_{Q_i} \setminus \nu_{l_i})$ represents the most specific generalization. This follows due to the fact that the smallest value in a domain always indicates the start of a new interval.

Example 1. With respect to the weights considered in Table 1, the set {83, 87} represents an anonymization where we split the domain D_{Q_i} into the intervals [81,83),[83,87),[87,88].

Considering this set representation, the problem reduces to a search through the power set of Σ_{all}. To solve this problem the authors suggest a specialized version of the OPUS framework [54], which is a branch-and-bound framework for unordered search. To measure the quality of an anonymization we again use the loss metric as proposed in Eq. (7). In addition we need a lower bounding function for the loss metric, which gives a lower bound for the cost in any sub-tree of a node in the search tree. We use the lower bounding function

$$LB_{LM} = \sum_{\forall t \in T} \begin{cases} \sum_{i=1}^{r} \beta_i \alpha & \text{if } H \text{ suppresses } t \\ \sum_{i=1}^{r} \beta_i \left(\dfrac{U_i^{a_t} - L_i^{a_t}}{U_i - L_i} \right) & \text{otherwise}, \end{cases} \tag{8}$$

where H is the generalization at the current node and $U_i^{a_t}$ and $L_i^{a_t}$ are respectively the lower and upper bound for the equivalence class in which the value t lies with respect to the generalization implied by the Σ_{all} of the current node. The upper case in the lower bounding function corresponds to the case where t is already suppressed in the current generalization. Clearly, this implies that t will be suppressed for the whole branch, resulting in the first sum as lower bound for this value t. The second case applies if entry t is not yet suppressed, meaning that the width of the equivalence class is always bounded from below by the most specific anonymization in this branch of the tree, that implied by the current Σ_{all}.

5. Optimization methods for privacy preserved input data

In this section, we study how we pack items of different weights into bins such that the number of used bins is minimized while the weight capacity of each bin is not violated by the packed items. Recall that, different from the standard bin-packing problem, here the weights of items are anonymized, i.e., they are the query results from the privacy-preservation methods described in Section 4. One straightforward optimization method is to directly consider the anonymized data as an input to the bin-packing problem, meaning that we for example directly use the lower bounds produced by generalization. However, this method suffers from the following two issues: (1) the solution quality, i.e., the number of used bins, can be largely overestimated or underestimated; and even more seriously, (2) the solutions can be infeasible because the weight capacity constraint of bins could be violated. These two issues are caused by the fact that this straightforward method disregards the data uncertainty, which is caused by the generalization and suppression of values in the anonymization.

One popular method in the operations research literature to handle such data uncertainty is that of robust optimization [7], for which the basic idea is to make sure that a solution is feasible for every choice of weight $w \in \mathcal{U}$. Here $\mathcal{U}$ denotes the uncertainty set for the weights. Then the problem

$$\min_{x,y} \left\{ \sum_{i \in N} y_i : \sum_{j \in N} w_j x_{i,j} \le c y_i \quad \forall i \in N, w \in \mathcal{U}, \sum_{i \in N} x_{i,j} = 1 \quad \forall j \in N, x_{i,j}, y_i \in \{0,1\} \quad \forall i, j \in N \right\} \tag{9}$$

is known as the robust counterpart of the bin-packing problem. However, it can easily be seen that this robust counterpart in case of k-anonymization methods, with column-wise uncertainty, simply corresponds to the case of using the upper bounds of the intervals as output statistic $f(T)$ from the generalization. In that case, the found solution will be feasible for any realization for the true weights.

Although this robust counterpart approach ensures feasible solutions to the bin-packing problem, it comes at a great cost to the solution quality. One possible solution to overcoming the conservatism of the robust optimization approach is to only enforce feasibility in a certain percentage of the cases. This approach corresponds to that of using chance constraints [15], where we want all constraints to be jointly satisfied with some probability $1 - \delta$ for $\delta \in [0, 1]$. To simplify further analysis, let us parameterize the uncertainty as in [5]. Here assume a vector of nominal weights w^0 and basic unit shifts w^l and let ζ be a perturbation vector taking realizations from distribution P. Then the true weights are given by

$$w = w^0 + \sum_{l=1}^{n} \zeta_l w^l, \quad \zeta \sim P, \tag{10}$$

after which we consider the joint chance constraint

$$Prob_{\zeta \sim P}\left[\zeta : [w^0]^T x_i + \sum_{l=1}^{n} \zeta_l [w^l]^T x_i \le c y_i \; \forall i \in N \right] \ge 1 - \delta \tag{11}$$

for the set of capacity constraints in Eq. (4). Here, $x_i = [x_{i,1}, \dots, x_{i,n}]^T$ corresponds to the vector of packing decisions for bin i. We now propose two ways to solve the bin-packing problem involving the joint chance constraint (Eq. (11)).

5.1. Anonymization-aware stochastic programming

An intuitive way to solve a problem involving the joint chance constraint (Eq. (11)) is through considering the support of the distribution of P, which we call Ω. In our setting we know that this support is discrete, as k-anonymity provides a discrete amount of different generalizations. Let E again indicate the set of equivalence classes and assume $|E| = q$. Let additionally the set of all permutations of the equivalence class $e_i \in E$ be given by Perm_{e_i}. Then every realization of P is some $p \in Z = \mathrm{Perm}_{e_1} \times \dots \times \mathrm{Perm}_{e_q}$.

Example 2. Consider the last four weights of Table 1 and the case in which two values are generalized to the interval [80,85] and two to the interval [86,90]. In this case Ω consists of the Cartesian product {{81, 83}, {83, 81}} × {{87, 88}, {88, 87}}.

In order to establish Ω we need for every equivalence class the values contained in it, more specifically, we need all underlying weights of the database. One may wonder whether it leads to a breach of k-anonymity. Although it clearly leaks more information to an adversary than only the upper bound on an interval, all values in an equivalence class are still generalized to the same generalization. This makes that each entry is still indistinguishable from at least $k - 1$ other entries. Hence, the criterion of k-anonymity is still satisfied. However, the extra information that is provided may make one liable to other type of attacks, making that a trade-off is applicable between privacy and optimization quality. This trade-off is studied further in Section 6.5.

We can now transform the uncertain problem with chance constraints to a deterministic model [53]. Considering that our distribution P is discrete, it can be represented by some finite support Ω. Let $\omega \in \Omega$ indicate a realization from this support, let p_ω be the probability of this realization occurring, $w_{j,\,\omega}$ be the associated weight for item j corresponding to realization ω and M be a sufficiently large number. Then we can solve our bin-packing problem with k-anonymized weights by solving the deterministic problem:

$$\min \sum_{i \in N} y_i \tag{12}$$

$$\text{s.t.} \sum_{j \in N} w_{j,\omega} x_{i,j} - M z_\omega \le c y_i \qquad \forall i \in N, \omega \in \Omega \tag{13}$$

$$\sum_{i \in N} x_{i,j} = 1 \qquad \forall j \in N \tag{14}$$

$$\sum_{\omega\in\Omega} p_\omega z_\omega \le \delta \tag{15}$$

$$y_i, x_{i,j}, z_\omega \in \{0,1\} \qquad \forall i, j \in N, \omega \in \Omega. \tag{16}$$

In this formulation we allow by introducing the binary variables z_ω in the constraints (13) that the capacity may be violated in the cases where $z_\omega = 1$. Then we enforce in the knapsack constraint (15) that this does not happen more often than with probability δ, such that the joint chance constraint is satisfied. Note that in general considering the whole support Ω is computationally infeasible even in the case of finite support, since many permutations are possible.

Example 3. Consider a case with only 25 weights, where we enforce 5-anonymity. Then we have 5 equivalence classes with 5 weights in the best case, which already corresponds to $(5!)^5 \approx 2.5{*}10^{10}$ different realizations.

Considering this computational difficulty in using the complete support, we use sampling to create a subset Ω_s of Ω using s realizations, as suggested by [13]. Then we solve the sample average approximation of the optimization problem (12)–(16) with Ω replaced by Ω_s. The level of s is dependent on the level of confidence we want. Following [13], we specify δ not too small a priori, after which we can determine through the means of Monte-Carlo techniques the constraint violation a-posteriori more exactly.

5.2. Robust optimization

A major disadvantage of the stochastic programming based solution is that it often results in large optimization problems. Hence, we use a robust counterpart approach that has been suggested to handle chance constraints in [5], which instead uses probability bounds. Here we relax the assumption that $\zeta \sim P$ and instead assume that P belongs to some family of functions $\mathcal{P}$. We apply the Bonferonni inequality to replace the relatively difficult joint chance constraints to n relatively easier individual chance constraints. This brings us to the ambiguous chance constraints:

$$Prob_{\zeta\sim\mathcal{P}}\left[\zeta : [w^0]^T x_i + \sum_{l=1}^{n} \zeta_l [w^l]^T x_i > cy_i\right] \le \frac{\delta}{n} \quad \forall i \in N \tag{17}$$

The exact assumptions made on $\mathcal{P}$ determine the robust approximations of these chance constraints.

To reduce computational complexity, we assume that our region of uncertainty is bounded with the mean in the middle of the interval. As we are mostly interested in the upper bounds of some interval $[l, u]$, we can consider our interval to be $[\mu - (u-\mu), \mu + (u-\mu)]$. We additionally assume that:

$$E[\zeta_l] = 0, l = 1,\dots,n \quad \& \quad |\zeta_l| \le 1, l = 1,\dots,n \quad \& \quad \{\zeta_l\}_{l=1}^n \text{ are independent} \tag{18}$$

While the last assumption of independence is clearly not realistic it allows for a more simple robust counterpart analysis in our case where describing covariances is difficult analytically. An important implication of this will be that later results will not tend to hold exactly and hence will only act as approximate bounds. However, as suggested before, one can always apply Monte-Carlo simulations to determine more exactly the feasibility afterwards.

In this paper, we consider a budget of uncertainty [5,8] to reduce computational complexity. This gives us the set of perturbations

$$\mathcal{Z} = \left\{\zeta \in \mathbb{R}^n : -1 \le \zeta_l \le 1, l = 1,\dots,n, \sum_{l=1}^{n} |\zeta_l| \le \gamma\right\}, \tag{19}$$

giving for the individual chance constraints the robust counterpart

$$\sum_{l=1}^{n} |z_{i,l}| + \gamma \max_l |q_{i,l}| + [w^0]^T x_i \le cy_i \qquad \forall i \in N, \tag{20}$$

$$z_{i,l} + q_{i,l} = -[w^l]^T x_i \qquad \forall i \in N, l \in N, \tag{21}$$

which can easily be transformed into a linear system. The γ in this case acts as a budget of uncertainty and to enforce the chance constraints we should take $\gamma = \sqrt{-2n\log\left(\frac{\delta}{n}\right)}$.

6. Experimental evaluation

In the last two sections, different methodologies were presented for both the phase of privacy preservation and optimization. We presented theoretical arguments to support the use of these methods in our setting of combining privacy preservation and optimization. The aim of this section is to evaluate these methods empirically.

Table 2
Experimental setup in terms of bin-packing input data.

setting	c	l	u	n	distribution
I: 25Item/LWeight/Eq	500	0.25	0.75	25	100%: U(125,375)
II: 50Item/LWeight/Eq	500	0.25	0.75	50	100%: U(125,375)
III: 25Item/SWeight/Eq	2500	0.05	0.15	25	100%: U(125,375)
IV: 50Item/SWeight/Eq	2500	0.05	0.15	50	100%: U(125,375)
V: 25Item/LWeight/UnEq	500	0.25	0.75	25	75%: $U(125, 250)$; 25%: $U(250, 375)$

6.1. Performance measures

As our final aim is to find anonymization and optimization methodologies that perform well in terms of the found solutions from the optimization, it is natural to use solution quality as a direct measure for evaluating the performance of a suggested methodology. Note here that, due to the uncertainty that is added during privacy preservation, the solutions are likely to deviate from the true results. Hence the first two performance measures are:

1. The performance ratio $\frac{o}{o_n}$, where $o = \sum_{i\in N} y_i$ is the found objective value and o_n is the objective value without considering privacy preservation (i.e., actual weight values are used in the bin-packing problem).
2. The feasibility f of the found solution with regard to satisfying the constraints (4)–(6) for the true weights w_j.

Note that the found objective value o is merely the number of used bins as the outcome of solving the bin-packing problem with anonymized weights, while o_n is the number of bins used in case the true weights are used. Moroever, note that in case of k-anonymity, solutions are by construction not unique. Due to generalization, all values in some equivalence class $e \in E$ are mapped to the same summary statistic, which implies that the weights w_j for items in the same equivalence class are the same. Items in the same class can thus always be interchanged in the optimal solution, meaning that every element $z \in Z = \text{Perm}_{e_1} \times \ldots \times \text{Perm}_{e_q}$ is an optimal solution to the considered problem. Let $f(z)$ indicate if z is feasible given the constraints and the true weights. Then we define the feasibility f_k to be the proportion: $f_k = \frac{\sum_{z\in Z} f(z)}{|Z|}$. As Z is in general too large to be explicitly enumerated, we instead use sampling to compute f_k.

Besides solution quality, we require that the methods are able to provide solutions in reasonable time. We thus additionally consider the metrics:

3. The time needed for anonymization t_a.
4. The solving time of the optimization problem t_o.

6.2. The experimental settings

We investigate the performance of our methodologies in varying bin-packing settings. There are essentially two important properties of bin-packing problem instances that are likely to influence the solution quality. The first is the distribution of the weights over the bin capacity. As all item weights should be smaller than the bin-capacity and be positive, we assume a distribution with finite support $[lc, uc]$, where l and u are respectively the lower and upper proportion of the bin capacity c that the weights may take. We consider two options for l and u. One takes into account relatively large weights compared to the bin capacity, which is applied in setting I, II and V. The second corresponds to relatively small weights, which is used in setting III and IV. We furthermore use an uniform distribution over $[lc, uc]$ for settings I–IV. To additionally research the effects of skewness in the input data we investigate the case in which 75% of the weights lies in $[lc, \frac{l+u}{2}c]$ and 25% of the weights in $[\frac{l+u}{2}c, uc]$, where in these intervals the weights are again uniformly distributed. This corresponds to setting V.

A second important property is the spread (average distances) of the weights. Different spreads for the same distributions are achieved through distinguishing different sizes of the instances, i.e., number of items $n = 25$ and $n = 50$. Thus, we have a larger spread in case of settings I, III, and V with a small number of items $n = 25$, whereas a smaller spread in settings II and IV with $n = 50$. The five experimental settings are summarized in Table 2. In order to obtain reliable results over these different settings, 10 different instances are considered for each experimental setting.

Lastly, we describe the computational environment. We implemented the K-Optimize privacy methods in Java, and used the ARX package [47] to implement the Flash algorithm. We use CPLEX 12.6.1 to solve the optimization models. A time limit of 10 min was granted to the solver, after which 2 min of solution polishing was applied if the best solution has not been proven optimal. All the experiments were run on the Lisa Supercomputer [51].

6.3. The performance of the suggested privacy preservation methods

We first investigate the impact of different anonymization methods on the solution quality of the optimization problem. We use the standard model of optimization for the bin-packing problem (see Eq. (3)–(6)). Two different output statistics are provided as output of the anonymization and as input to the bin-packing problem: mean and upper bound. In case of the mean summary statistic, we use $\Sigma_{t\in e} w_t/|e|$ as input for w_j for $j \in e$. In the case of the upper bound summary statistic

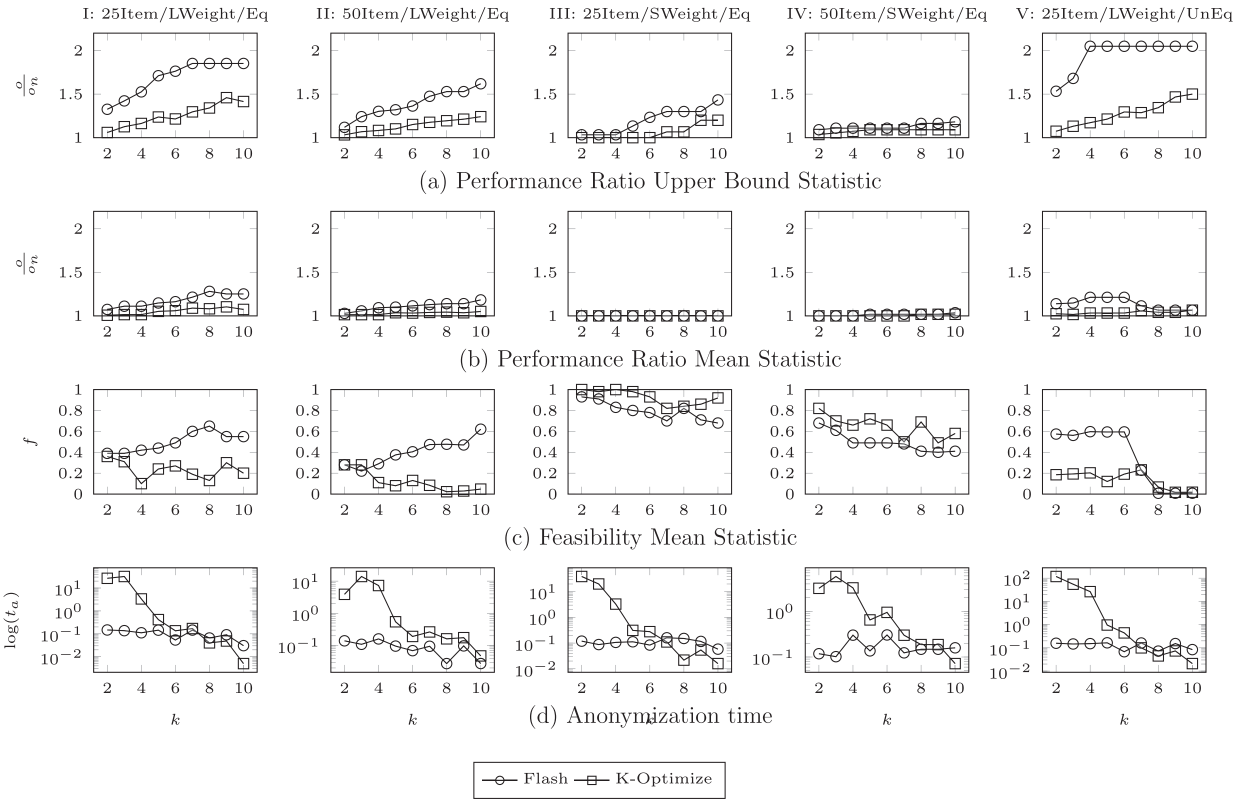


Fig. 3. Performance of two *k*-anonymity methods (Flash and K-Optimize) with upper bound and mean as input to the standard bin-packing optimization model. The x-axis shows increasing levels of *k*-anonymity, with varying *k* from 2 to 10.

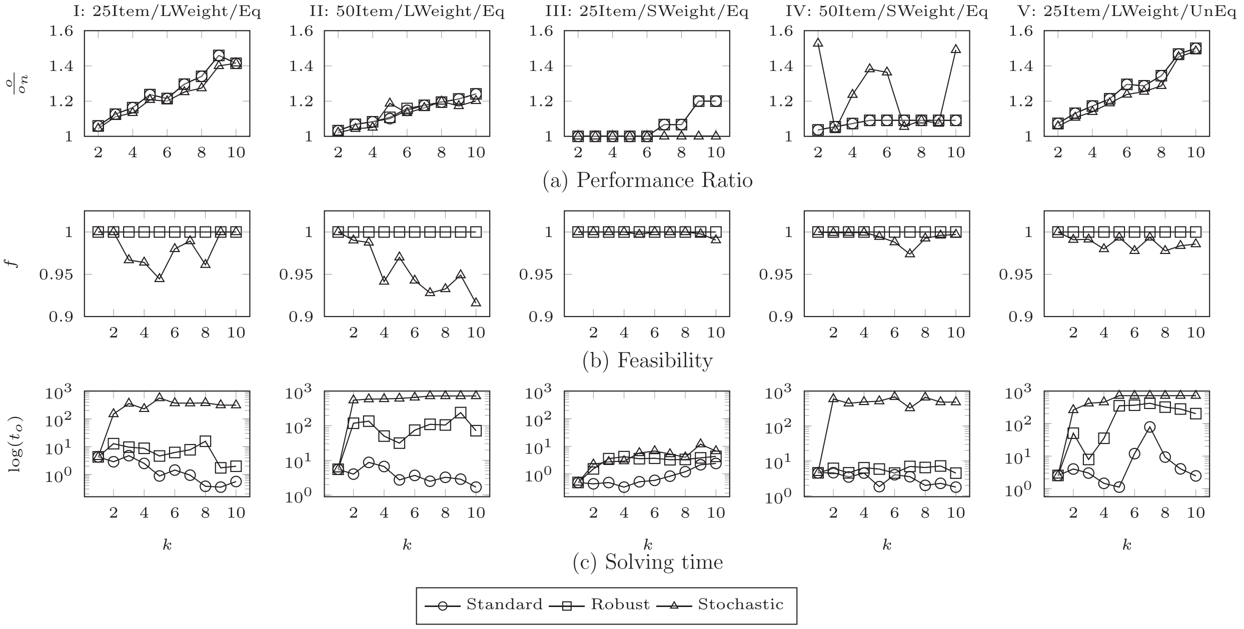
we use U_e. We expect the mean statistic to give a good estimate of the true objective value, whereas the upper bound will ensure full feasibility of the solution.

Fig. 3 shows the experimental results. The objective values obtained by using the true weights are equal to 13.5, 27.8, 3, 5.5 and 12.2 for settings I to V respectively and these values are thus used to determine the objective ratios. The level of *k* to achieve *k*-anonymity is varied from 2 to 10, with steps of size 1. Comparing the different summary statistics used, we observe that using mean values (Fig. 3(b)) in general leads to better objective values than using upper bounds (Fig. 3(a)), at a cost of lower feasibility. As using upper bound statistics always ensures feasibility of the solutions, we exclude the feasibility measure for upper bounds from Fig. 3.

One might expect the mean summary statistics to result in reasonable estimate of the true objective value, however, we see that in most settings the mean summary statistic (in Fig. 3(b)) results in objective values that are greater than the value of the optimal solution without anonymized weights. Especially in setting I, the differences are considerable. This over-estimation is due to the fact that packing items with equal values (the summary statistics) into bins is more difficult than the case of varying weights (the actual weights), in which items may more easily fit in empty spots left. The only exception seems to be the case of setting III in Fig. 3(b). This can be explained by the fact that c/μ is equal to 2.5, leaving on average some spare capacity in the last bin which acts as a buffer for deviations due to data uncertainty.

In terms of the returned objective value, we notice that the Flash algorithm, which enforces full-domain generalization, performs worse than the K-Optimize algorithm in all settings (see Fig. 3(a) and (b)). This is expected as Flash typically generalizes values further than K-Optimize. In terms of feasibility (Fig. 3(c)), Flash outperforms K-Optimize in the settings with large weights (i.e., settings I, II, V), but with a large increase in objective value (Fig. 3(b)).

Interestingly, the performance of the anonymization methods is dependent on the problem instance at hand and the level of privacy preservation. It is not surprising that a higher *k* value leads to a worse performance ratio for both Flash and K-Optimize algorithms. Nevertheless, K-Optimize performs well in most settings, especially when using the mean statistic. The solution quality obtained using the Flash anonymization with the upper bound statistic is much worse in settings I, II and V (with relatively large weights) than in settings III and IV (with relatively small weights). In particular, when *k* is larger than 7 and 4 in settings I and V, respectively, the Flash algorithm uses all available bins. This may be explained by the fact that in bin-packing problem instances items can only be combined in a bin when their weight is smaller than half the bin-capacity threshold. But in settings with relatively large and only few weights (such as settings I and V in Fig. 3 (a))



(a) Performance Ratio

(b) Feasibility

(c) Solving time

Fig. 4. Solution quality and solving time for normal, robust and stochastic optimization and the K-Optimize algorithm.

a large k may imply that all upper bounds for the equivalence classes become larger than this threshold. In such cases, the combination of items becomes impossible, leading to a situation in which all bins are used.

6.4. The performance of the suggested optimization methods

Let us now move to an evaluation of the suggested optimization methods. We consider the methods of robust optimization and stochastic programming. Additionally, we include the upper bound as a benchmark for the suggested methods. We do not consider all methods for achieving k-anonymization, but only use K-Optimize, which has overall shown the best performance in the last section. Furthermore, we enforce that all constraints are satisfied with a probability of 95% in respectively stochastic optimization and robust optimization, corresponding to a value of $\delta = 0.05$ in the formulation (12)–(16) and in Eq. (11). Furthermore, we use $s = 100$ samples in stochastic optimization.

The results of these optimizations are displayed in Fig. 4. It is clear from Fig. 4(a) that using the proposed optimization methods for anonymized data improves the solution quality compared to the standard optimization using the upper bound statistic for most of the instances. The robust approach with the budgeted uncertainty set performs not as good as we expected, as it always provides the same objective value and feasibility as the upper bound statistic. What we expected is that the chance constraints would reduce the found objective value, while ensuring an acceptable level of feasibility. The unexpected behavior can be explained as follows. First of all the implementation we considered is only an approximation as we explained in subsection 5.2. In reality our random variables ζ_l are not independent, but rather negatively correlated. There are only a finite number of realizations, meaning that finding a positive deviation reduces the chance of finding a positive deviation for the other weights. Secondly, the parameter γ is not related to the values of $x_{i,j}$. Hence, in cases where only few of the $x_{i,j}$ variables are set to 1 in an optimal solution, this method tends to over-insure against the worst-case scenario. A possible way to improve this would be to pick γ on the basis of experience, but this may be difficult in the case of privacy preservation, especially in single-shot optimization problems.

The stochastic optimization approach does give better solutions than the robust approach, and hence the upper bound benchmark, for most of the settings. Especially for instances with a small number of items (i.e., settings I, III, V), it consistently outperforms the robust approach in terms of performance ratio (Fig. 4(a)), regardless of the level of anonymity. For the settings II and IV with large problem sizes (i.e. more items), the performance of the stochastic approach is not dominant anymore. Given the current time limit set, we obtain a worse solution than by using the upper bound in 2 cases in setting II and for 5 cases in setting IV in Fig. 4(a). These results can be fully attributed to the fact that in these cases the solver was unable to find an optimal solution in the given time-frame, as can be seen by the solving times that almost equal the time limit of 12 min in Fig. 4(c). Thus, while stochastic programming is a reliable method in case of smaller settings with few items, its usefulness for larger cases is strongly dependent on the problem size and correspondingly the solving time granted.

When considering the feasibility of the obtained solutions, as shown in Fig. 4(b), we find that the anonymization-aware stochastic optimization methods satisfies the chance constraints in almost all of the instances. This implies that for most

Table 3
An example data set with attributes Disease (sensitive attribute) and Age (quasi-identifier). The original data is shown left, the middle shows the anonymized Age attribute using lower and upper bound statistics, and the right part using multiset statistics. Both anonymizations satisfy 6-anonymity.

Name	Disease	Age	lower and upper bounds	multisets
Person 1	Hepatitis	21	[21–25]	{21, 22, 23, 23, 24, 25}
Person 2	HIV	22	[21–25]	{21, 22, 23, 23, 24, 25}
Person 3	Flu	23	[21–25]	{21, 22, 23, 23, 24, 25}
Person 4	HIV	23	[21–25]	{21, 22, 23, 23, 24, 25}
Person 5	Hepatitis	24	[21–25]	{21, 22, 23, 23, 24, 25}
Person 6	Cancer	25	[21–25]	{21, 22, 23, 23, 24, 25}
Person 7	Cancer	31	[31–35]	{31, 31, 31, 32, 34, 35}
Person 8	Cancer	31	[31–35]	{31, 31, 31, 32, 34, 35}
Person 9	Cancer	31	[31–35]	{31, 31, 31, 32, 34, 35}
Person 10	Flu	32	[31–35]	{31, 31, 31, 32, 34, 35}
Person 11	HIV	34	[31–35]	{31, 31, 31, 32, 34, 35}
Person 12	Hepatitis	35	[31–35]	{31, 31, 31, 32, 34, 35}

settings we indeed obtain a feasible solution with a probability of 95%, the specified probability. Clearly, the level to which the required probability is obtained depends on the sample size. However, our sample of size $s = 100$ seems to be perform well in these settings.

6.5. The trade-off between privacy and solution quality

We now evaluate the trade-off that occurs between privacy and solution quality when a privacy preserving database is used as input to a bin-packing problem. This trade-off has proven to be dependent on the purpose of the optimizer: whether we want to find a good objective value (i.e., a correct amount of bins), or if we also want to ensure the feasibility of the allocation. In case we are mainly interested in finding a good objective value, the trade-off is acceptable in many settings. Especially in the case of the settings III and IV, where we consider only relatively small weights, the deviation from the true objective value remains limited to less than 2% when using the mean summary statistic (Fig. 3). In settings I, II, and V, the trade-off is considerably larger, where the increase in performance ratio going from a level of no privacy protection to a level of $k = 10$ corresponds to respectively 7%, 5% and 7% when using the method of k-anonymity with the mean summary statistic (see Fig. 3).

However, achieving feasible solutions is considerably more costly. For example, for setting I the objective value under stochastic optimization and $k = 8$ is about 27% larger than when using the true weights and about 20% higher than when using the mean weights. Clearly, the impact of such an increase in practical applications is large. However, with a feasibility of only 13% when using the mean weights, such an increase is hard to overcome when the found solution has to be implemented in practice.

However, the trade-off is also dependent on the instance at hand. In case of setting III it is especially acceptable, where for k up until 10 there is no trade-off at all when using the stochastic optimization and K-Optimize algorithm as seen in Fig. 4. In the case of large weights as seen in setting I the results are very dependent on k, where the increase in objective value is proportional to the value of k. This entails for example an increase of 4% in objective value in going from no privacy preservation at all to $k = 2,$ whereas going to $k = 10$ raises the objective value by about 41% (when using the methods of K-Optimize and stochastic programming, as presented in Fig. 4). In such settings it thus seems only possible to enforce a low level of privacy. When there are more items, the stochastic programming approach seems to break down, sometimes returning even worse results than simply considering upper bounds as summary statistics. This is due to the running time limit of 10 min. Hence, to be used in practice, longer running time should be granted. In addition, more advanced optimization methods may be required to improve the solution quality for large problem instances.

7. Attacks on multiset statistics

In the experiments above, we clearly see the benefit of using multiset statistics for stochastic optimization. In this section, we analyze the privacy preservation qualities of this new type of statistic, compared to the commonly used lower and upper bound values of the aggregation ranges. We consider three commonly used attack models for privacy-preserving data publishing schemes: record linkage, attribute linkage, and table linkage. In order to clarify the privacy risks, we provide a small example dataset as illustrated in Table 3.

Record Linkage

In this attack model, it is possible to uniquely identify a data owner (Person) using quasi-identifiers [21]. In the anonymized datasets, both anonymizations satisfy k-anonymity, since each record has at least $k > 1$ matching records for the corresponding quasi-identifiers. Therefore, even if an attacker has access to external knowledge, the privacy preservation quality of the multisets is the same as the lower and upper bound statistics. For instance, an attacker knows that Alice is in Table 3 and also knows her age is 21. The attacker can observe that Alice's record is one of the first six records with both multisets and lower and upper bounds, but cannot identify which one is her. Furthermore, the frequencies of age values do not provide additional information to an attacker that could identify Alice, although she is the only one in the dataset with age 21.

Attribute Linkage

In an attribute linkage attack, the data owner's record does not have to be identified uniquely but his/her sensitive attribute may be inferred through a homogeneity or background knowledge attack [21]. Similar to record linkage, the privacy of using multisets is identical to that of using lower and upper bounds. Although people in the same age set have the same disease (Person 7/8/9 are 31 years old and have cancer), the additional information of knowing the exact age values in the multisets does not help to perform a homogeneity attack since these people cannot be identified. Similarly, in case of background knowledge attack, the multisets do not leak more information than lower and upper bounds.

Table Linkage

A table linkage attack is possible if an attacker can infer the presence or absence of a record owner in a dataset [21]. While the lower and upper bounds is not a guarantee against a table linkage attack, the multiset statistics are more vulnerable against it since the values of quasi-identifiers are visible. For an attacker, inferring the presence of a data owner is probabilistic in the multiset statistics as in the lower and upper bounds. However, (s)he can confidently infer the absence of a data owner if no matching quasi-identifier value is present. For instance, if an attacker knows that Bob is 33 years old, (s)he can confidently state that Bob's record is not present in the data.

8. Conclusion

We investigated the trade-off between privacy and solution quality that occurs when a k-anonymized database is used as input to a bin-packing problem. An important contribution in this respect was to suggest a framework in which the enforcement of a formal privacy criterion introduced a minimal loss of solution quality. In this respect, we started of by considering two methods for achieving k-anonymity, out of which one was a heuristic one and the other exact. A second step was to consider, optimization methods that could reduce the effects of data uncertainty, as created by the anonymization, on solution quality. This resulted in enforcing chance constraints, for which we made use of the frameworks of robust optimization and stochastic programming.

All these methods were then empirically tested in a variety of bin-packing instances. Here we found that using an exact method, in this case K-Optimize, to achieve k-anonymity lead to significant improvements in solution quality. Furthermore, the approach of stochastic programming offered a good solution quality, by balancing feasibility and objective value in case of small problems. However, the increase in solving time made that this method was not dominant for large problem instances, where using the upper bound statistics may provide an alternative method.

The computational results show that one can obtain reasonable estimates of the true objective value when one is not interested in feasibility. However, enforcing feasibility comes at greater cost to the objective value, especially when considering a higher level of privacy preservation, i.e. a higher level of k. We conclude that using current state-of-the-art methods for privacy preservation, it is very hard to obtain feasible solutions during optimization without increasing the objective value unrealistically, or significantly decreasing the level of privacy protection. By developing new optimization methods that take the anonymization into account (k-anonymity with stochastic programming), we demonstrate that optimization performance can be increased significantly. This new method only requires a small amount of additional information that was lost during anonymization: the multiset of values used to construct an aggregation range, instead of only the lower and upper bounds.

There is however a downside to publishing multisets instead of lower and upper bounds: the database becomes susceptible to a table-linkage attack to identify the absence of individual rows. In our opinion, this is for most applications a small price to pay for the improvement in solution quality of an optimization method that uses these multisets as input. Our results clearly show the need for new ways of anonymizing data that take the optimization that will be using the resulting data into account, and vice versa.

Conflict of interest

None.

References

[1] R. Agrawal, R. Srikant, Privacy-preserving data mining, in: SIGMOD '00, 29(2), ACM, 2000, pp. 439–450.

[2] A.B. Alexandru, K. Gatsis, G.J. Pappas, Privacy preserving cloud-based quadratic optimization, in: 55th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2017, Monticello, IL, USA, October 3–6, 2017, 2017, pp. 1168–1175, doi:10.1109/ALLERTON.2017.8262869.

[3] A.B. Alexandru, K. Gatsis, Y. Shoukry, S.A. Seshia, P. Tabuada, G.J. Pappas, Cloud-based quadratic optimization with partially homomorphic encryption, 2018. CoRR abs/1809.02267, http://arxiv.org/abs/1809.02267.

[4] R.J. Bayardo, R. Agrawal, Data privacy through optimal k-anonymization, in: ICDE '05, 2005, pp. 217–228.

[5] A. Ben-Tal, L. El Ghaoui, A. Nemirovski, Robust Optimization, Princeton University Press, 2009.

[6] A. Ben-Tal, A. Nemirovski, Robust solutions of linear programming problems contaminated with uncertain data, Math. Program. 88 (3) (2000) 411–424.

[7] A. Ben-Tal, A. Nemirovski, Robust optimization–methodology and applications, Math. Program. 92 (3) (2002) 453–480.

[8] D. Bertsimas, D.B. Brown, C. Caramanis, Theory and applications of robust optimization, SIAM Rev. 53 (3) (2011) 464–501.

[9] M. Beye, Z. Erkin, R.L. Lagendijk, Efficient privacy preserving k-means clustering in a three-party setting, in: 2011 IEEE International Workshop on Information Forensics and Security, Foz do Iguaçu, Brazil, 2011, pp. 1–6, doi:10.1109/WIFS.2011.6123148.

[10] J.R. Birge, F. Louveaux, Introduction to Stochastic Programming, Springer Science & Business Media, 2011.

[11] A. Blum, K. Ligett, A learning theory approach to non-interactive database privacy, in: STOC '08, ACM, 2008, pp. 609–618.

[12] P. Bunn, R. Ostrovsky, Secure two-party k-means clustering, in: CCS '07, ACM, 2007, pp. 486–497.

[13] G. Calafiore, M.C. Campi, Uncertain convex programs: randomized solutions and confidence levels, Math. Program. 102 (1) (2005) 25–46.

[14] O. Catrina, S. De Hoogh, Secure multiparty linear programming using fixed-point arithmetic, in: ESORICS 2010, Springer, 2010, pp. 134–150.

[15] A. Charnes, W.W. Cooper, G.H. Symonds, Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil, Manage. Sci. 4 (3) (1958) 235–263.

[16] K. Chen, L. Liu, Privacy preserving data classification with rotation perturbation, in: ICDM '05, IEEE, 2005, pp. 589–592.

[17] C. Clifton, M. Kantarcioglu, J. Vaidya, Tools for privacy preserving distributed data mining, ACM SIGKDD Explor. 4 (2003) 2003.

[18] M. Delorme, M. Iori, S. Martello, Bin packing and cutting stock problems: mathematical models and exact algorithms, Decision Models for Smarter Cities, 2014.

[19] C. Dwork, Differential privacy, in: ICALP 2006, Springer, 2006, pp. 1–12.

[20] Z. Erkin, T. Veugen, T. Toft, R. Lagendijk, Privacy-preserving user clustering in a social network, in: WIFS '09, 2009, pp. 96–100.

[21] B.C. Fung, K. Wang, A.W.-C. Fu, S.Y. Philip, Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques, CRC Press, 2010.

[22] C.J. Geyer, Markov chain monte carlo maximum likelihood(1991).

[23] A. Gupta, K. Ligett, F. McSherry, A. Roth, K. Talwar, Differentially private combinatorial optimization, in: SODA '10, SIAM, 2010, pp. 1106–1125.

[24] S. Han, U. Topcu, G.J. Pappas, Differentially private convex optimization with piecewise affine objectives, in: 53rd IEEE Conference on Decision and Control, 2014, pp. 2160–2166, doi:10.1109/CDC.2014.7039718.

[25] S. Han, U. Topcu, G.J. Pappas, Differentially private distributed constrained optimization, IEEE Trans. Automat. Control 62 (1) (2017) 50–64.

[26] J. Hsu, A. Roth, T. Roughgarden, J. Ullman, Privately solving linear programs, in: J. Esparza, P. Fraigniaud, T. Husfeldt, E. Koutsoupias (Eds.), Automata, Languages, and Programming, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 612–624.

[27] Z. Huang, S. Mitra, N. Vaidya, Differentially private distributed optimization, in: Proceedings of the 2015 International Conference on Distributed Computing and Networking, in: ICDCN '15, ACM, New York, NY, USA, 2015, pp. 4:1–4:10, doi:10.1145/2684464.2684480.

[28] V.S. Iyengar, Transforming data to satisfy privacy constraints, in: KDD '02, 2002, pp. 279–288.

[29] G. Jagannathan, K. Pillaipakkamnatt, R.N. Wright, A new privacy-preserving distributed k-clustering algorithm, in: Proceedings of the 2006 SIAM International Conference on Data Mining, 2006, pp. 494–498, doi:10.1137/1.9781611972764.47.

[30] G. Jagannathan, R.N. Wright, Privacy preserving distributed K-means clustering over arbitrarily partitioned data, in: KDD '05, ACM, New York, NY, USA, 2005, pp. 593–599. http://doi.acm.org/10.1145/1081870.1081942.

[31] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[32] S. Kisilevich, L. Rokach, Y. Elovici, B. Shapira, Efficient multidimensional suppression for k-anonymity, IEEE Trans. Knowl. Data Eng. 22 (3) (2010) 334–347.

[33] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, K.A. Kuhn, Flash: efficient, stable and optimal k-anonymity, in: PASSAT '12, IEEE, 2012, pp. 708–717.

[34] V. Kolesnikov, A.-R. Sadeghi, T. Schneider, Improved garbled circuit building blocks and applications to auctions and computing minima, CANS '09, LNCS, Springer, 2009.

[35] R.L. Lagendijk, Z. Erkin, M. Barni, Encrypted signal processing for privacy protection: conveying the utility of homomorphic encryption and multiparty computation, IEEE Signal Process. Mag. 30 (1) (2013) 82–105, doi:10.1109/MSP.2012.2219653.

[36] M. Laszlo, S. Mukherjee, Approximation bounds for minimum information loss microaggregation, IEEE Trans. Knowl. Data Eng. 21 (11) (2009) 1643–1647.

[37] K. LeFevre, D.J. DeWitt, R. Ramakrishnan, Incognito: efficient full-domain k-anonymity, in: SIGMOD '05, ACM, 2005, pp. 49–60.

[38] K. LeFevre, D.J. DeWitt, R. Ramakrishnan, Mondrian multidimensional k-anonymity, ICDE '06, IEEE, 2006. pp. 25–25

[39] Y. Lindell, B. Pinkas, Privacy preserving data mining, J. Cryptol. 15 (3) (2002) 177–206.

[40] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, L-diversity: privacy beyond k-anonymity, ACM Trans. Knowl. Discov. Data 1 (1) (2007) 3.

[41] T.-P. Martello, Silvano, Knapsack Problems : Algorithms and Computer Implementations, J. Wiley & Sons, Chichester; New York, 1990.

[42] L. Martino, V. Elvira, D. Luengo, J. Corander, F. Louzada, Orthogonal parallel mcmc methods for sampling and optimization, Digit Signal Process. 58 (2016) 64–84.

[43] F. McSherry, K. Talwar, Mechanism design via differential privacy, in: FOCS '07, IEEE, 2007, pp. 94–103.

[44] A. Meyerson, R. Williams, On the complexity of optimal k-anonymity, in: PODS '04, ACM, 2004, pp. 223–228.

[45] S. Oliveira, O. Zaiane, Privacy preserving clustering by data transformation, in: Proceedings of the 18th Brazilian Symposium on Databases, 2003, pp. 304–318.

[46] S. Oliveira, O. Zaiane, Achieving privacy preservation when sharing data for clustering, in: SDM '04, in: LNCS, Springer, 2004, pp. 67–82.

[47] F. Prasser, F. Kohlmayer, R. Lautenschläger, K.A. Kuhn, Arx-a comprehensive tool for anonymizing biomedical data, in: Proceedings of the 2014 AMIA Annual Symposium (AMIA 2014), 2014, American Medical Informatics Association, 2014, p. 984.

[48] P. Samarati, Protecting respondents identities in microdata release, IEEE Trans. Knowl. Data Eng. 13 (6) (2001) 1010–1027.

[49] P. Samarati, L. Sweeney, Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression, Technical Report, SRI International, 1998.

[50] Y. Shoukry, K. Gatsis, A. Al-Anwar, G.J. Pappas, S.A. Seshia, M.B. Srivastava, P. Tabuada, Privacy-aware quadratic optimization using partially homomorphic encryption, in: 55th IEEE Conference on Decision and Control, CDC 2016, Las Vegas, NV, USA, December 12–14, 2016, 2016, pp. 5053–5058, doi:10.1109/CDC.2016.7799042.

[51] SURFsara, Lisa cluster, 2015, http://surfsara.nl.

[52] L. Sweeney, K-anonymity: a model for protecting privacy, Int. J. Uncertain. Fuzziness Knowl. Based Syst. 10 (05) (2002) 557–570.

[53] M. Tanner, E. Beier, A general heuristic method for joint chance-constrained stochastic programs with discretely distributed parameters, Optim. Online (2010).

[54] G.I. Webb, Opus: an efficient admissible algorithm for unordered search, J. Artif. Intell. Res. (1995) 431–465.