Hierarchical adaptive evolution framework for privacy-preserving data publishing

World Wide Web (2024) 27:49
https://doi.org/10.1007/s11280-024-01286-z

Hierarchical adaptive evolution framework for privacy-preserving data publishing

Mingshan You[1] · Yong-Feng Ge[1] · Kate Wang[2] · Hua Wang[1] · Jinli Cao[3] · Georgios Kambourakis[4]

Received: 28 February 2024 / Revised: 10 May 2024 / Accepted: 2 July 2024 /
Published online: 12 July 2024

Abstract

The growing need for data publication and the escalating concerns regarding data privacy have led to a surge in interest in Privacy-Preserving Data Publishing (PPDP) across research, industry, and government sectors. Despite its significance, PPDP remains a challenging NP-hard problem, particularly when dealing with complex datasets, often rendering traditional traversal search methods inefficient. Evolutionary Algorithms (EAs) have emerged as a promising approach in response to this challenge, but their effectiveness, efficiency, and robustness in PPDP applications still need to be improved. This paper presents a novel Hierarchical Adaptive Evolution Framework (HAEF) that aims to optimize t-closeness anonymization through attribute generalization and record suppression using Genetic Algorithm (GA) and Differential Evolution (DE). To balance GA and DE, the first hierarchy of HAEF employs a GA-prioritized adaptive strategy enhancing exploration search. This combination aims to strike a balance between exploration and exploitation. The second hierarchy employs a random-prioritized adaptive strategy to select distinct mutation strategies, thus leveraging the advantages of various mutation strategies. Performance bencmark tests demonstrate the effectiveness and efficiency of the proposed technique. In 16 test instances, HAEF significantly outperforms traditional depth-first traversal search and exceeds the performance of previous state-of-the-art EAs on most datasets. In terms of overall performance, under the three privacy constraints tested, HAEF outperforms the conventional DFS search by an average of 47.78%, the state-of-the-art GA-based ID-DGA method by an average of 37.38%, and the hybrid GA-DE method by an average of 8.35% in TLEF. Furthermore, ablation experiments confirm the effectiveness of the various strategies within the framework. These findings enhance the efficiency of the data publishing process, ensuring privacy and security and maximizing data availability.

Keywords Privacy-preserving data publishing · t-closeness anonymization · Genetic algorithm · Differential evolution · Adaptive strategy

✉ Yong-Feng Ge
yongfeng.ge@vu.edu.au

Extended author information available on the last page of the article

1 Introduction

In today's digital age, data plays a crucial role in driving innovation and decision-making, while the issue of privacy has become increasingly pertinent [1–5]. Privacy-Preserving Data Publishing (PPDP) has emerged as a paramount need, seeking to strike a delicate balance between sharing valuable information and safeguarding individuals' sensitive data [6–9]. This practice entails the dissemination of datasets that have been carefully anonymized or transformed to protect the privacy of individuals while still allowing researchers and organizations to extract meaningful insights [10–13]. Due to a heightened awareness of privacy risks, stricter regulations related to data protection, and an understanding of the necessity of responsible data processing, the popularity of privacy-protected data is on the rise [14–16].

Preserving data privacy while maintaining data utility is indeed a challenging task, and it remains one of the most pressing and complex challenges in the field [2, 17–19]. Various approaches have emerged to address this challenge, including data anonymization, differential privacy, secure multiparty computation, and homomorphic encryption [20–23]. Among these techniques, data anonymization is widely adopted. It involves modifying or removing identifying information from a dataset to protect individual privacy. Techniques such as generalization, suppression, perturbation, or data synthesis are employed to mitigate the risk of re-identification while ensuring the usefulness of the analyzed data [24, 25]. However, despite its widespread adoption and effectiveness in enhancing privacy protection, data anonymization also faces significant computational challenges due to NP-hardness restrictions on optimization [26], making it often impractical to find an exact solution within a reasonable time frame.

Some studies have introduced Evolutionary Algorithms (EAs), a common solution to NP-hard problems [27], to optimize data anonymization schemes. Among EAs, Differential Evolution (DE) and Genetic Algorithm (GA) are popular choices [28]. These algorithms find wide applications in fields such as engineering, machine learning, and bioinformatics, providing flexible and powerful optimization methods [25, 29]. DE uses differential mutation and crossover to evolve populations, emphasizing exploration. On the other hand, GA models natural selection and genetics, focusing more on exploitation. Both methods have shown effectiveness in solving complex optimization problems. This paper leverages an innovative framework that combines GA and DEs (including DE variants) for better performance and robustness in PPDP.

Previous academic studies have delved into using GA and DE algorithms for data anonymization. However, it is important to note that both GA and DE have multiple variants, and the efficacy of these variants, when applied to this problem, has yet to be thoroughly tested and evaluated. The performance of these different variants on data anonymization problems can vary significantly and needs to be explored experimentally. This exploration can help identify the most effective algorithm variants for optimizing data anonymization schemes. Additionally, the performance of different algorithms on different datasets can be uneven. Therefore, there is an urgent need for the development of more robust and effective algorithms.

This paper aims to enable a more practical application of EA in data anonymization processing, by improving the performance and robustness of the algorithm. We develop an effective adaptive strategy that dynamically combines the strengths of GA and DE mutation strategies, resulting in improved algorithm performance. Furthermore, we design a GA priority strategy and a random-based-DE priority strategy, which use GA and random-based-DE with greater probability in the early stage of evolution to enhance the population diversity and explorative search. In the later stages of evolution, best-based-DE is predominantly utilized to expedite the convergence of the search process and explorative search. This paper contributes to PPDP in the following ways.

- This paper introduces an innovative Hierarchical Adaptive Evolutionary Framework (HAEF), seamlessly integrating GA, DE, and variants of DE. In contrast to previous algorithms utilizing solely GA or both GA and DE with a single mutation strategy [30–32], HAEF integrates a richer array of evolutionary strategies. As a result, it offers more outstanding performance and stronger robustness anonymization algorithm for PPDP.
- In addition, this paper develops a novel adaptive strategy that intelligently leverages the unique characteristics of GA and various DE mutation strategies. Compared with the previous approach that mechanically alternated between GA and DE strategies [32], our method selects the subsequent evolution strategy based on the historical success probability of each strategy.
- Furthermore, we design a GA-prioritized strategy and a random-prioritized strategy. The GA-prioritized strategy utilizes GA and random-based-DE strategies more in the early to middle stages of the evolution process, enhancing the algorithm's robustness. The best-based-DE strategies are utilized more in the middle to late stages, improving the algorithm's convergence speed.
- To validate the effectiveness of our proposed method, we conduct comprehensive experiments. Comparative experiments with previous methods confirm the superiority of our framework. Ablation experiments further demonstrate the effectiveness of this strategy.

The paper is structured in the following manner: Section 2 offers an overview of the current literature in the field. Section 3 delves into a detailed definition of the optimal anonymization problem of t-closeness. Section 4 elucidates the HAEF approach that has been proposed. The experimental setup is specified in Section 5, and the experimental results are analyzed in Section 6. Section 7 discusses the limitations of this work and its potential real-world applications. Lastly, Section 8 concludes the paper by summarizing the essential findings and contributions.

2 Related work

Data anonymization is a crucial method for PPDP. Various privacy assessment models have been proposed to ensure the privacy and confidentiality of sensitive data, with some of the most classic models including k-anonymity, l-diversity, and t-closeness [33–35]. k-anonymity ensures that each record in a dataset is not distinguishable from at least k-1 other records, thereby protecting individual identities [33]. On the other hand, l-diversity prevents attribute leakage by requiring that each equivalence class contains at least l distinct sensitive attribute values [34]. t-closeness, an important model, requires that the distribution of a sensitive attribute in any equivalence class be very similar to the distribution of that attribute in the whole dataset. This approach enhances privacy guarantees by ensuring that sensitive attributes are not overly concentrated in certain equivalence classes [35].

There has been considerable research on finding the optimal way to anonymize data for k-anonymity. However, more research is needed to find optimal techniques for t-closeness anonymization. A study by Liang et al. [26] has proven that finding the optimal t-closeness anonymization solution is NP-hard. The limited number of studies addressing this issue underscores the need for further research.

Traditional search methods for finding the best anonymization scheme rely on depth-first or breadth-first traversal methods, supplemented by search space optimization techniques. For instance, Kohlmayer et al. [36] proposed the Flash algorithm, which uses a depth-first traversal search to achieve the best k-anonymity. While this method can guarantee the optimal solution for simple datasets, it may not be effective for complex datasets.

EAs have emerged as powerful tools for solving the NP-hard problem of data anonymization, particularly for complex datasets [1]. These algorithms combine evolutionary principles with optimization techniques to search for high-quality solutions efficiently. Ge et al. proposed the Information-Driven Genetic Algorithm (IDGA) [30] and the Information-Driven Distributed Genetic Algorithm (ID-DGA) [31], which use GA-based approaches to optimize k-anonymity. The Two-Layer Evolutionary Framework (TLEF), introduced by You et al. [32], is a notable improvement to EAs for optimizing data anonymization schemes. By integrating GA and DE, TLEF delivers enhanced performance. However, the simplicity of TLEF's hybrid strategy suggests that further refinement is possible.

In the domain of EAs, research has explored the adaptation of different strategies, particularly within DE [37, 38]. One such example is the Self-adaptive Differential Evolution algorithm (SaDE) [38], which introduces a novel approach where the learning strategy and control parameters need not be pre-specified. This adaptability enables SaDE to adjust its strategy dynamically during optimization, potentially resulting in even better performance.

3 Problem formulation

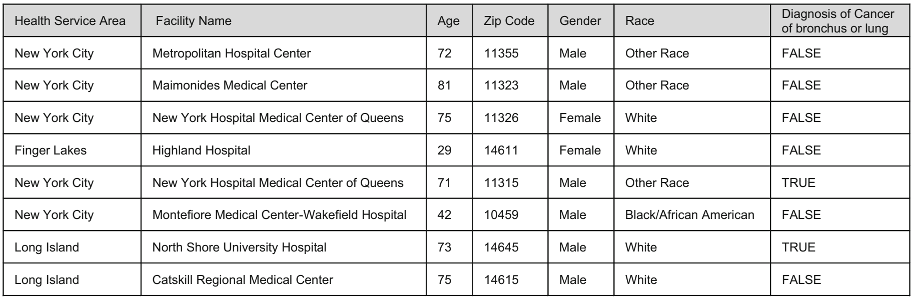
3.1 Data anonymization

When publishing data, it is common practice to anonymize it using various techniques (such as generalization and suppression) to convert the original dataset D into an anonymous dataset T. The original dataset, often in table format, typically contains explicit identifiers, quasi identifiers, sensitive attributes, and non-sensitive attributes.

- Explicit Identifiers: These are attributes that directly identify record owners. Examples include names, social security numbers, and email addresses.
- Quasi Identifiers (QID): These attributes can potentially reveal the identity of individuals when combined with other information. Examples include zip codes, birth dates, and gender.
- Sensitive Attributes (SA): These attributes contain private or sensitive information that needs to be protected. Examples include medical conditions, sexual orientation, and financial status.
- Non-Sensitive Attributes: These attributes include other information that is not considered sensitive. Examples include job titles, educational background, and hobbies.

When it comes to anonymizing data, the primary goal is to transform the $QIDs$ within the table into anonymized $QID^{'}s$ to eliminate the risk of data re-identification. Explicit identifiers, which directly identify individuals, are typically removed from the table altogether. Non-sensitive attributes, which lack sensitive information, do not require special handling and may remain unchanged. However, SAs contain essential information analysts need and are therefore retained.

The process of anonymizing data, known as $\mathcal{M}$, typically involves implementing techniques such as generalization, suppression, perturbation, data synthesis, etc. This paper adheres to the framework of IDGA [30] and ID-DGA [31], employing the strategies of



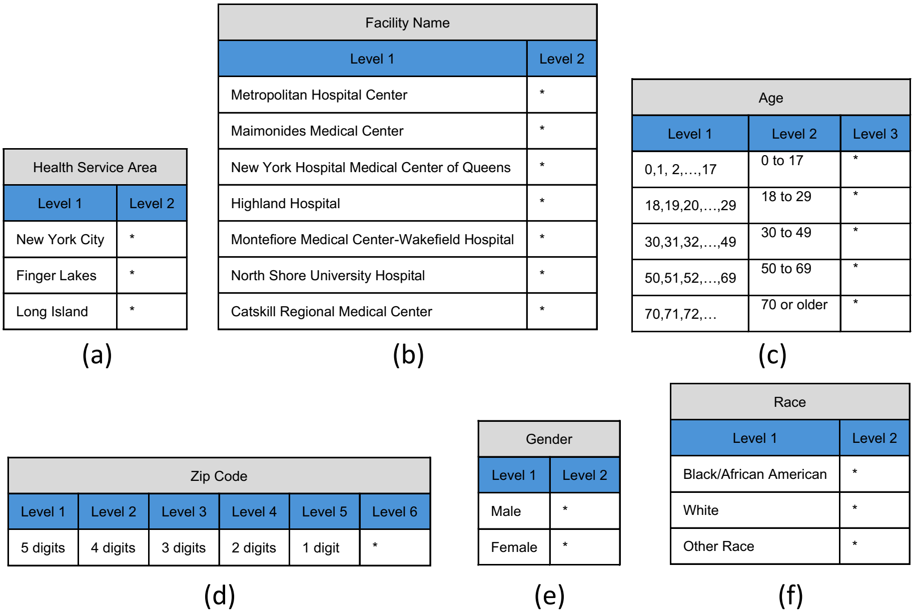
Health Service Area	Facility Name	Age	Zip Code	Gender	Race	Diagnosis of Cancer of bronchus or lung
New York City	Metropolitan Hospital Center	72	11355	Male	Other Race	FALSE
New York City	Maimonides Medical Center	81	11323	Male	Other Race	FALSE
New York City	New York Hospital Medical Center of Queens	75	11326	Female	White	FALSE
Finger Lakes	Highland Hospital	29	14611	Female	White	FALSE
New York City	New York Hospital Medical Center of Queens	71	11315	Male	Other Race	TRUE
New York City	Montefiore Medical Center-Wakefield Hospital	42	10459	Male	Black/African American	FALSE
Long Island	North Shore University Hospital	73	14645	Male	White	TRUE
Long Island	Catskill Regional Medical Center	75	14615	Male	White	FALSE

Figure 1 An example of original dataset with the generalization and suppression operations

generalization and suppression to achieve data anonymization, denoted as $\mathcal{M}\{G, S\}$. Generalization (G) entails substituting particular attribute values with broader or less precise ones, reducing the likelihood of re-identification, whereas Suppression (S) involves entirely removing specific records to prevent the disclosure of sensitive information. An example of an anonymized table is shown in Figure 3, which hides part information and makes privacy more difficult to be exposed.

$$\mathcal{D}\{QID, SA\} \xrightarrow{\mathcal{M}\{G,S\}} \mathcal{T}\{QID^{'}, SA\} \tag{1}$$

For example, the original table in Figure 1 initially contains six $QIDs$, one SA, and r records. To protect privacy, the table is anonymized using generalization (G) and suppression (S) techniques. As shown in Figure 3, the G operator is a sequence of natural numbers with a length of n, which designates the necessary level of generalization for each QID. Typically, G is used alongside a generalization hierarchy (illustrated in Figure 2), where each value



Health Service Area	
Level 1	Level 2
New York City	*
Finger Lakes	*
Long Island	*

(a)

Facility Name	
Level 1	Level 2
Metropolitan Hospital Center	*
Maimonides Medical Center	*
New York Hospital Medical Center of Queens	*
Highland Hospital	*
Montefiore Medical Center-Wakefield Hospital	*
North Shore University Hospital	*
Catskill Regional Medical Center	*

(b)

Age		
Level 1	Level 2	Level 3
0,1, 2,…,17	0 to 17	*
18,19,20,…,29	18 to 29	*
30,31,32,…,49	30 to 49	*
50,51,52,…,69	50 to 69	*
70,71,72,…	70 or older	*

(c)

Zip Code					
Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
5 digits	4 digits	3 digits	2 digits	1 digit	*

(d)

Gender	
Level 1	Level 2
Male	*
Female	*

(e)

Race	
Level 1	Level 2
Black/African American	*
White	*
Other Race	*

(f)

Figure 2 An example of generalization hierarchies

corresponds to a generalization level in the taxonomy tree [11]. Another operator, S, is a sequence consisting of 0s and 1s with a length of 8 (the same as the number of records in the original table), where zero indicates a record to be deleted, and one suggests a record to be retained (Figure 3).

3.2 Privacy model

In PPDP, data to be released must first undergo a privacy assessment to ensure proper anonymization before it can be released. Only anonymized data that meets privacy requirements is considered qualified for release. This paper employs the t-closeness privacy model for analysis, which aims to establish a certain level of similarity between the distributions of sensitive attributes in both the original and anonymized datasets. Please refer to Definition 1 for a more detailed explanation.

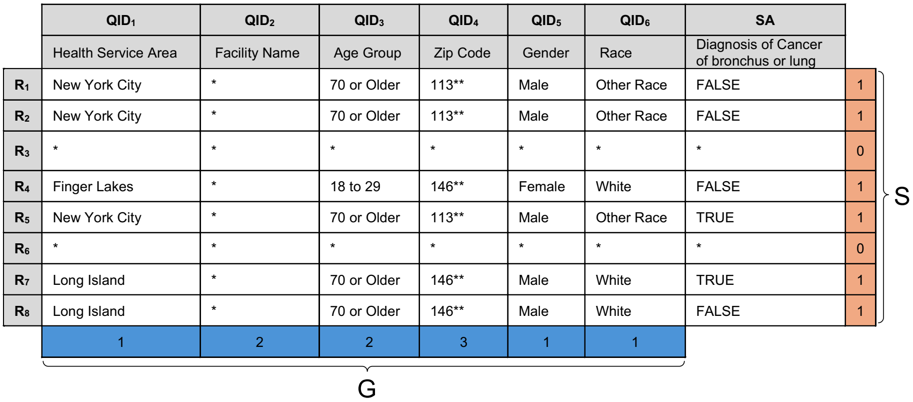
Definition 1 (The t-closeness Principle). *An equivalence class is said to have t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold t. A table is said to have t-closeness if all equivalence classes have t-closeness.*

Given a distribution of equivalence class $S = (s_1, s_2, \cdots, s_m)$, and the distribution of the whole table $Q = (q_1, q_2, \ldots, q_m)$, one well-known way to define the distance between the distribution is Euclidean distance:

$$D[S, Q] = \sqrt{(s_1 - q_1)^2 + (s_2 - q_2)^2 + \cdots + (s_m - q_m)^2} \tag{2}$$

where $(s_1, s_2, \cdots, s_m)$ and $(q_1, q_2, \cdots, q_m)$ represent the proportion of a sensitive attribute in the corresponding class and the whole table.

It is noteworthy that regardless of whether the original data contains one single or multiple sensitive attributes, the calculation for t-closeness of a dataset remains consistent. In the case of a single sensitive attribute, the proportional distribution of each value within the sensitive attribute can be directly computed. In scenarios involving multiple attributes, it becomes necessary to calculate the proportional distribution of combinations of values across all sensitive attributes.



	QID$_1$	QID$_2$	QID$_3$	QID$_4$	QID$_5$	QID$_6$	SA	
	Health Service Area	Facility Name	Age Group	Zip Code	Gender	Race	Diagnosis of Cancer of bronchus or lung	
R$_1$	New York City	*	70 or Older	113**	Male	Other Race	FALSE	1
R$_2$	New York City	*	70 or Older	113**	Male	Other Race	FALSE	1
R$_3$	*	*	*	*	*	*	*	0
R$_4$	Finger Lakes	*	18 to 29	146**	Female	White	FALSE	1
R$_5$	New York City	*	70 or Older	113**	Male	Other Race	TRUE	1
R$_6$	*	*	*	*	*	*	*	0
R$_7$	Long Island	*	70 or Older	146**	Male	White	TRUE	1
R$_8$	Long Island	*	70 or Older	146**	Male	White	FALSE	1
	1	2	2	3	1	1		

Figure 3 An example of anonymized dataset

For an anonymized dataset (T) containing l equivalent classes, its Anonymity Degree $AD(T)$ is defined by the furthest class, as shown in (3).

$$AD(T) = Max(D[S_1, Q], D[S_2, Q], \ldots, D[S_l, Q]) \tag{3}$$

Therefore, according to Definition 1, if an anonymized dataset has t-closeness , then:

$$AD(T) \leq t \tag{4}$$

3.3 Utility metrics

The utility of dataset T is calculated according to its Transparency Degree (TD) [11], which implication is how much useful information remains in the released data after suppression and generalization:

$$TD(T) = \sum_{r' \in T} \sum_{v_g \in r'} \frac{1}{\left|v_g\right|} \tag{5}$$

where r' indicates records remained in T after the suppression process; v_g is the generalized value in record r'; $\left|v_g\right|$ is the number of domain values that are descendants of v_g.

3.4 Optimal anonymization

Optimal anonymization describes the solution that results in minimal information loss according to a given metric [36, 39]. In the specific context of this paper, it is defined as follows.

Definition 2 (Optimal anonymization). *For anonymized dataset T, the optimal anonymization solution can satisfy the privacy requirement $AD(T) \leq t$ and achieves the highest utility degree $Max(TD(T))$.*

As the study [26] demonstrated, for every constant t, this is an NP-hard problem. In other words, finding the optimal solution is computationally expensive or impossible. A second best approach is to find a relatively optimal solution within limited computational time $(maxEvaluationTime)$.

Therefore, the optimization problem to be solved in this paper is to find the most efficient combination of suppression and generalization solutions $\mathcal{M}\{G, S\}$ within the $maxEvaluationTime$ to maximize the data utility metric $TD(T)$ within a given t-closeness threshold t.

$$\begin{cases} \text{Max}(TD(T)) \\ AD(T) \leq t \\ EvaluationTime \leq \text{max}EvaluationTime \end{cases} \tag{6}$$

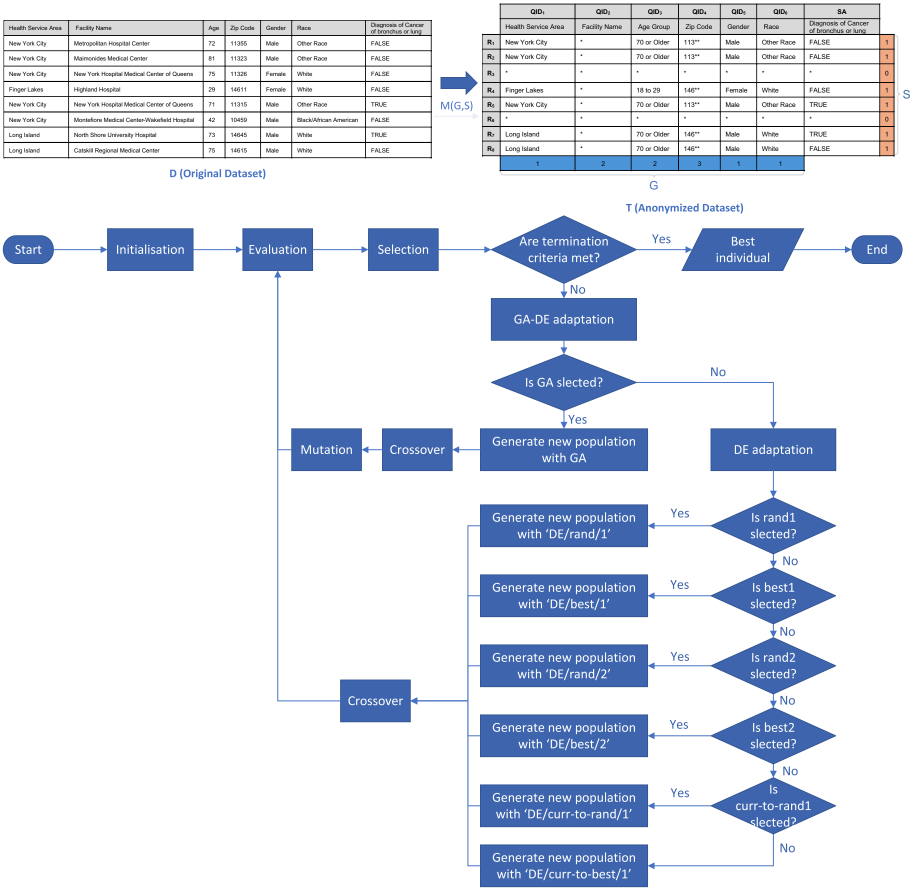
4 Hierarchical adaptive evolution framework

The HAEF algorithm utilizes a two-hierarchical adaptive architecture that combines the benefits of GA and DE variant algorithms. GA utilizes natural selection and genetics to generate promising solutions, while DEs offer potent search mechanisms and exploration capabilities. By harnessing the power of both techniques, HAEF can effectively navigate the solution space and uncover optimal solutions that enhance optimization. Furthermore, the innovative adaptive strategy of HAEF prioritizes using GA and random-based-DE mutation strategies during the early stages of iteration to broaden the search space. Later, it utilizes best-based-DE strategies to approach the global optimum, resulting in a more efficient and effective optimization process.

4.1 Workflow of HAEF

The overall process of HAEF is illustrated in Figure 4. It is a method that anonymizes original data using two technique s: generalization and suppression. As mentioned in Section 3.1, the generalization operator is a list of natural numbers signifying the levels of the generalization hierarchy (Figure 2) to which the values of QID should be mapped. The suppression operator is represented as a binary sequence of values (0 and 1) equal in length to the number of records in the dataset. ‘0’ indicates the removal of the data in the corresponding row, while ‘1’ denotes the retention of data in the respective record.

Given a G and an S, a table D’s anonymization method is established. By calculating the AD and TD of the anonymized table T, the effectiveness of this anonymization scheme can be evaluated. Taking the $M\{G, S\}$ in Figure 3 as an example, the anonymized table



Health Service Area	Facility Name	Age	Zip Code	Gender	Race	Diagnosis of Cancer of bronchus or lung
New York City	Metropolitan Hospital Center	72	11355	Male	Other Race	FALSE
New York City	Maimonides Medical Center	81	11323	Male	Other Race	FALSE
New York City	New York Hospital Medical Center of Queens	75	11326	Female	White	FALSE
Finger Lakes	Highland Hospital	29	14611	Female	White	FALSE
New York City	New York Hospital Medical Center of Queens	71	11315	Male	Other Race	TRUE
New York City	Montefiore Medical Center-Wakefield Hospital	42	10459	Male	Black/African American	FALSE
Long Island	North Shore University Hospital	73	14645	Male	White	TRUE
Long Island	Catskill Regional Medical Center	75	14615	Male	White	FALSE

D (Original Dataset)

M(G,S)

	QID₁	QID₂	QID₃	QID₄	QID₅	QID₆	SA	
	Health Service Area	Facility Name	Age Group	Zip Code	Gender	Race	Diagnosis of Cancer of bronchus or lung	
R₁	New York City	*	70 or Older	113**	Male	Other Race	FALSE	1
R₂	New York City	*	70 or Older	113**	Male	Other Race	FALSE	1
R₃	*	*	*	*	*	*	*	0
R₄	Finger Lakes	*	18 to 29	146**	Female	White	FALSE	1
R₅	New York City	*	70 or Older	113**	Male	Other Race	TRUE	1
R₆	*	*	*	*	*	*	*	0
R₇	Long Island	*	70 or Older	146**	Male	White	TRUE	1
R₈	Long Island	*	70 or Older	146**	Male	White	FALSE	1
	1	2	2	3	1	1		

T (Anonymized Dataset)

Start
Initialisation
Evaluation
Selection
Are termination criteria met?
Yes
Best individual
End
No
GA-DE adaptation
Is GA slected?
No
Yes
Generate new population with GA
Crossover
Mutation
DE adaptation
Is rand1 slected?
Yes
Generate new population with ‘DE/rand/1’
No
Is best1 slected?
Yes
Generate new population with ‘DE/best/1’
No
Is rand2 slected?
Yes
Generate new population with ‘DE/rand/2’
No
Is best2 slected?
Yes
Generate new population with ‘DE/best/2’
No
Is curr-to-rand1 slected?
Yes
Generate new population with ‘DE/curr-to-rand/1’
No
Generate new population with ‘DE/curr-to-best/1’
Crossover

Figure 4 The overall work process of HAEF

contains 6 records. After generalizing the QIDs, these records are grouped into three groups: (R_1, R_2, R_5); (R_4); (R_7, R_8).

As long as the anonymized table T meets the privacy constraints (4), scheme M satisfies PPDP. However, the optimal anonymization not only needs to satisfy the privacy constraints but also requires finding a scheme that maximizes TD.

In order to seek the optimal solution, HAEF employs a hybrid approach combining EA and DE. Initially, the algorithm generates an initial population comprising two subpopulations: population a represents schemes for generalization (G), while population b represents schemes for suppression (S). The size of the population is denoted by NP, indicating the number of individuals within the population. Subsequently, it is imperative to compute and validate the privacy and data transparency of each anonymized table generated by every individual in the population. Following the attainment of fitness values, a selection process is conducted among the individuals within the population. This selection method adheres to the criteria proposed in IDGA [30] and ID-DGA [31]. The best-performing individual selected is then utilized for subsequent evolution or is directly outputted as the determined optimal anonymization scheme.

The algorithm iteratively searches for the optimal solution within the predefined maximum validation iteration range. HAEF incorporates a dual-layer adaptive scheme, intelligently leveraging the advantages of GA, DE, and the variants of DE. In the hierarchy of GA-DE adaptation, HAEF employs a probabilistic approach to randomly select either the GA or DE method for the evolution of the new generation. The probability of selecting GA or DE is determined by the ratio of successful offspring generated by GA and DE in previous generations, which proceed to the subsequent round of evolution. Additionally, HAEF incorporates a GA-prioritized strategy, wherein GA is given higher probability usage during the initial stages of evolution, and DE is favored with higher probability towards the later stages.

Upon selecting GA, two individuals are randomly chosen from the population to serve as parents. These parents then undergo crossover, wherein segments of their genetic material are exchanged, generating offspring. Subsequently, the mutation is applied to randomly alter portions of the genetic makeup of some newly generated individuals, forming a new generation of individuals. Through validation and selection of the new individuals, those that outperform their parent individuals are retained, while those that do not meet the criteria are eliminated. This process results in the formation of the next generation of the population.

When DE is selected, HAEF incorporates an adaptation layer, intelligently choosing the mutation strategy for the DE layer based on probabilities. The selection probabilities of all mutation strategies are determined by the ratio of successful offspring generated by each mutation strategy in past generations to the total number of offspring generated by DE. Furthermore, HAEF implements a random-prioritized strategy, favoring the use of random-based DE mutation strategies in the initial stages of evolution and shifting towards the use of best-based-DE mutation strategies in the later stages.

Once one of the six mutation strategies is selected, each individual in the original population will generate a new individual accordingly. Subsequently, this new individual will exchange portions of its elements with the original individual through crossover. The fitness of each new individual is then validated, and these individuals are compared with the original individuals. If a new individual outperforms the original individual, it replaces the original individual; otherwise, it is discarded. This process forms a new population generated by DE for the next generation.

When the maximum validation iteration is reached, the HAEF ends and outputs the current best anonymization scheme.

4.2 GA-DE adaptation hierarchy

As shown in Algorithm 1, the HAEF follows the general procedure of an EA. Firstly, in Step 2, a $Population$ $P_0\{X_{i,0}|i = 1, 2, \cdots, NP\}$ contains NP $Individual$s is initialized. Each $Individual$ includes two parts: $X^a_{i,0}$ is for method of generalization $\mathcal{M}\{G\}$ and $X^b_{i,0}$ is for method of suppression $\mathcal{M}\{S\}$.

$$X^a_{i,0} = (x^a_{1,i,0}, x^a_{2,i,0}, \cdots, x^a_{j,i,0})|i = 1, 2, \cdots, NP \tag{7}$$

$X^a_{i,0}$ is a list contains j natural numbers which elements are randomly generated according to a uniform distribution $0 \leq x^a_{j,i,0} \leq x^{up}_j$, where x^{up}_j is the highest generalization hierarchy level of the jth attribute. For $\mathcal{M}\{G\}$, $j = 1, 2, \cdots, n$, where n is the number of QID attributes.

$$X^b_{i,0} = (x^b_{1,i,0}, x^b_{2,i,0}, \cdots, x^b_{j,i,0})|i = 1, 2, \cdots, NP \tag{8}$$

$X^b_{i,0}$ is randomly generated list according to a uniform distribution $x^b_{j,i,0} \in \{0, 1\}$. For $\mathcal{M}\{S\}$, $j = 1, 2, \cdots, r$, where r is the number of entries recorded in table D.

In the following Step 3, HAEF calculate the $AD(T)$ and $TD(T)$ of each $Individual$ in P_0 with (3) and (5). Then, following the selection criteria introduced in [30], the best $Individual$ in P_0 ($X_{best,0}$) is selected in Step 4 for the later process.

From Steps 6 to 16, HAEF enters a $while$ loop for evolutionary operations. In the loop, firstly, HAEF performs the first adaptation hierarchy deciding either GA or DE evolutionary strategy will be applied to update the current population $P_g\{X^a_{i,g}, X^b_{i,g}\}$ (shown in Step 9). HAEF randomly selects GA or DE according to its corresponding probabilities p_{ga} and p_{de} (Step 9). If GA is selected, HAEF will perform Algorithm 2 (Steps 10-11). Otherwise, DE is selected and Algorithm 3 will work (Steps 12-13).

The value of g representing the current generation will increase by 1 each time through the loop. The loop will be broken when $g \leq \gamma$ where γ is the pre-defined max generation.

Algorithm 1 Outline of HAEF.

```
1: Input: $NP$, $\gamma$, $CR_{de}$, $F$, $CR_{ga}$, $MR_{ga}$
2: Initialize population $P_0\{X^a_{i,0}, X^b_{i,0}\}$ with size $NP$
3: Evaluate fitness for each $Individual$ in $P_0$ according to (3) and (5)
4: Select the $X_{best,0}$ in $P_0$
5: Set generation counter $g = 0$
6: while $g < \gamma$ do
7:   Update $p_{ga}$ and $p_{de}$ every $UpdateInterval$ generation according to (9) and (10)
8:   Update $p^1_{de}, p^2_{de}, p^3_{de}, p^4_{de}, p^5_{de}, p^6_{de}$ every $UpdateInterval$ generation
9:   Select EA according probability $p_{ga}$ and $p_{de}$
10:  if $SelectedEA = GA$ then
11:    Update $P_g\{X^a_{i,g}, X^b_{i,g}\}$ with Algorithm 2
12:  else
13:    Update $P_g\{X^a_{i,g}, X^b_{i,g}\}$ with Algorithm 3
14:  end if
15:  $g = g + 1$
16: end while
17: Output: the best anonymization solution $X_{best,\gamma}$ found in $P_\gamma$
```

4.2.1 GA-prioritized adaptation

GA and DE are commonly utilized, each with distinct advantages and drawbacks. GA excels at preserving a varied population during optimization, efficiently exploring the search space and minimizing the risk of local optima. It achieves this through a crossover operation that blends information from multiple candidate solutions to generate novel solutions, thereby uncovering new areas of the search space and potentially discovering superior solutions. While DE may struggle to balance exploration and exploitation, GA tackles these challenges adeptly with its crossover operations. Additionally, DE can prematurely converge to suboptimal solutions in complex multimodal optimization problems.

Hence, this paper designs a GA-DE adaptation hierarchy to dynamically combine GA and DE, predominantly employed in the early stages of evolution to preserve population diversity and avoid premature convergence to the local optimum. Conversely, DE dominates the middle and late stages of evolution to approach the global optimum quickly.

In the GA-DE adaptation hierarchy, HAEF adopts a GA-prioritized adaptation strategy. For the first evolution, the EA strategy is chosen according to the initial probability $p_{ga}^0 = 1$ and $p_{de}^0 = 0$. Then p_{ga} and p_{de} are updated every $UpdateInterval$ generations according to (9) and (10) (shown in Step 7), where $UpdateInterval$ is a pre-defined parameter.

$$p_{de} = \frac{1}{2}\left(\frac{ns_{de} \cdot (ns + nf_{ga})}{ns_{ga} \cdot (ns_{de} + nf_{de}) + ns_{de} \cdot (ns_{ga} + nf_{ga})} + \frac{g}{\gamma}\right) \tag{9}$$

$$p_{ga} = \frac{1}{2}\left(\frac{ns_{ga} \cdot (ns_{de} + nf_{de})}{ns_{ga} \cdot (ns_{de} + nf_{de}) + ns_{de} \cdot (ns_{ga} + nf_{ga})} + 1 - \frac{g}{\gamma}\right) \tag{10}$$

where ns_{de} and ns_{ga} are, respectively, the counts of successful trial vectors generated by the DE and GA during the $UpdateInterval$ generation. Correspondingly, nf_{de} and nf_{ga} are the count of discarded trial vectors generated by DE and GA.

The algorithm stops when it reaches the maximum number of generations. The best anonymization solution, labeled as $X_{best,\gamma}$, is determined from the population P_γ. This solution is the most effective configuration found by the HAEF and provides optimal anonymization for the given dataset.

4.2.2 GA

This paper presents an implementation of a basic GA, detailed in Algorithm 2. The algorithm iterates through each pair of parents in the parental population using a for loop (from Step 2 to Step 9). For each pair, the algorithm performs a $crossover$ operation between the $father$ and $mother$, producing an offspring that inherits traits from both parents (Step 3). The crossover rate is regulated by CR_{ga}. Then, in Step 4, the offspring undergoes a $mutation$ operator that introduces small, random changes to its genetic composition. The proportion of random changes is controlled by MR_{ga}. In Step 5, the offspring's AD and TD are evaluated based on (3) and (5). If an offspring is deemed more competitive than any parent individual, that parent is replaced by the offspring (Steps 6-8). This process repeats until all parent individuals are considered. Once GA evolution is completed, the updated population P_{g+1} and the best individual $Individual_{best,g+1}$ of this round are returned to the mainstream.

Algorithm 2 Pseudo-code of GA.

```
Input: $P_g\{X^a_{i,g}, X^b_{i,g}\}$, $MR_{ga}$, $CR_{ga}$, $X_{best,g}$
for each two parent individuals in $P_g$ do
    Perform crossover operator on two individuals and generate offspring

    Execute mutation operator on offspring
    Evaluate offspring according to (3) and (5)
    if offspring is more competitive than one parent individual then
        Replace the parent individual with offspring
    end if
end for
Output: Updated population $P_{g+1}\{X^a_{i,g+1}, X^b_{i,g+1}\}$ and $X_{best,g+1}$
```

4.3 DEs adaptation hierarchy

As shown in Algorithm 3, DE uses varying mutation strategies during execution. As shown in Steps 2-20, through the for loop, each input individual of the previous generation enters a cycle of DE: $mutation$, $crossover$, and $selection$.

Algorithm 3 DEs Adaptation hierarchy.

```
1: Input: $P_g\{X^a_{i,g},X^b_{i,g}\}, CR_{de}, F, X_{best,g}$
2: for each individual in $P_g$ do
3:     Select DE mutation strategy according to $p^1_{de}, p^2_{de}, p^3_{de}, p^4_{de}, p^5_{de}, p^6_{de}$
4:     if $SelectedDE = DE/rand/1$ then
5:         Generate mutant vectors $V^a_{i,g}$, $V^b_{i,g}$ according to (14)
6:     else if $SelectedDE = DE/best/1$ then
7:         Generate mutant vectors $V^a_{i,g}$, $V^b_{i,g}$ according to (15)
8:     else if $SelectedDE = DE/rand/2$ then
9:         Generate mutant vectors $V^a_{i,g}$, $V^b_{i,g}$ according to (16)
10:    else if $SelectedDE = DE/best/2$ then
11:        Generate mutant vectors $V^a_{i,g}$, $V^b_{i,g}$ according to (17)
12:    else if $SelectedDE = DE/current-to-rand/1$ then
13:        Generate mutant vectors $V^a_{i,g}$, $V^b_{i,g}$ according to (18)
14:    else
15:        Generate mutant vectors $V^a_{i,g}$, $V^b_{i,g}$ according to (19)
16:    end if
17:    Crossover $u_i$ and $x_i$ according to (20)
18:    Evaluate fitness for $v_i$ according to (3) and (5)
19:    Perform selection operation according to (21)
20: end for
21: Output: Updated population $P_{g+1}\{X^a_{i,g+1}, X^b_{i,g+1}\}$, $X_{best,g+1}$
```

4.3.1 Random-prioritized adaptation

In DE, the mutation strategy plays a crucial role in exploring the search space and reaching the optimal solution. Random-based mutation strategies (e.g., DE/rand/1, DE/rand/2, DE/current-to-rand/1) and best-related mutation strategies (e.g., DE/best/1, DE/best/2, DE/current-to-best/1) differ in how they select candidate solutions for mutation.

The random-based mutation strategies facilitate the search space exploration by selecting random solutions from the population for mutation. This can help to escape from local optima and find diversified solutions. This approach is generally more robust to noise and disturbances in the objective function since it relies on various mutation candidate solutions. However, the convergence rate of these strategies may be slower than that of best-based strategies, especially in optimization problems where exploiting the best solution is crucial for convergence.

Best-based strategies focus on exploiting the best solutions in the population, leading to faster convergence to the optimal solution. These strategies tend to produce higher-quality solutions since they prioritize the best solution for mutation. However, the best-based strategies may prematurely converge to a sub-optimal solution, especially in problems where search space exploration is crucial.

In summary, DE mutation strategies should be chosen based on the optimization problem's requirements for exploration and exploitation. Random-based strategies are better for exploring extensive search spaces, while best-based strategies are better for converging toward the best solution. Therefore, this paper uses an adaptive hierarchy to select the DE mutation strategy dynamically. Random-based mutation strategies tend to be used more frequently in the early stage of the evolution process. In contrast, best-based mutation strategies tend to be used more regularly in the middle and late stages of evolution.

When updating the population with DE, each individual has the opportunity to select a predefined mutation strategy based on the corresponding probability (Step 3). In the first round, this probability is given by the initialization state. The subsequent probability p_{de}^{δ} is renewed every $UpdateInterval$ generations by the (11), (13) and (12) in the mainstream.

$$sm_{de}^{\delta} = \frac{ns_{de}^{\delta}}{(ns_{de}^{\delta} + nf_{de}^{\delta}) + \varepsilon} + \varepsilon \tag{11}$$

where $\delta = 1, 2, \cdots, 6$ represents a corresponding mutation strategy in (14)-(19). ns_{de}^{δ} and nf_{de}^{δ} are, respectively, the counts of successful and failed trial vectors corresponding to δ during $UpdateInterval$ generations.

This paper presents different evolutionary probability updating methods for random-based and best-based mutation strategies. The aim is to increase the usage of the 'random' strategy in the early stages of evolution and the 'best' strategy in the middle and late stages. There are three types of "random" based mutation strategy (refer to (14), (16), and (18)), and the corresponding probability of evolution $p_{DE}^{\delta}|\delta = 1, 3, 5$ is updated using (12). Similarly, there are three types of "best" based mutation strategies (refer to (15), (17), and (19)), and the corresponding probability of evolution $p_{DE}^{\delta}|\delta = 2, 4, 6$ is updated using (13).

$$p_{de}^{\delta} = \frac{1}{4}(\frac{sm_{de}^{\delta}}{sm_{de}^{1} + sm_{de}^{2} + sm_{de}^{3} + sm_{de}^{4} + sm_{de}^{5} + sm_{de}^{6}} + 1 - \frac{g}{\gamma})|\delta = 1, 3, 5 \tag{12}$$

$$p_{de}^{\delta} = \frac{1}{4}(\frac{sm_{de}^{\delta}}{sm_{de}^{1} + sm_{de}^{2} + sm_{de}^{3} + sm_{de}^{4} + sm_{de}^{5} + sm_{de}^{6}} + \frac{g}{\gamma})|\delta = 2, 4, 6 \tag{13}$$

4.3.2 Mutation operation

At generation g, DE employs the $mutation$ and $crossover$ operations to produce a *trial vector* $U_{i,g}$ for each *individual vector* $X_{i,g}$, also called *target vector*, in the current population.

For each *target vector* $X_{i,g}$ in generation g, an associated *mutant vector* $V_{i,g} = \left\{v_{1i,g}, v_{2i,g}, ..., v_{ni,g}\right\}$ can usually be generated by using one of the following 6 strategies:

"$DE/rand/1$":

$$V_{i,g} = X_{r_1,g} + F \cdot \left(X_{r_2,g} - X_{r_3,g}\right). \tag{14}$$

"$DE/best/1$":

$$V_{i,g} = X_{best,g} + F \cdot \left(X_{r_1,g} - X_{r_2,g}\right). \tag{15}$$

"$DE/rand/2$":

$$V_{i,g} = X_{r_1,g} + F \cdot \left(X_{r_2,g} - X_{r_3,g}\right) + F \cdot \left(X_{r_4,g} - X_{r_5,g}\right). \tag{16}$$

"$DE/best/2$":

$$V_{i,g} = X_{best,g} + F \cdot \left(X_{r_1,g} - X_{r_2,g}\right) + F \cdot \left(X_{r_3,g} - X_{r_4,g}\right). \tag{17}$$

"$DE/current - to - rand/1$":

$$V_{i,g} = X_{i,g} + F \cdot \left(X_{r_1,g} - X_{i,g}\right) + F \cdot \left(X_{r_2,g} - X_{r_3,g}\right). \tag{18}$$

"$DE/current - to - best/1$":

$$V_{i,g} = X_{i,g} + F \cdot \left(X_{best,g} - X_{i,g}\right) + F \cdot \left(X_{r_1,g} - X_{r_2,g}\right). \tag{19}$$

where indices r_1, r_2, r_3, r_4, r_5 are random and mutually different integers generated in the range $[1, NP]$, which should also be different from the current *trial vector*'s index i. F is a factor in (0, 2] for scaling differential vectors, and $X_{best,g}$ is the individual vector with the best fitness value in the population at generation g.

If the value of some elements in the newly generated *mutant vector* exceeds the corresponding upper and lower bounds, we deal it with the following rules. For population a, if $v^a_{j,i,g} < 0$, replace its value with 0; if $v^a_{j,i,g} > x^{up}_j$, replace its value with a random integer in $(0, x^{up}_j]$. For population b, if $v^b_{j,i,g} < 0$, randomly replace its value with 0 or 1; if $v^b_{j,i,g} > 1$, replace its value with1; if $0 \le v^b_{j,i,g} \le 1$, then randomly and uniformly replace its value with 0 or 1 with probability $v^b_{j,i,g}$.

4.3.3 Crossover operation

Subsequently, the "binominal" $crossover$ operation is performed between the generated *mutant vector* $V_{i.g}$ and its corresponding *target vector* $X_{i,g}$, resulting in a *trial vector* $U_{i,g} = \left(u_{1i,g}, u_{2i,g}, ..., u_{ni,g}\right)$.

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & if(rand_j[0,1] \le CR_{de}) or j = j_{rand}) \\ x_{j,i,g}, & otherwise \end{cases} \tag{20}$$

where CR_{de} is a user-specified crossover constant in the range [0, 1) and j_{rand} is a randomly chosen integer in the range $[1, NP]$ to guarantee that the *trial vector* $U_{i,g}$ will differ from its corresponding *target vector* $X_{i,g}$ by at least one element. For population a, j takes on the values of 1 to n, and for population b, j ranges from 1 to r.

4.3.4 Selection operation

Next, the *trial vectors*' fitness values are evaluated using (3) and (5). Then, a selection operation similar to the one described in [30] is performed. If neither the *trial vector* nor it's corresponding *target vector* meet the privacy preservation requirement, the $Individual$ with a higher AD is considered more competitive. If only the *trial vector* meets the requirement, it is considered more competitive. Finally, if both the *trial vector* and its corresponding *target vector* meet the privacy preservation requirements, the $Individual$ with a higher TD value replaces the *target vector* and enters the population of the next generation. The mathematical expression of this operation is as follows:

$$X_{i,g+1} = \begin{cases} U_{i,g}, & \text{if } AD(T(X_{i,g})) > AD(T(U_{i,g})) > t \\ U_{i,g}, & \text{if } AD(T(U_{i,g})) \leq t,\, AD(T(X_{i,g})) > t \\ U_{i,g}, & \text{if } AD(T(U_{i,g})) \leq t,\, AD(T(X_{i,g})) \leq t,\, TD(T(U_{i,g})) > TD(T(U_{i,g})) \\ X_{i,g}, & \text{otherwise} \end{cases} \tag{21}$$

This process continues until all *target vector*s in $P_g\{X^a_{i,g}, X^a_{i,g}\}$ have been considered. Finally, the updated population P_{g+1} and $X_{best,g+1}$ is output as the result of the algorithm, capturing the improvements made through the iteration process.

5 Experimental setup

This section describes the experimental setup we used in our study, including the dataset, hardware, software configurations, and specific steps taken to ensure reproducible and reliable results.

5.1 Dataset and test cases

This study employs the Hospital Inpatient Discharges 2015 database[1], which is an official database provided by the New York State Department of Health. The database consists of 34 attributes, including Health Service Area, Facility Name, Age Group, Zip Code, and more, and contains over two million records.

This paper extracted four datasets from the original database to validate the proposed approach. Each dataset has different characteristics in QIDs and SA. These datasets involve sensitive attributes such as whether a patient is at high risk for emergency hospitalization, diagnosed with a mental disorder, an HIV-infected individual, or diagnosed with bronchial or lung cancer.

Considering the potential impact of varying the number of QIDs, record counts, and the balance of sensitive attribute values on model performance, we randomly construct four test cases based on each dataset. These test cases differ in terms of the number of QIDs, record counts, and the proportion of classes for the sensitive attributes. For specific details, please refer to Table 1 to explore the number of sensitive attributes (nSA), the number of QIDs (nQID), and the record count (nR).

In Figure 5, we present an analysis of the balance/imbalance about sensitive attributes across each dataset. We examine diverse proportions of $Class0$ and $Class1$ for sensitive attributes: spanning from roughly balanced (D_9, D_{10}, D_{11}, D_{12}) to extremely imbalanced situations (D_{13}, D_{14}, D_{15}, D_{16}). Additionally, intermediate states are explored within the spectrum of D_1 to D_8.

5.2 Algorithm implementation

The algorithms in our study, including HAEF and the compared algorithms, are implemented in Python and executed on a local workstation running Windows 10 Pro. The workstation

[1] https://health.data.ny.gov/Health/Hospital-Inpatient-Discharges-SPARCS-De-Identified/82xm-y6g8

Table 1 Properties of 16 test cases

Dataset	Test case	$nQID$	nSA	nR
Dataset 1	D_1	6	1	300
	D_2	6	1	600
	D_3	10	1	300
	D_4	10	1	600
Dataset 2	D_5	6	1	300
	D_6	6	1	600
	D_7	10	1	300
	D_8	10	1	600
Dataset 3	D_9	5	1	300
	D_{10}	5	1	600
	D_{11}	9	1	300
	D_{12}	9	1	600
Dataset 4	D_{13}	6	1	300
	D_{14}	6	1	600
	D_{15}	10	1	300
	D_{16}	10	1	600

features an AMD Ryzen Threadripper PRO 3995WX CPU with 64 cores and a clock speed of 2.70 GHz, along with 256 GB RAM, providing ample computational resources for the experiments.

6 Experimental result

6.1 Comparison with existing approaches

To assess the efficacy of the proposed HAEF algorithm in achieving t-closeness, we conducted a series of experiments comparing it to three existing algorithms: DFS [36], ID-DGA [31],

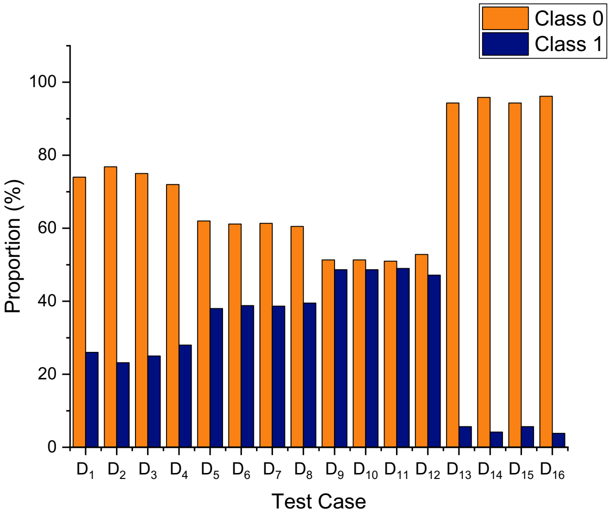


Figure 5 Proportion of $Class0$ and $Class1$ in the sensitive attribute of each test case

and TLEF [32]. DFS uses a traditional depth-first traversal search method, while ID-DGA and TLEF are advanced EAs developed for data anonymization.

The experiments included three different t-closeness thresholds: 0.1, 0.2, and 0.3. By testing the performance of each algorithm under varying thresholds, we are able to gain valuable insights into the effectiveness of the proposed method and how it compares to existing techniques.

6.1.1 Parameter settings

The experiments set the maximum fitness evaluation number as ten times the product of the number of quasi-identifiers and the number of records ($10 \times nQID \times nR$) for the above four methods. Please note that we do not care whether DFS has traversed all solutions; we only define its maximum number of evaluations to facilitate fair comparison with EA-based methods (ID-DGA, TLEF and HAEF). Therefore, the solution finally obtained by DFS may or may not be optimal.

The population size (NP) for all three EAs is set to 30. For ID-DGA, TLEF and HAEF, the GA related parameters crossover rate CR_{ga} was set to 0.5, while the mutation rate MR_{ga} was set to 0.2. For TLEF and HAEF, the DE-related parameters scaling factor (F) is set to 1.3, and the crossover rate (CR_{de}) is set to 0.3. In addition, parameter $UpdateInterval$ related to the update frequency of the HAEF adaptive strategy is set to be updated every ten generations. These parameter settings provided a standardized framework for evaluating the performance and behavior of each algorithm in our experiment.

6.1.2 *TD* comparison

The experimental results are summarized in Tables 2, 3, and 4, which showcase the performance of different algorithms for a specific number of evolutions. The TD values are achieved at t-closeness thresholds of 0.1, 0.2, and 0.3, respectively. Each table displays the average (Avg) and standard deviation (Std) of TD based on 25 independent runs for ID-DGA, TLEF and HAEF on 16 test datasets. The DFS only needs to be run once, and its single $Result$ is listed.

In the tables, bold text highlights the maximum TD mean for datasets. The † symbol indicates statistical significance test results using the Wilcoxon rank-sum test with a 0.05 level. Based on the results presented in the three tables, it is evident that the proposed HAEF algorithm has significant advantages under various privacy constraints.

Compared to the baseline algorithms DFS, ID-DGA, and TLEF, the proposed HAEF method demonstrates significant advantages under various privacy constraints. At $t = 0.1$, except for D_5, D_7, D_9, and D_{11}, HAEF achieves the highest average TD on 12 out of 16 test datasets. Additionally, HAEF statistically outperforms the control group in 9 of these test datasets (excluding D_3, D_6, and D_{15}). At $t = 0.2$, the HAEF algorithm achieves the highest average TD on 14 of 16 test datasets, except for D_5 and D_9. From a significance testing perspective, HAEF performs better than the baseline in 13 test datasets (excluding D_3). At $t = 0.3$, HAEF attains the highest TD mean on most test datasets (excluding D_5 and D_{14}) and exhibits significant advantages in 11 datasets (excluding D_1, D_3, D_9 and D_{11}). This underscores the substantial impact of HAEF's performance in the majority of the test datasets.

Figure 6 shows the overall performance (comparing the sum of TD values across 16 test cases) improvement of HAEF compared to the other three algorithms at different t values.

Table 2 TD comparison when t=0.1

Case	DFS [36]	ID-DGA [31]		TLEF [32]		HAEF	
	Result	*Avg*	*Std*	*Avg*	*Std*	*Avg*	*Std*
D_1	5.32E+02	5.32E+02	0.00E+00	6.66E+02	1.42E+00	**6.67E+02**†	1.25E+00
D_2	9.12E+02	9.26E+02	4.10E+01	1.20E+03	1.40E+02	**1.26E+03**†	1.26E+02
D_3	5.94E+02	6.16E+02	0.00E+00	6.44E+02	4.89E+01	**6.65E+02**	6.43E+01
D_4	1.06E+03	1.10E+03	0.00E+00	1.27E+03	4.39E+01	**1.32E+03**†	6.17E+01
D_5	3.84E+02	3.85E+02	6.64E-01	**4.42E+02**†	3.23E+01	4.03E+02	2.12E+01
D_6	9.19E+02	9.19E+02	0.00E+00	9.17E+02	1.32E+01	**9.27E+02**	1.41E+01
D_7	4.82E+02	5.04E+02	0.00E+00	**7.37E+02**†	2.63E+01	7.34E+02	3.24E+01
D_8	1.19E+03	1.23E+03	0.00E+00	1.38E+03	1.22E+02	**1.44E+03**†	8.71E+01
D_9	5.65E+02	5.65E+02	0.00E+00	**7.04E+02**†	9.38E-01	6.91E+02	2.54E+01
D_{10}	1.13E+03	1.15E+03	8.07E+01	1.42E+03	3.38E+00	**1.43E+03**†	2.16E+01
D_{11}	5.55E+02	5.79E+02	0.00E+00	**7.91E+02**	2.89E+01	7.88E+02	3.30E+01
D_{12}	1.12E+03	1.17E+03	0.00E+00	1.34E+03	8.93E+01	**1.42E+03**†	1.01E+02
D_{13}	7.95E+02	7.95E+02	0.00E+00	1.04E+03	2.42E+02	**1.16E+03**†	1.98E+02
D_{14}	1.51E+03	1.52E+03	4.84E+01	2.08E+03	1.38E+02	**2.29E+03**†	1.08E+02
D_{15}	1.01E+03	1.03E+03	1.47E+01	2.69E+03	4.34E+02	**2.83E+03**	1.96E+00
D_{16}	1.71E+03	2.14E+03	9.22E+00	2.80E+03	6.17E+02	**4.46E+03**†	1.33E+03

When $t = 0.1$, HAEF outperforms DFS, ID-DGA, and TLEF by 55.41%, 48.31%, and 11.75%, respectively; when $t = 0.2$, the improvements are 43.14%, 37.38%, and 7.91%, respectively; and when $t = 0.3$, HAEF increases 44.79%, 37.59%, and 5.39% respectively.

Table 3 TD comparison when t=0.2

Case	DFS [36]	ID-DGA [31]		TLEF [32]		HAEF	
	Result	*Avg*	*Std*	*Avg*	*Std*	*Avg*	*Std*
D_1	5.70E+02	5.74E+02	2.21E+01	6.83E+02	1.18E+01	**6.89E+02**†	1.34E+01
D_2	1.14E+03	1.14E+03	0.00E+00	1.34E+03	6.50E+01	**1.35E+03**†	2.48E+00
D_3	5.96E+02	6.54E+02	0.00E+00	6.66E+02	4.35E+01	**7.02E+02**	5.93E+01
D_4	1.18E+03	1.30E+03	0.00E+00	1.57E+03	6.13E+01	**1.58E+03**†	4.70E+01
D_5	5.32E+02	5.32E+02	0.00E+00	**5.64E+02**	3.43E+01	5.46E+02	1.51E+01
D_6	9.89E+02	9.90E+02	2.09E+00	1.19E+03	8.18E+01	**1.27E+03**†	6.11E+01
D_7	6.32E+02	6.54E+02	0.00E+00	7.57E+02	1.94E+01	**7.68E+02**†	1.44E+01
D_8	1.49E+03	**1.53E+03**†	0.00E+00	1.52E+03	1.78E+01	**1.53E+03**†	3.43E+00
D_9	**7.15E+02**	**7.15E+02**†	0.00E+00	7.05E+02	3.08E+01	7.14E+02	1.65E+00
D_{10}	1.13E+03	1.23E+03	1.34E+02	1.52E+03	6.74E+01	**1.56E+03**†	2.85E+01
D_{11}	6.30E+02	6.54E+02	0.00E+00	7.99E+02	3.85E+00	**8.02E+02**†	3.70E+00
D_{12}	1.56E+03	1.61E+03	0.00E+00	1.67E+03	6.88E+01	**1.73E+03**†	3.48E+01
D_{13}	8.32E+02	8.36E+02	2.14E+01	1.00E+03	1.46E+02	**1.14E+03**†	1.56E+02
D_{14}	1.66E+03	1.66E+03	1.79E+01	2.28E+03	1.32E+02	**2.36E+03**†	5.52E+01
D_{15}	1.08E+03	1.08E+03	3.39E+01	2.69E+03	4.28E+02	**2.83E+03**†	1.96E+00
D_{16}	2.01E+03	2.29E+03	0.00E+00	3.26E+03	7.60E+02	**4.40E+03**†	1.19E+03

Table 4 TD comparison when t=0.3

Case	DFS [36]	ID-DGA [31]		TLEF [32]		HAEF	
	Result	*Avg*	*Std*	*Avg*	*Std*	*Avg*	*Std*
D_1	5.72E+02	5.76E+02	2.16E+01	6.82E+02	2.67E+01	**6.84E+02**	8.21E+00
D_2	1.14E+03	1.14E+03	0.00E+00	1.40E+03	3.52E+01	**1.42E+03**†	2.01E+01
D_3	5.96E+02	6.54E+02	0.00E+00	6.59E+02	3.72E+01	**7.04E+02**	6.30E+01
D_4	1.33E+03	1.37E+03	1.78E+01	1.57E+03	6.79E+01	**1.59E+03**†	2.81E+00
D_5	**6.45E+02**	**6.45E+02**†	0.00E+00	6.23E+02	3.31E+01	5.86E+02	4.52E+01
D_6	1.14E+03	1.14E+03	0.00E+00	**1.37E+03**†	1.37E+01	**1.37E+03**†	1.61E+01
D_7	6.32E+02	6.51E+02	8.90E+00	7.91E+02	6.45E+01	**8.44E+02**†	7.17E+01
D_8	1.49E+03	1.54E+03	0.00E+00	1.56E+03	5.55E+01	**1.61E+03**†	5.85E+01
D_9	7.15E+02	7.15E+02	0.00E+00	7.15E+02	1.16E+01	**7.18E+02**	1.77E+01
D_{10}	1.43E+03	1.43E+03	0.00E+00	1.61E+03	1.26E+02	**1.68E+03**†	1.35E+02
D_{11}	7.80E+02	8.04E+02	0.00E+00	**8.07E+02**	1.28E+01	**8.07E+02**	4.13E+00
D_{12}	1.57E+03	1.61E+03	0.00E+00	1.77E+03	9.06E+01	**1.81E+03**†	1.00E+02
D_{13}	8.32E+02	8.49E+02	4.24E+01	1.15E+03	1.30E+02	**1.24E+03**†	1.27E+02
D_{14}	1.67E+03	1.70E+03	6.11E+01	**2.40E+03**†	1.01E+02	2.38E+03	2.80E+00
D_{15}	1.08E+03	1.14E+03	2.25E+01	**2.83E+03**†	4.49E+00	**2.83E+03**†	1.96E+00
D_{16}	2.02E+03	2.60E+03	6.98E+00	4.30E+03	8.96E+02	**5.27E+03**†	7.85E+02

Under the three privacy constraints, HAEF's overall performance compared to DFS, ID-DGA, and TLEF increased by an average of 47.78%, 41.09%, and 8.35%, respectively.

Looking at the overall trend, as t increases, the advantage of HAEF is diminished, especially when compared to TLEF. This is because a larger t value implies looser privacy constraints and lower optimization complexity. Both HAEF and TLEF use the same evolutionary methods, namely GA and DE. However, TLEF combines GA and DE using a

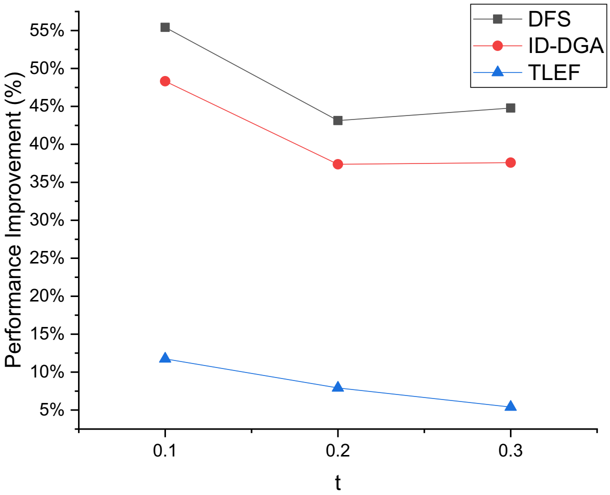


Figure 6 The overall performance improvement of HAEF over the other three baseline models at different t values

mechanical parity-interleaved approach, while HAEF intelligently selects the evolutionary path.

6.1.3 Convergence curves

This subsection presents convergence curves of DFS, ID-DGA, TLEF and HAEF at $t = 0.2$ in Figures 7 and 8. These curves offer valuable insights into the algorithms' performance and visually illustrate how the algorithms advance iteratively, giving us a better understanding of their convergence behavior and optimization capabilities.

As shown in Figures 7 and 8, each sub-figure features a legend in the lower right corner, showcasing the algorithms through distinct symbols and varying colors. The horizontal axis denotes the number of fitness evaluations (NFEs), while the vertical axis corresponds to the value of TD.

The convergence curves displayed in Figure 7 demonstrate multiple instances where HAEF outperforms other algorithms, such as D_4, D_6, D_7, D_8, D_{10}, D_{12}, D_{13}, D_{14} and D_{15}. In most cases, the results of HAEF and TLEF are significantly superior to DFS and ID-DGA, with HAEF displaying exceptional performance overall, particularly in D_6, D_{12}, D_{13}, and D_{15}. We also observe that HAEF lags significantly behind other algorithms in the initial iteration phase. For example, compared to TLEF, which uses GA and DE in a balanced manner at each stage, the convergence speed of HAEF is not dominant in the beginning stage of evolution. This may be due to the intensive updating of GA at the beginning of its evolution. While this

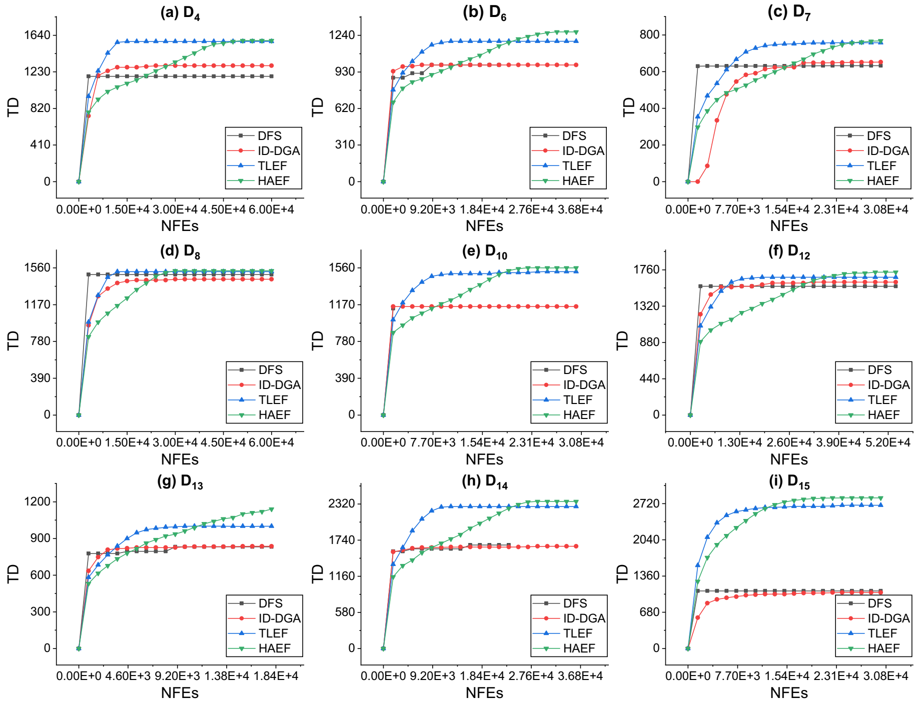


Figure 7 Comparison of the convergence curves of HAEF and comparison algorithms on the 9 test cases that achieve the maximum TD mean

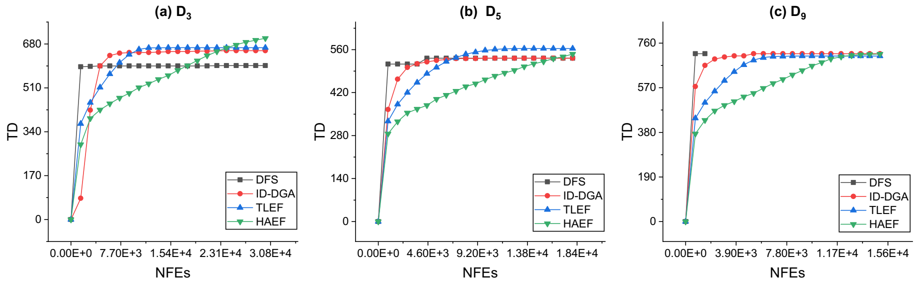


Figure 8 Comparison of the convergence curves of HAEF and comparison algorithms on three test cases that did not achieve the maximum TD mean

may slow the convergence rate, the overall algorithm is better suited to finding the global optimal solution, thanks to the GA algorithm's development capabilities.

In Figure 7, DFS shows fewer results in some subfigures (e.g., b, e, h). This is because DFS has traversed all solutions within the maximum number of evaluations to obtain the optimal solution.

Figure 8 shows the algorithm performance of HAEF on the datasets that did not achieve the maximum TD mean (D_5, D_9) or the significant best (D_3). It can be seen that although HAEF did not achieve the maximum value, it provided a final result comparable to other methods that achieved the maximum TD mean. This shows the robustness of the HAEF.

6.1.4 Population diversity

To further analyze the changes in the population diversity of the HAEF method during the evolution process, this section uses Euclidean distance and Hamming distance to represent the diversity of the G population and S population, respectively. A larger Euclidean distance or Hamming distance between two individuals of the population means that the difference between two individuals is more significant, and vice versa [40]. The sum of the Euclidean distance or Hamming distance between each pair of individuals in the population indicates the diversity of the population.

This paper calculates the changing trend of the diversity of the G and S populations of HAEF and the compared EAs (ID-DGA and TLEF) on each test set when t=0.2 as the number of fitness evaluations increases. As shown in Figure 9, three typical changing trends (D_3, D_6 and D_{16}) of each algorithm are listed. The subgraphs a, b, and c respectively represent the Euclidean distance of the G population of each algorithm on the test set D_3, D_6, and D_{16}, and the subgraphs d, e, and f represent the Hamming distance of the corresponding S population. As can be seen from the figure, regardless of population G or population S, the HAEF proposed in this paper can maintain higher population diversity than the compared EAs in the early and middle stages of evolution. A higher population diversity can enhance exploratory search and help identify promising search regions.

6.2 Ablation test

This subsection examines the efficacy of various strategies employed in HAEF using ablation experiments. Three comparison algorithms were obtained by partially preventing the relevant strategies from taking effect, whose performance is shown in Table 5. In the table, the bold

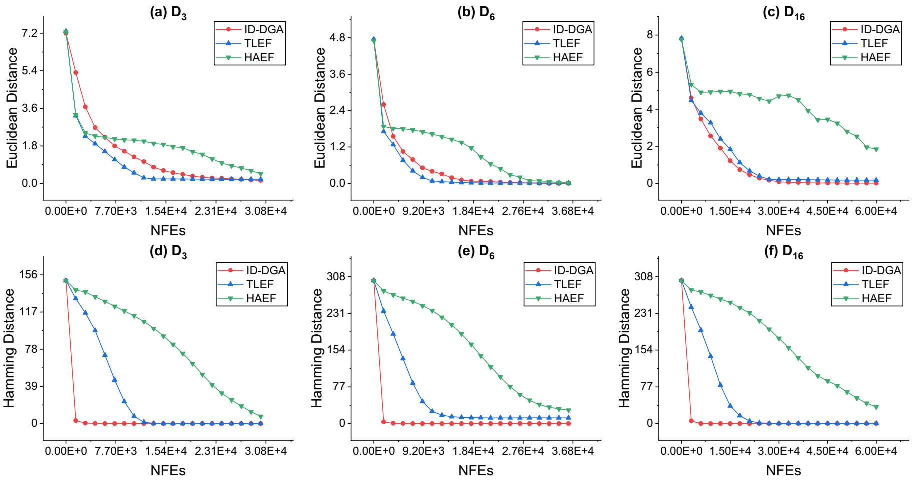


Figure 9 Comparison of the population diversity curves of HAEF and compared algorithms on three typical test cases

text highlights the maximum TD mean for datasets and the symbol † indicates the statistical significance test results using the Wilcoxon rank-sum test at the 0.05 level.

Among them, the 'without priority strategy' approach halts the GA-prioritized adaptation strategy in the GA-DE adaptation hierarchy and the Random-prioritized adaptation in the DEs adaptation hierarchy. The 'without GA-DE adaptation' scheme employs only 6 DEs for adaptive modifications. Finally, the 'without DEs adaptation' approach discards the other five variant DEs and solely utilizes the '$DE/best/1$' mutation strategy and GA adaptive evolution.

In the ablation test, HAEF achieved the highest average TD on 13 out of 16 test sets and had a significant advantage on 11. When comparing 'without GA-DE adaptation' and 'without DEs adaptation', both had advantages and disadvantages, but DEs are found to have more advantages than GA in solving problems of optimal t-closeness. Combining GA and DEs with a two-hierarchy adaptive strategy, the 'without priority strategy' scheme performed better on more datasets than the 'without GA-DE adaptation' scheme (which only uses DEs adaptation) and the 'without DEs adaptation' scheme. It achieved the optimal value and was the second most competitive solution. When comparing the first two sets of data in Table 5, HAEF performed better than 'without priority strategy' on most sets, indicating that the GA-prioritized and random-prioritized adaptation strategies proposed in the paper further improved the performance of the algorithm.

7 Implication and limitation

The implication of this paper is significant from both academic and industry perspectives. From the academic standpoint, the paper proposes a high-performance hierarchical adaptive evolution framework. Given its advantages in optimization performance and versatility, it is worth utilizing in other data-driven scenarios. Additionally, the proposed adaptive strategies can be embedded into existing state-of-the-art evolutionary algorithms for further improvement. For the industry, this paper showcases a practical and outstanding application of evolutionary computation in data anonymization, leading to the potential utilization of existing data publishing systems.

Table 5 Comparison of TD values across HAEF ablation trials when $t = 0.2$

Case	HAEF		Without priority strategy		Without GA-DE adaptation		Without DEs adaptation	
	Avg	*Std*	*Avg*	*Std*	*Avg*	*Std*	*Avg*	*Std*
D_1	**6.89E+02**†	1.34E+01	6.84E+02	1.04E+01	6.82E+02	8.75E+00	6.81E+02	9.55E+00
D_2	**1.35E+03**	2.48E+00	**1.35E+03**	4.69E+01	**1.35E+03**†	2.48E+00	1.34E+03	6.57E+01
D_3	**7.02E+02**	5.93E+01	6.99E+02	6.60E+01	6.80E+02	5.67E+01	6.72E+02	5.40E+01
D_4	**1.58E+03**†	4.70E+01	1.57E+03	6.90E+01	1.55E+03	8.52E+01	1.54E+03	8.28E+01
D_5	5.46E+02	1.51E+01	5.42E+02	3.01E+01	5.50E+02	2.14E+01	**5.62E+02**	3.74E+01
D_6	1.27E+03	6.11E+01	1.25E+03	7.19E+01	**1.28E+03**	4.55E+01	1.24E+03	8.05E+01
D_7	**7.68E+02**†	1.44E+01	7.53E+02	3.15E+01	7.56E+02	1.07E+01	7.56E+02	2.67E+01
D_8	**1.53E+03**†	3.43E+00	1.52E+03	6.83E+00	1.51E+03	4.34E+01	1.52E+03	8.82E+00
D_9	**7.14E+02**†	1.65E+00	7.07E+02	2.29E+01	**7.14E+02**	9.26E-01	7.04E+02	3.12E+01
D_{10}	**1.56E+03**†	2.85E+01	1.52E+03	6.71E+01	1.51E+03	7.18E+01	1.50E+03	7.12E+01
D_{11}	**8.02E+02**†	3.70E+00	7.95E+02	2.03E+01	7.91E+02	3.08E+01	7.94E+02	2.90E+01
D_{12}	**1.73E+03**†	3.48E+01	1.68E+03	6.82E+01	1.66E+03	6.70E+01	1.65E+03	7.06E+01
D_{13}	**1.14E+03**†	1.56E+02	1.04E+03	1.17E+02	1.04E+03	1.12E+02	1.05E+03	1.99E+02
D_{14}	**2.36E+03**†	5.52E+01	2.32E+03	1.01E+02	2.27E+03	1.34E+02	2.27E+03	1.29E+02
D_{15}	**2.83E+03**†	1.96E+00	2.80E+03	1.25E+02	**2.83E+03**	6.37E+00	2.82E+03	9.38E+00
D_{16}	4.40E+03	1.19E+03	4.40E+03	1.20E+03	**5.23E+03**	9.85E+02	3.48E+03	1.02E+03

While the proposed HAEF method in this paper has made significant progress compared to the baseline algorithms, one limitation is that it only considers the static privacy release of data, making it inapplicable in dynamic data publishing scenarios. A potential future work could be to adapt the proposed HAEF method to dynamic data systems.

8 Conclusion

In conclusion, the increasing demand for data release and the simultaneous need for privacy protection have emphasized the importance of PPDP. This problem remains challenging, especially with complex datasets with ineffective traditional traversal search methods. Although EAs show promise in tackling this challenge, they require refinement when applied to PPDP. This study introduces a new approach, the HAEF, which optimizes the t-closeness anonymity method by utilizing attribute generalization and record suppression. HAEF's innovative two-layered design, consisting of a GA-prioritized adaptive strategy in the first layer and a 'random' prioritized adaptive strategy in the second layer, enhances exploration and search capabilities. Benchmark tests demonstrate HAEF's superiority over traditional depth-first traversal search algorithms and the outperformance of existing algorithms like ID-DGA and TLEF on most datasets. Ablation experiments further confirm the effectiveness of various strategies within the framework. The proposed framework significantly improves data release efficiency, ensuring privacy, security, and maximum availability. Future research could explore extending HAEF's application to other privacy-preserving techniques and evaluating its scalability to larger datasets.

Author Contributions Mingshan You contributes to conceptualization, methodology, software, writing - original draft.
Yong-Feng Ge contributes to conceptualization, methodology, software, writing - original draft.
Kate Wang contributes to supervision, project administration, writing - review & editing.
Hua Wang contributes to validation, formal analysis, writing - review & editing.
Jinli Cao contributes to supervision, validation, writing - review & editing.
Georgios Kambourakis contributes to supervision and validation.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Availability of Data and Materials The data utilized in this paper is available at: https://health.data.ny.gov/Health/Hospital-Inpatient-Discharges-SPARCS-De-Identified/82xm-y6g8.

Declarations

Conflicts of Interest The authors declare that they have no conflict of interest.

Ethical Approval Not applicable.

Competing Interests The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Li, J.-Y., Zhan, Z.-H., Wang, H., Zhang, J.: Data-driven evolutionary algorithm with perturbation-based ensemble surrogates. IEEE Trans. Cybern **51**(8), 3925–3937 (2021). https://doi.org/10.1109/tcyb.2020.3008280
2. Wang, H., Cao, J., Zhang, Y.: Ticket-based service access scheme for mobile users. Austral. Comput. Sci. Comm. **24**(1), 285–292 (2002)
3. Wang, H., Sun, L.: Trust-involved access control in collaborative open social networks. In: 2010 Fourth International Conference on Network and System Security, pp. 239–246 (2010) IEEE
4. Yin, J., Tang, M., Cao, J., Wang, H., You, M., Lin, Y.: Vulnerability exploitation time prediction: an integrated framework for dynamic imbalanced learning. World Wide Web, 1–23 (2022)
5. Venkateswaran, N., Prabaharan, S.P.: An efficient neuro deep learning intrusion detection system for mobile adhoc networks. EAI Endorsed Trans. Scalable Inf. Syst **9**(6), 7–7 (2022)
6. Kabir, M.E., Wang, H., Bertino, E.: A role-involved purpose-based access control model. Inf. Syst. Front. **14**(3), 809–822 (2011). https://doi.org/10.1007/s10796-011-9305-1
7. Sun, X., Li, M., Wang, H., Plank, A.: An efficient hash-based algorithm for minimal k-anonymity. In: Conferences in Research and Practice in Information Technology, vol. 74, pp. 101–107 (2008)
8. Sun, X., Wang, H., Li, J., Pei, J.: Publishing anonymous survey rating data. Data Min. Knowl. Disc. **23**(3), 379–406 (2010). https://doi.org/10.1007/s10618-010-0208-4
9. Wang, H., Wang, Y., Taleb, T., Jiang, X.: Editorial: special issue on security and privacy in network computing. World Wide Web **23**(2), 951–957 (2019). https://doi.org/10.1007/s11280-019-00704-x
10. Cheng, K., Wang, L., Shen, Y., Wang, H., Wang, Y., Jiang, X., Zhong, H.: Secure k-NN query on encrypted cloud data with multiple keys. IEEE Trans. Big Data **7**(4), 689–702 (2017). https://doi.org/10.1109/tbdata.2017.2707552
11. Fung, B.C.M., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: a survey of recent developments. ACM Computing Surveys **42**(4), (2010) https://doi.org/10.1145/1749603.1749605
12. Lau, B.P.L., Marakkalage, S.H., Zhou, Y., Hassan, N.U., Yuen, C., Zhang, M., Tan, U.-X.: A survey of data fusion in smart city applications. Information Fusion **52**, 357–374 (2019) https://doi.org/10.1016/j.inffus.2019.05.004
13. Romero, C., Ventura, S.: Educational data mining and learning analytics: an updated survey. WIREs Data Mining and Knowledge Discovery **10**(3), (2020) https://doi.org/10.1002/widm.1355
14. Ge, Y.-F., Orlowska, M., Cao, J., Wang, H., Zhang, Y.: Knowledge transfer-based distributed differential evolution for dynamic database fragmentation. Knowl.-Based. Syst. **229**, 107325 (2021) https://doi.org/10.1016/j.knosys.2021.107325
15. Yin, J., Tang, M., Cao, J., Wang, H.: Apply transfer learning to cybersecurity: predicting exploitability of vulnerabilities by description. Knowl.-Based Syst. **210**, 106529 (2020)
16. Patil, D.R., Pattewar, T.M.: Majority voting and feature selection based network intrusion detection system. EAI Endorsed Trans. Scalable Inf. Syst. **9**(6), 6–6 (2022)
17. Sun, X., Li, M., Wang, H.: A family of enhanced (l, α)-diversity models for privacy preserving data publishing. Futur. Gener. Comput. Syst. **27**(3), 348–356 (2011). https://doi.org/10.1016/j.future.2010.07.007
18. Wang, H., Sun, L., Bertino, E.: Building access control policy model for privacy preserving and testing policy conflicting problems. J. Comput. Syst. Sci. **80**(8), 1493–1503 (2014). https://doi.org/10.1016/j.jcss.2014.04.017
19. Yang, J., Li, Y., Liu, Q., Li, L., Feng, A., Wang, T., Zheng, S., Xu, A., Lyu, J.: Brief introduction of medical database and data mining technology in big data era. J. Evid. Based Med. **13**(1), 57–69 (2020). https://doi.org/10.1111/jebm.12373
20. Zhu, T., Li, G., Zhou, W., Yu, P.S.: Differentially private data publishing and analysis: a survey. IEEE Trans. Knowl. Data Eng. **29**(8), 1619–1638 (2017). https://doi.org/10.1109/tkde.2017.2697856
21. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)
22. Lindell, Y.: Secure multiparty computation. Commun. ACM **64**(1), 86–96 (2020)
23. Acar, A., Aksu, H., Uluagac, A.S., Conti, M.: A survey on homomorphic encryption schemes: theory and implementation. ACM Computing Surveys (Csur) **51**(4), 1–35 (2018)

24. Ge, Y.-F., Yu, W.-J., Cao, J., Wang, H., Zhan, Z.-H., Zhang, Y., Zhang, J.: Distributed memetic algorithm for outsourced database fragmentation. IEEE Trans. Cybern. **51**(10), 4808–4821 (2021). https://doi.org/10.1109/tcyb.2020.3027962
25. Ge, Y.-F., Wang, H., Bertino, E., Zhan, Z.-H., Cao, J., Zhang, Y., Zhang, J.: Evolutionary dynamic database partitioning optimization for privacy and utility. IEEE Trans. Dependable and Secure Comp. (2023)
26. Liang, H., Yuan, H.: On the complexity of t-closeness anonymization and related problems. In: Database Systems for Advanced Applications: 18th International Conference, DASFAA 2013, Wuhan, China, 22-25 April 2013. Proceedings, Part I 18, pp. 331–345 (2013). Springer
27. Kesavan, V., Kamalakannan, R., Sudhakarapandian, R., Sivakumar, P.: Heuristic and meta-heuristic algorithms for solving medium and large scale sized cellular manufacturing system np-hard problems: a comprehensive review. Materials Today: Proceedings **21**, 66–72 (2020) https://doi.org/10.1016/j.matpr.2019.05.363 . International Conference on Recent Trends in Nanomaterials for Energy, Environmental and Engineering Applications
28. Pant, M., Zaheer, H., Garcia-Hernandez, L., Abraham, A.: Differential evolution: a review of more than two decades of research. Eng. Appl. Artif. Intell. **90**,103479 (2020)
29. Ge, Y.-F., Bertino, E., Wang, H., Cao, J., Zhang, Y.: Distributed cooperative coevolution of data publishing privacy and transparency. ACM Trans. Knowl. Discov. Data **18**(1), 1–23 (2023)
30. Ge, Y.-F., Wang, H., Cao, J., Zhang, Y.: An information-driven genetic algorithm for privacy-preserving data publishing. In: Web Information Systems Engineering–WISE 2022: 23rd International Conference, Biarritz, France, 1–3 November 2022, Proceedings, pp. 340–354 (2022). Springer
31. Ge, Y.-F., Wang, H., Cao, J., Zhang, Y., Jiang, X.: Privacy-preserving data publishing: an information-driven distributed genetic algorithm. World Wide Web **27**(1), 1 (2024)
32. You, M., Ge, Y.-F., Wang, K., Wang, H., Cao, J., Kambourakis, G.: Tlef: two-layer evolutionary framework for t-closeness anonymization. In: Web Information Systems Engineering–WISE 2023 24th International Conference, Melbourne, VIC, Australia, 25–27 October 2023, Proceedings, pp. 235–244 (2023). Springer
33. Sweeney, L.: k-anonymity: a model for protecting privacy. Internat. J. Uncertain. Fuzziness Knowl.-Based Syst. **10**(05), 557–570 (2002)
34. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. ACM Trans. Knowl. Discov. Data (TKDD) **1**(1), 3 (2007)
35. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: privacy beyond k-anonymity and l-diversity. In: 2007 IEEE 23rd International Conference on Data Engineering, pp. 106–115 (2006). IEEE
36. Kohlmayer, F., Prasser, F., Eckert, C., Kemper, A., Kuhn, K.A.: Flash: efficient, stable and optimal k-anonymity. In: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, pp. 708–717 (2012). IEEE
37. Zhang, J., Sanderson, A.C.: Jade: adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput. **13**(5), 945–958 (2009)
38. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 1785–1791 (2005). IEEE
39. Bayardo, R.J., Agrawal, R.: Data privacy through optimal k-anonymization. In: 21st International Conference on Data Engineering (ICDE'05), pp. 217–228 (2005). IEEE
40. Corriveau, G., Guilbault, R., Tahan, A., Sabourin, R.: Review and study of genotypic diversity measures for real-coded representations. IEEE Trans. Evol. Comput. **16**(5), 695–710 (2012)

Authors and Affiliations

Mingshan You[1] · Yong-Feng Ge[1] · Kate Wang[2] · Hua Wang[1] · Jinli Cao[3] · Georgios Kambourakis[4]

Mingshan You
mingshan.you@live.vu.edu.au

Kate Wang
kate.wang@rmit.edu.au

Hua Wang
hua.wang@vu.edu.au

Jinli Cao
J.Cao@latrobe.edu.au

Georgios Kambourakis
gkamb@aegean.gr

[1] Institute for Sustainable Industries and Liveable Cities, Victoria University, 70-104 Ballarat Road, Melbourne 3011, Victoria, Australia

[2] School of Health and Biomedical Sciences, RMIT University, 30 Janefield Drive, Bundoora 3083, Victoria, Australia

[3] Department of Computer Science and Information Technology, La Trobe University, Melbourne 3086, Victoria, Australia

[4] Department of Information and Communication Systems Engineering, University of the Aegean, Karlovasi 83200, Samos, Greece