Adaptive-parameter memetic algorithm for privacy-preserving trajectory data publishing: A multi-objective optimization approach

Computing (2025) 107:151
https://doi.org/10.1007/s00607-025-01504-0

SPECIAL ISSUE ARTICLE

Adaptive-parameter memetic algorithm for privacy-preserving trajectory data publishing: A multi-objective optimization approach

Samsad Jahan[1] · Yong-Feng Ge[1] · Hua Wang[1] · Enamul Kabir[2]

Received: 28 February 2025 / Accepted: 27 May 2025

Abstract

Trajectory data has grown pervasive, benefiting practical applications, including transportation administration and location-based operations. Nevertheless, trajectories may reveal extremely sensitive information about an individual, including movement patterns, personal profiles, geographical locations, and social contacts, necessitating privacy protection while disseminating trajectory data. Therefore, prioritizing privacy protection is crucial while analyzing trajectory data. Current methods of protecting privacy concentrate on single objective optimizing techniques such as maximizing data utility but often disregard various privacy constraints. To overcome this challenge, this study aims to improve both data privacy and usability by balancing competing goals-maximizing privacy while maintaining useful information-through a Multi-Objective Optimization (MOO) approach in trajectory data publishing. We provide a unique algorithm named Adaptive-Parameter Memetic Algorithm (APMA) that employs a non-dominated sorting multi-objective technique and a Memetic Algorithm (MA). This algorithm utilizes adaptive memory-based mutation and crossover strategies to dynamically adjust the mutation and crossover parameters and improve the solution's quality. The proposed innovative local search strategy helps to achieve better population diversity and solution quality. Comprehensive studies illustrate the efficacy of the proposed method regarding solution quality and convergence outcomes.

Keywords Trajectory data privacy · Memetic algorithm · Adaptive parameter · Privacy-utility trade-off · Multi-objective optimization

1 Introduction

The increasing prevalence of devices equipped with GPS capabilities has resulted in the development of numerous services based on geographical location, which monitor moving entities and generate a wealth of spatial trajectory data. Analyzing this

Extended author information available on the last page of the article

Published online: 17 June 2025

trajectory data is now a standard practice, demonstrated by various trajectory-associated methods that could improve numerous practical applications, including city planning, traffic oversight, and personalized suggestions [4, 22, 28, 46]. Trajectory data offers novel perspectives on human mobility, thus enabling public transportation systems and traffic pattern optimization. However, the complex nature of this data might compromise privacy by revealing private information like social contacts, routines, and locations of sensitive information [6]. People can be easily identified by their past movements through linking and re-identification attacks [5, 28, 43, 44]. Consequently, stringent privacy-preserving measures are needed to protect trajectory data while ensuring its usability.

Traditional privacy-preserving methods for trajectory or location data frequently fall short in optimizing data anonymity while preserving utility. In addition, these kinds of strategies have limitations in high-dimensional and sequential trajectory data scenarios. These approaches depend on greedy algorithms to optimize information, which can become trapped in local optima. Trajectory data privacy is typically improved using single objective optimization methods [16, 47, 53], focusing particularly on maximization of utility while frequently neglecting the broad spectrum of privacy protection needs [41]. These techniques face two challenges. First, they fail to accommodate the varied characteristics of trajectory data, potentially resulting in inferior performance in real-world circumstances. Second, they frequently employ static parameters inadequately suited for the privacy-utility trade-off necessary in Multi-Objective Optimization (MOO) tasks within the Privacy Preserving Trajectory Data Publishing (PPTDP) problem.

To overcome these obstacles, our study seeks to find solutions to the following critical research gaps:

Adaptive parameter adjustment Whereas traditional approaches depend on the fixed or static parameter, our study uses adaptive parameters that dynamically adjust the key parameters during evolutionary stages, which is efficient for finding better privacy-utility solutions. This mechanism also guarantees that the algorithms remain effective in various data scenarios.

Memetic algorithm integration Memetic Algorithm (MA) is proposed, drawing inspiration from natural evolution and the concepts of memes. It can be defined as an integration of population-based search methodologies and local improvement techniques. Earlier research on MA has shown that it tends to find a balance between exploring and exploiting throughout the optimization process [12]. It can be used to solve a lot of complex optimization tasks like automatic data clustering [36], the vehicle routing problem [45], and database fragmentation for enhancing privacy and utility [12]. However, the use of MA in improving the privacy and utility solutions in trajectory data still remains unexplored. In this study, we employ an MA framework featuring an adaptive memory mechanism and adaptive parameters to improve search capability and solution quality. We are the first to employ adaptive memory-based mutation and crossover in the context of trajectory data privacy, to the best of our knowledge.

Multi-objective optimization and evolutionary algorithms MOO has become a potential way to maximize both privacy and utility in trajectory data by means of a balance between the two. The Non-dominated Sorting Genetic Algorithm

(NSGA-II) [10] and Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) [54] have been used in privacy preserving data publishing [38]. These techniques might help to identify Pareto optimal solutions balancing utility and privacy. For instance, Tian et al. treated the location privacy model as MOO problem and combined it with particle swarm optimization techniques to balance privacy protection and service efficiency [39]. Zheng et al. employed a non-linear MOO strategy to guarantee data utility through a semantic-aware privacy-preserving mechanism in an online location-based trajectory data scenario [57]. However, most conventional models for protecting trajectory data privacy concentrate on single objective optimization, i.e., either maximizing privacy or maximizing utility. Therefore, our MOO approach maximizes both privacy and utility and avoids being trapped in a local optimum solution.

Evolutionary Algorithms (EAs), such as genetic algorithms (GAs) and MAs, have been widely employed in prior research to augment data privacy and refine the data anonymization [13–15, 17, 48–50, 56]. The high efficacy of EAs in safeguarding the privacy of trajectory data is indicated by their successful implementation in both MOO problem and privacy protection [3, 19, 35]. This article provides a framework that concurrently addresses privacy and utility maximization, conceptualizing it as a multi-objective PPTDP problem using an MA.

Here is a list of our contributions to this article: **(i)** we propose an Adaptive-Parameter Memetic Algorithm (APMA) that employs an MA with a non-dominated sorting MOO method to optimize the PPTDP problem to improve privacy-utility solutions for trajectory data; **(ii)** we employ a novel local search technique to enhance population quality; **(iii)** our proposed algorithm adaptively adjust the crossover and mutation parameter along with adaptive memory mechanism strategy to optimize the privacy-utility solutions; **(iv)** we conduct comprehensive experiments to validate the efficacy of our suggested algorithm in enhancing privacy, utility, and convergence results.

The organization of the subsequent sections of this work is as follows. Section 2 describes the related works. Section 3 delineates the preliminaries of trajectory data and the problem statement. Section 4 delineates our suggested algorithm. Section 5 delineates the experimental results, and Sect. 6 describes the experimental settings. In the end, Sect. 7 presents the conclusion.

2 Related work

Preserving the privacy of trajectory data and maintaining its utility is a challenging task. The commonly used privacy-preserving methods for trajectory data are *k*-anonymity [25, 29, 37, 41], *l*-diversity [33], *t*-closeness [32], and Differential Privacy (DP) [11]. Among these techniques, *k*-anonymity has achieved significant interest in the area of PPTDP. It mitigates re-identification attacks by guaranteeing the indistinguishability of an individual within a *k*-anonymous group while concurrently minimizing information loss, which refers to the extent of distortion necessary to obscure the individual within the group [27]. It has been demonstrated that achieving optimal k-anonymity is NP-hard [34]. Prominent implementations, such as NWA [1], W4M

[2], and GLOVE [20], are widely used benchmarks. NWA and W4M use a two-step greedy approach: first, they group k comparable trajectories; second, they anonymize every group. On the other hand, GLOVE computes trajectory-wise merge costs and then performs hierarchical clustering, where merge cost directly affects the utility and privacy. Extensions of k-anonymity try to further reduce information loss by using the minimum description length concept [31], timestamp coarsening [9], and tailored k values depending on sensitivity [24, 30].

Although k anonymity reduces re-identification risks and mitigates record linkage, group members with identical sensitive characteristics face concerns about attribute linkage. l-diversity [33] addresses this attribute linkage concerns by ensuring that there are at least l distinct sensitive values within each group. However, skewed or semantically similar distributions may jeopardize privacy. t-closeness [32] further refines protection by restricting the distance between group-level and global sensitive attribute distributions. KLT [41] uniquely integrates l-diversity and t-closeness in trajectory protection by adopting GLOVE's architecture and creating semantic zones based on points of interest (POIs), merging neighboring regions until diversity and closeness criteria are achieved. DP [11] limits the disclosure risk of individual data using a privacy budget ϵ. Representative DP techniques, including DPT [23], SPLT [7], and AdaTrace [21], model statistical characteristics of raw trajectories and generate synthetic datasets. These techniques vary in modeling, sampling, and noise mechanisms: DPT adds Laplace noise to distributions, SPLT preserves semantic similarity under DP restrictions, and AdaTrace enhances attack resilience while maximizing utility.

The MOO method, alongside EAs, has recently been utilized to achieve optimal anonymization solutions [25]. Tachioka's proposed method balances required utility with privacy, using MOO and NSGA-II to maintain privacy while optimizing parameter tuning [38]. However, Tachioka specifically applied his method within healthcare contexts. Jahan et al. [26] contributed significantly to this domain by introducing the Dynamic Parameter Genetic Algorithm (DPGA) for multi-objective trajectory data publication. This research argues that conventional static parameter sets often provide suboptimal outcomes, particularly in dynamic environments where attributes may change over time. Ge et al. developed the Federated Genetic Algorithm (FGA) in related research; it employs a two-layer optimization framework to highlight the privacy aspects of trajectory data and also incorporates a federated learning approach, thus allowing distributed data processing [19]. Moreover, Ge et al. [18] offer an Information-Driven Distributed Genetic Algorithm (ID-DGA), designed to attain optimum anonymization through attribute generalization. This method emphasizes the necessity of information-driven strategies in improving the efficacy of privacy-preserving approaches and suggests that a more informed approach to data anonymization might provide better results for data utility as well as privacy.

You et al. [50] introduce a Hierarchical Adaptive Evolution Framework (HAEF) designed to enhance t-closeness anonymization via attribute generalization and record suppression, employing GA and Differential Evolution (DE). The initial hierarchy of HAEF utilizes a GA-prioritized adaptive strategy to enhance exploration search, thereby balancing GA and DE. Within our framework, APMA focuses on MOO by incorporating an MA augmented with an adaptive crossover and mutation strategy, along with a local search strategy, thus yielding a superior solution for balancing privacy and utility. Table 1 compares existing privacy-preserving works using EAs with our proposed study.

3 Preliminaries of trajectory data

This section presents the definitions related to our PPTDP.

Definition 1 (*Uncertain trajectory* [40]) The trajectory of a moving object in a three-dimensional space can be defined as a sequence of spatio-temporal points $(x_1, y_1, t_1), (x_2, y_2, t_2), \dots, (x_n, y_n, t_n)$ within a time interval where $t_1 < t_2 < \cdots < t_n$. The object is presumed to move in a straight line from (x_i, y_i) to (x_{i+1}, y_{i+1}) at a constant speed during the interval $[t_i, t_{i+1}]$. An uncertain trajectory is the pair (τ, δ), where τ is a trajectory between times t_1 and t_n and δ is an uncertainty threshold.

Definition 2 (*Co-localization* [40]) Given a collection of trajectories, τ_1 and τ_2 are said to be co-localized with respect to a cylindrical volume of radius δ if there exist points $p_1 \in \tau_1$ and $p_2 \in \tau_2$ such that their Euclidean distance (ED) is less than or equal to δ. The ED is calculated as: $\mathrm{Dist}(x_1, y_1, x_2, y_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ where $p_1 = (x_1, y_1, t_1)$ and $p_2 = (x_2, y_2, t_2)$.

In our framework, the (k, δ)- anonymity criterion is set as the privacy model and is defined as follows:

Definition 3 ((k, δ)-*Anonymity* [2]) Achieving (k, δ)-anonymity involves transforming the dataset D into a new dataset T, where D is a dataset of trajectories, and δ and k are uncertainty and anonymity thresholds, respectively. Each trajectory τ' in this transformed dataset is a member of an anonymous set (k, δ). The goal is to minimize the distortion between the original dataset D and the transformed dataset T.

Table 1 Privacy preserving methods using EAs

Study	Implementation	Single objective optimization	MOO	EA	Local Search
Auto parameter tuning [38]	Health data	✗	✓	GA	✗
DPGA [26]	Trajectory data	✗	✓	GA	✗
FGA [19]	Trajectory data	✗	✓	GA	✗
ID-DGA [18]	Health Data	✓	✗	Distributed GA	✗
HAEF [50]	Health Data	✓	✗	GA and DE	✗
APMA	Trajectory Data	✗	✓	MA	✓

Definition 4 (*Translation distortion* [2]) Consider a translated version $\tau^{'}$ of τ within a dataset, where $\tau \in D$ and $\tau^{'} \in T$. The translation distortion cost from τ to $\tau^{'}$ is given by $TD(\tau, \tau^{'}) = \sum_{t \in T} ED(\tau[t], \tau^{'}[t])$. Additionally, the total distortion of the anonymized dataset T can be expressed as $TTD(D, T) = \sum_{\tau \in D} TD(\tau, \tau^{'})$.

3.1 Problem statement

The goal of the PPTDP problem is to transfer the original dataset D to an anonymous dataset T in a way that satisfies the highest privacy requirements set by a privacy model while maximizing its utility. The following case can be considered as the application of our proposed APMA. Consider the scenario of a busy metropolitan city, where three Uber drivers-D_1, D_2, and D_3-and three passengers-P_1, P_2, and P_3-are seeking efficient rides. To improve passenger satisfaction and minimize travel time, a central Uber service provider optimizes the assignment of drivers to passengers. To protect the drivers' privacy, they employ (k, δ) -anonymity to obfuscate their precise GPS locations, sharing only the obfuscated data with the service provider. The provider aims to maximize the privacy of both drivers and passengers while ensuring optimal utility. Each driver strives to minimize their travel distance while maximizing their privacy. In this context, our algorithm can offer better privacy and utility solutions, accommodating the varying privacy requirements of Uber drivers.

Our proposed algorithm take into consideration of maximizing two objective simultaneously. The two objectives are as follows:

Definition 5 (*Maximizing privacy degree*) The objective is to maximize the privacy degree (PD), which is achieved by enhancing the value of k across all possible anonymization solutions for an anonymized dataset D.

Definition 6 (*Maximizing utility degree*) The goal is to maximize the utility degree (UD) by minimizing the total distortion among all possible utility solutions for the anonymized dataset D.

In the PPTDP problem, trajectory data can be easily targeted by re-identification attacks [27]; hence, we use (k, δ)-anonymity as a way to protect privacy against this attack. This method can obscure individual trajectory data by abstracting and clustering similar paths based on identical routes or areas. It is notably more simplistic and direct compared to other privacy measures and is renowned for its robust generalization capabilities. So, the objective is to find the anonymization solution S that transforms D into T, ensuring it meets the (k, δ)-anonymity privacy standard. The parameters k and δ are determined by data publishers according to privacy requirements. An increased value of k and a decreased value of δ signify enhanced privacy protection. Furthermore, reducing the overall translation distortion, $\tau(D, T)$, provides enhanced utility.

3.2 Example: Privacy-preserving trajectory anonymization of taxi drivers

Let us consider an example of three taxi drivers, denoted as D_1, D_2, and D_3, operating within an urban environment. Every driver tries to find a balance between two criteria: keeping their location private (privacy) and minimizing their travel distance (utility). Each driver's original trajectory and translated trajectories across three timestamps are given below in Table 2, where X is the latitude, Y is the longitude, and T stands for the timestamp.

Assume that they form a cluster, and they will be space translated to create an anonymous set of trajectories. To enhance privacy, a centralized trajectory τ_c is computed based on the average locations at each timestamp: $\tau_c = (8, 3, 1), (7, 5.333, 2), (4, 5.6667, 3)$. When $\delta = 0$ (i.e., all trajectories are transformed into centralized trajectories), the *TTD* cost is computed by taking sum of all *TD*'s from the original and centralized trajectories, which is approximately 23.388. Each driver's trajectory is translated toward the centralized trajectory to obscure their original paths while preserving the general movement patterns.

Example of translated trajectory for D_1 at Timestamp 1: $(5 - 8,\ 4 - 3) = (-3, 1)$, which is considered as a direction vector. For $\delta = 1$ the point shifts from the cluster center to the disk perimeter $\frac{\delta}{2}$. Therefore, the translated trajectory for the D_1 and the first timestamp is calculated as follows: $(5 + 3 \times .5, 4 + (-1 \times 0.5) = (6.5, 3.5)$.

Likewise, translated trajectories are calculated across all timestamps for each driver. The *TTD* cost for the translated trajectories and centralized trajectories is now 11.76. The *TTD* cost for the translated trajectories is lower than the *TTD* cost of the original trajectories at the same level of privacy protection i.e. $k = 3$ and $\delta = 0$. In this example, we aim to satisfy the privacy criteria of $k = 3$ and $\delta = 1$. The original trajectories do not adhere to this criterion and potentially be re-identified by an attacker. However, by translating them into the new form, both privacy and utility can be concurrently achieved.

This example illustrates how trajectory centralization and transformation reduce distinguishability among users while maintaining reasonable privacy and

Table 2 Original and translated trajectories of drivers

Original Trajectory				Translated Trajectory		
Driver	X	Y	T	X	Y	T
D_1	5	4	1	6.5	3.5	1
	6	9	2	6.5	7.1665	2
	7	8	3	5.5	6.835	3
D_2	10	3	1	9	3	1
	8	6	2	7.5	5.665	2
	3	5	3	3.5	5.3335	3
D_3	9	2	1	7.5	2.5	1
	7	1	2	7	3.1665	2
	2	4	3	3	4.8335	3

usability. By minimizing *TTD*, this approach achieves a balance between preserving privacy and maintaining utility for service operations.

4 The proposed APMA

This section describes our proposed APMA that utilizes an EA with an adaptive memory mechanism and MOO approach. Our key objective is to maximize both privacy and utility degrees.

4.1 Representation of individual

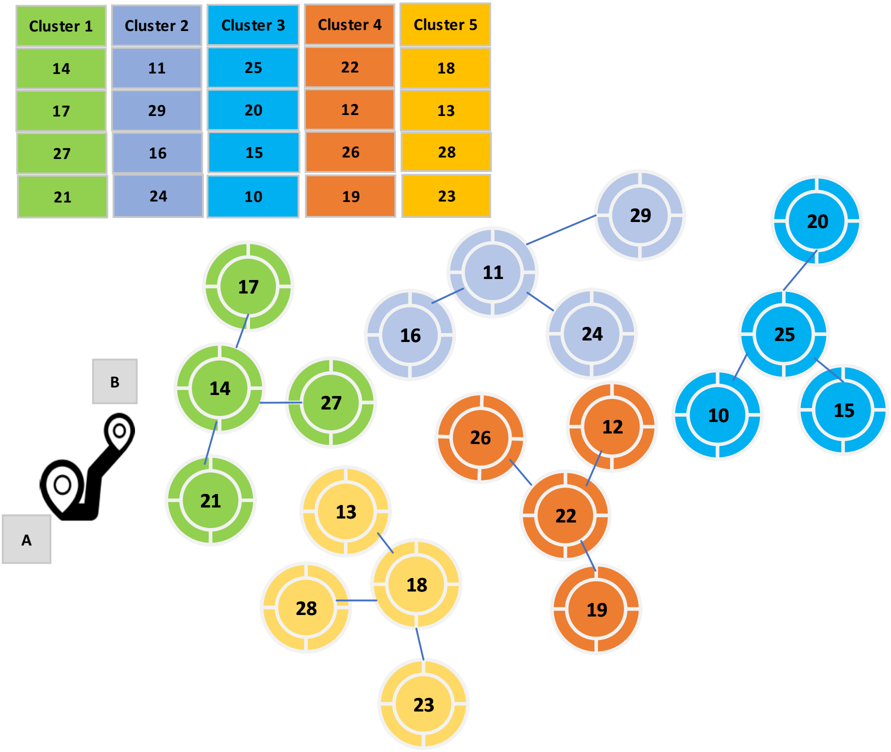
In APMA, each individual is considered as an anonymization solution for the PPTDP. A vector of size proportional to the number of clusters in the trajectory data represents each individual. Each gene, or element, in the person, or vector, signifies a cluster centroid. Subsequently, we use a grouping-based approach to provide the comprehensive clustering solution. To form a group, we choose the trajectories that are closest to each cluster centroid using the *ED*. The privacy preservation requirement is directly fulfilled when each group size equals the anonymity threshold k in anonymization.

Figure 1 describes how 20 trajectories are divided into 5 clusters. Each cluster is constructed according to similar routes or timezone. Each number represents a trajectory. To satisfy the privacy criterion $k = 4$, we randomly choose five trajectories that serve as cluster centroids. For the first centroid (trajectory 14), we establish a cluster with its three closest trajectories (17,27,21) that have not been selected as centroids. Likewise, for the second centroid (trajectory 11), we choose the remaining three trajectory data based on their distances. Ultimately, all trajectories are categorized into five clusters, with each cluster containing four trajectories. This arrangement satisfies the privacy criterion $k = 4$.

The proposed representation method offers two advantages. Initially, it incorporates only the cluster centroids within the individual, thereby decreasing the complexity of the search space for optimizing clustering solutions. Secondly, a distance-based grouping method is employed to convert the centroids within an individual into a comprehensive clustering solution. This method utilizes heuristic distance information among trajectories, leading to enhanced quality in trajectory clustering [19].

4.2 Adaptive crossover and mutation parameter

The rates of crossover and mutation, along with the selection of chromosomes, substantially affect the efficacy of GAs. Proper tuning of these parameters may significantly improve the performance of GAs. We designed the crossover operator in our APMA to maintain the common attributes of both parents while introducing variety via an integration of the distinct traits. This parameter executes the crossover operation between two parent solutions using an adaptive crossover rate (*CR*). We obtain each gene in the child (except the last one) from either the father or mother individual, based on a random value.



Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
14	11	25	22	18
17	29	20	12	13
27	16	15	26	28
21	24	10	19	23

Fig. 1 An illustration of the representation of 20 trajectory data points into 5 clusters under $k = 4$ privacy requirements [26]

If the random value exceeds the *CR*, the child inherits information from the father; if not, it inherits information from the mother. We estimate the last gene by averaging the information from both individuals. The illustration of adaptive crossover is given in Fig. 2. In this figure, the first four genes from the father and mother represent the cluster centroid, and the last one represents the effective cluster number in both individuals. The mother individual is represented as $[2, 7, 8, 5, 4]$ and father individual is $[4, 6, 3, 1, 2]$. A list of random number between 0 and 1 is generated as $[0.8, 0.6, 0.3, 0.5]$. For the first index, since $0.8 > 0.7$, the information is selected from the father. For the 2nd index, since $0.6 < 0.7$, the information is taken from the mother, and so on. The information for the last index is estimated from the average of both parents' effective cluster number information, which is 3 (See Fig. 2). Consequently, the resulting child individual is [4,2,8,5,3]. This probability may be adjusted dynamically to balance exploration (discovery of novel gene combinations) with exploitation (emphasis on successful combinations) throughout the evolutionary process.

In the genetic search process, the mutation operator is essential because it helps avoid local optima by providing new information at random, increasing diversity, and making it easier to explore better solutions. The procedure introduces small random changes to genes within a chromosome directed by a small

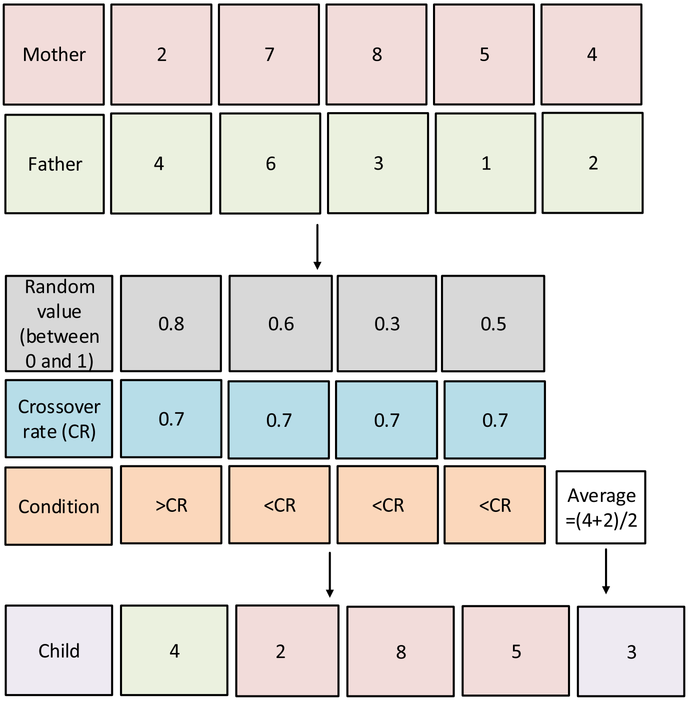


Fig. 2 An illustration of adaptive crossover strategy

mutation probability. The main focus of this strategy is to preserve genetic diversity while selecting best-fitted individuals. The adaptive mutation strategy in our proposed algorithm introduces a controlled diversity in the population by modifying the individual solutions while ensuring the feasibility probabilistically.

Mutation condition Mutation is applicable just when the problem space permits variability, specifically when the boundary size exceeds one. To avoid unnecessary calculations, we omit mutation if the boundary size is less than 1. In our mutation strategy, every gene experiences a mutation with probability F, where F is an adaptive mutation factor that is dynamically modified over the course of evolution. The value of F determines the probability of mutation for each gene index that affects the search space's balance of exploration and exploitation.

Our mutation strategy ensures that the newly generated value is not duplicated (except the last element); if a duplicate value is identified, it generates a new random value until a unique value is found. This process ensures solution feasibility and prevents duplicate assignments within the individual. After iterating through all genes, the modified solution returns as offspring. This ensures a balance between diversity and structural integrity.

This mutation technique is capable of global search, perhaps revealing answers that a conservative mutation strategy may miss. It may also increase the genetic diversity in a population, which is important for making the search area bigger and lowering the risk of local optima.

4.3 Adaptive memory mechanism

Adaptive memory is a self-adaptive parameter tuning method designed to dynamically control the mutation and crossover rates in EAs. By maintaining a historical recollection of effective parameter values, it repeatedly changes them to improve optimally [42]. The memory starts with initial default values such as $CR = 0.5$ and $F = 0.5$ in each iteration, new values are drawn from the Gaussian distribution centered on the previously recorded parameters, guaranteeing controlled exploration. The memory is then updated using a simple mean while the mutation factor is updated using a weighted sum of squares, preferring larger values. By substituting more efficient new values with the older ones, a cycling memory indexing method guarantees continuous learning. By allowing the algorithm to dynamically balance exploration and exploitation, this adaptive technique helps improve convergence speed and robustness, thereby reducing the requirement for manual parameter adjustment.

4.4 Local search parameter

The local search strategy iteratively enhances the candidate solution by exploring the neighborhood that helps to obtain a similar set of solutions by making small changes in the solution. This approach improves the offspring by replacing less common components with more common ones in the population. Within the evolutionary framework, this method improves the general quality of the candidate solutions.

The local search builds an element frequency dictionary by assessing the frequency of every element throughout the whole population omitting the final element of each individual, therefore representing the effective cluster number. The elements in the population are sorted in decreasing order according to their frequency, therefore determining the most often occurring element and enabling the refinement process. The components in the child solution are simultaneously organized in ascending order, with the least frequent element under special attention. The dictionary records the frequency of each element's appearance within the population.

Using this information, we calculate the Element Existing Ratio (EER) as follows:

$$EER = \frac{f_i}{N} \tag{1}$$

where f_i is the frequency of the ith element and N is the total population size. The local search strategy involves the following process:

(i) To carry out the refinement process, the elements in the population are arranged in descending order based on their frequency, identifying the most common element. Simultaneously, the elements in the child solution are arranged in ascending order, where the element with the least frequency is highlighted. The lowest rank in the child population is selected as a candidate for replacement.

(ii) The algorithm now seeks the high-frequency element from the population that is absent in the child's solution. This guarantees that newly added components are both highly representative of the population and diversified in the context of children. If no viable substitute is discovered, the child solution remains unaltered.

(iii)

Once a suitable high-frequency element are identified, it replaces the least frequent element in the child solution. Throughout the process, the uniqueness constraint guarantees that the new element is not previously present in the child.

(iv) The last item in the child individual represents an efficient cluster count, which is retained throughout the local search and then reattached to the solution post-modification. This process guarantees the preservation of the structural features of the clustering.

The rationale behind the proposed four steps is to identify low-frequency elements in the child individual and replace them with high-frequency elements from the population. Since the low-frequency elements in the child might represent weaker choices, we focus on identifying these elements and substituting them with the high-frequency ones to obtain more effective solutions. Throughout the process, we attempt to improve the solution quality while maintaining the uniqueness of the elements. Overall, the aim of the local search strategy is to improve the solution quality while preserving the structural integrity. The dynamic replacement of less frequent elements with more frequent and stronger ones helps us achieve the better solutions.

Algorithm 1 Pseudo-code of APMA

```
Initialize generation index $G_i \leftarrow 0$
Create initial population $P$
Evaluate population $P$
**while** stopping criterion is not satisfied **do**
    **for** each pair of parent individuals $(p_1, p_2)$ in $P$ **do**
        $(F, CR) \leftarrow$ AdaptiveMemory.SampleParameters()
        $child \leftarrow$ AdaptiveCrossover$(p_1, p_2, CR)$
        $child \leftarrow$ AdaptiveMutation$(child, F,$ boundary_size$)$
        $child \leftarrow$ LocalSearch$(child, P)$
        Evaluate $child$ using privacy and utility objectives
        Update adaptive memory if $child$ improves the corresponding parent(s)
        Add $child$ to the offspring population $P_{\text{offspring}}$
    **end for**
    $P \leftarrow$ NonDominatedSortingSelection$(P \cup P_{\text{offspring}},$ population_size$)$
    Update hypervolume indicator and record convergence information
    $G_i \leftarrow G_i + 1$
**end while**
**Output** non-dominated solutions in $P$
```

4.5 Overall process of APMA

The general process of APMA is presented in Algorithm 1. This algorithm begins with initializing a generation index and creating a population of potential solutions. Then, the population is assessed based on the multi-objective privacy and utility criteria, which measure the effectiveness of the solutions in the optimization problems. The next step is to start a loop that will persist until a specified termination (i.e. maximum generation/evaluation number) condition is met. After that, the algorithm chooses the adaptive parameters, such as F and CR, from the adaptive memory for every pair of parent individuals. Then the adaptive crossover and mutation operation is performed to produce the child individual. The child individual then undergoes a local search operation for refinement. After that, the child individual is evaluated based on the privacy and utility criteria, and if it improves the corresponding parents, it is added to the offspring population. The adaptive memory is also updated. Following processing all parent pairs, the offspring are merged with the present population, and the next generation is produced using a selection mechanism grounded on non-dominated sorting and hypervolume. Finally, after the stopping criterion is satisfied, the algorithm generates the non-dominated solutions from the last population.

5 Experimental results

In this section, we present the experimental results of APMA and compare them to some existing approaches.

5.1 Performance evaluation metric

We evaluate the performance of APMA and its variants using the hypervolume (HV) metric [55], which is widely used in MOO problems for performance comparison. HV is a Pareto-compliant metric that quantifies the volume of the objective space dominated by a collection of non-dominated solutions, constraint by specific reference points. In the context of privacy-utility optimization, HV effectively captures the trade-off by simultaneously achieving high privacy preservation and high data utility. This metric assess the quality of solutions relative to the Pareto front-the set of non-dominated solutions for which no others are equivalent or better across all objectives [16]. The HV value corresponds to the spatial volume enclosed between the reference point and the obtained solutions, where the reference points is typically defined as the anti-optimal point in the objective space [17]. Higher HV values indicate better convergence and diversity of the solution set, thus reflecting superior algorithmic performance [8].

5.2 Comparison with existing methods

We assess the effectiveness of the novel APMA by comparing it with the established NSGA-II and DPGA algorithms. A comprehensive explanation of these two algorithms is provided below.

1. **NSGA-II** [10]: NSGA-II is a fundamental GA that has been effectively employed in numerous complex optimization tasks. It implements a non-dominated sorting strategy for the MOO problem. We have adapted this algorithm to achieve our MOO goal of maximizing privacy and utility for trajectory data and observed its performance. This algorithm is well known for producing solutions that are not dominated by others.
2. **DPGA** [26]: DPGA is a current state-of-the-art EA that implements dynamic crossover and mutation rates and a scramble mutation for multi-objective trajectory data publishing. This algorithm performs better than NSGA-II in generating privacy-utility solutions.

We estimate the average (Avg) and standard deviation (Std) of the HV of these algorithms and see their performance. The algorithms with larger HV indicate that the corresponding algorithm is more efficient in producing better privacy-utility solutions. Firstly, we compare APMA with NSGA-II and see from Table 3 that APMA has better HV in all test instances than NSGA-II. When we compare it with DPGA, we observe that APMA has better HV in 12 test instances than the DPGA algorithm. Better HV values are marked in **bold**.

To emphasize the significance of our solution, we perform the Wilcoxon signed-rank test for NSGA-II versus APMA at a 5% level of significance. Out of 15 test instances, 12 are found statistically significant; also, for DPGA versus APMA, 7 test instances show statistically significant outcomes. It clearly establishes the superiority of our proposed algorithm over existing state-of-the-art algorithms. All the significant results are marked using the † symbol.

We also illustrate the convergence value of our algorithm and compare it with NSGA-II and DPGA. In Fig. 3, time is plotted on the horizontal axis, while HV is plotted on the vertical axis. From Fig. 3, it is evident that APMA converges more effectively than both NSGA-II and DPGA. This indicates the advantages of the adaptive crossover and mutation parameters, as well as the local search strategy, in identifying better solutions for privacy and utility.

5.3 Impact of adaptive parameter and local search

We see the impact of the local search operator on our APMA algorithm and compare it with a variant of the APMA algorithm and a variant of the NSGA-II algorithm.

1. **APMA without local search:** When we remove the local search operator from APMA, we call it Adaptive Parameter Genetic Algorithm (APGA).
2. **NSGA-II with local search:** After adding the local search strategy to the NSGA-II algorithm, the variant becomes a non-dominated sorting Memetic Algorithm (NSMA). In this algorithm, uniform crossover and random mutation settings are utilized.

Table 4 shows that when APMA is compared to NSMA, all test cases have better HV than NSMA. Moreover, 13 of the 15 outcomes are statistically significant, which means that the adaptive parameter and local search strategy have a significantly better effect on APMA's results. When APMA is compared to APGA, it is seen that APMA has better HV in 12 test cases, with six (6) being statistically significant. This shows that the adaptive parameter makes APMA better than APGA. Better solutions are marked **bold** and significant outcomes are marked with † symbol.

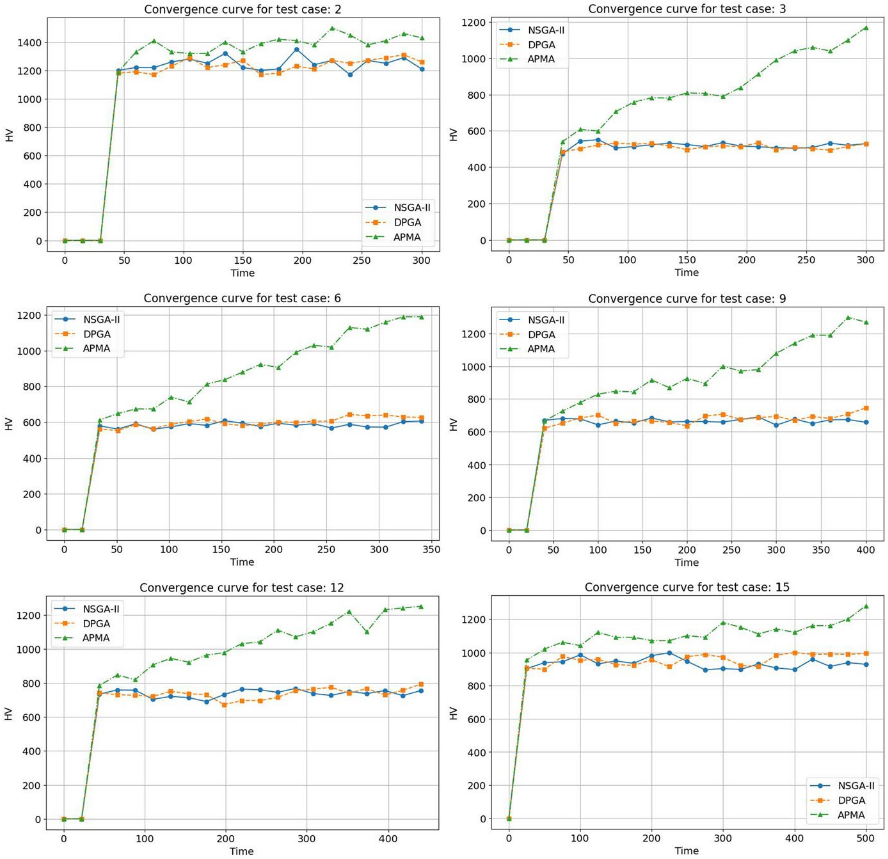


Fig. 3 Comparison of convergence outcome between NSGA-II, DPGA, and APMA

Table 3 Comparison of HV for NSGA-II versus APMA, DPGA versus APMA

Test cases	NSGA-II vs APMA				DPGA vs APMA			
	NSGA-II		APMA		DPGA		APMA	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
1	1.56E+03	2.35E+02	**1.73E+03**†	3.11E+02	1.71E+03	3.23E+02	**1.73E+03**	3.11E+02
2	1.21E+03	1.94E+02	**1.43E+03**†	3.24E+02	1.26E+03	2.51E+02	**1.43E+03**†	3.24E+02
3	5.29E+02	8.09E+01	**1.17E+03**†	4.08E+02	5.28E+02	9.06E+01	**1.17E+03**†	4.08E+02
4	1.69E+03	3.89E+02	**1.80E+03**	2.64E+02	**1.93E+03**	3.41E+02	1.80E+03	2.64E+02
5	1.26E+03	2.22E+02	**1.72E+03**†	2.62E+02	1.47E+03	2.30E+02	**1.72E+03**†	2.62E+02
6	5.90E+02	1.04E+02	**1.22E+03**†	3.57E+02	6.32E+02	9.97E+01	**1.22E+03**†	3.57E+02
7	1.82E+03	2.88E+02	**2.18E+03**†	2.98E+02	2.03E+03	4.36E+02	**2.18E+03**	2.98E+02
8	1.54E+03	3.02E+02	**1.58E+03**	2.50E+02	**1.62E+03**	2.81E+02	1.58E+03	2.50E+02
9	6.57E+02	1.01E+02	**1.27E+03**†	4.22E+02	7.45E+02	8.74E+01	**1.27E+03**†	4.22E+02
10	2.08E+03	4.85E+02	**2.19E+03**	2.46E+02	**2.29E+03**	4.20E+02	2.19E+03	2.46E+02
11	1.55E+03	3.58E+02	**1.75E+03**†	2.63E+02	1.72E+03	3.17E+02	**1.75E+03**	2.63E+02
12	7.55E+02	1.16E+02	**1.25E+03**†	3.09E+02	7.93E+02	1.37E+02	**1.25E+03**†	3.09E+02
13	2.30E+03	5.14E+02	**2.82E+03**†	4.05E+02	2.77E+03	6.23E+02	**2.82E+03**	4.05E+02
14	1.89E+03	4.02E+02	**2.23E+03**†	2.84E+02	2.16E+03	4.24E+02	**2.23E+03**	2.84E+02
15	9.29E+02	1.73E+02	**1.28E+03**†	2.59E+02	9.95E+02	1.69E+02	**1.28E+03**†	2.59E+02

This also verifies the performance of adaptive parameter and novel local search operator in our APMA.

6 Experimental tools and environment

In this section, we describe information about test cases, parameter settings, and details of the algorithm environment used in the experiment.

6.1 Test cases

The algorithm is evaluated using 15 test instances derived from the T-Drive trajectory dataset, which comprises the one-week trajectories of 10,357 taxis in Beijing [51, 52]. This dataset contains some unique trajectories which pose a risk of location disclosure if an attacker identifies any part of the trajectory. Our focus is on preventing the re-identification attack within this dataset. The test cases vary according to the number of trajectories (T_n), the size of the trajectories (T_s), the number of data centers (ndc), and the privacy thresholds k and δ. Three values of k-20, 25, and 50-are selected to represent different levels of privacy protection. Table 5 presents the characteristics of the test cases used in this study.

Table 4 Impact of adaptive parameter and local search

Test cases	NSMA vs APMA				APGA vs APMA			
	NSMA		APMA		APGA		APMA	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
1	1.56E+03	2.48E+02	**1.73E+03**†	3.11E+02	**1.84E+03**	3.99E+02	1.73E+03	3.11E+02
2	1.20E+03	2.01E+02	**1.43E+03**†	3.24E+02	1.35E+03	3.46E+02	**1.43E+03**	3.24E+02
3	5.30E+02	8.32E+01	**1.17E+03**†	T4.08E+02	7.79E+02	1.84E+02	**1.17E+03**†	4.08E+02
4	1.75E+03	2.90E+02	**1.80E+03**†	2.64E+02	1.71E+03	3.81E+02	**1.80E+03**	2.64E+02
5	1.30E+03	2.54E+02	**1.72E+03**†	2.62E+02	1.58E+03	3.65E+02	**1.72E+03**	2.62E+02
6	5.82E+02	7.78E+01	**1.22E+03**†	3.57E+02	7.29E+02	1.72E+02	**1.22E+03**†	3.57E+02
7	1.78E+03	3.29E+02	**2.18E+03**†	2.98E+02	2.14E+03	4.27E+02	**2.18E+03**†	2.98E+02
8	1.54E+03	3.04E+02	**1.58E+03**	2.50E+02	**1.64E+03**	4.60E+02	1.58E+03	2.50E+02
9	6.74E+02	9.54E+01	**1.27E+03**†	4.22E+02	8.60E+02	1.82E+02	**1.27E+03**†	4.22E+02
10	1.96E+03	3.73E+02	**2.19E+03**†	2.46E+02	2.10E+03	3.44E+02	**2.19E+03**	2.46E+02
11	1.63E+03	3.43E+02	**1.75E+03**	2.63E+02	**1.81E+03**	3.75E+02	1.75E+03	2.63E+02
12	6.84E+02	1.25E+02	**1.25E+03**†	3.09E+02	8.84E+02	1.42E+02	**1.25E+03**†	3.09E+02
13	2.36E+03	4.71E+02	**2.82E+03**†	4.05E+02	2.66E+03	4.58E+02	**2.82E+03**	4.05E+02
14	1.92E+03	4.00E+02	**2.23E+03**†	2.84E+02	2.20E+03	3.43E+02	**2.23E+03**	2.84E+02
15	8.83E+02	1.55E+02	**1.28E+03**†	2.59E+02	1.01E+03	1.96E+02	**1.28E+03**†	2.59E+02

6.2 Parameter settings

The population size in our approach is fixed at 30. The mutation rate (M_r) for NSGA-II, NSMA, and DPGA is set to 0.05, while for APMA and APGA, the initial values of CR and F are both 0.5. The maximum evaluation time is determined by dividing the number of trajectories by 10. All algorithms are run 25 times on 15 test cases. These parameter values were chosen based on the algorithms' better performance under these settings.

6.3 Impact of the key parameters

For every three test cases, T_n, T_s, ndc, and δ are fixed, and only the privacy parameter k is varied. The values of CR and F are dynamically adjusted to achieve a better privacy-utility solution. We observed that the solutions of APMA, as well as those of the other algorithms, vary according to the k values, where the CR and F are adaptively adjusted to achieve the privacy and utility solution. For k=20 the average HV for the first test case is 1.73E+3. As the value of k increases to 25, the solution quality decreases to 1.43E+3, and for $k = 50$, it further declines to 1.17E+03. This illustrates how solution quality varies with different k values. The trade-off between privacy and utility is adaptively adjusted based on varying k values. As k increases, the utility of the solutions decreases correspondingly, and vice versa. This demonstrates how the

Table 5 Characteristics of test cases [19]

Test cases	T_n	T_s	*ndc*	*k*	δ
1	3000	10	5	20	1.0
2	3000	10	5	25	1.0
3	3000	10	5	50	1.0
4	3500	10	5	20	1.0
5	3500	10	5	25	1.0
6	3500	10	5	50	1.0
7	4000	10	5	20	1.0
8	4000	10	5	25	1.0
9	4000	10	5	50	1.0
10	4500	10	5	20	1.0
11	4500	10	5	25	1.0
12	4500	10	5	50	1.0
13	5000	10	5	20	1.0
14	5000	10	5	25	1.0
15	5000	10	5	50	1.0

privacy-utility trade-off is achieved. Furthermore, through the comparisons with other algorithms, the proposed algorithm achieved best privacy -utility trade-off.

6.4 Computational complexity and runtime comparison

The overall total computation complexity of the proposed algorithm is $O(MN^2 + maxEvalT_nT_s)$ where M is the number of objectives, N is the population size, *maxEval* represents the maximum fitness evaluation number, T_n indicates the number of trajectories, and T_s is the size of trajectories. The running time of NSGA-II, DPGA, and APMA are shown in Table 6.

6.5 Algorithm implementation

All the algorithms, including NSGA-II, DPGA, NSMA, APGA, and APMA, are executed on a Windows 10 system with an Intel(R) Core(TM) i5-8500 CPU @ 3.00 GHz and 8.00 GB RAM, using a Python 3.11 (64-bit) environment.

7 Conclusion

In this paper, we present our Adaptive Parameter Memetic Algorithm (APMA) for achieving better privacy and utility solutions in trajectory data publishing. It uses a new local search strategy and an adaptive memory-based crossover and

Table 6 Running time results (in milliseconds)

Test cases	NSGA-II	DPGA	APMA
1	66017.22	64830.56	65563.07
2	61239.10	60279.31	60012.42
3	48416.74	47698.65	47261.97
4	93954.99	98664.41	99171.27
5	84624.47	89429.83	97217.41
6	67646.49	67457.27	84113.56
7	134271.75	129449.81	145069.39
8	120247.14	116011.00	125684.46
9	90472.19	94476.54	92494.08
10	180446.45	183114.41	179186.49
11	160264.93	160630.22	161422.82
12	119475.28	125224.91	122476.30
13	248159.99	247796.68	239082.83
14	218769.03	214868.56	211378.03
15	160089.68	160748.37	160243.28

mutation operator. Our method guarantees an improved privacy-utility tradeoff solution over the current approaches by using Multi-Objective Optimization (MOO) strategies. Extensive experiments show that our proposed algorithm can enhance the convergence result and the solution quality. The main benefit of the adaptive parameter tuning mechanism is its better exploration and exploitation capacity than the existing methods. The proposed local search operator has the advantages of ensuring better search capacity and improving population diversity. In the future, Differential Privacy (DP) can be integrated into our framework to develop a hybrid privacy-preserving approach that enhances both clustering methodologies and privacy protection. Furthermore, this method is applicable to offline location scenarios; we anticipate extending our algorithm for online settings in future investigations.

Author Contributions Samsad Jahan contributes to conceptualization, methodology, software, writing - original draft. Yong-Feng Ge contributes to conceptualization, methodology, software, writing - original draft. Hua Wang contributes to supervision, project administration, writing - review & editing. Enamul Kabir contributes to validation, formal analysis, writing - review & editing.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. This paper has no funding information.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

1. Abul O, Bonchi F, Nanni M (2008) Never walk alone: Uncertainty for anonymity in moving objects databases. In: 2008 IEEE 24th international conference on data engineering, Ieee, pp 376–385
2. Abul O, Bonchi F, Nanni M (2010) Anonymization of moving objects databases by clustering and perturbation. Inf Syst 35(8):884–910
3. Assayag Y, Souto E, Barreto R et al (2025) Efficient exploration of indoor localization using genetic algorithm and signal propagation model. Computing 107(1):30
4. Barhamgi M, Perera C, Ghedira C et al (2018) User-centric privacy engineering for the internet of things. IEEE Cloud Comput 5(5):47–57
5. Barhamgi M, Huhns MN, Perera C et al (2021) Introduction to the special section on human-centered security, privacy, and trust in the internet of things
6. Barhamgi M, Masmoudi A, Lara-Cabrera R et al (2024) Social networks data analysis with semantics: application to the radicalization problem. J Ambient Intell Humaniz Comput pp 1–15
7. Bindschaedler V, Shokri R (2016) Synthesizing plausible privacy-preserving location traces. In: 2016 IEEE symposium on security and privacy (SP), IEEE, pp 546–563
8. Chen H, Wu G, Pedrycz W et al (2019) An adaptive resource allocation strategy for objective space partition-based multiobjective optimization. IEEE Trans Syst Man Cybern A Syst 51(3):1507–1522
9. Chiba T, Sei Y, Tahara Y et al (2019) Trajectory anonymization: Balancing usefulness about position information and timestamp. 2019 10th IFIP international conference on new technologies. IEEE, Mobility and Security (NTMS), pp 1–6
10. Deb K, Pratap A, Agarwal S et al (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans Evol Comput 6(2):182–197
11. Dwork C (2006) Differential privacy. In: International colloquium on automata, languages, and programming, Springer, pp 1–12
12. Ge YF, Cao J, Wang H et al (2020) Distributed differential evolution for anonymity-driven vertical fragmentation in outsourced data storage. In: Web information systems engineering – WISE 2020. Springer International Publishing, p 213–226, https://doi.org/10.1007/978-3-030-62008-0_15
13. Ge YF, Orlowska M, Cao J et al (2021) Knowledge transfer-based distributed differential evolution for dynamic database fragmentation. Knowl-Based Syst 229:107325. https://doi.org/10.1016/j.knosys.2021.107325
14. Ge YF, Orlowska M, Cao J et al (2022) MDDE: Multitasking distributed differential evolution for privacy-preserving database fragmentation. VLDB J. https://doi.org/10.1007/s00778-021-00718-w
15. Ge YF, Wang H, Cao J et al (2022b) An information-driven genetic algorithm for privacy-preserving data publishing. In: International conference on web information systems engineering, Springer, pp 340–354
16. Ge YF, Bertino E, Wang H et al (2023) Distributed cooperative coevolution of data publishing privacy and transparency. ACM Trans Knowl Discov Data 18(1):1–23
17. Ge YF, Wang H, Bertino E et al (2024) Evolutionary dynamic database partitioning optimization for privacy and utility. IEEE Trans Dependable Secure Comput 21(4):2296–2311
18. Ge YF, Wang H, Cao J et al (2024) Privacy-preserving data publishing: An information-driven distributed genetic algorithm. World Wide Web 27(1):1
19. Ge YF, Wang H, Cao J et al (2024c) Federated genetic algorithm: Two-layer privacy-preserving trajectory data publishing. In: Proceedings of the genetic and evolutionary computation conference, pp 749–758

20. Gramaglia M, Fiore M (2015) Hiding mobile traffic fingerprints with glove. In: Proceedings of the 11th ACM conference on emerging networking experiments and technologies, pp 1–13
21. Gursoy ME, Liu L, Truex S et al (2018) Utility-aware synthesis of differentially private and attack-resilient location traces. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp 196–211
22. Haider SA, Zubairi JA, Idwan S (2024) Mapping and just-in-time traffic congestion mitigation for emergency vehicles in smart cities. Computing 106(12):4109–4130
23. He X, Cormode G, Machanavajjhala A et al (2015) Dpt: Differentially private trajectory synthesis using hierarchical reference systems. Proc VLDB Endow 8(11):1154–1165
24. Hu Z, Yang J, Zhang J (2018) Trajectory privacy protection method based on the time interval divided. Comput Security 77:488–499
25. Jahan S, Ge YF, Kabir E et al (2023) Analysis and protection of public medical dataset: From privacy perspective. In: International conference on health information science, Springer, pp 79–90
26. Jahan S, Ge YF, Wang H et al (2024) Dynamic-parameter genetic algorithm for multi-objective privacy-preserving trajectory data publishing. In: International conference on web information systems engineering, Springer, pp 46–57
27. Jin F, Hua W, Francia M et al (2022) A survey and experimental study on privacy-preserving trajectory data publishing. IEEE Trans Knowl Data Eng 35(6):5577–5596
28. Jin F, Hua W, Ruan B et al (2022b) Frequency-based randomization for guaranteeing differential privacy in spatial trajectories. In: 2022 IEEE 38th International conference on data engineering (ICDE), IEEE, pp 1727–1739
29. Kabir ME, Mahmood AN, Wang H et al (2020) Microaggregation sorting framework for k-anonymity statistical disclosure control in cloud computing. IEEE Trans Cloud Comput 8(2):408–417. https://doi.org/10.1109/tcc.2015.2469649
30. Kopanaki D, Theodossopoulos V, Pelekis N et al (2016) Who cares about others' privacy: Personalized anonymization of moving object trajectories. In: EDBT, pp 425–436
31. Li F, Gao F, Yao L et al (2016) Privacy preserving in the publication of large-scale trajectory databases. In: Big data computing and communications: Second international conference, BigCom 2016, Shenyang, China, July 29-31, 2016. Proceedings 2, Springer, pp 367–376
32. Li N, Li T, Venkatasubramanian S (2006) t-closeness: Privacy beyond k-anonymity and l-diversity. In: 2007 IEEE 23rd international conference on data engineering, IEEE, pp 106–115
33. Machanavajjhala A, Kifer D, Gehrke J et al (2007) l-diversity: Privacy beyond k-anonymity. ACM Trans Knowl Discov Data (TKDD) 1(1):3–es
34. Meyerson A, Williams R (2004) On the complexity of optimal k-anonymity. In: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp 223–228
35. Monte Sousa F, Callou G (2025) Analysis of evolutionary multi-objective algorithms for data center electrical systems. Computing 107(2):65
36. Sheng W, Chen S, Sheng M et al (2016) Adaptive multisubpopulation competition and multiniche crowding-based memetic algorithm for automatic data clustering. IEEE Trans Evol Comput 20(6):838–858
37. Sweeney L (2002) k-anonymity: A model for protecting privacy. Int J Uncertain Fuzziness Knowledge-Based Syst 10(05):557–570
38. Tachioka Y (2022) Privacy preservation satisfying utility requirements based on multi-objective optimization. In: 2022 joint 12th international conference on soft computing and intelligent systems and 23rd international symposium on advanced intelligent systems (SCIS &ISIS), IEEE, pp 1–4
39. Tian C, Xu H, Lu T et al (2021) Semantic and trade-off aware location privacy protection in road networks via improved multi-objective particle swarm optimization. IEEE Access 9:54264–54275
40. Trajcevski G, Wolfson O, Hinrichs K et al (2004) Managing uncertainty in moving objects databases. ACM Trans Database Syst (TODS) 29(3):463–507
41. Tu Z, Zhao K, Xu F et al (2019) Protecting trajectory from semantic attack considering *k*-anonymity, *l*-diversity, and *t*-closeness. IEEE Trans Netw Serv Manage 16(1):264–278. https://doi.org/10.1109/tnsm.2018.2877790
42. Wang GG, Gao D, Pedrycz W (2022) Solving multiobjective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm. IEEE Trans Industr Inf 18(12):8519–8528
43. Wang H, Zhang Y, Cao J (2006) Ubiquitous computing environments and its usage access control. In: Proceedings of the 1st international conference on Scalable information systems, pp 6–es

44. Wang H, Jiang X, Kambourakis G (2015) Special issue on security, privacy and trust in network-based big data. Inf Sci 318:48–50
45. Wang J, Ren W, Zhang Z et al (2018) A hybrid multiobjective memetic algorithm for multiobjective periodic vehicle routing problem with time windows. IEEE Trans Syst Man Cybern A Syst 50(11):4732–4745
46. Yao L, Chen Z, Hu H et al (2022) Privacy preservation for trajectory publication based on differential privacy. ACM Trans Intell Syst Technol (TIST) 13(3):1–21
47. Yin J, Tang M, Cao J et al (2022) Knowledge-driven cybersecurity intelligence: Software vulnerability coexploitation behavior discovery. IEEE Trans Industr Inf 19(4):5593–5601
48. Yin J, Chen G, Hong W et al (2024) A heterogeneous graph-based semi-supervised learning framework for access control decision-making. World Wide Web 27(4):35
49. You M, Yin J, Wang H et al (2023) A knowledge graph empowered online learning framework for access control decision-making. World Wide Web 26(2):827–848
50. You M, Ge YF, Wang K et al (2024) Hierarchical adaptive evolution framework for privacy-preserving data publishing. World Wide Web 27(4):49
51. Yuan J, Zheng Y, Zhang C et al (2010) T-drive: driving directions based on taxi trajectories. In: Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems, pp 99–108
52. Yuan J, Zheng Y, Xie X et al (2011) Driving with knowledge from the physical world. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 316–324
53. Zhang J, Huang Q, Huang Y et al (2023) Dp-trajgan: A privacy-aware trajectory generation model with differential privacy. Futur Gener Comput Syst 142:25–40
54. Zhang Q, Li H (2007) Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731
55. Zhang X, Tian Y, Jin Y (2014) A knee point-driven evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 19(6):761–776
56. Zhao B, Chen WN, Wei FF, et al (2024) Pega: A privacy-preserving genetic algorithm for combinatorial optimization. IEEE Trans Cybern
57. Zheng Z, Li Z, Jiang H et al (2022) Semantic-aware privacy-preserving online location trajectory data sharing. IEEE Trans Inf Forensics Secur 17:2256–2271

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Samsad Jahan[1] · Yong-Feng Ge[1] · Hua Wang[1] · Enamul Kabir[2]

✉ Samsad Jahan
samsad.jahan@live.vu.edu.au

✉ Yong-Feng Ge
yongfeng.ge@vu.edu.au

Hua Wang
hua.wang@vu.edu.au

Enamul Kabir
enamul.kabir@usq.edu.au

[1] Institute for Sustainable Industries and Liveable Cities, Victoria University, 70-104 Ballarat Rd, Footscray, VIC 3011, Australia

[2] School of Mathematics, Physics and Computing, University of Southern Queensland, 487-535 West St, Toowoomba, QLD 4350, Australia