



A survey on soft computing-based high-utility itemsets mining

Rajiv Kumar¹ · Kuldeep Singh²

Accepted: 25 November 2021

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

Traditional frequent itemsets mining (FIM) suffers from the vast memory cost, small processing speed and insufficient disk space requirements. FIM assumes only binary frequency value for items in the dataset and assumes equal importance value for items. In order to target all these limitations of FIM, high-utility itemsets (HUIs) mining has been presented. HUIs mining is more complicated and difficult than FIM. HUIs mining algorithms spend more execution time because of large search space. Therefore, soft computing techniques-based HUIs mining has been proposed. Soft computing techniques provide a systematic process to discover the optimum solutions by using the concept of natural evolution. This article explores the usage of soft computing techniques in HUIs mining. The article presents a taxonomy of soft computing techniques-based HUIs mining including evolutionary computation and fuzzy logic-based approaches. This article enhances understanding of HUIs mining problems, the current status of provided solutions. The paper provides a comparison analysis of key techniques and discusses theoretical aspects of the various nature-inspired and fuzzy logic-based techniques along with their pros and cons. In addition, an overview of strategies and approaches is also presented. Finally, the article mentions key research opportunities and future directions.

Keywords Pattern mining · High-utility itemsets mining · Utility mining · Evolutionary techniques · Fuzzy logic · Soft computing · Nature-inspired techniques

1 Introduction

In the past two decades, knowledge discovery database (KDD) is an evolving issue to mine useful knowledge from very large datasets. The mining of knowledge is generally grouped into association rules (Agrawal et al. 1993; Agrawal and Srikant 1994; Chen et al. 1996; Zaki 2000), sequential patterns (Agrawal and Srikant 1995), classification [6], clustering (Berkhin 2006) and high-utility itemsets (Chan et al. 2003, [9]). The frequent itemsets mining (FIM) or association rule mining (ARM) is proposed to mine the frequent itemsets where the occurrence of frequencies is no less than

the minimum support threshold or their confidence is no less than user defined minimum confidence threshold (Agrawal and Srikant 1994; Han et al. 2000). In order to mine the association rules (ARs), Agrawal and Srikant (1994) presented apriori algorithm in a level-wise manner. But, it consumes a high amount of time to identify the frequent itemsets and generation of the numerous candidates. In order to target the limitations of required a high amount of execution time and a large number of candidate itemsets, evolutionary computation (EC) techniques have been used to mine ARs (Telikani et al. 2020).

In real-life situation, there are many itemsets with high profit value, but less occurrence. In that case, FIM or ARM is extremely inadequate to discover the high-profit itemsets. For example, breads are likely to purchase more as compared to diamonds. The bread itemsets have higher frequency with low-profit value, while diamond itemsets have low frequency with high-profit value. Therefore, the information of high-utility itemsets is highly beneficial as compared to frequent itemsets. In order to overcome the limitations of FIM, high-utility itemsets mining (HUIM) algorithms (Ahmed et al. 2009; Yao and Hamilton 2006; Singh et al. 2018, [9]) are

✉ Kuldeep Singh
ksingh@cs.du.ac.in

Rajiv Kumar
rajiv.kumar@bothouniversity.ac.bw

¹ Faculty of Engineering and Technology, Botho University, Gaborone, Botswana

² Department of Computer Science, University of Delhi, Delhi 110-007, India

proposed to find the high-profit itemsets from the quantitative datasets. An itemset can be chosen as a high-utility itemset if its utility value is higher than the minimum utility threshold. The utility of the itemset is measured according to the user preferences such as profit or cost.

Several algorithms are proposed to efficiently mine the complete set of HUIs in the literature. Chan et al. (2003) firstly proposed the concept of HUIs. Yao et al. [9] discover the complete set of HUIs by using two factors. The first is the item's quantity as the internal utility, while the second is the item's profit value as the external utility. Due to combinatorial issue that occurs in earlier algorithms, Liu et al. (2005) designed a two-phase transaction weighted utility (TWU) model and transaction weighted downward closure property (TWDCP) to mine HUIs. Guo-Cheng Lan and Tseng (2014) develop the pruning strategy to effectually discover HUIs and the mining approach based on index-projection technique. Tseng et al. (2010) proposed UP-Growth algorithm and UP-tree structure to mine HUIs. Liu and Qu (2012) develop an efficient list-based HUI-Miner method to mine HUIs without candidate generation.

The traditional HUIM algorithms spend more computation time to search an item from the large datasets. This issue can be resolved by using evolutionary computation which sets the strict termination condition to decrease the computational time for a process (Cattral et al. 2009). The genetic algorithm is an example of evolutionary computation. It provides an optimal solution to resolve the NP-hard and nonlinear issues.

A genetic algorithm, named high-utility pattern extracting using genetic algorithm with ranked mutation using minimum utility threshold (HUPE_{umu}-GRAM) (Kannimuthu and Premalatha 2014), is proposed to efficiently mine HUIs. In the same paper, authors proposed another genetic algorithm, named HUPE_{wumu}-GRAM is also proposed to find HUIs without a specific minimum utility threshold. But, both these algorithms are inadequate to obtain the promising HUIs because the computations are more if the distinct items are very large in the dataset. Zhang and Ji (2015) proposed particle swarm optimization (PSO) to handle the continuous problem to obtain the optimum solutions. But, in real-time applications, PSO is insufficient to optimize for a binary search space. In order to address this issue, a binary PSO (BPSO) algorithm is presented to discover the optimum solutions to handle the binary search space (Lin et al. 2017). However, BPSO approach requires a large number of computations to obtain high accuracy.

Another solution to the computational evolutionary is to use the fuzzy theory system to handle the transaction datasets based on the crisp sets. Fuzzy frequent itemsets (FFIs) discover the fuzzy data in a level-wise way to greatly reduce the computation time; however, some information may be lost (Hong et al. 1999). Several algorithms are also developed to

mine MFFIs based on their designed tree structures (Hong et al. 2014; Lin et al. 2015). Even though it is better than apriori-like approaches, it still requires more computation to mine MFFIs.

Several surveys (Freitas 2003; Ventura and Luna 2016; Telikani et al. 2020; Badhon et al. 2021) exist on evolutionary computation-based frequent or association itemsets mining. But no survey is available for evolutionary computation or soft computing techniques-based HUIs mining. To the best of our knowledge, this work is the first that presents a survey on soft computing-based HUIs mining approaches. Our survey is distinct from FIM-based state-of-the-art works on the following aspects: First, we present a taxonomy of evolutionary computation and fuzzy-logic-based HUIs mining approaches. Second, the present theoretical and detailed analysis of the state-of-the-art approaches. Third, we also present the advantages and disadvantages of all the state-of-the-art approaches. Fourth, the paper shows the comparison of characteristics and features of the available state-of-the-art approaches.

This paper presents a comprehensive survey of soft computing techniques based on HUIs mining to provide the current developments and key opportunities in the related fields. The main contributions of this article are:

1. This paper presents an in-depth understanding of the different approaches of soft computing techniques-based HUIs mining, including their component features utilized techniques and comparisons with related concepts.
2. This survey presents a taxonomy of soft computing techniques-based HUIs mining, including evolutionary computation and fuzzy-based approaches, as shown in Fig. 1.
3. It further categorize soft computing techniques-based HUIs mining into GA, PSO, ACO, multi-objective, hybrid, simple fuzzy, temporal fuzzy and GA-based fuzzy approaches.
4. This paper presents a comparative analysis and highlights the comparative pros and cons of the current state-of-the-art approaches.
5. This article also discusses research opportunities and future remarks.

The rest of the paper is organized as follows: Sect. 2 discusses preliminaries, definitions and properties to describe the soft computing-based HUIs mining approaches. Section 3 reviews the soft computing techniques-based HUIs mining approaches. Section 4 provides discussions of HUIs mining approaches. Section 5 discusses the research opportunities and future directions. Lastly, Sect. 6 concludes the survey.

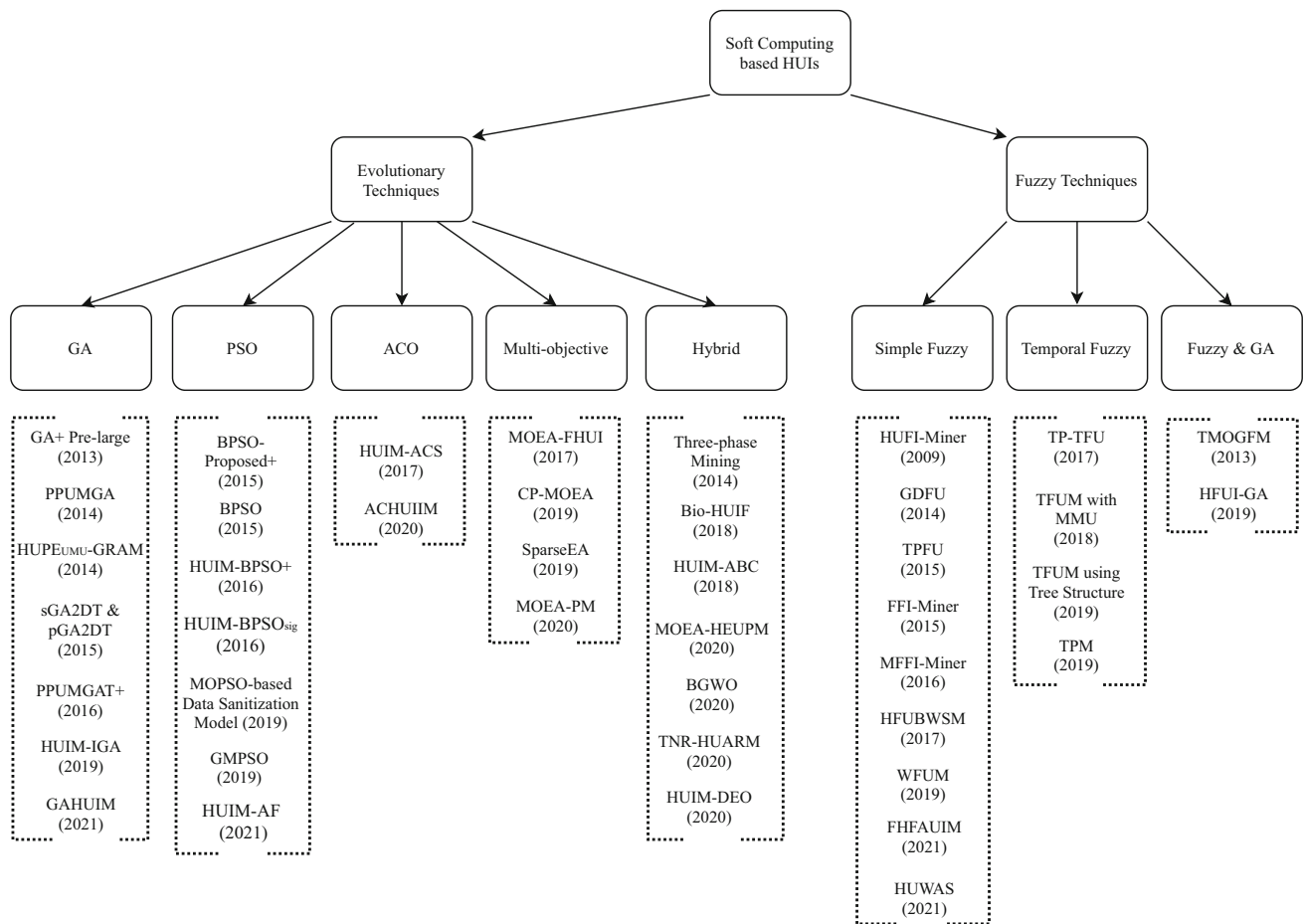


Fig. 1 Taxonomy of soft computing techniques-based high-utility itemsets mining approaches

2 Preliminaries and definitions

Most of HUIs mining approaches are expanded from the conventional FIM methods. This section introduces preliminaries, definitions and the notion of HUIs mining. This section also defines the problem statement of HUIs mining.

Let $I = \{i_1, i_2, \dots, i_n\}$ be a finite set of n distinct items (attributes) in a quantitative dataset D which consists the set of transactions, $D = \{T_1, T_2, \dots, T_m\}$, where each transaction $T_q = D(1 \leq q \leq m)$, (1) is a subset of I and (2) has a unique identifier q , called T_{ID} (transaction identifier). Each item i in a transaction T_q is associated with a purchase quantity (or internal utility), denoted as $IU(i, T_q)$. A quantitative dataset D is shown in Table 1, which will be used as a running example. The dataset D consists of eight transactions ($T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8$) and six distinct items A, B, C, D, E and F . The external utility (or profit value) of each item is denoted as $PR(i)$ as shown in Table 2. The minimum utility threshold is set as $(\delta = 20)$. For example, the transaction T_1 consists of four items A, B, D, F and has internal utility 3, 1, 3, 2, respectively. The profit value of A, B, D and F is, respectively, 2, 8, 5 and 2.

Table 1 Transactional dataset

T_{ID}	Transaction
T_1	$(A, 3), (B, 1), (D, 3), (F, 2)$
T_2	$(B, 3), (C, 2), (E, 1)$
T_3	$(B, 3), (C, 1), (E, 3), (F, 2)$
T_4	$(A, 3), (B, 2), (D, 2)$
T_5	$(A, 2)$
T_6	$(C, 2), (D, 1), (E, 3), (F, 3)$
T_7	$(A, 3), (C, 2), (D, 3)$
T_8	$(D, 3), (E, 2), (F, 1)$

Table 2 External utility

Item	A	B	C	D	E	F
Profit	2	8	1	5	7	2

Definition 1 The utility of an item i in a transaction T_q is calculated as: $U(i, T_q) = IU(i, T_q) \times PR(i)$. For example, the utility of item A in T_1 is $U(A, T_1) = IU(A, T_1) \times PR(A)$ is $3 \times 2 (= 6)$. The utility of items A, B, D and F in T_1 is,

Table 3 TU for the running example

TID	Transaction	TU
T_1	A, B, D, F	33
T_2	B, C, E	33
T_3	B, C, E, F	50
T_4	A, B, D	32
T_5	A	4
T_6	C, D, E, F	34
T_7	A, C, D	23
T_8	D, E, F	31

respectively, computed as $U(A, T_1) = IU(A, T_1) \times PR(A)$ is $3 \times 2 (= 6)$, $U(B, T_1) = IU(B, T_1) \times PR(B)$ is $1 \times 8 (= 8)$, $U(D, T_1) = IU(D, T_1) \times PR(D)$ is $3 \times 5 (= 15)$ and $U(F, T_1) = IU(F, T_1) \times PR(F)$ is $2 \times 2 (= 4)$.

Definition 2 The utility of an itemset X in transaction T_q is calculated as: $U(X) = \sum_{i \in X} U(X, T_q)$.

For example, the utility of itemsets $\{B, D\}$ and $\{B, D, F\}$ in transaction T_1 is, respectively, computed as: $U(\{B, D\}, T_1) = U(B, T_1) + U(D, T_1) = 8 + 15 (= 23)$ and $U(\{B, D, F\}, T_1) = U(B, T_1) + U(D, T_1) + U(F, T_1) = 8 + 15 + 4 (= 27)$.

Definition 3 The utility of an itemset X in a dataset D is calculated as: $U(X) = \sum_{X \subseteq T_q \wedge T_q \in D} U(X)$.

For example, the utility of $\{B\}$ and $\{B, D\}$ in dataset D is, respectively, calculated as: $U(B) = U(B, T_1) + U(B, T_2) + U(B, T_3) + U(B, T_4) = 8 + 24 + 24 + 16 (= 72)$ and $U(B, D) = U(\{B, D\}, T_1) + U(\{B, D\}, T_4) = 23 + 26 (= 49)$.

Definition 4 The transaction utility of transaction T_q is calculated as: $TU(T_q) = \sum U(i, T_q)$.

For example, $TU(T_1) = U(A, T_1) + U(B, T_1) + U(D, T_1) + U(F, T_1) = 6 + 8 + 15 + 4 (= 33)$. Table 3 shows TU for the running example.

Definition 5 The total utility of a dataset D is calculated as: $T_D = \sum_{T_q \in D} TU(T_q)$.

For example, the total utility of a dataset D is calculated as: $T_D = TU(T_1) + TU(T_2) + TU(T_3) + TU(T_4) + TU(T_5) + TU(T_6) + TU(T_7) + TU(T_8) = 33 + 33 + 50 + 32 + 4 + 34 + 23 + 31 (= 240)$.

Definition 6 The transaction weighted utilization (TWU) of an itemset X is calculated as: $TWU(X) = \sum_{X \subseteq T_q \wedge T_q \in D} TU(T_q)$.

For example, TWU of A is $TWU(A) = \sum_{A \subseteq T_q \wedge T_q \in D} TU(T_q) = T_1 + T_4 + T_5 + T_7 = 33 + 32 + 4 + 23 (= 92)$. Table 4 shows the TWU of each item for the running example.

Table 4 TWU of each item

Item	A	B	C	D	E	F
Profit	92	148	140	153	148	148

Table 5 HUIs of the running example

Itemsets	Utility	Itemsets	Utility
$\{B\}$	72	$\{D, F\}$	47
$\{D\}$	60	$\{E, F\}$	68
$\{E\}$	63	$\{A, B, D\}$	61
$\{A, D\}$	58	$\{B, C, E\}$	76
$\{B, C\}$	51	$\{B, E, F\}$	49
$\{B, D\}$	49	$\{C, E, F\}$	55
$\{B, E\}$	76	$\{D, E, F\}$	63
$\{C, E\}$	54	$\{B, C, E, F\}$	50
$\{D, E\}$	55	...	

Property 1 (Pruning using TWU). If TWU of an itemset X is less than min_util , then itemset X and none of its supersets are HUIs.

HUIM algorithms use TWU to satisfy the *anti-monotonic property*.

Definition 7 An itemset X in a dataset D is a high-utility itemset (HUI) if and only if its utility is no less than minimum utility count as: $HUI \leftarrow \{X | U(X) \geq T_D \times \delta\}$.

For example, the utility of itemsets $\{B\}$ and $\{A, B\}$ is, respectively, calculated as $U(\{B\}) = 72$ and $U(\{A, B\}) = 36$. Thus, $\{B\}$ is a HUI since $U(\{B\}) = 72 \geq 240 \times 0.2 (= 48)$, while $\{A, B\}$ is not a HUI since $U(\{A, B\}) = 36 < 240 \times 0.2 (= 48)$. Table 5 shows HUIs for the running example, where $min_util(T_D \times \delta)$ is $240 \times 0.2 (= 48.0)$.

3 Mining of high-utility itemsets using soft-computing approaches

The traditional HUIM algorithms perform on extremely high search space to find HUIs in the large datasets, thereby resulting in the exponential problem. To address this issue, soft computing techniques are proposed to discover the optimal solutions based on the principles of natural evolutionary (Cattral et al. 2009).

The soft computing techniques are broadly classified into two main categories, namely evolutionary computation (EC) and fuzzy logic. The EC soft computing techniques are further classified into genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), multi-objective optimization EC and hybrid approaches. On the

other hand, fuzzy logic soft computing techniques are further classified into simple fuzzy, temporal fuzzy and GA-based fuzzy approaches.

In the following subsections, the EC-based and fuzzy logic-based soft computing techniques on HUIs mining are discussed in detail.

3.1 evolutionary computation-based approaches

Evolutionary computation is motivated by the concept of biological evolution, which provides the optimal solutions and solves the NP-hard problems in various domains and applications such as engineering design problems (Thiruganasambandam et al. 2019) and image processing [30]. Many algorithms are proposed in privacy preserving data mining (PPDM) to hide the sensitive data itemsets (Atallah et al. 1999; Dasseni et al. 2001; Hong et al. 2013; Lin et al. 2013; Daniel 1991). The privacy preserving utility mining (PPUM) extends PPDM to utilize the concept of utility to modify an item's quantity (Yeh and Hsu 2010). Authenticity is the major concern for data sanitization in the PPUM. Several evolutionary computation-based soft computing approaches are presented to hide the sensitive information in the PPUM.

The genetic algorithms (GAs) (Goldberg 1989; Holland 1992) become an active research area that provides probable solutions in a stipulated period of time and can be used to hide the sensitive HUIs in PPUM. GA is a meta-heuristic technique that is influenced by the concept of natural selection [39] and is widely used to solve the nonlinear and NP-hard problems (Karakatić and Podgorelec 2015; Mirjalili et al. 2020). It uses mainly three populations, namely selection, crossover and mutation to find promising solutions (Yang 2010; Goldberg and Holland 1988). GAs adopt various optimization problems like design, robotics, combinatorial optimization and data mining (Berson and Smith 1997).

Another major research area of evolutionary computation is particle swarm optimization (PSO) (Kennedy and Eberhart 1995) that is influenced by the flocking characteristics of birds to resolve the issues of optimization. In PSO, each particle is denoted by a set of velocities in the evolution process. PSO considers personal best (*pbest*) and global best (*gbest*) to keep the previous best particles and neighbor best particles, respectively, in the memory. PSO-based algorithms provide faster optimal solutions as compared to the GA-based algorithms when the number of populations is set low. PSO is used in many real-time scenarios like routing problems (Nouaouria et al. 2013), cuckoo search (Li and Yin 2016), clustering (Tsai et al. 2015), differential solutions (Gong and Ling 2010; Storn and Price 1997). Other applications based on PSO are still in progress to solve the optimization problems (Agrawal and Silakari 2014; Menhas et al. 2011; Nouaouria et al. 2013).

Most of the traditional algorithms of FIM and HUIM separately use the two measures, support and utility. Recently, the FIM and HUIM approaches are composed as a multi-objective optimization problem (MOP) (Zhang et al. 2018) in which the support and utility are considered simultaneously (Zhang et al. 2016) as the multi-objective view. The multi-objective optimization evolutionary (MOE) is another major research area of evolutionary computation to address the optimization problems (Zhang et al. 2016; Cheng et al. 2015; Zhang et al. 2015). MOE has various real-time applications in data mining (Mukhopadhyay et al. 2014), for example, clustering [59], classification (Levesque et al. 2012), association rule mining (Minaei-Bidgoli et al. 2013), feature selection (Venkatadri and Rao 2010) and pattern recommendation (Zhang et al. 2017). Several MOE algorithms are proposed where the Pareto optimal solutions are sparse in nature like sparse reconstruction (Li et al. 2018), neural network training (Fieldsend and Singh 2005) and portfolio optimization (Ponsich et al. 2013).

The bio-inspired computing optimizes a problem by iteratively improving a candidate solution concerning a given measure of quality. Biological systems provide abundant inspiration for the construction of high-performance computing models and intelligent algorithms, enabling the production of problem solving techniques with enhanced robustness and flexibility under complex optimization scenarios. Ant colony optimization (ACO) has been successfully applied to solve several optimization problems. They are especially efficient and effective in finding nearly optimal solutions from a very large solution space. The ant population selects similar paths and makes sure that other possible solutions have a good chance to be selected. Dolphin echolocation (DE) is another optimization problem in which the procedure for food capturing using dolphin's echolocation is comparable in finding an optimum solution. The dolphins originally search completely in the search space for food. No sooner it identifies a target; there is a search restriction made by that of the animals. The technique pretends DE is restraining its examination based on the distance of food (Soto et al. 2016).

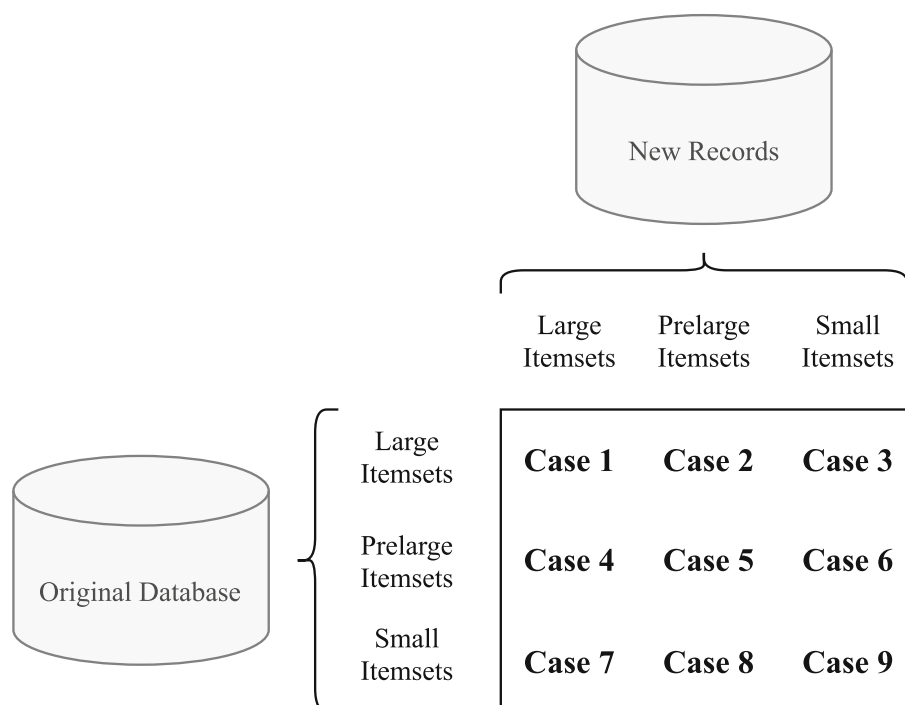
In the following subsections, we discuss the GA-based, PSO-based, multi-objective optimization EC and hybrid-based soft computing techniques on HUIs mining.

3.1.1 GA-based approaches

GA+ pre-large: In the past few decades, genetic algorithms (Berkhin 2006; Dasseni et al. 2001) become an active research area to provide the optimal solution for data sanitization in a limited amount of time.

In 2013, Lin et al. (2013) proposed a GA-based privacy-preserving utility mining algorithm to hide the sensitive HUIs in PPUM. It adopts three evaluation functions, namely maximal deleted utility for transaction deletion, downward closure

Fig. 2 Nine cases of the pre-large concept generated due to transaction deletion (Lin et al. 2013)



property (Liu et al. 2005) and pre-large concept ([69], Hong et al. 2001) to hide the sensitive HUIs. The pre-large concept avoids multiple dataset scans and efficiently maintains the discovered information in the dynamic datasets. It sets two thresholds for knowledge maintenance. Therefore, when some transactions are deleted from the datasets, nine possible cases may occur that are shown in Fig. 2. Cases 2, 3, 4, 7 and 8 do not disturb the final frequent itemsets. Case 1 may remove some existing frequent itemsets, and Cases 5, 6 and 9 may generate new frequent itemsets.

PPUM approach keeps the authenticity of the original dataset as much as possible. Therefore, it is a tedious task to modify the quantities of items in the datasets. The three side effects of the proposed algorithm, namely hiding failure, missing cost and artificial cost, are evaluated to determine the number of sensitive HUIs, number of missing HUIs and number of artificial HUIs, respectively. It is observed that the proposed algorithm performs well in terms of these side effects. The experimental results show that the proposed GA+pre-large algorithm significantly works well as compared to the simple GA algorithm (sGA) in terms of run-time and computational cost.

PPUMGA: GA algorithms (Berkhin 2006; Dasseni et al. 2001) provide the optimal solutions to solve difficult optimization problems. GAs are successfully applied in the field of optimization (Lin and Hong 2013; Lindell and Pinkas 2000), machine learning (Lindell and Pinkas 2000), neural networks (Agrawal and Srikant 2000) and many more.

In 2014, Lin et al. (2014) proposed a GA-based privacy-preserving utility mining approach named PPUMGA,

through transaction insertion to hide the sensitive HUIs in PPUM. It is the first work that uses a GA-based approach to disconcert the initial dataset for data sanitization via transaction insertion. It firstly finds a recommendable size of transactions as the number of genes in a chromosome, which can be determined by the maximal inserted utility value and the total utility value in the original datasets. The designed algorithm extracts the transactions from the original datasets as the optimal solution for a chromosome to be inserted into the dataset. A lower-utility threshold is also obtained to find the pre-large transaction-weighted utilization itemsets, thus avoiding the re-scanning time of the original dataset. An evaluation function is designed to consider the three side effects, namely hiding failure, missing cost and artificial cost. The adjustable weights are also considered for the three side effects according to users.

It adopts downward closure property (Liu et al. 2005) and pre-large concept (Hong et al. 2001) to reduce the computation for re-scanning the original dataset in the evaluation process. A pre-large concept is used to maintain the discovered information in the dynamic datasets (Hong et al. 2001). When some transactions are inserted into the datasets, nine cases are possible which are shown in Table 6. It is observed that if all large and pre-large itemsets are kept in the original dataset, Cases 2, 3 and 4 can be easily handled. In order to find the exact utility threshold, a sliding count technique is designed to filter the unpromising itemsets to reduce the evaluation cost. The pre-large concepts with two thresholds and a proposed sliding count are used to reduce the re-scanning

Table 6 Nine cases of pre-large concept generated by transaction insertion (Lin et al. 2014)

Cases	Original/new	Merged results
Case 1	Large/large	Large
Case 2	Large/pre-large	Large or pre-large, decided through existing information
Case 3	Large/small	Large, pre-large, or small, decided through existing information
Case 4	Pre-large/large	Large or pre-large, decided through existing information
Case 5	Pre-large/pre-large	Pre-large
Case 6	Pre-large/small	Pre-large or small, decided through existing information
Case 7	Small/large	Pre-large or small, when the number of transactions is small
Case 8	Small/pre-large	Pre-large or small
Case 9	Small/small	Small

time of the original dataset, thus enhancing the performance in the evaluation process.

In PPUM, data sanitization is divided into two types, namely item sanitization and transaction sanitization, to hide sensitive information. The proposed algorithm incorporates transaction sanitization to protect sensitive information. The experimental results show that the proposed algorithm has only a few side effects of hiding the failure of sensitive HUIs. It significantly reduces the computation time for re-scanning the original dataset because of using the pre-large concept. The proposed algorithm performs well as compared to the existing simple GA (GAs) and GA+ pre-large (Lin et al. 2013) in terms of execution time. The reason is that the proposed algorithm uses the sliding count to reduce the execution time.

HUPE_{UMU}-GARM and HUPE_{WUMU}-GARM: The major issue with two-phase HUIM algorithms (Liu et al. 2005; Chu et al. 2009) is that they generate a large number of candidates, thereby resulting in a significant amount of time to mine HUIs. Therefore, data analysts face two following major challenges: (1) Search space is exponential that leads to enormous time to mine HUIs and (2) it is a tough task to specify the minimum utility threshold to mine HUIs. To address these challenges, GA-based algorithms (Beasley et al. 1993) are proposed to efficiently mine HUIs from transactional datasets. Data mining is one of the major application areas of GAs (Berson and Smith 1997).

In 2014, Kannimuthu and Premalatha (2014) proposed two algorithms, namely HUPE_{UMU}-GARM (high-utility pattern extraction using genetic algorithm with ranked mutation using minimum utility threshold) and HUPE_{WUMU}-GARM (high-utility pattern extraction using genetic algorithm with ranked mutation without using minimum utility threshold) that utilize GA with ranked mutation to significantly mine HUIs from the transactional datasets. The authors claim that it is the first work that considers negative item values for mining HUIs and utilizes a genetic algorithm. HUPE_{UMU}-GARM utilizes a minimum utility threshold along with a transaction dataset. This algorithm

is preferred when search space and memory are the constraints. On the other hand, HUPE_{WUMU}-GARM is used to generate optimal HUIs without specifying the minimum utility threshold. The proposed algorithms adopt TWU model (Liu et al. 2005) to eliminate the unpromising itemsets at the initial stage. These algorithms utilize fitness function (Yao and Hamilton 2006) to compute the fitness value.

The experimental results show that HUPE_{UMU}-GARM efficiently works and achieves scalability, especially in the case of large datasets. On the other hand, HUPE_{WUMU}-GARM shows linear scalability and scales much better. Both algorithms efficiently extract HUIs from the datasets consisting of negative profit values. However, it is a non-trivial task to obtain HUIs as the initial chromosomes for the later evolution process. Therefore, it requires high amount of computational time to find the initial particles. Moreover, it may lead to generation of non-HUIs from the dataset.

sGA2DT and pGA2DT: Genetic algorithms [39] find optimal solutions using the principles of natural evolution. Earlier PPDM approaches (Agrawal and Srikant 2000; Verykios et al. 2004; Lindell and Pinkas 2000) use heuristic methods to sanitize the dataset by deleting partial items.

In 2015, Lin et al. (2015) proposed two algorithms, namely sGA2DT (simple genetic algorithm to delete transaction) and pGA2DT (pre-large genetic algorithm to delete transaction) for data sanitization to hide the sensitive itemset through transaction deletion in PPDM. The proposed algorithms delete the necessary transactions to hide the sensitive itemsets. The sGA2DT adopts the simple genetic algorithm (Holland 1992) to eliminate the transactions for hiding sensitive itemsets, while pGA2DT adopts a pre-large concept (Hong et al. 2001) to compute the side effects of artificial and missing itemsets by evaluating the fitness function without re-scan of the original dataset. The pre-large itemsets (PL) act as buffers and are used to reduce the movement of itemsets directly from large to small and vice versa when transactions are deleted.

An evaluation function is proposed to determine the optimal number of deleted transactions to hide the sensitive

itemsets. The adjustable weighting and dataset dissimilarities are also taken into account to measure the side effects as per the requirement of users. A fitness function is also designed to consider the three side effects, hiding failures, missing costs and artificial costs to determine the goodness of the chromosomes. The number of deleted transactions is pre-computed and pre-determine to minimize the side effects. The proposed algorithms adopt the pre-large concept to accelerate the execution time of re-scanning the original dataset. It also reduces the population size.

The experimental results show that pGA2DT performs well compared to sGA2DT in terms of the execution time, number of hiding failures, number of missing itemsets, number of artificial itemsets and dataset dissimilarity from the benchmark datasets. The pGA2DT algorithm performs well as compared to the sGA2DT algorithm in terms of the execution time. However, the dissimilarity between datasets requires improvement in case of the larger dataset. The greedy approach has good performance in execution time since it does not consider any of the side effects to evaluate transactions for hiding the sensitive itemsets; however, it incurs low performance due to these side effects. The dissimilarity of the dataset in the greedy approach shows better performance because it is randomly executed to delete transactions and those transactions may contain fewer itemsets without optimal selection for hiding sensitive itemsets. Although the proposed sGA2DT and pGA2DT algorithms can achieve a good performance in terms of side effects; they rely on the pre-defined weights of three side effects in their pre-defined fitness function. This problem causes the results of the side effects to be seriously influenced by the weighted values; the results are occasionally not optimized, which requires prior knowledge to set up these weighted values.

PPUMGAT+: Generally, collaboration among industries can work together to share information to achieve higher benefits and profits in business. However, the shared information can be extracted and analyzed by the other collaborators or competitors, thus decreasing its own benefits and causing security threats. To solve this issue, several PPDM (Amiri 2007; Dunning and Kresman 2013; Han and Ng 2007) and/or PPUM (Yeh and Hsu 2010; Lin et al. 2014; Yun and Kim 2015) algorithms are proposed to perturb the original dataset and produce a sanitized dataset. However, these algorithms are insufficient because of NP-hard problem. It is necessary to hide the sensitive information but discover the required information in decision-making.

In 2016, Lin et al. (2016) proposed the following two evolutionary algorithms: (1) PSO-based algorithm to efficiently mine HUIs in PPUM and HUIM. (2) GA-based privacy-preserving algorithm in PPDM.

The binary PSO model (Kennedy and Eberhart 1997) consists of the following four processes to mine HUIs: (1)

pre-processing (2) particle encoding (3) fitness evaluation (4) updating process. The pre-processing process first discovers the 1-HTWUIs based on the TWU model (Liu et al. 2005). The updating process uses the maximal pattern tree (MP-tree) to compress the valid combinations of the itemsets. The experimental results show that the proposed algorithm performs well as compared to HUPE_{UMU}-GARM approach (Kannimuthu and Premalatha 2014) in terms of execution time. The reason is that the proposed algorithm only generates the valid combinations of itemsets existing in the dataset, which can greatly avoid the computational problem in the evolution process.

The GA-based algorithm uses a pre-large concept (Hong et al. 2001) to accelerate the evaluation process and reduces the cost and space of the evolution process. It efficiently updates the discovered information in the dataset. The proposed algorithm calculates the utility of each item, utility of each transaction and utility of the whole dataset in the corresponding utility table in the original dataset. The experimental results show that GA-based algorithm with pre-large concept performs well as compared to the naive GA-based algorithm (Holland 1992) in terms of various minimum utility thresholds on the benchmark datasets. It is 2 or 3 times faster than the naive GA-based algorithm. The reason is that the Pre-large concept (Hong et al. 2001) is adopted in the proposed algorithm which is used to reduce the re-scanning of the original dataset. Therefore, the proposed algorithm only scans the original dataset once in the whole evolution process. The proposed PPUMGAT+ algorithm not only effectively mines HUIs but also provides a trade-off between privacy-preserving for the sensitive HUIs and mining performance for HUIM.

HUIM-IGA: The major issue with traditional HUIM algorithms (Liu et al. 2005; Yao and Hamilton 2006; Ahmed et al. 2009; Tseng et al. 2010; Liu and Qu 2012) is the growth of search space exponentially as the number of transactions and items in the dataset increase (Kannimuthu and Premalatha 2014). This issue occurs due to the combinatorial optimization problem (Ventura and Luna 2016). To address the limitations of HUIM, several EC-based approaches (Kannimuthu and Premalatha 2014; Song and Huang 2018; Lin et al. 2016, 2017; Wu et al. 2017) are proposed to find the optimal solutions in a limited amount of time. However, they consume lots of time to mine HUI that satisfy the minimum utility threshold. Moreover, some HUIs maybe miss during the evolution process (Song and Huang 2018).

In 2019, Zhang et al. (2019) proposed a HUIM approach based on the improved genetic algorithm (HUIM-IGA) to efficiently mine HUIs. The authors proposed the following four strategies: (1) A population diversity maintenance strategy is presented which persuasively decreases the number of missing HUIs in the search process and expands the search space to provide high quality solutions (Song and Huang

2018). The bitmap structure (Song and Huang 2018; Song et al. 2014) is used to mine HUIs according to TWU value of each 1-HTWUI (Liu et al. 2005). (2) A neighborhood exploration strategy for repeated HUIs is also designed to improve the local search ability of the proposed HUIM-IGA algorithm in the optimum solution regions. It is also used to accelerate the exploration of the new HUIs. (3) An individual repair strategy based on TWU sorting is also designed that reduces the invalid combinations to find HUIs. This process can ensure that a repaired individual is an effective combination of datasets. (4) An elite strategy (Deb et al. 2002) is also introduced which prevents the information loss of promising itemsets.

In order to check the effectiveness of population diversity maintenance strategy and elite strategy, two versions of the proposed algorithms are developed, namely HUIM-IGA_{DMS} (HUIM-IGA without diversity maintenance strategy) and HUIM-IGA_{ES} (HUIM-IGA without elite strategy). It is observed that the convergence speeds of HUIM-IGA_{DMS} and HUIM-IGA were similar in the early stage of evolution. However, in the middle and late stages of evolution, HUIM-IGA converged faster than HUIM-IGA_{DMS}. Therefore, the proposed HUIM-IGA can mine HUIs significantly faster than HUIM-IGA_{DMS}. Furthermore, the elite strategy can reduce the missing of HUIs and the loss of excellent solutions, so the convergence speed of HUIM-IGA with elite strategy is significantly faster than that HUIM-IGA_{ES}. Experimental results show that the proposed algorithm outperforms the state-of-the-art EC-based HUIM algorithms, Bio-HUIF-GA (Song and Huang 2018), Bio-HUIF-PSO (Song and Huang 2018), HUIM-BPSO (Lin et al. 2017), HUIM-BPSO_{sig} (Lin et al. 2016) and HUPE_{umu}-GRAM (Kannimuthu and Premalatha 2014), and three non-EC-based algorithms UP-Growth (Tseng et al. 2010), UP-Hist Growth (Dawar and Goyal 2015) and IHUP (Ahmed et al. 2009), in terms of the run-time, converging speed and promising HUIs.

GAHUIM: Several GA-based algorithms (Kannimuthu and Premalatha 2014; Song and Huang 2018; Zhang et al. 2019) are proposed to efficiently mine HUIs. However, there are still lots of scopes to further speed up the mining process by significantly consumption of memory space.

In 2021, Javangula (2021) proposed an algorithm named GAHUIM (HUI mining using genetic algorithm) to mine HUIs by parsing the dataset only once. In the proposed system, due to the requirements imposed by the high-utility itemset mining problem, a non-binary representation of the genetic algorithm has been proposed. The operators of genetic algorithm such as crossover and selection operation have been customized to efficiently operate in the mining problem. To reduce the explosion of candidate generation, an upper bound based on the sum of the itemset utility and remaining utility is used.

GA performs a search over a space of solutions to obtain the optimal solutions. The fitness function obtains the solutions from the initial population and refines it to obtain the optimal solutions. HUIM problem can be modeled as an optimization problem which consists of the following entities: (1) gene; (2) chromosome; (3) population; (4) fitness function; (5) breeding process; (6) convergence. Gene is a 1-D array that consists of the information about the set of items in a transaction. Figure 3 shows the structure of a gene. A chromosome is a set of genes representing an itemset in the transaction. Figure 4 shows the structure of a chromosome.

The population is a set of chromosomes. In HUIM problem, fitness selection has two functions: (1) HUISelect and (2) MatSelect. HUISelect copies chromosomes representing HUIs to output, while MatSelect chooses chromosomes with high probabilities to generate HUIs to mating pool. Two chromosomes are merged to form offspring. In the breeding phase, the chromosomes from the primary section in the mating pool are picked up one by one and the offspring are generated. The itemset utility is compared with the threshold to select the high-utility itemsets. The remaining utility is used to select prospective candidates for superset generation. The algorithm starts with an initial population comprising a single itemset. The fitness function computes the sum of the itemset utility and the remaining utility and selects the prospective itemsets for breeding. The children are generated by merging two prospective itemsets. The selection and breeding process are repeated until there are no more chromosomes in the population for further breeding.

Experimental results show that the proposed algorithm outperforms the existing state-of-the-art algorithms (Ahmed et al. 2009; Qu et al. 2019; Tseng et al. 2010, 2013) in terms of operational speed and memory requirement.

3.1.2 PSO-based approaches

BPSO_Proposed+: The genetic algorithm [39] is an optimization approach to solve the NP-hard and nonlinear problems. They are used to find the optimal solutions based on the designed fitness functions. The concept of PSO is motivated from the flocking characteristic of birds which is used to resolve the optimization issues (Kennedy and Eberhart 1995). The PSO was originally defined to solve the continuous valued spaces. In real-world applications (Kuo et al. 2011; Pears and Koh 2011), many problems are set as discrete variable spaces, such as scheduling and routing problems. Discrete (binary) PSO (BPSO) algorithm (Kennedy and Eberhart 1997) is proposed to address the problem of continuous PSO. Another optimization approach, BPSO is developed to solve the association rules (Sarath and Ravi 2013).

In 2015, Lin et al. (2015) presented a novel approach that adopts a binary PSO-based (BPSO) algorithm (Kennedy

Bill No. (BN)	Item Set (IS)	Utility (U)	Remaining Utility (RU)	Remaining Item Set (RIS)
------------------	------------------	----------------	---------------------------	-----------------------------

Fig. 3 Structure of a gene (Javangula 2021)

Chromosome - IS					
Gene Representing IS	Bill No. (BN)	Item Set (IS)	Utility (U)	Remaining Utility (RU)	Remaining Item Set (RIS)
	Bill No. (BN)	Item Set (IS)	Utility (U)	Remaining Utility (RU)	Remaining Item Set (RIS)
	*				
	*				

Fig. 4 Structure of a chromosome (Javangula 2021)

and Eberhart 1997), to efficiently mine HUIs. The proposed algorithm first finds the 1-HTWUIs based on TWU model (Liu et al. 2005). The particle is thus encoded as the set of binary variables corresponding to the sorted order of 1-HTWUIs. The particles are then evaluated to find the satisfied HUIs. After that, the particles are correspondingly updated by velocities, *pbest*, *gbest* and *sigmoid* function. This iteration is repeated until the termination criteria is achieved. After that, the set of HUIs is discovered. It greatly reduces the combinatorial problem in the evolution process. A *sigmoid* function is also utilized for discovering HUIs in the evolution process.

The experimental results show that the proposed algorithm performs better compared to the GA-based HUPE_{umu}-GRAM (Kannimuthu and Premalatha 2014) algorithm in terms of run-time and discovered HUIs. Furthermore, the proposed PSO-based algorithm can generate nearly the same number of HUIs compared to the state-of-the-art FHM algorithm (Fournier-Viger et al. 2014) especially when the minimum utility threshold is set higher in the condensed datasets. The reason is that the size of a particle is associated with the number of 1-HTWUIs, fewer computations are required when the minimum utility threshold is set higher.

BPSO_Proposed: The traditional algorithms of HUIM (Liu et al. 2005; Yao and Hamilton 2006; Ahmed et al. 2009;

Tseng et al. 2010; Liu and Qu 2012) incur the “exponential problem” of a very large search space while the number of distinct items or the size of the dataset is very large. Evolutionary computation is an efficient way to find the optimal solutions using the principles of natural evolution (Cattal et al. 2009).

In 2015, Lin et al. [100] presented a binary PSO-based (BPSO) algorithm with an extended maximal pattern (MP)-tree structure (Kennedy and Eberhart 1997) to efficiently mine HUIs. It consists of the following steps: (1) pre-processing; (2) particle encoding; (3) fitness evaluation; (4) updating processes. In the first pre-processing step, it first finds the 1-HTWUIs based on TWU model (Liu et al. 2005). Then, MP-tree structure reduces the multiple dataset scans by early pruning the invalid combinations of the particles. It accelerates the mining process to extract HUIs in the evolution process. MP-tree structure stores only valid combinations of the itemsets to avoid the invalid generations of the particles in the updation process. It greatly reduces the computations of multiple dataset scans. The OR and NOR operators determine that the valid combinations of the combined itemsets are generated or not. In the second particle encoding process, the items are sorted according to the increasing order of their 1-HTWUIs. In the third fitness evaluation process, the particles are evaluated to find the *pbest* and

gbest particles in the evolution process. Finally, in the updating process, the particles are correspondingly updated by velocities, *pbest*, *gbest* and *sigmoid* function. The MP-tree structure is used here to generate only the valid combinations of the particles. The proposed binary (discrete) PSO-based algorithm presents a sigmoid updating strategy to further generate the valid combinations of itemsets.

Experimental results show that the proposed algorithm works well as compare to the HUPE_{umu} -GRAM algorithm (Kannimuthu and Premalatha 2014) in terms of run-time and discovered HUIs under different minimum thresholds. The reason is that HUPE_{umu} -GRAM generates many unpromising itemsets and scans the dataset multiple times. It hardly finds promising HUIs. However, the proposed algorithm can efficiently find the valid HUIs by concerning both the local and global results. It greatly avoids the combinational problem in the evolution process. The proposed algorithm can generate nearly the same number of HUIs compared to the state-of-the-art two-phase algorithm, especially when the minimum utility threshold is set higher. The reason is that the size of a particle is associated with the number of 1-HTWUIs, and fewer computations are required when the minimum utility threshold is set higher.

HUIM-BPSO⁺: As we have seen previously, two heuristic GA-based algorithms, HUPE_{umu} -GRAM (Kannimuthu and Premalatha 2014) and HUPE_{wumu} -GRAM (Kannimuthu and Premalatha 2014) are proposed to efficiently mine HUIs with and without specifying minimum utility threshold, respectively. However, they require the crossover and mutation operations to randomly generate the next solutions in the evolution process. Moreover, a high amount of computation is needed to discover the satisfied HUIs in the early stage. This is insufficient, especially when the number of distinct items are very large in the dataset.

In 2016, Lin et al. (2017) proposed a discrete PSO-based algorithm, named HUIM-BPSO (HUIM-binary PSO) which integrates the sigmoid updating strategy and TWU model (Liu et al. 2005), to find the complete set of HUIs from the condense datasets. The authors presented a BPSO-based algorithm (Kennedy and Eberhart 1997) with an extended OR/NOR tree structure to discover HUIs. As the number of iterations is increased, the proposed algorithm obtains accurate precision to get the identical number of rules as the conventional two-phase algorithm (Liu et al. 2005). This OR/NOR-tree structure is used to determine whether the combined items of a particle exist in the dataset, which can be used to avoid the redundant combinations in the evolution process.

The experimental results show that HUIM-BPSO⁺ algorithm reduces the search space to obtain valid itemsets, therefore, speeding up the computations as compared to the GA-based algorithms, HUPE_{umu} -GRAM (Kannimuthu and Premalatha 2014) and HUPE_{wumu} -GRAM (Kannimuthu and

Premalatha 2014) in terms of run-time, number of discovered HUIs, number of determining patterns, convergence speed, under different minimum utility thresholds. The proposed HUIM-BPSO⁺ algorithm generates nearly the same number of HUIs as the TWU model, especially when the minimum utility threshold is set high in the condensed datasets. The reason is that the size of a particle is associated with the number of 1-HTWUIs; fewer computations are required when the minimum utility threshold is set higher. However, when the number of iterations is increased, HUIM-BPSO⁺ can achieve higher accuracy to generate the same number of HUIs as compared to the traditional TWU model.

HUIM-BPSO_{sig}: The PSO-based algorithms (Kennedy and Eberhart 1995, 1997) require few parameters as compared to GA-based approaches (Salleb et al. 2007; Kannimuthu and Premalatha 2014). PSO is a bio-inspired and population-based approach to find the optimal solutions by adopting the velocity to update the particles (Kennedy and Eberhart 1995).

In 2016, Lin et al. (2016) presented a binary particle swarm optimization (BPSO)-based algorithm (Kennedy and Eberhart 1997) to efficiently mine HUIs from the transactional dataset. It consists of pre-processing, particle encoding, fitness evaluation and the updating phases to mine HUIs. The proposed algorithm adopts the TWU model to find the number of 1-HTWUIs as the particle size, which can greatly reduce the combinational problem in the evolution process. HUIM-BPSO_{sig} encrypts the particles as the binary variables. It first sets the number of discovered 1-HTWUIs based on the TWU model (Liu et al. 2005), to significantly reduce the combinatorial issue in the evolution process. The *sigmoid* function is used in the updation process of the particles to design the HUIM-BPSO_{sig} algorithm.

The experimental results show that the proposed HUIM-BPSO_{sig} algorithm outperforms the state-of-the-art algorithm, HUPE_{umu} -GRAM (Kannimuthu and Premalatha 2014) in terms of run-time, number of discovered HUIs, number of patterns and convergence speed under different minimum thresholds. The reason is that HUPE_{umu} -GRAM suffers from combinational problem of the generated itemsets and requires several parameters in the evolution process.

MOPSO-based data sanitization model: The data sanitization in PPDM is an important research issue, which hides the sensitive information using the distortion method (Agrawal and Srikant 1994; Oliveira and Zaïane 2002; Verykios et al. 2004). The sanitization process has the following three side effects (Verykios et al. 2004): (1) hiding failures; (2) missing costs; (3) artificial costs (Verykios et al. 2004). Another side effect is dataset dissimilarity which shows the size of deleted transactions before and after sanitization process (Lin et al. 2015). Therefore, it is a tough task to obtain the optimal solutions because of NP-hard problem. Several GA-based algorithms (Lin et al. 2014, 2015)

are proposed to mitigate the effect of these side effects; however, they delete the important sensitive information in the dataset during the evolution process.

To alleviate the above side effects, Wu et al. (2019) proposed a data sanitization model, based on a multi-objective PSO (MOPSO) algorithm (Coello and Lechuga 2002) to mine HUIs in PPDM. However, MOPSO algorithm cannot be directly applicable to PPDM problem as dominance relations should be used to optimize the transactions for deletion. Hence, a sanitization algorithm based on the hierarchical clustering method is presented for PPDM. In the pre-processing phase, the items have to be placed in the confidential information (CI) list before the data sanitization process. In the evolution process, the particles are evaluated to obtain the optimized solutions in the next iteration. Each particle has MDT vectors (maximized deleted transactions) in the sanitization process. Since the fitness value of each particle cannot be determined by *pbest*, therefore, the non-domination relation is adopted in the updation process for *pbest*. The proposed algorithm resolves the problem of PPDM with transaction deletion and minimizes the side effects and generates the non-dominated solutions. It uses two strategies, namely LUS (local updating strategy) and GUS (global updating strategy) to obtain the non-dominated solutions.

The proposed algorithm achieves significant performance in contrast to the existing single-objective approach, cpGA2DT (Lin et al. 2014) and the multi-objective approach, NSGA2DT [106] in terms of hiding failure, missing cost, artificial cost and dataset dissimilarity. It is observed that in the proposed algorithm, dataset dissimilarity increases because more redundant information is eliminated from the dataset.

GMPSO: Data sanitization is a process to hide sensitive information by perturbing the dataset; however, it incurs side effects, namely hiding failure, missing costs, and artificial costs. Two GA-based algorithms, namely cpGA2DT (Lin et al. 2014) and pGA2DT (Lin et al. 2015), are proposed to mitigate these side effects; however, they depend on the pre-defined weights of side effects in their fitness function. It is quite challenging to obtain the weight of side effects due to lack of prior knowledge. To address this issue, EMO-based algorithm (Cheng and Pan 2014) is proposed to consider “data distortion” and “knowledge distortion” in the sanitization process through item deletion. However, it suffers from the incomplete knowledge of the decision-making process as it directly deletes the items in the dataset.

In 2019, Tsu-Yang Wu et al. [108] proposed a grid-based multi-objective particle swarm optimization algorithm, named GMPSO, based on the MOPSO framework (Coello and Lechuga 2002), to discover the optimal solutions for data sanitization in privacy-preserving data mining. This is the first work of its kind to design the MOPSO-based framework in PPDM. In the pre-processing step, the frequent itemsets

and pre-large itemsets are obtained and, respectively, placed into FI (Set of frequent itemsets) and PF (Set of pre-large itemsets) lists for later processing. In the evolution step, two strategies, namely global best (*gbest*) and personal best (*pbest*), are used to update the evolution process for PPDM to obtain the non-dominated solutions. The pre-large concept (Hong et al. 2001; Lin et al. 2009) is used to decrease the multiple dataset scans and speeds up the evolution process. The local updating strategy (LUS) and global updating strategy (GUS) are used to obtain the diverse set of non-dominated solutions of the MOP. They can be able to discover a better diversity of the desired solutions.

The experimental results show that the proposed algorithm significantly works well as compared to the state-of-the-art single-objective cpGA2DT (Lin et al. 2014) and multi-objective NSGA-II-based [106] approaches, in terms of flexibility, hiding failure, missing costs, artificial costs, execution time and scalability. The proposed GMPSO algorithm does not require crossover and mutation operations to generate the next populations. However, it suffers from a minor side effect of data dissimilarity as compared to the benchmark algorithms.

HUIM-AF: Heuristic methods are suitable to solve combinatorial problems because they can explore a very large search space to obtain the optimal solutions within the specified time. The artificial fish swarm (AFS) (LI and QIAN 2002) is a heuristic method that is inspired by the collective movement of fish and their social behavior. As compared to the GA (Kannimuthu and Premalatha 2014; Zhang et al. 2019), PSO (Lin et al. 2016; Song and Li 2020) and other heuristic approaches (Song and Huang 2018; Yang et al. 2017) for HUIM problems, the artificial fish swarm algorithm (AFSA) is more suitable because it only records the current position, but not other previous knowledge, of each artificial fish (AF). This approach is quite useful to get the optimal diversified results for the large search space of HUIM problems.

In 2021, Song et al. (2021) proposed an algorithm named HUIM-AF (high-utility itemset mining for artificial fish swarm) to mine HUIs from the transactional dataset. The proposed HUIM-AF models the following three behaviors of AF: (1) swarm; (2) prey; and (3) follow. Swarm is the behavior whereby a fish tends to assemble to protect from danger while avoiding overcrowded areas. Preying is the behavior whereby a fish moves to a specific location with the highest probability (concentration) of food. Follow is a behavior whereby a fish finds a location with a high concentration of food, and other fish follows. A position vector (PV) is used to represent the position of AF. PV keeps the 1-HTWUIs in the binary vector as per their increasing order. The utility of itemset is used directly as the object of optimization as used in Lin et al. (2016), Song and Huang (2018).

The experimental results show that the proposed HUIM-AF outperforms the existing state-of-the-art algorithms, $HUPE_{UMU}$ -GARM (Kannimuthu and Premalatha 2014) and $HUIM-BPSO_{sig}$ (Lin et al. 2016) in terms of execution time and a number of discovered HUIs. It is observed that the proposed algorithm can leap over local optima effectively, thereby resulting in the consistency with HUIM problems with a large number of optimized diversified results.

3.1.3 ACO-based approaches

HUIM-ACS: Ant colony algorithms with other heuristic methods are successfully applied in the field of data mining (Zhang and Ji 2013). Ant colony system (ACS) is an EC algorithm that is suitable for discrete solution space. If the heuristic function is properly set in ACS, then it always performs better than other EC algorithms (Kennedy and Eberhart 1997; Kannimuthu and Premalatha 2014; Lin et al. 2016).

In 2016, Wu et al. (2016) proposed an ant-based framework named HUIM-ACS (high-utility itemsets mining by ant colony system) to mine HUIs by using a specific routing graph and TWU model. This work is later extended in Wu et al. (2017). ACS (Dorigo and Gambardella 1997) is an expanded version of the ant system (Coloni et al. 1991; Dorigo et al. 1996), based on the observation that natural ant colonies use to search the food. The proposed HUIM-ACS algorithm first finds 1-HTWUIs based on the TWU model (Liu et al. 2005) by constructing an ant routing graph. Then, it removes unused items and reduces the size of the routing graph. A routing graph is generated before all of the ants start their tours. Each node in the routing graph represents an itemset that can be estimated whether it is HUI or not. A heuristic function is used to guide ants and quickly obtains HUIs. It effectively guides the ant's population. HUIM-ACS only calculated the TWU of an itemset when an ant arrives at the related node for the first time. There are following three situations arises for the TWU of the itemsets: (1) There is no information about the TWU for the paired items. (2) The process keeps all the TWUs for the paired items. (3) Some of the TWUs for the paired items exist in the process, but some do not.

There are following two reasons to perform the pruning processes: (1) to avoid tracing the duplicate tour by an ant and (2) to remove some branches which have no chance to discover any HUIs from the routing graph. Therefore, two pruning techniques, namely positive pruning rule and recursive pruning rule, are proposed to generate same routes which can be followed by ants as used before. They prune the routing graph to reduce the search space for HUIs. The proposed algorithm applies the local updating rule and global updating rule for mining HUIs.

The experimental results show that the proposed algorithm performs well as compared to two evolutionary computation

algorithms, $HUPE_{umu}$ -GRAM (Kannimuthu and Premalatha 2014) and HUIM-BPSO (Lin et al. 2017), and two non-evolutionary computation algorithms, HUI-Miner (Liu and Qu 2012) and EFIM (Zida et al. 2015), in terms of the execution time, a number of discovered HUIs and convergence speed from the large datasets. The proposed algorithm not only enhances the performance for mining HUIs by ACS operators but also provides a checking mechanism to see if all of HUIs in the dataset are being discovered or not. However, an efficient tree structure (routing graph) could be adopted to map the total solution space in order to obtain the optimal global solutions.

ACHUIM: The infrequent (rare) items may sometimes produce exceptionally interesting patterns, thereby resulting in high profits. Therefore, it is not a good idea to ignore infrequent items in the dataset from the business point of view. Although several algorithms (Haglin and Manning 2007; Pillai et al. 2013; Gupta and Chauhan 2015) are proposed to efficiently mine the infrequent itemsets from the transactional datasets. However, there is still lot of research gap to mine the high-utility infrequent itemsets ([124], Goyal et al. 2015). Several ant colony optimization techniques (Dorigo and Gambardella 1997; Wu et al. 2017; Seidlova et al. 2019) are proposed to obtain the best optimal solutions from the large search space in the dataset. However, there is still a need to design an efficient framework to produce better results.

In 2020, Arunkumar et al. (2020) proposed a customized ant colony algorithm named ACHUIM, to efficiently mine the high-utility infrequent itemsets (HUIIs) from the transactional dataset. The procedure of the proposed algorithm is as follow: (1) Firstly, the initial utility value and TWU of each item or itemset are calculated for every node and update these values according to the ant traversal system. As the ant system is populated with more ants, then all ants traverse the same route concurrently. (2) Secondly, the pruning method using downward closure property on TWU, is adopted to reduce the search space according to the lexicographical order of values. If TWU of a node is no less than the minimum utility value, then it is considered in the mining process, otherwise not. The ant in the proposed ACHUIM system visits only the node that has been set as its target, which reduces the unnecessary time spent on traversing the other nodes on the graph. As the unnecessary traversal has been evaded, the execution time of the proposed system gets considerably reduced. The ant's traversal toward the specified target is achieved by adding a modest intelligence to the ants artificially.

The experimental results prove the effectiveness of the proposed algorithm over the state-of-the-art algorithm, named HURI (High utility rare itemset) (Pillai et al. 2013), in terms of execution time and a number of generated HUIs on the benchmark datasets. As ants in the system work in

parallel, the execution time reduces proportionally with the increase in number of ants.

3.1.4 Multi-objective evolutionary computation-based approaches

MOEA-FHUI: FIM (Agrawal and Srikant 1994; Han et al. 2000) and HUIM (Ahmed et al. 2009; Tseng et al. 2010; Liu and Qu 2012, [9]) algorithms need to provide minimum support threshold and minimum utility threshold, respectively. However, due to a lack of prior knowledge, it is quite challenging to provide the optimal values for the minimum threshold. Various evolutionary computation-based algorithms (Kannimuthu and Premalatha 2014; Lin et al. 2016, 2017; Wu et al. 2017) are proposed to efficiently mine HUIs. However, they consider both measure's support and utility separately. An algorithm named TKU-Miner (Zhang et al. 2016) is proposed to consider both measures together. However, the users need to specify parameters, minimum support (*min_sup*) and minimum utility (*min_util*) in advance, but it is a tough task to provide the value for these parameters properly without prior knowledge.

In 2018, Zhang et al. (2018) proposed an efficient multi-objective evolutionary algorithm to consider both support and utility simultaneously. The proposed MOEA-FHUI (multi-objective evolutionary algorithm for mining frequent and high-utility itemsets) algorithm based on the MOEA/D method (Zhang et al. 2017) mines frequent and high-utility itemsets without specifying the prior parameter, *min_sup* and *min_util*, from the transactional dataset. MOEA/D gives an adequate balance between the convergence and populations diversity in the evolution process. This is the first work of its kind that considers both frequent and high-utility itemsets as a multi-objective optimization problem. MOEA/D breaks down a complex multi-objective optimization problem into several simple multi-objective optimization sub-problems. MOEA/D generates good convergence (Cai et al. 2015) and evenly distributes solutions than that of NSGA-II (Deb et al. 2002) and SPEA-II [129]. It uses the Tchebycheff value (Miettinen 2012) for optimization sub-problems. MOEA-FHUI checks the performance with two metrics, namely hypervolume (Zitzler and Thiele 1998) and coverage (Bobadilla et al. 2013). Hypervolume is used in multi-objective optimization problems. The larger hypervolume value shows the better solutions obtained. Coverage is used in recommendation systems to compute the diversity of the recommendation lists. The higher coverage value shows the more diverse items in the recommendation lists. A population initialization strategy is also suggested for HUIM problem that gives a guarantee to generate diverse solutions for the recommendation.

In order to show the effectiveness of MOEA-FHUI, two baseline versions, namely NSGA2-FHUI and SPEA2-FHUI,

are proposed that are based on framework of NSGA2 (Deb et al. 2002) and SPEA2 [129], respectively. It is observed that MOEA-FHUI is slightly better than NSGA2-FHUI and significantly performs well than SPEA2-FHUI. The experimental results show that the proposed MOEA-FHUI performs well as compared to the baseline algorithms, FP-Growth (Han et al. 2000), HUI-Miner (Liu and Qu 2012), HUIM-ACS (Wu et al. 2017) and TKU-Miner (Zhang et al. 2016) in terms of hypervolume and coverage for most of the datasets. It is observed that MOEA-FHUI can simultaneously provide many recommendations for decision-makers in just one run. Moreover, the efficiency of MOEA-FHUI could be further improved, especially when the number of transactions (or items) in the dataset becomes very large. However, the run-time and memory usage of MOEA-FHUI are escalated as the number of transactions and/or items grows.

CP-MOEA: The main advantage of MOEA-FHUI algorithm (Zhang et al. 2018) is that the users do not need to specify the minimum support threshold and minimum utility threshold. Moreover, it provides multiple items recommendation for decision-makers in only one run. However, it does not take into consideration the important property of the closest itemset.

To further continue this work to utilize the potential of MOEA, Cao et al. (2019) proposed closed itemset property-based multi-objective evolutionary approach (CP-MOEA) adopts a similar framework used in NSGA-II (Deb et al. 2002) to mine the frequent and high-utility itemsets. The closed itemset property means that if the superset of an itemset is closed, then this itemset must be dominated by its superset. CP-MOEA optimizes the notion of support (Han et al. 2007) and utility (Gade et al. 2004; Tseng et al. 2010; Liu and Qu 2012, [9]) simultaneously. The proposed algorithm adopts binary encoding as the individual representation. If the item appears in each individual itemset, then the corresponding bit is set to 1 or 0 for otherwise. Two updating strategies, namely updating strategy based on closed itemsets (USC) and updating strategy based on approximate-closed itemset (USA), are designed to enhance the quality of the mining process. It is observed that the proposed updating strategies can not only accelerate the convergence of population but also improve the diversity of the population.

In order to check the effectiveness of the proposed strategies of CP-MOEA, two variants, namely CP-MOEA without USC strategy (CP-MOEA(-USA)) and CP-MOEA without USA strategy (CP-MOEA(-USC)), are proposed. It is observed that CP-MOEA outperforms its variants CP-MOEA(-USA) and CP-MOEA(-USC), which adopt only one updating strategy on the benchmark datasets. The experimental results show that the proposed algorithm works well as compared to the state-of-the-art FHUI-MOEA (Zhang et al. 2018) approach in terms of convergence speed and hypervolume value.

SparseEA: In the literature, several sparse multi-objective optimization problems (MOPs) (Ma et al. 2016; Zhang et al. 2018; Zille et al. 2018) are proposed to obtain the optimal diversified results. However, they are ineffective for large sparse datasets because they do not consider the sparse nature of Pareto optimal solutions in the evolution process. Moreover, they incur very high computation cost.

In 2019, Tian et al. (2020) presented multi-objective optimization evolutionary algorithm (MOEA) named sparseEA to resolve the exhaustive sparsity MOPs. The proposed algorithm has a similar framework as used in NSGA-II (Deb et al. 2002). The following four sparse MOPs are introduced to increase the usefulness of the SparseEA: (1) feature selection (Xue et al. 2013) (2) pattern mining (Tang et al. 2012) (3) critical node detection (Lalou et al. 2018) (4) neural network training (Jin et al. 2004). The feature selection reduces the number of features and improves the performance of classification (Dash and Liu 1997). It is bi-objective MOP and used by several MOEAs (Xue et al. 2013; Hamdani et al. 2007). Pattern mining is used to maximize completeness and frequency. The critical node detection is used by the several MOEAs (Ventresca et al. 2018; Faramondi et al. 2019) to minimize the selected points and the pair-wise connectivity in the graph. The neural network training is used to minimize the error and complexity for optimization of the weights of a feed-forward neural network. Although these MOPs have different objectives and are from different fields, they consist of sparsity Pareto optimum solutions, therefore, grouped together in the same category of MOPs. The proposed algorithm solves the sparsity of MOPs with real and discrete variables. A novel population initialization strategy, named score assignment strategy, works efficiently and effectively to solve the sparse MOPs. The genetic operators are also suggested to guarantee the sparsity of the obtained solutions by considering the sparse characteristics of the Pareto optimum solutions. The proposed algorithm uses existing genetic operators for real variables, while new genetic operators are used for binary variables (Simões and Costa 1999; Ishibuchi et al. 2010). Furthermore, a test suite is proposed to assess the effectiveness of SparseEA. The test suite consists of eight sparse MOPs, namely SMOP1–SMOP8 to define various shape functions and landscape functions.

The experimental results show the effectiveness of the proposed sparseEA as compared to the state-of-the-art methods, NSGA-II (Deb et al. 2002), SPEA2 [129], SMS-EMOA (Beume et al. 2007), and EAGMOEA/D (Cai et al. 2015) in terms of hypervolume, test instances, convergence speed, diversity, complexities and error rates for sparse MOPs. However, the proposed algorithm evolves the population without consideration of interaction between decision variables. Therefore, it could be further improved by incorporating other effective strategies, for example, decision variable clus-

tering (Zhang et al. 2018) and Bayesian optimization (Wang et al. 2016).

MOEA-PM: There are few research works (Coello 2006; Zuo et al. 2015) which consider the three measures, support, occupancy and utility simultaneously and transformed as a three-objective optimization model. The model can be further improved to meet the actual needs.

In 2020, Fang et al. (2020) proposed an improved three-objective multi-objective optimization model for high-quality pattern mining (HQPM). The proposed MOEA-PM (multi-objective evolutionary algorithm for HQPM) discovers the high-quality patterns without specifying minimum support, minimum occupancy and minimum utility. The proposed work is an extended work of NSGA-II (Deb et al. 2002). The authors claim that this is the first work that considers three measures, namely support, occupancy and utility as a multi-objective optimization problem (MOP) (Zhang et al. 2017) for high-utility patterns mining (HUPM) (Fournier-Viger et al. 2019; Ahmed et al. 2021; Srivastava et al. 2021). The MOP involves two or more conflict objective functions to be maximized or minimized. The support measures the frequency of patterns that appear in the dataset. The occupancy measures the completeness of the patterns. The utility measures the profit of the patterns. These three measures are conflicting in nature. Therefore, the proposed algorithm performs the population evolution to find the balance among support, utility and occupancy. Hence, the users can get the best accurate recommendations.

Two population initializing strategies, namely MOEA-PM (initialization 1) and MOEA-PM (initialization 2), are developed to effectively distribute the population in the feasible solution space from the three perspectives, namely randomness, generality and compositional (Kazimipour et al. 2014). A bitmap (bit-table) structure is used to represent the dataset in the concise form that is quite common in the pattern mining algorithms (Dong and Han 2007; Burdick et al. 2005; Song et al. 2008, 2014). An auxiliary tool is designed to effectively reduce the search space and refine the exploration ability of solution space. The top- k pattern recommendation technique (Zhang et al. 2017) recommends the user to choose the best k solutions by using the Euclidean distance method.

In order to check the effectiveness of auxiliary tool used in MOEA-PM (initialization 1) and MOEA-PM (initialization 2), two variants are proposed, namely MOEA-PM(initialization 1)[−] and MOEA-PM(initialization 2)[−] in which the auxiliary tool is removed. It is observed that the convergence speed of MOEA-PM(initialization 1) and MOEA-PM(initialization 2) is significantly faster than that of MOEA-PM(initialization 1)[−] and MOEA-PM(initialization 2)[−]. Therefore, it is proved that the auxiliary tool speeds up the convergence of the proposed algorithm. The experimental results show that MOEA-PM works well as compared to the state-of-the-art methods, SparseEA (Tian et al. 2020),

MOPM (Zhang et al. 2017) and MOEA-FHUI (Zhang et al. 2018) in terms of the quality of results, efficiency and converging speed. It is observed that MOEA-PM (initialization 2) has a better convergence speed than that of MOEA-PM (initialization 1). The proposed algorithm is not only relatively complete, but at the same time, it also has both frequent and high utility in the dataset. Since there is less research on modeling problems from the perspective of multi-objective optimization, the proposed algorithm could further improve the modeling problems to be suitable for real-world applications.

3.1.5 Hybrid approaches

Three-phase mining: In the past decades, several utility mining algorithms (Erwin et al. 2007; Ahmed et al. 2009; Li et al. 2012) are proposed to consider the utility of itemsets (e.g., profit, quantity, etc.) as a piece of useful information to mine HUIs from the transactional datasets. However, the utility of an itemset is considered constant and independent. The concept of discount strategies is widely applicable in the supermarkets and may differ in different time periods. Hence, they play a vital role in mining of HUIs. It is a quite challenging task to calculate the profits of itemsets using discount strategies because HUIs in a supermarket may differ under the different discount strategies. Therefore, the discount strategy is a vital factor in HUIM problems.

In 2014, Lin et al. (2015) proposed a three-phase mining algorithm to discover HUIs with discount strategies. The proposed algorithm consists of the following three phases: (1) During the first phase, the utilities of items are calculated in the transaction-item-utility (TIU) table which consists of three tables, namely the transaction table, discount strategies table and cost-sale price table. The transaction table includes the records of the purchase history of customers. The discount strategies table includes the records of discount strategies of all items with their corresponding values. The cost-sale price table includes the cost and sale price of all items. (2) In the second phase, all HTWUIs are discovered level-by-level based on the TWU model (Liu et al. 2005). (3) During the third phase, one more dataset scan is performed to check all HTWUIs for their actual utility values. These three phases discover all HUIs with discount strategies. The TIU table enhances the execution efficiency of the proposed algorithm. A complete TIU table includes the TIU of all items, transaction utility (TU) of all transactions and TWU of all items.

The experimental results show that the proposed algorithm performs well in terms of execution time and scalability on the benchmark datasets. The execution time decreases with the increase in minimum utility threshold. However, it increases approximately linearly as the data size increases. The reason is that more computation is required for longer

transactions. It is observed that the changes in discount strategies have a significant impact on the mining of HUIs.

Bio-HUIF: The previous bio-inspired algorithms (Kannimuthu and Premalatha 2014; Lin et al. 2017, 2016) maintain the optimal values of one population in the next iteration. However, the distribution of HUIs is not even. Therefore, some results may miss with a certain number of iterations because there may be misleading values if the desired result is searched with the best values from the previous one.

In 2018, Song and Huang (2018) proposed a bio-inspired-based HUIM framework (Bio-HUIF), adopts the roulette wheel selection to find the complete set of HUIs from the transactional datasets. The roulette wheel selection is applied to all the discovered HUIs to determine the initial target of the next population. The authors proposed three novel Bio-HUIF algorithms, namely Bio-HUIF-GA, Bio-HUIF-PSO, and Bio-HUIF-BA which are based on GA (Holland 1992), PSO (Kennedy and Eberhart 1995) and BA (Bat Algorithm) (Yang 2011), respectively. The GA [39] is a bio-inspired technique in which each individual has a fitness value to represent the quality of solutions. PSO (Kennedy and Eberhart 1995) is also a bio-inspired technique that simulates the foraging behavior of birds or fish. It uses the population-based approach to determine the optimal solutions by incorporating the velocity of updated particles. BA (Yang 2011) is another bio-inspired technique that uses the echolocation behavior of bats which vary the pulse rates and loudness of emissions when searching for prey and avoiding obstacles. They are also used to solve optimization problems.

The three strategies, namely bitmap dataset representation (Song et al. 2014), promising encoding vector checking (PEVC) and bit difference sets, are used to speed up HUI discovery process. The original dataset is firstly transformed into a bitmap which is an effective representation way for mining HUIs (Song et al. 2014). The size of the encoding vector represents each individual (chromosome, particle, or bat) is equal to the number of 1-HTWUIs in the dataset. During the first dataset scan, the 1-HTWUIs are determined based on TWU model (Liu et al. 2005). Then, a bitmap representation is constructing by using PEVC pruning strategy. Finally, a comprehensive set of HUIs is obtained.

The experimental results show that the proposed Bio-HUIF efficiently works well as compared to the existing bio-inspired algorithms, HUPE_{umu}-GRAM (Kannimuthu and Premalatha 2014) and HUIM-BPSO (Lin et al. 2017), and two state-of-the-art HUIM algorithms IHUP (Ahmed et al. 2009) and UP-Growth (Tseng et al. 2013) in terms of the execution time, number of obtained HUIs and converging speed. It is observed that the proposed Bio-HUIF discovers more HUIs than HUPE_{umu}-GRAM and HUIM-BPSO with fewer number of iterations. It also improves the diversity of solutions within a limited number of iterations.

HUIM-ABC: The main issue with GA-based algorithms, HUPE_{umu} -GRAM (Kannimuthu and Premalatha 2014) and HUPE_{wumu} -GRAM (Kannimuthu and Premalatha 2014) is the pre-mature convergence, which means that these algorithms can easily fall into local optima.

In 2018, Song and Huang (2018) proposed an efficient algorithm named HUIM based on artificial bee colony (HUIM-ABC) to mine HUIs. The proposed algorithm is based on the artificial bee colony (ABC) [166] which is inspired by the foraging characteristics of bees. The ABC approach has two unique properties: (1) decentralized control (2) self-organization. The authors claim that this is the first work that applies ABC approach to mine HUIs. The proposed HUIM-ABC algorithm consists of the following three steps: (1) Formulate the problem from the perspective of ABC approach. (2) Bitmap structure is used to represent both information representation and pruning of search space. It significantly speeds up the mining process. (3) The size of information of the discovered HUIs is used to produce new nectar sources instead of the generation of new candidates.

The bitmap structure is used to transform the original dataset into a nectar resource. It represents the information in the compact form and prunes the search space and thereby significantly speeds up the process to discover HUIs. After transforming the dataset into a bitmap, it is natural to encode each nectar source in a bit vector. The length of this vector is equal to the number of 1-HTWUIs. Based on the bit vector, a new nectar source generation in the HUIM-ABC algorithm can be achieved by only changing the value of 1 bit in the old nectar source, either from zero to one or from one to zero.

ABC algorithm has three types of bees, namely employed bees, onlooker bees and scout bees. The number of employed bees equals the number of nectar sources. Every employed bee produces a new solution from the old solution. When all employed bees have finished their searching process, they share the fitness (nectar) information of their solution (nectar sources) with the onlookers. Each onlooker bee selects one of the memorized nectar sources depending on the fitness value obtained from the employed bees. The scout bees use the direct nectar source generation (DNSG) strategy to frequently search the new nectar sources to generate more HUIs. The bit-wise operation and direct utility computation are performed to limit the search space.

The experimental results show that the proposed HUIM-ABC approach is very effective as compared to the algorithms HUPE_{umu} -GRAM (Kannimuthu and Premalatha 2014), HUPE-BPSO_{sig} (Lin et al. 2016) and HUPE-BPSO (Lin et al. 2017) in terms of the run-time, the number of obtained HUIs and converging speed. The reason is that the DNSG strategy of the HUIM-ABC algorithm generated new nectar sources by making use of the discovered HUIs rather than completely at random. Moreover, the useless operations of utility calculation can be avoided by performing the bit-wise

operations to detect the unpromising bit vectors. Hence, new nectar sources can be generated within the range of size of most discovered results.

MOEA-HEUPM: The traditional mining algorithms (Fournier-Viger et al. 2014; Gan et al. 2021) are designed to mine the high frequent or high-utility patterns by considering support and utility individually from the uncertain datasets. They are not designed to mine the required information from the uncertain datasets which consider both measures (utility and uncertainty) together as a multi-objective optimization problem. The utility measure is considered as a semantic method to access the value of a pattern; on the other hand, the uncertain measure is considered as an objective method to access the reliability and existence of a pattern. It is a non-trivial task to consider both measures to mine HUIs from uncertain datasets. The reason is that both measures conflict with each other, thereby resulting in the useless extracted patterns.

In 2020, Ahmed et al. (2021) proposed the multi-objective evolutionary approach to discover the high expected utility patterns mining (MOEA-HEUPM) framework in a limited time-period from the uncertain dataset. The proposed algorithm considers the utility and uncertainty simultaneously to extract the set of non-dominated high expected utility patterns (HEUPs) based on the evolutionary computation from the uncertain environment. The proposed algorithm does not need prior knowledge (minimum utility threshold and minimum uncertain threshold) to discover the information. But, instead of prior knowledge, it mines more meaningful and unique non-dominated patterns to meet the decision.

Two strategies, namely meta-itemset-selection and transaction-itemset-selection, are proposed to select the patterns. For transaction-itemset-selection, 50 percent of the population are selected from the dataset based on the utility probability. On the other hand, remaining 50 percent of the population for the meta-itemset-selection, only one item in a transaction is encoded and selected based on the uncertainty probability. The transaction itemset quickly achieves the stable convergence of the solutions, while meta-itemset generates new off-springs from parents with high diversity.

Firstly, a weight-based algorithm, named Tchebycheff (Zhang et al. 2018), is incorporated to quickly obtain the non-dominated solutions based on the utility and uncertainty (multi-objective) concept. Secondly, two encoding schemes, namely binary encoding and value encoding, are used to obtain the diversity and high convergence of the derived solutions (Mukhopadhyay et al. 2014). Finally, the proposed algorithm utilizes MOEA/D procedure (Wang et al. 2016) for the evolution progress. After the iteratively evolutionary progress of MOEA/D, the final population is generated and produced. The individuals in the population are then sorted and selected as the non-dominated solutions for the final results.

The experimental results show that the proposed MOEA-HEUPM performs well compared to the existing state-of-the-art baseline algorithms, U-apriori (HOOSHSADAT et al. 2012) and EFIM (Zida et al. 2015) in terms of the hyper-volume, convergence, promising patterns and scalability. MOEA-HEUPM generates patterns at least 5 to 10 times smaller than that of U-apriori and EFIM. The value encoding is converged with the low number of generations, whereas the binary encoding performs better when more generations are achieved especially in the sparse dataset. Therefore, binary encoding is more suitable for the dataset that has a large number of itemsets. On the other hand, when the number of several itemsets is lower or dataset is a dense type, then value encoding schema is considered to be used. It is observed that the proposed algorithm generates less but useful non-dominated HEUPs for decision-makers without the need to specify the threshold value in the uncertain environment.

BGWO: In the literature, meta-heuristic optimization approaches are used to solve the various engineering problems because of the following: (1) They are simple and easy to implement. (2) There is no requirement for gradient information. (3) They avoid local optima. (4) They can be used in diverse disciplines. Grey wolf optimization (GWO) approach (Mirjalili et al. 2014) is proposed to solve the combinatorial optimization problems. This bio-inspired algorithm replicated the leadership foraging strategy by simulation of the behavior of the grey wolf. Grey wolf is considered as the top-level predators. They usually live in a group size of 5–12 wolves. However, it only deals with continuous optimization problems.

In 2020, Pazhaniraja and Kumar (2020) proposed a Boolean-based modified grey wolf optimization (BGWO) algorithm based on the traditional GWO (Mirjalili et al. 2014), to mine HUIs from large datasets. The grey wolf optimizer approach (Wang and Li 2019) is easy to implement and search the solution space with less number of parameters and high accuracy. The proposed algorithm is inspired by the optimization model which is based on the biotic behavior of the grey wolf. BGWO algorithm uses five Boolean operators, namely binary adder, difference, de Morgan's AND, multiplexer (with uncertainty) and circular-shift, to handle the binary solution representation. The Boolean operators are used for the conversion of continuous GWO into modified GWO where each operator in GWO is redefined with unconventional Boolean operators. The flow of GWO is preserved to retain the originality of the searching algorithm, whereas the operators are redefined to handle Boolean values.

Unlike standard GWO, the modified GWO does not possess any boundary check procedure which ensures the entire search is made within the boundary region. The fitness function is used as the total utilization of an itemset in every transaction in the dataset. An item pruning strategy is employed to enhance the proficiency of mining process in

terms of run-time prior to the initialization of the population. It saves lots of time to compute the itemset combination to identify HUIs from the given transactions.

The experimental results show that the proposed algorithm performs well compared to the existing state-of-the-art algorithms, HUPE_{umu}-GRAM (Kannimuthu and Premalatha 2014), HUIM-BPSO (Lin et al. 2017), IHUP (Ahmed et al. 2009), UP-Growth (Tseng et al. 2013), Bio-HUIF-GA (Song and Huang 2018), Bio-HUIF-PSO (Song and Huang 2018) and Bio-HUIF-BA (Song and Huang 2018), in terms of the run-time, converging speed and promising HUIs. It is observed that the proposed BGWO reduces overall time complexity in a significant manner.

TNR-HUARM: In the literature, evolutionary computation (Kannimuthu and Premalatha 2014; Lin et al. 2016, 2017; Song and Huang 2018) techniques are never designed to mine top non-redundant high-utility association rule mining.

In 2020, Krishna and Ravi (2020) proposed an evolutionary computation-based top non-redundant high-utility association rule mining (TNR-HUARM) algorithm with binary differential evolution (BDE) and adaptive binary differential evolution (ABDE), to mine the top non-redundant high-utility association rules without using the downward closure property. This work is the first of its kind where the proposed TNR-HUARM algorithm uses and compares BDE and ABDE. Differential evolution (DE) has four critical segments, namely initialization, mutation, recombination and selection for the continuous optimization problems (Storn and Price 1997). DE (Das and Suganthan 2011; Das et al. 2016) is chosen because of the following reasons: (1) It can work on nonlinear, non-differentiable and multi-modular search spaces. (2) It is robust and fast. (3) It can also be utilized for discrete problem spaces. The adaptive differential evolution (ADE) changes DE with parameters through roulette wheel selection (Bäck 1996) and random localization (Kaelo and Ali 2006). The BDE uses the sigmoid function by passing the continuous values to DE (Engelbrecht and Pampara 2007), while ABDE uses the binary version of continuous ADE. TNR-HUARM algorithm consists the following five steps: (1) initialization; (2) mutation; (3) recombination; (4) discretization; (5) selection. The proposed algorithm uses the Michigan approach for the rule coding.

TNR-HUAUM has various applications for analytical customer relation management (CRM), namely Bayesian networks (Tian et al. 2013; Xiao et al. 2016), recommender systems (Park et al. 2012; Bobadilla et al. 2013; Najafabadi et al. 2019), binary vectors (Li et al. 2019) and neural binary representation learning (Zhang et al. 2019). The proposed algorithm is applied to the online Retail dataset from the UCI repository (Chen and Guo 2012). The *t*-test is performed on the benchmark datasets where the null supposition

is accepted. The experimental results prove the effectiveness of the proposed algorithm in terms of memory usage, run-time and convergence on the benchmark datasets. The proposed TNR-HUAUM does not perform an exhaustive search of the search space to mine top non-redundant high-utility association rules. Moreover, it does not depend on another algorithms to generate top non-redundant high-utility association rules. It generates high-utility association rules first without depending on any closed itemset miner and then after generating the high-utility association rules removes the redundant rules.

HUIM-DEO: In 2020, Pazhaniraja and Sountharajan (2021) proposed a meta-heuristic evolutionary computation algorithm named high-utility itemset mining dolphin echolocation optimization (HUIM-DEO), based on genetic algorithm and dolphin echolocation optimization (DEO) to discover HUIs. It is inspired by the echolocation in dolphins to search the optimal path and direct toward the food. This technique further simulates the Dolphin echolocation (DE) by restraining its exploration, which is proportional to the distance from the target (Soto et al. 2016). DEO provides global search solutions to find HUIs.

The proposed algorithm consists of the following two phases: (1) During the first phase, it finds all the possible search space. (2) During the second phase, it examines better solutions. The proposed HUIM-DEO finds 1-HTWUIs to be the dolphin echo search size, based on TWU model (Liu et al. 2005). It incorporates meta-heuristic features which regulate the values of the randomly created solutions.

The experimental results show that HUIM-DEO algorithm performs well as compared to the state-of-the-art algorithms, HUPE_{umu}-GRAM (Kannimuthu and Premalatha 2014), HUIM-BPSO (Lin et al. 2017), Bio-HUIF-GA (Song and Huang 2018) and Bio-HUIF-PSO (Song and Huang 2018), in terms of the execution time, promising HUIs and convergence. It is observed that the proposed HUIM-DEO makes use of a global search solution and can discover the HUI with the corresponding parameters in terms of profit and quantity.

3.1.6 Discussion of evolutionary computation-based approaches

In the previous section, evolutionary computation including GA, PSO, ACO, multi-objective optimization and hybrid approaches-based HUIs mining approaches is discussed.

GA-based algorithms are based on the principle of survival of the fittest that generate the next population by various operations with each individual represents a possible solution. The meta-heuristic GA-based techniques are used to obtain the optimal solutions to solve the difficult optimization problems by using genetic operations, for example, crossover, mutation and selection. These genetic operations are then per-

formed to generate the next generations. They used the fitness function to efficiently mitigate the three side effects, hiding failure, missing costs and artificial costs. However, GA-based approaches suffer from the combinatorial problem of assigning the initial chromosomes in the later evolution process. Moreover, it is a non-trivial task to obtain the appropriate parameters for the GAs.

To address the above discussed issues, the EC-based PSO algorithms are proposed to efficiently mine HUIs. They are based on bio-inspired and population-based approaches to find the optimal solutions by adopting the velocity to update the particles. The PSO-based approach can be easily implemented without the trivial task to set the appropriate crossover and mutation rates as the GA-based algorithm does. A hierarchical classification MOPSO framework is used for data sanitization to hide the sensitive itemsets in PPDM. However, it cannot be directly applied to solve the problem of PPDM because the dominance relationships should be utilized to obtain the optimized transactions for deletion. Therefore, MOPSO framework using grid-based algorithm is presented to obtain the un-dominated solutions in PPDM by adopting the pre-large concept and updating the *gbest* and *pbest* values in the evolution process. It effectively mitigates the three side effects, hiding failure, missing costs and artificial costs. The heuristic method based on HUIM-AF approach explores very large space to efficiently find the optimal diverse solutions within limited amount of time. PSO-based approaches provide better results as compare to GA-based methods in terms of execution time, number of discovered HUIs and convergence. However, some useful information may be missing to discover the complete set of HUIs in the dataset.

To address the above limitations, EC-based ant colony optimization techniques are proposed, which not only efficiently mine HUIs, but also provide a checking mechanism to ensure the complete set of HUIs in the dataset. However, they require to set the heuristic function in a proper way which is quite challenging task.

The previous studies focus on the efficiency of mining frequent itemsets or high-utility itemsets. They do not consider both measures, support and utility, separately. Therefore, it significantly affects the performance of the mining process. On the other hand, multi-objective EC-based methods consider both measures, support and utility simultaneously to mine the frequent itemsets and high-utility itemsets without specifying the minimum support threshold and minimum utility threshold from a multi-objective view. This task can be further extended by considering the closed itemsets in the dataset. However, it is computationally costly to obtain the optimized solutions from large datasets. To address this issue, an MOEA-based approach is proposed to mine HUIs from large-scale sparse MOPs. A general multi-objective HQPM

is proposed to balance support, occupancy and utility simultaneously.

In order to improve the optimal diverse solutions, HUIM framework, from the perspective of bio-inspired algorithms, is proposed to discover HUIs. It targets the values of the next population rather than maintaining the current optimal values in the next population. Another HUIM framework, based on ABC algorithm, is proposed to mine HUIs by using nectar sources and bitmap structure. A multi-objective EC-based framework is proposed to mine HEUPs by considering both utility and uncertainty in a limited amount of time from an uncertain environment. An optimization model based on the biological behavior of grey wolf is proposed to mine HUIs by using less parameters and high accuracy. A multi-objective EC algorithm, based on BDE and ABDE, to mine high-utility association rules, is proposed to mine HUIs from the online retail sector.

The overview of EC-based soft computing techniques is shown in Table 7 which includes the following fields: (1) name of algorithm; (2) data structure; (3) dataset; (4) mining; (5) pruning strategies; (6) the state-of-the-art algorithms; (7) base algorithms. The paper has encompassed the comprehensive list of research papers in the domain of soft computing including GA-based, PSO-based, ACO-based, MOE-based, bio-inspired, ABC-based, Dolphin echolocation-based and others. The data structures are used to store the useful information in concise form to mine the high-utility itemsets. Most of the research papers have considered transactional datasets. The paper has discussed several pruning strategies to efficiently prune the search space, thereby resulting in the enhancement in the mining process. The survey has shown an in-depth comparison of each algorithm with their state-of-the-art algorithms and base algorithms.

A brief summary of EC-based soft computing techniques along with their pros and cons is shown in Table 8. The article has provided the theoretical basis of each algorithm that discuss the core concept used in the algorithm. The pros and cons of each algorithm are discussed in depth. This will provide new ideas to the researchers on how they can build novel frameworks, pruning strategies and new developments pertaining to similar studies.

3.2 Fuzzy logic-based approaches

Crisp set theory is used to determine whether an element is a member of a set or not and, thereby, has a limited scope to map a quantitative value in the dataset. On the other hand, fuzzy set theory is taken into consideration whether an element is a member of a set with a definite level such as very modest, modest, average, extreme and very extreme (Zadeh 1975) and, thereby, provides an easy interpretation of the data. It transforms the numerical value of the transactions into the representation of the semantic terms in the dataset.

The fuzzy set theory is a major research area of data mining (Kuok et al. 1998). Many fuzzy data mining algorithms based on apriori are available in the literature (Alcalá-Fdez et al. 2009; Chen et al. 2009; Chan and Au 1997; Hong et al. 2003; Lee et al. 2008). However, they generate a large number of candidates. To overcome this issue, various HUIM algorithms are proposed to discover HUIs (Ahmed et al. 2009; Chan et al. 2003; Hu and Mojsilovic 2007; Yeh et al. 2008). However, these algorithms only provide the utility information to make business decisions. Wang et al. (2009) first combine the high-utility mining with the fuzzy set theory to discover the high fuzzy utility itemsets from the dataset. However, fuzzy utility mining considers only pre-defined fuzzy membership functions.

To address this issue, temporal-based fuzzy utility mining algorithms are proposed. They adopt the concept of temporal property of the itemsets to find the different time intervals of the items in the transactions in the dataset (Huang et al. 2017; Hong et al. 2019). However, they consume a high amount of time to process an itemset. Moreover, the temporal fuzzy utility mining approaches consider the single-conditional measurement which does not precisely reflect the itemsets in the real-time applications (Wang et al. 2009; Lan et al. 2015; Huang et al. 2017). To address this issue, several fuzzy utility mining algorithms are proposed which consider the concept of multi-conditional measurement (Chen et al. 2007; Ouyang and Huang 2011) where each item is assigned with an individual threshold. However, it is a more challenging task to consider the multiple item thresholds.

Several genetic fuzzy mining (GFM) algorithms (Alcalá-Fdez et al. 2009; Chen et al. 2012; Hong et al. 2006; Kaya and Alhajj 2005, 2006) are proposed that combine the genetic concept and fuzzy utility mining to discover the fuzzy association rules from the datasets. However, these algorithms consider only the single-objective optimization which does not reflect the true characteristics of the real-time applications. To address this issue, several multi-objective optimization GFM algorithms (Alhajj and Kaya 2007; Chen et al. 2013; Kaya 2005) are proposed with multiple objective functions. However, it is a challenging task to process the items to find the appropriate optimal solutions.

3.2.1 Simple fuzzy logic-based approaches

HUFI-Miner: The existing utility mining algorithms (Erwin et al. 2007; Liu et al. 2005; Yao and Hamilton 2006, [9]) cannot deal with both integer and real quantities. In 2009, Wang et al. (2009) proposed HUFI-Miner (High utility fuzzy itemsets miner) algorithm, first of its kind, to find the high-utility fuzzy itemsets (HUFIs) over numeric and nominal attributes from the transaction dataset. The authors proposed certain constraints and accompanying algorithms for efficiently pruning the search space in high-utility fuzzy itemset

Table 7 Comparison of the existing evolutionary computation-based high-utility itemsets mining approaches

Algorithm	Data structure	Dataset type	Mining	Pruning strategy	The state-of-the-art algorithms	Base algorithms
<i>GA-based techniques</i>						
GA+ pre-large (Lin et al. 2013)		Transactional	A set of appropriate transactions HT to be delete	TWU, pre-large concepts	None (first of its kind)	None
PPUMGA (Lin et al. 2014)		Transactional	TID	TWU, Pre-large concepts	None (first of its kind)	None
HUPE _{UMU} -GARM (Kannimuthu and Premalatha 2014), HUPE _{WUMU} -GARM (Kannimuthu and Premalatha 2014)	Utility-list	Transactional	HUI	TWU-based	None (first of its kind)	None
sGA2DT (Lin et al. 2015), pGA2DT (Lin et al. 2015)		Transactional	Sanitized dataset	Fitness function, Pre-large concept	Traditional greedy approach	None
PPUMGAT+ (Lin et al. 2016)	MP-tree	Transactional	HUIs	TWU, Pre-large concepts	HUPE _{UMU} -GARM (Kannimuthu and Premalatha 2014)	Traditional GA-based algorithm
HUIM-JGA (Zhang et al. 2019)	Bitmap	Transactional	HUIs	TWU-based	Bio-HUIF-GA (Song and Huang 2018), Bio-HUIF-PSO (Song and Huang 2018), HUIM-BPSO (Song and Huang 2018), HUIM-BPSO _{sig} (Lin et al. 2016), HUPE _{UMU} -GARM (Kannimuthu and Premalatha 2014)	UP-Growth (Tseng et al. 2010), UP-Hist Growth (Dawar and Goyal 2015), IHUP (Ahmed et al. 2009)
GAHUIM (Javangula 2021)	Utility-list	Transactional	HUIs	Fitness function	IHUP (Ahmed et al. 2009), HUI-Miner (Qu et al. 2019), UP-Growth (Tseng et al. 2010), UP-Growth ⁺ (Tseng et al. 2013)	None
<i>PSO-based techniques</i>						
BPSO-Proposed+ (Lin et al. 2015)		Transactional	HUIs	TWU-based	HUPE _{UMU} -GARM (Kannimuthu and Premalatha 2014), FHM (Fournier-Viger et al. 2014)	None
Proposed algorithm (BPSO) [100]	MP-tree	Transactional	HUIs	TWU-based	HUPE _{UMU} -GARM (Kannimuthu and Premalatha 2014)	None
HUIM-BPSO ⁺ (Lin et al. 2017)		Transactional	HUIs	TWU-based	HUPE _{UMU} -GARM (Kannimuthu and Premalatha 2014)	None
HUIM-BPSO _{sig} (Lin et al. 2016)		Transactional	HUIs	TWU-based	HUPE _{UMU} -GARM (Kannimuthu and Premalatha 2014)	Two-Phase (Liu et al. 2005)

Table 7 continued

Algorithm	Data structure	Dataset type	Mining	Pruning strategy	The state-of-the-art algorithms	Base algorithms
MOPSO-based data sanitization model (Verykios et al. 2004)	Hierarchical cluster	Transactional	Non-dominated solutions	LUS, GUS	cpGA2DT (Lin et al. 2014), NSGA2DT [106]	None
GMPSO [108]		Transactional	Non-dominated solutions	LUS, GUS	cpGA2DT (Lin et al. 2014), NSGA2DT [106]	None
HUIM-AF (Song et al. 2021)	Binary array	Transactional	HUIs	TWU-based	HUPE _{U-MU} -GARM (Kannimuthu and Premalatha 2014) and HUIM-BPSO _{sig} (Lin et al. 2016)	None
<i>ACO-based techniques</i>						
HUIM-ACS (Wu et al. 2016)	Hash table, routing graph	Transactional	HUIs	TWU-based, positive pruning rule, Recursive pruning rule	HUPE _{U-MU} -GARM (Kannimuthu and Premalatha 2014), HUIM-BPSO (Lin et al. 2017)	HUI-Miner (Liu and Qu 2012), EFIM (Zida et al. 2015)
ACHUIM (Arunkumar et al. 2020)		Transactional	HUIs	TWU	HURI (Pillai et al. 2013)	None
<i>Multi-objective evolutionary computation-based techniques</i>						
MOEA-FHUI (Zhang et al. 2018)		Transactional	Frequent and high-utility itemsets	Tchebycheff value	None (first of its kind)	FP-Growth (Han et al. 2000), HUI-Miner (Liu and Qu 2012), HUIM-ACS (Wu et al. 2017), TKU-Miner (Zhang et al. 2016)
CP-MOEA (Cao et al. 2019)	Bitmap	Transactional	Frequent and high-utility itemsets	USC, USA	FHUI-MOEA (Zhang et al. 2018)	None
SparseEA (Tian et al. 2020)	Bitmap	Transactional	Generated optimal solutions	Score assignment strategy, Genetic operator	None (first of its kind)	None
MOEA-PM (Fang et al. 2020)	Bitmap	Transactional	High-quality patterns	Support, TWU	SparseEA (Tian et al. 2020), MOPM (Zhang et al. 2017), MOEA-FHUI (Zhang et al. 2018)	None
<i>Hybrid techniques</i>						
Three phase mining (Lin et al. 2015)	TIU table	Transactional	HUIs	TWDC	None	None
Bio-HUIF (Song and Huang 2018)	Bitmap	Transactional	HUIs	TWU-based, PVCE	HUPE _{U-MU} -GARM (Kannimuthu and Premalatha 2014), HUIM-BPSO (Lin et al. 2017)	IHUP (Ahmed et al. 2009), UP-Growth (Tseng et al. 2013)

Table 7 continued

Algorithm	Data structure	Dataset type	Mining	Pruning strategy	The state-of-the-art algorithms	Base algorithms
HUIM-ABC (Song and Huang 2018)	Bitmap	Transactional	HUIs	HTWUI, DNSG	HUPE _{U/MU} -GARM (Kannimuthu and Premalatha 2014), HUIM-BPSO _{sig} (Lin et al. 2016), HUIM-BPSO (Lin et al. 2017)	None
MOEA-HEUPM (Ahmed et al. 2021)		uncertain dataset	HEUPs	Tchebycheff value	None	U-apriori (HOOSHSADAT et al. 2012), EFIM (Zida et al. 2015)
BGWO (Pazhaniraja and Kumar 2020)	Bitmap	Transactional	HUIs	Item pruning, Fitness function	HUPE _{U/MU} -GARM (Kannimuthu and Premalatha 2014), HUIM-BPSO (Lin et al. 2017), Bio-HUIF (Song and Huang 2018)	IHUP (Ahmed et al. 2009), UP-Growth (Tseng et al. 2013)
TNR-HUARM (Krishna and Ravi 2020)	Memory-list	Transactional	Top non-redundant high-utility association rules	Fitness function	None	None
HUIM-DEO (Pazhaniraja and Sountharajan 2021)		Transactional	HUIs	TWU	HUPE _{U/MU} -GARM (Kannimuthu and Premalatha 2014), Bio-HUIF-GA (Song and Huang 2018), Bio-HUIF-PSO (Song and Huang 2018)	None

Table 8 The pros and cons of the evolutionary computation-based HUIs mining approaches

Algorithm	Theoretical basis	Pros and cons
<i>GA-based techniques</i>		
GA+ pre-large Lin et al. (2013)	The proposed GA-based privacy-preserving utility mining method deletes appropriate transactions for hiding sensitive HUIs from the datasets	GA+ Pre-large adopts downward closure property and pre-large concepts to decrease the dataset re-scan cost. However, pre-large concept could be further applied in case of transaction insertion and modification in the dataset
PPUMGA Lin et al. (2014)	A GA-based privacy preserving utility mining algorithm is proposed by adopting pre-large concepts to discover the appropriate transactions to be inserted to hide the sensitive HUIs in the dataset	The pre-large concept significantly reduces the re-scan cost, thereby, resulting in the enhancement of mining process. However, the proposed algorithm only uses transaction insertion in the dataset
HUPE _{UMU} -GARM and HUPE _{WUMU} -GARM Kannimuthu and Premalatha (2014)	Two GA-based HUIs mining algorithms with ranked mutation are proposed with and without minimum utility threshold	TWU is used for removing unpromising items at an earlier stage and the measure for fitness value computation. However, it is time-consuming to discover the complete or large portion of HUIs. Moreover, they cannot estimate the duplicate itemsets in the dataset. Furthermore, the proposed algorithms suffer from the combinational problem of the generated itemsets and require several parameters in the evolution process
sGA2DT and pGA2DT Lin et al. (2015)	A heuristic GA-based approach with two optimization techniques is proposed to sanitize data with transaction deletion in PPDM	The proposed algorithms perform well than the greedy approach. However, the greedy approach shows better execution time because it does not take into account the three side effects to hide the sensitive itemsets. The proposed algorithms are efficient at finding the optimal transactions for deletion, but they still require pre-defined weights of the side effects, thereby, resulting in the incomplete information in the mining process
PPUMGAT+ Lin et al. (2016)	PPUMGAT+ is GA-based privacy preserving HUIs mining algorithm and hides the sensitive HUIs in PPUM	The pre-large concept significantly reduces the dataset scan cost and speeds up the mining process. PPUMGAT+ enables the effectiveness of privacy preserving for secure information in real-time world
HUIM-IGA Zhang et al. (2019)	An HUIM approach based on the improved genetic algorithm is proposed to efficiently mine HUIs	HUIM-IGA consumes less time to find the same proportions of HUIs than that of other EC-based HUIM algorithms. However, more robust searching strategies are required to avoid the generation of non-HUIs and reduce the unnecessary use of computational resources. The proposed algorithm could be further improved to enhance the overall ability to find HUIs
GAHUIM Javangula (2021)	A non-binary GA-based algorithm is proposed to mine HUIs	The proposed algorithm mines HUIs by scanning the dataset only once. It converges from initial population of itemsets to high-utility itemsets

Table 8 continued

Algorithm	Theoretical basis	Pros and cons
<i>PSO-based techniques</i>		
BPSO-Proposed+ Lin et al. (2015)	A binary PSO-based algorithm is proposed to efficiently mine HUIs. It uses extended MP-tree structure to avoid the invalid generations of the particles	It produces the valid combinations of itemsets in the dataset. The <i>pbest</i> , <i>gbest</i> and <i>sigmoid</i> function in the evolution process, efficiently discover HUIs. However, it incurs the problem of combinatorial explosion
Proposed algorithm (BPSO) Lin et al. [100]	An evolutionary computation binary PSO-based approach, based on the particle swarm optimization and adopts a <i>sigmoid</i> function, to efficiently mine HUIs	The proposed algorithm performs well when the minimum utility threshold is set high in the dense datasets. It efficiently finds HUIs by using the local and global results in the mining process. However, it suffers from the combinatorial explosion. Moreover, the minimum utility depends on the dataset
HUIM-BPSO+ Lin et al. (2017)	The BPSO-based model using an improved OR/NOR-tree structure, is proposed to efficiently discover HUIs by integrating the <i>sigmoid</i> updating strategy and TWU model	HUIM-BPSO algorithm provides accurate precision to obtain the same number of HUIs as the conventional TWU model when the number of iterations increases in the mining process. However, it takes a high amount of time to discover the complete set of HUIs. Moreover, it fails to estimate the duplicate itemsets present in the dataset
HUIM-BPSO _{sig} Lin et al. (2016)	A binary PSO-based technique is used for the evolutionary computation, to efficiently find HUIs by integrating the <i>sigmoid</i> updating strategy and TWU model	HUIM-BPSO _{sig} achieves the speed of convergence and accuracy for generating the same number of HUIs compared to the HUPE _{um} -GRAM algorithm from the condensed datasets. However, it consumes lots of time to find the complete set of HUIs
MOPSO-based data sanitization model Verykios et al. (2004)	A sanitized algorithm, based on the MOPSO and hierarchical clustering method, is proposed to discover the optimized solutions for PPDM	The proposed algorithm effectively eliminates the four side effects, hiding failure, missing cost, artificial cost and data dissimilarity. However, the multi-optimization framework could be further explored in terms of PPUM
GMPSO Wu et al. [108]	A grid-based multi-objective particle swarm optimization algorithm is proposed to obtain the optimum solutions of sanitized dataset in PPDM	The proposed algorithm does not require crossover and mutation operations to generate the next populations. Therefore, it effectively eliminates the four side effects and time cost to obtain the non-dominated solutions. However, it still obtains the minor side effect of dataset dissimilarity as compared to benchmark algorithms (Lin et al. 2014, [106])
HUIM-AF Song et al. (2021)	A heuristic artificial fish swarm algorithm is proposed to efficiently find HUIs	The proposed algorithm significantly discovers the large set of diverse solutions within limited amount of time in HUIM problem. However, more effective pruning strategies could be further designed to obtain the better optimization results

Table 8 continued

Algorithm	Theoretical basis	Pros and cons
<i>ACO-based techniques</i>		
HUIM-ACS Wu et al. (2016)	The proposed HUIM-ACS algorithm, based on ant system framework (Colomi et al. 1991; Dorigo et al. 1996), finds HUIs by a specific routing graph and TWU model	It proposed two pruning rules, hash table based rule and recursive pruning rule, to improve the performance of mining process. It performs well in case of the very large datasets. However, it obtains less HUIs in few datasets. Moreover, it consumes lots of time to estimate each candidate in the selection process. Therefore, there is need to design more effective heuristic function and pruning rules to save the time in the evaluation process
ACHUIM Arunkumar et al. (2020)	A customized ant colony algorithm is proposed to discover HUIs in an effective way	The proposed algorithm obtains the optimal solutions to generate more HUIs in less time. However, it can be further improved and applied to increase the efficiency of the proposed system
<i>Multi-objective evolutionary computation-based techniques</i>		
MOEA-FHUI Zhang et al. (2018)	A multi-objective evolutionary algorithm, based on the MOEA/D method (Zhang et al. 2017), is proposed to discover the frequent and high-utility itemsets by considering the both measures, support and utility, simultaneously	The proposed algorithm does not require to specify the minimum support threshold and minimum utility support, thereby, generates a set of non-dominated solutions for recommendations and brings more convenience for the users. However, the run-time and memory consumption of the proposed algorithm increase as the number of transactions and/or items increase in the dataset. Moreover, it consumes lots of computational time in the evaluation process to calculate the support and utility of itemsets
CP-MOEA Cao et al. (2019)	A closed itemset property-based multi-objective evolutionary approach, based on similar framework as of NSGA-II (Deb et al. 2002), is proposed to discover the frequent and high-utility itemsets by considering both measures, support and utility, together	The proposed updating strategies, USC and USA, not only accelerating the population convergence, but also improve the population diversity
SparseEA Tian et al. (2020)	An evolutionary algorithm, based on similar framework like NSGA-II (Deb et al. 2002), is proposed for exhaustive sparsity multi-objective optimization problems to ensure the sparsity of the generated solutions	The population strategy and genetic operator guarantee the sparsity of the generated optimal solutions. However, the proposed algorithm could be integrated with effective customized search strategies to solve specific sparse MOPs in applications
MOEA-PM Fang et al. (2020)	An improved multi-objective evolutionary algorithm is proposed to mine the high-quality patterns using three measures, support, occupancy and utility	The proposed algorithm is effective in terms of the quality of results, efficiency and converging speed. However, more effective strategies could be designed for MOEAs to discover high-quality patterns

Table 8 continued

Algorithm	Theoretical basis	Pros and cons
<i>Hybrid techniques</i>		
Three phase mining Lin et al. (2015)	The proposed three-phase algorithm efficiently mines HUIs by considering the discount strategies	The initial two phases of proposed algorithm are similar to two-phase algorithm, but it obtains the whole HUIs with discount strategies in the third phase. However, more effective discount strategies could be designed and applied in other domains
Bio-HUIF Song and Huang (2018)	Three novel bio-inspired algorithms Bio-HUIF-GA, Bio-HUIF-PSO and Bio-HUIF-BA, based on GA, PSO and BA, respectively, are proposed to discover all HUIs	The three strategies, bitmap dataset representation, promising encoding vector checking, and bit difference sets, are used to speed up the mining process. The proposed algorithms improve the diversity of solution by performing limited number of iterations. However, Bio-HUIF-GA spends lots of time to mine the complete set of HUIs
HUIM-ABC Song and Huang (2018)	The proposed algorithm, based on the artificial bee colony (ABC) [166], efficiently discovers HUIs by using the bitmap structure and nectar sources	The bitmap structure and DNSG strategy efficiently prune the search space and reduce the computational cost, thereby, discovering more HUIs in limited number of iteration cycles
MOEA-HEUPM Ahmed et al. (2021)	A multi-objective evolutionary framework is proposed to efficiently discover the high expected utility patterns by considering both factors, utility and uncertainty, together in a limited time from the uncertain dataset	The proposed framework is effective and efficient with respect to the hyper-volume, converging speed and number of obtained patterns. Moreover, it does not require the prior knowledge of minimum utility threshold and minimum uncertain threshold to discover the information in the dataset
BGWO Pazhaniraja and Kumar (2020)	A meta-heuristic optimization model-based modified grey wolf optimization algorithm which uses five Boolean operators (De Morgan's AND, adder, difference, multiplexer, circular shift), is proposed to discover HUIs	The BGWO algorithm searches the solution space with high precision and less parameters. Moreover, it does not possess any boundary check procedure which ensures the entire search is made within the boundary region
TNR-HUARM Krishna and Ravi (2020)	The authors proposed BDE- and ABDE-based top non-redundant high-utility association rules mining algorithm to find the top high-utility association rules	The proposed algorithm is used in several applications, for example, customer segmentation, recommender systems, market basket analysis, retail shelf allocation and healthcare. It is observed that BDE-based approach performs well than ABDE. But when the <i>t</i> -test is performed, both approaches are found statistically equal on all the benchmark datasets. However, BDE-based approach consumes lots of time to tune the control parameters, but ABDE does not require to tune control parameters manually
HUIM-DEO Pazhaniraja and Sountharajan (2021)	A meta-heuristic algorithm is proposed to mine HUIs using dolphin echolocation optimization and genetic algorithm	The greedy search method obtains the optimal solutions to find the more HUIs in less time. The reason is that it significantly reduces the combinatorial explosion in the evaluation process

mining problems. Moreover, the theoretical and experimental of these intelligent heuristics can effectively be applied to discover HUFU. The concept of quantitation is easy to interpret because of the fuzzification concept into linguistic regions. Hence, it finds less redundant quantitative itemsets from the dataset. It not only simplifies the mining problem but also provides an accessible amount of itemsets. HUFU-Miner incorporates the downward closure property of HMFI (high transaction-max utility fuzzy itemsets) and small look-up tables to solve the memory requirements in the case of large datasets. Moreover, it needs at most two scans of the original dataset. It significantly prunes HUFU candidates, thus resulting in the reduction of execution time cost considerably.

The experimental results show that the proposed HUFU-Miner algorithm significantly reduces the search space and HUFU candidates, which results in the reduction of the execution cost. Moreover, almost half of the cost is reduced in the second scan. It is observed that the proposed algorithm not only simplifies the mining problem but also produces more profitable patterns.

GDFU: However, TPFU (two-phase fuzzy utility mining) algorithm (Lan et al. 2015) efficiently avoids information loss in fuzzy utility mining by using FUUB (fuzzy utility upper-bound) model. But, there are lots of generated unpromising candidates because there is no effective strategy to reduce the upper-bound of fuzzy utilities of itemsets in the mining process.

In 2014, Lan et al. (2014) designed an approach named GDFU (gradual data reduction fuzzy utility mining) to mine high fuzzy utility itemsets from the quantitative transactional dataset. The profits, quantities and fuzzy utility of each item is considered in a transaction. The authors proposed an efficient data reduction strategy to improve the performance and prune the unpromising candidate itemsets. The fundamental of this proposed method is that the unpromising fuzzy itemsets in the transactions at the k -th pass are early eliminated by using the members emerged in the set of candidate fuzzy utility upper-bound k -itemsets before discovering high fuzzy utility upper-bound $(k + 1)$ -itemsets. This strategy accelerates the execution speed in terms of matching and counting the fuzzy utilities of itemsets in the mining process.

The experimental results show that the proposed GDFU approach significantly works well as compared to TPFU (Lan et al. 2015) in terms of the execution time. The reason is that the proposed algorithm incorporates the effective strategy to further prune a large number of unpromising candidates and thereby significantly improves the mining performance.

TPFU: As we have seen that fuzzy utility mining (Wang et al. 2009) integrates the concept of fuzzy set concept with utility mining to mine high fuzzy utility itemsets from quantitative datasets. However, the common membership degree value of fuzzy itemsets is not considered in the fuzzy util-

ity function. Moreover, the fuzzy utility values of items are difficult to calculate and understand.

In 2015, Lan et al. (2015) proposed an algorithm named TPFU (two-phase fuzzy utility mining) to discover the actual high fuzzy utility itemsets (HFUs) from the quantitative transactional dataset. Since the downward closure property cannot hold in the fuzzy utility mining, a novel fuzzy utility measure is designed to measure the fuzzy utility itemsets using the fuzzy minimum operator. The fuzzy utility upper-bound (FUUB) method is proposed to avoid information loss in fuzzy utility mining. TPFU considers not only the quantities and profit values of items in transactions but also the common degrees of fuzzy itemsets by using the concept of fuzzy minimum operator. TPFU consists of the following two phases: (1) During the first phase, the FUUB of all distinct fuzzy itemsets is calculated and 1-HFUUB is obtained from their FUUB value in a level-wise way. (2) During the second phase, the actual fuzzy utility values of all itemsets in the set of all 1-HFUUB are obtained by scanning the dataset. Finally, the complete set of HFUs are obtained.

The experimental results show that the proposed algorithm performs well in terms of execution time under various parameter settings on the benchmark datasets. The reason is that TPFU effectively reduces the search space to find fuzzy utility itemsets in comparison with the full search space of itemsets. It is observed that the number of HFUs is less than that of high-utility itemsets (HFs) under various parameters. The reason is that TPFU uses the concept of both original utility function and minimum operator.

FFI-Miner: In 2015, Lin et al. (2015) proposed a fuzzy frequent itemset miner (FFI-Miner) algorithm to discover the fuzzy frequent itemsets (FFIs) without candidates generation. A fuzzy-list structure is used to maintain the required fuzzy data for the mining process. It efficiently speeds up the computation process to directly find the discovered FFIs. It consists of the following three attributes: (1) transaction TID (*tid*); (2) internal fuzzy value (*if*); and (3) resting fuzzy value (*rf*).

Two strategies, namely maximum scalar cardinality ([211], Lin and Hong 2014) and support-ascending order, are adopted to transform the original datasets into fuzzy datasets. The maximum scalar cardinality strategy is used to easily identify the fuzzy values between the transformed fuzzy terms based on the designed fuzzy-list structure. The fuzzy terms are sorted in their support-ascending order to perform the intersection operations for discovering their support values among the fuzzy k -terms ($k \geq 2$). The search space of FFI-Miner is represented as an enumeration tree according to the support-ascending strategy. However, the complete search space of the enumeration tree is very large to find the complete set of FFIs. Therefore, a pruning strategy is proposed to prune the search space, thereby accelerate the mining process to directly obtain FFIs. If the summation of

rf values of an itemset is no less than the minimum support count, then any extensions of an itemset will not be FFIs. Hence, it can be directly ignored to avoid the construction phase of fuzzy-list structures of the extensions of that itemset.

The experimental results show that the proposed FFI-Miner performs well as compared to the existing state-of-the-art, apriori-based FDFA (Hong et al. 1999) and, tree-based algorithms, CFFP-tree [211], GDF (Lan et al. 2015) and UBFFP-tree (Lin and Hong 2014), in terms of the run-time and number of visited nodes for discovering FFIs under variants of membership functions. It is observed that the proposed algorithm is almost up to 1 or 2 orders of magnitude faster than other benchmark algorithms.

MFFI-Miner: Although tree-based methods ([211], Lan et al. 2015; Lin and Hong 2014) perform well as compare to apriori-based FDFA (Hong et al. 1999), to discover the fuzzy frequent itemsets (FFIs). However, they incur high computation cost. In addition, they require multiple dataset scans to mine FFI in a level-wise way. Moreover, they consider the maximum scalar cardinality to discover, at most, one represented item from the transformed linguistic terms, which results in the missing of some useful FFIs. Therefore, the set of discovered FFIs may be incomplete.

In 2016, Lin et al. (2017) proposed multiple fuzzy frequent itemset miner (MFFI-Miner) algorithm to find the multiple fuzzy frequent itemsets (MFFIs) without candidates generation. The fuzzy-list structure is proposed to efficiently retain the required data to decrease the computation of dataset scan of the mining process in a level-wise way. The proposed algorithm consists of two phases: (1) transformation and (2) fuzzy-list construction. During the transformation phase, if the fuzzy itemset with its support count is no less than the minimum support count, then it is considered as the FFI and kept in the transformed datasets. In the fuzzy-list construction phase, the remaining fuzzy itemsets build their own fuzzy-list structure to maintain the desired information. Then, the upper-bound value of processed fuzzy itemsets is obtained to maintain the downward closure property for a later combinational procedure. The search space of the fuzzy-list structure is represented as an enumeration tree according to the designed support-ascending-order strategy using depth-first search. Two pruning strategies, namely pruning by internal value (PI) and pruning by resting value (PR), are designed for pruning the search space of the enumeration tree and thereby accelerate the mining process to directly mine the MFFIs. Thus, if the summation of the internal fuzzy (if) values or resting fuzzy (rf) values of a fuzzy itemset is less than the minimum support count, any extensions of that itemset will not be the FFIs and can be directly ignored to avoid the construction phase for its fuzzy-list structure. Thus, the search space can be reduced in the enumeration tree.

In order to evaluate the effectiveness of the designed pruning strategies, two versions of the proposed algorithm, namely MFFI-Miner_{PI} and MFFI-Miner_{PR}, are developed. It is observed that these pruning strategies significantly reduce the search space to discover MFFIs. MFFI-Miner_{PR} always performs well than MFFI-Miner_{PI} for dense datasets. Experimental results show that the proposed MFFI-Miner algorithm efficiently works well as compared to the state-of-the-art algorithms, generate-and-test GDF (Lan et al. 2015) and pattern-growth UBFFP-tree (Lin and Hong 2014), in terms of the memory usage, run-time, node analysis and a number of determined candidates from the real and synthetic datasets. MFFI-Miner_{PI} and MFFI-Miner_{PR} are almost 2 to 3 orders of magnitude faster than GDF or UBFFP-tree algorithms. Although the proposed MFFI-Miner algorithm required extra memory in order to retain the fuzzy-list structures, the number of candidates could be greatly reduced when the search space was eliminated by using the designed pruning strategies.

HFUBWSM: Several fuzzy utility mining algorithms (Wang et al. 2009; Lan et al. 2014, 2015; Lin et al. 2017) are proposed to efficiently mine the fuzzy utility itemsets from the quantitative transaction datasets. However, in the literature, less work is done on the fuzzy utility-based web-page sets mining from weblog datasets. In 2017, Kumar and Thakur (2018) proposed a high fuzzy utility-based web-page sets mining (HFUBWSM) algorithm to find the sets of the frequent web-page without candidates generation from the weblog transaction dataset. However, it only stores fuzzy-based frequent web page sets. It requires only one dataset scan to significantly reduce the computation time, thereby resulting in the acceleration of the mining process. A new fuzzy set membership function is defined for proper value assignment to each member in the set. A minimum fuzzy utility threshold value (MFUTV) and user-defined percentile (UDP) are applied to reduce the search space of the mining process.

The proposed algorithm consists of the following steps: (1) In the first step, it collects the weblog data sets from web servers to create the weblog transaction dataset (WDB). Then, it defines two parameters, namely maximum fuzzy transaction utility (MFTU) and external utility (EU) in the mining process. (2) In the second step, it creates the web transaction table with internal utility and fuzzification of web transactions into fuzzy regions using the membership function. (3) In the third step, it calculates the value of MFTU for each transaction in WDB and the value of FTWU (Fuzzy transaction weighted utilization) using the IU (Internal utility) and EU in the weblog dataset. (4) In the fourth step, if the value of FTWU is no less than that of MFTU, then it is desired web page set, otherwise not. (5) In the last step, it prunes all the superset of infrequent generators which do

not satisfy the minimum fuzzy threshold value. Finally, it extracts the high fuzzy utility web page sets.

The experimental results show that the proposed HFUBWSM algorithm works well as compared to the state-of-the-art algorithms, IHUP (Ahmed et al. 2009), UP-Growth (Tseng et al. 2013), FHM (Fournier-Viger et al. 2014) and HUI-Miner (Liu and Qu 2012), in terms of the memory usage, run-time and extracted high-utility web page sets from weblog datasets. It is observed that HFUBWSM is approximately 50 percent, 35 percent, 32 percent, and 30 percent more efficient than IHUP, UP-Growth, HUI-Miner and FHM, respectively. These results are highly useful for decision-makers to understand the behavior of customers and do business planning according to the discovered knowledge.

WFUM: Weighted-based frequent itemsets mining algorithms (Lin et al. 2015, 2017) are proposed to discover the weighted frequent itemset by considering the weight for items as a significant parameter. However, the concept of weighted items is still to be explored more to get useful information about items in the dataset.

In 2019, Hong et al. (2019) proposed a two-stage algorithm, named weighted fuzzy utility mining (WFUM), to mine the high weighted fuzzy utility itemsets (HWFUIs) in a level-wise. The proposed work is the extended version of TPFU approach (Lan et al. 2015), to handle the different weights of items. WFUM consists of two phases: (1) In the first phase, the weighted fuzzy utility upper-bound (WFUUB) model is designed to obtain all the high weighted fuzzy utility upper-bound itemsets (HUFUUBI) to satisfy the downward closure property. It is used to avoid information loss in WFUM model. (2) In the second phase, the actual weighted fuzzy utility (*awfu*) value of HUFUUBI is computed. If *awfu* value is no less than the minimum threshold, then the itemset belongs to the HWFUI, otherwise not.

The experimental results show that the proposed WFUM works well than the state-of-the-art TPFU algorithm (Lan et al. 2015) in terms of the number of transactions, mean-length of frequent itemsets and mean-length of items per transaction. It is observed that the number of itemsets, in proposed WFUM, is always less than that of TPFU.

FHFAUIM: The main issue with two-stage utility mining (UM) (Liu et al. 2005) is that it considers the same threshold value to evaluate the itemsets, irrespective of their length. Fuzzy utility mining (FUM) (Lan et al. 2015) integrates the utility mining with the fuzzy concept to identify the high fuzzy utility itemsets. It has the advantage over UM as it efficiently represents quantitative information. However, it has a measurement issue in which the fuzzy utility value for an itemset may be higher than that of its subsets. To address this issue, fuzzy average-utility itemsets using two-stage strategy (Hong et al. 2020) is proposed to mine high fuzzy average-utility itemsets. However, it incurs high computational cost.

In 2021, Hong et al. (2021) proposed a single-stage algorithm, named fast high fuzzy average-utility itemset mining (FHFAUIM), based on FP-tree (Han et al. 2000), to identify fuzzy average-utility itemsets. It integrates a single-stage strategy with a tree structure to prune the search space to store information in the external array list at the node level. The external list stores the transaction ID of the fuzzy item, fuzzy value and utility value is added to the tree node. It directly performs mined information in a single phase. An over-estimation model (Hong et al. 2020) is used for fuzzy average-utility mining.

The experimental results show that the proposed algorithm performs well as compared to high fuzzy average-utility itemset mining (HFAUIM) (Hong et al. 2020) in terms of execution time and the number of generated candidates. The proposed FHFAUIM algorithm has 95.39 percent efficiency improvement rate of execution time over HFAUIM. However, FHFAUIM consumes lots of memory to add list information for each node. The reason is that it speeds up the execution time by using an external list for each node to store mined information which needs extra memory.

HUWAS: High-utility product mining (HUPM) generates the utility of products by considering their duration. In 2021, Ganesan and Shankar (2021) proposed a novel structure, named high-utility product sets using fuzzy (HUFPM), to mine high-utility products from the web datasets by integrating the concept of fuzzy with utility mining. A high-utility web access sequence (HUWAS) tree structure is designed to create the sequential tree by using the internal and external utility of web-access sequence.

The data collection of web service consists of the following phases: (1) pre-processing execution logs; (2) Utility analysis for the product set; (3) predicting maximum utility based on fuzzy logic (FL). In the pre-processing phase, the weblog data are processed to get the statistical information. In the utility analysis phase, the utility of the product set is considered to estimate the high utility in the mining process. HUWAS tree consists of the number of sequences in the dataset. In the last step, the generated pattern is arranged according to the fuzzy membership function into the fuzzy regions. The proposed algorithm performs a single scan to discover the high-utility web access sequence from the candidate sequence. After the construction of the tree, the proposed algorithm finds candidate patterns from the tree. It generates web access patterns to discover frequent sub-sequences from web sequence dataset. The high-utility patterns are obtained in two dataset scans. It prunes the duplicate patterns from the tree to enhance the efficiency of the mining process.

The experimental results show that the proposed algorithm performs well as compared to the state-of-the-art algorithms, ANN and top-k HUI (Duong et al. 2016), in terms of execution time, memory consumption and accuracy from the large web pages from complex datasets. It is observed that the pro-

posed algorithm effectively discovers all HUIs in a dataset with the utilization of optimization methods.

3.2.2 Temporal fuzzy logic-based approaches

TP-TFU: The existing fuzzy utility mining algorithms (Wang et al. 2009; Lan et al. 2014, 2015) are not suitable to mine high temporal fuzzy utility itemsets from temporal datasets. The reason is that these datasets consist of the quantitative data with different time stamps, thereby resulting in high computational complexity.

In 2017, Huang et al. (2017) proposed two-phased temporal-based fuzzy utility mining (TP-TFU) algorithm to find the high temporal fuzzy utility itemsets (HTFUIs) from the temporal quantitative dataset (TQD). The proposed algorithm considers the sale of the items, the amount of the items and transaction periods of transactions in order to find HTFUIs in product datasets. An upper-bound fuzzy utility model is also designed to effectively mine the HTFUIs without loss of any information in temporal fuzzy utility mining. It prunes the unpromising candidates to speed up the mining process. TP-TFU consists of the following two phases: (1) During the first phase, the high temporal fuzzy utility upper-bound (HTFUUB) is derived if their high temporal fuzzy utility upper-bound ratio ($htfuubr$) is no less than the minimum threshold value. (2) During the second phase, the dataset is scanned to obtain the fuzzy utility values of the derived HTFUUB. If the fuzzy utility value of the derived candidate is no less than the user-specific minimum threshold, then it belongs to HTFUIs, otherwise not.

The experimental results show that the proposed TP-TFU algorithm performs well as compared to the two-phased high fuzzy utility (TP-HFU) (Lan et al. 2015) in terms of the total number of different items, the average length of items per transaction, number of transactions and execution time under different threshold settings and fuzzy membership regions. However, the proposed algorithm is required more computational time and slower than the TP-HFU. But, it obtains more useful information to make better decisions. It is observed that the proposed algorithm discovered more useful items than that of TP-HFU.

TFUM with MMU: The existing fuzzy utility mining algorithms (Wang et al. 2009; Huang et al. 2017) are proposed to mine fuzzy utility itemsets by considering only a single minimum utility threshold. However, the single minimum utility threshold does not reflect the true characteristics of real-world applications. Moreover, the threshold requirement should depend on the item. For example, people may consider the utility of emerging items and aged ones in different criteria. An emerging item may not currently get much profit, but it may have a high potential to earn more profit in the future. Therefore, it is not a fair selection to choose a single threshold only. On the other hand, multiple minimum

utility thresholds generally reflect the actual requirements of real-world applications. However, they incur high complexity in the mining process.

In 2018, Huang et al. (2018) proposed the temporal fuzzy utility mining (TFUM) with multiple minimum utility (MMU) threshold to find all the HTFUIs from the quantitative transaction dataset. The minimum utility threshold itself depends on the item. The proposed algorithm consists of two phases: (1) During phase one, the proposed algorithm finds the high temporal fuzzy utility upper-bound itemset based on the minimum constraint (HTFUUBMC) of the promising candidate itemsets. Then, the upper-bound $tfuubr$ values are obtained and compared against the corresponding minimum thresholds. (2) During phase two, the actual fuzzy utility value of each promising candidate itemset is obtained by performing the additional dataset scan. If the actual fuzzy utility values are no less than their corresponding minimum thresholds, then it belongs to HTFUIs, otherwise not. Finally, it finds the complete set of high temporal fuzzy utility itemsets (HTFUIs) satisfying the multiple minimum constraints.

The experimental results show the effectiveness of the proposed algorithm in terms of execution time on the benchmark datasets. It is observed that the multiple constraints of the proposed algorithm give good performance.

TFUM using tree structure: As compared to the traditional fuzzy utility mining algorithms (Huang et al. 2017; Lan et al. 2015), temporal fuzzy utility data mining considers the temporal factor of transactions, purchased quantities, item profits and linguistic terms.

In 2019, Hong et al. (2019) proposed a temporal fuzzy utility mining algorithm to extract the HTFUIs from the temporal quantitative transaction dataset (TQD). A tree structure is designed, based on the frequent pattern tree (FP-tree) (Han et al. 2000), to store the required information in the mining process. The proposed algorithm consists of two phases: (1) In phase one, it discovers the set of high temporal fuzzy utility upper-bound itemsets (HTFUUBI) using FP-Growth approach. The maximum temporal fuzzy utility ($mtfu$) is calculated. The temporal fuzzy utility upper-bound ratio ($tfuubr$) is computed and compared to the user-defined threshold. If $tfuubr$ is no less than the minimum threshold, then the itemsets belong to HTFUIs. (2) During the second phase, the dataset scan is performed to examine whether the candidate itemsets are HTFUIs or not. Finally, it discovers the complete set of HTFUIs.

The experimental results show that the proposed algorithm performs well as compared to the apriori-based mining algorithm, named TP-FPU (Huang et al. 2017) with regard to the run-time and memory consumption on the benchmark datasets.

TPM: The utility mining algorithms using a single-conditional threshold (Tseng et al. 2015; Chu et al. 2009; Wang et al. 2009) are proposed to efficiently mine HUIs from

the datasets. However, they do not reflect the true characteristics of all items. To address this issue, several algorithms using multiple-conditional thresholds (Liu et al. 1999; Chen et al. 2007; Ouyang and Huang 2011; Lin et al. 2016) are proposed to represent the true characteristics of real-world applications. However, these algorithms assign an individual threshold for each item, thereby resulting in the complexity of the mining process.

In 2019, Huang et al. (2019) proposed an algorithm named, two-phase multi-conditional algorithm for temporal fuzzy utility mining (TPM) to find the HTFUIs with multi-conditional minimum thresholds from the quantitative transaction dataset. The proposed algorithm considers the temporal behavior, importance of items and usage of linguistic terms in the mining process. TPM is the generalized version of TFUM (Huang et al. 2017) which considers the single-conditional minimum threshold ($TFUMP_{smt}$). The proposed temporal fuzzy utility mining algorithm considers the multi-conditional minimum threshold ($TFUMP_{mmt}$) to hold the downward closure property and avoids information loss in the mining process. TPM consists of the following two phases: (1) During phase one, the proposed algorithm discovers the HTFUUBI by checking whether their $tfuubr$ values are larger than or equal to their corresponding minimum thresholds. A pre-processing procedure is built to transform the occurrence time of each transaction to a time period and to convert the quantitative information of the dataset to fuzzy regions. (2) During phase two, the temporal fuzzy utility ratio ($tfur$) values of HTFUUBIs are computed and compared against their corresponding minimum thresholds. If $tfur$ values are no less than their corresponding minimum thresholds, then the candidate itemsets belong to HTFUIs, otherwise not.

The experimental results show that the proposed algorithm is highly effective as compared to the traditional TFUM algorithm (Huang et al. 2017), in terms of execution time under different parameter settings. It is observed that the proposed TPM derives fewer itemsets than that of TFUM and performs good performance from both the effectiveness of multi-conditional view-point and the execution efficiency.

3.2.3 Fuzzy logic and GA-based approaches

TMOGFM: In the literature, several genetic fuzzy mining (GFM) algorithms (Hong et al. 2006; Kaya and Alhadj 2005, 2006) are proposed with diverse objective functions, for example, number of large itemsets, number of rules and suitability of membership functions. Although these algorithms provide non-dominated solutions to choose different market strategies for decision-makers. However, there are still lots of challenges how to choose appropriate solutions among different strategies.

In 2013, Chen et al. (2013) proposed a two-stage multi-objective fuzzy mining approach named TMOGFM (two-stage multi-objective genetic-fuzzy mining) to discover the linguistic information from the quantitative transaction dataset. The proposed algorithm not only discovers Pareto solutions but also provides representative solutions for the decision-making process. In the first stage, the proposed algorithm determines non-dominated solutions with the given objective functions. In the second stage, the clustering technique obtains the optimal solutions from the set of the non-dominated solutions. Therefore, each clustered group is a trade-off among objective functions. Finally, the valid solutions of each cluster is used to find the promising itemsets.

The experimental results prove the effectiveness of the proposed algorithm in terms of different generated solutions and clustering results on the benchmark datasets. It is observed that the proposed algorithm provides a simple and useful method to derive non-dominated solutions according to the favorites of decision-makers. These corresponding results could be mined and the obtained outputs could be further examined for business purposes.

HFUI-GA: Traditional high fuzzy utility itemset mining (HFUIM) algorithms (Lan et al. 2015; Wang et al. 2009) mine a set of satisfied high fuzzy utility itemsets (HFUIs) on small datasets, but in real-life applications, the datasets are very large. Thus, it takes a too long time to find the required information, which makes real-time decisions inefficient or impossible. Evolutionary computation (EC) is a technique to find the desired information in a limited time. Several EC-based algorithms (Kannimuthu and Premalatha 2014; Lin et al. 2017; Wu et al. 2017) are proposed to mine the required knowledge in different domains. However, no existing works focus on mining high fuzzy utility itemsets using EC-based algorithms.

In 2019, Wu et al. (2019) proposed high fuzzy utility itemsets for using a genetic algorithm (HFUI-GA) framework to discover the HFUIs in a definite time period from the large datasets. First, the proposed algorithm finds the various lengths of fuzzy itemsets separately. Then, it performs an evolutionary process to discover the HFUIs recursively until the complete set of promising candidate itemsets are examined. The proposed algorithm consists of the following steps: (1) chromosome encoding; (2) crossover operator; (3) mutation operator; (4) fitness function and evaporation policy; (5) selection operator; (6) termination conditions; (7) relationship array; (8) candidate pool and initialize population. In the designed HFUI-GA, each chromosome in the population is represented as a candidate of HFUI. The proposed process will randomly select two chromosomes from the population according to a relationship array. Each fuzzy item in a chromosome will perform the mutation operator. The mutation operator simply selects a fuzzy item from the candidate

pool to replace the selected one randomly or according to a relationship array. A chromosome with higher fuzzy utility illustrates a higher fitness value. An evaporation policy is utilized to keep the diversity of the population (Dorigo and Gambardella 1997). The tournament selection method is chosen as the selection operator. The termination conditions are employed to terminate the evolutionary process. Relationship array records the transaction weighted fuzzy utilization for each pair of items. The proposed HFUI-GA processes reveal not only HFUIs but also HTWFIUs. If a fuzzy itemset contains a subset that is not HTWFIUs, it is not HFUIs. Thus, the GA process in HFUI-GA generates an initial population from the previous level GA process.

Two improved strategies, namely evaporation strategies (HFUIGA(E)) and relationship array strategies (HFUIGA(R)), are designed to efficiently and effectively generate the HFUIs. HFUIGA(E) and HFUIGA(R) are evaluated and compared against the two strategies, namely HFUIGA algorithm with no strategy (HFUIGA(O)) and HFUIGA algorithm with two strategy (HFUIGA(F)), to show the effectiveness of the proposed algorithm. It is observed that HFUIGA(R) reveals fewer HFUIs in most of the cases. HFUIGA(O) shows similar behavior as of HFUIGA(R). Although the convergence of HFUIGA(O) is slower than that of HFUIGA(R), the HFUIGA(O) also focuses on some patterns due to the fitness values. The HFUIGA(E) and HFUIGA(F) have better performance than the other two algorithms due to the evaporation strategy. The evaporation strategy can discover more potential patterns to generate more HFUIs effectively. Sometimes, HFUIGA(F) can discover more HFUIs in the early iterations since it can quickly combine the suitable patterns to find HFUIs. However, this benefit is not very obvious. Finally, the HFUIGA(E) can try more possible patterns and keep the diversity in the population. Thus, the HFUIGA(E) can find more HFUIs in all cases. Generally, the HFUIGA(E) outperforms the other variants and always reveals the highest number of HFUIs. Experimental results prove the effectiveness of the proposed algorithms in terms of discovered HFUIs in a limited time for different variants.

3.2.4 Discussion of fuzzy logic-based approaches

The transactions may consist of the quantitative values in the retail datasets that is a tedious task. Therefore, the concept of fuzzy data mining is adopted to convert the quantitative values into linguistic regions. The fuzzy utility mining algorithms are proposed to efficiently mine fuzzy utility itemsets by integrating the fuzzy set concept and utility mining. However, it consumes lot of time to prune search space. To address this issue, an efficient fuzzy-list structure is used to efficiently

store the information for later mining process. However, it may miss some useful information to mine a complete set of HUIs. Therefore, an efficient algorithm with fuzzy-list structure and pruning strategies is proposed to mine multiple fuzzy utility itemsets without candidates generation. However, it suffers from high memory consumption.

In real-time applications, transactions may include transaction time and transaction stamps in the dataset. Therefore, temporal-based fuzzy utility mining problems find attention to these issues according to the fuzzy set concept. However, it is quite complex as compared to traditional fuzzy utility mining. A tree-based temporal fuzzy utility mining algorithm is proposed to extract temporal fuzzy utility patterns. However, these approaches use the single minimum utility threshold for all the items in the dataset. Therefore, they are failed to represent the true characteristics of the mining process. To address this issue, the temporal fuzzy utility mining with multiple minimum utility thresholds is proposed to extract interesting highly temporal fuzzy utility itemsets from the quantitative dataset.

The GA-based fuzzy utility itemsets mining efficiently discover the multi-level fuzzy utility itemsets in the dataset. It provides a simple way to use the derived non-dominated solutions according to the users' favorites. However, the performance can be further improved by utilizing other GA-based optimization techniques.

The overview of fuzzy logic soft computing techniques is shown in Table 9, including data structure, mining techniques, datasets used, pruning strategies, comparison with the state-of-the-art algorithms and compared base algorithms. The data structure, for example, fuzzy-list, is used to efficiently store the mining information in an appropriate way. The several pruning strategies, for example, fuzzy utility upper-bound, downward closure property and others are used to efficiently prune the search space and reduce the computational cost, thereby resulting in the improvement of mining performance. The proposed algorithms were compared and evaluated against the respective existing state-of-the-art algorithms and base algorithms to show the effectiveness and efficiency of these proposed methods from the benchmark datasets.

A brief summary of the fuzzy logic-based soft computing techniques, along with their pros and cons, is shown in Table 10. The survey has given the theoretical aspects of the proposed algorithms in order to emphasize on the core concept used in these methods. The pros and cons of each proposed algorithm are discussed in detail. The new researchers may work to improve these algorithms and also get novel ideas and insights into the novel potential methods in the field of fuzzy high-utility mining.

Table 9 Comparison of the existing fuzzy logic-based high-utility itemsets mining approaches

Year	Algorithm	Data structure	Dataset	Mining	Pruning strategy	The state-of-the-art algorithms	Base algorithms
<i>Simple fuzzy logic techniques</i>							
2009	HUFI-miner (Wang et al. 2009)	Small look-up tables	Transactional	HUFIs	Downward closure property of HMFI	None (first of its kind)	None
2014	GDFU (Lan et al. 2014)	FUUB	Transactional	HFUs	Data reduction strategy, HFUUBs	TPFU (Lan et al. 2015)	Two-phase (Liu et al. 2005)
2015	TPFU (Lan et al. 2015)	FUUB	Transactional	HFUs	Downward closure property in fuzzy sets, HFUUBs	None	Two-phase (Liu et al. 2005)
2015	FFI-Miner (Lin et al. 2015)	Fuzzy-list	Transactional	FFIs	PI-prune, PR-prune	FDTA (Hong et al. 1999), CFFP-tree [211], GDF (Hong et al. 2013), UBFFP-tree (Lin and Hong 2014)	None
2016	MFFI-Miner (Lin et al. 2017)	Fuzzy-list	Transactional	MFFIs	PI-prune, PR-prune	GDF (Hong et al. 2013), UBMPFP-tree (Lin et al. 2015)	None
2017	HFUBWSM (Kumar and Thakur 2018)	MFUTV	WDB	High utility-based web-page	fuzzy MFUTV and UDP	IHUP (Ahmed et al. 2009), UP-Growth (Tseng et al. 2013), HUI-Miner (Liu and Qu 2012), FHM (Fourmier-Viger et al. 2014)	None
2019	WFUM (Hong et al. 2019)	WFUUB	Transactional	HFUs	HUFUUBI, awfu	TPFU (Lan et al. 2015)	Two-phase (Liu et al. 2005)
2021	FHFAUIM (Hong et al. 2021)	WFUUB	Transactional	HFUs	HUFUUBI, awfu	TPFU (Lan et al. 2015)	Two-phase (Liu et al. 2005)
2021	HUWAS (Ganesan and Shankar 2021)	HUWAS-tree	Web log	High-utility product sets	EU, Fuzzy function	ANN and Top-k HUI (Duong et al. 2016)	None
<i>Temporal fuzzy logic techniques</i>							
2017	TP-TFU (Huang et al. 2017)	Periodic fuzzy utility table	TQD	HFUs	HTFUUB and htfuubr	TPFU (Lan et al. 2015)	Two-phase (Liu et al. 2005)
2018	TFUM with MMU (Huang et al. 2018)	Fuzzy-list	Transactional	HTFUs	HTFUUBMC	None	None
2019	TFUM using tree structure (Hong et al. 2019)	Tree structure	TQD	HTFUs	HTFUUBI and tfuubr	TP-TFU (Huang et al. 2017)	Two-phase (Liu et al. 2005)
2019	TPM (Huang et al. 2019)	Fuzzy-list	TQD	HFUs	HTFUUBI and tfur	TP-TFU (Huang et al. 2017)	Two-phase (Liu et al. 2005)
<i>Fuzzy logic & GA-based techniques</i>							
2013	TMOGFM (Chen et al. 2013)	Fuzzy-list	Transactional	Fuzzy association rules	Non-dominated solutions (NDS)	None	None
2019	HFUI-GA (Wu et al. 2019)	Fuzzy-list	Transactional	HFUs	HTWFUI	None (first of its kind)	None

Table 10 The pros and cons of the fuzzy logic-based HUIs mining approaches

Algorithm	Theoretical basis	Pros and cons
<i>Simple fuzzy logic techniques</i>		
HUFI-Miner Wang et al. (2009)	The proposed fuzzy approach discovers high fuzzy utility itemsets by considering both profits and quantities of items. The theoretical and empirical study are also provided	HUFI-Miner scans the original dataset at most twice. It can handle both integer and real quantities
GDFU Lan et al. (2014)	The proposed algorithm efficiently handles the problem of fuzzy utility mining by incorporating the effective reduction strategy	The proposed algorithm resolves the issue of fuzzy utility mining to prune unpromising candidate itemsets by adopting the fuzzy utility upper-bound which guarantees to find all the high fuzzy utility itemsets. However, it consumes high amount of time to mine fuzzy utility itemsets
TPFU Lan et al. (2015)	A two-phase TPFU proposed fuzzy utility upper-bound model and efficiently solves the problem of fuzzy utility mining	It significantly performs well on the real dataset with respect to the effectiveness of HFUs and run-time. However, it spends lots of computational time to mine fuzzy utility itemsets
FFI-Miner Lin et al. (2015)	The authors proposed a fast algorithm to discover FFIs without candidates generation	The proposed algorithm needs low memory to retain the relevant fuzzy information in the fuzzy-list structure. However, some information may be miss to discover the complete set of FFTs
MFFI-Miner Lin et al. (2017)	The authors proposed the MFFI-Miner algorithm to efficiently mine the complete set of MFFIs without candidates generation	The proposed pruning strategies greatly reduce the number of candidates. However, the proposed algorithm needs extra amount of memory to keep the fuzzy-list structure
HFUBWSM Kumar and Thakur (2018)	A high fuzzy utility strategy is proposed to mine the web page sets from the weblog dataset by using MFUTV and UDP pruning strategies	The proposed algorithm is efficient and effective with regards to the memory usage and run-time
WFUM Hong et al. (2019)	The authors proposed efficient two-stage weighted utility mining algorithm to mine the high weighted fuzzy utility itemsets	The designed WFUUB model prevents the loss of information in the mining process in level-wise manner
FHFAUIM Hong et al. (2021)	A single-stage tree structure-based method, based on FP-tree (Han et al. 2000), is proposed to efficiently discover fuzzy average-utility itemsets by using an external list for each node in the tree	The mined process can be performed directly in a single-phase. However, it consumes lots of memory to store the list information for each node
HUWAS Ganesan and Shankar (2021)	The proposed algorithm efficiently discovers all HUIs by adopting the optimization techniques	It significantly improves the execution speed by using the minimum utility threshold
<i>Temporal fuzzy logic soft computing techniques</i>		
TP-TFU Huang et al. (2017)	The authors proposed two-phased temporal fuzzy utility mining algorithm to mine the high temporal fuzzy utility itemsets from the temporal quantitative dataset	The proposed algorithm considers the period of transaction of the items. However, it consumes more computation time and slower than existing TP-HFU (Lan et al. 2015) method
TFUM with MMU Huang et al. (2018)	The temporal fuzzy utility mining algorithm with multiple minimum constraint is proposed to mine the HFUIs	The proposed algorithm is effective with respect to the execution time
TFUM using tree structure Hong et al. (2019)	The authors proposed temporal fuzzy utility mining algorithm using a modified tree structure, based on the FP-tree (Han et al. 2000), to mine the high temporal fuzzy utility itemsets from the quantitative dataset using the temporal property	The proposed algorithm performs well with regards to the memory usage and run-time

Table 10 continued

Algorithm	Theoretical basis	Pros and cons
TPM Huang et al. (2019)	An extended two-phased multi-conditional minimum threshold approach is proposed to mine the high temporal fuzzy utility itemsets by considering the temporal behavior, items' importance and the linguistic terms' usage	The proposed algorithm is efficient in execution time under different parameter settings. However, when TPM consists of all the identical value, it requires more execution time than TP-TFU (Huang et al. 2017) because it requires to find itemset constraints
<i>Fuzzy logic & GA-based techniques</i>		
TMOGFM Chen et al. (2013)	TMOGFM works in two stages; firstly, the multi-objective genetic algorithm is utilized and a set of non-dominated membership functions is derived. Secondly, the non-dominated solutions are divided into groups	TMOGFM provides a useful way to use the derived non-dominated solutions according to the users favorites. However, the performance of proposed algorithm could be further improved by incorporating other multi-objective genetic methods
HFUI-GA Wu et al. (2019)	A GA-based framework is proposed to mine the high fuzzy utility itemsets in a limited time for small datasets	Two proposed strategies enhance the performance of the mining process

4 Discussion

The paper discussed the soft computing techniques on high-utility itemsets mining that include evolutionary computation and fuzzy logic approaches, available in literature till-date. Evolutionary computation soft computing techniques are classified into GA, PSO, ACO, multi-objective optimization and hybrid approaches, while fuzzy logic-based soft computing techniques are classified into simple fuzzy, temporal fuzzy and GA-based fuzzy approaches.

GA is a directed optimization and search technique that can solve highly complex and nonlinear problems. GA-based techniques are more robust and general-purpose search methods to discover HUIs in the large exponential search space. The GA algorithms with pre-large concept significantly reduce the computational time to re-scan the dataset, thereby significantly improving the mining performance. They hide the sensitive high-utility itemsets in PPUM by using the concept of pre-large. A GA-based framework of sGA2DT and pGA2DT algorithms (Lin et al. 2015) find the appropriate transactions to be deleted for hiding sensitive itemsets. The $HUPE_{UMU}$ -GARM algorithm mines HUIs by using a minimum utility threshold. However, it generates a large number of HUIs if the threshold is set too low. On the other hand, $HUPE_{WUMU}$ -GARM algorithm mines HUIs without specifying the minimum utility threshold.

In contrast to GA-based approaches, the PSO-based approach [100] can efficiently discover the relevant HUIs by considering of the global and local results. A binary PSO-based (BPSO) algorithm with improved MP-tree, is proposed to efficiently mine HUIs by using the *sigmoid* updating strategy and TWU model. Another efficient PSO-based HUI-M-BPSO_{sig} is proposed to mine HUIs by using the *sigmoid* function and TWU model. A multi-objective PSO-based framework is proposed to hide the sensitive

information in PPDM by using LUS and GUS strategies. The GA-based multi-objective PSO approach with pre-large concept greatly reduces the computing time of the mining process. A multi-objective grid-based method is proposed to efficiently mine HUIs by adopting the pre-large concept and updating the *gbest* and *pbest* values in the evolution process.

The HUI-M-based ACO approaches (Wu et al. 2016, 2017) provide faster convergence speed and reduce the computational time to better judge the utility of the duplicate itemsets. The infrequent itemsets may play a significant role in gaining high profits from the business perspective. Therefore, a customized ant colony algorithm is proposed to efficiently mine the high-utility infrequent itemsets from the transactional dataset.

The previous algorithms consider frequent itemsets or high-utility itemsets separately. To address this issue, MOEA-FHUI framework is proposed to efficiently mine frequent and high-utility itemsets by considering both support and utility together, thereby resulting in the significant improvement of the mining process. CP-MOEA approach further extends the MOEA framework by considering the closed itemsets for the task of mining frequent and high-utility itemsets. However, they incur high costs for sparse datasets. To deal with this issue, SparseEA algorithm is proposed to obtain the effective sparse Pareto optimal solutions from the large sparse datasets.

The bio-inspired-based HUI-M approaches enhance the optimal solutions diversity in minimum iterations rather than to maintain the best values of population, significantly discover HUIs. The HUI mining based on the artificial bee colony algorithm using nectar source and bitmap structure is proposed to mine HUIs from the large dataset. The MOEA-HEUPM approach produces non-dominated solutions to make the decision without setting the pre-threshold in the uncertain dataset. A GWO-based approach produces

significant outputs in a limited number of iterations. The TNR-HUAUM approach is applicable to retail sectors for managing and allocating the space of shelf efficiently to get the maximum profits in the business.

The fuzzy utility itemsets mining not only discovers HUIs but also give relevant information about the purchase items. As the traditional DCP property does not hold for the fuzzy itemsets, therefore, the fuzzy utility upper-bound is used to avoid information loss. Similarly, weighted fuzzy utility upper-bound holds the monotonic property to process the itemsets in the weighted fuzzy utility itemset mining. FFI-Miner utilizes the fuzzy-list structure to efficiently reduce the computation time and search space, thereby significantly enhances the performance of the mining process. However, it generates a large number of candidates. To address this issue, MFFI-Miner algorithm is proposed to mine MFFIs without candidates generation. However, it consumes high amount of memory. Two-stage-based fuzzy utility mining algorithm (Hong et al. 2020) is computationally expensive. Therefore, a single-stage fuzzy average-utility mining is proposed to mine fuzzy average utility itemset by using external list for each node in the tree. However, memory is a critical issue.

The temporal fuzzy utility itemsets mining not only performs well in terms of items profit but also significantly obtains the transaction period of an itemset in the temporal dataset. The utility itemsets mining or fuzzy utility mining with multiple minimum thresholds provide the true characteristics of the real-time applications to obtain the maximum benefits in the businesses. To further enhance the mining performance, an extended two-phase multi-conditional minimum threshold approach is proposed to mine high temporal fuzzy utility mining by considering the temporal behavior and fuzzy linguistic terms.

To obtain the optimal Pareto solutions, a GA-based TMOGFM algorithm is proposed to find more accurate non-dominated solutions for decision-making process. When the search space is very large or the distinct items are big in the dataset, it is a challenging task to mine fuzzy utility itemsets. To address this issue, HFUI-GA framework is proposed to mine fuzzy utility itemsets in a limited amount of time. However, there is still a wide scope to use the fuzzy utility mining problem in other domains of data mining, including on-shelf, top-k, sequential mining, data stream, negative itemsets and many more.

The soft computing techniques in high-utility itemsets mining are applied on various real and synthetic datasets. The details of these real and synthetic datasets are shown in Table 11 and Table 12, respectively. The real datasets have been taken from various resources (Fournier-Viger et al. 2016; Asuncion and Newman 2007; Goethals 2003). Accident dataset anonymized traffic accident data. Chess dataset is prepared based on the UCI chess. Connect dataset is prepared based on the UCI connect-4. Foodmart dataset consists

of the customer transactions from a retail store, obtained and transformed from SQL-Server 2000. Mushroom dataset is prepared based on the UCI mushrooms. kddcup99 is transformed from the KDD CUP 1999. OnlineRetail is transformed from the Online Retail dataset. Pamp dataset is transformed from the PAMAP2 dataset. Pumsb dataset consists of the census data for population and housing. Kosarak is a very large dataset containing 990000 sequences of click-stream data from a Hungarian news portal. The dataset was converted in SPMF format using the original data. Retail dataset is the customer transactions from an anonymous Belgian retail store. Susy dataset is related to physics. It is related to particles detected using a particle accelerator. The dataset contains instances prepared using the original dataset. The instances were transformed to transactions, such that the value field of every attribute is divided into 10 equal parts and every part is represented by a number. As a result, the dataset has 190 items. USCensus is transformed from the US Census 1990 dataset. The real datasets BMSWebview-1 and BMSWebview-2 are taken from (Zheng et al. 2001). These two datasets were used in KDD CUP 2000. It contains click-stream data from e-commerce. On the other hand, the synthetic datasets have been taken from resources (Srikant 1999; Almaden 1994; Goethals 2003, [237]). These datasets are generated by using the random transaction dataset generator provided in SPMF.

5 Research opportunity and future directions

In the recent decade, lots of research advancement have been taken place in the high-utility itemsets mining with soft computing techniques. However, there are more emerging issues to address the conventional methods. In this section, we discuss key opportunities in high-utility itemsets mining with soft computing techniques.

GA-based algorithm with pre-large concept (Lin et al. 2014) can also be applied on deletion and modification of the transactions to hide the sensitive information in the dataset. Combining pre-large concept with insertion, deletion and modification of the transactions for PPUM may be a good research opportunity. The MOEAs using PPUM can be explored as a research opportunity. Other interesting areas are bound estimation of pattern measures (Zhang et al. 2016) and surrogate-based model (Vu et al. 2017) into MOEAs that decrease the computational time of the fitness evaluation. The MOEAs include applications in the area of personalized recommendation such as travel landscapes recommendation (Liu et al. 2011) and online shopping platforms [240] that will provide greater research opportunities. Furthermore, the MOEAs with sparsity evolution that includes Bayesian optimization (Wang et al. 2016) and decision vari-

Table 11 Statistical information about real datasets

Dataset	# of transactions	# of distinct items	Average length	Maximum length	Type
Accidents_10%	34018	469	34	46	Dense
Chess	3196	76	37	37	Dense
Connect	67557	130	43	43	Dense
Connect_50%	33779	129	43	43	Dense
Kddcup9910%	100000	133	16	16	Dense
Mushroom	8124	120	23	23	Dense
Pamp_10%	100000	142	24	24	Dense
Pumsb	49045	21113	74	74	Dense
Susy_1%	50000	191	19	19	Dense
USCensus_10%	100000	396	48	48	Dense
BMSWebview-1	59602	497	2.5	181	Sparse
BMSWebview-2	77512	3340	5		Sparse
BMS-POS	515597	1657	6.53	164	Sparse
Foodmart	21557	1559	4	11	Sparse
Kosarak	9899	1044	8		Sparse
OnlineRetail_10%	54191	2585	4.4	8	Sparse
Powerc_10%	104000	133	7	7	Sparse
Retail	88162	16470	10	76	Sparse

Table 12 Statistical information about synthetic datasets

Dataset	# of transactions	# of distinct items	Average length	Type
T6I4D10K	10000	100	6	Sparse
T10I10D100k	100000	870	10.1	Sparse
T10I4D10k	10000	100	10	Sparse
T10I4D100k	100000	870	10.1	Sparse
T10I4NXXDYK	100000-500000	2000-10000	10	Sparse

able clustering (Zhang et al. 2018) can be used to resolve the sparsity of MOPs. The HUIM-based bio-inspired algorithms such as parallelization of Bio-HUIF and bee swarm optimization (Ozturk et al. 2015) can be explored as a new research opportunity. The fuzzy utility itemset mining approaches could be used in the data streams, sequential pattern, on-shelf, tree-based and dynamic environment. The GA-based fuzzy mining approaches can be further enhanced by adopting multi-objective GAs and iterative learning methods. These may be good research opportunities in the field of fuzzy logic-based soft computing high-utility itemsets mining.

Online social platforms significantly influence the spreading of information in the real-world applications. The soft computing techniques may be good future scope in the area of social spider optimization (Singh et al. 2020). Jaya optimization could be used to resolve the knapsack problem. The combination of Jaya optimization and differential evolutionary techniques produces significant results to explore the vital information in the dataset (Wu and He 2020). It can be further explored in the field of soft computing techniques and researchers can adopt the Jaya optimization in high-utility

itemsets mining as a future direction. Another interesting area in moth flame optimization (Mirjalili 2015) which is inspired by nature optimization. The moth travels a long distance by maintaining an angle with respect to the moon. The behavior of moth can be used as a future direction to model the optimization techniques in HUIMs. A multi-objective binary bat optimization approach (Song et al. 2016) effectively mines the association rule and produces better results as compared to apriori-like approaches. The bat optimization could be a good future direction in the high-utility itemsets mining. There are lots of tremendous research opportunities and future directions in soft computing techniques based on high-utility itemsets mining. The researchers can adopt these novel areas of soft computing to further explore the various techniques and perform significant research.

6 Conclusions

In this article, we presented a comprehensive survey of a number of state-of-the-art algorithms on mining high utility

itemsets using soft computing techniques. We classify the soft computing-based techniques into two categories evolutionary computation and fuzzy logic. The evolutionary techniques included nature-inspired approaches like GA, PSO, ACO multi-objective optimization and hybrid approaches. The fuzzy logic techniques included simple fuzzy, temporal fuzzy and GA-based fuzzy approaches. The article presented a taxonomy of the state-of-the-art algorithms of evolutionary computation and fuzzy logic-based soft computing techniques for HUIs mining. We have provided a brief description as well as a detailed summary of the evolutionary computation and fuzzy logic-based HUIs mining approaches. The article also presented pros and cons of the nature-inspired and fuzzy logic-based techniques on HUIs mining. Finally, the key research opportunities and future directions of soft computing techniques in HUIs mining are discussed. Moreover, the paper provided insights to end-users for developing appropriate algorithms for mining high-utility itemset using evolutionary computation-based techniques.

Author Contributions K.S conceived of the presented idea. R.K. developed the theory and performed the computations. R.K. implemented the example and tables related to the running example. R.K. provided the data for Tables 6, 8, 10 and 11. K.S. provided the data for Tables 7 and 9. Both the authors analyzed and contributed to the final manuscript.

Declarations

Conflict of interest The authors declare no conflicts of interest. The article does not contain any studies with human or animal subjects. This article presents a survey of soft computing-based high-utility itemsets mining approaches.

References

- Agrawal S, Silakari S (2014) Frpso: Fletcher—reeves based particle swarm optimization for multimodal function optimization. *Soft Comput* 18(11):2227–2243. <https://doi.org/10.1007/s00500-013-1196-2>
- Agrawal R, Srikant R (2000) Privacy-preserving data mining. *SIGMOD Rec.* 29(2):439–450. <https://doi.org/10.1145/335191.335438>
- Agrawal R, Imielinski T, Swami A (1993) Database mining: a performance perspective. *IEEE Trans Knowl Data Eng* 5(6):914–925. <https://doi.org/10.1109/69.250074>
- Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large databases. In: *Proceedings of the 20th international conference on very large data bases, VLDB '94*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p 487–499
- Agrawal R, Srikant R (1995) Mining sequential patterns. In: *Proceedings of the eleventh international conference on Data engineering*, pp 3–14. <https://doi.org/10.1109/ICDE.1995.380415>
- Ahmed CF, Tanbeer SK, Jeong B-S, Lee Y-K (2009) Efficient tree structures for high utility pattern mining in incremental databases. *IEEE Trans Knowl Data Eng* 21(12):1708–1721. <https://doi.org/10.1109/TKDE.2009.46>
- Ahmed U, Lin JC-W, Srivastava G, Yasin R, Djenouri Y (2021) An evolutionary model to mine high expected utility patterns from uncertain databases. *IEEE Trans Emerg Topic Comput Intell* 5(1):19–28. <https://doi.org/10.1109/TETCI.2020.3000224>
- Alcalá-Fdez J, Alcalá R, Gacto MJ, Herrera F (2009) Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. *Fuzzy Sets Syst* 160(7):905–921. <https://doi.org/10.1016/j.fss.2008.05.012>
- Alhaji R, Kaya M (2007) Multi-objective genetic algorithms based automated clustering for fuzzy association rules mining. *J Intell Inf Syst* 31(3):243–264. <https://doi.org/10.1007/s10844-007-0044-1>
- Almaden I (1994) Quest synthetic data generation code
- Amiri A (2007) Dare to share: protecting sensitive knowledge with data sanitization, decision Support Systems 43(1), 181–191, mobile commerce: strategies, technologies, and applications. <https://doi.org/10.1016/j.dss.2006.08.007>. <https://www.sciencedirect.com/science/article/pii/S0167923606001230>
- Arun Kumar MS, Suresh P, Gunavathi C (2020) High utility infrequent itemset mining using a customized ant colony algorithm. *Int. J. Parallel Program.* 48(5):833–849. <https://doi.org/10.1007/s10766-018-0621-7>
- Asuncion A, Newman D (2007) UCI machine learning repository
- Atallah M, Bertino E, Elmagarmid A, Ibrahim M, Verykios V (1999) Disclosure limitation of sensitive rules. In: *Proceedings 1999 workshop on knowledge and data engineering exchange (KDEX'99)* (Cat. No.PR00453), pp 45–52. <https://doi.org/10.1109/KDEX.1999.836532>
- B V (2019) Biomedical image analysis using semantic segmentation. *J Innov Image Process* 1:91–101. <https://doi.org/10.36548/jiip.2019.2.004>
- Bäck T (1996) *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press Inc, USA
- Badhon B, Kabir MMJ, Xu S, Kabir M (2021) A survey on association rule mining based on evolutionary algorithms. *Int J Comput Appl* 43(8):775–785. <https://doi.org/10.1080/1206212X.2019.1612993>
- Beasley D, Bull D, Martin R (1993) An overview of genetic algorithms: Pt1, fundamentals. *Univer Comput* 15:58–69
- Berkhin P (2006) *A survey of clustering data mining techniques*. Springer, Berlin, Heidelberg, pp 25–71. https://doi.org/10.1007/3-540-28349-8_2
- Berson A, Smith SJ (1997) *Data warehousing, data mining, and olap*, 1st edn. McGraw-Hill Inc, USA
- Beume N, Naujoks B, Emmerich M (2007) Sms-emoa: multiobjective selection based on dominated hypervolume. *Eur J Oper Res* 181(3):1653–1669. <https://doi.org/10.1016/j.ejor.2006.08.008>. <https://www.sciencedirect.com/science/article/pii/S0377221706005443>
- Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. *Knowl-Based Syst* 46:109–132. <https://doi.org/10.1016/j.knosys.2013.03.012>. <https://www.sciencedirect.com/science/article/pii/S0950705113001044>
- Burdick D, Calimlim M, Flannick J, Gehrke J, Yiu T (2005) Mafia: a maximal frequent itemset algorithm. *IEEE Trans Knowl Data Eng* 17(11):1490–1504. <https://doi.org/10.1109/TKDE.2005.183>
- Cai X, Li Y, Fan Z, Zhang Q (2015) An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization. *IEEE Trans Evol Comput* 19(4):508–523. <https://doi.org/10.1109/TEVC.2014.2350995>
- Cao H, Yang S, Wang Q, Wang Q, Zhang L (2019) A closed itemset property based multi-objective evolutionary approach for mining frequent and high utility itemsets. In: *2019 IEEE congress on evolutionary computation (CEC)*, pp 3356–3363. <https://doi.org/10.1109/CEC.2019.8789985>
- Cattral R, Oppacher F, Graham KJL (2009) Techniques for evolutionary rule discovery in data mining. In: *2009 IEEE congress on evolutionary computation*, pp 1737–1744. <https://doi.org/10.1109/CEC.2009.4983151>

- Chan KCC, Au W-H (1997) An effective algorithm for mining interesting quantitative association rules. In: Proceedings of the 1997 ACM symposium on applied computing, SAC '97, association for computing machinery, New York, NY, USA, p 88–90. <https://doi.org/10.1145/331697.331714>
- Chan R, Yang Q, Shen Y-D (2003) Mining high utility itemsets. In: Third IEEE international conference on data mining, pp 19–26. <https://doi.org/10.1109/ICDM.2003.1250893>
- Chen C-H, He J-S, Hong T-P (2013) A two-stage multi-objective genetic-fuzzy mining algorithm. In: 2013 IEEE international workshop on genetic and evolutionary fuzzy systems (GEFS), pp 16–20. <https://doi.org/10.1109/GEFS.2013.6601050>
- Chen C-H, Hong T-P, Tseng VS, Lee C-S (2007) A genetic-fuzzy mining approach for items with multiple minimum supports. In: 2007 IEEE international fuzzy systems conference, pp 1–6. <https://doi.org/10.1109/FUZZY.2007.4295628>
- Chen DC Daqing, Guo K (2012) Data mining for the online retail industry: A case study of rfm model-based customer segmentation using data mining. *J Database Market Customer Strateg Manag* 19:197–208. <https://doi.org/10.1057/dbm.2012.17>. <https://link.springer.com/article/10.1057/dbm.2012.17>
- Chen M-S, Han J, Yu PS (1996) Data mining: an overview from a database perspective. *IEEE Trans Knowl Data Eng* 8(6):866–883. <https://doi.org/10.1109/69.553155>
- Chen C-H, Hong T-P, Tseng VS (2009) An improved approach to find membership functions and multiple minimum supports in fuzzy data mining. *Expert Syst Appl* 36(6):10016–10024. <https://doi.org/10.1016/j.eswa.2009.01.067>
- Chen C-H, Hong T-P, Lee Y-C (2012) Genetic-fuzzy mining with taxonomy. *Int J Uncert Fuzz Knowl -Based Syst* 20(supp02):187–205. <https://doi.org/10.1142/S021848851240020X>
- Chen C-H, He J-S, Hong T-P (2013) Moga-based fuzzy data mining with taxonomy. *Knowl -Based Syst* 54(C):53–65
- Cheng R, Jin Y, Narukawa K, Sendhoff B (2015) A multiobjective evolutionary algorithm using gaussian process-based inverse modeling. *IEEE Trans Evol Comput* 19(6):838–856. <https://doi.org/10.1109/TEVC.2015.2395073>
- Cheng P, Pan J (2014) Association rule hiding based on evolutionary multi-objective optimization by removing items. In: Brodley CE, Stone P (Eds.), Proceedings of the twenty-eighth AAAI conference on artificial intelligence, July 27–31, 2014, Québec City, Québec, Canada, AAAI Press, pp. 3100–3101. <http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8564>
- Chu C-J, Tseng VS, Liang T (2009) An efficient algorithm for mining high utility itemsets with negative item values in large databases. *Appl Math Comput* 215(2):767–778. <https://doi.org/10.1016/j.amc.2009.05.066>. <https://www.sciencedirect.com/science/article/pii/S009630030900561X>
- Coello C Coello, Lechuga M (2002) Mopso: a proposal for multiple objective particle swarm optimization. In: Proceedings of the 2002 congress on evolutionary computation. CEC'02 (Cat. No.02TH8600), Vol 2, pp 1051–1056. <https://doi.org/10.1109/CEC.2002.1004388>
- Coello C Coello (2006) Evolutionary multi-objective optimization: a historical view of the field. *IEEE Comput Intell Magaz* 1(1):28–36. <https://doi.org/10.1109/MCI.2006.1597059>
- Colomi A, Dorigo M, Maniezzo V (1991) Distributed optimization by ant colonies
- Daniel EL (1991) Knowledge discovery as a threat to database security. In: Proceedings of the 1st international conference on knowledge discovery and databases, Vol 107, p 516
- Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. *IEEE Trans Evol Comput* 15(1):4–31. <https://doi.org/10.1109/TEVC.2010.2059031>
- Dash M, Liu H (1997) Feature selection for classification. *Intell Data Anal* 1(1):131–156. [https://doi.org/10.1016/S1088-467X\(97\)00008-5](https://doi.org/10.1016/S1088-467X(97)00008-5). <https://www.sciencedirect.com/science/article/pii/S1088467X97000085>
- Das S, Mullick SS, Suganthan P (2016) Recent advances in differential evolution—an updated survey. *Swarm Evol Comput* 27:1–30. <https://doi.org/10.1016/j.swevo.2016.01.004>. <https://www.sciencedirect.com/science/article/pii/S2210650216000146>
- Dasseni E, Verykios V, Elmagarmid A, Bertino E (2001) Hiding association rules by using confidence and support. *Inf Hiding*. https://doi.org/10.1007/3-540-45496-9_27
- Dawar S, Goyal V (2015) Up-hist tree: An efficient data structure for mining high utility patterns from transaction databases. In: Proceedings of the 19th international database engineering and applications symposium, IDEAS '15, association for computing machinery, New York, NY, USA, p 56–61. <https://doi.org/10.1145/2790755.2790771>
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans Evol Comput* 6(2):182–197. <https://doi.org/10.1109/4235.996017>
- Dong J, Han M (2007) Bittablefi: An efficient mining frequent itemsets algorithm. *Knowl -Based Syst* 20(4):329–335. <https://doi.org/10.1016/j.knosys.2006.08.005>. <https://www.sciencedirect.com/science/article/pii/S09507051060001493>
- Dorigo M, Gambardella L (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans Evol Comput* 1(1):53–66. <https://doi.org/10.1109/4235.585892>
- Dorigo M, Maniezzo V, Colomi A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B Cybern* 26(1):29–41. <https://doi.org/10.1109/3477.484436>
- Dunning LA, Kresman R (2013) Privacy preserving data sharing with anonymous id assignment. *IEEE Trans Inf Forens Secur* 8(2):402–413. <https://doi.org/10.1109/TIFS.2012.2235831>
- Duong Q-H, Liao B, Fournier-Viger P, Dam T-L (2016) An efficient algorithm for mining the top-k high utility itemsets, using novel threshold raising and pruning strategies. *Knowl -Based Syst* 104(C):106–122. <https://doi.org/10.1016/j.knosys.2016.04.016>
- Engelbrecht A, Pampara G (2007) Binary differential evolution strategies. In: 2007 IEEE congress on evolutionary computation, pp 1942–1947. <https://doi.org/10.1109/CEC.2007.4424711>
- Erwin A, Gopalan RP, Achuthan N (2007) Ctu-mine: an efficient high utility itemset mining algorithm using the pattern growth approach. In: 7th IEEE international conference on computer and information technology (CIT 2007), pp 71–76. <https://doi.org/10.1109/CIT.2007.120>
- Fang W, Zhang Q, Sun J, Wu X-J (2020) Mining high quality patterns using multi-objective evolutionary algorithm. *IEEE Trans Knowl Data Eng* 1–1. <https://doi.org/10.1109/TKDE.2020.3033519>
- Faramondi L, Oliva G, Panzieri S, Pascucci F, Schlueter M, Munetomo M, Setola R (2019) Network structural vulnerability: a multiobjective attacker perspective. *IEEE Trans Syst Man Cybern Syst* 49(10):2036–2049. <https://doi.org/10.1109/TSMC.2018.2790438>
- Fieldsend J, Singh S (2005) Pareto evolutionary neural networks. *IEEE Trans Neural Netw* 16(2):338–354. <https://doi.org/10.1109/TNN.2004.841794>
- Fournier-Viger RNBV, Philippe, Lin JC-W, Tseng VS (2019) High-utility pattern mining—theory, algorithms and applications, 1st Edition, Springer, Cham. <https://doi.org/10.1007/978-3-030-04921-8>. <https://link.springer.com/book/10.1007/978-3-030-04921-8>
- Fournier-Viger P, Wu C-W, Zida S, Tseng VS (2014) Fhm: faster high-utility itemset mining using estimated utility co-occurrence pruning. In: Andreassen T, Christiansen H, Cubero J-C, Raś ZW (eds) *Found Intel Syst*. Springer, Cham, pp 83–92
- Fournier-Viger P, Lin JC-W, Gomariz A, Gueniche T, Soltani A, Deng Z, Lam HT (2016) The spmf open-source data mining library version 2. In: Berendt B, Bringmann B, Fromont É, Garriga G, Miettinen

- P, Tatti N, Tresp V (eds) *Mach Learn Knowl Discov Databases*. Springer, Cham, pp 36–40
- Freitas AA (2003) A survey of evolutionary algorithms for data mining and knowledge discovery. Springer, Berlin, Heidelberg, pp 819–845. https://doi.org/10.1007/978-3-642-18965-4_33
- Gade K, Wang J, Karypis G (2004) Efficient closed pattern mining in the presence of tough block constraints. In: *Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining, KDD '04*, association for computing machinery, New York, NY, USA, p 138–147. <https://doi.org/10.1145/1014052.1014070>
- Gan W, Lin JC-W, Fournier-Viger P, Chao H-C, Tseng VS, Yu PS (2021) A survey of utility-oriented pattern mining. *IEEE Trans Knowl Data Eng* 33(4):1306–1327. <https://doi.org/10.1109/TKDE.2019.2942594>
- Ganesan M, Shankar S (2021) High utility fuzzy product mining (hufpm) using investigation of huwas approach. *J Amb Intell Human Comput*, 1–11 <https://doi.org/10.1007/s12652-021-03231-8>. <https://link.springer.com/article/10.1007/s12652-021-03231-8>
- Goethals B (2003) Frequent itemset mining dataset repository. *Freq Itemset Min Implement (FIMI'03)*
- Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. *Mach Learn* 3:95–99. <https://doi.org/10.1023/A:1022602019183>. <https://link.springer.com/article/10.1023/A:1022602019183>
- Goldberg DE (1989) *Genetic algorithms in search, optimization and machine learning*, 1st edn. Addison-Wesley Longman Publishing Co. Inc, USA
- Goyal V, Dawar S, Sureka A (2015) High utility rare itemset mining over transaction databases, pp 27–40. https://doi.org/10.1007/978-3-319-16313-0_3
- Guidotti A, Ricciardo, Gabrielli L, Giannotti F Discovering temporal regularities in retail customers' shopping behavior. *EPJ Data Sci* 7(6). <https://doi.org/10.1140/epjds/s13688-018-0133-0>. <https://epjdatascience.springeropen.com/articles/10.1140/epjds/s13688-018-0133-0>
- Guo-Cheng Lan T-PH, Tseng VS (2014) An efficient projection-based indexing approach for mining high utility itemsets. *Knowl Inf Syst* 38:85–107. <https://doi.org/10.1007/s10115-012-0492-y>. <https://link.springer.com/article/10.1007/s10115-012-0492-y>
- Gupta D, Chauhan AS (2015) Ant colony based optimization from infrequent itemsets. In: Satapathy SC, Biswal BN, Udgata SK, Mandal J (eds) *Proceedings of the 3rd international conference on frontiers of intelligent computing: theory and applications (FICTA) 2014*. Springer, Cham, pp 797–804
- Haglin D, Manning A (2007) On minimal infrequent itemset mining, pp 141–147
- Hamdani TM, Won J-M, Alimi AM, Karray F (2007) Multi-objective feature selection with NSGA ii. In: Beliczynski B, Dzielinski A, Iwanowski M, Ribeiro B (eds) *Adapt Nat Comput Algorith*. Springer, Berlin, Heidelberg, pp 240–247
- Han DXJ, Cheng H, Yan X (2007) Frequent pattern mining: current status and future directions. *Data Mining Knowl Disc* 15:55–86. <https://doi.org/10.1007/s10618-006-0059-1>. <https://link.springer.com/article/10.1007/s10618-006-0059-1>
- Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. *SIGMOD Rec* 29(2):1–12. <https://doi.org/10.1145/335191.335372>
- Han S, Ng WK (2007) Privacy-preserving genetic algorithms for rule discovery. In: *Proceedings of the 9th international conference on data warehousing and knowledge discovery, DaWaK'07*, Springer, Berlin, Heidelberg, p 407–417
- Holland J Adaptation in natural and artificial systems: an introductory analysis with application to biology. *Control Artif Intell*
- Holland JH (1992) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence*. MIT Press, Cambridge, MA, USA
- Hong T-P, Chiang P-H, Huang W-M, Lin JC-W (2019) Fuzzy utility mining under minimum weight constraint of multiple item weights. In: *2019 international conference on data mining workshops (ICDMW)*, pp 982–986. <https://doi.org/10.1109/ICDMW.2019.00142>
- Hong T-P, Lin C-Y, Huang W-M, Li S-M, Wang S-L, Lin JC-W (2019) Mining temporal fuzzy utility itemsets by tree structure. In: *2019 IEEE international conference on big data (big data)*, pp 2659–2663. <https://doi.org/10.1109/BigData47090.2019.9006317>
- Hong T-P, Lin K-Y, Wang S-L (2003) Fuzzy data mining for interesting generalized association rules. *Fuzzy Sets Syst* 138(2):255–269. [https://doi.org/10.1016/S0165-0114\(02\)00272-5](https://doi.org/10.1016/S0165-0114(02)00272-5). <https://www.sciencedirect.com/science/article/pii/S0165011402002725>
- Hong T-P, Wang C-Y Maintenance of association rules using pre-large itemsets <https://doi.org/10.4018/978-1-59904-120-9.ch003>
- Hong T-P, Kuo C-S, Chi S-C (1999) Mining association rules from quantitative data. *Intell Data Anal* 3(5):363–376. [https://doi.org/10.1016/S1088-467X\(99\)00028-1](https://doi.org/10.1016/S1088-467X(99)00028-1)
- Hong T-P, Wang C-Y, Tao Y-H (2001) A new incremental data mining algorithm using pre-large itemsets. *Intell Data Anal* 5(2):111–129
- Hong T-P, Chen C-H, Wu Y-L, Lee Y-C (2006) A ga-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions. *Soft Comput* 10(11):1091–1101. <https://doi.org/10.1007/s00500-006-0046-x>
- Hong T-P, Lan G-C, Lin Y-H, Pan S-T (2013) An effective gradual data-reduction strategy for fuzzy itemset mining. *Int J Fuzzy Syst* 15(2):170–181
- Hong T-P, Lin C-W, Yang K-T, Wang S-L (2013) Using tf-idf to hide sensitive itemsets. *Appl Intell* 38(4):502–510. <https://doi.org/10.1007/s10489-012-0377-5>
- Hong T-P, Lin C-W, Lin T-C (2014) The mffp-tree fuzzy mining algorithm to discover complete linguistic frequent itemsets. *Comput Intell* 30(1):145–166. <https://doi.org/10.1111/j.1467-8640.2012.00467.x>
- Hong T-P, Ku M-P, Chiu H-W, Huang W-M, Li S-M, Lin JC-W (2021) A single-stage tree-structure-based approach to determine fuzzy average-utility itemsets. In: Fujita H, Selamat A, Lin JC-W, Ali M (eds) *Advances and trends in artificial intelligence. artificial intelligence practices*. Springer, Cham, pp 66–72
- Hong T, Ku M, Huang W, Li S, Lin JC (2020) A tree-based fuzzy average-utility mining algorithm. In: Fatta GD, Sheng VS, Cuzocrea A, Zaniolo C, Wu X (Eds.), *20th international conference on data mining workshops, ICDM workshops 2020, Sorrento, Italy, November 17–20, 2020, IEEE*, pp 669–672. <https://doi.org/10.1109/ICDMW51313.2020.00094>
- Hooshadad M, Bayat S, Naimi P, Mirian MS, Zaiane OR (2012) UAPRIORI: an algorithm for finding sequential patterns in probabilistic data. *World scientific proceedings series on computer engineering and information science*, pp. 907–912. https://doi.org/10.1142/9789814417747_0145. https://www.worldscientific.com/doi/abs/10.1142/9789814417747_0145
- Huang W-M, Hong T-P, Lin JC-W, Chiang M-C (2018) Temporal fuzzy utility mining with multiple minimum utility thresholds. In: *1st international workshop on utility-driven mining*
- Huang W-M, Hong T-P, Lan G-C, Chiang M-C, Lin JC-W (2017) Temporal-based fuzzy utility mining. *IEEE Access* 5:26639–26652. <https://doi.org/10.1109/ACCESS.2017.2774510>
- Huang W-M, Hong T-P, Chiang M-C, Lin JC-W (2019) Using multi-conditional minimum thresholds in temporal fuzzy utility mining. *Int J Comput Intell Syst* 12:613–626. <https://doi.org/10.2991/ijcis.d.190426.001>
- Hu J, Mojsilovic A (2007) High-utility pattern mining: a method for discovery of high-utility item sets. *Pattern Recogn* 40(11):3317–

3324. <https://doi.org/10.1016/j.patcog.2007.02.003>. <https://www.sciencedirect.com/science/article/pii/S0031320307000866>
- Ishibuchi H, Tsukamoto N, Nojima Y (2010) Diversity improvement by non-geometric binary crossover in evolutionary multiobjective optimization. *IEEE Trans Evol Comput* 14(6):985–998. <https://doi.org/10.1109/TEVC.2010.2043365>
- Javangula V et al (2021) Mining of high utility item sets using genetic algorithm. *Turkish J Comput Math Educ (TURCOMAT)* 12(9):2437–2448
- Lin PFVTPHJMTWJCH, Li T, Zhan J (2017) Efficient mining of multiple fuzzy frequent itemsets. *Int J Fuzzy Syst* 19:1032–1040. <https://doi.org/10.1007/s40815-016-0246-1>. <https://link.springer.com/article/10.1007/s40815-016-0246-1>
- Jerry PFVTPH, Chun-Wei L, Lu Y, Voznak M (2017) A binary pso approach to mine high-utility itemsets. *Soft Comput* 21:5103–5121. <https://doi.org/10.1007/s00500-016-2106-1>. <https://link.springer.com/article/10.1007/s00500-016-2106-1>
- Jin Y, Okabe T, Sendhoff B (2004) Evolutionary multi-objective optimization approach to constructing neural network ensembles for regression. In: *Advances in natural computation—applications of multi-objective evolutionary algorithms*, World Scientific, pp. 635–673. https://doi.org/10.1142/9789812567796_0027
- Kaelo P, Ali M (2006) A numerical study of some modified differential evolution algorithms. *Eur J Oper Res* 169(3):1176–1184. <https://doi.org/10.1016/j.ejor.2004.08.047>. <https://www.sciencedirect.com/science/article/pii/S037722170500281X>
- Kalaipriyan TVSSP, Sourabh P, Thirumal V (2019) Reinforced cuckoo search algorithm-based multimodal optimization. *Appl Intell* 49(6):2059–2083. <https://doi.org/10.1007/s10489-018-1355-3>. <https://link.springer.com/article/10.1007/s10489-018-1355-3>
- Kannimuthu S, Premalatha K (2014) Discovery of high utility itemsets using genetic algorithm with ranked mutation. *Appl Artif Intell* 28(4):337–359
- Karaboga D An idea based on honey bee swarm for numerical optimization, technical report—tr06, Technical Report, Erciyes University
- Karakatić S, Podgorelec V (2015) A survey of genetic algorithms for solving multi depot vehicle routing problem. *Appl Soft Comput* 27(C):519–532. <https://doi.org/10.1016/j.asoc.2014.11.005>
- Kaya M (2005) Multi-objective genetic algorithm based approaches for mining optimized fuzzy association rules. *Soft Comput* 10:578–586. <https://doi.org/10.1007/s00500-005-0509-5>. <https://link.springer.com/article/10.1007/s00500-005-0509-5>
- Kaya M, Alhajj R (2005) Genetic algorithm based framework for mining fuzzy association rules. *Fuzzy Sets Syst* 152(3):587–601. <https://doi.org/10.1016/j.fss.2004.09.014>
- Kaya M, Alhajj R (2006) Utilizing genetic algorithms to optimize membership functions for fuzzy weighted association rules mining. *Appl Intell* 24(1):7–15. <https://doi.org/10.1007/s10489-006-6925-0>
- Kazimipour B, Li X, Qin AK (2014) A review of population initialization techniques for evolutionary algorithms. In: *2014 IEEE congress on evolutionary computation (CEC)*, pp 2585–2592. <https://doi.org/10.1109/CEC.2014.6900618>
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of ICNN'95—international conference on neural networks*, Vol 4, pp 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>
- Kennedy J, Eberhart R (1997) A discrete binary version of the particle swarm algorithm. In: *1997 IEEE international conference on systems, man, and cybernetics*. *Comput Cybern Simul* Vol 5, pp 4104–4108. <https://doi.org/10.1109/ICSMC.1997.637339>
- Krishna GJ, Ravi V (2020) Mining top high utility association rules using binary differential evolution. *Eng Appl Artif Intell* 96:103935. <https://doi.org/10.1016/j.engappai.2020.103935>. <https://www.sciencedirect.com/science/article/pii/S0952197620302578>
- Kumar V, Thakur RS (2018) High fuzzy utility strategy based webpage sets mining from weblog database. *Int J Intell Eng Syst* 11(1):191–200
- Kuo R, Chao C, Chiu Y (2011) Application of particle swarm optimization to association rule mining. *Appl Soft Comput* 11(1):326–336. <https://doi.org/10.1016/j.asoc.2009.11.023>. <https://www.sciencedirect.com/science/article/pii/S1568494609002361>
- Kuok CM, Fu A, Wong MH (1998) Mining fuzzy association rules in databases. *SIGMOD Rec* 27(1):41–46. <https://doi.org/10.1145/273244.273257>
- Lalou M, Tahraoui MA, Kheddouci H (2018) The critical node detection problem in networks: a survey. *Compu Sci Rev* 28:92–117. <https://doi.org/10.1016/j.cosrev.2018.02.002>. <https://www.sciencedirect.com/science/article/pii/S1574013716302416>
- Lan G-C, Hong T-P, Lin Y-H, Wang S-L (2014) Fast discovery of high fuzzy utility itemsets. In: *2014 IEEE international conference on systems, man, and cybernetics (SMC)*, pp 2764–2767. <https://doi.org/10.1109/SMC.2014.6974346>
- Lan G-C, Hong T-P, Lin Y-H, Wang S-L (2015) Fuzzy utility mining with upper-bound measure. *Appl Soft Comput* 30:767–777. <https://doi.org/10.1016/j.asoc.2015.01.055>. <https://www.sciencedirect.com/science/article/pii/S1568494615000769>
- Lee Y-C, Hong T-P, Wang T-C (2008) Multi-level fuzzy mining with multiple minimum supports. *Exp Syst Appl* 34(1):459–468. <https://doi.org/10.1016/j.eswa.2006.09.011>. <https://www.sciencedirect.com/science/article/pii/S0957417406002958>
- Levesque J-C, Durand A, Gagne C, Sabourin R (2012) Multi-objective evolutionary optimization for generating ensembles of classifiers in the roc space. In: *Proceedings of the 14th annual conference on genetic and evolutionary computation, GECCO '12, association for computing machinery*, New York, NY, USA, p 879–886. <https://doi.org/10.1145/2330163.2330285>
- Li F-g, Sun Y-j, Ni Z-w, Liang Y, Mao X-m (2012) The utility frequent pattern mining based on slide window in data stream. In: *2012 fifth international conference on intelligent computation technology and automation*, pp 414–419. <https://doi.org/10.1109/ICICTA.2012.110>
- LI ZJS Xiao Lei, QIAN JX (2002) An optimizing method based on autonomous animats: fish-swarm algorithm. *Syst Eng-Theory Pract* 22(11):32–38. [https://doi.org/10.12011/1000-6788\(2002\)11-32](https://doi.org/10.12011/1000-6788(2002)11-32). <http://www.sysengi.com/EN/Y2002/V22/I11/32>
- Li H, Zhang Q, Deng J, Xu Z (2018) A preference-based multiobjective evolutionary approach for sparse optimization. *IEEE Trans Neural Netw Learn Syst* 29(5):1716–1731. <https://doi.org/10.1109/TNNLS.2017.2677973>
- Lin C-W, Hong T-P, Lu W-H. An efficient tree-based fuzzy data mining approach. *Int J Fuzzy Syst* 12
- Lin C-W, Hong T-P, Lu W-H (2009) The pre-fufp algorithm for incremental mining. *Exp Syst Appl* 36(5):9498–9505. <https://doi.org/10.1016/j.eswa.2008.03.014>. <https://www.sciencedirect.com/science/article/pii/S0957417408001826>
- Lin C-W, Hong T-P (2013) A survey of fuzzy web mining, Wiley interdisciplinary reviews. *Data mining and knowledge discovery* 3(3):190–199. <https://doi.org/10.1002/widm.1091>. <https://wires.onlinelibrary.wiley.com/doi/10.1002/widm.1091>
- Lin C-W, Hong T-P, Wong J-W, Lan G-C (2013) Privacy preserving high utility mining based on genetic algorithms. In: *2013 IEEE international conference on granular computing (GrC)*, pp 191–195. <https://doi.org/10.1109/GrC.2013.6740406>
- Lin C-W, Hong T-P, Wong J-W, Lan g-c, Lin W-Y (2014) A ga-based approach to hide sensitive high utility itemsets. *Sci World J* 2014, 804629. <https://doi.org/10.1155/2014/804629>. <https://www.hindawi.com/journals/tswj/2014/804629/>

- Lin C-W, Yang L, Viger P, Fournier, Frnda J, Sevcik L, Vozňák M (2015) An evolutionary algorithm to mine high-utility itemsets. *Adv Electr Electron Eng* 13. <https://doi.org/10.15598/aece.v13i4.1474>
- Lin C-W, Yang L, Viger P, Fournier, Wu M-T, Hong T-P, Wang L-L (2015) A swarm-based approach to mine high-utility itemsets, Vol 540, pp 572–581. https://doi.org/10.1007/978-3-662-48319-0_48
- Lin C-W, Yuyu Z, Zhang B, Viger P, Fournier, Djenouri Y (2015) Hiding sensitive itemsets with multiple objective optimization. *Soft Comput* 23. <https://doi.org/10.1007/s00500-019-03829-3>
- Lin JC-W, Gan W, Fournier-Viger P, Hong T-P (2015) Rwfim: recent weighted-frequent itemsets mining. *Eng Appl Artif Intell* 45:18–32. <http://dblp.uni-trier.de/db/journals/eaai/eaai45.html#LinGFH15>
- Lin JC-W, Gan W, Fournier-Viger P, Hong T-P, Tseng VS (2015) Mining high-utility itemsets with various discount strategies. In: 2015 IEEE international conference on data science and advanced analytics (DSAA), pp 1–10. <https://doi.org/10.1109/DSAA.2015.7344861>
- Lin JC-W, Gan W, Fournier-Viger P, Yang L, Liu Q, Frnda J, Sevcik L, Voznak M (2016) High utility-itemset mining and privacy-preserving utility mining. *Perspect Sci* 7:74–80, 1st Czech-China scientific conference 2015. <https://doi.org/10.1016/j.pisc.2015.11.013>. <https://www.sciencedirect.com/science/article/pii/S2213020915000580>
- Lin JC-W, Hong T-P, Lin T-C (2015) A cmffp-tree algorithm to mine complete multiple fuzzy frequent itemsets. *Appl Soft Comput* 28:431–439. <https://doi.org/10.1016/j.asoc.2014.11.049>. <https://www.sciencedirect.com/science/article/pii/S1568494614006139>
- Lin C-W, Hong T-P (2014) Mining fuzzy frequent itemsets based on ubffp trees. *J Intell Fuzzy Syst* 27(1):535–548. <https://doi.org/10.3233/IFS-131022>
- Lin C-W, Hong T-P, Chang C-C, Wang S-L (2013) A greedy-based approach for hiding sensitive itemsets by transaction insertion. *J Inf Hiding Multimed Signal Process* 4:201–214
- Lin C-W, Zhang B, Yang K-T, Hong T-P (2014) Efficiently hiding sensitive itemsets with transaction deletion based on genetic algorithms. *Sci World J* 2014:398269. <https://doi.org/10.1155/2014/398269>
- Lin C-W, Li T, Viger P, Fournier, Hong T-P (2015) A fast algorithm for mining fuzzy frequent itemsets. *J Intell Fuzzy Syst* 29:2373–2379. <https://doi.org/10.3233/IFS-151936>
- Lin C-W, Hong T-P, Lin T-C, Pan S-T (2015) An ubmffp tree for mining multiple fuzzy frequent itemsets. *Int J Uncert Fuzz Knowl-Based Syst* 23:861–879. <https://doi.org/10.1142/S0218488515500385>
- Lin C-W, Hong T-P, Yang K-T, Wang S-L (2015) The ga-based algorithms for optimizing hiding sensitive itemsets through transaction deletion. *Appl Intell* 42(2):210–230. <https://doi.org/10.1007/s10489-014-0590-5>
- Lin JC-W, Gan W, Fournier-Viger P, Hong T-P, Zhan J (2016) Efficient mining of high-utility itemsets using multiple minimum utility thresholds. *Knowl-Based Syst* 113(C):100–115. <https://doi.org/10.1016/j.knosys.2016.09.013>
- Lin JC-W, Yang L, Fournier-Viger P, Wu JM-T, Hong T-P, Wang LS-L, Zhan J (2016) Mining high-utility itemsets based on particle swarm optimization. *Eng Appl Artif Intell* 55(C):320–330. <https://doi.org/10.1016/j.engappai.2016.07.006>
- Lin JC-W, Gan W, Fournier-Viger P, Hong T-P, Chao H-C (2017) Mining weighted frequent itemsets without candidate generation in uncertain databases. *Int J Inf Technol Decis Making* 16(06):1549–1579. <https://doi.org/10.1142/S0219622017500341>
- Lindell Y, Pinkas B (2000) Privacy preserving data mining. In: Annual international cryptology conference, Springer, pp 36–54
- Liu Y, Liao W-K, Choudhary A (2005) A two-phase algorithm for fast discovery of high utility itemsets. In: Ho TB, Cheung D, Liu H (eds) *Advances in knowledge discovery and data mining*. Springer, Berlin, Heidelberg, pp 689–695
- Liu Q, Ge Y, Li Z, Chen E, Xiong H (2011) Personalized travel package recommendation. In: 2011 IEEE 11th international conference on data mining, pp 407–416. <https://doi.org/10.1109/ICDM.2011.118>
- Liu B, Hsu W, Ma Y (1999) Mining association rules with multiple minimum supports. In: Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining, KDD '99, association for computing machinery, New York, NY, USA, p 337–341. <https://doi.org/10.1145/312129.312274>
- Liu M, Qu J (2012) Mining high utility itemsets without candidate generation. In: Proceedings of the 21st ACM international conference on information and knowledge management, CIKM '12, association for computing machinery, New York, NY, USA, p 55–64. <https://doi.org/10.1145/2396761.2396773>
- Li Y, Wang S, Pan Q, Peng H, Yang T, Cambria E (2019) Learning binary codes with neural collaborative filtering for efficient recommendation systems. *Knowl-Based Syst* 172:64–75. <https://doi.org/10.1016/j.knosys.2019.02.012>. <https://www.sciencedirect.com/science/article/pii/S0950705119300735>
- Li X, Yin M (2016) A particle swarm inspired cuckoo search algorithm for real parameter optimization. *Soft Comput* 20:1389–1413. <https://doi.org/10.1007/s00500-015-1594-8>. <https://link.springer.com/article/10.1007/s00500-015-1594-8>
- Ma X, Liu F, Qi Y, Wang X, Li L, Jiao L, Yin M, Gong M (2016) A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Trans Evol Comput* 20(2):275–298. <https://doi.org/10.1109/TEVC.2015.2455812>
- Menhas MI, Fei M, Wang L, Fu X (2011) A novel hybrid binary pso algorithm. *ICSI'11*. Springer, Berlin, Heidelberg, pp 93–100
- Miettinen K (2012) Nonlinear multiobjective optimization, Vol 12, Springer
- Minai-Bidgoli B, Barmaki R, Nasiri M (2013) Mining numerical association rules via multi-objective genetic algorithms. *Inf Sci* 233:15–24. <https://doi.org/10.1016/j.ins.2013.01.028>. <https://www.sciencedirect.com/science/article/pii/S0020025513001072>
- Mirjalili S (2015) Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl-Based Syst* 89:228–249. <https://doi.org/10.1016/j.knosys.2015.07.006>. <https://www.sciencedirect.com/science/article/pii/S0950705115002580>
- Mirjalili S, Dong J, Song, Sadiq AS, Faris H (2020) Genetic algorithm: theory, literature review, and application in image reconstruction. Springer, Cham, pp 69–85. https://doi.org/10.1007/978-3-030-12127-3_5
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>. <https://www.sciencedirect.com/science/article/pii/S0956997813001853>
- Mukhopadhyay A, Maulik U, Bandyopadhyay S, Coello CAC (2014) A survey of multiobjective evolutionary algorithms for data mining: part i. *IEEE Trans Evol Comput* 18(1):4–19. <https://doi.org/10.1109/TEVC.2013.2290086>
- Mukhopadhyay A, Maulik U, Bandyopadhyay S (2014) A survey of multiobjective evolutionary clustering. *ACM Comput Surv* 47(4). <https://doi.org/10.1145/2742642>
- Najafabadi MK, Mohamed AH, Mahrin MN (2019) A survey on data mining techniques in recommender systems. *Soft Comput* 23(2):627–654. <https://doi.org/10.1007/s00500-017-2918-7>. <https://link.springer.com/article/10.1007/s00500-017-2918-7>
- Nouaouria N, Boukadoum M, Proulx R (2013) Particle swarm classification: a survey and positioning. *Pattern Recog* 46(7):2028–2044. <https://doi.org/10.1016/j.patcog.2012.12.011>. <https://www.sciencedirect.com/science/article/pii/S0031320313000022>
- Oliveira SRM, Zaiane OR (2002) Privacy preserving frequent itemset mining. In: Proceedings of the IEEE international conference on

- privacy, security and data mining, Vol 14, CRPIT '14, Australian Computer Society, Inc., AUS, p 43–54
- Ouyang W, Huang Q (2011) Mining direct and indirect fuzzy association rules with multiple minimum supports in large transaction databases. In: 2011 eighth international conference on fuzzy systems and knowledge discovery (FSKD), Vol 2, pp 947–951. <https://doi.org/10.1109/FSKD.2011.6019747>
- Ozturk C, Hancer E, Karaboga D (2015) A novel binary artificial bee colony algorithm based on genetic operators. *Inf Sci* 297(C):154–170. <https://doi.org/10.1016/j.ins.2014.10.060>
- Park DH, Kim HK, Choi IY, Kim JK (2012) A literature review and classification of recommender systems research. *Exp Syst Appl* 39(11):10059–10072. <https://doi.org/10.1016/j.eswa.2012.02.038>. <https://www.sciencedirect.com/science/article/pii/S0957417412002825>
- Pazhaniraja SSN, Kumar BS (2020) High utility itemset mining: a boolean operators-based modified grey wolf optimization algorithm. *Soft Comput* 24(21):16691–16704. <https://doi.org/10.1007/s00500-020-05123-z>. <https://link.springer.com/article/10.1007/s00500-020-05123-z>
- Pazhaniraja N, Sountharajan S (2021) High utility itemset mining using dolphin echolocation optimization. *J Amb Intell Human Comput* 12:8413–8426. <https://doi.org/10.1007/s12652-020-02571-1>. <https://link.springer.com/article/10.1007/s12652-020-02571-1>
- Pears R, Koh YS (2011) Weighted association rule mining using particle swarm optimization. In: Proceedings of the 15th international conference on new frontiers in applied data mining, PAKDD'11, Springer, Berlin, Heidelberg, p 327–338. https://doi.org/10.1007/978-3-642-28320-8_28
- Pillai J, Vyas OP, Mueyba M (2013) Huri—a novel algorithm for mining high utility rare itemsets. In: Meghanathan N, Nagamalai D, Chaki N (eds) *Adv Comput Inf Technol*. Springer, Berlin Heidelberg, pp 531–540
- Pillai J, Vyas OP CSHURI—modified HURI algorithm for customer segmentation and transaction profitability. *CoRR arXiv:1205.1609*
- Ponsich A, Jaimes AL, Coello CAC (2013) A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Trans Evol Comput* 17(3):321–344. <https://doi.org/10.1109/TEVC.2012.2196800>
- Projection CIQDM (1996)
- Qu J-F, Liu M, Viger P Fournier (2019) Efficient algorithms for high utility itemset mining without candidate generation, pp 131–160. https://doi.org/10.1007/978-3-030-04921-8_5
- Ross Quinlan R, Ghosh J, Yang Q, Motoda H, McLachlan GJ, Ng A, Liu B, Yu PS, Zhou Z-H, Steinbach M, Hand DJ, Wu X, Kumar V, Steinberg D (2008) Top 10 algorithms in data mining. *Knowl Inf Syst* 14(1):1–37. <https://doi.org/10.1007/s10115-007-0114-2>. <https://link.springer.com/article/10.1007/s10115-007-0114-2>
- Salleb A, Vrain C, Nortet C (2007) Quantminer: a genetic algorithm for mining quantitative association rules, Vol 7, pp 1035–1040
- Sarath K, Ravi V (2013) Association rule mining using binary particle swarm optimization. *Eng Appl Artif Intell* 26(8):1832–1840. <https://doi.org/10.1016/j.engappai.2013.06.003>. <https://www.sciencedirect.com/science/article/pii/S0952197613001048>
- Seidlova R, Poživil J, Seidl J (2019) Marketing and business intelligence with help of ant colony algorithm. *J Strategic Mark* 27(5):451–463. <https://doi.org/10.1080/0965254X.2018.1430058>
- Simões A, Costa E (1999) Transposition: a biological-inspired mechanism to use with genetic algorithms. In: Dobnikar A, Steele NC, Pearson DW, Albrecht RF (Eds.), *Proceedings of the international conference on artificial neural nets and genetic algorithms, ICANNGA 1999, Portorož, Slovenia, 1999*, Springer, pp 178–186. https://doi.org/10.1007/978-3-7091-6384-9_31
- Singh SS, Kumar A, Singh K, Biswas B (2020) Im-ssso: maximizing influence in social networks using social spider optimization. *Concurr Comput Pract Exper* 32(2):e5421. <https://doi.org/10.1002/cpe.5421>
- Singh K, Shakya HK, Singh A, Biswas B (2018) Mining of high-utility itemsets with negative utility. *Exp Syst* 35(6):e12296–e12296. <https://doi.org/10.1111/exsy.12296>. <https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12296>
- Song W, Huang C (2018) Mining high utility itemsets using bio-inspired algorithms: a diverse optimal value framework. *IEEE Access* 6:19568–19582. <https://doi.org/10.1109/ACCESS.2018.2819162>
- Song W, Li J (2020) Discovering high utility itemsets using set-based particle swarm optimization. In: Yang X, Wang C-D, Islam MS, Zhang Z (eds) *Advanced data mining and applications*. Springer, Cham, pp 38–53
- Song W, Yang B, Xu Z (2008) Index-bittablefi: an improved algorithm for mining frequent itemsets. *Knowl-Based Syst* 21:507–513. <https://doi.org/10.1016/j.knosys.2008.03.011>
- Song W, Liu Y, Li J (2014) Bahui: fast and memory efficient mining of high utility itemsets based on bitmap. *Int J Data Warehous Min* 10(1):1–15. <https://doi.org/10.4018/ijdw.2014010101>
- Song A, Ding X, Chen J, Li M, Cao W, Pu K (2016) Multi-objective association rule mining with binary bat algorithm. *Intell Data Anal* 20:105–128. <https://doi.org/10.1023/IDA-150796>
- Song W, Li J, Huang C (2021) Artificial fish swarm algorithm for mining high utility itemsets. In: Tan Y, Shi Y (eds) *Advances in swarm intelligence*. Springer, Cham, pp 407–419
- Song W, Huang C (2018) Discovering high utility itemsets based on the artificial bee colony algorithm, pp 3–14. https://doi.org/10.1007/978-3-319-93040-4_1
- Soto R, Crawford B, Carrasco C, Almonacid B, Reyes V, Araya I, Misra S, Olguín E (2016) Solving manufacturing cell design problems by using a dolphin echolocation algorithm. In: Misra S, Gervasi O, Torre C, Taniar D, Stankova E, Rocha A, Wang S, Murgante B, Apduhan B (eds) *Computational science and its applications—16th international conference, ICCSA 2016, proceedings, lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture Notes in Bioinformatics)*, Springer, pp 77–86, publisher Copyright: Springer, Switzerland 2016.; null; conference date: 04-07-2016 Through 07-07-2016. https://doi.org/10.1007/978-3-319-42092-9_7
- Srikant R (1999) Ibm quest synthetic data generation code, not available online anymore
- Srivastava G, Lin JC-W, Pirouz M, Li Y, Yun U (2021) A pre-large weighted-fusion system of sensed high-utility patterns. *IEEE Sens J* 21(14):15626–15634. <https://doi.org/10.1109/JSEN.2020.2991045>
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359. <https://doi.org/10.1023/A:1008202821328>. <https://link.springer.com/article/10.1023/A:1008202821328>
- Tang L, Zhang L, Luo P, Wang M (2012) Incorporating occupancy into frequent pattern mining for high quality pattern recommendation. In: *Proceedings of the 21st ACM international conference on information and knowledge management, CIKM '12, ssociation for Computing Machinery, New York, NY, USA, p 75–84*. <https://doi.org/10.1145/2396761.2396775>
- Telikani A, Gandomi AH, Shahbahrami A (2020) A survey of evolutionary computation for association rule mining. *Inf Sci* 524:318–352. <https://doi.org/10.1016/j.ins.2020.02.073>. <https://www.sciencedirect.com/science/article/pii/S002002552030164X>
- Tian Y, Zhang X, Wang C, Jin Y (2020) An evolutionary algorithm for large-scale sparse multiobjective optimization problems. *IEEE Trans Evol Comput* 24(2):380–393. <https://doi.org/10.1109/TEVC.2019.2918140>
- Tian D, Gledson A, Antoniadis A, Aristodimou A, Dimitrios N, Sahay R, Pan J, Stivaros S, Nenadic G, Zeng X-j, Keane J (2013)

- A bayesian association rule mining algorithm. In: 2013 IEEE international conference on systems, man, and cybernetics, pp 3258–3264. <https://doi.org/10.1109/SMC.2013.555>
- Tsai C-S, Chun-Wei, Huang K-W, Chiang M-C (2015) A fast particle swarm optimization for clustering. *Soft Comput* 19(2):321–338. <https://doi.org/10.1007/s00500-014-1255-3>. <https://link.springer.com/article/10.1007/s00500-014-1255-3>
- Tseng VS, Wu C-W, Shie B-E, Yu PS (2010) Up-growth: An efficient algorithm for high utility itemset mining. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '10, association for computing machinery, New York, NY, USA, p 253–262. <https://doi.org/10.1145/1835804.1835839>
- Tseng VS, Shie B-E, Wu C-W, Yu PS (2013) Efficient algorithms for mining high utility itemsets from transactional databases. *IEEE Trans Knowl Data Eng* 25(8):1772–1786. <https://doi.org/10.1109/TKDE.2012.59>
- Tseng VS, Wu C-W, Fournier-Viger P, Yu PS (2015) Efficient algorithms for mining the concise and lossless representation of high utility itemsets. *IEEE Trans Knowl Data Eng* 27(3):726–739. <https://doi.org/10.1109/TKDE.2014.2345377>
- Venkatadri M, Rao KS (2010) A multiobjective genetic algorithm for feature selection in data mining. *Int J Comput Sci Inf Technol* 1(5):443–448
- Ventresca M, Harrison KR, Ombuki-Berman BM (2018) The bi-objective critical node detection problem. *Eur J Oper Res* 265(3):895–908. <https://doi.org/10.1016/j.ejor.2017.08.053>. <https://www.sciencedirect.com/science/article/pii/S0377221717307841>
- Ventura S, Luna JM (2016) Pattern mining with evolutionary algorithms, 1st edn., Springer. <https://doi.org/10.1007/978-3-319-33858-3>. <https://www.springer.com/gp/book/9783319338576>
- Verykios VS, Bertino E, Fovino IN, Provenza LP, Saygin Y, Theodoridis Y (2004) State-of-the-art in privacy preserving data mining. *SIGMOD Rec.* 33(1):50–57. <https://doi.org/10.1145/974121.974131>
- Vu KK, d'Ambrosio C, Hamadi Y, Liberti L (2017) Surrogate-based methods for black-box optimization. *Int Trans Oper Res* 24(3):393–424
- Wang C-M, Chen S-H, Huang Y-F (2009) A fuzzy approach for mining high utility quantitative itemsets. In: 2009 IEEE international conference on fuzzy systems, pp 1909–1913. <https://doi.org/10.1109/FUZZY.2009.5277408>
- Wang J-S, Li S-X (2019) An improved grey wolf optimizer based on differential evolution and elimination mechanism. *Sci Rep* 9(1):1–21, article number: 7181. <https://doi.org/10.1038/s41598-019-43546-3>. <https://www.nature.com/articles/s41598-019-43546-3>
- Wang Z, Hutter F, Zoghi M, Matheson D, De Freitas N (2016) Bayesian optimization in a billion dimensions via random embeddings. *J Artif Int Res* 55(1):361–387
- Wang Z, Zhang Q, Zhou A, Gong M, Jiao L (2016) Adaptive replacement strategies for moea/d. *IEEE Trans Cybern* 46(2):474–486. <https://doi.org/10.1109/TCYB.2015.2403849>
- Wenyn GZC, Ling CX (2010) De/bbo: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. *Soft Comput* 15:645–665. <https://doi.org/10.1007/s00500-010-0591-1>. <https://link.springer.com/article/10.1007/s00500-010-0591-1>
- Wu JM-T, Lin J Chun-Wei, Djenouri Y, Fournier-Viger P, Zhang Y (2019) A swarm-based data sanitization algorithm in privacy-preserving data mining. In: 2019 IEEE congress on evolutionary computation (CEC), pp 1461–1467. <https://doi.org/10.1109/CEC.2019.8790271>
- Wu JM-T, Lin JC-W, Fournier-Viger P, Wiktorski T, Hong T-P, Pirouze M (2019) A ga-based framework for mining high fuzzy utility itemsets. In: 2019 IEEE international conference on big data (big data), pp 2708–2715. <https://doi.org/10.1109/BigData47090.2019.9006171>
- Wu JM-T, Zhan J, Lin JC-W (2016) Mining of high-utility itemsets by aco algorithm. In: Proceedings of the the 3rd multidisciplinary international social networks conference on social informatics 2016, data science 2016, MISNC, SI, DS 2016, Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/2955129.2955179>
- Wu T-Y, Lin JC-W, Zhang Y, Chen C-H. A grid-based swarm intelligence algorithm for privacy-preserving data mining. *Appl Sci* 9(4). <https://doi.org/10.3390/app9040774>. <https://www.mdpi.com/2076-3417/9/4/774>
- Wu JM-T, Zhan J, Lin JC-W (2017) An aco-based approach to mine high-utility itemsets. *Know -Based Syst* 116(C):102–113. <https://doi.org/10.1016/j.knosys.2016.10.027>
- Wu C, He Y (2020) Solving the set-union knapsack problem by a novel hybrid jaya algorithm. *Soft Comput* 24(3):1883–1902. <https://doi.org/10.1007/s00500-019-04021-3>. <https://link.springer.com/article/10.1007/s00500-019-04021-3>
- Xiao S, Hu Y, Han J, Zhou R, Wen J (2016) Bayesian networks-based association rules and knowledge reuse in maintenance decision-making of industrial product-service systems. *Procedia CIRP* 47:198–203. <https://doi.org/10.1016/j.procir.2016.03.046>
- Xue B, Zhang M, Browne WN (2013) Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Trans Cybern* 43(6):1656–1671. <https://doi.org/10.1109/TSMCB.2012.2227469>
- Yang X-S (2010) Engineering optimization: an introduction with metaheuristic applications. Wiley. <https://doi.org/10.1002/9780470640425>
- Yang X-S (2011) Bat algorithm for multi-objective optimisation. *Int J Bio-Inspired Comput* 3(5):267–274. <https://doi.org/10.1504/IJIBC.2011.042259>
- Yang R, Xu M, Jones P, Samatova N (2017) Real time utility-based recommendation for revenue optimization via an adaptive online top-k high utility itemsets mining model. In: 2017 13th international conference on natural computation, fuzzy systems and knowledge discovery (ICNC-FSKD), pp 1859–1866. <https://doi.org/10.1109/FSKD.2017.8393050>
- Yao H, Hamilton HJ, Butz CJ A foundational approach to mining item-set utilities from databases, pp 482–486. <https://doi.org/10.1137/1.9781611972740.51>. <https://epubs.siam.org/doi/abs/10.1137/1.9781611972740.51>
- Yao H, Hamilton HJ (2006) Mining itemset utilities from transaction databases. *Data Knowl Eng* 59(3):603–626, including: ER 2003. <https://doi.org/10.1016/j.datak.2005.10.004>. <https://www.sciencedirect.com/science/article/pii/S0169023X05001643>
- Yeh J-S, Chang C-Y, Wang Y-T (2008) Efficient algorithms for incremental utility mining. In: Proceedings of the 2nd international conference on ubiquitous information management and communication, ICUIMC '08, Association for Computing Machinery, New York, NY, USA, p 212–217. <https://doi.org/10.1145/1352793.1352839>
- Yeh J-S, Hsu P-C (2010) Hhuif and msicf: novel algorithms for privacy preserving utility mining. *Expert Syst Appl* 37(7):4779–4786. <https://doi.org/10.1016/j.eswa.2009.12.038>
- Yun U, Kim J (2015) A fast perturbation algorithm using tree structure for privacy preserving utility mining. *Expert Syst Appl* 42(3):1149–1165. <https://doi.org/10.1016/j.eswa.2014.08.037>
- Zadeh L (1975) The concept of a linguistic variable and its application to approximate reasoning-i. *Inf Sci* 8(3):199–249. [https://doi.org/10.1016/0020-0255\(75\)90036-5](https://doi.org/10.1016/0020-0255(75)90036-5). <https://www.sciencedirect.com/science/article/pii/0020025575900365>
- Zaki MJ (2000) Scalable algorithms for association mining. *IEEE Trans Knowl Data Eng* 12(3):372–390. <https://doi.org/10.1109/69.846291>

- Zhang SW, Yudong, Ji G (2013) A rule-based model for bankruptcy prediction based on an improved genetic ant colony algorithm. *Mathematical Prob Eng*, article ID 753251. <https://doi.org/10.1155/2013/753251>. <https://www.hindawi.com/journals/mpe/2013/753251/>
- Zhang SW, Yudong, Ji G (2015) A comprehensive survey on particle swarm optimization algorithm and its applications. *Math Prob Eng*, article ID 931256. <https://doi.org/10.1155/2015/931256>
- Zhang X, Tian Y, Jin Y (2015) A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 19(6):761–776. <https://doi.org/10.1109/TEVC.2014.2378512>
- Zhang X, Tian Y, Jin Y (2016) Approximate non-dominated sorting for evolutionary many-objective optimization. *Inf Sci* 369(C):14–33. <https://doi.org/10.1016/j.ins.2016.06.007>
- Zhang X, Duan F, Zhang L, Cheng F, Jin Y, Tang K (2017) Pattern recommendation in task-oriented applications: a multi-objective perspective [application notes]. *IEEE Comput Intell Magaz* 12(3):43–53. <https://doi.org/10.1109/MCI.2017.2708578>
- Zhang X, Tian Y, Cheng R, Jin Y (2018) A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. *IEEE Trans Evol Comput* 22(1):97–112. <https://doi.org/10.1109/TEVC.2016.2600642>
- Zhang Y, Wu J, Wang H (2019) Neural binary representation learning for large-scale collaborative filtering. *IEEE Access* 7:60752–60763. <https://doi.org/10.1109/ACCESS.2019.2915331>
- Zhang Q, Fang W, Sun J, Wang Q (2019) Improved genetic algorithm for high-utility itemset mining. *IEEE Access* 7:176799–176813. <https://doi.org/10.1109/ACCESS.2019.2958150>
- Zhang L, Fu G, Cheng F, Qiu J, Su Y (2018) A multi-objective evolutionary approach for mining frequent and high utility itemsets. *Appl Soft Comput* 62:974–986. <https://doi.org/10.1016/j.asoc.2017.09.033>. <https://www.sciencedirect.com/science/article/pii/S1568494617305720>
- Zhang L, Luo P, Chen E, Wang M (2016) Revisiting bound estimation of pattern measures: a generic framework. *Inf Sci* 339:254–273. <https://doi.org/10.1016/j.ins.2015.12.036>. <https://www.sciencedirect.com/science/article/pii/S0020025516000190>
- Zheng Z, Kohavi R, Mason L (2001) Real world performance of association rule algorithms. In: *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining, KDD '01*, Association for Computing Machinery, New York, NY, USA, p 401–406. <https://doi.org/10.1145/502512.502572>
- Zida S, Fournier-Viger P, Lin JC-W, Wu C-W, Tseng VS (2015) Efim: a highly efficient algorithm for high-utility itemset mining. In: *Sidorov G, Galicia-Haro SN (eds) Advances in artificial intelligence and soft computing*. Springer, Cham, pp 530–546
- Zille H, Ishibuchi H, Mostaghim S, Nojima Y (2018) A framework for large-scale multiobjective optimization based on problem transformation. *IEEE Trans Evol Comput* 22(2):260–275. <https://doi.org/10.1109/TEVC.2017.2704782>
- Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—a comparative case study. In: *Eiben AE, Bäck T, Schoenauer M, Schwefel H-P (eds) Parallel Prob Solv Nature PPSN V*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 292–301
- Zitzler E, Laumanns M, Thiele L *Spea2: improving the strength pareto evolutionary algorithm*, TIK-report 103
- Zuo Y, Gong M, Zeng J, Ma L, Jiao L (2015) Personalized recommendation based on evolutionary multi-objective optimization [research frontier]. *Comp Intell Mag* 10(1):52–62. <https://doi.org/10.1109/MCI.2014.2369894>

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