



Research article

Multi-objective Boolean grey wolf optimization based decomposition algorithm for high-frequency and high-utility itemset mining

N. Pazhaniraja¹, Shakila Basheer², Kalaipriyan Thirugnanasambandam³, Rajakumar Ramalingam⁴, Mamoon Rashid^{5,*} and J. Kalaivani⁶

¹ Department of Artificial Intelligence and Machine Learning, Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India

² Department of Information Systems, College of Computer and Information Science, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

³ Centre for Smart Grid Technologies, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India

⁴ School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India

⁵ Department of Computer Engineering, Faculty of Science and Technology, Vishwakarma University, Pune, 411048, India

⁶ Department of Computing Technologies, SRMIST, Kattankulathur, Chennai, India

* **Correspondence:** Email: mamoon873@gmail.com.

Abstract: In itemset mining, the two vital goals that must be resolved from a multi-objective perspective are frequency and utility. To effectively address the issue, researchers have placed a great deal of emphasis on achieving both objectives without sacrificing the quality of the solution. In this work, an effective itemset mining method was formulated for high-frequency and high-utility itemset mining (HFUI) in a transaction database. The problem of HFUI is modeled mathematically as a multi-objective issue to handle it with the aid of a modified bio-inspired multi-objective algorithm, namely, the multi-objective Boolean grey wolf optimization based decomposition algorithm. This algorithm is an enhanced version of the Boolean grey wolf optimization algorithm (BGWO) for handling multi-objective itemset mining problem using decomposition factor. In the further part of this paper decomposition factor will be mentioned as decomposition. Different population initialization strategies were used to test the impact of the proposed algorithm. The system was evaluated with 12 different real-time datasets, and the results were compared with seven different recent existing multi-objective

models. Statistical analysis, namely, the Wilcoxon signed rank test, was also utilized to prove the impact of the proposed algorithm. The outcome shows the impact of the formulated technique model over other standard techniques.

Keywords: Boolean operators; Grey Wolf Optimization; Frequency and Utility mining; Swarm intelligence; multi-objective algorithm

Mathematics Subject Classification: 05C85, 78M50

1. Introduction

In data mining, frequent itemset mining is the process of retrieving a frequently used itemset from a transaction database to determine the relationship among the individual items to improve productivity. For example, in a supermarket database, identifying frequently purchased combinations of items will lead the investor to buy more stock of the items together and thus improve the supplies, which will result in more profit. This identification also helps to promote the itemset with attractive offers, so that the likelihood of purchase in a shop will be increased significantly. Due to this advantage, a significant number of researchers have contributed to frequent itemset mining to find the best combined itemsets [1].

In general, an itemset can be termed as a frequent itemset if and only if its appearance in the database is not less than the user specified value (min_{freq}). However, considering only the frequency of an itemset does not lead to an optimal discovery of the itemset which gives us the most profit. Users who check for more profit may feel that frequency based mining alone is not that effective. Utility is another consideration for improving profits, as used in what is usually called high-utility itemset mining (HUIM) [2], which counts the profit achieved through every itemset in the transaction database. In this utility mining, every item has a profit value and a number of occurrences in the database. An itemset can be termed as a high-utility itemset if and only if its profit in the database is not less than the user specified value (min_{util}).

Most of the studies on itemset mining consider either frequency or utility. In other words, the two factors are considered as separate objectives. With this perspective, if an itemset needs to be discovered with high frequency and high utilization, a set of solutions which cross the minimum threshold for frequency must first be discovered, and then the minimum threshold value of utility will be applied to obtain an itemset. With this procedure, there may be a possibility of obtaining empty sets if the transaction database is very large. Also, applying an algorithm twice to solve a problem is not computationally efficient. For example, an itemset with high-frequency items may have a very low utility value, and vice versa. Hence, addressing this as a multi-objective optimization problem is worthy.

Evolutionary algorithms, based on evolution in nature, perform well in terms of solving optimization algorithms in multi-objective solution space. Algorithms such as the genetic algorithm [3], particle swarm optimization [4], the ant colony optimization algorithm [5], the cuckoo search algorithm [6], etc. are prominent evolutionary methods for solving optimization problems. This work extends the work of Pazhaniraja et al. (2020), which sought to perform high utility itemset mining with the Boolean grey wolf optimization algorithm (BGWO) [7]. In this work, the BGWO is modeled to solve the multi-objective optimization problem of high-frequency and high-utility mining using decomposition factor.

The grey wolf optimization algorithm (GWO) organizes the roles in the wolf pack according to the pack hierarchy [8]. GWO is used in many optimization tasks since it has fewer constraints and does not need imitative knowledge in the first search. This inventive technique is simple and accomplishes a balanced diversification and intensification procedure over the search area [9]. GWO has quickly garnered a broad research audience from many fields. Recently, a multi-objective GWO (MOGWO) was developed to solve optimization problems with various and competing objectives [10]. MOGWO integrates a static auxiliary archive for preserving and retrieving Pareto optimal solutions. The societal and hunting behaviors of grey wolves are then simulated using this algorithm. Even though MOGWO has outperformed other multi-objective optimizers, there remains potential for further development.

The contributions of this research work are as follows:

- An effective multi-objective Boolean grey wolf optimization algorithm (MOBGWO) has been designed and developed to handle binary-based multi-objective optimization problems.
- Effective Boolean operators are designed to handle the generation of feasible solutions that include stochastic randomness and uncertainty in solution generation to maintain non-determinism.
- The proposed MOBGWO model is utilized for HFUI from a multi-objective perspective considering frequency and utility of the itemset.

The proposed algorithm has been analyzed by evaluating it with different sizes of datasets, from small to extremely large datasets. The formulated MOBGWO model is compared with other standard techniques to prove the significance of the proposed algorithm. The motivating factor for proposing a multi-objective optimization algorithm is the “No Free Lunch Theorem” proposed by Wolpert, D. H., and Macready, W. G., in 1997 [11]. As per Wolpert’s theorem, there exists no single algorithm that can solve all optimization problems. When we compare the performances of all optimization algorithms, the performance averages will be equal to one another. This is the base motivation of our proposed algorithm. However, there are a significant number of approaches for solving multi-objective optimization problems, and every individual problem has its own features to be handled which are not generic in nature. Our proposed algorithm addresses one such aspect of itemset mining in this research work.

The rest of the work is structured as follows: Section 2 discusses related studies on recent algorithms that have been proposed for HFUI. Section 3 illustrates the multi-objective problem formulation of HFUI. Section 4 discusses the proposed multi-objective Boolean grey wolf optimization based decomposition algorithm for performing HFUI. Section 5 deals with the experimental evaluation of the formulated model, and the final section concludes the paper with future directions.

2. Literature survey

In 1994, Agrawal and Srikant formulated the frequent itemset mining problem [12] and solved it using the Apriori algorithm. Since then, several approaches have been framed and proposed, such as Partition-1 [13], FP-Growth [14], Eclat [15] and KDCI [16]. These techniques are effective only for mining that does not care about profit, since high frequency may not always lead to high profit. Hence, in 2004, Huo et al. [17] formulated utility-based itemset mining, through which they discovered a high-utility itemset from the transaction database. In a proposal by Lie et al. [18], the items are pruned to reduce the cost of computation, and the method uses the transaction-weight utility property. Since then, several approaches have been proposed [19–23] for performing high-utility itemset mining. When

there exists a greater number of itemset on the perspective of either frequent or utility there exists a problem which are all to be preserved. To address this problem, an effective top k high utility itemset mining algorithm was proposed which utilizes a threshold value [24,25], and some open-source software tools were also proposed [26].

The approaches of traditional algorithms are effective, but the exhaustive search-based models are highly time-consuming processes. Meanwhile, stochastic optimization models using evolutionary techniques are intended to solve the optimization issues in appropriate ways with computational efficiency. The first such approach for HUIM is the genetic algorithm [27], which uses a ranked mutation model for mining high-utility itemsets (HUIs). Using particle swarm optimization, the discrete particle swarm optimization algorithm was proposed for HUIM, and some of the same authors later proposed a binary version of the particle swarm optimization algorithm for mining HUIs [28,29]. Another model, called an ant colony system [30], has been proposed for HUIM, in which processes such as the two-way pruning model were imposed for effective handling of a transaction database for the mining process.

In a similar way, several approaches have been presented for performing HUIM and high-frequency and high-utility mining. Evolutionary algorithms are not only intended to solve mining problems but are also used in other research domains such as mathematical benchmark functions and many more. Other applications can be found in [31–34]. The intention of this research is to propose an effective multi-objective model for performing HFUI.

The choice of GWO has been made based on its simplicity and the wide usage of the algorithm for solving various applications in different domains from both single-objective and multi-objective perspectives. Some of the applications include a routing problem in a dual supply chain which considers both the price and the duration of delivery from a case study on a construction material supplier in 2022 by Abbaspour et al. [35]. In 2022, Asgari et al. proposed a multi-objective variant of GWO for analyzing a solar based tri-generation system [36]. In 2022, Hasanzadeh et al. proposed another multi-objective variant of GWO for solving the combination of solid oxide fuel cell with other hybrid and standalone gas turbines [37]. However, based on the “No Free Lunch Theorem” [7], our proposed algorithm is developed to perform multi-objective HFUI.

In 2022, Xuan Liu, Genlang Chen and Wanli Zuo [38] proposed the EMSFUI-D and EMSFUI-B algorithms to identify high utility itemsets. These algorithms holds two different pruning mechanisms to rule out the least utilized items without considering them for fitness evaluation. High average-utility itemset mining involves evaluating a quantitative consumer transactional database in order to discover high average-utility itemsets, or groupings of items with high average usefulness. In 2022, Le. B. et al. [39] proposed to mine frequent HAUI that involves evaluating a quantitative consumer transactional database in order to discover high average-utility itemsets, or groupings of items with high average usefulness. In 2023, Luna et al. [40] proposed top- k high utility itemset mining through genetic algorithms for performing high utility itemset extraction. It directs the search process by evaluating the usability of each item to provide initial solutions and combines solutions appropriately, hence lowering runtime and memory usage. However, the effective mechanism to identify high utility itemset mining is still in debate, and, as per the statement of the No Free Lunch theorem, the proposed algorithm is being used for identifying the high utility itemset from a different objective’s perspective.

3. System design

In this section a detailed description of frequency and utility itemset mining is given along with an example model. Based on this problem definition, the objectives of the problem are also given in this section.

3.1. Problem formulation of high frequency and high utility itemset mining

Let us consider Table 1 as a transaction database (D) consisting of eight transactions $T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$, each with a unique id ranging from 1 to 8. In general, a transaction database consists of m finite sets of transactions $T = \{t_1, t_2, \dots, t_m\}$ in which each transaction has a varied number of items in it. As a single set, the number of distinct items (n) in all the transactions is represented as $I = \{i_1, i_2, \dots, i_n\}$. A transaction can be named as a supporting transaction when a set of items (X) is presented in that transaction. The supporting transactions of the itemset is the number of overall transactions that contains the itemset X , and it can be represented as D_X . The itemset X is a frequent itemset if $freq(X) \geq min_{sup}$, where min_{sup} is the threshold value specified by the user, and $freq(X)$ denotes the total number of transactions that contain itemset X .

Table 1. Transaction table.

t_{id}	Transaction	$W(t_{id})$
t_1	$\{(a, 3), (b, 4), (c, 4), (d, 7)\}$	18
t_2	$\{(a, 8), (b, 13), (c, 3), (e, 201)\}$	225
t_3	$\{(a, 4), (b, 7), (d, 3), (f, 2), (h, 2)\}$	18
t_4	$\{(a, 2), (d, 4), (g, 6)(i, 2), (j, 1)\}$	15
t_5	$\{(c, 8), (d, 8), (e, 5), (h, 1), (i, 1)\}$	23
t_6	$\{(a, 8), (b, 6), (i, 2)\}$	16
t_7	$\{(c, 3), (d, 2), (h, 1)\}$	6
t_8	$\{(a, 9), (b, 12), (i, 4)(j, 2)\}$	27

In Table 1, the items are represented as pairs of item names and numbers. For example, in transaction t_1 , the term $(a, 3)$ denotes that the item a is utilized 3 times in transaction t_1 . Likewise, in every transaction, each item is specified with a utilization value of the item which is used in that transaction, t_{id} . The overall efficacy of each transaction is given as $W(t_{id})$ in the last column of Table 1.

In Table 1, there are 10 distinct items in the transaction database (D) $I = \{a, b, c, d, e, f, g, h, i, j\}$ and eight transactions $T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$. For example, let us consider an itemset $X = \{a, b\}$. The supporting transactions of itemset X in database D are $D_X = \{t_1, t_2, t_3, t_6, t_8\}$, and thus $freq(X) = \frac{|D_X|}{|T|}$, which is $\frac{5}{8}$ for the itemset $\{a, b\}$. The itemset X can be termed as a frequent itemset if min_{freq} is set in the interval (0 to 0.6).

The weight of an item in a transaction is represented as $W(t_{id}, i)$. For example, the weight of item a in transaction t_1 is 3. Similarly, for weight of an itemset X in a transaction t_i can be represented as $W(t_{id}, X)$. For example, the weight of itemset $X = \{a, b\}$ in the transaction t_1 is $W(t_1, a) + W(t_1, b) = 3 + 4 = 7$.

In a similar way, the utility of an itemset (X) in the entire transaction database D can be represented as follows:

$$util(X, D) = \frac{\sum_{i=id \in D_X}^{|D_X|} \sum_{X \subseteq t} W(t_i, X)}{\sum_{id=1}^{|T|} W(t_{id})}. \quad (1)$$

For example, in transaction database D , $util(\{a, b\}, D) = \frac{7+21+11+14+21}{18+225+18+15+23+16+6+27} = \frac{74}{348} =$

0.21. Similarly, $util(\{b, e\}, D)$ is 0.61. If min_{util} , through which high utility itemsets can be defined, is set to 0.5, itemset $\{b, e\}$ will be denoted as a high utility itemset, and the itemsets $\{a, b\}$ will be discarded. Similarly, the other high utility itemset in the transaction database D are identified as $util(\{a, e\}, D) = 0.60$, $util(\{c, e\}, D) = 0.62$, $util(\{a, b, e\}, D) = 0.63$, $util(\{a, c, e\}, D) = 0.60$, $util(\{b, c, e\}, D) = 0.62$ and $util(\{a, b, c, e\}, D) = 0.64$.

3.2. Multi-objective formulation for high frequency and high utility itemset mining

In high frequency and high utility itemset mining, the two factors of frequency of an itemset and its utility in the database can be observed as two different objectives in a multi-objective optimization model, as discussed here. The two factors are considered as two different objectives based on the inference as follows:

Theorem 1. In a transaction database D , a high-utility itemset may have low frequency, and a low-utility itemset may have high frequency.

Proof: From Table 1, in a transaction database D , the frequency of item $\{a\}$ is 0.8, and its utility is 0.05. In this example, the frequency of the itemset meets the user constraint when min_{freq} is set to 0.5. However, the utility does not meet the user constraint when min_{util} is set to 0.5. On the other hand, if the itemset X is $\{a, e\}$, then the frequency of X in the transaction database D is 0.2, and its utility is 0.60. Thus, the utility of the itemset X satisfies the user constraint, whereas the frequency does not, provided that $min_{freq} = min_{util} = 0.5$. Thus, the transformation of high frequency and high utility itemset mining into a multi-objective optimization is valid.

Hence, the objectives of the problem can be represented as follows:

$$Maximize F(X) = \begin{cases} \max(Freq(X)) \\ \max(Util(X)) \end{cases}. \quad (2)$$

4. Multi-objective Boolean grey wolf optimization based decomposition algorithm

In this section, detailed descriptions of the standard grey wolf optimization algorithm (GWO), the conversion of the standard GWO into the Boolean GWO and the multi-objective Boolean GWO with decomposition algorithm for performing multi-objective high frequency and high utility itemset mining are given.

4.1. Standard GWO

In the past decade, bio-inspired optimization algorithms have been applied by researchers to solve a number of significant optimization problems, including combinatorial problems such as the Traveling Salesman Problem, knapsack problems, virtual machine placement in cloud computing, etc. They have such wide applications because they possess simplicity, such as straightforward implementation irrespective of gradient information. One such optimization algorithm is the grey wolf optimization algorithm [8]. It is inspired from the social behavior of wolves, as they possess good leadership and high-level hunting strategies toward their prey. They are among the high-level predators and work in groups with sizes varying from 5 to 12 naturally.

The wolves are further classified as Alpha, Beta, Delta and Omega wolves. Alpha wolves (α) are responsible for decision making, and it is the most dominant wolves' category when compared to other categories of wolves. These wolves instruct the other wolves of the group. Alpha wolves are responsible for production of new wolves. Beta wolves (β) are the wolves with the category next to the Alpha wolves. These wolves help the Alpha wolves in planning. Beta wolves will assume leadership of the wolves' group in the absence of Alpha wolves. In the presence of Alpha wolves, these Beta wolves will act as the responding wolves for the decisions made by the Alpha wolves. In the third category are the Delta wolves (δ). These wolves are the subordinate wolves in the pack and are the next level of wolves. These wolves fall under the classifications of elders, sentinels, hunters, scouts and caregivers. As the last category of wolves, the Omega wolves (ω) are the lowest ranking wolves and play the role of scapegoat. They obey the instructions provided by all other wolves in the group.

The formation of GWO is based on tracking, chasing and approaching the prey of the wolves. The wolves perceive, encircle and harass the prey until the prey's movement stops. Then, they attack the prey in an effective manner. The GWO is effective for exploration and the exploitation phase of a search strategy in solution space. However, the standard GWO is proposed for solving continuous optimization problems such as mathematical benchmark problems, tension/compression spring design, welded beam design, etc. HFUI, on the other hand, is a binary based solution representation in which the conventional operators of standard GWO will not be very effective for generation of new solutions for the next iterations. Hence, Pazhaniraja et al. proposed a Boolean based grey wolf optimization algorithm [7] for performing HUI mining. In the next section, a description of BGWO is given in brief.

4.2. Boolean based grey wolf optimization algorithm

In this section, the Boolean based GWO is described in brief. It is highly recommended to visit the origin of this algorithm in [7]. For further proper understanding of this paper a brief note is given as follows: There are, in total, 5 Boolean operators used in BGWO which replace the conventional operators in the standard GWO for processing the HFUI problem. The representation of a solution is first given in this section, through which the further processing can be applied.

4.2.1. Solution representation

From the example of Table 1, the database D consists of eight transactions $T = \{t_1, t_2, \dots, t_8\}$. In the first transaction, out of 10 distinct items, only 4 items, namely, $\{a, b, c, d\}$, are included. This can be represented as a set $D(t_1) = [1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0]$ which shows that items $\{a, b, c, d\}$ are

included, and items $\{e, f, g, h, i, j\}$ are not included. In this way, the whole database can be represented as follows:

$$D(\text{Bit}_{Map}) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad (3)$$

where T_d denotes the row of the transaction, and $\{a, b, c, d, e, f, g, h, i, j\}$ specifies the column items.

4.2.2. De Morgan's AND (\wedge)

De Morgan's law specifies a sequence of conversion rules that includes valid rules of inference. In a De Morgan's AND gate, the resultant factor will be 1 if any input value holds 0, and the output value will be 0 if both the input values are 1. A 2-bit input model of a De Morgan's AND gate is shown in Table 2.

Table 2. Two-bit input De Morgan's AND.

Input		Output
A	B	A+B
0	0	1
1	0	1
0	1	1
1	1	0

4.2.3. Difference (\ominus)

The Boolean difference gives output in two vectors, namely, Difference and Borrow. For a 2-bit input, the borrow will return as 1 only if the first bit is 0 and the second bit is 1. In all other cases, the value of borrow will be 0. For the difference vector, the output will be one if and only if only one of the inputs is 1; otherwise, the result will be 0. Table 3 shows the values of 2-bit vectors.

Table 3. 2-bit input Difference (\ominus) gate values.

Input		Output	
A	B	Diff.	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

4.2.4. Binary Adder (\oplus)

Table 4 shows the input and output of the 2-bit input model of Binary Adder. Like the Difference operator, this Binary Adder also returns two different vectors, namely, Sum and Carry. Carry will return

1 only if both the inputs are 1. For Sum, the output will be 1 if and only if only one of the inputs is 1; otherwise, the resultant will be 0.

Table 4. 2-bit input Binary Adder (\oplus) gate values.

Input		Output	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

4.2.5. Multiplexer (with uncertainty) (\bowtie)

A multiplexer of 2^n inputs has n select lines, which are used to select which input line to send to the output. (S_0). A generalized equation for the Multiplexer is given below. In our model the multiplexer is responsible for the uncertainty factors A and C .

$$z = (A \wedge \neg S_0) \vee (B \wedge S_0). \quad (4)$$

There are two coefficient components, A and C , in the suggested model. The uncertainty factors are caused by these two components. These two factors are responsible for the stochastic randomness during the search process.

4.2.6. 2-bit circular shift (left and right) (\mapsto, \mapleftarrow)

The process of moving the last element in a tuple to the first place while shifting all other entries to the next position, or carrying out the opposite operation, is known as circular shift. A two-bit circular shift alternates between two places.

4.3. Multi-objective Boolean grey wolf optimization based decomposition algorithm for HFUI

For solving multi-objective optimization problems, the proposed Boolean GWO is integrated with a decomposition-based optimization model. In decomposition, the problem is divided into subproblems which can be solved in an effective manner. In this section, the Boolean GWO is modeled to perform multi-objective HFUI. After the population initialization, the problem is divided into subproblems, and it proceeds by the finding of neighbors. Later, the solutions are evolved to next generation iteration.

4.3.1. Initialization strategy

In general, optimization algorithms start with an initial population in a random manner. However, initializing in a random manner for itemset mining will often lead to infeasible solutions. For example, there is no combination of items $\{b, g\}$. Suppose a solution is generated as $[0, 1, 0, 0, 0, 0, 1, 0, 0, 0]$ and sent for fitness calculation; it will return a value 0. If this is chosen to be a crossover solution, then the generation of a child solution will also be ineffective. Thus, for an effective initialization, two

things are required: 1. The generated solutions should indicate a valid itemset combination from the database. 2. The generated solutions after the search strategy through BGWO should also be a valid itemset. If these two conditions are met, then the convergence of optimal solutions in a shorter number of iterations is highly possible in the search strategy. Hence, two selection strategies are adopted from Multi-Objective Evolutionary Algorithm based Decomposition (MOEA/D) based HFUI mining.

Transaction Itemset: The transaction-itemset utility can be denoted as $util(t_i)$. The sum of all transaction utilities can be computed as $\sum_{i=1}^{|D|} util(t_i)$. Individuals can be selected based on the probability of $\frac{util(t_i)}{\sum_{i=1}^{|D|} util(t_i)}$

Meta-Itemset: Likewise for meta-itemsets, let us consider that there are 10 meta-itemsets in the transaction database D , namely, $\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}, \{h\}, \{i\}$ and $\{j\}$. The frequency of items can be calculated as $freq(M_i)$. The collective calculation of frequencies can be computed as $\sum_{i=1}^{|M|} freq(M_i)$. Now, the choice if an individual can be selected is based on the probability $\frac{freq(M_i)}{\sum_{i=1}^{|M|} freq(M_i)}$.

4.3.2. Creation of subproblems

Let us consider that there are N_{Pop} individuals in the population. N_{Pop} subproblems are generated as follows:

$$\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{N_{Pop}}\} \quad (5)$$

are the evenly spread vectors in which each λ corresponds to 2 values, namely, $\lambda_i = (\lambda_i^1, \lambda_i^2)$ where the 2 refers to the number of objectives that satisfy $\lambda_i^1 + \lambda_i^2 = 1$. These subproblems are then solved using Tchebycheff method [41] which is formulated as

$$\text{minimize } g(X|\lambda_i, z^*) = \max_{j=1:2} \{(\lambda_i^j \cdot |F_j(X) - z^*|)\}, \quad (6)$$

where $z^* = (z_1^*, z_2^*)$ is the reference point that records the maximum value for individual fitness function.

4.3.3. Finding neighbors

After the creation of subproblems, the solutions used to find their neighbors for further processing. Euclidian distance for every subproblem with other subproblem is computed and then sort the individual's based on the Euclidian distance in descending order. Then extract the first NB number of individuals and name it as the neighbours of every corresponding individual.

4.3.4. Dominant determination

In Section 3.2 description of multi-objective optimization model is given, and the domination

strategy that identifies the solutions with better results in different objective functions is defined. In this section, the pseudocode of dominant determination is given. A solution x can be termed as a non-dominant solution iff none of the objective value of that solution is lesser than the other solution's objective values. The pseudocode is given in Algorithm 1.

Table 5. Nomenclature for Algorithms 1 and 2.

Parameter name	Variable representation
Population / Solutions in a Generation	$\vec{\psi}$
A Single Individual / Solution	$\vec{\psi}_i$
Total number of solutions in a population	N_{pop}
Fitness Function	F
Fitness function of first objective	f_1
Fitness function of second objective	f_2
Dimension	d
Transaction Database	\mathbb{D}
Number of Objectives	N_{obj}
Current iteration	t
Maximum number of iterations	MAX_{IT}
Number of Neighbors	T
Random vectors	$\vec{a}, \vec{r}_1, \vec{r}_2$
Coefficient Vectors	\vec{A}, \vec{C}
Markers for population split of wolves as alpha, beta and gamma.	L_1, L_2, L_3, L_4
Optimal solution index	z
Sorted population	S
Dominated solution indices	D_D
External Population	EP
Nearest Neighbor Index	T_g
Decomposed cost of every individual	g_i

Table 5 represents the parameters used in Algorithms 1 and 2 and the respective variables that represent the parameters. Algorithm 1 represents the dominance identification model. In Algorithm 2, the actual working procedure of the Boolean multi-objective optimization-based decomposition algorithm is presented.

Algorithm 1: Dominant determination

Input: Population $(\vec{\psi})$, $N_{pop} = |\vec{\psi}|$, **Objective Function** $F = (f_1, f_2)$
 for each $i \in N_{pop}$ do
 for each $j \in N_{pop}$ do
 if $\left(\left(\forall (F(\vec{\psi}_i) \leq F(\vec{\psi}_j)) \right) \&\& \left(\exists (F(\vec{\psi}_i) < F(\vec{\psi}_j)) \right) \right)$ then
 $\vec{\psi}_j.isDominated = True$
 elseif $\left(\left(\forall (F(\vec{\psi}_j) \leq F(\vec{\psi}_i)) \right) \&\& \left(\exists (F(\vec{\psi}_j) < F(\vec{\psi}_i)) \right) \right)$ then
 $\vec{\psi}_i.isDominated = True$
 end if
 end for
end for
Output: $\vec{\psi}.isDominated$

After calculating the domination of individuals, the solution for the next generation is computed using BGWO, which is given in Algorithm 2. In the later part, once again, the decomposed cost and determination of domination of individuals are computed to update the external population (EP).

Algorithm 2: Multi-objective Boolean grey wolf optimization based decomposition algorithm on HFUI

Input: Transaction Database (\mathbb{D}), Number of Objectives (N_{obj}), Objective Function (f_1, f_2) , initial iteration (t), Population Size (N_{pop}), Dimension of an individual (d), Population Space $(\vec{\psi}(N_{pop}, d) = \emptyset)$, Maximum Iteration (MAX_{IT}), Number of Neighbors (T)

Begin

// Parameter Initialization

$t \leftarrow 1$

$\vec{a}, \vec{r}_1, \vec{r}_2 \leftarrow randi([0, 1], 1, d)$

$\vec{A} = (2 \leftarrow \vec{a}) \wedge (\vec{r}_1 \ominus \vec{a})$

$\vec{C} = 2 \mapsto \vec{r}_2$

Initialize L_1, L_2, L_3, L_4

Initialize z

// Population Initialization & Fitness Computation

for each $i \in N_{pop}$ do

$\vec{\psi}_i \leftarrow randi([0, 1], 1, d)$

$[Fit_i] \leftarrow f_1(\vec{\psi}_i), f_2(\vec{\psi}_i)$

$z \leftarrow \max(z, Fit_i)$

end for

/ Creating Sub Problems

for each $i \in N_{pop}$ do

$\lambda_i \leftarrow rand(N_{obj}, 1)$

$\lambda_i \leftarrow \lambda_i / norm(\lambda_i)$

end for

$D \leftarrow Euclid(\lambda, \lambda)$

// Finding Neighbours (NB)

```

for each  $i \in N_{Pop}$  do
     $S \leftarrow \text{sort}(D(i, :))$ 
     $NB_i \leftarrow S(1:T)$ 
end for
// Calculation of Decomposed Cost
for each  $i \in N_{Pop}$  do
     $g_i \leftarrow \max(\lambda_i \cdot \text{abs}(Fit_i - z))$ 
end for
// Domination Determination
 $D_D \leftarrow \text{Domination}(\vec{\psi})$  // Logical Value
 $EP \leftarrow \vec{\psi}(\sim[D_D])$ 
// Iteration Starts
do
// Sorting and extraction of  $\alpha, \beta$  and  $\gamma$  Individuals
 $\vec{\psi}_{Sorted} \leftarrow \text{sort}(\vec{\psi}, 'descend')$ 
 $\vec{\psi}_\alpha \leftarrow \vec{\psi}_{Sorted} \left( \text{randi} \left( \left[ 1, \frac{N_{Pop}}{100} \times L_1 \right] \right) \right)$ 
 $\vec{\psi}_\beta \leftarrow \vec{\psi}_{Sorted} \left( \text{randi} \left( \left[ \frac{N_{Pop}}{100} \times L_2, \frac{N_{Pop}}{100} \times L_3 \right] \right) \right)$ 
 $\vec{\psi}_\gamma \leftarrow \vec{\psi}_{Sorted} \left( \text{randi} \left( \left[ \frac{N_{Pop}}{100} \times L_4, N_{Pop} \right] \right) \right)$ 

for each  $i \in N_{Pop}$  do
     $NB_{T1}, NB_{T2}, NB_{T3} \leftarrow \text{randsample}(T, 3)$ 
     $\vec{\psi}_{T1} \leftarrow \vec{\psi}(NB_{T1}); \vec{\psi}_{T2} \leftarrow \vec{\psi}(NB_{T2}); \vec{\psi}_{T3} \leftarrow \vec{\psi}(NB_{T3})$ 
     $\vec{D}_\alpha \leftarrow |\vec{C}_1 \bowtie (\vec{\psi}_{T1} \ominus \vec{\psi}_i)|, \vec{D}_\beta \leftarrow |\vec{C}_2 \bowtie (\vec{\psi}_{T2} \ominus \vec{\psi}_i)|, \vec{D}_\gamma \leftarrow |\vec{C}_3 \bowtie (\vec{\psi}_{T3} \ominus \vec{\psi}_i)|$ 
     $\vec{\psi}_1 \leftarrow |\vec{\psi}_\alpha \ominus (\vec{A}_1 \bowtie \vec{D}_\alpha)|, \vec{\psi}_2 \leftarrow |\vec{\psi}_\beta \ominus (\vec{A}_2 \bowtie \vec{D}_\beta)|, \vec{\psi}_3 \leftarrow |\vec{\psi}_\gamma \ominus (\vec{A}_3 \bowtie \vec{D}_\gamma)|$ 
     $\vec{\psi}_{sum}, \vec{\psi}_{Carry} \leftarrow \vec{\psi}_1 \oplus \vec{\psi}_2 \oplus \vec{\psi}_3$ 
    if  $(f(\vec{\psi}_{sum}) < f(\vec{\psi}_{Carry}))$  then
         $\vec{\psi}_T \leftarrow \vec{\psi}_{sum}$ 
         $Fit_T \leftarrow f(\vec{\psi}_{sum})$ 
    else
         $\vec{\psi}_T \leftarrow \vec{\psi}_{Carry}$ 
         $Fit_T \leftarrow f(\vec{\psi}_{Carry})$ 
    end if
     $z \leftarrow \max(z, Fit_T)$ 
    for each  $j \in T$  do
         $T_g \leftarrow \max(\lambda_j \cdot \text{abs}(Fit_T - z))$ 
        if  $(T_g \leq g_i)$  then
             $\vec{\psi}_i \leftarrow \vec{\psi}_T$ 
        end if
    end for
end for
end for
// Domination Determination

```

$D_D \leftarrow \text{Domination}(\vec{\psi})$ // Logical Value

$EP \leftarrow \vec{\psi}(\sim[D_D])$

$\vec{a}, \vec{r}_1, \vec{r}_2 \leftarrow \text{randi}([0, 1], 1, d)$

$\vec{A} = (2 \leftarrow \vec{a}) \wedge (\vec{r}_1 \ominus \vec{a})$

$\vec{C} = 2 \mapsto \vec{r}_2$

$t \leftarrow t + 1$

Until ($t \leq \text{MAX}_{IT}$)

$\vec{\psi}^* \leftarrow \vec{\psi}(EP(\max(\text{Fit})))$

End

Output: EP

4.3.5. Working model of MOBGWO

In the initial phase, three different random binary vectors are generated, namely, $\vec{a}, \vec{r}_1, \vec{r}_2$, which are the basic blocks for the construction of coefficient factors \vec{A}, \vec{C} respectively. For the construction of coefficient vector \vec{A} , a Boolean operation “AND” gate is used between the vector \vec{a} with left two-bit circular shift and a difference vector between \vec{a} and \vec{r}_1 . Similarly, the coefficient vector \vec{C} is prepared by right two-bit circular shift on \vec{r}_2 .

In the population initialization phase, every solution vector ($\vec{\psi}_i$) is prepared randomly, and its respective fitness values are also calculated using two objective functions, namely, f_1 and f_2 . The best fitness value is recorded in the variable z .

A subset of vectors λ is generated for decomposing the problem based on the number of objectives. This vector is responsible for the weightage given to the objectives for every solution. Since a different combination strikes out during the iteration, the combinations will have high probability to end up with optimal Pareto front solutions.

For every solution, the near neighbor solutions (i.e., T solutions) are identified based on the Euclidian distance between the λ vector representations of every solution. Then, for every solution, the decomposed cost will be identified as g_i . Now, the domination factor of every solution is identified based on Algorithm 1.

Based on the default threshold values to segregate the wolves as alpha, beta and gamma wolves, the solution is sorted based on the decomposition cost of every solution. Then, from sorted solution index 1 to L_1 , the solutions will be treated as Alpha wolves. From L_2 (i.e., L_1+1) to L_3 , the individuals will be treated as beta wolves, and the rest will be treated as Gamma wolves. Now, the evolution of wolves from one generation to the next generation will be carried out based on the Boolean operations described in Section 4.2.

After the identification of optimal solutions from the generated solutions, the best will be promoted for the further iteration evolution. The process will be continued till it reaches the maximum number of iterations, and the optimal solution will be registered at the end of the algorithm from the

external population archive (EP). Figure 1 shows the flowchart of the proposed MOBGWO.

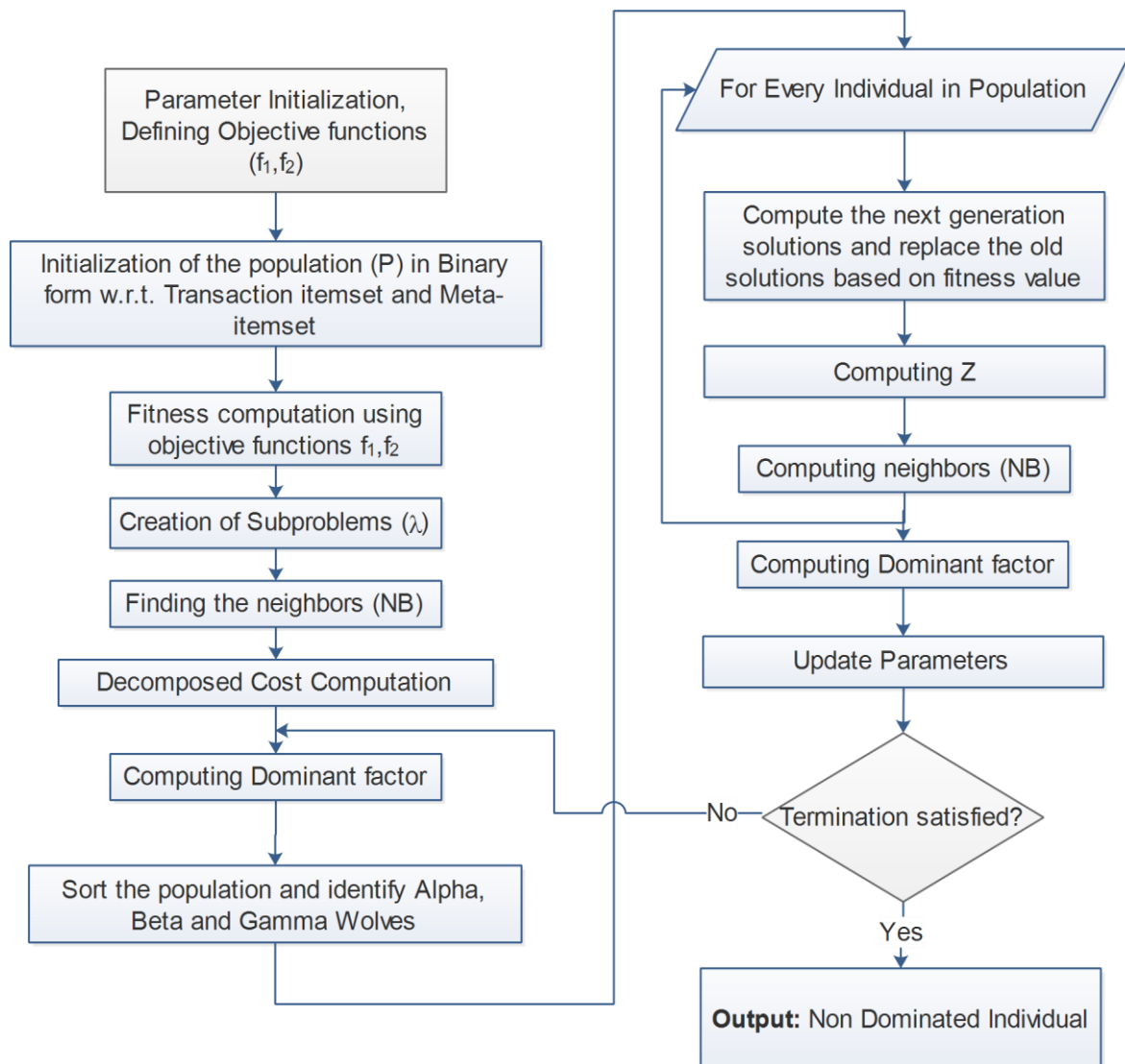


Figure 1. Flowchart of MOBGWO.

5. Experimental evaluation

In this section, the experimental setup for evaluation of the proposed algorithm, the datasets used for evaluation, the performance metrics for evaluation, and the results analysis are discussed in detail.

5.1. Experimental setup and dataset

The formulated technique was executed in MATLAB v9.2 in a system configured with a Core i7 9th generation processor, 8 GB of RAM and a 1 TB HDD. The model bounds are shown in Table 6. The performance of the proposed algorithm is compared with existing multi-objective models, such as MOEA-HFUI [42], HUI-Miner [23], HUIM-ACS [30], FP-Growth [14], TKU-Miner [43], UBP-

Miner [44] and BOEA-HFUI [45]. For evaluating the proposed algorithm, 12 real datasets were downloaded from the SPMF repository [46]. The details of the datasets are given in Table 7. None of the datasets holds the weight of each item in any transaction. Hence, the distribution of weights of each item is done based on the previous works [20,23,43] that generated the weight values for each item in the range 1 to 1000.

Table 6. Simulation parameters.

Parameters	Values
Population Size	100
Runs	10
Iterations/Run	500
Circular Position Shift	2
min_{freq}	0.5
min_{util}	0.5
Neighbors (T)	15
N_{obj}	2

Table 7. Details of the dataset.

Dataset	Transaction Length	No. of Items	No. of Transactions	% of dataset used
Accident	46	469	34018	20%
Foodmart	11	1559	21557	80%
USCensus	48	396	100000	20%
PowerAC	7	133	104000	90%
Chess	37	76	3196	100%
Susy	19	191	50000	5%
Connect	43	129	33779	80%
BMS-web-view-1	181	497	59602	80%
KDDcup99	16	133	100000	20%
Pamp	24	142	100000	20%
OnlineRetail	8	2585	54191	20%
Mushroom	23	120	8124	100%

5.2. Performance metrics

Two performance indicators, namely, Hypervolume [47] and coverage [48], were used to measure the formulated technique in a multi-objective solution space.

Hypervolume (HV): It is utilized to calculate the convergence and diversity of the population in the given solution space. Larger values of hypervolume indicate better performance of the algorithm.

Coverage (Cov): It determines the diversity in population of the algorithm. It can be mathematically represented as

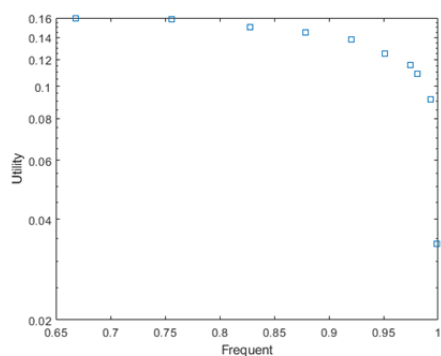
$$Cov = \frac{N_d}{N}, \quad (7)$$

where N indicates the total number of items and N_d indicates the different items present in the external population. Larger values of coverage indicate better diversity achieved by the algorithm.

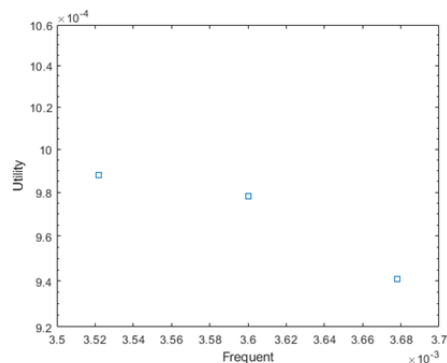
5.3. Experimental analysis

The results of the proposed algorithm are compared with the other standard algorithms in terms of non-dominated solutions achieved from the multi-objective perspective, the curve of Pareto front solutions and the values achieved in terms of coverage and hypervolume for the given solution space.

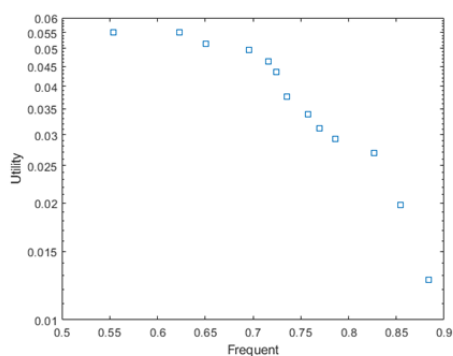
Figure 2 shows the non-dominated solutions and their curves achieved with respect to the two objectives, namely, frequency and utility of the itemset. The achieved solutions are plotted in the graph for all 12 real time datasets which were obtained from the SPMF data repository.



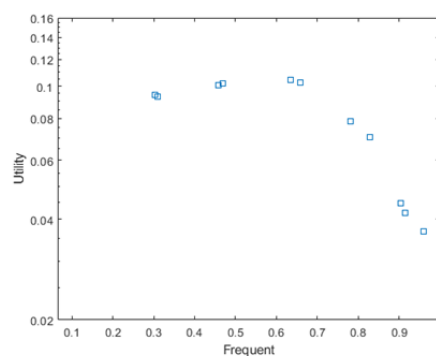
a) Accident



b) Foodmart



c) USSensus



d) PowerAC

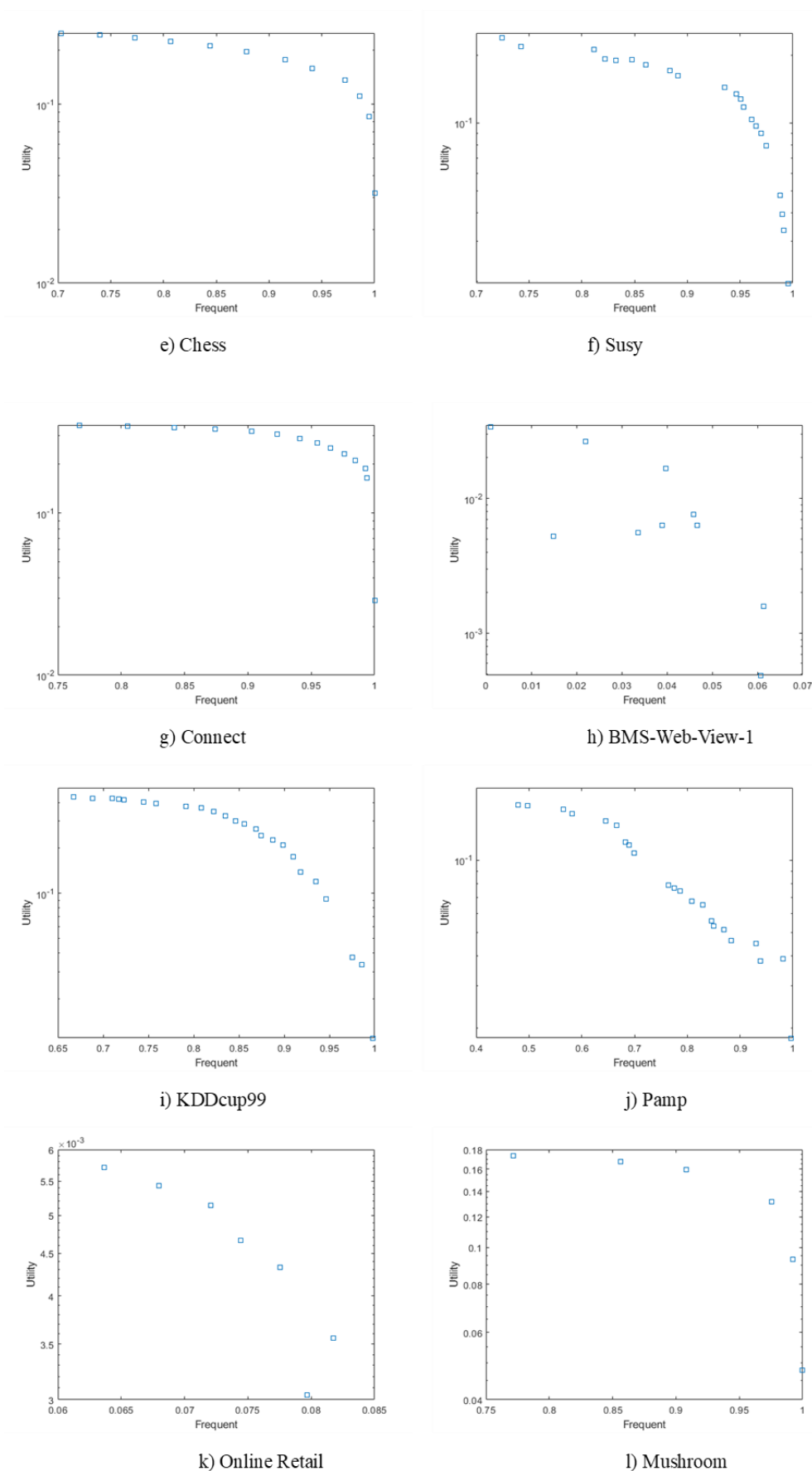


Figure 2. Non-dominated solutions of MOBGWO-HFUI on all 12 datasets.

The total number of non-dominated solutions achieved using MOBGWO for HFUI on the datasets Chess, Connect, Accidents, Mushroom, Foodmart, OnlineRetail, USCensus, Pamp, KDDcup99, Susy, BMS-Web-View-1 and Powerc were 12, 14, 10, 6, 6, 7, 13, 22, 24, 21, 10 and 11, respectively. The comparison of the achieved solutions over the existing algorithms are shown in Figure 3.

In Figure 3, the initialization strategy of the proposed algorithm is examined. Four different initialization strategies were being tested over a small number of datasets, namely, the Chess, Connect and Accidents datasets. Both the HV and Coverage were examined to fix the optimal initialization strategy. The compared models include MOBGWO-HFUI (rand), which represents the random generation of solutions irrespective of their infeasibility. The next model is MOBGWO-HFUI (Trans), which represents the probability initialization strategy through the transaction-itemset model. The next model is MOBGWO-HFUI (Meta), which represents the probability initialization strategy through the meta-itemset model. Finally, a hybrid initialization model MOBGWO-HFUI was used which is the fusion of Trans and Meta itemset initialization. This model holds half of the population with Trans and the other half with the meta itemset initialization model. The inferences and the analysis are discussed in this section.

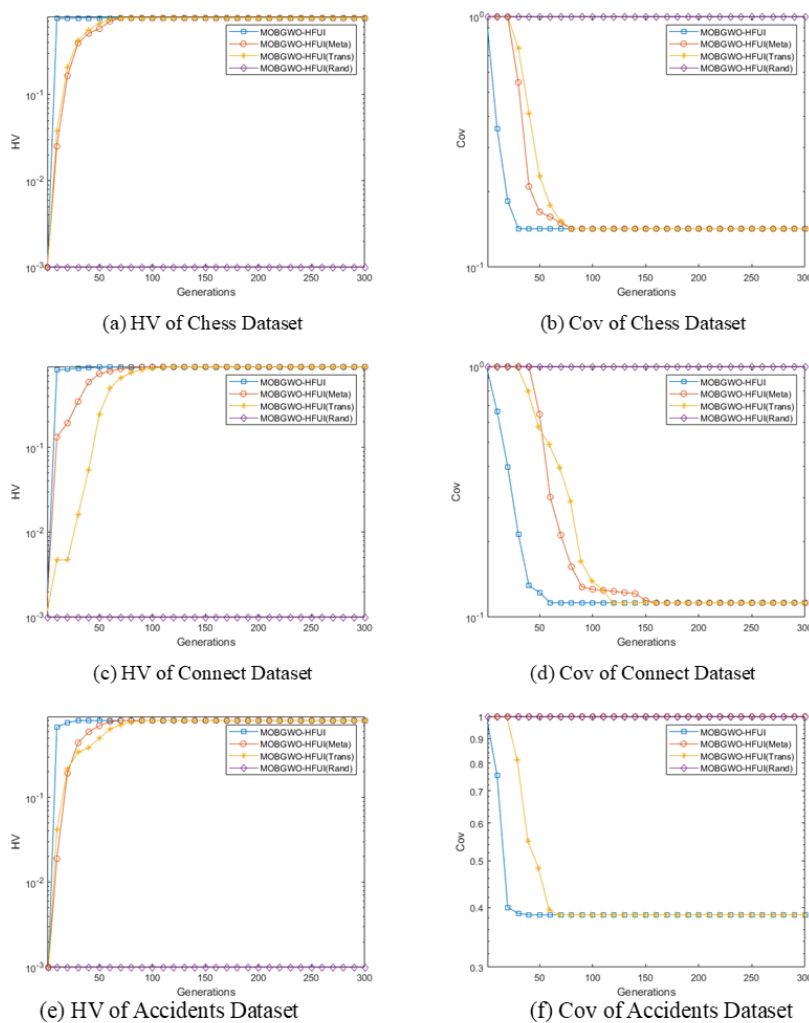


Figure 3. Performance indicators with different initialization strategies for MOBGWO-HFUI.

From Figure 3 it can be inferred that both on convergence and diversity preservation the hybrid initialization model outperforms the existing initialization models. Figure 3(a),(c),(e) shows the performance of different initialization models for the performance indicator HV for the dataset Chess, Connect and Accident, respectively. Similarly, Figure 3(b),(d),(f) shows the performances of different initialization models for the performance indicator coverage for the datasets Chess, Connect and Accident, respectively. From the observation of Figure 3, it is worth it to use the hybrid initialization strategy for further implementation of the proposed algorithm.

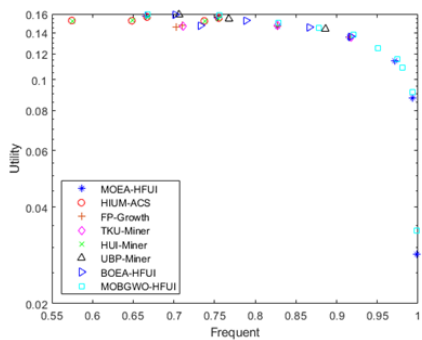
Figure 4 shows the non-dominant solution set plot with respect to frequency and utility of the itemset obtained. There are 12 subplots showing the performance graphs of all algorithms for the 12 real time datasets, as indicated from the a to l subgraphs.

From Figure 4, on comparing the counts of non-dominant individuals, the following inferences are obtained. For the Chess dataset, the number of obtained non-dominant individuals for the algorithm MOEA-FHUI was 12, for HUIM-ACS was 5, for FP Growth was 6, for TKU Miner was 4, for HUI-Miner was 3, for UBP-Miner was 4, for BOEA-HFUI was 6 and for MOBGWO-HFUI was 12. Similarly, from the subgraphs we can perceive that the proposed algorithm obtained a greater number of Pareto front solutions when compared with the existing algorithms.

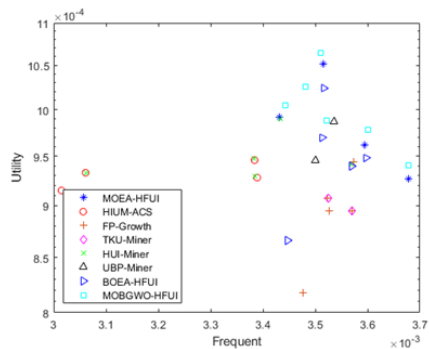
Tables 8 and 9 show the performance indicator values Coverage and Hypervolume, respectively. The values of the indicators range from 0 to 1. As the number approaches 1, it indicates the improvement of the algorithm. In Tables 8 and 9, a \pm sign indicates the results in the range specified. The other three symbols ($+ / - / \approx$) indicate better, worse and equal performance compared to our proposed algorithm.

From Table 8, it is evident that the proposed algorithm outperforms the existing algorithms in eight datasets out of 12. Next to the proposed algorithm, MOEA-HFUI outperforms our proposed algorithm in 2 datasets, namely, Accidents and Powerc. The proposed algorithm performs significantly equally to the existing algorithm MOEA-HFUI in five datasets, namely, Chess, Connect, Mushroom, KDDcup99 and Susy. In Foodmart dataset TKU-Miner outperforms all existing algorithms including our proposed algorithm. On BMS-Web-View-1 dataset HUI-Miner, HUIM-ACS and our proposed MOBGWO-HFUI perform equally and achieve the maximum value 1. On comparing the proposed algorithm results with UDP-Mnier, the results effectively for 10 datasets out of 12 and on comparing with BOEA-HFUI proposed algorithm outperforms in eight different datasets out of 12.

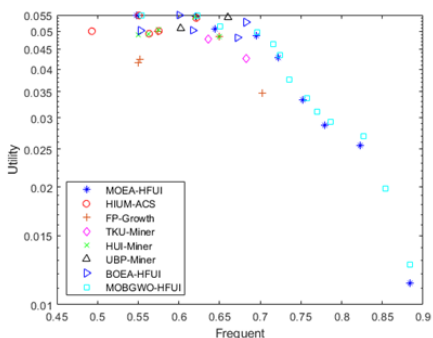
From Table 9, the proposed algorithm MOBGWO-HFUI outperforms all the existing algorithm in eight datasets. The existing algorithm MOEA-HFUI outperforms the proposed algorithm on three datasets, namely, Mushroom, Foodmart and Susy dataset. The existing algorithm MOEA-FHUI significantly performs equally with our proposed algorithm MOBGWO-HFUI with Chess, Accidents, Pamp, KDDcup99 and Powerc datasets. On comparing the proposed algorithm results with UDP-Miner, the results are effective for 11 datasets out of 12 except BMS-Web-View dataset, and on comparing with BOEA-HFUI, the proposed algorithm outperforms in nine different datasets out of 12.



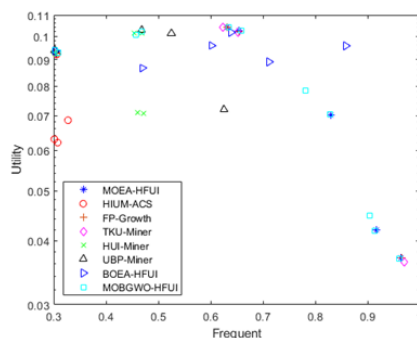
(a) Accident



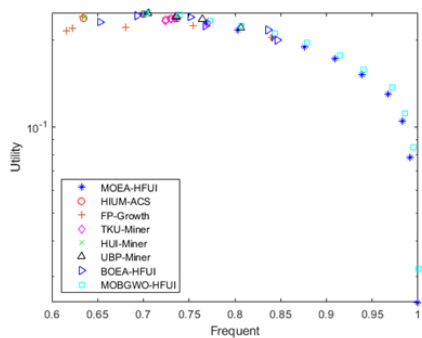
(b) Foodmart



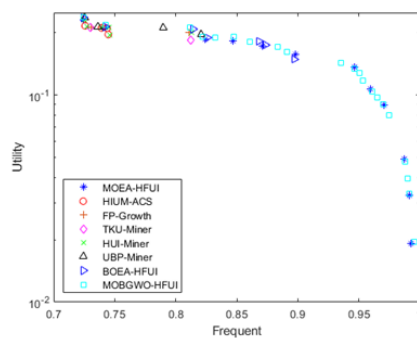
(c) USSensus



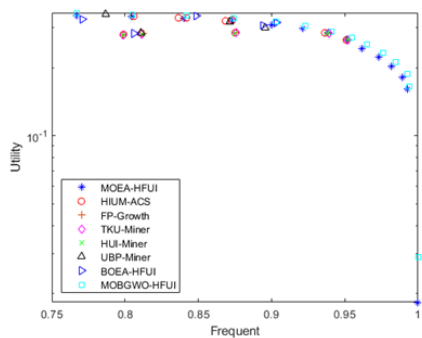
(d) PowerAC



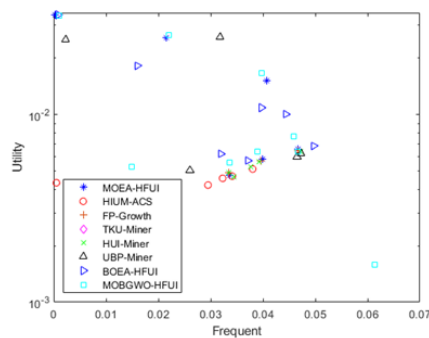
(e) Chess



(f) Susy



(g) Connect



(h) BMS-Web-View-1

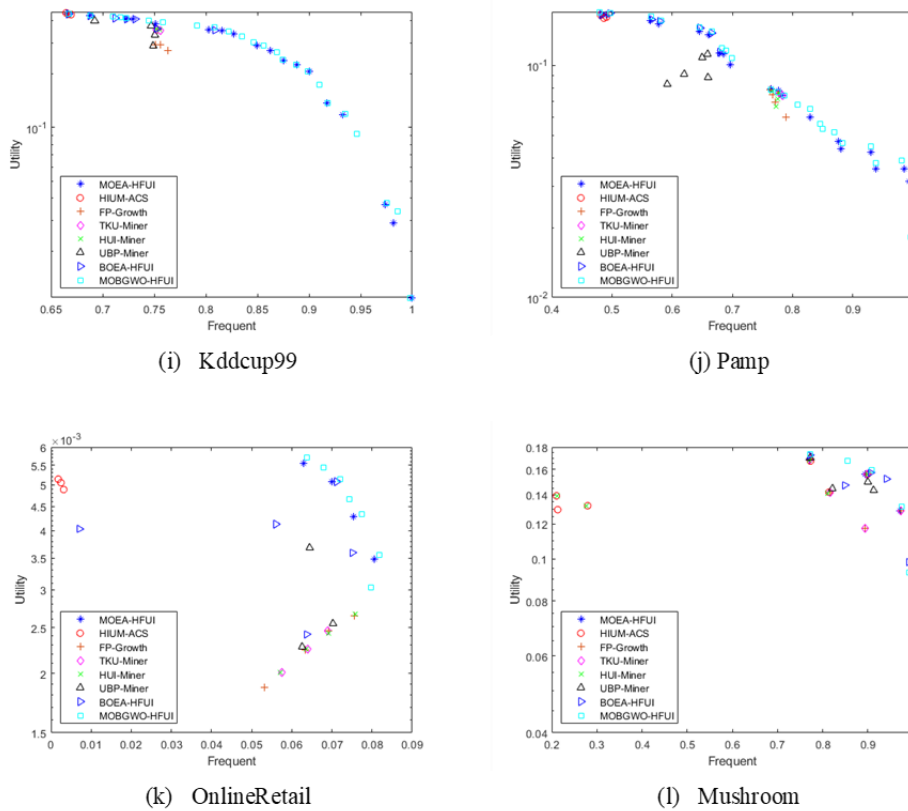


Figure 4. Non-dominated solutions of MOBGWO-HFUI and existing algorithms on all 12 datasets.

Table 8. Performance indicator coverage for 12 real time datasets.

Dataset	TKU-Miner	FP-growth	HUI-Miner	HUIM-ACS	MOEA-FHUI	UPI-Miner	BOEA-FHUI	MOBGWO-HFUI
Accidents	0.253 ⁻	0.237 ⁻	0.267 ⁻	0.252 ⁻ ± 0.02	0.275⁺ ± 0.002	0.271 ⁻	0.276⁺ ± 0.001	0.271 ± 0.001
Foodmart	0.979⁺	0.915 ⁻	0.955 ⁻	0.943 ⁻ ± 0.01	0.938 ⁻ ± 0.001	0.975 [≈]	0.973 [≈] ± 0.002	0.971 ± 0.004
USCensus	0.305 ⁻	0.275 ⁻	0.289 ⁻	0.285 ⁻ ± 0.02	0.328 ⁻ ± 0.01	0.352 [≈]	0.357 [≈] ± 0.01	0.356 ± 0.04
Powerc	0.378 ⁻	0.372 ⁻	0.334 ⁻	0.334 ⁻ ± 0.01	0.451 ⁺ ± 0.01	0.411 ⁻	0.452⁺ ± 0.01	0.448 ± 0.01
Chess	0.159 ⁻	0.151 ⁻	0.150 ⁻	0.152 ⁻ ± 0.01	0.193 [≈] ± 0.004	0.198⁺	0.192 ⁻ ± 0.003	0.196 ± 0.002
Susy	0.273 ⁻	0.264 ⁻	0.259 ⁻	0.254 ⁻ ± 0.02	0.312 [≈] ± 0.06	0.318 [≈]	0.314 ⁻ ± 0.02	0.321 ± 0.04
Connect	0.120 ⁻	0.108 ⁻	0.132 ⁻	0.134 ⁻ ± 0.03	0.151[≈] ± 0.005	0.144 ⁻	0.152 [≈] ± 0.001	0.151 ± 0.002
BMS-Web-View-1	0.698 ⁻	0.713 ⁻	1[≈]	1[≈]	0.764 ⁻ ± 0.01	1[≈]	1[≈]	1
KDDcup99	0.257 ⁻	0.283 ⁻	0.255 ⁻	0.268 ⁻ ± 0.03	0.299 [≈] ± 0.01	0.299 ⁻	0.301[≈] ± 0.02	0.301 ± 0.01
Pamp	0.317 ⁻	0.286 ⁻	0.313 ⁻	0.327 ⁻ ± 0.004	0.345 ⁻ ± 0.01	0.353 ⁻	0.362 ⁻ ± 0.01	0.368 ± 0.04
OnlineRetail	0.974 ⁻	0.948 ⁻	0.967 ⁻	0.965 ⁻ ± 0.01	0.995 ⁻ ± 0.001	0.987 ⁻	0.994 ⁻ ± 0.002	0.998 ± 0.001
Mushroom	0.342 ⁻	0.311 ⁻	0.351 ⁻	0.337 ⁻ ± 0.04	0.431 [≈] ± 0.006	0.436 [≈]	0.434 [≈] ± 0.002	0.437 ± 0.002

Table 9. Performance indicator Hypervolume for 12 real time datasets.

Dataset	TKU-Miner	FP-growth	HUI-Miner	HUIM-ACS	MOEA-FHUI	UPI-Miner	BOEA-FHUI	MOBGWO-HFUI
Accidents	0.756 ⁻	0.741 ⁻	0.672 ⁻	0.686 ⁻ ± 0.03	0.769 [≈] ± 0.02	0.783 ⁻	0.802[≈] ± 0.03	0.801 ± 0.04
Foodmart	0.053⁺	0.040 ⁻	0.050 ⁻	0.035 ⁻ ± 0.04	0.036 ⁺ ± 0.001	0.051 ⁺	0.035 ⁺ ± 0.001	0.034 ± 0.002
USCensus	0.617 ⁻	0.615 ⁻	0.617 ⁻	0.624 ⁻ ± 0.009	0.650 ⁻ ± 0.01	0.658 ⁻	0.679 ⁻ ± 0.03	0.687 ± 0.04
Powerc	0.411 ⁻	0.398 ⁻	0.397 ⁻	0.409 ⁻ ± 0.01	0.424 [≈] ± 0.008	0.421 ⁻	0.429⁺ ± 0.003	0.421 ± 0.002
Chess	0.776 ⁻	0.730 ⁻	0.759 ⁻	0.754 ⁻ ± 0.02	0.804 ⁻ ± 0.01	0.801 ⁻	0.806 ⁻ ± 0.01	0.811 ± 0.03
Susy	0.819 ⁻	0.822 ⁻	0.824 ⁻	0.811 ⁻ ± 0.03	0.835⁺ ± 0.03	0.825 ⁻	0.827 ⁻ ± 0.01	0.829 ± 0.02
Connect	0.905 ⁻	0.896 ⁻	0.870 ⁻	0.902 ⁻ ± 0.01	0.928 ⁻ ± 0.008	0.917 ⁻	0.936 ⁻ ± 0.002	0.946 ± 0.004
BMS-Web-View-1	0.050 ⁻	0.048 ⁻	0.069 ⁻	0.061 ⁻ ± 0.02	0.085 ⁻ ± 0.01	0.093[≈]	0.085 ⁻ ± 0.02	0.092 ± 0.04
KDDcup99	0.885 ⁻	0.869 ⁻	0.890 ⁻	0.895 ⁻ ± 0.01	0.897 [≈] ± 0.002	0.892 ⁻	0.898 [≈] ± 0.001	0.902 ± 0.04
Pamp	0.817 ⁻	0.823 ⁻	0.812 ⁻	0.801 ⁻ ± 0.01	0.858 [≈] ± 0.01	0.812 ⁻	0.859 [≈] ± 0.01	0.862 ± 0.04
OnlineRetail	0.090 ⁻	0.085 ⁻	0.103 ⁻	0.096 ⁻ ± 0.04	0.111 ⁻ ± 0.002	0.111 ⁻	0.113 ⁻ ± 0.001	0.121 ± 0.005
Mushroom	0.832 ⁻	0.745 ⁻	0.791 ⁻	0.789 ⁻ ± 0.02	0.839 ⁺ ± 0.01	0.837 ⁺	0.840⁺ ± 0.02	0.832 ± 0.04

Table 10. Wilcoxon signed rank test on coverage for 12 real time datasets.

Datasets	TKU-Miner Vs MOBGWO-HFUI				FP-growth Vs MOBGWO-HFUI				HUI-Miner Vs MOBGWO-HFUI				HUIM-ACS Vs MOBGWO-HFUI				MOEA-FHUI Vs MOBGWO-HFUI				UDP-Miner Vs MOBGWO-HFUI				BOEA-FHUI Vs MOBGWO-HFUI			
	P-	T+	T-	Winner	P-	T+	T-	Winner	P-	T+	T-	Winner	P-	T+	T-	Winner	P-	T+	T-	Winner	P-	T+	T-	Winner	P-	T+	T-	Winner
	Value				Value				Value				Value				Value				Value							
Accidents	0	0	55	+	1.73E-06	0	55	+	1.73E-06	0	55	+	0	0	55	+	3.38E-03	30	25	-	0	0	55	+	3.38E-03	30	25	-
Foodmart	1.65E-01	55	0	-	0	0	55	+	0	0	55	+	0	0	55	+	1.73E-06	0	55	+	1.80E-05	18	37	+	1.73E-06	0	55	+
USCensus	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	1.73E-06	0	55	+	0	0	55	+
Powerc	0	0	55	+	1.73E-06	0	55	+	1.73E-06	0	55	+	0	0	55	+	3.38E-03	30	25	-	0	0	55	+	3.38E-03	30	25	-
Chess	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	3.38E-03	30	25	-	0	0	55	+
Susy	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+
Connect	1.83E-03	6	49	+	1.80E-05	18	37	+	4.68E-03	12	43	+	1.73E-06	0	55	+	1	0	0	=	0	0	55	+	1	0	0	=
BMS-Web-View-1	0	0	55	+	0	0	55	+	1	0	0	=	1	0	0	=	0	0	55	+	1	0	0	=	0	0	55	+
KDDcup99	1.83E-03	4	51	+	1.80E-05	16	39	+	4.68E-03	9	46	+	1.73E-06	0	55	+	0	0	55	+	0	0	55	+	4.68E-03	46	9	-
Pamp	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+
OnlineRetail	1.65E-01	55	0	-	0	0	55	+	0	0	55	+	0	0	55	+	1.73E-06	0	55	+	0	0	55	+	1.73E-06	0	55	+
Mushroom	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+	0	0	55	+
+/-/-	10/0/2				12/0/0				11/1/0				11/1/0				9/1/2				10/1/1				8/1/3			

Table 11. Wilcoxon signed rank test on Hypervolume for 12 real time datasets.

Function	TKU-Miner Vs				FP-growth Vs MOBGWO-				HUI-Miner Vs				HUIM-ACS Vs				MOEA-FHUI Vs				UDP-Miner Vs				BOEA-FHUI Vs							
	MOBGWO-HFUI				HFUI				MOBGWO-HFUI				MOBGWO-HFUI				MOBGWO-HFUI				MOBGWO-HFUI				MOBGWO-HFUI							
	P-Value	T+	T-	Winne r	P-Value	T+	T-	Winne r	P-Value	T+	T-	Winne r	P-Value	T+	T-	Winne r	P-Value	T+	T-	Winne r	P-Value	T+	T-	Winne r	P-Value	T+	T-	Winne r				
Accidents	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	4.68E	42	1	-
			5				5				5				5				5				5				5		-03		3	
Foodmart	1	55	0	-	0	0	5	+	0	0	5	+	1.83E	13	4	+	1.73E	0	5	+	1	55	0	-	1.73E	0	5	+	1.73E	0	5	+
							5				5		-03		2		-06		5						-06		5		-06		5	
USCensus	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+
			5				5				5				5				5				5				5				5	
Powerc	3.38E	0	5	+	0	0	5	+	0	0	5	+	1.73E	0	5	+	1	55	0	-	0	0	5	+	1	55	0	-	1	55	0	-
	-03		5				5				5		-06		5								5				5				5	
Chess	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+
			5				5				5				5				5				5				5				5	
Susy	3.38E	0	5	+	0	0	5	+	0	0	5	+	1.73E	0	5	+	1	55	0	-	0	0	5	+	1	55	0	-	1	55	0	-
	-03		5				5				5		-06		5								5				5				5	
Connect	1.83E	13	4	+	0	0	5	+	0	0	5	+	1.73E	0	5	+	1.73E	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+
	-03		2				5				5		-06		5		-06		5				5				5				5	
BMS-Web-View-1	1.83E	14	4	+	1.80E	10	4	+	4.68E	0	5	+	1.73E	0	5	+	1.73E	0	5	+	4.68E	46	9	-	0	0	5	+	0	0	5	+
	-03		1		-05		5		-03		5		-06		5		-06		5		-03						5				5	
KDDcup99	0	0	5	+	0	0	5	+	0	0	5	+	1.73E	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+
			5				5				5		-06		5				5				5				5				5	
Pamp	0	0	5	+	4.11E	15	4	+	0	0	5	+	1.73E	0	5	+	1.73E	0	5	+	0	0	5	+	1.73E	0	5	+	1.73E	0	5	+
			5		-03		0				5		-06		5		-06		5				5		-06		5		-06		5	
OnlineRetail	0	0	5	+	0	0	5	+	0	0	5	+	1.73E	0	5	+	1.73E	0	5	+	0	0	5	+	0	0	5	+	0	0	5	+
l			5				5				5		-06		5		-06		5				5				5				5	
Mushroom	3.38E	0	5	+	0	0	5	+	0	0	5	+	1.73E	0	5	+	1	55	0	-	1	55	0	-	1	55	0	-	1	55	0	-
	-03		5				5				5		-06		5																	
+/-	11/0/1				12/0/0				12/0/0				12/0/0				9/0/3				9/0/3				8/0/4							

5.4. Statistical analysis–Wilcoxon signed rank test on MOBGWO-HFUI vs other state-of-art algorithms

In Tables 10 and 11, we utilized non-parametrical pairwise Wilcoxon signed rank test comparing MOBGWO-HFUI with other existing algorithms on 12 real time datasets to demonstrate impact difference. “+” denotes the cases in which the Null Hypothesis H_0 is rejected, and the proposed MOBGWO-HFUI shows superior performance to the compared algorithm. “=” indicates that there exists no statistical difference between the compared algorithms. “-” indicates that H_0 is rejected, and MOBGWO-HFUI has worse performance than the other existing technique. At the end of each table, the overall cases of pairwise comparison are given. The p-values that fall below $1.76E-6$ are rounded.

From Table 10 it is evident that the proposed algorithm is effective when compared to the other existing state of art algorithms. On analyzing the Wilcoxon signed rank test on Coverage values, MOBGWO-HFUI outperforms FP-Growth on all datasets. With HUI-Miner and HUIM-ACS, MOBGWO-HFUI competes equally on one dataset, and in the remaining 11 datasets MOBGWO-HFUI outperforms both the algorithms with significant differences. On competing with TKU-Miner, the proposed MOBGWO-HFUI outperforms in 10 different datasets, but TKU-Miner outperforms MOBGWO-HFUI in FoodMark and OnlineRetail datasets. With respect to MOEA-FHUI, MOBGWO-HFUI outperforms on 9 different datasets, competes equally with 1 dataset and loses on the other 2 datasets. Competing with UDP-Miner, MOBGWO-HFUI outperforms in 10 datasets, competes equally with one BMS-View dataset and lost to chess dataset in all the runs. Competing with BOEA-FHUI, MOBGWO-HFUI outperforms in eight datasets, competes equally with Connect dataset and loses to Accidents, PowerC and KDDcup99 datasets marginally. Overall, the performance of MOBGWO-HFUI is significant in positive aspects.

From Table 11 it is evident that the proposed algorithm is effective when compared to the other existing state of art algorithms. On analyzing the Wilcoxon signed rank test on Hypervolume values, MOBGWO-HFUI outperforms FP-Growth, HUIM-ACS and HUI-Miner on all datasets. On competing with TKU-Miner, proposed MOBGWO-HFUI outperforms in 11 different datasets, but TKU-Miner outperforms MOBGWO-HFUI in FoodMark dataset. With respect to MOEA-FHUI, MOBGWO-HFUI outperforms on 9 different datasets and loses on the other 3 datasets. Competing with UDP-Miner, MOBGWO-HFUI outperforms in nine datasets out of 12. Competing with BOEA-FHUI, MOBGWO-HFUI outperforms in eight datasets and lost to 4 other datasets marginally.

6. Conclusions

In this paper, a multi-objective Boolean grey wolf optimization based decomposition technique was introduced for solving the high-frequency and high-utility itemset mining problem by addressing both the frequency and utility as objectives in a simultaneous manner. For solving this problem, the high-frequency and high-utility itemset problem was modeled in a multi-objective domain space with 2 objectives. Then, a Boolean based grey wolf optimization algorithm was modeled to solve the multi-objective optimization problem using decomposition factor. The decomposition factor solves the problem by breaking the problem into subproblems. In this scenario the multi-objective problem has been broken into subproblems and solved effectively. To evaluate the proposed algorithm, two

performance indicators, namely, Hypervolume and coverage, were used to find the convergence and diversity achieved through the algorithm in the given solution space. Seven different multi-objective algorithms were tested in the same testbed to determine the impact of the proposed algorithm. A total of 12 real time datasets were used to prove the significance. statistical analysis using the Wilcoxon signed rank test shows the significance of the proposed algorithm over other existing algorithms. From all the classical, statistical analysis and interpretation of results in the above sections, it is proven that the proposed algorithm is effective when compared with the existing algorithms. The future direction of this research work can be extended with a greater number of objectives in itemset mining.

Acknowledgments

This research was supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R195), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflict of interest

The authors declare no conflicts of interest.

References

1. C. C. Aggarwal, J. Han, *Frequent Pattern Mining*, 1 Ed., Cham: Springer, 2014. https://doi.org/10.1007/978-3-319-07821-2_1
2. S. Ventura, J. M. Luna, *Pattern Mining with Evolutionary Algorithms*, 1 Ed., Cham: Springer, 2016. <https://doi.org/10.1007/978-3-319-33858-3>
3. J. R. Sampson, Adaptation in Natural and Artificial Systems (John H. Holland), *SIAM Review*, **18** (1976), 529–530. <https://doi.org/10.1137/1018105>
4. J. Kennedy, R. Eberhart, Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, **4** (1995), 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>
5. M. Dorigo, V. Maniezzo, A. Coloni, Ant system: Optimization by a colony of cooperating agents, *IEEE T. Syst. Man Cy. B*, **26** (1996), 29–41. <https://doi.org/10.1109/3477.484436>
6. X. S. Yang, S. Deb, Cuckoo search: Recent advances and applications, *Neural Comput. Applic.*, **24** (2014), 169–174. <https://doi.org/10.1007/s00521-013-1367-1>
7. N. Pazhaniraja, S. Sountharajan, B. Sathis Kumar, High utility itemset mining: a Boolean operators-based modified grey wolf optimization algorithm, *Soft Comput.*, **24** (2020), 16691–16704. <https://doi.org/10.1007/s00500-020-05123-z>
8. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
9. H. Faris, Hossam, I. Aljarah, M. A. Al-Betar, S. Mirjalili, Grey wolf optimizer: A review of recent variants and applications, *Neural Comput. Appl.*, **30** (2018), 413–435. <https://doi.org/10.1007/s00521-017-3272-5>

10. S. Mirjalili, S. Saremi, S. M. Mirjalili, L. S. Coelho, Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization, *Expert Syst. Appl.*, **47** (2016), 106–119. <https://doi.org/10.1016/j.eswa.2015.10.039>
11. Wolpert, David H., William G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.*, **1** (1997), 67–82. <https://doi.org/10.1109/4235.585893>
12. L. Huang, H. Chen, X. Wang, G. Chen, A fast algorithm for mining association rules, *J. Comput. Sci. Technol.*, **15** (2000), 619–624. <https://doi.org/10.1007/BF02948845>
13. A. Savasere, E. R. Omiecinski, S. B. Navathe, An efficient algorithm for mining association rules in large databases, *Proceedings of the 21th International Conference on Very Large Data Bases*, 1995, 432–444. <https://doi.org/10.5555/645921.673300>
14. J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, *ACM Sigmod Rec.*, **29** (2000), 1–12.
15. M. J. Zaki, Scalable algorithms for association mining, *IEEE Trans. Knowl. Data Eng.*, **12** (2000), 372–390. <https://doi.org/10.1109/69.846291>
16. C. Lucchese, S. Orlando, P. Palmerini, R. Perego, F. Silvestri, kDCI: A Multi-Strategy Algorithm for Mining Frequent Sets, *Proceedings of the IEEE ICDM Workshop of Frequent Itemset Mining Implementations (FIMI)*, 2003.
17. H. Yao, H.J. Hamilton, C. J. Butz, A foundational approach to mining itemset utilities from databases, *Proceedings of the 2004 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics*, 2004, 482–486. <https://doi.org/10.1137/1.9781611972740.51>
18. Y. Liu, W. K. Liao, A. Choudhary, A fast high utility itemsets mining algorithm, *Proceedings of the 1st international workshop on Utility-based data mining*, 2005, 90–99. <https://doi.org/10.1145/1089827.1089839>
19. K. Gade, J. Wang, G. Karypis, Efficient closed pattern mining in the presence of tough block constraints, *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2004, 138–147. <https://doi.org/10.1145/1014052.1014070>
20. C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, Y. K. Lee, Efficient tree structures for high utility pattern mining in incremental databases, *IEEE Trans. Knowl. Data Eng.*, **21** (12) (2009) 1708–1721. <https://doi.org/10.1109/TKDE.2009.46>
21. V. S. Tseng, C. W. Wu, B. E. Shie, P. S. Yu, Up-growth: An efficient algorithm for high utility itemset mining, *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2010, 253–262. <https://doi.org/10.1145/1835804.1835839>
22. C. W. Wu, B. E. Shie, V. S. Tseng, P. S. Yu, Mining top-k high utility itemsets, *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2012, 78–86. <https://doi.org/10.1145/2339530.2339546>
23. M. Liu, J. Qu, Mining high utility itemsets without candidate generation, *Proceedings of ACM International Conference on Information and Knowledge Management*, 2012, 55–64. <https://doi.org/10.1145/2396761.2396773>
24. H. Ryang, U. Yun, Top-k high utility pattern mining with effective threshold raising strategies, *Knowl.-Based Syst.*, **76** (2015), 109–126. <https://doi.org/10.1016/j.knosys.2014.12.010>

25. V. S. Tseng, C. W. Wu, P. Fournier-Viger, P. S. Yu, Efficient algorithms for mining top-k high utility itemsets, *IEEE Trans. Knowl. Data Eng.*, **28** (2016), 54–67. <https://doi.org/10.1109/TKDE.2015.2458860>
26. A. H. Altalhi, J. M. Luna, M. A. Vallejo, S. Ventura, Evaluation and comparison of open source software suites for data mining and knowledge discovery, *Wiley Interdiscip. Rev.: Data Min. Knowl. Discov.*, **7** (2017), e1204. <https://doi.org/10.1002/widm.1204>
27. S. Kannimuthu, K. Premalatha, Discovery of high utility itemsets using genetic algorithm with ranked mutation, *Appl. Artif. Intell.*, **28** (2014), 337–359. <https://doi.org/10.1080/08839514.2014.891839>
28. J. C. Lin, L. Yang, P. Fournier-Viger, J. M. Wu, T. Hong, L. S. Wang, J. Zhan, Mining high-utility itemsets based on particle swarm optimization, *Eng. Appl. Artif. Intell.*, **55** (2016), 320–330. <https://doi.org/10.1016/j.engappai.2016.07.006>
29. J. C. W. Lin, L. Yang, P. Fournier-Viger, T. P. Hong, M. Voznak, A binary pso approach to mine high-utility itemsets, *Soft Comput.*, **21** (2017), 5103–5121. <https://doi.org/10.1007/s00500-016-2106-1>
30. J. M. Wu, J. Zhan, J. C. Lin, An ACO-based approach to mine high-utility itemsets, *Knowl.-Based Syst.*, **116** (2017), 102–113. <https://doi.org/10.1016/j.knosys.2016.10.027>
31. K. Thirugnanasambandam, S. Prakash, V. Subramanian, S. Pothula, V. Thirumal, Reinforced cuckoo search algorithm-based multimodal optimization, *Appl. Intell.*, **49** (2019), 2059–2083. <https://doi.org/10.1007/s10489-018-1355-3>
32. R. S. Raghav, K. Thirugnansambandam, D. K. Anguraj, Beeware Routing Scheme for Detecting Network Layer Attacks in Wireless Sensor Networks. *Wireless Pers. Commun.*, **112** (2020), 2439–2459. <https://doi.org/10.1007/s11277-020-07158-9>
33. D. Saravanan, S. Janakiraman, K. Chandraprabha, T. KalaiPriyan, R. Raghav, S. Venkatesan, Augmented Powell-Based Krill Herd Optimization for Roadside Unit Deployment in Vehicular Ad Hoc Networks, *J. Test. Eva.*, **47** (2019), 4108–4127. <https://doi.org/10.1520/JTE20180494>
34. K. Thirugnanasambandam, R. S. Raghav, D. Saravanan, U. Prabu, M. Rajeswari, Experimental Analysis of Ant System on Travelling Salesman Problem Dataset TSPLIB, *EAI Endorsed Trans. Pervasive Health Technol.*, **5** (2019), e4. <https://doi.org/10.4108/eai.13-7-2018.163092>
35. S. Abbaspour, A. Aghsami, F. Jolai, M. Yazdani, An Integrated Queueing-Inventory-Routing Problem in a Green Dual-Channel Supply Chain Considering Pricing and Delivery Period, a Case Study of Construction Material Supplier, *J. Comput. Des. Eng.*, **9** (2022), 1917–1951. <https://doi.org/10.1093/jcde/qwac089>
36. A. Asgari, M. Yari, S. M. S. Mahmoudi, U. Desideri, Multi-objective grey wolf optimization and parametric study of a continuous solar-based tri-generation system using a phase change material storage unit, *J. Energy Storage*, **55** (2022), 105783. <https://doi.org/10.1016/j.est.2022.105783>
37. A. Hasanzadeh, A. Chitsaz, A. Ghasemi, P. Mojaver, R. Khodaei, S. M. Alirahmi, Soft computing investigation of stand-alone gas turbine and hybrid gas turbine–solid oxide fuel cell systems via artificial intelligence and multi-objective grey wolf optimizer, *Energy Rep.*, **8** (2022), 7537–7556. <https://doi.org/10.1016/j.egy.2022.05.281>
38. L. Xuan, G. Chen, W. Zuo, Effective algorithms to mine skyline frequent-utility itemsets, *Eng. Appl. Artif. Intell.*, **116** (2022), 105355. <https://doi.org/10.1016/j.engappai.2022.105355>

39. B. Le, T. Truong, H. Duong, P. Fournier-Viger, H. Fujita, H-FHAUI: Hiding frequent high average utility itemsets, *Inf. Sci.*, **611** (2022), 408–431. <https://doi.org/10.1016/j.ins.2022.07.027>
40. J. M. Luna, R. U. Kiran, P. Fournier-Viger, S. Ventura, Efficient Mining of Top-k High Utility Itemsets through Genetic Algorithms, *Inf. Sci.*, **624** (2023), 529–553. <https://doi.org/10.1016/j.ins.2022.12.092>
41. K. Miettinen, *Nonlinear Multiobjective Optimization*, Norwell: Kluwer, 1999.
42. L. Zhang, G. Fu, F. Cheng, J. Qiu, , Y. Su, A multi-objective evolutionary approach for mining frequent and high utility itemsets, *Appl. Soft Comput.*, **62** (2018), 974–986. <https://doi.org/10.1016/j.asoc.2017.09.033>
43. L. Zhang, P. Luo, E. Chen, M. Wang, Revisiting bound estimation of pattern measures: A generic framework, *Inf. Sci.*, **339** (2016), 254–273. <https://doi.org/10.1016/j.ins.2015.12.036>
44. W. Peng, X. Niu, P. Fournier-Viger, C. Huang, B. Wang, UBP-Miner: An efficient bit based high utility itemset mining algorithm, *Knowl.-Based Syst.*, **248** (2022), 108865. <https://doi.org/10.1016/j.knosys.2022.108865>
45. F. Wei, C. Li, Q. Zhang, X. Zhang, J. C. W. Lin, An efficient biobjective evolutionary algorithm for mining frequent and high utility itemsets, *Appl. Soft Comput.*, **140** (2023) 110233. <https://doi.org/10.1016/j.asoc.2023.110233>
46. P. Fournier-Viger, C. W. Lin, A. Gomariz, T Gueniche, A. Soltani , Z. Deng , et al., The SPMF Open-Source Data Mining Library Version 2, *Proc. 19th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2016) Part III, Springer LNCS 9853* (2016), 36–40. <https://www.philippe-fournier-viger.com/spmf/>
47. E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms a comparative case study, *Proceedings of International Conference on Parallel Problem Solving from Nature*, (1998) 292–301. <https://doi.org/10.1007/BFb0056872>
48. J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, *Knowl.-Based Syst.*, **46** (2013) 109–132. <https://doi.org/10.1016/j.knosys.2013.03.012>



AIMS Press

© 2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>).