mathematics

MDPI

Article

Association Rule Mining through Combining Hybrid Water Wave Optimization Algorithm with Levy Flight

Qiyi He [1], Jin Tu [1], Zhiwei Ye [1,*], Mingwei Wang [1], Ye Cao [1], Xianjing Zhou [2] and Wanfang Bai [3]

[1] School of Computer Science, Hubei University of Technology, Wuhan 430068, China
[2] Wuhan Zhuoer Information Technology Co., Ltd., Wuhan 430312, China
[3] Xining Data Services Authority, Xining 810007, China
* Correspondence: hgcsyzw@hbut.edu.cn; Tel.: +86-189-7159-3240

Abstract: Association rule mining (ARM) is one of the most important tasks in data mining. In recent years, swarm intelligence algorithms have been effectively applied to ARM, and the main challenge has been to achieve a balance between search efficiency and the quality of the mined rules. As a novel swarm intelligence algorithm, the water wave optimization (WWO) algorithm has been widely used for combinatorial optimization problems, with the disadvantage that it tends to fall into local optimum solutions and converges slowly. In this paper, a novel hybrid ARM method based on WWO with Levy flight (LWWO) is proposed. The proposed method improves the solution of WWO by expanding the search space through Levy flight while effectively increasing the search speed. In addition, this paper employs the hybrid strategy to enhance the diversity of the population in order to obtain the global optimal solution. Moreover, the proposed ARM method does not generate frequent items, unlike traditional algorithms (e.g., Apriori), thus reducing the computational overhead and saving memory space, which increases its applicability in real-world business cases. Experiment results show that the performance of the proposed hybrid algorithms is significantly better than that of the WWO and LWWO in terms of quality and number of mined rules.

Keywords: data mining; association rule mining; water wave optimization algorithm; hybrid algorithm; Levy flight

MSC: 68T09

check for updates

Citation: He, Q.; Tu, J.; Ye, Z.; Wang, M.; Cao, Y.; Zhou, X.; Bai, W. Association Rule Mining through Combining Hybrid Water Wave Optimization Algorithm with Levy Flight. *Mathematics* **2023**, *11*, 1195. https://doi.org/10.3390/math11051195

Academic Editors: Weihua Xu, Jinhai Li and Xibei Yang

Received: 12 January 2023
Revised: 22 February 2023
Accepted: 27 February 2023
Published: 28 February 2023

1. Introduction

Significant amounts of data are generated every day in all industries, and the field of data mining has expanded quickly in recent years. Data mining, also known as knowledge discovery, aims to find valuable implicit information in large amounts of data [1]. Association rule mining (ARM) is a powerful data mining technique that has been used in a variety of applications to uncover valuable patterns and relationships in large datasets [2]. ARM has long been in the spotlight as a fundamental strategy and is still used today in a variety of applications including market basket analysis, medical diagnostics, network intrusion detection, and other areas [3,4].

Frequent itemset mining (FIM), as a major phase in the traditional ARM algorithms, mines all possible association rules by first mining all the frequent item sets. However, the large number of possible association rules generated leads to expensive computational overhead. For example, in a network intrusion detection system based on association rules, extracting the features of network traffic and building a rule will be difficult if the network data grow quickly, as FIM might not be able to effectively mine the association relationship between features.

Traditional algorithms to resolve the problems of optimization need enormous computational efforts that are inclined to fail as the scale of the problem rises [5,6], which is time- and space-consuming [7,8]. To overcome this issue, some swarm-intelligence-based FIM methods, such as ant colony optimization (ACO) [9], bee swarm optimization (BSO) [10], bat algorithm (BA) [11,12], cuckoo search (CS) [13], penguin search optimization algorithm (PeSOA) [14], and particle swarm optimization (PSO) [15], have been utilized for ARM. From the perspective of runtime performance, these are far more effective, but the quality of the solutions still needs to be improved.

The water wave optimization (WWO) algorithm is a promising swarm intelligence algorithm that has gained attention from researchers because of its impressive performance in solving optimization problems. It has several advantages compared with other swarm intelligence algorithms, such as fewer parameters, high computational efficiency, and easy implementation. Zheng [16] first proposed the WWO algorithm based on shallow water wave models. As a novel algorithm, WWO has a broad development prospect in optimization problems and engineering applications [17], such as the traveling salesman problem and high-speed rail dispatching [18,19]. Zheng et al. [20] proposed a systematic approach to adapting WWO to specific heuristic algorithms for various combinatorial optimization problems. The proposed approach tested the flow-shop scheduling problem, and the results demonstrated that it is competitive. Yan et al. [21] proposed an enhanced WWO based on the elite opposition-based learning strategy and the simplex method (ESWWO), and the experimental results demonstrated that ESWWO is a practical and useful approach for path planning and function optimization problems. Given the effectiveness of WWO in engineering scheduling problems and path optimization problems, in this paper, it is employed to improve the performance of the primary ARM algorithm. However, WWO tends to get stuck in local optima and converge slowly when there are many transaction items in the datasets. To address this issue, an improved WWO with Levy flight (LWWO) is proposed to achieve better solutions in the global search process.

The wave in WWO makes probabilistic decisions based on the search length of the water wave in the current dimension when generating a feasible solution. The probability distribution used to select the next feasible solution is crucial when solving a locally optimal problem [22,23]. Moreover, the levy distribution has a heavy tail feature, which means that it has a thicker tail than other distributions [24–26]. Therefore, in the levy distribution, the tails have a higher probability of being selected, which gives a variety of solutions for LWWO. Additionally, expanding the variety of solutions increases the likelihood of discovering the optimal solution. Levy flight strategy has been shown to be successful in a variety of swarm intelligence algorithms, such as PSO [27–29], artificial bee colony (ABC) algorithm [30], CS algorithm [31,32], and so on.

To further improve the performance of swarm intelligence, scholars have conducted a lot of research on the hybrid strategy and achieved good results [33,34]. In addition, there are successful cases to prove that WWO has good performance when hybridizing with other algorithms. Rekha et al. [35] designed the water moth search algorithm (WMSA) for training a deep recurrent neural network to detect malicious network activities by combining WWO and moth search optimization (MSO). Zhang et al. [36] proposed an improved sine cosine WWO algorithm (SCWWO), which parallels the sine cosine algorithm (SCA) and WWO in the wave propagation and breaking phases. Experimental results demonstrate that SCWWO significantly improves convergence speed and computational accuracy. In this paper, a novel hybrid strategy on LWWO is proposed to obtain better performance. The basic idea of the hybrid strategy is to combine the characteristics of different algorithms. During the iterative update of the population, the algorithms with the appropriate characteristics are selected according to the different needs of the pre- and post-iteration periods. In this study, seeking appropriate algorithms to combine with LWWO is important to further improve the global search capability of water waves. In recent years, scholars have applied various forms of swarm intelligence to ARM and achieved good performance. Zhou et al. [37] applied ACO to solve the sequential rule mining problem, and interesting temporal association rules were extracted. Khan et al. [38] applied the ARM technique to protein classification, which combines the ARM and supervised classification mechanisms using ACO. According to experimental findings, the classifier performed well by identifying the most precise and brief rules. Heraguemi et al. [39,40] proposed an algorithm based on BA for ARM, and the proposed algorithm performed better than the FP-growth algorithm in terms of computation speed and memory usage. Afshari et al. [41] proposed an efficient approach that benefits from the CS algorithm for hiding sensitive association rules. Based on the effective application of the above algorithms on ARM, in this paper, LWWO is hybridized with the ACO, BA, and CS, respectively.

This paper proposes a novel LWWO-based ARM method with a hybrid strategy for discovering hidden relationships between items in datasets. The proposed method not only fuses the advantages of different algorithms but also facilitates a balance between the exploration and exploitation capabilities of WWO, which allows the algorithm to obtain the global optimal solution to the maximum extent and improve the quality of the mined association rules. The main contributions of this paper are summarized as follows:

(1) Levy flight is used in the WWO algorithm's location update process to improve the variety of solutions and prevent it from being stuck on the local optimal solution.
(2) A hybrid strategy is proposed to help the algorithm balance exploration and exploitation. A hybrid algorithm with greater global search capability was obtained by hybridizing ACO, BA, and CS with LWWO, respectively.
(3) The proposed hybrid LWWO–ARM method generates rules directly. Different from traditional ARM algorithms, the proposed method does not generate frequent item sets, reducing the computational overhead of the algorithm, which improves its applicability in practical cases.

The proposed hybrid LWWO–ARM method is tested on six datasets of different sizes and dimensions, which exhibits significantly better performance than primary WWO in terms of the number and quality of rules.

2. Theory Background

2.1. Association Rule Mining

ARM was first proposed by Agrawal as an important research task in data mining, and is also known as the market-basket problem: In a set of items and sales records, which include information about transactions related to the item, association rules are mined to obtain important relationships between the items. In summary, ARM aims to extract relationships between features of different elements [42]. Similar to functional dependency (FD), ARM expresses important relationships between database attributes. For example, with 100% confidence, an association rule is a constant conditional FD. Many studies have shown that ARM techniques can significantly accelerate the general FD discovery approaches [43,44].

Association rule analysis has been widely used in various industries. It can extract relevant information from complex and large amounts of data, which has been used in the financial industry to predict customer needs and provide more suitable product recommendations [45]. By mining the information data of the enterprise, it can predict the sales situation of the enterprise and provide a more specific development plan for the development of the enterprise. On online e-commerce platforms, intelligent recommendations can be made based on the correlation information of users' browsing behavior, and the correlation between different products can be studied to improve the sales of products. Moreover, in the medical field, some researchers use ARM for the identification of cancer-related genes [46].

Association rules provide an expression ($A \to B$) to present the correlation between features, indicating that if event A has occurred, then event B is also likely to occur [47]. In association rules, there are various indicators to evaluate their quality. The proposed hybrid algorithms based on LWWO consider two standards: support and confidence.

(1) Support

Support is one of the criteria for assessing the quality of a rule; it is a measure of how often an item or a set of items appear in a dataset. The value of support can be calculated using Equation (1):

$$Sup(A \rightarrow B) = \frac{P(A \cup B)}{|N|} \tag{1}$$

where $P(A \cup B)$ is the number of transitions that contain both A and B and $|N|$ is the number of all transactions in the database.

(2) Confidence

Confidence is also a criterion for the quality of association rules; it is a measure of how likely an item or a set of items are to appear together in a transaction. The value of confidence can be calculated using Equation (2):

$$Conf(A \rightarrow B) = \frac{P(A \cup B)}{P(A)} \tag{2}$$

where $P(A)$ is the number of transactions that only contain A.

2.2. Brief Introduction to WWO

WWO is a powerful algorithm that can be used to solve a variety of optimization problems. It is based on the principles of shallow water wave motion, which involves three operations: propagation, wave breaking, and refraction. WWO is computationally efficient and is capable of finding high-quality solutions.

(1) Propagation.

The process of propagation can be seen as moving from deep water to shallow water, and the propagated solution is updated ${X_d}'$ according to Equation (3):

$${X_d}' = X_d + rand_{[-1,1]} \cdot \lambda L_d, d = 1, 2, \ldots, n \tag{3}$$

where $rand_{[-1,1]}$ is a uniformly distributed random number with mean zero and standard deviation one, and L_d is the length of the d th dimension of the search space. If the updated location ${X_d}'$ exceeds the search range, it is randomly reset to a location within the search range.

If $f(x') > f(x)$, ${X_d}'$ replaces X_d in the population, and its height is reset to $h_{\max}$; Otherwise, the height of x is decreased by one and x is retained.

After each generation update, the wavelength λ of each wave x is calculated as in Equation (4):

$$\lambda = \lambda \alpha^{-(f(x) - f_{\min} + \varepsilon)/(f_{\max} - f_{\min} + \varepsilon)} \tag{4}$$

where $f_{\max}$ is the maximum fitness value and $f_{\min}$ is the minimum fitness value in the current population, α is the wavelength reduction factor, and ε is a very small constant to avoid division-by-zero.

(2) Refraction.

When the water wave x propagates many times without improvement, its wave height decreases to $h = 0$. Refraction is performed on it as Equation (5) to avoid search stagnation:

$${X_d}' = {x_d}^* + N(\frac{{x_d}^* + x_d}{2}, \left|\frac{{x_d}^* - x_d}{2}\right|), d = 1, 2, \ldots, n \tag{5}$$

where ${x_d}^*$ denotes the current optimal solution and $N(\mu, \sigma^2)$ denotes the Gaussian random number with mean μ and variance σ^2.

Afterwards, the wave height of $X_d{}'$ is reset to $h_{\max}$, and its wavelength is updated by Equation (6):

$$\lambda^{new} = \lambda \frac{f(x)}{f(x')} \tag{6}$$

(3) Breaking.

Once the WWO algorithm searches for a new optimal water wave $x_d{}^*$, then the wave breaking operation is performed as Equation (7):

$$X_d{}' = x_d{}^* + N(0,1) \cdot \beta L_d, d = 1, 2, \ldots, k \tag{7}$$

where β is the wave breaking parameter. Additionally, $N(0,1)$ is a function that generates a random number with a Gaussian distribution that has a mean of zero and a standard deviation of one. If the wave $X_d{}'$ is better than $x_d{}^*$, $X_d{}'$ replaces $x_d{}^*$ in the population.

2.3. Brief View of Levy Flight

Levy flight refers to a random walk with a heavy-tailed probability distribution of step lengths, which means that there is a high probability of a large change in position during the random walk [48–50]. Abundant studies have shown that the individual behavior of many animals in nature is well-represented by the characteristics of Levy flight. The essence of Levy flight is a method to randomize the step length to simulate flight. Levy flight is a method to represent the levy distribution in random steps, as shown in Equation (8):

$$Levy(s) \sim |s|^{-1-\beta}, 0 < \beta \leq 2 \tag{8}$$

where s is the step length of Levy flight, it can be calculated according to Equations (9) and (10):

$$s = \frac{\mu}{|v|^{1/\beta}}, \mu \sim N(0, \sigma_\mu^2), v \sim N(0, \sigma_v^2) \tag{9}$$

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta) \cdot \sin(\pi\beta/2)}{\Gamma[(1+\beta)/2] \cdot \beta \cdot 2^{(\beta-1)/2}} \right\}^{\frac{1}{\beta}}, \sigma_v = 1 \tag{10}$$

where the parameters $\beta = 1.5$, μ, and ν are both normal distributions and Γ is the standard gamma function.

3. Proposed Approach

3.1. Integrating Levy Flight with WWO

The primary WWO algorithm can deal with low-dimensional unimodal optimization problems with mathematical functions simply and efficiently. However, when the same method is applied to high-dimensional complex optimization problems, the solutions obtained by traditional WWO are not very satisfactory and the computation time is long. To improve the global search and local exploration capabilities of WWO, Levy flight is integrated into the location update process of the algorithm. Levy flight maximizes the diversity of the search space, which ensures that the algorithm updates the water wave positions efficiently, helping the WWO to achieve better search results. Therefore, the WWO position update formula is optimized and can be expressed by Equation (11):

$$x'(d) = x(d) + Levy \cdot \lambda L(d) \tag{11}$$

In brief, the pseudo-code of the LWWO is described in Algorithm 1.

Algorithm 1: Pseudo-Code of LWWO

```
Input: population ($P$) and maximum number of iterations ($T$);
Output: Best solution $x^*$
Randomly Initialize a population $P$ of n waves and the parameters ($\alpha, \beta, \lambda$)
While stop criterion is not satisfied do
  for each $x \in P$ do
    propagate $x$ to a new $x'$ using Equation (3)
    if $f(x') > f(x)$ then
      if $f(x') > f(x^*)$ then
        Break $x'$ using Equation (11)
        Update $x^*$ with $x'$
      Replace $x$ with $x'$
    else
      $x.h = x.h - 1$
      if $x.h = 0$ then
        Refract $x$ to a new $x'$ using Equations (5) and (6)
  Update the wavelengths using Equation (4)
Return $x^*$
```

3.2. The Hybrid Strategy for LWWO

To prevent the algorithm from falling into a local optimum to the greatest extent, it is necessary to apply a method that can explore the entire search space and exploit local regions. An effective mixture of different algorithms takes advantage of different search strategies to balance exploration and exploitation. A hybrid strategy is proposed to improve the global optimization capability of LWWO. ACO, BA, and CS were selected for hybridization with LWWO because they have a strong exploratory capability and have been proven to perform well on ARM in previous studies. Then, hybrid algorithms LWWO–ACO, LWWO–BA, and LWWO–CS were applied to ARM.

In this study, the proposed method comprises four stages: data pre-processing, hybrid algorithms for ARM, rule evaluation, and generation of association rule sets. The overall flow of the hybrid algorithm for ARM is shown in Figure 1. The process of selecting the optimal solution for the hybrid algorithm is illustrated in the dashed box, using LWWO–CS as an example. The proposed hybrid algorithm obtains the current optimal solution by comparing the values of the fitness functions of LWWO and CS.

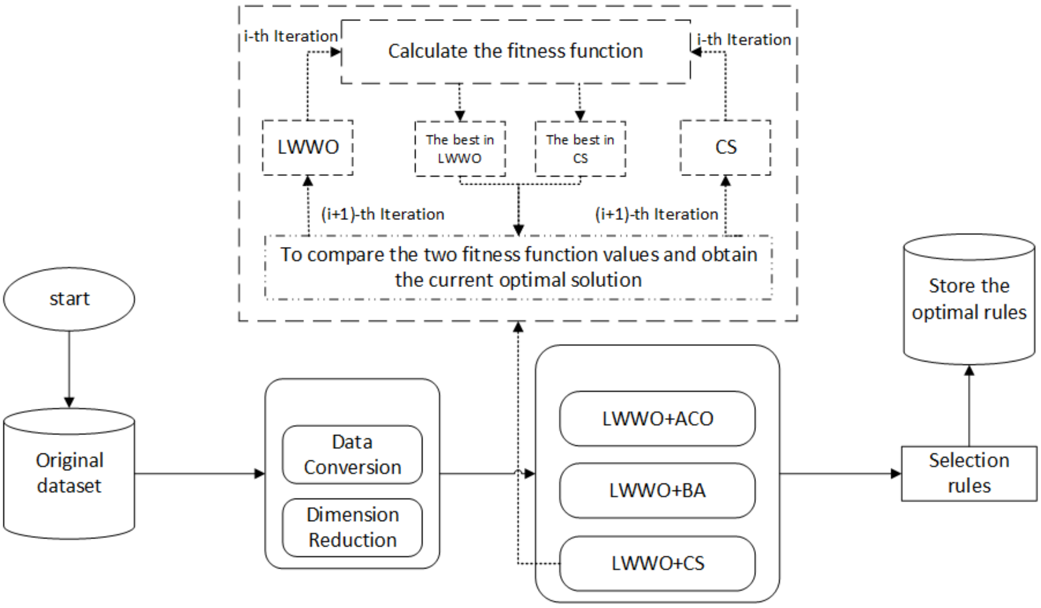


Figure 1. The overall model of the hybrid algorithm for ARM.

In this paper, we combine the strengths of multiple algorithms to compensate for the shortcomings of the primary WWO, such as slow convergence speed and weak ability to jump out of the local optimum. In the whole process, the number of populations is the same for each algorithm, and the total number of iterations is the sum of the iterations of two algorithms. When the iterations of LWWO are complete, the current population is retained and used as the initial population for the next stage.

In general, swarm intelligence algorithms converge slowly during the population initialization phase. Therefore, at the beginning of the iteration, it is advisable to choose an algorithm with good convergence and strong search capabilities to approach the optimal solution. In the middle and late stages of the iteration, individuals of the population tend to cluster around local optimal solutions; thus, algorithms with greater exploratory power are needed.

As WWO is relatively slow to converge in the early stages, the ACO algorithm, which has a good convergence and search capabilities, is chosen to be mixed with it. In the later stage of the iteration, since WWO is weak in jumping out of the local optimum, it is suggested to combine with an algorithm with a stronger ability to jump out of the local optimum, such as the BA and the CS algorithms, which can further search the solution space to find other possible optimal solutions.

The data structure of the proposed algorithm is HashSet, which limits the storage of duplicate elements, making it well-suited for filtering duplicate association rules and storing the individuals in the population that meet the requirements. At the same time, the proposed method uses HashMap to store the rules, their corresponding support and confidence, which are stored in memory as key-value pairs. The proposed hybrid algorithm based on LWWO pseudo-code is provided in Algorithm 2.

Algorithm 2: Pseudo-Code of the proposed hybrid algorithm based on LWWO

```
Input: population and maximum number of iterations, datasets;
Output: rules stored in HashSet, the support and confidence stored in HashMap to an Excel table
Scan the datasets and count the number of attribute items
Convert the transactions in the datasets into a 0-1 matrix
Randomly Initialize the population and the parameters of algorithm A
Iterations = $T$; $T = T1 + T2$; $t = 1$
While $t < T1$ do
    for each individual in the population do
        Compute the fitness values for individuals in the population
        Update the individuals in the population by location update formula of algorithm A
        Binarize the updated individuals using sigmoid activation function using Equation (13)
        Evaluate individuals in the population, deposit the individuals that meet the rules in HashSet, deposit the corresponding support and confidence in HashMap
    end for
    $t = t + 1$
end while
Use the population after algorithm A iterations as the initial population of algorithm B
Initialize the parameters for algorithm B;
while t < T2 do
    for each individual in the population do
        Compute the fitness values for individuals in the population
        Update the individuals in the population by location update formula of algorithm B
        Binarize the updated individuals using sigmoid activation function using Equation (13).
        Evaluate individuals in the population, deposit the individuals that meet the rules in HashSet, deposit the corresponding support and confidence in HashMap
    end for
    $t = t + 1$
end while
Count the execution time of hybrid algorithm and the number of rules searched
```

3.3. The Hybrid Algorithms for Association Rule Mining

The ARM process can be summarized as follows: First, a mapping needs to be created between the records in the dataset; in other words, for existing data records, relationships between attributes are revealed by association rules.

Figure 2 illustrates the entire operation of the hybrid algorithm based on the LWWO generation rules. In the process, the data are first converted into binary format, and then candidate solutions are encoded as individuals in the population. It is important to define a suitable fitness function to assess the quality of each individual. Then, a hybrid algorithm is used to mine the data for association rules and continues the search process until the iteration is complete. Finally, when the evolution of the individuals is complete, the hybrid algorithm outputs strong association rules.

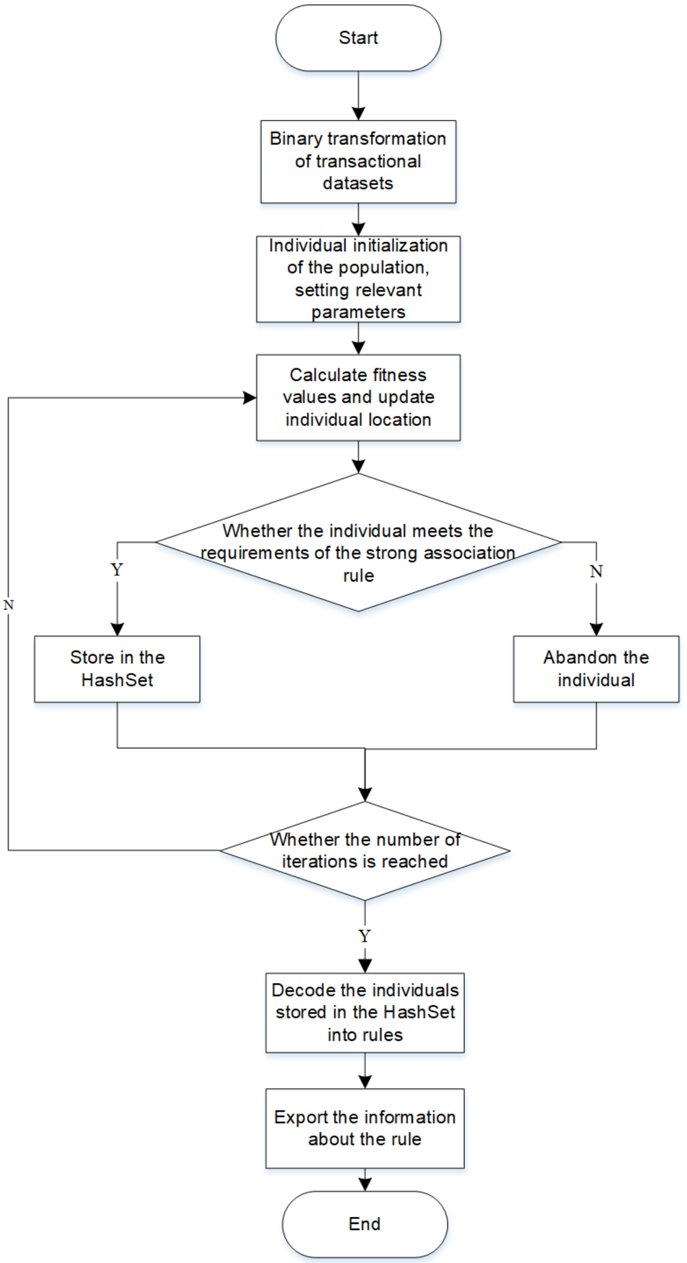
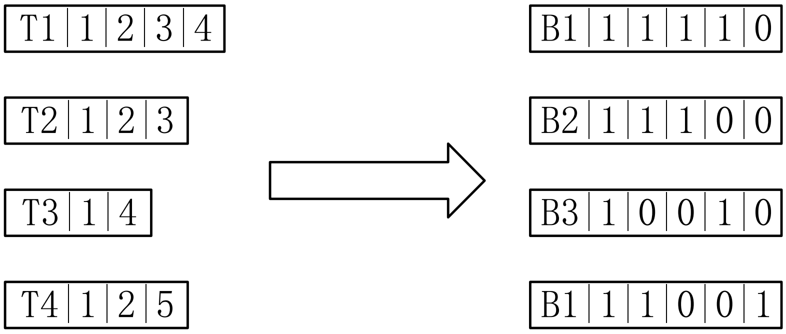

Start
Binary transformation of transactional datasets
Individual initialization of the population, setting relevant parameters
Calculate fitness values and update individual location
Whether the individual meets the requirements of the strong association rule
Y
N
N
Store in the HashSet
Abandon the individual
Whether the number of iterations is reached
Y
Decode the individuals stored in the HashSet into rules
Export the information about the rule
End

Figure 2. Flowchart of the proposed hybrid algorithms for ARM.

3.4. Rule Encoding

In swarm intelligence, each individual in the population represents a possible solution in the solution space. Aiming at the problem of ARM, each individual represents a possible rule. By encoding them, each solution is transformed into a rule. For the algorithms mentioned in this paper, binary encoding is used to encode the raw data. Binary encoding means that the records in a dataset are transformed into a zero–one matrix and all data records are stored in zero or one format, which makes it easy to read the data records and also increases the speed of calculation. The conversion method is shown in Figure 3. For an attribute item present in a transaction, the corresponding position is set to one. Otherwise, it is set to zero.



T1 | 1 | 2 | 3 | 4

T2 | 1 | 2 | 3

T3 | 1 | 4

T4 | 1 | 2 | 5

B1 | 1 | 1 | 1 | 1 | 0

B2 | 1 | 1 | 1 | 0 | 0

B3 | 1 | 0 | 0 | 1 | 0

B1 | 1 | 1 | 0 | 0 | 1

Figure 3. Transaction data binary transformation.

There are two approaches to encoding association rules: in the Michigan approach, a rule represents a solution, while in the Pittsburgh approach, it uses a set of rules to represent a solution. Considering that each individual represents a rule, for the problem of how to distinguish the antecedent from the latter, the proposed algorithm uses the Michigan method, which consists of two attribute terms that jointly represent the transaction terms in a rule; each rule consists of a rule antecedent and a rule consequent.

As an example, the rules generated from the transaction in Figure 3 are encoded as shown in Table 1, where each attribute item consists of two digits, the first digit indicating whether the attribute is included in the rule, and the second digit indicating whether the attribute is the antecedent of the rule or the consequence of the rule. For example, if I_1 is coded as 1–1, it means that the generated rule has the attribute item I_1, and I_1 is in the antecedent of the rule. Thus, this individual can be decoded as a rule $I_1 \rightarrow I_2$.

Table 1. Randomly generated Michigan-coded individuals.

$\mathbf{I_1}$		$\mathbf{I_2}$		$\mathbf{I_3}$		$\mathbf{I_4}$		$\mathbf{I_5}$	
1	1	1	0	0	1	0	0	0	1

After the positional update of population individuals, the code of population individuals is longer coded zero–one, so it is necessary to recode the updated individuals. The updated equation is as in Equation (12):

$$x(t)_i'' = \begin{cases} 1 & r_3 < Sig(x(t)_i') \\ 0 & r_3 \geq Sig(x(t)_i') \end{cases} \tag{12}$$

where r_3 is a random number in the range between zero and one, and $Sig(x)$ is the sigmoid function, as shown in Equation (13):

$$Sig(x) = \frac{1}{1+e^{-x}} \tag{13}$$

3.5. The Fitness Function

Constructing a suitable fitness function is a key step in solving optimization problems and is used to evaluate the quality of the solution. Specifically applied to ARM, support and confidence are important criteria for evaluating whether a rule is valid or not.

Support indicates the probability that all data items in a rule will appear in the dataset at the same time and reflects the level of support for a rule. Confidence reflects the feasibility of a rule. They are used to determine the importance and validity of a rule. We consider combining support and confidence to define the required fitness function. The fitness function is formed as in Equation (14):

$$fitness_i = \alpha \times Support_i + \beta \times Confidence_i \tag{14}$$

where $Support_i$ is the support for randomly generated rules, $Confidence_i$ is its confidence, and α and β determine the weight of support and confidence in the iterative process, respectively. α and β are all in the range (zero, one), and $\alpha + \beta = 1$. If α is set to zero, only the influence of confidence on the rules is considered, and rules with a weak association between the antecedent and the consequent may be mined. If β is set to zero, only the influence of the support on the rules is considered, and rules with very low accuracy may be mined, ignoring rare rules. Considering appropriate weights can balance support and confidence, which has a good effect on mining strong association rules.

In addition, whether a rule is determined to be a strong association rule requires consideration of minimum support and minimum confidence. Finally, whether an individual can be retained as a strong association rule is calculated by Equation (15):

$$f_i = \frac{Support_i + Confidence_i}{Min\_Sup + Min\_Conf} \tag{15}$$

where Min_Sup and Min_Conf are the minimum support and the minimum confidence defined by the user, respectively. If $f_i < 1$, it means that the rule represented by the individual is not strongly associated and should be discarded. Otherwise, it can be retained as a strong association rule, and the rule will be written to the output file.

4. Experimental Results and Discussion

For the purpose of validating the performance of the proposed method, the corresponding computational experiments for hybrid algorithms, WWO, and LWWO implemented on ARM were conducted. Each algorithm was run 30 times on different datasets, and the average results were recorded. All algorithms were written in Java and executed on an Intel Core i7 machine with 16 GB of memory, running on Windows 10. The datasets and parameter settings are explained in detail in the following sections.

4.1. Datasets

The datasets were downloaded from the "LUCS-KDD Discretised/Normalised" database. To test the performance of the algorithms applied on ARM on datasets of different sizes and dimensions, six different datasets were selected for experiments to compare the performance of the five algorithms (WWO, LWWO, LWWO–ACO, LWWO–BA, LWWO–CS). The information of the datasets is shown in Table 2; these data were preprocessed by discretization and dimension reduction and then used in the experiments.

Table 2. Dataset descriptions.

Dataset	Rows	Columns
Iris	150	19
Heart	303	52
Ecoli	336	34
Breast	699	20
Flare	1389	39
Led7	3200	24

4.2. Parameters Settings

As for ARM, the quality of a rule is usually determined by its support and confidence. Through the analysis in Section 3.5, the proposed method uses Equation (14) as the fitness function of the proposed algorithm. The weight of support and confidence jointly determine the direction of population convergence and the search target. We tested the performance of different parameter combinations on a shopping basket dataset. As shown in Table 3, when the support weight (α) is 0.7 and the confidence weight (β) is 0.3, most algorithms perform a relatively large number of association rules and a good quality of the mined rules. Therefore, to ensure the fairness of the experiment, the experiment had the following settings:

- 500 for the total number of iterations (T);
- 60 for population size;
- 0.7 for support weight (α);
- 0.3 for confidence weight (β);
- 0.1 for the minimum support (Min_Sup);
- 0.5 for the minimum confidence (Min_Conf).

Table 3. Different weight of support and confidence tests.

Support Weight (α)	Confidence Weight (β)	Evaluation Indicators	ACO	BA	CS	WWO	LWWO
0.1	0.9	Rule_num	166	1671	1356	775	1665
		Avg_conf	0.9569	0.9497	0.8505	0.9635	0.9497
0.2	0.8	Rule_num	487	1331	736	619	1319
		Avg_conf	0.9852	0.9329	0.9431	0.9557	0.9329
0.3	0.7	Rule_num	387	1573	1672	733	1531
		Avg_conf	0.9564	0.9494	0.9255	0.9449	0.9494
0.4	0.6	Rule_num	214	1834	868	743	1743
		Avg_conf	0.8941	0.9246	0.9506	0.9349	0.9246
0.5	0.5	Rule_num	149	1633	1702	822	1621
		Avg_conf	0.9646	0.9376	0.9583	0.9529	0.9376
0.6	0.4	Rule_num	234	1539	2266	722	1712
		Avg_conf	0.9529	0.9364	0.9167	0.9250	0.9364
0.7	0.3	Rule_num	470	1833	1795	876	1867
		Avg_conf	0.9694	0.9363	0.8562	0.9481	0.9363
0.8	0.2	Rule_num	190	1382	653	786	1368
		Avg_conf	0.9558	0.9133	0.8365	0.9675	0.9133
0.9	0.1	Rule_num	229	1288	451	775	1257
		Avg_conf	0.8904	0.8886	0.9109	0.9635	0.8886

4.3. Evaluation Standard

It is insufficient to use only the number of rules mined as the evaluation standard. Consequently, the quality of the rules obtained should also be used as one of the criteria for determining the efficacy of the algorithms. This paper uses the average confidence and the average support of the rules mined as a supplementary evaluation criterion, to observe the quality of the rules mined by the algorithm. In summary, it was finally determined that the performance of the algorithm was evaluated from four aspects: average mining time (avg_time), average number of rules mined (rule_nums), average confidence (avg_conf), and average support (avg_sup).

4.4. Performance Comparison

Since the Led7 dataset contains the largest number of records, it was used as a sample to count the execution time of each algorithm for different numbers of records, and the results are shown in Table 4 and Figure 4. Overall, LWWO had the shortest execution time, with a reduction of 44.06–51.08% over WWO. This indicates that the Levy flight strategy was effective in improving the search speed of WWO. The time taken by LWWO–CS was closest to that of WWO, while LWWO–BA and LWWO–ACO took more time than WWO. Compared with WWO, when the number of records was 1600, LWWO–ACO and LWWO–BA took 41.27% and 32.82% more time, respectively, and LWWO–CS took only 12.00% more time; when the number of records was 3200, LWWO–ACO and LWWO–BA took 26.33% and 19.66% more time, respectively, while LWWO–CS took only 7.58% more time. This indicates that LWWO–CS has an advantage over the other two hybrid algorithms in terms of execution time.

Table 4. Running time of each algorithm with different data volume (ms).

Number of Records	WWO	LWWO	LWWO–ACO	LWWO–BA	LWWO–CS
400	331	174	418	382	342
800	552	281	767	702	629
1200	791	387	1136	1079	923
1600	1042	552	1472	1384	1167
2000	1271	711	1837	1672	1389
2400	1498	834	2092	1952	1661
2800	1809	976	2388	2218	1937
3200	2085	1137	2634	2495	2243

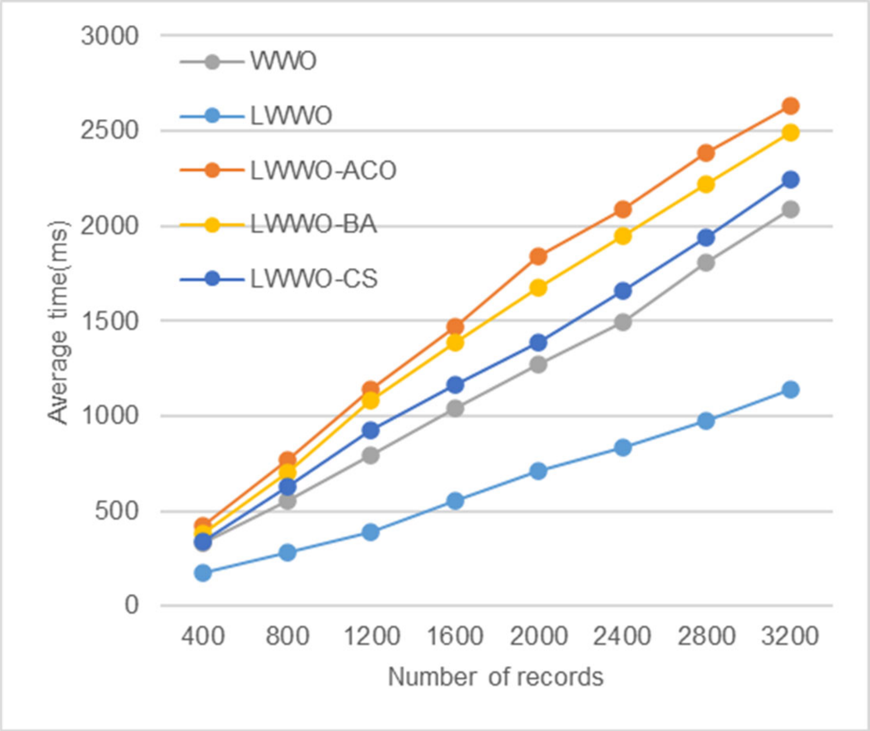


Figure 4. Time consumption of each algorithm with different data volume.

4.5. Comprehensive Evaluation

In this section, the average execution time of each algorithm, the average number of rules mined, the average confidence, and the average support are tested on six benchmark datasets of varying dimensions and sizes, and the detailed statistics are shown in Table 5.

Figure 5 shows a comparison of the average execution times of the five algorithms on different datasets. It can be seen that the execution time of the algorithms increased on the dataset with a higher number of transactions. The Heart dataset had a similar number of transaction records as the Ecoli dataset, but it had a higher dimensionality; the algorithms had a longer average execution time on the Heart dataset. On all datasets, the execution time of the LWWO was 32.28–55.67% shorter than that of the primary WWO. All three hybrid algorithms showed a slight increase in execution time compared with primary WWO. LWWO–ACO took the longest time and LWWO–CS showed some advantages in terms of time consumption.

Table 5. Detailed data of each algorithm on different datasets.

Dataset	Algorithm	Time(ms)	Num	Sup	Conf	Fitness
Iris	WWO	163	39	0.128	0.804	0.601
	LWWO	**78**	53	0.159	0.833	0.631
	LWWO–ACO	236	97	0.252	0.846	0.668
	LWWO–BA	229	**272**	0.248	**0.867**	0.681
	LWWO–CS	193	263	**0.304**	0.856	**0.690**
Heart	WWO	381	81	0.188	0.722	0.562
	LWWO	**258**	113	0.217	0.756	0.594
	LWWO–ACO	692	168	0.226	0.794	0.624
	LWWO–BA	674	**334**	0.224	0.803	0.629
	LWWO–CS	564	242	**0.235**	**0.824**	**0.647**
Ecoli	WWO	194	124	0.303	0.792	0.645
	LWWO	**86**	173	0.325	0.799	0.657
	LWWO–ACO	256	272	0.332	0.816	0.671
	LWWO–BA	237	**513**	0.336	0.815	0.671
	LWWO–CS	201	371	**0.339**	**0.826**	**0.680**
Breast	WWO	586	65	0.307	0.797	0.650
	LWWO	**323**	76	0.332	0.806	0.664
	LWWO–ACO	769	127	0.346	0.818	0.676
	LWWO–BA	756	**391**	**0.365**	**0.823**	**0.686**
	LWWO–CS	622	376	0.359	0.819	0.681
Flare	WWO	822	252	0.127	0.827	0.617
	LWWO	**484**	364	0.156	0.831	0.629
	LWWO–ACO	1336	619	0.172	0.863	0.656
	LWWO–BA	1261	**1099**	0.180	0.904	0.687
	LWWO–CS	1056	781	**0.193**	**0.930**	**0.709**
Led7	WWO	2085	155	0.151	0.786	0.596
	LWWO	**1137**	179	0.168	0.837	0.636
	LWWO–ACO	2634	271	0.197	0.852	0.656
	LWWO–BA	2495	**938**	0.224	0.851	0.663
	LWWO–CS	2143	896	**0.286**	**0.868**	**0.693**

The bolded numbers in the table are the optimal values for that group of experiments.

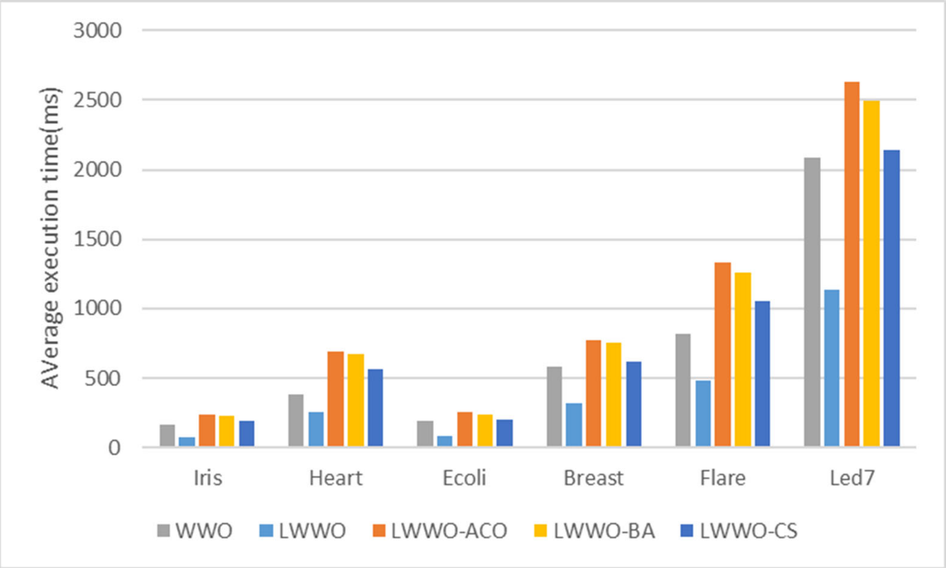


Figure 5. Comparison of the execution time of each algorithm.

Figure 6 shows the comparison of time consumption and the number of rules required by each algorithm on different datasets. Across all datasets, the number of rules mined by LWWO was 15.48–44.44% more than that of WWO, which shows that Levy flight can effectively increase the diversity of solutions. The number of rules mined by LWWO–ACO was 1.75–2.48 times that of WWO, and the time-consumption increased by 26.33–81.63%. The number of rules mined by LWWO–BA was 4.12–6.97 times that of WWO, and the time-consumption increased by 19.66–76.90%. The number of rules mined by LWWO–CS was 2.99–6.74 times that of WWO, and the time-consuming increased by 2.78–48.03%. It shows that the proposed hybrid algorithms have a significant increase in the number of association rules mined compared with WWO, with LWWO–BA showing the largest improvement. This indicates that the proposed hybrid algorithms are able to effectively increase the diversity of solutions and have a strong global search capability. Furthermore, the time-consumption of the hybrid algorithms is increased by a reasonable degree.

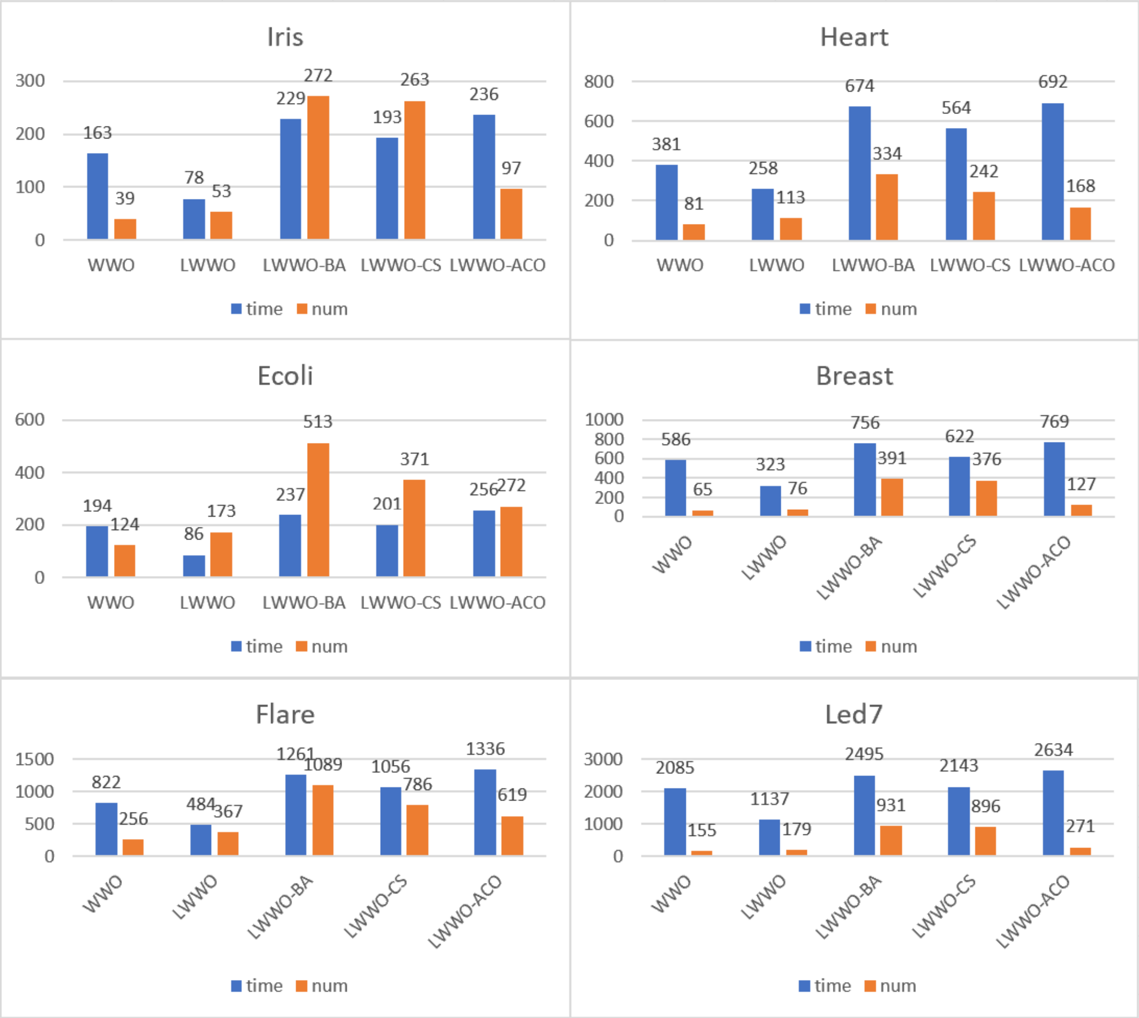


Figure 6. Comparison of average execution time and the number of rules mined.

Figure 7 shows the comparison of the average support of the algorithms. On the six datasets, the improved algorithm had a significantly higher average support compared with WWO. Specifically, the support of LWWO–ACO ranged from 0.172 to 0.346, that of LWWO–BA from 0.180 to 0.365, that of LWWO–CS from 0.193 to 0.359, that of WWO from 0.127 to 0.307, and that of LWWO from 0.156 to 0.332. The average support of all the hybrid algorithms was higher than that of WWO and LWWO, with a more obvious advantage on the Iris and Led7 datasets.

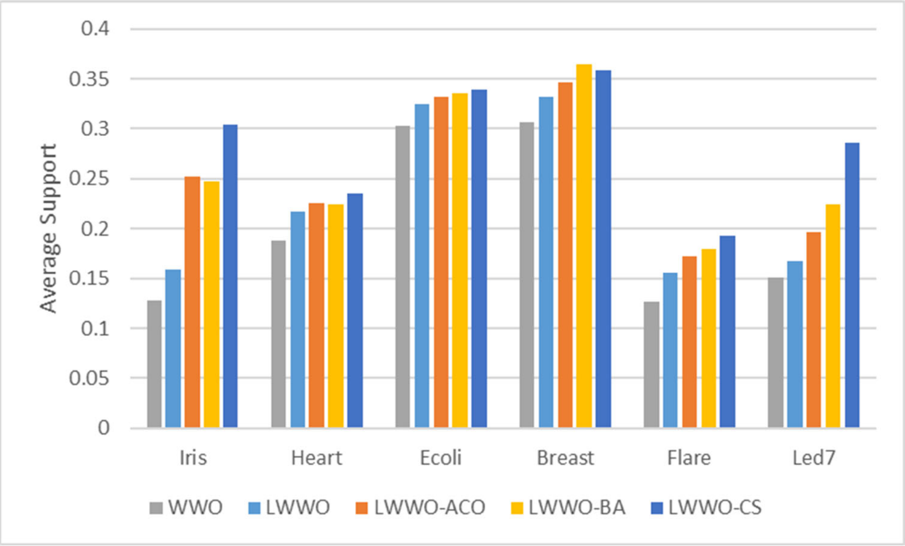


Figure 7. Comparison of average support.

Figure 8 shows a comparison of the average confidence of the algorithms on the six datasets. It indicates that the hybrid algorithms slightly improved the confidence of the association rules mined. Compared with WWO, the average confidence of LWWO–PSO improved by 2.63–9.97%, the average confidence of LWWO–BA improved by 2.90–11.22%, and the average confidence of LWWO–CS improved by 2.76–14.13%. It indicates that the hybrid algorithms also had slightly improved confidence in the association rules, with LWWO–CS performing the best.

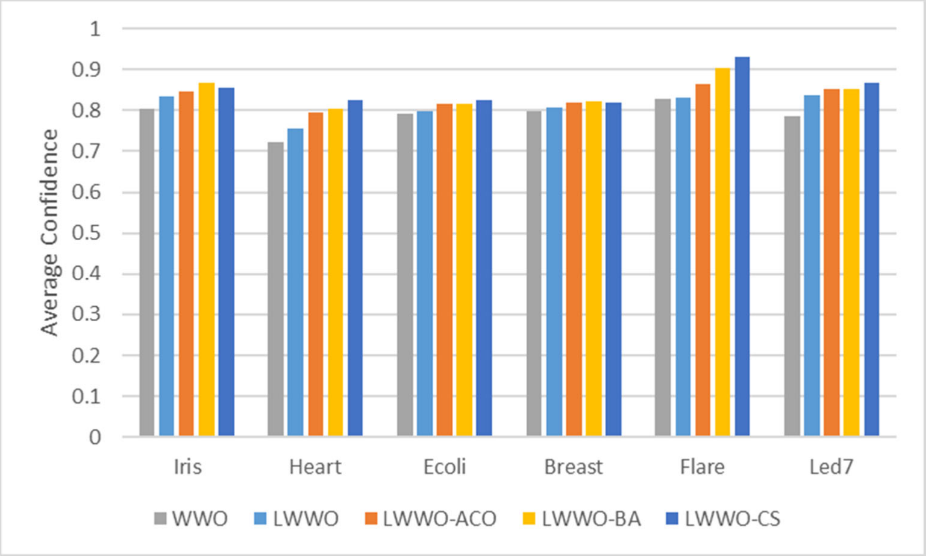


Figure 8. Comparison of average confidence.

Figure 9 shows a comparison of the average fitness for the five algorithms. On all datasets, the average fitness of LWWO–ACO improved by 3.95–11.08%, that of LWWO–BA by 4.03–13.32%, and that of LWWO–CS by 4.77–16.44% compared with WWO. Compared with LWWO, the average fitness of LWWO–ACO improved by 1.81–5.86%, that of LWWO–BA by 2.13–9.22%, and that of LWWO–CS by 2.56–12.72%. It indicates that the quality of the association rules mined by the three hybrid algorithms improved, with a more significant advantage on the Flare dataset; LWWO–CS had the best performance.

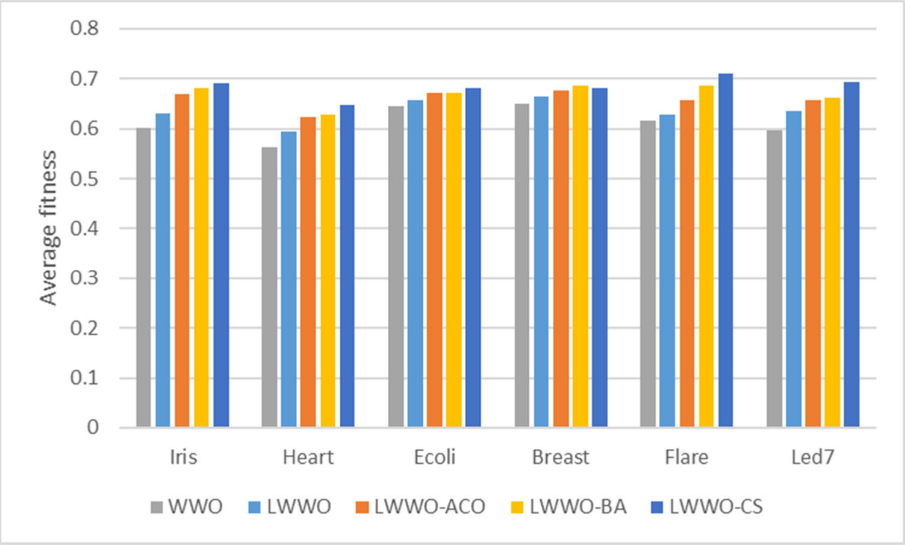


Figure 9. Comparison of average fitness.

By comparing the performance of the algorithms on different datasets through the above graphs, it can be inferred that the proposed hybrid algorithm can obtain strong association rules with higher support and confidence. In addition, LWWO–BA obtained the highest number of rules on all datasets. The best performance on all datasets was LWWO–CS, which had the highest average fitness value and the shortest time consumption among the hybrid algorithms. Compared with WWO, the three hybrid algorithms significantly increased the number of association rules mined and improved the quality of the association rules mined with an acceptable increase in time consumption, indicating that the proposed hybrid algorithms are effective and robust for ARM.

5. Summary and Conclusions

ARM is an important technique for uncovering relationships between different items in large datasets. Therefore, in the era of big data, it is worthwhile to deeply study ARM algorithms. In this paper, three hybrid algorithms, called LWWO–ACO, LWWO–BA, and LWWO–CS, are proposed for ARM. These algorithms combine the search strategies of different algorithms to maximize the global optimal solution in the search process. Experiments on six datasets of various dimensions and sizes show that the proposed hybrid algorithms outperform WWO and LWWO in terms of the number and quality of rules mined. Moreover, LWWO–CS shows the best competitiveness, which indicates the effectiveness of the hybrid strategy on ARM. Furthermore, the proposed ARM approach does not require the generation of frequent item sets, thus improving mining efficiency and maintaining a good balance between rule quality and mining efficiency, making it suitable for practical applications.

So far, in our experiments, we have only focused on discovering association rules between items in the dataset; the specific relationships of the items in the obtained association rules will be further investigated in the future. Another possible direction for our future research is to focus on the similarity of the mined association rules and to filter the redundant rules in the rule sets more efficiently. In future work, it would be interesting to apply the proposed hybrid algorithms to different datasets and compare them with other ARM methods to see how well they work in real-world settings.

Author Contributions: Conceptualization, Q.H. and Z.Y.; methodology, J.T. and Y.C.; software, J.T. and Y.C.; validation, Q.H., M.W. and W.B.; writing, J.T. and Y.C.; review, Z.Y., M.W. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research and publication of the paper were funded by the National Natural Science Foundation of China (No. 42201464) and the Wuhan Science and Technology Bureau 2022 Knowledge Innovation Dawning Plan Project (No. 2022010801020270).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

ARM	association rule mining
WWO	water wave optimization algorithm
LWWO	water wave optimization algorithm with Levy flight
FIM	frequent item sets mining
ACO	ant colony optimization algorithm
BSO	bee swarm optimization algorithm
BA	bat algorithm
CS	cuckoo search algorithm
PeSOA	penguin search optimization algorithm
PSO	particle swarm optimization algorithm
ABC	artificial bee colony algorithm
WMSA	water moth search algorithm
SCWWO	sine cosine water wave optimization algorithm
SCA	sine cosine algorithm
FD	functional dependency
LWWO–ACO	water wave optimization algorithm with Levy flight hybrid with ant colony optimization algorithm
LWWO–BA	water wave optimization algorithm with Levy flight hybrid with bat algorithm
LWWO–CS	water wave optimization algorithm with Levy flight hybrid with cuckoo search algorithm

References

1. Dogan, A.; Birant, D. Machine learning and data mining in manufacturing. *Expert Syst. Appl.* **2021**, *166*, 114060. [CrossRef]
2. Saxena, A.; Rajpoot, V.A. Comparative Analysis of Association Rule Mining Algorithms. In Proceedings of the 2016 International Conference on Inventive Computation Technologies (ICICT), Jaipur, India, 22–23 December 2020.
3. Dhaenens, C.; Jourdan, L. Metaheuristics for data mining: Survey and opportunities for big data. *Ann. Oper. Res.* **2022**, *314*, 117–140. [CrossRef]
4. Cheng, S.; Liu, B.; Ting, T.O.; Qin, Q.; Shi, Y.; Huang, K. Survey on data science with population-based algorithms. *Big Data Anal.* **2016**, *1*, 3. [CrossRef]
5. Karthikeyan, T.; Ravikumar, N. A survey on association rule mining. *Int. J. Adv. Res. Comput. Comm. Eng.* **2014**, *3*, 2278-1021.
6. Fournier-Viger, P.; Lin, J.C.; Vo, B.; Chi, T.T.; Zhang, J.; Le, H.B. A survey of itemset mining. *WIREs Data Min. Knowl. Discov.* **2017**, *7*, e1207. [CrossRef]
7. Luna, J.M.; Fournier-Viger, P.; Ventura, S. Frequent itemset mining: A 25 years review. *WIREs Data Min. Knowl. Discov.* **2019**, *9*, e1329. [CrossRef]
8. Logeswaran, K.; Andal, R.K.S.; Ezhilmathi, S.T.; Khan, A.H.; Suresh, P.; Kumar, K.R.P. A Survey on metaheuristic nature inspired computations used for Mining of Association Rule, Frequent Itemset and High Utility Itemset. *IOP Conf. Series: Mater. Sci. Eng.* **2021**, *1055*, 012103. [CrossRef]
9. Patel, B.; Chaudhari, V.K.; Karan, R.K.; Rana, Y.K. Optimization of association rule mining apriori algorithm using ACO. *Int. J. Soft Comput. Eng.* **2011**, *1*, 24–26.
10. Djenouri, Y.; Drias, H.; Habbas, Z. Bees swarm optimisation using multiple strategies for association rule mining. *Int. J. Bio-Inspired Comput.* **2014**, *6*, 239. [CrossRef]
11. Song, A.; Song, J.; Ding, X.; Xu, G.; Chen, J. Utilizing Bat Algorithm to Optimize Membership Functions for Fuzzy Association Rules Mining. In Proceedings of the International Conference on Database and Expert Systems Applications, Lyon, France, 28–31 August 2017.
12. Heraguemi, K.E.; Kamel, N.; Drias, H. Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies. *Appl. Intell.* **2016**, *45*, 1021–1033. [CrossRef]
13. Mlakar, U.; Zorman, M.; Fister, I., Jr.; Fister, I. Modified binary cuckoo search for association rule mining. *J. Intell. Fuzzy Syst.* **2017**, *32*, 4319–4330. [CrossRef]
14. Gheraibia, Y.; Moussaoui, A.; Djenouri, Y.; Kabir, S.; Yin, P.Y. Penguins Search Optimisation Algorithm for Association Rules Mining. *J. Comput. Inf. Technol.* **2016**, *24*, 165–179. [CrossRef]

15. Krishnamoorthy, S.; Sadasivam, G.S.; Rajalakshmi, M.; Kowsalyaa, K.; Dhivya, M. Privacy Preserving Fuzzy Association Rule Mining in Data Clusters Using Particle Swarm Optimization. *Int. J. Intell. Inf. Technol.* **2017**, *13*, 1–20. [CrossRef]
16. Zheng, Y.-J. Water wave optimization: A new nature-inspired metaheuristic. *Comput. Oper. Res.* **2015**, *55*, 1–11. [CrossRef]
17. Zheng, Y.; Xia, L.; Yu, Q. A method for identifying three-dimensional rock blocks formed by curved fractures. *Comput. Geotech.* **2015**, *65*, 1–11. [CrossRef]
18. Zheng, Y.J.; Zhang, B. A simplified water wave optimization algorithm. In Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 25–28 May 2015.
19. Wu, X.-B.; Liao, J.; Wang, Z.-C. Water Wave Optimization for the Traveling Salesman Problem. In Proceedings of the International Conference on Intelligent Computing, Fuzhou, China, 20–23 August 2015; pp. 137–146.
20. Zheng, Y.-J.; Lu, X.-Q.; Du, Y.-C.; Xue, Y.; Sheng, W.-G. Water wave optimization for combinatorial optimization: Design strategies and applications. *Appl. Soft Comput.* **2019**, *83*, 105611. [CrossRef]
21. Yan, Z.; Zhang, J.; Tang, J. Path planning for autonomous underwater vehicle based on an enhanced water wave optimization algorithm. *Math Comput. Simul.* **2021**, *181*, 192–241. [CrossRef]
22. Yun, X.; Feng, X.; Lyu, X.; Wang, S.; Liu, B. A novel water wave optimization based memetic algorithm for flow-shop scheduling. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016.
23. Zheng, Y.; Zhang, B.; Xue, J. Selection of key software components for formal development using water wave optimization. *J. Softw.* **2016**, *27*, 933–942.
24. Tarkhaneh, O.; Isazadeh, A.; Khamnei, H.J. A new hybrid strategy for data clustering using cuckoo search based on Mantegna levy distribution, PSO and k-means. *Int. J. Comput. Appl. Technol.* **2018**, *58*, 137–149. [CrossRef]
25. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009.
26. Dou, R.; Duan, H. Lévy flight based pigeon-inspired optimization for control parameters optimization in automatic carrier landing system. *Aerosp. Sci. Technol.* **2017**, *61*, 11–20. [CrossRef]
27. Hariya, Y.; Kurihara, T.; Shindo, T.; Jin'no, K. Lévy flight PSO. In Proceedings of the 2015 IEEE congress on evolutionary computation (CEC), Sendai, Japan, 25–28 May 2015.
28. Jensi, R.; Jiji, G.W. An enhanced particle swarm optimization with levy flight for global optimization. *Appl. Soft Comput.* **2016**, *43*, 248–261. [CrossRef]
29. Yan, B.; Zhao, Z.; Zhou, Y.; Yuan, W.; Li, J.; Wu, J.; Cheng, D. A particle swarm optimization algorithm with random learning mechanism and Levy flight for optimization of atomic clusters. *Comput. Phys. Commun.* **2017**, *219*, 79–86. [CrossRef]
30. Aydoğdu, I.; Akın, A.; Saka, M. Design optimization of real world steel space frames using artificial bee colony algorithm with Levy flight distribution. *Adv. Eng. Softw.* **2016**, *92*, 1–14. [CrossRef]
31. Li, X.; Yin, M. A hybrid cuckoo search via Lévy flights for the permutation flow shop scheduling problem. *Int. J. Prod. Res.* **2013**, *51*, 4732–4754. [CrossRef]
32. Yang, X.-S.; Deb, S. Cuckoo search: Recent advances and applications. *Neural Comput. Appl.* **2013**, *24*, 169–174. [CrossRef]
33. Nenavath, H.; Jatoth, R.K. Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. *Appl. Soft Comput.* **2018**, *62*, 1019–1043. [CrossRef]
34. Khalilpourazari, S.; Khalilpourazary, S. An efficient hybrid algorithm based on Water Cycle and Moth-Flame Optimization algorithms for solving numerical and constrained engineering optimization problems. *Soft Comput.* **2017**, *23*, 1699–1722. [CrossRef]
35. Rekha, P.M.; Shahapure, N.H.; Punitha, M.; Sudha, P.R. Water Moth Search Algorithm-based Deep Training for Intrusion Detection in IoT. *J. Web Eng.* **2021**, *20*, 1781–1812.
36. Zhang, J.; Zhou, Y.; Luo, Q. An improved sine cosine water wave optimization algorithm for global optimization. *J. Intell. Fuzzy Syst.* **2018**, *34*, 2129–2141. [CrossRef]
37. Zhou, H.; Hirasawa, K. Evolving temporal association rules in recommender system. *Neural Comput. Appl.* **2017**, *31*, 2605–2619. [CrossRef]
38. Khan, M.A.; Shahzad, W.; Baig, A.R. Protein classification via an ant-inspired association rules-based classifier. *Int. J. Bio-Inspired Comput.* **2016**, *8*, 51. [CrossRef]
39. Heraguemi, K.E.; Kamel, N.; Drias, H. Association Rule Mining Based on Bat Algorithm. *J. Comput. Theor. Nanosci.* **2015**, *12*, 1195–1200. [CrossRef]
40. Heraguemi, K.E.; Kamel, N.; Drias, H. Multi-objective bat algorithm for mining numerical association rules. *Int. J. Bio-Inspir. Com.* **2018**, *11*, 239–248. [CrossRef]
41. Afshari, M.H.; Dehkordi, M.N.; Akbari, M. Association rule hiding using cuckoo optimization algorithm. *Expert Syst. Appl.* **2016**, *64*, 340–351. [CrossRef]
42. Agrawal, R.; Imieliński, T.; Swami, A. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, 25–28 May 1993.
43. Fan, W.; Geerts, F.; Li, J.; Xiong, M. Discovering Conditional Functional Dependencies. *IEEE Trans. Knowl. Data Eng.* **2010**, *23*, 683–698. [CrossRef]
44. Caruccio, L.; Deufemia, V.; Polese, G. On the discovery of relaxed functional dependencies. In Proceedings of the 20th International Database Engineering & Applications Symposium, Montreal, QC, Canada, 11–13 July 2016.

45. Sato, Y.; Izui, K.; Yamada, T.; Nishiwaki, S. Data mining based on clustering and association rule analysis for knowledge discovery in multiobjective topology optimization. *Expert Syst. Appl.* **2018**, *119*, 247–261. [CrossRef]
46. Gakii, C.; Rimiru, R. Identification of cancer related genes using feature selection and association rule mining. *Informatics Med. Unlocked* **2021**, *24*, 100595. [CrossRef]
47. Dorigo, M.; Gambardella, L. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66. [CrossRef]
48. Barthelemy, P.; Bertolotti, J.; Wiersma, D.S. A Lévy flight for light. *Nature* **2008**, *453*, 495–498. [CrossRef] [PubMed]
49. Kamaruzaman, A.F.; Zain, A.M.; Yusuf, S.M.; Udin, A. Levy Flight Algorithm for Optimization Problems–A Literature Review. *Appl. Mech. Mater.* **2013**, *421*, 496–501. [CrossRef]
50. Haklı, H.; Uğuz, H. A novel particle swarm optimization algorithm with Levy flight. *Appl. Soft Comput.* **2014**, *23*, 333–345. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.