Web usage mining with evolutionary extraction of temporal fuzzy association rules

Stephen G. Matthews a,*, Mario A. Gongora b, Adrian A. Hopgood c, Samad Ahmadi b

a Intelligent Systems Laboratory, Department of Engineering Mathematics, University of Bristol, Bristol BS8 1UB, UK
b Centre for Computational Intelligence, Department of Informatics, De Montfort University, Leicester LE1 9BH, UK
c Sheffield Business School, Sheffield Hallam University, Sheffield S1 1WB, UK

Article info
Article history:
Available online 27 September 2013

Keywords:
Fuzzy association rules
Temporal association rules
Evolutionary fuzzy system
Genetic algorithm
Data mining
Analytics
Rule discovery
2-tuple linguistic representation

Abstract
In Web usage mining, fuzzy association rules that have a temporal property can provide useful knowledge about when associations occur. However, there is a problem with traditional temporal fuzzy association rule mining algorithms. Some rules occur at the intersection of fuzzy sets' boundaries where there is less support (lower membership), so the rules are lost. A genetic algorithm (GA)-based solution is described that uses the flexible nature of the 2-tuple linguistic representation to discover rules that occur at the intersection of fuzzy set boundaries. The GA-based approach is enhanced from previous work by including a graph representation and an improved fitness function. A comparison of the GA-based approach with a traditional approach on real-world Web log data discovered rules that were lost with the traditional approach. The GA-based approach is recommended as complementary to existing algorithms, because it discovers extra rules.

1. Introduction

Web usage mining is one type of Web mining [23] that attempts to discover patterns of user behaviours that are recorded in the logs of Web servers as users browse Web sites [10]. In this paper, temporal fuzzy association rules are used for Web usage mining. For example, “On a Friday evening, visitors who viewed history.html for a large amount of time also viewed contact-us.html for a medium amount of time”. Such rules extend traditional Boolean association rules [1] by incorporating temporal and fuzzy quantitative features. The temporal feature of the rule is on a Friday evening, and the fuzzy features are the large and medium descriptions.

Matthews et al. [25] discovered a problem of losing some rules when using traditional methods on synthetic market basket data. Traditional methods follow a two-step process of defining the linguistic labels and membership functions of those labels first, and using them in the mining process. However, the contextual meaning of the linguistic labels can change with events such as seasonal weather, sports games [28], or unforeseen events, e.g., hurricanes [22]. The problem is that although the meaning can change in a temporal period the membership functions remain the same. For example, a low quantity of ice cream sales in summer has a different meaning to a low quantity in winter. The membership function does not accurately define the linguistic label for some temporal periods.

Matthews et al. [25] created a solution that combined the flexibility of the 2-tuple linguistic representation [16] with the search power of a GA. The 2-tuple linguistic representation displaces membership functions laterally along the universe of discourse whilst the linguistic label remains the same. Previous work is improved in this paper by incorporating a graph data structure with an enhanced fitness function. The enhancements enable the approach to work on datasets with real-world complexity in a different domain.

This article is structured as follows: Section 2 provides an overview of related work, Section 3 describes the traditional approach and the original GA-based algorithm, Section 4 introduces enhancements to the GA-based algorithm that was applied to Web log data, Section 5 presents the evaluation of our approach compared with a traditional method, and conclusions are made in Section 6.

2. Related work

The application of Web usage mining has been categorised as either personalised for learning user profiles, or unpersonalised for user navigation patterns [30]. In this paper, we focus on user navigation patterns represented with fuzzy association rules. Web usage mining can be used for the personalisation of web content, pre-fetching and caching, enhancing Web site design, and
customer relationship management in e-commerce [13]. Recent work has also applied similar techniques to those in this paper. GAs have mined sequence rules in Web log data [31] and have also performed subgroup discovery [6]. Fuzzy sets have been used to represent the time spent viewing Web pages for fuzzy association rules [33] and fuzzy sequence rules [19]. The temporal and fuzzy features of association rules that are mined in this paper are now reviewed.

The term temporal is ambiguous, because it can have different interpretations in temporal data mining [26]. In this paper, a temporal association rule expresses associations between items from the same transaction, and that association is repeated (occurs frequently) in multiple transactions of a subset of a dataset. For example, a rule may be present in several transactions, and that rule may occur more frequently on a Friday than any other day of that week. Exhibition periods [21] are temporal patterns that take into consideration the time when items were introduced into the dataset, e.g., new publications in a publications database. Cyclic patterns [27] have rules that occur more frequently in regular periods, such as a rule that occurs every weekend. Temporal patterns with partial periodicity [15] relax the regularity of cyclic patterns, so the rule may not be present in some cycles of the temporal pattern. These types of temporal association rules are intra-transactional, which is different to inter-transactional where rules contain items from several transactions spread over a period of time, such as sequence rules [2].

Quantitative association rule mining extends Boolean association rule mining by discovering rules in quantitative attributes [29]. For example, the time spent viewing a Web page, or the quantities of items sold in a shopping basket. Quantitative association rule mining discretises quantitative attributes into bins. Quantitative association rules suffer from the crisp boundary problem, so fuzzy association rules better deal with unnatural boundaries of crisp intervals [20] and inaccuracies with physical measurements [7]. Fuzzy sets [34] allow the quantities to be described with linguistic terms [35], such as low and high.

The temporal property of not discovering rare fuzzy itemsets [32] is different to our research, because we focus on how the fuzzy sets are defined instead of only the temporal property. Au and Chan [5] also mine fuzzy association rules in temporal partitions of the dataset, and they follow the same two-step process, which can lose rules.

3. Temporal fuzzy association rule mining

Two approaches for mining temporal fuzzy association rules were run on the United States Environmental Protection Agency (EPA) dataset. The purpose is to demonstrate how the flexibility of the 2-tuple linguistic representation approach can help to discover rules on real-world data that a traditional approach cannot. The two approaches are described here, and enhancements to the GA-based approach are explained in Section 4.

3.1. FuzzyApriori

FuzzyApriori [18] is an extension to the Apriori algorithm [1] that uses a breadth-first search. FuzzyApriori uses fuzzy sets to express quantities of items with linguistic terms, but it does not consider any temporal pattern. So, the dataset is partitioned according to its temporal dimension, such as by hour, and FuzzyApriori is executed on each dataset partition separately. The systematic search of the temporal dimension allows for the discovery of temporal features of fuzzy association rules. This is similar to the first approach for mining cyclic association rules [27] where the dataset is also partitioned according to the temporal dimension. The rules mined from each dataset partition are aggregated into a final rule set, which is the end result.

Due to the static nature of membership functions in existing approaches, not all temporal fuzzy association rules can be discovered, hence some are lost. Au and Chan [5] also mine fuzzy association rules in temporal partitions of the dataset, which has been discussed in Section 2. Au and Chan [5] use a different search method in the two-step process, but in theory the same problem of losing rules exists, because the fuzzy sets are defined first and they are static. For this reason, a method based on the seminal Apriori algorithm is only compared, i.e., FuzzyApriori.

3.2. CHC with 2-tuple linguistic representation

The GA-based approach was first described in Matthews et al. [25], so an overview is given before introducing enhancements in Section 4. The pseudocode of the algorithm is described in Section A. The GA-based approach by Matthews et al. [25] is not considered to be traditional like FuzzyApriori, because it is not an exhaustive search method. Instead, a stochastic search method is applied – a GA called Cross-generational elitist selection, Heterogeneous re-combination, and Cataclysmic mutation (CHC) [12]. The contextual change of meaning for linguistic labels is modelled with the 2-tuple linguistic representation, which is a flexible representation. The crucial difference from other temporal fuzzy association rule mining approaches is that Matthews et al. [25] simultaneously search for membership function parameters and the items in the rule, as well as the temporal period when the rule occurs. This overcomes the problem of membership functions remaining the same when there is a contextual change in the meaning of linguistic labels. Alternative GA-based approaches that simultaneously search for fuzzy sets and rules do exist, but they perform different tasks, i.e., control [17], classification [36], and fuzzy modelling [11]. The GA-based approach uses Iterative Rule Learning (IRL) [14]. IRL represents one rule in a chromosome. One rule is used from the final population of a GA. More rules are learnt by repeating the GA and penalising previously learnt rules in the fitness function.

4. Enhanced temporal fuzzy association rule mining

The GA-based approach is extended with an enhanced fitness function. A weight in the fitness function provides a preference-based multi-objective model to overcome confidence dominating the fitness [25]. Previous approaches also use Pareto-based multi-objective models [24], however, selecting a single rule from the Pareto front (for IRL) is a challenging problem. A chromosome C has mixed types, and is defined as $C = (c_1, c_2, c_3, \ldots, c_n)$ where the lower temporal endpoint is $e_l$ (start of time window), the upper temporal endpoint is $e_u$ (end of time window), i is the uniform resource locator (URL), $s$ is the linguistic label expressing the page view time for that URL (e.g., medium), $a$ is the lateral displacement of that linguistic label, $i$ determines the antecedent/consequent part, and $k$ is the number of URIs in a rule. For example, a chromosome $(807127200,807130800, "Rules.html","medium",0.42,antecedent)"/""medium",0.31, consequent"$ represents the rule "IF view time of/Rules.html is (medium,0.42) THEN view time of/is (medium,0.31) during the period 807127200 to 807130800 (unixtime). A single rule is represented and extracted from a chromosome, because the lateral displacements of a fuzzy set are specific to each rule.

The fuzzy support count of a chromosome C in a single transaction $t_i$ is defined from Hong et al. [18] as

$$FuzSupTran(C|t_i) = \min_{a_k} \{f_{t_i}(a_k)\}$$

(1)
where \( \mu \) is the degree of membership for a linguistic label \( s_k \) and lateral displacement \( x_k \) for item \( i_k \) with a rule of length \( k \) and for one transaction \( t_j \) where \( j \) is a dataset transaction ID (TID). The minimum is used for intersection of all the clauses, which is the same method of intersection used in FuzzyApriori.

FuzSupTran is then used to calculate fuzzy support counts across multiple transactions and the fitness is defined as

\[
\text{Fitness}(C) = \left( \sum_{j=1}^{N} \text{FuzSupTran}(C_{X}^{(b)} \cap C_{Y}^{(b)}) \right) \frac{e}{e_t} + w \left( \frac{\sum_{j=1}^{N} \text{FuzSupTran}(C_{X}^{(b)} \cap C_{Y}^{(b)})}{\sum_{j=1}^{N} \text{FuzSupTran}(C_{X}^{(b)})} \right),
\]

where \( C \) is a chromosome, \( X \) is the rule antecedent, \( Y \) is the rule consequent, \( j \) is a dataset TID from the \( e_l \) lower endpoint to the \( e_u \) upper endpoint, and \( w \) is a weight applied to the confidence measure. Hence, \( C_{X}^{(b)} \) and \( C_{Y}^{(b)} \) are the rule antecedent and the rule consequent respectively for one transaction in a dataset. A weight is required to avoid local minima that occur as a result of the magnitude of confidence being higher than the magnitude of temporal fuzzy support; a GA produces high confidence values [25,3] compared with a smaller magnitude of support values. For example, a temporal fuzzy support value of 0.001 is smaller than a confidence value of 0.1, so the confidence value has more influence than the temporal fuzzy support. The weight was determined from multiple runs of the algorithm so that the temporal fuzzy support and weighted confidence had the same order of magnitude.

We also extend our previous work with a graph representation to enhance the efficiency of searching for URLs/items in a rule. Our previous approach allowed the generation of invalid chromosomes with rules that did not exist in the dataset. Such rules are detrimental to the search process, and are undesirable in the final rule set.

The dataset is transformed from rows and columns to a cyclical undirected graph. The purpose is to ensure that chromosomes contain valid itemsets that are present in the dataset and also to reduce the itemset search space. The tabular representation is used for fitness evaluation, and the graph representation is used during initialisation and crossover.

An undirected graph \( G \) is a pair of finite sets \( (V,E) \) where \( V \) is a non-empty set of vertices and \( E \) is a set of pairs \( (e,t) \). Each pair in \( E \) consists of an edge \( e \) and a non-empty finite set of TIDs \( t \). Each edge \( e \) is an unordered pair of vertices \((a,b)\). The definition extends regular graphs by including a set of TIDs for each edge.

An example is presented to demonstrate the construction of the graph. Table 1 is a small example of a quantitative dataset transformed into the graph of Fig. 1. Each edge represents the co-occurrence of two items. Items are vertices. The TIDs of the co-occurrence are also on an edge. Edges are paired with a set of TIDs to identify the co-occurrence of items. If there is no set of TIDs for an edge then an edge does not exist.

The vertices for the example graph are \( V = \{A,B,C\} \), and the edges are \( E = \{((A,B),(1,3)),((B,C),(1,2)),((A,C),(1,4)),((A,A),(1,3,4)),((B,B),(1,2,3)),((C,C),(1,2,4))\} \). A loop connects a vertex to itself. These edges are shown in Fig. 1 as lines that loop to the same vertex, i.e., TIDs \( \{1,3,4\} \) for vertex \( A \), TIDs \( \{1,2,3\} \) for vertex \( B \), and TIDs \( \{1,2,4\} \) for vertex \( C \).

The graph representation is incorporated into initialisation and crossover of chromosomes. The algorithms are defined in Appendix A. The algorithm HybridCrossover prevents crossover from producing invalid itemsets. Potential offspring are first checked to identify if the resulting itemsets are present in the specific temporal partition of the dataset. If the resulting offspring are not present then the items are not swapped. Algorithm CheckGraph in Appendix A uses the graph data structure to ensure offspring are valid itemsets in a temporal period.

### 5. Evaluation

The dataset and methodology for analysing the enhanced GA-based approach are discussed and results are then presented.

#### 5.1. Data

A Web log dataset has both temporal and quantitative features. The temporal feature is the timestamp of a request made to the server, and the quantitative feature is the page view time in seconds.

The EPA dataset is a collection of Hypertext Transfer Protocol (HTTP) requests to a Web server collected from a 24-h period. The geographical location of the Web server is Research Triangle Park, NC, USA. The EPA dataset was recorded from 23:53:25 29th August 1995 EDT to 20:53:07 30th August 1995 EDT. The EPA dataset has 47748 requests: 46014 GET requests, 1622 POST requests, 107 HEAD requests, and 6 invalid requests. Table 2 shows a sample of records from the EPA dataset before cleaning and preprocessing.

The EPA dataset was cleaned by removing all records assumed to be the Web site’s design or a non-traversable Web page (suffixes: gif, xbm, zip, pdf, exe, gz, wpd, wp, dct, jpg, and imf). All records that did not have a GET request method were removed. After preprocessing, there were 2688 transactions and 5147 URLs. Preprocessing consisted of:

1. A 10-min time window was used [9], which assumes that visitors do not view the page for more than 10 min.
2. Maximal forward reference transaction identification [8] produced lists of URLs, which are referred to as transactions.
3. Some resulting transactions contained the same URLs next to each other. This is likely to be caused by refreshing the Web page, so subsequent occurrences of the same URL were removed.
4. Two URL requests are required to determine the page view time of a URL by calculating the difference in timestamps. For example, history.html accessed at 12:00:10 and contact-us.html accessed at 12:00:30 has a view time of

#### Table 1

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>13</td>
</tr>
</tbody>
</table>

\(^1\) Available from The Internet Traffic Archive (http://ita.ee.lbl.gov/).
20 s. Transactions with 2 or fewer URLs were removed to ensure that a page view time can be calculated.

5. For FuzzyApriori, the dataset was partitioned by hour.

5.2. Methodology

The methodology for evaluating the approaches on the real-world EPA dataset is described. The aim of the evaluation is to identify whether the GA-based approach can discover rules that the traditional approach cannot. This is a novel approach to traditional methodologies because the focus is on discovering lost rules on a real-world dataset, which is a new unrecognised problem that warrants a different methodology [25]. The methodology for evaluation is the same as Matthews et al. [25], but the analysis of the results is simplified to improve clarity.

The linguistic labels and membership functions are defined first. The traditional and GA-based approaches were run using the same linguistic labels and membership functions. The result was two sets of temporal fuzzy association rules: one set containing the traditional fuzzy set representation, and the other containing the 2-tuple linguistic representation. The two sets of rules were compared to identify rules that matched and rules that did not match.

The method of rule comparison from IRL was used [25]. Each clause of the rule is compared. If the items/URLs and linguistic labels of two clause match, then the lateral displacements are compared. The lateral displacements are considered to be the same if the difference in absolute values of lateral displacements is less than a lateral displacement threshold of 0.5. For example, for a lateral displacement threshold of 0.5 and two lateral displacements, −0.45 and −0.05, the absolute difference is 0.4 so the fuzzy sets are considered to be the same.

5.3. Results

The two approaches for discovering temporal fuzzy association rules were run and the rules were compared. Results of the comparison and an example of a lost rule are presented here. The algorithms were implemented in Java within the KEEL tool (Knowledge Extraction based on Evolutionary Learning) [4]. The experiments were conducted on a personal computer with a 64-bit 2 GHz dual-core processor and 3 GB RAM. FuzzyApriori had a minimum temporal fuzzy support of 0.0011 and a minimum confidence of 0.5. Rules are discarded because their measures fall below either the minimum temporal fuzzy support or the minimum confidence. The minimum confidence was not set high, so that rules are not discarded because of low confidence when the rules have high temporal fuzzy support. The reason for the minimum confidence value is that the temporal fuzzy support is a key factor in a temporal pattern. Furthermore, the minimum confidence was increased from 0.05 in our previous approach [25]. The population size was 50, and the PCBLX crossover operator parameter was 1 [25]. IRL was configured to produce the same percentage of rule lengths as FuzzyApriori. For example, if FuzzyApriori produced 50% with length 2 and 50% with length 3, then IRL produced 50% with length 2 and 50% with length 3.

<table>
<thead>
<tr>
<th>Measure</th>
<th>CHC</th>
<th>FuzzyApriori</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Rules</td>
<td>100</td>
<td>782</td>
</tr>
<tr>
<td>Arithmetic mean of temporal fuzzy support (4 s.f.)</td>
<td>0.0039</td>
<td>0.0017</td>
</tr>
<tr>
<td>Arithmetic mean of confidence (4 s.f.)</td>
<td>0.6078</td>
<td>0.7918</td>
</tr>
<tr>
<td>Execution time (minutes)</td>
<td>1422.65</td>
<td>1.52</td>
</tr>
</tbody>
</table>

Fig. 2. Boxplot of temporal fuzzy support for FuzzyApriori (FA) and one run of CHC.

Initially, the GA-based approach was run once to assess a typical run. Table 3 shows statistics of both approaches. The GA-based approach was limited to 100 rules and the systematic search with FuzzyApriori discovered 762 rules. The arithmetic mean of temporal fuzzy support was higher for the GA-based approach, but it had a lower confidence value. The differences in distributions of both measures are shown in Figs. 2 and 3. The GA-based approach takes longer, but this is outweighed by its benefit of discovering rules with higher temporal fuzzy support.

The evolution of the best fitness for one run of CHC (one iteration of IRL) is shown in Fig. 4. It can be observed that the best fitness increases through the generations. The large jumps in best fitness are likely to be caused by a change in nominal data (e.g., item, linguistic label) in the chromosome rather than interval data (e.g., lateral displacement).

The method of evaluation from Matthews et al. [25], which is stated in Section 5, is now used. The purpose is to identify what rules the GA-based approach can discover that a traditional approach cannot discover in Web log data. The experiments that follow were conducted by running the GA-based approach 30 times and the percentages are arithmetic means of all runs. Tables 4,5 show that 49.23% of the 100 rules discovered with the GA-based approach were also discovered with FuzzyApriori. Of those
49.23% from the GA-based approach, 2.93% had a reduction in temporal fuzzy support and 46.30% had an increase in temporal fuzzy support. When analysing confidence of the same 49.23% of rules, 9.20% had a reduction, 0.30% did not change and 39.73% had an increase. The GA-based approach has rediscovered 49.23% of rules that were already discovered by FuzzyApriori. However, the quality increased significantly for the majority of these rules. Over half the rules (50.77%) were only discovered with the GA-based approach. These 50.77% were not discovered by FuzzyApriori because they fell below one or both thresholds.

It is important to understand which of the 50.77% rules in the EPA dataset are now above the thresholds. When a rule is above both thresholds, it is deemed to be significant. Table 6 shows the percentage of rules that were below the threshold(s), and the percentage of rules now above the threshold(s). Rules now above the thresholds are significant to this research because the GA-based approach has learnt the lateral displacement of membership functions so that a rule is now above the threshold(s).

The results are reported using percentage, which is a relative measure of the 100 rules. It is important to note that increasing the number of rules in IRL may not discover more lost rules than those discovered in the 100 rules. In such case, the percentage would decrease. However, lost rules are still discovered, and it may only be one rule that is of great significance/interest to the user.

An example of a temporal fuzzy association rule is presented below. The rule was not discovered with FuzzyApriori, because the temporal fuzzy support of 0.0005 was below the threshold of 0.0011, and the confidence of 0.44 was below the threshold of 0.5.

Endpoints (unixtime): 807127200–807130800
Rule: IF view time of / is (medium, –0.49) THEN view time of / is medium
Temporal Fuzzy Support: 0.0005
Confidence: 0.44

The same example rule was discovered with the GA-based approach, as shown below, but with lateral displacements from the 2-tuple linguistic representation. The rule demonstrates knowledge that was lost with a traditional approach, but learnt with the GA-based approach.

### Table 4

<table>
<thead>
<tr>
<th>Arithmetic mean of change in temporal fuzzy support</th>
<th>–ve (%)</th>
<th>0(%)</th>
<th>+ve (%)</th>
<th>Total (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHC and FA</td>
<td>2.93</td>
<td>0.00</td>
<td>46.30</td>
<td>49.23</td>
</tr>
<tr>
<td>CHC only</td>
<td>0.00</td>
<td>0.00</td>
<td>50.77</td>
<td>50.77</td>
</tr>
</tbody>
</table>

Further experiments were conducted with CHC and FuzzyApriori on a synthetic market basket dataset to examine scalability and parameter settings. Preliminary results showed that increasing the number of transactions from 10,000 to 90,000 transactions (same as Web site visitors) produced a constant number of lost rules and execution time is linear. And, increasing the number of items from 1000 to 5000 items (same as URL) decreased the number of lost rules and the execution time is linear. Full experimentation and statistical analysis on more real-world examples are subject of our ongoing future work.

### 6. Conclusions

We have demonstrated the problem of losing temporal fuzzy association rules on real-world Web log data for the first time and presented a novel solution. Our previous approach of using a GA and the 2-tuple linguistic representation has been improved by transforming the dataset to a graph, which ensures valid itemsets are discovered, and modifying the fitness function.

The execution time of the GA-based approach is longer, however, the contribution to knowledge is that it can discover rules that a traditional algorithm cannot, and the rules have higher temporal fuzzy support. The GA-based approach is recommended as complementary to existing algorithms, because it discovers extra rules that a traditional algorithm does not. The decision to use this complementary approach can rely on understanding what temporal changes may be present in the application domain (e.g., seasonal and/or scheduled events).

It is important to note that lowering minimum support/confidence would overcome the problem of losing rules with traditional approaches, however, the number of rules increases, which is undesirable in association rule mining. Further work will explore different enhancements, and different approaches to tackle the same problem.
Algorithm 1. IRL with CHC

Begin
While maximum number of iterations not reached do
  Generate initial population
  Evaluate initial population and initialise L
  While maximum number of fitness evaluations not reached do
    Select individuals from parents
    Recombine individuals to form offspring
    Evaluate offspring
    Combine offspring with parents, and select the best N individuals for the next population.
    If there are no new individuals, or the best chromosome does not change, then $L = L - 1$.
    If $L < 0$ then reinitialise the population
  End (While)
  Add best rule to final rule set
End (While)

Algorithm 2. HybridCrossover

Inputs:
$k$— Length of rule;
$p$— Two parent chromosomes;

Outputs:
$O$: Two offspring chromosomes

Begin
  $n$— 0; initialise loop variable to first index
  $O$— $p$; Create offspring from identical copies of parents;
If offspring have matching items Then
  Move clauses, containing matching items, to same loci;
End (If)
While $n < k$ do// Loop through every index in rule
  If $0$ items are identical AND linguistic labels are identical Then
    Uniform crossover of lateral displacement using PCBLX-parcentric BLX-c;
    Uniform crossover of antecedent-consequent parameter using swap;
    End (If)
  If $0$ items are identical AND linguistic labels are not identical Then
    Uniform crossover of [linguistic label, lateral displacement] using swap;
    Uniform crossover of antecedent-consequent parameter using swap;
    End (If)
  If $0$ items are not identical Then
    If 0 is present in endpoints of $O_2$ using Algorithm
      CheckGraph AND $O_2$ is present in endpoints of $O_1$ using Algorithm
      CheckGraph Then
        Uniform crossover of [item, linguistic label, lateral displacement] using swap;
        Uniform crossover of antecedent-consequent parameter using swap;
      End (If)
  End (If)
  $n$— $n + 1$; Increment loop variable
End (While)

Algorithm 3. CheckGraph

Begin
$p$— Clause index of candidate item $j$;
While $n < k$ do// Loop through every index in itemset
  If index of candidate item not equal to current index
    If $n = p$ Then
      If $T$ used for first time
        If $n = 0$ OR $p = 0$ Then
          Initialise $T = M_{i,j}$ that are $> e_1$ and $< e_2$;
        Else
          $T = T \cap M_{i,j}$; Update transaction IDs set with transactions containing candidate item $j$ and current item ($I_0$)
        End (If)
      End (If)
    If $T = \emptyset$ Then return FALSE;
    Else return TRUE;
  End (While)

References


