Journal of Theoretical and Applied Information Technology
29th February 2020. Vol.98. No 04

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HYBRID ANT COLONY OPTIMIZATION AND ITERATED LOCAL SEARCH FOR RULES-BASED CLASSIFICATION

HAYDER NASER KHRAIBET AL-BEHADILI[1], KU RUHANA KU-MAHAMUD[2], RAFID SAGBAN[3]
[1]Computer Science Department, Shatt Alarab University, Iraq
[2]School of Computing, Universiti Utara Malaysia, Malaysia
[3]Department of Software, Faculty of Information Technology, University of Babylon, Iraq
E-mail: [1]haider872004@gmail.com, [2]ruhana@uum.edu.my, [3]rsagban@uobabylon.edu.iq

ABSTRACT

This research presents the ILS-AntMiner rules-based algorithm, a hybrid Iterated Local Search and Ant Colony Optimization, to improve classification accuracy and the size of the classification model. This hybridisation aims to enhance the classification performance in both accuracy and simplicity by increasing the profit of neighbourhood structures in the exploitation mechanism. The experimental results in this research are compared with the most related ant-mining classifiers, including ACO/PSO2 and ACO/SA across various datasets. The results indicate that the proposed classification algorithm can effectively search the training space based on multiple structures to escape from local optima and achieve high classification accuracy and model size.

Keywords: *Data Mining*, *Rule Discovery, Ant-Miner, Metaheuristics, Swarm Intelligence.*

1. INTRODUCTION

Data mining is the process of unveiling hidden insights from data. It is considered by different institutions and companies as the most important opportunity to raise revenue. Data mining is widely used in various fields, such as medicine, science, recognition, business and engineering [1]. In data mining, two types of learning are available, namely, supervised techniques and unsupervised approaches of learning [2]. Unsupervised learning techniques discover patterns from data without any previous knowledge of the data (i.e., unlabelled) [3]–[5]. Conversely, supervised learning techniques use labelled data to build the data mining model [6]–[8] Such techniques can be considered a powerful approach with an accurate and rapid result in a wide range of applications (e.g., businesses). One of the supervised learning techniques to gain significant attention is the rules-based classification which extracts classification rules from the data. One of the prominent ant colony optimisation (ACO) algorithms used for rules-classification is the Ant-Miner variant [9], [10]. The Ant-Miner produces a comprehensive classification model by finding a list of classification rules in the form of (IF <*term1*> AND <*term2*> AND … <*term-n*> THEN <class> from the data. The advantages of these rules can be easily translated in the natural language.

The Ant-Miner introduced by [11] is inspired by the foraging behaviour of an actual ant colony. The Ant-Miner is a metaheuristic, swarm-based, stochastic and separate-and-conquer approach. It consists of three major stages, namely, construction rule, pruning rule and updating pheromone.

In the rule-building stage, each ant begins to add terms to be included in the rule. The term acts as a particular duo (attribute and value) from the attribute in the dataset, and each term can be added only once under the building rule. The Ant-Miner classifier adds terms that increase classification performance according to its pheromone concentration and amount of information.

In rule pruning, the overfitting problem can be avoided by decreasing the length and increasing the simplicity of the constructed rules. The procedure removes one term at a time whilst enhancing quality. The pruning repeats until improvement ceases. The pheromone update has two main stages, namely, updating the pheromone amount for all terms in the current rule on the basis of its quality and updating all terms that do not appear in the current rule.

The Ant-Miner classifier also suffers from premature exploitation because of the absence of any local search in its structure. The Ant-Miner was not designed to explore the neighbourhoods of the current rule and does not consume more time in improving it iteratively. The neighbourhood structures are not fully covered. In this way, this type of search is over-explorative because it is either a single neighbourhood structure movement as exemplified in [12] or does not profit from local

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

search at all [13]. Therefore, various neighbourhood structures can be developed to catapult the search to another point which gives the possibility to exploit the neighbourhood completely. This study proposes a hybridized Iterated Local Search (ILS) with Ant-Miner classifier algorithm for more mature exploitation.

The remainder of this research is structured as follows. Section 2 discusses the literature review of hybridization between ACO and local search algorithms. Section 3 illustrates the proposed hybrid algorithm. Section 4 discusses the experimental results of the proposed method. Section 5 provides conclusions and suggestions for possible future research.

2. RELATED WORKS

Combinatorial Optimisation problems arise in various application domains, such as scheduling problems, routing problems, network routing and data classification. These problems have simple definitions and complex solutions [14]. The algorithms used to solve such problems are classified into exact and approximation algorithms. The exact algorithms always ensure the discovery of the optimal solution. With the large instance size of the problem, the run time often increases abnormally and, in some cases, the exact algorithms produce unsatisfactory solutions for hard optimisation problems. The approximation algorithms aim to introduce a trade-off between solution quality and run time. These algorithms can efficiently determine the optimal (approximate) solution for hard optimisation problems [15].

The approximation algorithms can be classified into two groups: heuristic and metaheuristic algorithms. Heuristic algorithms have two categories: constructive and local search algorithms. Constructive algorithms build the solutions from scratch, whereas local search algorithms search initial solutions and iteratively use the neighbourhood search for improvement. The high-level framework and strategy that guide the local search and constructive algorithms are the metaheuristic algorithms [16].

Metaheuristic algorithms are high-level strategies or general heuristics designs that explore the search space and search for a high-quality solution to an optimisation problem. Some well-known metaheuristic algorithms include ACO, Simulating Annealing (SA), Tabu Search (TB), Genetic Algorithm (GA) and ILS. Metaheuristic algorithms can be classified into two groups: constructive or local search based (iterative).

Constructive metaheuristic algorithms build the optimal solution from their constituent elements. The greedy algorithm concept is often used to supplement the best available element to the solution. The iterative metaheuristic algorithms determine the optimal solution by iteratively replacing the current solution with its neighbourhood to achieve further improvement [17]. The ACO belongs to the former [18], whereas the others belong to the latter. In recent works, ACO has been determined to use stochastic movements and memory in the search process and accept solutions worse than previously. These features will allow ACO to explore large search spaces without becoming trapped in the local minima.

However, ACO suffers from a local optimisation problem, i.e., the searching strategy is restricted to a global search only, which produces vulnerable solutions. Local search algorithms are fast and efficient. They start with an initial solution and spend a long time to achieve improvement using the available neighbourhood space. However, local search algorithms cannot escape from local minima. Hence, hybridisation between the ACO and local search algorithm is required to solve the disadvantages of both algorithms. The ACO is used to explore the new area within the search space and produce the initial solution, whereas local search algorithms use this solution to utilise and improve the search in the neighbourhood space. Many studies have reported the improvement of hybridisation between ACO and local search.

A recent work adopted the hybrid elitist ant system variant of the ACO algorithm for DM clustering tasks. This study combined the elitist ant system with the ILS algorithm and tuned the importance of the constraints' parameter for each dataset. The goal of this application is to introduce an elitist ant system that can utilise the search space in the clustering domain. The experiment was conducted using six medical benchmark datasets from UCI. The computational results show that the proposed algorithm has produced solutions with higher quality than other methodologies [19].

Lin, Dai, Contreras, and Zhang in [20] combined ACO with a 1-opt local search procedure for forest transportation planning problems (FTPPs). The proposed mechanism is introduced to improve the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

solution quality further obtained by ACO. The 1-opt local search procedure is applied at the end of the iteration to search for potential shortcuts at each vertex and its adjacent vertices along the individual origin–destination routes to improve the quality. The performance of the proposed approach was tested in a hypothetical FTPP and then tested using 10 FTPP instances generated using the same FTPP. Afterwards, the proposed method was applied in a real large-scale FTPP. The results show that the proposed approach can match the feasible solutions for all cases of all FTPP tests.

This study combined elitist–AS, a variant of ACO, with the ILS algorithm on the well-known symmetric traveling salesman NP-hard problem. The case study involved different instances of TSP, including two instances that consist of 26 cities and 1094 locations in Jordan. This research aims to maintain the elitist–AS exploitation mechanism by using ILS to intensify the search around the elite solution. The results display that the proposed hybridisation can produce optimal results compared with other algorithms [21].

A successful hybridisation between ACO and ILS algorithms for the forecasting of Turkey's domestic electricity consumption has been achieved. This research aimed to combine the advantages of both algorithms to perform forecast estimations. The evaluation performance was conducted using different economic indicators, such as population, export, GDP and import, on Turkey's data sets between 2004 and 2013. The evaluation results exhibited high quality compared with the actual and predicted electricity consumption datasets. Therefore, the proposed hybridisation is used to forecast Turkey's domestic electricity consumption until 2030 [22].

Ezzat et al. [23] have proposed an extension of the EigenAnt ant colony system (EAAS) algorithm called probabilistic EAAS (PEAAS). The algorithm is applied to the sequential ordering problem (SOP), which is used to model real-world vehicle routing and transportation problems. The enhancement has been applied using two procedures. Firstly, the stochastic degree in the solution construction process was increased. Secondly, the SOP-3-exchange local search procedure was added to enhance all solutions in the solution construction steps. The results indicate that PEAAS is better than the original EAAS and state-of-the-art enhanced ACS (EACS) algorithms.

Liao et al. [24] introduced an incremental ant colony algorithm with local search (IACORLS) to solve a continuous optimisation problem. IACORLS is an improved version of the ACOR algorithm with an extra procedure for search diversification, which consists of a growing solution archive, followed by the hybridisation between the ACOR algorithm and the local search procedure to increase the intensified capabilities. The experiment used three local search procedures: Powell's BOBYQA, Lin-Yu Tseng's Mtsls1 and Powell's conjugate directions set. The results were tested using two benchmark functions in SOCO and the special session on continuous optimisation. The results show that the combination of the ACOR algorithm and Mtsls1 obtains the best results amongst the possible combinations. Moreover, IACROLS is considered one of the best continuous optimisation algorithms.

Drias et al. [25] proposed a hybridisation of ACO and TS in medical web information foraging. The framework of this study consisted of two main steps. The first phase performed learning on several web pages' instances for localising the related pages of interest to the user. The second phase considered the dynamicity and openness of the web by using the results obtained from the first step. Then, the changes that occurred on the web were observed. The experiment used the website of the US National Library of Medicine (MedlinePlus), which consists of 1903 web pages for 900 diseases. The results from this experiment were promising from two aspects: response time and strong web relatedness.

Eswaramurthy and Tamilarasi in [26] presented the global optimisation of the hybrid ACO with TS in job shop scheduling problems. The aim of this coupling is to utilise pheromone trails and dynamic tabu in selecting the best available neighbours to enhance the quality of the solution. This coupling helps escape from the local optima, avoids cycling and accelerates the convergence of the search process. The performance was tested using two matrices: quality of solution and CPU run time. The results show that the proposed algorithm performs better than well-known algorithms.

Another study introduced different approaches based on the hybridisation between ACS and local search procedures in different optimisation problems. This study was conducted by Gambardella in a PhD thesis [27] on proposed approaches published with other co-authors. The

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

first approach consisted of parallel stochastic ACS coupling with restricted three-opt procedure as a local search procedure in a TSP optimisation problem. The study showed that ACS provides good starting solutions for local search optimisers. The experiment results display that ACS performs better than other nature-inspired algorithms, i.e., evolutionary computation and SA. In addition, this paper was the second most cited in IEEE Transactions on Evolutionary Computation. The second approach proposed the coupling between multiple ACS with CROSS exchanges as local searches for time window vehicle routing optimisation problems. This study introduced two colonies: the first reduces the number of vehicles, whereas the second reduces the travel distances, subsequently sharing information between colonies through pheromone updating. The performance of this approach was tested in terms of quality and execution time, showing that it is comparable with well-known methods. The third approach highlighted other couplings between ACS and a new local search mechanism to work with a sequential ordering optimisation problem called the SOP-3-exchange. This approach introduced the SOP-3 exchange local optimiser, which was designed for the sequential ordering problem regarded as the contribution of this study. The experiment's evidence shows that the implementation of the local search greatly improves ACS performance. The performance results show that the proposed approach is better than existing methods in the sequential ordering optimisation problem.

Another approach was proposed by Gambardella, Montemanni and Weyland in [28] by using hybrid ACS with strong local searches for three transportation optimisation problems. The proposed approach aims to improve the performance of the ACS algorithm in the constructive stage and integrates it with a strong local search optimiser. In this mechanism, the local search runs if and only if the obtained solution in the constructive stage is within 20% of the best quality solution to accelerate the ACS algorithm. The evaluation results display that EACS obtains a higher quality of solutions than classical ACS methods. Finally, the ACS algorithm was applied with local search procedure in different real-world vehicle routing problems. The performance in these real problems showed that ACO is one of the best and most successful metaheuristics in vehicle routing optimisation problems.

Guan and Lin in [29] proposed the hybridisation between ACO and variable neighbourhood search for single-row facility layout as an optimisation problem. The enhancement approach modified the original ACO in different directions. The first modification, conducted in the construction stage, proposed new objective function formulas to compute the values in different neighbourhoods. A feedback criterion based on edit distance was measured to avoid local optima. Then, a new pheromone updated the strategy on the basis of the best and worst solutions introduced. In addition, an extra variable neighbourhood search procedure was added as a local search mechanism, which uses the first improvement via three different neighbourhood structures to change in a random search. The experiment results show that the proposed algorithm is comparable with other algorithms in single-row facility layout optimisation problems.

De La Cruz, Paternina-Arboleda, Cantillo and Montoya-Torres [30] combined the ACS algorithm with TS in the heterogeneous vehicle routing problem with multiple products and time windows, namely, HVRPTWMP. This approach is based on two pheromone strategies to accelerate the ants in the searching process. In addition, HVRPTWMP uses two characteristics, namely, recent and frequent event memories and diversification. This approach utilises regency and frequency memories from TS to search for further improvement in the quality of solutions. Meanwhile, diversification is applied using the best solution of the current neighbourhood instead of the standard TS that uses the best solutions to guide search diversification. In the second stage, the standard TS stores the best solution explicitly, whereas this approach stores the attributes. The performance uses instances from the literature and shows that the quality of the proposed approach is comparable with the best-known solutions from the literature.

Sagban, Ku-Mahamud and Abu Bakar in [31] proposed overcoming the problem of the exploitation in ACO when hybrids are included in local searches. This approach applies two mechanisms to solve this drawback, namely, recursive local search method and reactive heuristics based on information recorded in two auxiliary memories in the search process to guide the search in the future. The first auxiliary memory indicates the arcs with low pheromone density during the pheromone update procedure at each step for further significance. The recursive local

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

search mechanisms use population-based memory to handle the solution that has been generated in the previous populations and used iteratively to utilise the neighbourhood of the current solution. These proposed techniques were implemented in the MMAS variant of ACO in two optimisation problems: TSP and the Quadratic assignment problem (QAP). The results were evaluated with different ACO variants coupled with local search procedures, displaying that the enhanced algorithm is better than six ACO variants.

Saian and Ku-Mahamud [12] proposed Ant-Miner coupled with SA to generate a list of classification rules. In the Ant-Miner, each ant discovers a rule. This study proposed SA as a local search procedure to improve this rule iteratively. The SA works for each rule on the basis of the temperature variable, which starts at a high value and then decreases on the basis of predefined factors. The search runs a certain number of iterations and selects the best rule from the available neighbourhood depending on its quality. Using the SA mechanism, which starts with high temperatures, allows the rule with low quality to be selected. Then, the temperature will be decreased, and the difference between the current and previous qualities will be crucial for selecting the rule with high quality. The performance matrix is indicated on the basis of the rule quality, the number of discovered rules and the terms per rule. The performance of this approach was tested using 13 datasets from the UCI repository, showing that it is comparable with the original Ant-Miner in predicative accuracy.

A hybridisation of ACO and ILS algorithms based on new pheromone-update rules on TSP and job shop scheduling NP-hard problems was proposed. The proposed pheromone-update rules are as follows: (1) rank-based and (2) IB/GB pheromone updates. The experiment results carried out on 16 TSP problems indicated that the hybrid ACO/ILS-rank-based method outperforms the non-hybrid rank-based one. Similarly, the hybrid ACO/IB/GB method outperforms the non-hybrid IB/GB-based method. In addition, the experimental results on job shop scheduling for FT06, FT10 and FT20 problems were promising [32].

Ji et al. [33] combined the ACO and GA as a local search procedure. The Ant-Miner algorithm generates the list of rules in each iteration and selects the best rule to be added in the final discover rule list. The proposed mechanism uses the best rule and the mutation mechanism used in GA to mutate the attribute value one at a time and then test the quality of the newly generated and original rules if and only if the quality increases. As such, the new rule will replace the original one. The experiment was conducted using six datasets from the UCI repository, showing slight improvement in the quality and comprehensibility of the rule.

In recent studies that have been reviewed, two types of local search procedure include hybrid with ACO. The first type is the heuristic method based on the iterative exploitation of neighbourhoods to improve the current solution by local changes [20], [23], [27]. Although the solution quality increases with a large number of neighbourhoods, these methods take exponential time. The second type, namely, metaheuristics, also uses strong local search, which increases the performance of heuristic methods by guiding the search with high-level principles and strategies (i.e., based on objective function, memory and prior performance) [19], [21], [31]–[34], [22], [24]–[30]. These algorithms overcome the drawbacks of the heuristic methods in a further intelligent way, which allows the local search to escape from local minima by generating a new initial solution or allows the worsening moves rather than simply providing random starting solutions. Overall, the results show that the local search plays a crucial role in improving the quality of the solution in different ACO variants.

The metaheuristic algorithms have been inspired by naturally occurring phenomena. The inspiration is embodied into some computational characteristics. The characteristics of metaheuristic algorithms include whether to use a single point or population of search. Another important feature of metaheuristics is using memory to guide the direction of future searches. In addition, some metaheuristic algorithms use single- or multiple-neighbourhood structures. The majority of local search algorithms use a single-neighbourhood structure that determines the type of allowed moves. This type of move is used in the TS and SA algorithms. Conversely, the ILS uses multiple-neighbourhood structures (N and N′).

In the ILS, the local search procedure starts with the neighbourhood of N solution until reaching local optimum. Perturbation then catapults the search to another point. For subsequently repeating the local search, the primary neighbourhood is used. The GA uses a mutation operator, which has the same effect as perturbation. The crossover

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

operator moves in hyper-neighbourhoods. Unlike the local search algorithm, the solution construction in ACO is not based on a specific neighbourhood structure. Table 1 shows the main characteristics of metaheuristic algorithms. '√' means the feature is present, and '–' indicates otherwise.

Table1: Characteristics of Metaheuristic Algorithms

Characteristics	ACO	SA	TB	GA	ILS
Population	√	-	-	√	-
Memory	√	-	√	-	√
Multiple neighbourhoods	-	-	-	√	√
Constructive	√	-	-	-	-
Iterative	-	√	√	√	√

Recent studies have shown that the performance of ACO improves when hybridisation is performed with a local search algorithm. The hybridisation must consider the characteristics of metaheuristic algorithms, as shown in Table 1. These characteristics must be studied deeply to determine which algorithmic component is to be integrated. For example, the multiple-neighbourhood structures of ILS are used to guide the search for new regions. An in-depth investigation is needed where local search improves the classification task; otherwise, it will not have any additional performance improvement. In addition, the original Ant-Miner and its variants do not use any local search to explore more terms and edges, except the two studies in the literature which also have drawbacks. In the first study, Ji et al. in [33] combined the Ant-Miner and GA as a local search procedure by using only a mutation mechanism to mutate one attribute value of the best rule found in the iteration; if and only if the quality improves, the new rule will replace the original one. The combination is only used in the mutation operator; it will not have a high-quality solution whilst it is not able to explore and does not iteratively improve the neighbourhoods of the current solution. In a further study, Saian and Ku-Mahamud in [12] coupled the Ant-Miner with SA as a local search procedure. Each ant in each colony uses the SA to discover one rule. The SA works for each rule on the basis of the temperature variable, which starts with a high value then decreases on the basis of pre-defined factors. The search runs a certain number of iterations and selects the best rule from available neighbourhoods on the basis of quality. The move results in minimal improvement because the local optimisers do not forbid the current neighbourhood structures from moving to the next iterations. Thus, the Ant-Miner algorithm will consume additional time in improving the same search regions to determine the best improvement. To the best of our knowledge, no research has been conducted on the Ant-Miner hybrid and ILS algorithm as a local search. The diversification of the Ant-Miner and the intensification of ILS can be coupled to balance local exploitation and global exploration via multiple-neighbourhood structures. In addition, restricting the ILS to the best iteration will save computation time whilst providing a strong possibility to improve the best solution obtained by the ants.

3. PROPOSED HYBRID ALGORITHM

The overall goal of hybridisation of the Ant-Miner with ILS (ILS-AntMiner) is to benefit from its characteristics to form the neighbourhood structures. In ILS, there is a strong inter-correlation between exploration (e.g., perturbation) and exploitation components (e.g., local search). Furthermore, ILS has succeeded in employing perturbation in exploiting multiple neighbourhood structures. Figure 1 shows how ILS has utilised perturbation to improve the search in two of its neighbourhood structures in the proposed hybridisation.

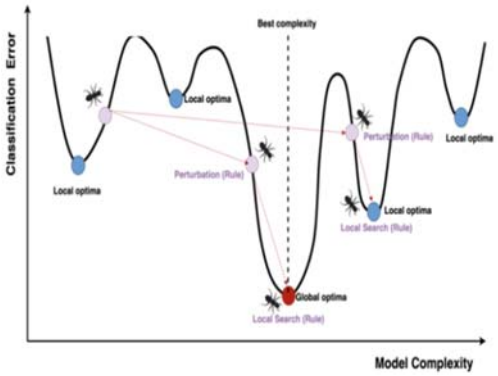

Figure 1: Proposed Hybrid Search Strategy

The ILS-AntMiner classifier starts with all training instances to discover one classification rule from the data. The discovered rule is then inserted in the rules list, in which every instance covered by the rule are removed from the training instances set. These operations stop when all the instances in the training dataset are lower than the pre-specified constant values known as *Max uncovered cases*. This approach has three main procedures, namely, rule building, pruning rule and updating pheromone. The initial procedure is the construction rule, where every ant starts to add

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

terms to be included in the rule. The ant adds one term at a time to improve the classification accuracy according to its probability value. The probability of the term selection to be added to the current rule is provided by Equation (1) [35] as follows:

$$\text{Probability} = \frac{\left[\tau_{ij(t)}\right]\left[\eta_{ij}\right]}{\sum_{i=1}^{a} xi \cdot \sum_{j=1}^{bi}\left[\tau_{ij(t)}\right]\left[\eta_{ij}\right]} \quad (1)$$

where $\left[\tau_{ij(t)}\right]$ is the amount of pheromone concentration for each term at iteration (t); $\left[\eta_{ij}\right]$ is the problem depending upon heuristic function; *a* is the attribute number in the dataset; *bi* is the number of different values for each attribute; and *xi* is set to 1 (if the attribute is not yet visited by the ant) and 0 (otherwise). In addition, the heuristic function value is used together with the pheromone value to decide on the term selection. In the ILS-AntMiner, the heuristic function is inspired by information theory. The ILS-AntMiner computes the amount of information contained in each term (entropy). The heuristic function is given by Equations (2) and (3) [35], as follows:

$$\eta_{ij} = \frac{log_2\, k - H(W|A_i = V_{ij})}{\sum_{i=1}^{a} xi \cdot \sum_{j=1}^{bi} (log_2\, k - H(W|A_i = V_{ij}))} \quad (2)$$

$$H(W|A_i = V_{ij}) = -\sum_{w=1}^{k}\left[\frac{P(W|A_i = V_{ij})}{T_{ij}}\right] * log_2\left[\frac{P(W|A_i = V_{ij})}{T_{ij}}\right] \quad (3)$$

where *w* is the class attribute, and k is the number of classes; $P(W|A_i = V_{ij})$ is the partition containing the instances, where attribute *Ai* has value *Vij* with class *w*; |*Tij*| is the total number of instances in partition *Tij* (instances where attribute *Ai* has value *Vij*); *a* represents the total number of attributes, and *bi* is the number of values in the attribute i.

This process is repeated, while certain attributes are not used yet; or the pre-specified minimum number of instances is not covered by the current rule. Once the rule is completed, the classifier chooses the consequence part of the rule by assigning the majority class among the instances covered by the rule.

The discovered rule will then undergo the pruning procedure, which aims to avoid the overfitting problem by reducing the size of the discovered rules to increase comprehensibility. The procedure prunes one term at a time as the rule quality improves. The procedure loops until no more improvement occurs or only one term is left under the rule. The class value of the dataset can potentially change during this procedure because the majority of the classes in the cases covered by rule pruning may change compared with cases covered by the original rules.

The pheromone is updated after rule construction and prune procedures. The approach of pheromone updates has two basic steps. Firstly, increasing the amount of the pheromone in all terms appears in the rule according to rule quality by Equations (4) and (5) [35].

$$\tau_{ij(t+1)} = \tau_{ij(t)} + \tau_{ij(t)} \cdot Q \quad (4)$$

$$Q = + \frac{TP}{TP+FN} * \frac{TN}{FP+TN} \quad (5)$$

where *TP* is the number of instances covered by the discovered rule and has the class predicted by the rule; *FN* is the total number of instances covered by the discovered rule and has a class different from the class predicted by the rule; *TN* is the total number of instances not covered by the discovered rule and does not have the class predicted by the rule; and *FP* is the total number of instances not covered by the discovered rule but has class predicted by the rule.

Secondly, evaporating each term does not occur in the rule by normalising unused terms. Another ant then builds its rule derived from the updated amount of pheromone. The process is completed based on the following stopping conditions being satisfied. In the first condition, the number of discovered rules must be equal to the number of ants. The second condition is according to the number of the rule convergence that is statically determined, where the ant starts to converge by building a rule similar to that previously constructed. The best among all construction rules will be added to the list of discovered rules.

To speed up the search in the proposed ILS-AntMiner, ILS is only applied for the iteration-best rule instead of applying it for all constructed rules so as to ensure the algorithm stays lightweight. The change that may be applied to the classification rule is defined by a neighbourhood structure. A neighbourhood structure is a function $N: S \rightarrow 2^S$ that assigns to every $s \in S$ a set of neighbourhood $N(s) \subseteq S$. $N(s)$ is also called the neighbourhood of *s*.
The procedure starts with training data instances to discover one classification rule. The rule will be added to the discovered-rule-list. This process will

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

stop when all instances in the dataset are less than pre-specified parameter values, which are known as max_uncovered_cases. The best rule in the discovered-rule-list will be treated by ILS local search. Therefore, the best discovered rule will be the kernel to form the neighbourhood structure.

The ILS-AntMiner is a multiple neighbourhood structure algorithm. Each iteration, the perturbation generates a new starting rule where the local search can be applied. To ensure that the perturbation is enough to escape from the local optima, the size of movement should be larger than local search. Perturbation strength depends on the number of classification rule components that needs to be modified. A strong perturbation leads to the modification of several terms and the structure of the current classification rule may be lost. Thus, the size of perturbation is twice the size of local search movement which is a four-terms exchange. The rule solution is represented as an integer, one-dimensional array that has a size equal to the number feature of the dataset. Each bit in the array is associated with a feature, and each has a different integer number of terms (data dependence). These constraints will not allow the array bits to change easily as the random movement will not be able to explore other terms that do not appear in the current rule. In addition, the constraint by the features and their possible integer number of terms can lead to non-existent solutions.

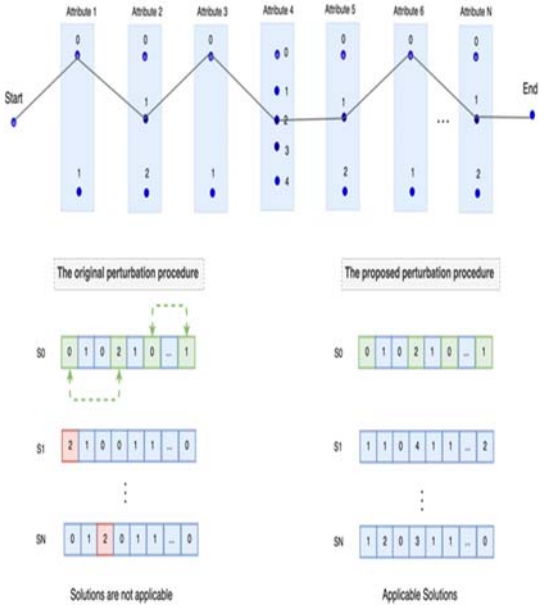

Figure 2: Four-Terms Exchange Perturbation

Figure 2 shows how the original perturbation of swapping between the values of bit 1 and bit 4 will lead to non-existent solutions (note that the value of bit 4 (Attribute 4) = 2 is not identified in bit 1 (Attribute 1)). Thus, the perturbation procedure has been modified to select four features from the rule. The perturbing moves were subsequently chosen within the neighbourhood structure of the selected feature for the current classification rule.

To guarantee a fully matured exploitation for such promising search regions, the size of local search movement in ILS-AntMiner is set to *two-terms exchange*. The *two-terms exchange* replaces up to two terms from the current rule and then adds other possible terms. The procedure then repeatedly performs operations from the given rule until no further improvement can be achieved. Equations (6) and (7) define the acceptance criterion used to accept the better-quality classification rule during the local search stages.

$$\text{AcceptanceCriterion} = \begin{cases} \text{BestRule}^*, \ if \ \text{Quality (BestRule}^*) > \text{Quality (BestRule)} \\ \text{BestRule}, \ Otherwise \end{cases} \quad (6)$$

$$\text{Quality (BestRule}^*) = \frac{TP}{TP+FN+FP} + \frac{TN}{FP+TN} \quad (7)$$

where *TP* is the number of instances covered by the discovered rule and class predicted by the rule. FN is the total number of instances covered by the discovered rule and class different from the class predicted by the rule. *TN* is the total number of instances not covered by the discovered rule and does not have the class predicted by the rule. *FP* is the total number of instances not covered by the discovered rule but has the class predicted by the rule.

The acceptance criteria in ILS will be the rudder to guide the search toward the bottom of the neighbourhood structure shape. Furthermore, it determines the size of movement in the current search region.

The ILS–Ant-Miner pseudocode (see Figure 3) shows the adaptation of the abovementioned components of the ILS-based algorithm (i.e., perturbation, local search, and acceptance criterion) in the AntMiner framework. The combined feature makes the proposed classifier substantially different from the previous Ant-Mining classification algorithm.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ILS-AntMiner

```
Input: arff dataset
Output: classification rule
1  TrainingSet= {all Training instances};
2  InitialConstructionRuleList =[];
3  WHILE(TrainingSet>MaxUncoveredInstances)
4  AntIndex=1; ConvergenceTest=1;
5  InitializePheromone;
6  REPEAT
7  RuleConstructs;
8  RulePrune;
9  UpdatePheromone;
10 IF(CurrentRule=PreviousRule)
11 THEN ConvergenceTest = ConvergenceTest + 1;
12 ELSE ConvergenceTest = 1;
13 END IF AntIndex = AntIndex + 1;
14 UNTIL(AntIndex>=AntNo) OR (ConvergenceTest>=RuleConvergenceNo)
15 SelectBestRule;
16      REPEAT
17        BestRule′ =Perturbation (BestRule);
18        BestRule* =LocalSearch(BestRule');
19        BestRule = AcceptanceCriterion (BestRule, BestRule*);
20      UNTIL termination condition met
21 AddBestRule to DiscoveredRuleList;
22 TrainingSet = TrainingSet-{InstancesSetCoveredByRule};
23 END-WHILE
```

Figure 3: ILS–Ant-Miner Pseudocode

4. EXPERIMENTAL RESULTS AND DISCUSSION

A 10-fold cross-validation procedure was used to evaluate the ant- mining classification algorithms. In this procedure, the dataset is split into 10 groups. Each group is equally sized, where nine groups are used for the training process. The remaining group is used in the testing stage. This process is repeated 10 times with a different group for training and testing to ensure that all groups are used. Subsequently, the performance of all folds is averaged, and the standard deviations are computed. The 10-fold cross-validation method is used in other ant-mining classifier studies [12], [36].

The evaluation is performed on the basis of three criteria. Firstly, the classification accuracy in discovering the rules list is called the accurate classification rate. This criterion is based on the accurately classified instances in the test data. Each time, the training subsets consist of n number of instances, and the classifier constructs the training and test subsets that are used to test the performance. The accurate classification instances determine the performance of the proposed classifier. Secondly, the size in discovering the rules list is measured by the amount of terms per rule. The number of terms (conditions) refers to the number of antecedents carried by each rule. Thirdly, the performance of the algorithms in the classification accuracy and model size is observed. The average classification accuracy rank versus the average model size rank of all classifiers is used in our experiments. A low average rank implies good algorithm performance.

Benchmark datasets are used to compare the proposed algorithms with the commonly related ant-mining classification algorithms in the literature. Benchmark datasets are selected in accordance with the ant-mining literature. This benchmark includes secondary datasets selected from UCI [37].

The parameter values for the ant-miner classifiers are adopted from Raul et al. (2016), López–Ibáñez et al. (2016) to ensure that each classifier works with the same parameter values and provides a fair evaluation of the results [38]–[40].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The parameters used for the Ant-Miner classifiers are listed in Table 2.

Table 2: Experimental Parameters

Parameter	Description	Value
NA	Number of ants in the colony	10
MCR	Minimum number of cases that each rule must cover	5
MUC	Maximum of uncovered cases by the discovered rules	10
RC	Number of rules for convergence	10
NI	Number of iterations	10
β	Beta	1
α	Alpha	1

The implementation of hybridising the Ant-Miner with ILS or ILS-AntMiner, is evaluated with two other hybrid classifiers, namely, ACO/PSO2 and ACO/SA. These classifiers are considered to be the most-related classifiers in the ant-mining literature. Then, the classification performance is analysed in detail for these algorithms by using 16 datasets from UCI under different benchmark scenarios. In the first stage, Tables 3 and 4 show the experimental results of the average classification accuracy, average number of discovered rules, and model size by using the 10-fold cross validation method. In each table the first row presents the average classification accuracy, and the numbers after the symbol "+/–" are standard deviations. For each dataset, the best result is highlighted in bold. The second row displays the performance rank for each dataset. The experimental results in Tables 3 and 4 are used to determine the best classifiers.

As shown in Table 3, the ILS-AntMiner generates the highest classification accuracy compared with the other two classifiers. The ILS-AntMiner achieves the best results in 14 and eight datasets compared with ACO/PSO2 and ACO/SA, respectively. The ILS-AntMiner achieves the first best result in eight datasets compared with the other two classifiers. The ILS-AntMiner obtains the second best result in six datasets. The ACO/SA achieves the overall best classification performance in six datasets. Finally, ACO/PSO2 obtains third place in two datasets.

Table 4 shows the simplicity (the less number of terms per rule) of the constructed rules. The ILS-AntMiner dominates 11 datasets compared with the ACO/PSO2 classifier. In addition, the ILS-AntMiner obtains the best result in 14 datasets compared with ACO/SA. In comparison with the other classifiers, the ILS-AntMiner achieves the best result with 11 datasets and obtains the second best result in four datasets. Meanwhile, the second best performance is achieved by the ACO/PSO2 classifier with five datasets. ACO/SA obtains no highest results compared with the other classifiers.

The ILS-AntMiner benefits from exploration components by employing the perturbation in exploiting multiple neighbourhood structures. Therefore, the ILS-AntMiner dominates the other classifiers in all evaluation criteria.

Table 3: Average Classification Accuracy (Average +/− Standard Deviation, Performance Rank) Obtained Using 10-Fold Cross-Validation Method for All Classifiers And ILS-Antminer

Dataset	ACO/PSO 2	ACO/SA	ILS-AntMiner
Balance Scale	68.66% +/- 4.97	71.04 % +/- 3.91	**71.19% +/- 1.19%**
	3	2	**1**
Breast Cancer (Ljubljana)	70.94% +/-5.37	72.39 % +/- 9.09	**73.57% +/- 2.91%**
	3	2	**1**
Breast Cancer (Wisconsin)	93.86% +/-4.56	**96.14 % +/-2.93**	95.14% +/- 0.53%
	3	**1**	2
Credit-a	84.69% +/-4.39	85.80 % +/-2.58	**86.67% +/- 1.58%**
	3	2	**1**
Credit-g	71.0% +/-4.52	**75.50 % +/-3.29**	72.4% +/- 1.43%
	3	**1**	2
Diabetes	76.31% +/-4.32	76.70 % +/- 4.11	**76.93% +/- 2.18%**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	3	2	**1**
Heart (Cleveland)	78.51% +/-6.16	81.78 % +/-7.29	**82.28% +/- 2.4%**
	3	2	**1**
Heart (Statlog)	78.89% +/-7.78	81.11 % +/-9.14	**82.06% +/- 2.22%**
	3	2	**1**
Hepatitis	76.13% +/-8.34	**83.25 % +/-5.79**	78.17% +/- 2.74%
	3	**1**	2
Ionosphere	65.51% +/-7.46	**90.89 % +/-3.98**	89.86% +/- 2.09%
	3	**1**	2
Iris	94.0% +/-8.14	93.33 % +/-8.43	**96% +/- 1.47%**
	2	3	**1**
Lymphography	77.19% +/-12.59	78.29 % +/-6.88	**80.36% +/- 4.62%**
	3	2	**1**
Mushroom	**100.0% +/-0.0**	99.01 % +/-2.55	98.02% +/- 0.13%
	1	2	3
Segment	82.08% +/-4.64	**92.42 % +/-1.60**	91.56% +/- 0.76%
	3	**1**	2
Sonar	54.86% +/-3.87	**80.36 % +/-7.40**	76.47% +/- 2.37%
	3	**1**	2
Tic-tac-toe	**100.0% +/-0.0**	97.18 % +/-1.88	81.93% +/- 1.52%
	1	2	3

Table 4: Average Model Size (Average +/− Standard Deviation, Performance Rank) Obtained Using 10-Fold Cross-Validation Method for All Classifiers And ILS-Antminer

Dataset	**ACO/PSO 2**	**ACO/SA**	**ILS-AntMiner**
Balance Scale	52 +/- 0	42.90 +/- 5.15	**8.9 +/- 0.38**
	3	2	**1**
Breast Cancer (Ljubljana)	26.8 +/- 6.196	33.20 +/- 3.74	**13.1 +/- 0.81**
	2	3	**1**
Breast Cancer (Wisconsin)	17.1 +/- 2.42	18.90 +/- 2.02	**9.8 +/- 0.49**
	2	3	**1**
Credit-a	70.6 +/- 7.6	53.50 +/- 8.88	**25.2 +/- 1.98**
	3	2	**1**
Credit-g	**30.5+/- 16.33**	127.90 +/- 14.82	45.9 +/- 1.28
	1	3	2
Diabetes	112.5 +/- 9.312	65.70 +/- 3.90	**29.6 +/- 2.09**
	3	2	**1**
Heart (Cleveland)	28.3 +/- 4.347	29.10 +/- 3.53	**20.9 +/- 1.14**
	2	3	**1**
Heart (Statlog)	25.9 +/- 4.30	27.60 +/- 3.98	**17.5 +/- 1.08**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	2	3	**1**
Hepatitis	**11.6 +/- 2.31**	15.70 +/- 3.47	16.2 +/- 1.08
	1	2	3
Ionosphere	**2.2 +/- 0.42**	24.50 +/- 3.75	12.7 +/- 0.87
	1	3	2
Iris	**3.3 +/- 0.94**	4.80 +/- 1.08	4.1 +/- 0.48
	1	3	2
Lymphography	42.8 +/- 6.48	16.50 +/- 2.97	**16.2 +/- 0.63**
	3	2	**1**
Mushroom	33.4 +/- 2.87	37.00 +/- 2.90	**7.7 +/- 0.15**
	2	3	**1**
Segment	59.3 +/- 7.9	121.60 +/- 5.97	**43 +/- 1.26**
	2	3	**1**
Sonar	**0.9 +/- 1.97**	25.70 +/- 3.61	21 +/- 1.37
	1	3	2
Tic-tac-toe	53.6 +/- 7.306	96.50 +/-11.14	**20.6 +/- 2.13**
	2	3	**1**

Table 5 and Figure 4 show the evaluation of the performance based on the nonparametric Friedman test with Holm's post hoc test. This test aims to find the dominant classifier among the 16 datasets. As shown in Table 5, in all cases, the lowest rank indicates a good algorithm performance. Thus, the ILS-AntMiner obtains the best average rank in classification accuracy, number of discovered rules, and model size.

Figure 4 displays the result of the average classification accuracy rank versus the model size rank. The ILS-AntMiner obtains the best classification accuracy and best model size. Under these circumstances, the ILS-AntMiner dominates the hybridisation with Ant-Miner classifier in all evaluation criteria. This result is due to the enhancement process achieved by the ILS algorithm to the rule produced by the Ant-Miner classifier. ILS uses the power of the multiple neighbourhood structures (i.e., perturbation and local search) procedures to escape from the local optima.

Table 5: Results Of ILS-Antminer And Other Classifiers Based on Average Performance Rank on All Datasets

	ACO/PSO 2	ACO/SA	ILS-AntMiner
Accuracy	2.6875	1.6875	**1.625**
Terms	1.9375	2.6875	**1.375**

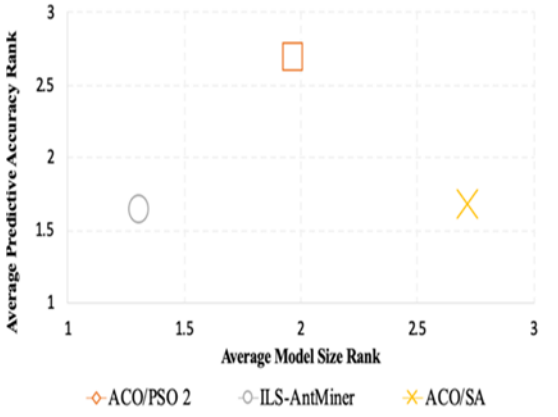

Figure 4: ILS–Ant-Miner Pseudocode

The overall goal of this research is to benefit from characteristics of ILS to form the neighbourhood structures. In ILS-AntMiner, there is a strong inter-correlation between exploration (e.g., perturbation) and exploitation components (e.g., local search). Furthermore, ILS-AntMiner concept has succeeded

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

in employing perturbation in exploiting multiple neighbourhood structures. ILS-AntMiner has utilised perturbation procedure to improve the search in two of its neighbourhood structures in the proposed hybridisation. Thus, ILS-AntMiner significantly different from previous researches in the literature.

5. CONCLUSION AND FUTURE RESEARCH

The Ant-Miner is a prominent ACO-based rules classification framework with high interpretation ability to find the relationships between features and outstanding performance in a simple manner. However, the Ant-Miner classifier also suffers from premature exploitation because it has not been designed to explore the neighbourhoods of the current rule and does not improve it iteratively. This work investigated a hybrid rules-based classifier, ILS-AntMiner, that hybridised the ACO and ILS for a more mature exploitation to improve classification accuracy with small model complexity. Experimental results concerning benchmark dataset outperform the well-known hybrid ACO classifiers namely, ACO/PSO2, and ACO/SA in terms of classification accuracy and model size. For future research directions, other stochastic local search algorithms, such as randomised iterative improvement, iterated greedy and evolutionary algorithms, can be hybridised with Ant-Miner algorithms. Another research direction, the proposed ILS-AntMiner algorithm, can be improved in terms of its capability to cope with continuous attributes. The proposed classifier can only cope with discrete attributes and, thus, needs a pre-processing step to deal with continuous attributes.

ACKNOWLEDGEMENTS

The authors thank the Ministry of Higher Education Malaysia for funding this study under the Transdisciplinary Research Grant Scheme, TRGS/1/2018/UUM/02/3/3 (S/O code 14163).

REFERENCES:

[1] H. N. K. Al-behadili, "Intelligent Hypothermia Care System using Ant Colony Optimization for Rules Prediction," *J. Univ. Babylon*, vol. 26, no. 2, pp. 47–56, 2018.

[2] N. C and S. V, "A Study on Applications of Machine Learning Techniques in Data Mining," *Shodhshauryam, Int. Sci. Ref. Res. J.*, vol. 1, no. 3, pp. 31–34, 2005.

[3] A. M. Jabbar and K. R. Ku-Mahamud, "Ant-based sorting and ACO-based clustering approaches: A review," in *In 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2018, pp. 217–223.

[4] A. M. Jabbar, R. Sagban, and K. R. Ku-Mahamud, "BALANCING EXPLORATION AND EXPLOITATION IN ACS ALGORITHMS FOR DATA CLUSTERING," *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 16, pp. 4320–4333, 2019.

[5] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "Modified ACS Centroid Memory for Data Clustering," *J. Comput. Sci.*, vol. 15, no. 10, pp. 1439–1449, 2019.

[6] H. N. K. Al-behadili, "Classification Algorithms for Determining Handwritten Digit," *Iraqi J. Electr. Electron. Eng.*, vol. 12, no. 1, pp. 96–102, 2016.

[7] H. B. Alwan and K. R. Ku-Mahamud, "Mixed-variable ant colony optimisation algorithm for feature subset selection and tuning support vector machine parameter," *Int. J. Bio-Inspired Comput.*, vol. 9, no. 1, pp. 53–63, 2017.

[8] H. N. K. Al-behadili, K. R. Ku-Mahamud, and R. Sagban, "Annealing strategy for an enhance rule pruning technique in ACO-based rule classification," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 3, pp. 1499–1507, 2019.

[9] H. N. K. Al-behadili, K. R. Ku-Mahamud, and R. Sagban, "Ant colony optimization algorithm for rule-based classification: Issues and potential solutions," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 21, pp. 7139–7150, 2018.

[10] H. N. K. Al-behadili, K. R. Ku-Mahamud, and R. Sagban, "Rule pruning techniques in the ant-miner classification algorithm and its variants: A review," in *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2018, pp. 78–84.

[11] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, "Data mining with an ant colony optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 321–332, 2002.

[12] R. Saian and K. R. Ku-Mahamud, "Ant colony optimization for rule induction with simulated annealing for terms selection," *Proc. - 2012 14th Int. Conf. Model.*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Simulation, UKSim 2012, no. March 2012, pp. 33–38, 2012.

[13] D. Martens, B. Baesens, and T. Fawcett, "Editorial survey: Swarm intelligence for data mining," *Mach. Learn.*, vol. 82, no. 1, pp. 1–42, 2011.

[14] M. Dorigo, M. Birattari, and T. Stützle, "Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique," Bruxelles, Belgium, 2006.

[15] T. Stutzle, "Local Search Algorithms for Combinatorial Problems- Analysis, Improvements, and New Applications," Technische Universiti at Darmstadt, 1998.

[16] M. Dorigo and T. Stutzle, "The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances," *Handb. Metaheuristics SE - Int. Ser. Oper. Res. Manag. Sci.*, vol. 57, pp. 250–285, 2003.

[17] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and Conceptual Comparison," *ACM Comput. Surv.*, vol. 140, no. 1, pp. 189–213, 2001.

[18] J. Levine and F. Ducatelle, "Ant colony optimization and local search for bin packing and cutting stock problems," *J. Oper. Res. Soc.*, vol. 55, no. 7, pp. 705–716, 2004.

[19] A. F. Abuhamdah, "Adaptive elitist-ant system for medical clustering problem," *J. King Saud Univ. - Comput. Inf. Sci.*, pp. 0–8, 2018.

[20] P. Lin, R. Dai, M. A. Contreras, and J. Zhang, "Combining ant colony optimization with 1-opt local search method for solving constrained forest transportation planning problems," *Artif. Intell. Res.*, vol. 6, no. 2, p. 27, 2017.

[21] G. M. Jaradat, "Hybrid elitist-ant system for a symmetric traveling salesman problem: case of Jordan," *Neural Comput. Appl.*, vol. 29, no. 2, pp. 565–578, 2016.

[22] M. D. Toksari, "A hybrid algorithm of Ant Colony Optimization (ACO) and Iterated Local Search (ILS) for estimating electricity domestic consumption: Case of Turkey," *Int. J. Electr. Power Energy Syst.*, vol. 78, pp. 776–782, 2016.

[23] A. Ezzat, A. M. Abdelbar, and D. C. Wunsch, "An extended EigenAnt colony system applied to the sequential ordering problem," *IEEE SSCI 2014 - 2014 IEEE Symp. Ser. Comput. Intell. - SIS 2014 2014 IEEE Symp. Swarm Intell. Proc.*, pp. 276–282, 2015.

[24] T. Liao, M. Montes, D. Aydin, T. Stützle, and M. Dorigo, "An Incremental Ant Colony Algorithm with Local Search for Continuous Optimization," *Gecco-2011 Proc. 13th Annu. Genet. Evol. Comput. Conf.*, no. January, pp. 125–132, 2011.

[25] Y. Drias, S. Kechid, and G. Pasi, "A Novel Framework for Medical Web Information Foraging Using Hybrid ACO and Tabu Search," *J. Med. Syst.*, vol. 40, no. 1, p. 5, 2015.

[26] V. P. Eswaramurthy and A. Tamilarasi, "Hybridizing tabu search with ant colony optimization for solving job shop scheduling problems," *Int. J. Adv. Manuf. Technol.*, vol. 40, no. 9–10, pp. 1004–1015, 2009.

[27] L. M. Gambardella, "Coupling Ant Colony System with Local Search," UNIVERSIT´E LIBRE DE BRUXELLES, 2015.

[28] L. M. Gambardella, R. Montemanni, and D. Weyland, "Coupling ant colony systems with strong local searches," *Eur. J. Oper. Res.*, vol. 220, no. 3, pp. 831–843, 2012.

[29] J. Guan and G. Lin, "Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem," *Eur. J. Oper. Res.*, vol. 248, no. 3, pp. 899–909, 2016.

[30] J. J. De La Cruz, C. D. Paternina-Arboleda, V. Cantillo, and J. R. Montoya-Torres, "A two-pheromone trail ant colony system - Tabu search approach for the heterogeneous vehicle routing problem with time windows and multiple products," *J. Heuristics*, vol. 19, no. 2, pp. 233–252, 2013.

[31] R. Sagban, K. R. Ku-Mahamud, and M. S. Abu Bakar, "Reactive max-min ant system with recursive local search and its application to TSP and QAP," *Intell. Autom. Soft Comput.*, vol. 23, no. 1, pp. 127–134, 2016.

[32] B. Fox, W. Xiang, and H. P. Lee, "Industrial applications of the ant colony optimization algorithm," *Int. J. Adv. Manuf. Technol.*, vol. 31, no. 7–8, pp. 805–814, 2007.

[33] J. Ji, N. Zhang, C. Liu, and N. Zhong, "An Ant Colony Optimization Algorithm for Learning Classification Rules," *WI '06 Proc. 2006 IEEE/WIC/ACM Int. Conf. Web*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Intell., pp. 1034–1037, 2006.

[34] R. Saian and K. R. Ku-Mahamud, "Hybrid Ant Colony Optimization and Simulated Annealing for Rule Induction," *2011 UKSim 5th Eur. Symp. Comput. Model. Simul.*, no. March 2016, pp. 70–75, 2011.

[35] R. Parpinelli, H. Lopes, and A. A. AFreitas, "Data Mining With an Ant Colony Optimization Algorithm," *IEEE Trans. Evol. Comput.*, vol. 47, no. 6 (4), pp. 321–332, 2002.

[36] K. M. Salama and A. M. Abdelbar, "Extensions to the Ant-Miner classification rule discovery algorithm," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6234 LNCS, pp. 167–178, 2010.

[37] D. Dua and T. Karra, "UCI Machine Learning Repository," *Irvine, CA: University of California, School of Information and Computer Science*, 2017. [Online]. Available: http://archive.ics.uci.edu/ml.

[38] R. Robu, C. Vacar, N. Robu, and S. Holban, "A study on Ant Miner parameters," in *6th International Conference on Information, Intelligence, Systems and Applications*, 2016.

[39] M. López-Ibáñez, T. Stützle, and M. Dorigo, "Ant Colony Optimization: A Component-Wise Overview," Bruxelles, Belgium, 2016.

[40] M. L. Raymer, W. Punch, E. Goodman, L. Kuhn, and A. Jain, "Dimensionality reduction using genetic algorithms - Evolutionary Computation," *IEEE Trans. Evol. Comput.*, vol. 4, no. 2, pp. 164–171, 2000.