DOI: 10.6977/IJoSI.202203_7(1).0004
IJoSI S. W. Kareem, A. S. Mohammed, etc. / Int. J. Systematic Innovation, 7(1), 46-65 (2022)

Bayesian Network Structure Discovery Using Antlion Optimization Algorithm.

Hoshang Qasim Awla[1], Shahab Wahhab Kareem[2], Amin Salih Mohammed[3]
[1] Computer science department, Faculty of Science, Soran University, Erbil, Iraq.
[*] Corresponding author E-mail: Shahab.kareem@epu.edu.iq
[2] Dep. of Technical Information Systems Engineering, Erbil Technical Engineering College, Erbil Poly-technic University, Erbil, Iraq.,
[3]Department of Information Technology, College of Engineering and Computer Science, Lebanese French University, Erbil, Kurdistan Region, Iraq
(Received 16 August 2021; final version received 18 February 2022; accepted 19 February 2022)

Abstract

Bayesian networks have recently been used for discovering an optimal learning structure in machine learning. Bayes networks can describe possible dependencies of explanatory variables. As a novel approach to studying the structure of a Bayesian network, the authors present the Antlion Optimization Algorithm (ALO). In the algorithm; deletion, rewind, insertion, and change are utilized to produce ALO to reach the best hull solution. Essentially, the technique used in the ALO algorithm imitates the antlions' behaviors while hunting. The suggested approach is contrasted with simulated Annealing, Simulated Annealing Hybrid Bee, Greedy Search Hybrid Bee, optimization inspired by Pigeon, and greedy search using the BDe Score function. The researchers also studied the representation of the confusion matrix of these techniques using different reference data sets. The findings of the assessments reveal that the proposed algorithm works better than the other algorithms and has better consistency and score values. As shown by the experimental evaluations, the proposed method has a more reliable performance than other algorithms (including the production of excellent scores and accuracy values).

Keywords:Bayesian Network, Metaheuristics, Antlion Optimization, Structure Learning, Hunting Search, Local & Global Search, Search and Score

1. Introduction

One of the simplified theoretical techniques for graphing the probabilistic framework of observed data in machine learning is Bayesian Networks (BN) (Junzhong, Wei, Liu, 2012). They can be defined and executed completely for purposes including; information design, inference, and argumentation (Fortier, Sheppard and Pillai, 2013). The structure of the Bayesian network is represented as a direct acyclic graph (DAG) which is designed based on two complementary parts; the structure and the parameters of the network. The structure represents dependencies between the variables and the parameters represent conditional probabilities. Discovering the learning structure of a Bayesian network is difficult without a proper search approach. Learning the optimal structure of a Bayesian network (BN) using a dataset is NP-hard class (Junyi and Chen, 2014). A direct acyclic graph (DAG) is the configuration of the Bayesian network, which is constructed based on two complementary parts; the network structure and parameters. Dependencies between the variables are expressed by the configuration and the parameters represent conditional probabilities. Without a proper search strategy, solving the learning structure of a Bayesian network is challenging (Margarritis, 2003). To analyze the space of BN structures, the score and scan method is used to continuously approximate all alternative network structures before the valid metric value is obtained.

Score-based processes focus on a network prediction function, possible data, and aim for a framework that optimizes the score, which is the target (Fast, 2010). Two key models are used to apply the score function approach: The Bayesian score and the information-theoretical score. In methods such as; Log-likelihood (LL), Bayesian Information Criterion (BIC), Normalized Minimum Likelihood (NML), Akaike Information Criterion (AIC), Minimum Description Length (MDL), and Mutual Information Tests (MIT), the information-theoretical score used. In several various methods, the Bayesian score was done, such as; BD (Bayesian Dirichlet), BDe (Bayesian Dirichlet ('e' for probability equivalence)), BDeu (Bayesian Dirichlet equivalent uniform ('u' for uniform joint distribution)), and K2 (Cooper and Herskovits,1992).

There are different search method approaches to simplify the issue of structure learning. These include the Ant Colony Algorithm for Optimization (Salama and Freitas, 2012), Particle Swarm Intelligence (Cowie, Oteniya, Cles, 2007), Bee Colony(Li and Chen, 2014), Hybrid Algorithm ((He and Gao, 2018)(Li, and Wang, 2017)(Kareem and Okur, 2018)), Bacterial Foraging Optimization (Yang, Junzhong, Liu, Jinduo and Yin, 2016), Simulated Annealing Algorithm(Hesar, 2013), Breeding Swarm Algorithm(Khanteymoori, Olyaee, Abbaszadeh and Valian, 2018), Genetic Algorithms(Larraiiaga, Poza, 1996), Pigeon Inspired Optimization(Kareem and Okur, 2019), Gene-Pool Optimal Mixing Evolutionary Algorithm (GOMEA) (Orphanou, Thierens, and Bosman, 2018), Elephant Swarm Water Search Algorithm (Kareem and Okur, 2020), Falcon Optimization algorithm (Kareem and Okur, 2021), Binary Encoding Water Cycle(Wang and Liu, 2018), Tightening Bounds(Fan, Malone, 2014), A* Search Algorithms(Malone, Wui, 2011), Scatter Search Documents(Patrick and Sahin, 2004), Quasi-Determinism Screening(Rahier, Marie, Girard, Forbes, 2019), Cuckoo Optimization Algorithm(Askari and Ahsaee, 2018), and Minimum Spanning Tree Algorithm (Sencer, Oztemel, Taksin and Torkul, 2013). Antlion optimization is a different case of metaheuristic methodology that can be applied in Bayesian networks for structure learning. As a novel approach to Bayesian network structure learning, this paper proposes and provides a comparative review of this process. ALO is applied in the different optimization problems like the parallel machine scheduling (Kilic and Yuzgec, 2019), governing loop of thermal generators (Gupta and Saxena, 2016), Optimal Reactive Power Dispatch (ORPD) problem (Mei, Sulaiman, Zuriani, 2015), and optimal route planning of unmanned aerial vehicles (Yao and Wang, 2017).

Because there are hundreds of nodes involved in high-dimensional data sets, which are now booming across a wide range of areas, the accompanying Bayesian network structure is very complex. Discovering the Bayesian network structure based on the data turns into an NP-hard problem. To acquire an ideal structure from complicated and high-dimensional data sets in a fair amount of time, one of the primary problems in Bayesian network research is to develop an effective structure learning approach that is both efficient and effective. For compared to an expert system based on empirical knowledge, the Bayesian network eliminates the uncertainty issue, which is particularly important when handling complicated problems, making it both more effective and intuitive to use.

Given that learning the network from data is generally considered to be an NP-hard optimization problem, it is necessary to find an efficient search algorithm; to this end, the heuristic algorithm, which has high search efficiency and is frequently used to find the best network structure in structure learning, has been developed.

Implementation of the stochastic search algorithm is straightforward, and the method's global search capability is enhanced. The global information may be employed more thoroughly than with other conventional search techniques, and the dependency on optimization function is less when compared to the other approaches. As a result, the network structure learned by the structure learning technique based on the stochastic search algorithm differs less from the actual structure and may converge to a better structure more rapidly, as well as the quality of the convergence itself is improved. The ALO has three major advantages: it can find a near-optimum solution independent of the starting parameter values, it has a quick convergence rate, and it can handle both integer and discrete constraints simultaneously.

The following is the layout of the remainder of this article. In Bayesian Networks, Part 2 discusses the principle of structure learning. A short introduction to the Antlion Optimization Algorithm is included in Part 3. We describe in part 4, the technique in-depth and demonstrate the experimental outcomes. The findings are found in Section 5.

2. Structure Learning of Bayesian Networks

Two components can be used to express the Bayesian Network: (G, P). The first, G (V; E), is the DAG that covers the calculable group of vertices (or nodes), V, interconnected by labeled edges (or connections), E. The second, P = {P (Xi Pa (Xi)))}, is a set of conditional probabilistic (CPD) distributions, entity to all Xi variables (chart vertices). In addition, Pa(Xi) represents the set of the node Xi parents in G (Cowie, Oteniya, Coles, 2007). A simple probabilistic combination for a (G; P) network can be represented based on this model via:

$$P(X_i, \ldots X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i)) \qquad (1)$$

On the other hand, a score function relies on many parameters, including Bayesian methods, data, and entropy, the minimum duration of explanation (Campos, 2006). Bayesian network posterior likelihood, according to Bayesian inference rules, can be expressed as:

$$P(G|D) = P(D|G)P(G) / \sum_{G'} P(D|G') \, P(G') \qquad (2)$$

In (2), P(D) is a conditional probability specified by using P(D) normalizing constant as:

$$P(D|G) = \int P(D|G,\theta) \, P(\theta|G) d\theta \qquad (3)$$

It is presumed that P(D) is independent of the Bayesian network G configuration. P(G) 'is the preceding likelihood, and θ represents the model parameter. As a result, the resulting distribution of the network configuration can be estimated as long as the marginal likelihood of all possible configurations is determined (Zhang, Liu, 2008). Structure learning approaches use score-based strategies by comparing the structure's existing and previous scores. The final expression of the ranking is (Heckerman, Geiger, and Chickering, 1995):

Score (G, D)=∑Score(Xi, Pa(Xi), D(Xi, pa(Xi))) (4)

3. Antlion Optimization Algorithm

Metaheuristics are algorithms motivated by nature to find approximate solutions to certain computationally complicated problems of optimization. In metaheuristics, swarming activities of firefly-BATT (Gadekallu, and Khare, 2016), cuckoo (Gadekallu, and Khare, 2017), ant, pigeon, fish, bee, etc. were used, (Gandomi, Yang, Marand, and Alavi, 2013). Some of the metaheuristic methods' supporting properties include adaptability, homogeneity, illation-free resources, and local optima eschewal ability (Mirjalili, Mirjalili, Lewis, 2014).

One of the newly proposed metaheuristic methods is Antlion Optimization (ALO) (Mirjalili, 2015). This is a life-cycle-based search algorithm inspired by evolution. The ALO algorithm simulates the relation in the trap area between antlions (doodlebugs) and ants. Ants are expected to walk blindly around the search area and antlions are often ready to hunt them and in time they become competent with traps. The ALO algorithm aims to model, the combat techniques used by antlions. The antlion's life cycle consists of two main levels: larvae and adults. Normally, antlion life can be rated for up to 3-4 years, which is effectively spent at the larval stage. In the larval and developmental stages, they search for prey daily. Antlion primarily uses five steps to hunt prey, including; arbitrary movement of ants, creation of traps, trapping of ants, hunting of prey, and traps reconstruction (Nischal, Mehta, 2015). In the ALO algorithm ants represent possible arbitrary solutions to a particular problem within the search field and antlion holes to pick up and eat ants in the ground. The ability to chase ants is coded and programmed according to the relationship between the ants and the antlion in the objective role. When describing optimization using the essence of the hunting abilities of antlions, there are impressive actions to consider. Inside the search field, the random motion produced by ants in ALO concerns the locations of ants and antlion-generated holes in the dimensions. The measurements of the cone-shaped hole are proportional to every antlion's health, i.e. the fitter antlions can create stronger holes and thus have a greater chance of capturing prey. The ants will pass into any antlion trap and adaptively decrease the size of their random motion as the antlion slides the ants towards the bottom of the pit. As a result, the eating antlions grow fitter than preys, use their position and restore the hole to maximize their chance of capturing other ants (Zawbaa, Emary, Parv, 2015) (Yogarajan, Revathi, 2016). This approach, which is based on the deep neural network (DNN) model, is used to pick ideal hyperparameters while using the least amount of time. An additional advantage of the suggested model was that it required just 38.13 percent of the total training time. It was shown via testing that the suggested paradigm was more effective (Gadekallu, Bhattacharya, Praveen, Zada, et. al.2020). ALO-SVR, a technique based on ant lion optimization algorithm and support vector regression, was developed to increase lithium battery SOH estimate accuracy. The Pearson correlation coefficient is used to examine the association between features and SOH in this technique, which picks characteristics that are strongly connected with current, voltage, and temperature. The Ant Lion Optimization Algorithm is used to refine the SVR model's main parameters before a final estimate model is developed. The findings reveal that the ALO-SVR approach has greater estimate accuracy and stability than the current GA-SVR and GS-SVR on the NASA public data set, proving the practicality of the estimation method. This research proposes the implementation of Multi-Objective Antlion Optimization (MALO) on solving Flight Scheduling and Aircraft Routing in the current pandemic conditions. The result showed an improvement in the estimated number of passengers and a decrease in the total cost. The result also revealed that MALO is capable of outperforming other well-known optimization algorithms and converged faster in the large data group while able to work faster than Genetic Algorithm (GA) across all experiments, proving MALO to be a more suitable method when dealing with large scheduling task (Awalivian, Raihan, Suyanto and Siti, 2021). Problems that the Ant Lion optimization method easily falls into the local optimum are the focus of this paper's discussion of dynamic random hill-climbing. By using hill-climbing mechanisms, an ant lion's jumping capacity is enhanced. By balancing exploration and development, a dynamic hill-climbing mechanism improves the algorithm's global search capabilities (GU, 2020).

The following conditions are assumed for the optimization process:

- Ants walk in the search area utilizing several arbitrary routes.
- Arbitrary walking is applied to every dimension of ants.
- Walking at random is affected by antlion traps.
- Antlions can create holes that fit their fitness (more powerful fitness, bigger hole).
- The possibility of grabbing more ants in large holes is higher.
- The fittest antlion will trap each ant in repetition.
- To mimic sliding ants in the direction of the antlion, the subjective movement scale has been adaptively decreased.
- When an ant grows fitter than antlion, it suggests that under the sand it is trapped and pulled by antlion.
- The antlion remains near the last captured prey and creates a pit to maximize its chance of capturing another prey.
- The mathematical model of ant is explained as follows, while exploring the region for food, ants walk at random (Kilic, Yuzgec, Karakuzu, 2019):

$$X(i) = [0, cusum(2r(i1-1)), cusum(2r(i2-1)), \ldots, cusum(2r(iT-1))] \quad (5)$$

cusum denotes cumulative sum, T is the largest number of iterations, where X(i) is the random movement of ants at iteration I and r(j) is an arbitrary function represented as:

$$r = \begin{cases} 1 & if\ rand > 0.5 \\ 0 & if\ rand \leq 0.5 \end{cases} \quad (6)$$

in the interval [0, 1], where rand is a random integer. The locations of ants are modified at each iteration, depending on the mechanism of random motion.

According to the upper and lower values, the spontaneous movement of ants should be normalized into the location in the real quest field. It is possible to determine the position of ants by using the following formula for each iteration.

$$X_i^t = \frac{(X_i^t - a_i)(d_i^t - c_i^t)}{b_i - a_i} + c_i^t \quad (7)$$

Where ai and bi are the minimum and maximum random motion values of ant and d_i^t, c_i^t denote the minimum and maximum location values of the antlion at the iteration of ith. X_i^t is standardized by the use of [0,1] $\frac{(X_i^t - a_i)(d_i^t - c_i^t)}{b_i - a_i}$. This shows the location of the selected antlion nearby. The ant walk is influenced by the antlion; the antlion hunts and drags it down to the pit as soon as the ant reaches the trap.

This procedure's mathematical model can be interpreted as follows (Kilic, Yuzgec, Karakuzu, 2019):

$$c_i^t = C^t + Antlion^t, \qquad d_i^t = d^t + Antlion^t \quad (8)$$

$$c^t = \frac{c^t}{I}, \qquad d^t = \frac{d^t}{I} \quad (9)$$

The antlion is the antlion location for each chosen antlion at the t^{th} iteration while d^t and c^t is the maximum and minimum for all variables relating to the i^{th} ant, and I is the unique constant ratio based on the iteration as seen in various scenarios:

$$I = \begin{cases} 1+10^2\frac{t}{T} & \text{if } 0.1T < t < 0.5T \\ 1+10^3\frac{t}{T} & \text{if } 0.5T < t < 0.75T \\ 1+10^4\frac{t}{T} & \text{if } 0.75T < t < 0.9T \\ 1+10^5\frac{t}{T} & \text{if } 0.9T < t < 0.95T \\ 1+10^6\frac{t}{T} & \text{if } 0.95T < t < T \\ 1 & \text{otherwise} \end{cases} \quad (10)$$

where T is denoted as the maximum iteration, and the current iteration is denoted as t. While calculating the X_i^t from equation 7, ants walk nearby to antlion and elite antlion, picked by the roulette wheel principle, in the current population. Ants' new locations are measured by the following equation as follow:

$$(11) \quad Ant_i^t = \frac{R_A^t + R_E^t}{2}$$

where R_A^t is arbitrary movement nearby antlion picked by the roulette wheel in the t-iteration, and R_E^t is the arbitrary movement nearby the elite antlion in the t-iteration. The antlions are needed to update their position to the last ant site to be caught to enhance the possibilities of catching new prey. It is described by the following equation:

$$Antlion_j^t = Ant_i^t \quad \text{if} \quad f(Antlion_j^t) > f(Ant_i^t) \quad (12)$$

where f($Antlion_j^t$) and f(Ant_i^t) denotes the fitness function of the current iteration of Antlion and Ant, $Antlion_j^t$ i indicates the j^{th} ant position specified at the t^{th} iteration and Ant_i^t indicates the position of i^{th} ant at t^{th} iteration.

The locations of ants are collected and stored in the matrix M_{ant} which is utilized for the optimization problem. Similarly, the positions of antlions are stored in the matrix $M_{antlion}$, which is shown as follow:

$$M_{ant} = \begin{bmatrix} Ant_{1,1} & \cdots & Ant_{1,d} \\ \vdots & \ddots & \vdots \\ Ant_{1,n} & \cdots & Ant_{n,d} \end{bmatrix},$$

$$M_{antlion} = \begin{bmatrix} Antlion_{1,1} & \cdots & Antlion_{1,d} \\ \vdots & \ddots & \vdots \\ Antlion_{1,n} & \cdots & Antlion_{n,d} \end{bmatrix} \quad (13)$$

where the $Ant_{i,j}$ denotes to the value of ith ant at jth iteration, and $Antlion_{i,j}$ denote to the value of i^{th} antlion at j^{th} iteration. n denotes the number of ants also antlions, d represents the number of variables (Kilic, Yuzgec, Karakuzu, 2019). The (F) is fitness function of the ants and antlions are determined and saved in the F_{ant} and $F_{antlion}$ matrix as follow:

$$F_{ant} = \begin{pmatrix} f(ant_{1,1}, ant_{1,2}, \ldots ant_{1,d} \\ f(ant_{12,1}, ant_{2,2}, \ldots ant_{2,d} \\ \vdots \\ f(ant_{n,1}, ant_{n,2}, \ldots ant_{n,d} \end{pmatrix},$$

$$F_{antlion} = \begin{pmatrix} f(antlion_{1,1}, antlion_{1,2}, \ldots antlion_{1,d} \\ f(antlion_{12,1}, antlion_{2,2}, \ldots antlion_{2,d} \\ \vdots \\ f(antlion_{n,1}, antlion_{n,2}, \ldots antlion_{n,d} \end{pmatrix} \quad (14)$$

4. Learning Structure of Bayesian Network Based on ALO

As a quest tool for the structural learning of Bayesian networks, the proposed method incorporates the Antlion Optimization (ALO) technique. For the measurement of the Bayesian network structure, the BDe metric is used as a score function. The ALO algorithm is essentially an iterated process consisting of a population of entities where a possible location in a given space is encoded by any antlion. The search area is known to be the space. In nature, the ALO algorithm simulates the chase process of antlions. By walking around a roundabout path and driving the sand with a wide jaw, antlion larvae generate a conical hole in the ground. The larvae shelter below the bottom of the cone after drilling the trap and readily anticipate their insect to pass through the hole as seen in Figure 1. It is removed and degraded as the prey is captured. When an ant approaches the cone, the antlion throws sand over the ant to slide it to the bottom of the hole. The antlion then raises the distance for the next catch. The ALO algorithm is defined as a function of three rows that converge as follows with the global optimum of optimization issues: ALO (A; B; C), where A is a function that produces arbitrary initial solutions, B, when reaching the ultimate norm, treats the first set given by function A and C. The antlion and ant are randomly generated in the ALO algorithm. The location of each ant relative to the antlion chosen by the roulette wheel operator and elite is changed at each iteration. The threshold for site changes is specified first about the current number of iterations. Then, by two random rounds of chosen ants and elites, the revised site is implemented. If all the ants walk randomly, a fitness function is used to estimate them. If any of the ants are more desirable than any other ant, their locations in the next iteration are intended to be new sites for the ants.

The best antlion is connected to the best antlion generated during optimization (elite) and is, if necessary, substituted. Such steps are iterative before false returns.

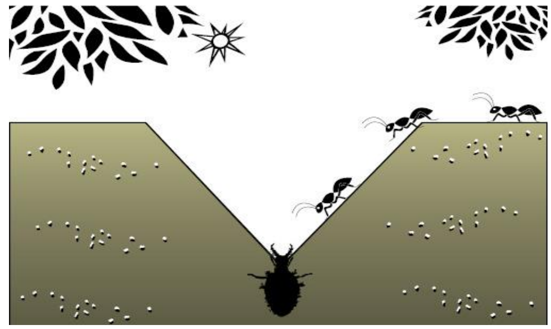

Fig. 1. the hunting process and behavior of antlion. (Anas Atef Amaireh, Asem Alzoubi, Nihad I Dib, 2017).

For each possible DAG, the Bayesian network solution region is generated for structure learning. For a given problem in the search space, the ants in the ALO algorithm represent the potential random solutions and the Antlions drill holes in the ground to trap and eat ants. An antlion’s hunting capacity is encoded in the objective feature and is optimized based on the relationship between ants and antlions. When modeling the optimization problem using the essence of the hunting activity of antlions, there are several laws to consider. A potential solution that represents a DAG having empty arcs is initiated by any antlion within the swarm. The exploration area for the roughly near-optimal or optimal solution, known as the BDe score, is later analyzed by an antlion. Equation (4) is used as the target function of the optimization to determine the BDe score. The goal of the exploration is to obtain a greater BDe score for the structure of the network. The initial solutions are created by iterative operations and the search space boundary is chosen. The arcs are appended one after another, beginning with a null graph (G0), given they are not included in the current graph solution. The append operation is done only if the new solution's score function is greater than the current score and the new solution also meets the DAG constraint. This approach proceeds until the sum of the arcs equals the amount specified in advance. The solution begins to allocate a population for each operator in the model and chooses the solution with a higher score function. According to the chosen operator, Antlion continues until the method has completed a sufficient number of iterations or no longer raises the BDe score. In local optimization, the processes usually involve four different operations: Elimination, Extension, Reversion, and Movement, as seen in Figure 2. Within this domain, the first three are basic operations, requiring only replacing an individual edge from a rival solution every time. This causes a relatively small region near the solution to be used. On the other hand, the existing edges modify the collection of parents for any movement process, which will allow a moderately significant change for the current solution. Therefore, if, after applying basic operators, the solution is not modified, the moving operator will boost it. Walking is the key force in local optimization using the preferred operation, which expands more as an antlion reaches the desired solution.

Walking directions, the turn of different local optimization operators, is becoming more widespread as an antlion constantly travels from a solution to a stronger one by experimentation. An antlion G0, which represents a DAG with arcs, attempts reversion, movement, extension, and deletion as seen in Figure 2, and reaches new G1, G2, G3, and G4 solutions, respectively. It will pick, thinking the best score is in G3, and it will begin to explore a similar method to get G+3 as the new solution. If G+3's BDe score is greater than that of G+1, the subsequent operator will continue to perform. Until the BDe score stabilizes, the operations can repeat, or the iteration loop completes the limit. The antlion chooses Elimination, Extension, Movement and Reversion among the directions in the entire process. The ALO algorithm pseudocode is seen in figure 3.

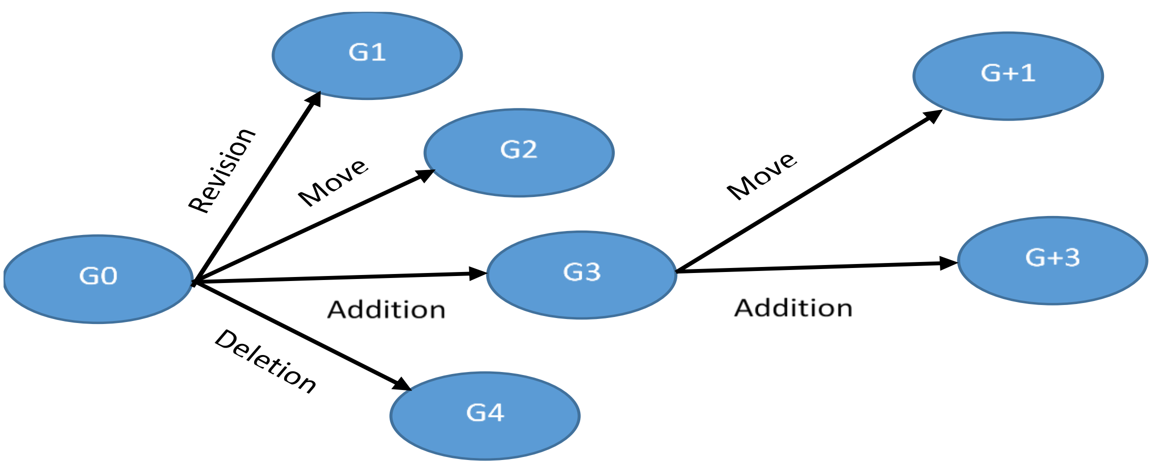

Fig. 2. Hunting searching steps for ALO

Algorithm: Antlion Algorithm for Bayesian Network Structure Learning.
INPUT*: - datasets*
NE: The size of the population
D: search space dimension
Search range: the search space border
tmax: maximum number of iterations; X_{max}*: upper boundary, and* X_{min}*: - lower boundary*
OUTPUT*: - Structure learning Bayesian Network with BDe Score of Best Structure.*

(1) The initialized empty structure and initialize parameters of ALO algorithm (dimension space D, size of population NE, the number of iteration, upper boundary (X_{max})and lower boundary (X_{min}), and $X_{max} > X_{min}$.

(2) Comparing each antlion by BDe score function, and find the elite antlion in the current position ($P^{t}_{best,i,d}$).

(3) While the end criterion is not satisfied

I. Find a new best position by comparing the BDe score function of each antlion, through updating the c_i^t, d_i^t in using equation (8)

II. Create a random walk and normalization using equation (5) and (7)

III. Update the antlion position using equation (11).

(4) Calculate the BDe score function.

(5) Evaluate BDeu score function of the new position using equation (12*).*

a. If the new position is better than the best, so the best position is updated.

b. If the (best global position) < current position, then update the best solution for global

c. The best score value and solution are saved.

d. If Xmin $\geq$ Xmax, stop the iteration process, and the results are present. If not, move into Step 5.

(6) Return the maximum BDe score.

Fig. 3. ALO Algorithm for Bayesian network structure learning

5. Experimental Evaluation

A common validation methodology is used to test the algorithm efficiency of ALO by using probabilistic datasets derived from famous Bayesian network benchmarks. A PC with the following properties is used in the experiment platform: Core i3, 2.1GHz Clock, 4 GB RAM, Ubuntu 14.04 operating system and uses Java to execute the algorithm. In multiple static datasets, the authors investigated the properties of the proposed algorithm, including; Andes (500 instance, 338 arcs, and 223 variables), Lucap02 (10000 instance, and 143 variables), win95pts (574 instance, 112 arcs, and 76 variables), Hepar (350 instance, 123 arcs, and 70 variables), Hailfinder (2656 instance , 66 arcs, and 56 variable), Alarm (10000 instance, 46 arcs, and 37 variables), Soybean (307 instance and 35 variables), Hepatitis(137 instance and 35 variables), Static Banjo (320 instance and 33 variables), Water (10083 instance, 66 arcs, and 32 variables), Epigenetics (72228 instance and 30 variable), Insurance (3000 instance, 52 arcs, and 27 variables), Sensors(5456 instance and 25 variables), Mushroom (1000 instance and 23 variables), Parkinsons (195 instance and 23 variables), Heart(267 instance and 22 variables), Imports(205 instance and 22 variables), Child (230 instance, 25 arcs, and 20 variables), Letter (20000 instance and 17 variables), Adult (30162 instance and 16 variables), Lucas01(10000 instance and 10 variables), WDBC (1000 instance and 9 variables), and Asia (3000 instance, 8 arcs, and 8 variables) (Kareem and Okur, 2019). This study is based on the stationary data presumption in the present form, and stationery sets are the learning datasets that we considered. It is a difficult job to apply the ALO algorithm to sensor data sets or other kinds of online stream data sets and should be tried after testing its output over stationary data sets. It is a difficult job to apply the ALO algorithm to sensor data sets or other kinds of online stream data sets and should be tried after testing its output over stationary data sets. Pigeon Inspired Optimization (PIO) (Kareem and Okur, 2019). Hybrid Simulated Annealing with Bee (BSA) (Kareem and Okur, 2017), Simulated Annealing, Hybrid Greedy with Bee (BGS) (Kareem and Okur, 2018), and Greedy Search methods are correlated with the performance of ALO by using the respective data set metrics. Both algorithms under the same conditions were tested by the authors. Global and Local searches are added to the datasets after specifying the parameters of the ALO algorithm. Population size N=50 and tmax = 10000 are both fixed parameters of ALO optimization for each event. The simulated annealing algorithm parameters are as follows: re-annealing temperature = 500, cooling factor = 0.8, original temperature = 1000. Greedy quest parameters are as follows: recommended maximum networks before reboot = 5000, suggested minimum networks before reboot = 3000, reboot by random network = yes, recommended minimum networks after maximum score = 1000 maximum parent count for operations Reboot = 5. Pigeon parameters are the search space dimension (D=20), the number of pigeons (NP=300), the maximum number of iterations for the map and compass operation (Nc1max=5000), the map and compass factor (P=0.3), and the maximum number of iterations for the landmark operation (Nc2max=10000). The Bee algorithm parameters are: Number of Scout Bees n= 200, Number of repetitions of algorithm steps imax= 10000, Number of best site e out of m chosen site= 7, Number of Sites m out of n visited sites= 30, Number of Bees needed for best e site n2= 90, Initial size of patches ngh including randomly chosen site= 200, Number of Bees needed for the other site (m-e) (n1) = 30. Three separate execution times have been applied by the algorithms: 60, 5, and 2 minutes.

In the data sets and time values listed, the results in Table 1,2,3 indicate the score for each algorithm. It can be observed from these tables that the approach suggested yields better score values for all conditions than the default Greedy Search, and Simulated Annealing Algorithms.

Table 1. The ALO, Simulated Annealing, PIO, Hybrid Bee with Simulated Annealing, Hybrid Bee with Greedy, and Greedy Score function in 2 minutes Execution time.

Dataset	2 Minutes					
	ALO	**PIO**	**Simulated Annealing**	**Hybrid Bee with Simulated Annealing**	**Greedy**	**Hybrid Bee with Greedy**
Asia	**-55049.9**	-55269.5	-56340.3	-56158.6	-56340.3	-56258.7
WDBC	**-6658.43**	-6666.04	-6682.72	-6675.42	-8089.41	-8080.83
lucas01	-11863.1	**-11860**	-12243.2	-12235.3	-13890.9	-13795.3
Adult	**-207805**	-207809	-211678	-211670	-211844	-211850
Letter	**-175185**	-175200	-178562	-178550	-184307	-184205
Child	-62364	**-62362**	-62343.7	-62341.8	-63336.6	-63325.2
Heart	-2424.49	**-2423.8**	-2432.19	-2423.8	-2576.93	-2570.56
Imports	**-1811.99**	**-1811.99**	-1828.91	-1820.26	-1994.15	-1982.59
spect.heart	**-2141.05**	-2142.5	-2141.47	-2141.23	-2144.65	-2144.2
Parkinson’s	**-1488.52**	-1598.91	-1601.3	-1600.92	-1732.76	-1715.57
Mushroom	**-3162.28**	-3372.51	-3375.31	-3374.18	-3745.46	-3745.46
Sensors	**-60341.9**	-60343.3	-60710.5	-60508.7	-69200.3	-68962.5
insurance	**-13896.4**	-138997	-13872.3	-13870.6	-13904.6	-13904
Epigenetics	**-176641**	-176657	-179910	-179906	-225346	-225340
Water	**-11563.4**	-13269.5	-13290.8	-13262.6	-14619.1	-13262.8
static. Data	-8427.12	**-8425.72**	-8451.5	-8449.49	-8585.21	-8570.26
Hepatitis	**-1326.58**	-1327.73	-1330.47	-1329.97	-1350.16	-1346.5
soybean	-2870.3	**-2870.2**	-2870.85	-2859.13	-3021.41	-3025.82
Alarm	-105155	**-105150**	-104927	-104927	-105972	-105552
Hail finder	**-75592.4**	-89521.6	-148193	-148180	-153602	-152038
Hepar	**-160095**	**-160095**	-161086	-161050	-169497	-161051
win95pts	**-46772.8**	-46779.5	-47085.1	-47032.4	-83749.3	-83650.8
Lucap2	**-112982**	-186368	-112261	-111413	-151215	-151243
Andes	**-498180**	-613197	-497353	-477461	-591871	-589927

Table 2. The ALO, Simulated Annealing, PIO, Hybrid Bee with Simulated Annealing, Hybrid Bee with Greedy, and Greedy Score function in 5 minutes Execution time.

ataset	5 Minutes					
	ALO	**PIO**	**Simulated Annealing**	**Hybrid Bee with Simulated Annealing**	**Greedy**	**Hybrid Bee with Greedy**
Asia	**-55157.2**	-55852.6	-56340.3	-56218.5	-56340.3	-56320.9
WDBC	-6662.24	**-6666.04**	-6682.72	-6675.52	-7954.65	-75236.7
lucas01	**-11512.5**	-11892.5	-12243.2	-12229.7	-12243.2	-12230.4
Adult	**-207328**	-207809	-211678	-211664	-211781	-211756
Letter	**-175200**	**-175200**	-178562	-178523	-184916	-182584
Child	**-62363.8**	-62369.2	-62343.7	-62140.7	-63799.4	-63235
Heart	-2424.81	**-2423.8**	-2423.8	-2423.8	-2560.43	-2545.2
Imports	**-1811.99**	**-1811.99**	-1828.91	-1824.3	-2012.21	-1950.3
spect.heart	**-2129.27**	-2132.82	-2143.73	-2140.85	-2142.89	-2141.25
Parkinson’s	**-1441.27**	-1598.91	-1601.3	-1600.58	-1721.16	-1701
Mushroom	**-3162.45**	-3372.51	-3375.31	-3375.51	-3709.7	-3625.4
Sensors	**-60343.3**	**-60343.3**	-60710.5	-60642.2	-69150	-66250
insurance	**13895.11**	**-13895.1**	-13872.3	-13842.7	-13904.6	-13892.3
Epigenetics	**-176637**	-176657	-179300	-179296	-224172	-224162
Water	**-11564.4**	-13269.5	-13290.8	-13262.6	-14644.7	-13264.5
static. Data	**-8414.4**	-8425.2	-8449.77	-8445.41	-8561.93	-8448.24
Hepatitis	**-1327.73**	**-1327.73**	-1330.47	-1328.62	-1350.16	-1340.3
soybean	**-2973.3**	**-2973.3**	-2857.82	-2863.82	-3011.38	-2991.81
Alarm	**-105167**	-105182	-104927	-104927	-106114	-106171
Hail finder	**-75583.9**	-75698	-148188	-148179	-153075	-151863
Hepar	**-160095**	**-160095**	-161086	-161049	-169881	-163375
win95pts	**-46779.5**	**-46779.5**	-47085.1	-47023.7	-83150.7	-75201.5
Lucap2	**-110425**	-175635	-112217	-110834	-152092	-151913
Andes	**-48572**	-613180	-489796	-480065	-588503	-584605

Table 3. The ALO, Simulated Annealing, PIO, Hybrid Bee with Simulated Annealing, Hybrid Bee with Greedy, and Greedy Score function in 60 minutes Execution time.

Dataset	60 Minutes					
	ALO	**PIO**	**Simulated Annealing**	**Hybrid Bee with Simulated Annealing**	**Greedy**	**Hybrid Bee with Greedy**
Asia	**-30584**	-30850	-56340.3	-56340	-56340.3	-56340
WDBC	-6662.25	-6666	-6682.72	-6679.63	-7841.35	-7752.35
lucas01	-11213.8	**12115.38**	-12243.2	-12212.9	-12243.2	-12236.4
Adult	**-207457**	-207809	-211678	-211664	-211762	-211739
Letter	**-175200**	**-175200**	-178562	-178510	-184118	-182269
Child	**-62245.7**	-62275.2	-62343.7	-62312.4	-63799.4	-63756.9
Heart	**-2422.57**	-2423.8	-2432.19	-2423.8	-2527.44	-2522395
Imports	**-1811.25**	-1812	-1828.91	-1824.3	-1995.76	-1950.2
spect.heart	**-2130.87**	-2135.4	-2144.13	-2144.1	-2142.24	-2142.24
Parkinson's	**-1442.87**	-1598.9	-1601.3	-1695.25	-1700.36	-1693.58
Mushroom	**-3019.91**	-3372.5	-3375.31	-3374.57	-3588.69	-3524.83
Sensors	-60343.3	**-60343**	-60710.5	-60612.5	-68364	-67825
insurance	**-13912.7**	-13950	-13872.3	-13850.6	-13904.6	-1385.62
Epigenetics	**-176642**	-176657	-179300	-179296	-217246	-217212
Water	**-11812.7**	-13270	-13290.8	-13262.2	-14272	-13262
static. Data	**-8325.27**	-8368.4	-8445.36	-8552.37	-8556.7	-8552.4
Hepatitis	**-1327.7**	**-1327.7**	-1330.47	-1328.62	-1350.16	-1346.52
soybean	**-2973.3**	**-2973.3**	-2973.83	-2992.99	-3012.72	-2993
Alarm	**-104884**	-104915	-104927	-105271	-105377	-105271
Hail finder	**-75852.4**	-78293	-148183	-151773	-152299	-151773
Hepar	**-160095**	**-160095**	-161086	-163231	-168871	-163231
win95pts	**-46780**	**-46780**	-47085.1	-470016	-83150.7	-80253.4
Lucap2	**-105289**	-105621	-111275	-151160	-150938	-151160
Andes	**-469254**	-469342	-480491	-480253	-586760	-587098

This means that with the minimal time needed, the ALO finds the best score. Another achievement of ALO optimization is observed for various values of population and the highest repetition quantity of the algorithm. The value of population and highest iteration number chosen of these sets {50, 75, 100, 1000, 2000, 3000, 4000, 5000}, respectively. It is observed that the score function is completely satisfying for all datasets. Furthermore, for some datasets, increment in the highest repetition number has less effect on score fitness. Nevertheless, if increase the population number, the score function weakens considerably. But, the increase in population and the repetition amount will reach to larger computational time. The confusion matrix is often used to measure the success of structure discovery. Using existing network architectures, each algorithm and data set may be assigned a confusion matrix value. It is the goal of this study to make a direct comparison between an existing network's structure and the one being constructed. To generate the confusion matrix, we first require a collection of predicted networks that can be compared with the real network. There are rows for the actual classes and columns for the expected classes in a confusion matrix. We need to construct the confusion matrix for each data set and its known network structure to verify the effectiveness of structure discovery.

The confusion matrix was being measured with each data set and its identified network structure to determine the performance of structure discovery. For each network, per algorithm, the metrics TN, TP, FP, and FN, were determined as well as the criteria; AHD, Accuracy (ACC), F1 Score, and Sensitivity (SE), defined as:

$$\text{AHD}= \frac{FN+FP}{TP+TN+FP+FN} \tag{15}$$

$$\text{F1Score} = \frac{2*TP}{2TP+FP+FN} \tag{16}$$

$$\text{Accuracy}= \frac{TP+TN}{TP+FP+TN+FN} \tag{17}$$

$$\text{Sensitivity} = \frac{TP}{TP+FN} \tag{18}$$

The definitions of certain criteria would be as follows: The FN is the arc in the routine, but not in the learning network. FP is the arc that is not in the normal network within the learning network. In neither the learning network nor the normal network, TN is the arc inside A TP is an arc (vertex or edge) within the learning network in the correct position.

In Figure 4, the sensitivity outcomes for ALO, PIO, Simulated Annealing and Greedy are shown. In the multiple datasets, the suggested strategy yields better values than PIO, Virtual Annealing and Greedy. Similarly, as seen in Figure 5, the suggested approach has high precision values in most datasets relative to the Simulated Annealing and Greedy algorithms. In finding the required structure, the proposed ALO Learning Algorithm performs well. As a consequence, the Iterative ALO algorithm is the best in most datasets from the point of estimation accuracy relative to other algorithms, and the ALO is even greater than the other algorithms from the point of construction times. For success indicators, we used F1 as a measure of the model's precision, in addition to the best score in Bayesian results. To measure the efficiency of the proposed algorithm, the F1- score, Accuracy, and Recall are used. In these cases, accuracy is the number of correctly identified guided edges divided by the number of all the edges in the predicted BN. The Recall is a partition of the number of directed edges identified in the real BN by the number of edges. It is known that the harmonic average of precision and recall is F1. The comparison of ALO is presented in figure 6, Greedy search, and Simulated Annealing. Perfection in these cases may be measured by finding all directed edges in a given BN and dividing it by the number of edges in the predicted BN. The number of directed edges detected divided by the total number of edges in the BN is what is represented by the Recall. Precision and recall, which constantly range between zero and one, make up the F1-score.

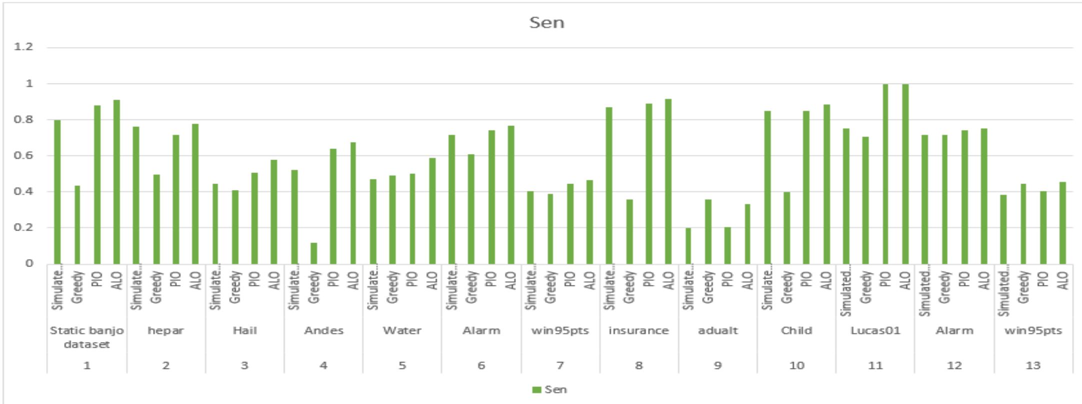

Fig. 4. Sensitivity for ALO, PIO, Simulated Annealing, and Greedy.

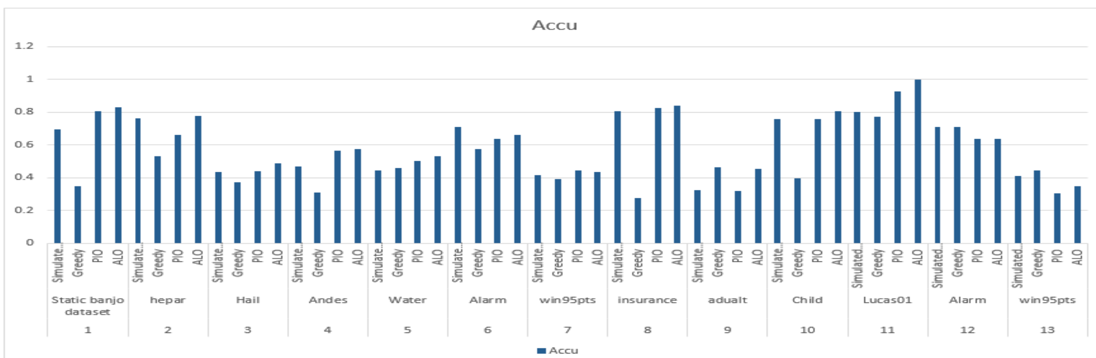

Fig. 5. Accuracy for ALO, PIO, Simulated Annealing Greedy

At 1 an F1 score is at its highest value, while at 0 it is at its lowest. The proposed methods are successful, as seen in Figure 6, than the Greedy search and Simulated Annealing Methods.

In addition, the model's ultimate aim is to provide a convenient representation of the real world, so consistency is a valuable model performance measurement metric.

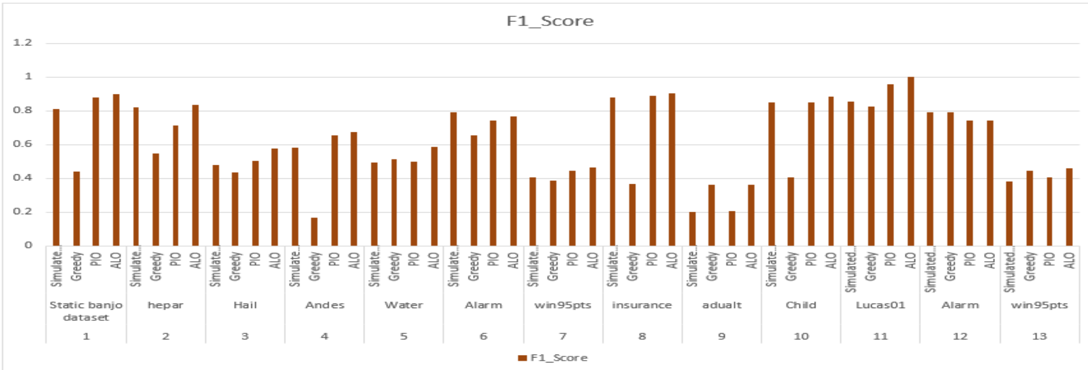

Fig. 6. F1_ Score for ALO, PIO, Simulated Annealing Greedy

The Hamming distances obtained by utilizing the DAG space are always much smaller than those obtained by using the suggested approach. As one of the most often used assessment measures for BN-structured learning, hamming distances fit the structure of students and the real networks, and they are focused exclusively on exploration rather than inference. The findings show that the suggested strategy outperforms the other methods that we've studied in terms of performance. Error correction often makes use of the Hamming distance.

From the Hamming distances, which are often considerably lower than those obtained by using the DAG space, the suggested algorithm is also preferable. Hamming distances are one of the most often used assessment criteria for BN structure learning, and often explicitly fits the learners' configuration and local networks are entirely geared towards discovery rather than inference. For the listed algorithms, Figure 7 shows the Average Hamming Distances. The findings suggest that the approach proposed provides higher output values than the other approaches we have considered

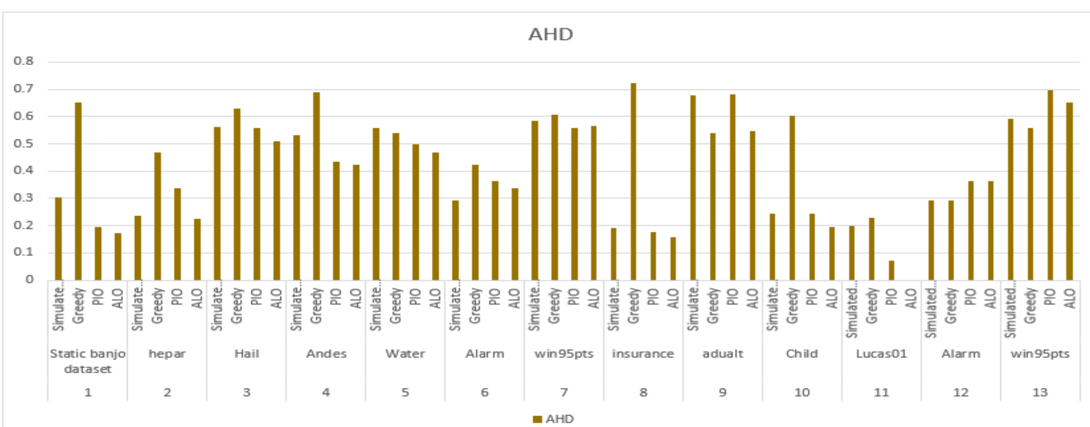

Fig. 7. AHD for ALO, PIO, Sim-

6. Conclusion

The authors concentrated on the learning issue of the Bayesian network structure and introduced the Antlion Motivated Optimization method for Bayesian network structure learning. The score and check strategy is used, using the ALO technique as a search and BDeu as a function of the score. ALO can be defined as a stochastic search method based on antlions' navigational behaviors. ALO is a general method of looking for a separate solution space; it can therefore be modified to accommodate any implementation field. Concentration management in ALO provides the global extremum with improved concentration by allowing the antlion to travel to the shortest available solution space. The suggested approach has a higher search capability, which means that better structure solution can be detected, higher score feature values measured, an excellent approximation to the structure of the network, and the results are correct. Algorithms accelerate global quest and easily contribute to global convergence. We expect to analyze other important ALO characteristics such as: run time analysis, use of energy, overall performance using additional data sets and experimental setups.

References

Khanteymoori, A., Olyaee, M.H, Abbaszadeh, O, & Valian, M. (2018). A Novel Method for Bayesian Networks Structure Learning Based on Breeding Swarm Algorithm. Soft Computing (9).

Gandomi, A.H, Yang, X., Marand, S., T., & Alavi, A. H., (2013). Metaheuristic Applications in Structures and Infrastructures. USA: Elsevier.

Amaireh, A., Alzoubi, A., & Dib, N., I., (2017). Design of Linear Antenna Arrays Using Antlion and Grasshopper Optimization Algorithms. Jordan Conference on Applied Electrical Engineering and Computing Technologies. Jordan: IEEE.

Campos, L. M. (2006). A Scoring Function for Learning Bayesian Networks Based On Mutual Information and Conditional Independence Tests. Journal of Machine Learning Research (7), 2149-2187.

Heyuan, Ch. Malonean, B & Wu X., J. (2011). Learning Optimal Bayesian Networks Using A* Search. NSF Grants IIS-0953723 and EPS-0903787, 21 IJCAI. Barcelona.

He, Ch., & Gao, X., (2018). Structure Learning of Bayesian Networks Based on The LARS-MMPC Ordering Search Method. Wuhan, China: Chinese Control Conference.

Yang, C., Junzhong, Ji., Liu, J., & Baocai Yin. (2016). Structural Learning of Bayesian Networks by Bacterial Foraging Optimization. International Journal of Approximate Reasoning, 69.

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. Machine Learning, 197-243.

Fast, A. S. (2010). Learning the Structure of Bayesian Networks with Constraint Satisfaction. Massachusetts: Ph.D. Thesis, Department of Computer Science, University of Massachusetts, February 2010.

Gadekallu, T. R., Bhattacharya, S., Maddikunta, P. R., Hakak, S., & Zada, W., et. al. (2020). Antlion Re-Sampling Based Deep Neural Network Model for Classification of Imbalanced Multimodal Stroke Dataset. Multimedia Tools and Applications, pp. 1573-7721.

Cooper, G. F., & Herskovits, E. (1992). A Bayesian Method for The Induction of Probabilistic Networks from Data. Machine Learning, 9, (Pp. 309–347).

Gadekallu, T. R., & Khare, N. (2017). Cuckoo Search Optimized Reduction and Fuzzy Logic Classifier for Heart Disease and Diabetes Prediction. International Journal of Fuzzy System Applications (IJFSA), 25-42.

GU, W. (2020). Improved Antlion Optimizer Algorithm Base On Dynamic Random Hill-Climbing Mechanism. IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 1675-1679.

GUPTA, E. & SAXENA, A. (2016). Performance Evaluation of Antlion Optimizer Based Regulator in Automatic Generation Control of Interconnected Power System. Journal of Engineering., 1-14.

Kilic, H., Yuzgec, U., & Karakuzu, C. (2019). Improved Antlion Optimizer Algorithm and Its Performance on Neuro Fuzzy Inference System. Neural Network World, 2019(4), 235- 254.

Hesar, A. S. (2013). Structure Learning of Bayesian Belief Networks Using Simulated Annealing Algorithm. Middle-East Journal of Scientific Research 18, 1343-1348.

Cowie, J., Oteniya, L., & Coles, R. (2007). Particle Swarm Optimizations for Learning Bayesian Networks. Engineering and Physical Sciences Research Council.

Wang, Ji. & liu, Sa. (2018). Novel Binary Encoding Water Cycle Algorithm for Solving Bayesian Network Structures Learning Problem. Knowledge-Based Systems, 150.

Junyi, L., & Jingyu C., (2014). A Hybrid Optimization Algorithm for Bayesian Network Structure Learning Based on Database. Journal of Computers, VOL. 9.

Junzhong, Ji., Hongkai, Wei., & Chunnian, Liu. (2012). An Artificial Bee Colony Algorithm for Learning Bayesian Networks. Springer-Verlag Berlin Heidelberg.

Orphanou, K., Thierens, D. & Peter, A. N. (2018). Learning Bayesian Network Structures with GOMEA. Kyoto, Japan: ACM.

Salama, K. M., & Freitas, A. (2012). ABC-Miner: An Ant-Based Bayesian Classification Algorithm.

KILIC, H. & YUZGEC, U. (2019). Improved Antlion Optimization Algorithm Via Tournament Selection and Its Application to Parallel Machine Scheduling. Computers & Industrial Engineering, 132, 166-186.

Qianglong, L., Xiaowei, Z., Kun, Z., Jianrui, S., & Kai, W. (2021). State of Health Estimation of Lithium-Ion Battery Based on Ant Lion Optimization and Support Vector Regression," 2021 IEEE International Conference on Electrical Engineering and Mechatronics Techno.

Awalivian, F., Raihan, M., Suyanto & Siti, S., (2021). Optimization of Aircraft Flight Scheduling and Routing Problem Using Multi-Objective Antlion Optimization. International Conference on Artificial Intelligence and Computer Science Technology (ICAICST). 1-6.

Askari, M. A., & Ahsaee, M. G. (2018). Bayesian Network Structure Learning Based on Cuckoo Search Algorithm. 2018 6th Iranian Joint Congress On Fuzzy and Intelligent Systems (CFIS). Kerman, Iran.

Margaritis, D. (2003). Learning Bayesian Network Model Structure from Data. Pittsburgh, PA. Available as Technical Report CMU-: Carnegie-Mellon University.

Mei R.N.S. M., Sulaiman, M.H., & Zuriani, M. (2015). Antlion Optimizer for Optimal Reactive Power Dispatch Solution. Journal of Electrical Systems, 3, 68-74.

Nischal, M. M., & Mehta, Sh., (2015). Optimal Load Dispatch Using Ant Lion Optimization. Journal of Engineering Research and Applications, 10-19.

Mirjalili, S. (2015). The Ant Lion Optimizer. Advance Engineering Software, 80–98.

Nathan, F., Sheppard, J., & Pillai, K. G. (2013). Abductive Inference Using Overlapping Swarm Intelligence. IEEE Symposium on Swarm Intelligence.

Larraiiaga, P., & Poza, M. (1996). Structure Learning of Bayesian Networks by Genetic Algorithms. Springer-Verlag Berlin Heidelberg Gmbh.

Patrick, O. D., & Sahin, F. (2004). Structural Learning of Bayesian Networks from Complete Data using the Scatter Search Documents. IEEE International Conference on Systems, Man and Cybernetics.

Nagarajan, R. Scutari, M. & Lèbre, S. (2013). Bayesian Networks in R with Applications in Systems Biology. New York: Springer.

Sencer, S., Oztemel, E., Harun Taskin, H., & Torkul, O. (2013). Bayesian Structural Learning with Minimum Spanning Tree Algorithm. The World Congress in Computer Science, Computer Engineering, and Applied Computing.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). A Grey Wolf Optimizer. Advances in Engineering Software, 69, 46-61.

Kareem, S., & Okur, M. C. (2018). Bayesian Network Structure Learning Using Hybrid Bee Optimization and Greedy Search. Adana / TURKEY: Çukurova University.

Kareem, S. & Okur, M. C. (2017). Evaluation of Bayesian Network Structure Learning. 2nd International Mediterranean Science and Engineering Congress (Pp. 1313-1319). Adana: ÇUKUROVA UNIVERSITY.

Kareem, S., & Okur, M. C. (2021, 22 4). Falcon Optimization Algorithm for Bayesian Network Structure Learning. Computer Science, Pp. 553-569.

Kareem, S., & Okur, M. C. (2019). Bayesian Network Structure Learning Based on Pigeon Inspired Optimization. International Journal of Advanced Trends in Computer Science and Engineering, 8(1.2), 131-137.

Kareem, S., & Okur, M. C. (2019). Pigeon Inspired Optimization of Bayesian Network Structure Learning and a Comparative Evaluation. Journal of Cognitive Science, 535-552.

Kareem, S., & Okur, M. C. (2020). Structure Learning of Bayesian Networks using Elephant Swarm Water Search Algorithm. International Journal of Swarm Intelligence Research, 19-30.

Zhang, S., & Liu, L. (2008). Mcmc Samples Selecting for Online Bayesian Network Structure Learning. Kunming, IEEE.

Shun, Li., & Wang, B., (2017). A Method for Hybrid Bayesian Network Structure Learning from Massive Data Using Mapreduce. IEEE.

Rahier, T., Marie, S., Girard, S., & Forbes, F.(2019). Fast Bayesian Network Structure Learning Using Quasi-Determinism Screening. HAL, 2, 14-24.

Gadekallu, T. R., & Khare, N. (2016). FFBAT-Optimized Rule Based Fuzzy Logic Classifier for Diabetes. International Journal of Engineering Research in Africa Trans Tech Publications, 137-152.

Fan, Xi., & Malone, C. B. (2014). Tightening Bounds for Bayesian Network Structure Learning. Association for The Advancement of Artificial Intelligence.

YAO, P. & Wang, H. (2017). Dynamic Adaptive Antlion Optimizer Applied to Route Planning for Unmanned Aerial Vehicle. Soft Computing., 21(18), 5475-5488.

Yogarajan, G., & Revathi, T. (2016). A Discrete Ant Lion Optimization (DALO) Algorithm for Solving Data Gathering Tour Problem in Wireless Sensor Networks. Middle-East Journal of Scientific Research, 3113±3120.

Zawbaa, H., Emary, E., & PARV B. (2015). Feature Selection Based on Antlion Optimization Algorithm. Third World Conference on Complex Systems (WCCS). Morocco: IEEE.

AUTHOR BIOGRAPHIES

Dr. Shahab Wahhab Kareem, Shahab Wahhab Kareem his BSc in Control and Computer Engineering from the University of Technology Baghdad in 2001, MSc in Software Engineering from Salahadeen University in 2009, and Ph.D. in Yasar University Izmir, Turkey in 2020. Dr. Wahhab's research interests include Machine learning and BIG DATA. Dr. Kareem is a lecturer at the Information System Eng. (ISE) Department (2011-till now). He can be contacted at email: shahab.kareem@epu.edu.iq or shahabwk@yahoo.com.Machine learning and BIG DATA. I'm a lecturer at the Information System Eng. (ISE) Department (2011-till now). He can be contacted at email: shahab.kareem@epu.edu.iq or shahabwk@yahoo.com.

Dr. Amin Salih Mohammed
Dr. Amin Salih Mohammed is currently serving as President of Lebanese French University, Erbil, Kurdistan region-Iraq. Before being promoted as President, he has served as Vice President, Scientific Affairs of the university. He has more than 15 years of teaching and research experience. Amin is an active researcher and acted as a resource person for various workshops and faculty development programs organized by different institutions. As a tech-savvy, He has completed his undergraduate, postgraduate engineering degree and Ph.D. in Computer Engineering from the Kharkiv National University of Radio Electronics, Kharkiv, Ukraine. His research fields are computer networks, wireless networks, and cloud computing. So far, he has published more than 50 research articles in various reputed international journals indexed in SCI & Scopus services and reputed conferences.

Hoshang Qasim Awla,
2012-BSc in Computer Science from Soran University/Iraq, 2016- MSc in Computer Engineering from Hassan Kalyouncu University/Turkey. Ph. D. student at Soran university. Assistant lecturer at computer science department/ Soran university, email address hoshang.awla@soran.edu.iq