

SHAHAB WAHHAB KAREEM
MEHMET CUDI OKUR

FALCON OPTIMIZATION ALGORITHM FOR BAYESIAN NETWORK STRUCTURE LEARNING

Abstract

In machine-learning, some of the helpful scientific models during the production of a structure of knowledge are Bayesian networks. They can draw the relationships of probabilistic dependency among many variables. The score and search method is a tool that is used as a strategy for learning the structure of a Bayesian network. The authors apply the falcon optimization algorithm (FOA) to the learning structure of a Bayesian network. This paper has employed reversing, deleting, moving, and inserting to obtain the FOA for approaching the optimal solution of a structure. Essentially, the falcon prey search strategy is used in the FOA algorithm. The result of the proposed technique is associated with pigeon-inspired optimization, greedy search, and simulated annealing that apply the BDeu score function. The authors have also examined the performances of the confusion matrix of these techniques by utilizing several benchmark data sets. 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, global search, falcon optimization algorithm, structure learning, search and score

Citation

Computer Science 22(4) 2021: 553–569

Copyright

© 2021 Author(s). This is an open access publication, which can be used, distributed and reproduced in any medium according to the Creative Commons CC-BY 4.0 License.

1. Introduction

One of the simplified analytical models for constructing the probabilistic structure of knowledge in machine-learning is a Bayesian network (BN) [13]. Such a model can be applied as a learning method for combining knowledge, arguments, and inference [8]. The structure of a Bayesian network is a directed acyclic graph (DAG) that is composed of two significant parts; the parameters, and the structure of the network. The conditional probabilities are represented as parameters, and the dependencies among the variables are displayed as the structure. It is difficult to solve a Bayesian network's structure learning without a proper search method. The difficulties for learning from a data set for the structure of a Bayesian network to achieve the optimal is NP-hard class [20]; however, a comprehensive investigation becomes conducted to improve the approximate approaches for the learning structure of the network. Usually, there are two types of structural learning procedures for Bayesian networks. The first type is a constraint-based procedure, and the second is a classification and search procedure [22]. The score and search method is applied to examine the range that concerns the structures of a BN, which include continuously estimating all candidate network structures until an actual metric score is obtained. Score-based methods depend on a function to evaluate the network and accessible data and constantly look for a structure that improves the score (which is the ultimate goal) [7]. The score function method is implemented by using two primary criteria: a Bayesian score, and an information-theoretic score. A Bayesian score is performed in some other techniques like K2, BD (Bayesian Dirichlet), BDe (Bayesian Dirichlet [the "e" stands for likelihood-equivalence]), and BDeu (Bayesian Dirichlet equivalent uniform [the "u" stands for uniform joint distribution]). The information-theoretic score is implemented in techniques such as the Akaike information criterion (AIC), log-likelihood (LL), minimum description length (MDL), Bayesian information criterion (BIC), mutual information test (MIT), and normalized minimum likelihood (NML) [3]. There are various techniques of a research strategy that are intended to improve the problem of structural learning; these include particle swarm intelligence [4], the ant colony optimization algorithm [27], bee colony [13], the hybrid algorithm ([11, 15, 21]), the simulated annealing algorithm [26], bacterial foraging optimization [33], genetic algorithms [19], the gene-pool optimal mixing evolutionary algorithm (GOMEA) [24], the breeding swarm algorithm [18], the binary encoding water cycle [32], pigeon-inspired optimization [16], tightening bounds [6], A* search algorithms [34], scatter search documents [5], the cuckoo optimization algorithm [1], quasi-determinism screening [25], and the minimum spanning tree algorithm [28]. Another additional metaheuristic technique that can be applied to learn the structure of Bayesian networks is falcon optimization. Here, this article proposes and presents a relative evaluation of this approach as a new method for solving the learning problem of a Bayesian network structure. The models of BN integrate with the administration for decision networks, the fundamental formulation of causal systems, mixed continuous and discrete variables, quantum probability, Bayesian neural networks, state-and-transition

standards, object-oriented and agent-based standards, geographic information systems, and other fields. BNs are becoming valuable mechanisms in risk management, risk analysis, and decision science for resource planning and environmental management. BNs are natural and compact graphical descriptions that can be utilized to manage causal reasoning and risk evaluation examination and allow many benefits beyond regression-based approaches [14]. A Bayesian network is used to present a short description of the relationship among the appearance of many chronic diseases and patient-level risk circumstances over time [14, 17]. A structure-learning challenge can be viewed as an inference problem where the variables define a selection of parents for any node within a graph. The major combinatorial problem arises from the global constraint that the graph structure must be acyclic. We called the structure learning problem a linear program over the polytope represented by valid acyclic structures. In decreasing this problem, the authors maintain an outer bound approximation to the polytope and imperatively stretch it by searching for a new kind of validity constraint. If a full solution is found, it is proven to be an optimal Bayesian network.

The arrangement of this article is as follows. Following the introduction, Section 2 presents the approach of Bayesian network structure learning in general. The short introduction of the falcon optimization algorithm is presented in Section 3. In Section 4, the authors present the methodology in detail and show the experimental results. The final section concludes the article.

2. Bayesian network structure learning

Essentially, one can express a Bayesian network utilizing two elements – $G; P$. The first element $G(V; E)$ is the DAG, which include the predictable group of nodes (or vertices), V , which is interconnected across identified links (or edges), and E . The second elements – $P = P(X_i | Pa(X_i))$ – describe the combination of conditional probabilistic distributions (CPD) that are specific to every variable X_i (vertices from a graph). Furthermore, $Pa(X_i)$ denotes the group of parents of node X_i in G [14]. Based on this model, a simple possibility group of a network ($G; P$) can be described by the following:

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)) \quad (1)$$

On the other hand, a score function is based on different principles (which include information and entropy), Bayesian approaches, and minimum description length [17]. According to the rules of Bayesian inference, the posterior probability of a Bayesian network can be expressed as follows:

$$P(G|D) = \frac{P(D|G).P(G)}{\sum_{G'} P(D|G').P(G')} \quad (2)$$

In (2), $P(D|G)$ is a finite probability, which is determined by using the normalization constant $P(D)$ as

$$P(D|G) = \int P(D|G, x) \cdot P(x|G) \cdot dx \quad (3)$$

$P(D)$ is assumed to be independent of Bayesian network structure G ; x represents the model parameter, and $P(G')$ is the prior probability. Therefore, as long as the limited probability of all potential structures is calculated, the posterior distribution of the network structure can be determined [4]. The methods of structural learning apply class-based procedures by analyzing past and current structure results. The final expression of the result is as follows [2]:

$$Score(G, D) = \sum Score(X_i, Pa(X_i), D(X_i, pa(X_i))) \quad (4)$$

3. Falcon optimization algorithm

Metaheuristics are nature-inspired algorithms for obtaining comparative solutions to any computationally difficult optimization problems. The swarming behaviors of animals (including firefly-BAT [35], cuckoo [9], ant, pigeon, fish, bee, etc.) have been used in metaheuristics [10]. The amazing characteristics behind the metaheuristics hold identity, illation-free tools, adaptability, and local optima eschewal ability [23].

In [31], the proposed metaheuristic algorithm depends on the falcon's behavior of hunting. The falcon optimization algorithm (FOA) is the reliable and robust algorithm of stochastic population-based problems that requires arrangements from several parameters to its three-stage action settlement.

The motivation of the proposed method was the chase style of falcons while they are seeking their prey during flight. Falcons are recluses, and their tactics for hunting depends on their requirements. However, specific tactics arise, and amazing models hold the fundamental precepts about the flight. Based on many products of Tucker [29,30], Among birds, high-performance flyers are falcons. In various states of elevated hunting, the fitting objectives are checked for the limits of flying achievement [30]. The implementation technique of flight in the framework include determining a standardized power about the flight, the flight average velocities, and adaptive responses to the wind [30]. One of the quickest animals in the world is a falcon; stoops have been shown to approach velocities that are faster than 300 km/h. Small thin tubercules in their beaks lead the air through high-speed stoops, allowing falcons to breath easily. The primary hunting is done throughout the day (including morning and night). They primarily feed on small and medium-sized birds, but their diets also include insects like cicadas, moths, and locusts (although such prey is rare) [12].

During flight, falcons take different routes to reach their prey. Each route has two parts: the first part is a logarithmic spiral on which a falcon continually keeps its head straight while peering at the prey with the highest visible acuity; and the second is when the falcon flies toward the prey in a straight segment – when the prey

is within the falcon's field of vision, the falcon dives. Therefore, a falcon's achieve locomotion can be classified into three steps: the initial step (first stage) – exploring for prey; the second step (second stage) – improving its dive through a logarithmic spiral; and the third step (third stage) – the dive itself (which can result in success; i.e., acquisition of prey). Otherwise, a falcon quickly reverses its action depending on it is experience.

The quick procedure, which includes five steps for the implementation of the FOA, is given below [31].

- Step 1: Start the algorithm by adjusting the parameters for the optimization problem, including the number of falcons (NP), highest speed ($Vmax$), cognitive rate (cc), social (sc) constant, following (fc) constant, dive probability (DP), and awareness probability (AP).
- Step 2: Set the velocity and position of the falcons randomly in a D-dimensional space based on the boundary conditions, where the position of each falcon is defined in consideration of the number of NP applicants within all of its D dimensions. The speeds are arbitrarily produced among the $Vmax$ and $Vmin$ limitations, where both are respectively determined as follows:

$$Vmax = 0.1 * ub \quad (5)$$

$$Vmin = -Vmax \quad (6)$$

where ub denotes the upper bound (the boundary area concerning each dimension). In the beginning, generate the pairs of numbers randomly (pAP, pDP) for each falcon for correspondence among the dive and awareness probabilities.

- Step 3: Calculate the fitness value and select the best ($xbest$) and global ($gbest$) sites. The selected positions will be used to produce new positions considering the logic that rules the move behind the dive and awareness probabilities.
- Step 4: New locations are produced, including updating the location of the falcon. Then, compare pAP with the probability of awareness AP ; if AP is bigger than pAP , the falcon moves from seeking for prey based on its activity (including some different experiences of the other falcons):

$$X_{iter+1} = X_{iter} + V_{iter} + cc(X_{best}, X_{iter} + sc(gbest, X_{iter})) \quad (7)$$

where V_{iter} is the current velocity and X_{iter} is the current position of the falcon. If pAP is bigger than AP , formerly compare a dive likelihood DP among pDP . If DP is less than pDP , then one of the targets is chosen as prey by the falcon (X_{chosen}), and it completes its fundamental step toward hunting. A logarithmic spiral is provided through

$$x_{iter+1} = X_{iter} + |X_{chosen} - x_{iter}|.exp^{bt} \cos(2\pi t) \quad (8)$$

where b is a fixed number that determines the state of the spiral logarithm (equal to 1), and t is an arbitrary number within range $(-1, 1)$ that determines

the next location of the falcon with respect to its exact destination [31]. If AD is bigger than pAP , formerly compare the score function of the preferred prey and the score function of the falcon. Wherever the prey is most appropriate, it will be followed through by the falcon related to a dive step:

$$X_{iter+1} = X_{iter} + V_{iter+1} + fc * rand(X_{chosen} - X_{iter}) \quad (9)$$

otherwise, falcon continues to fly based in its best position:

$$X_{iter+1} = X_{iter} + V_{iter+1} + cc * rand(X_{best}, X_{iter}) \quad (10)$$

The new location that is evaluated later concerns the velocities and location boundaries. Next, its new score function is computed, and the new values of X_{best} and $gbest$ are determined.

Step 5: Last, subsequent evaluations of Step 4 are continued until the highest number of iterations (itermax) is reached.

Algorithm: Structure Learning of Bayesian Network Based on
Falcon Optimization Algorithm

INPUT: – datasets Population size, NP ; Maximum speed, $Vmax$; Values of cognitive Cc , social, Sc and following Fc , constant. Value of awareness Probability (AP) and Dive probability (DP); $tmax$: maximum iteration number; $Xmax$: upper boundary, and $Xmin$: lower boundary.

OUTPUT: – learning Bayesian network.

1. Initialized empty structure and initialize parameters of FOA algorithm (dimension space D , size of population NP , constant values of Cc , Sc , and Fc , Awareness AP and Dive DP probability, number of iterations, upper boundary, and lower boundary, $(G_{best,i,d}^t)$.
2. Set velocity and position for all falcons randomly, Compare each falcon by BDe score function and find best in current position $(P_{best,i,d}^t)$.
3. For loop to maximum iteration number.
4. For loop to size of population.
5. Generate random value pAP , pDP . Select new best position by comparing BDe score function of each falcon.
6. if $pAP < AP$, update falcon velocity (V_i, d) using Equation (7); else, if $pDP > DP$, update falcon velocity using Equation (8). else, compare score function of current and previous one. If this one is better, update falcon velocity using Equation (9); otherwise, use Equation (10).
7. Update position X_i .

8. Evaluate BDeu score function of new position ($X_{i,d}^t$).
 - (a) If current position ($X_{i,d}^t$) is better than best position ($P_{best,i,d}^t$), then update best position by ($P_{best,i,d}^t = X_{i,d}^t$).
 - (b) If ($G_{best,i,d}^t < X_{i,d}^t$) greater than current position, then update best solution for global by ($G_{best,i,d}^t < X_{i,d}^t$).
 - (c) Best score value and solution are saved.
 - (d) If ($X_{min} \geq X_{max}$), stop iteration process; results are presented. If not, move to Step 5.
 9. Return maximum BDe score.
-

4. Structure learning of Bayesian network using FOA

The FOA is a proposed algorithm that can be applied for the structure learning of a Bayesian network. It uses the BDeu score function as a score metric for evaluating the structure of the Bayesian network. The FOA algorithm is an effective iterative method that depends on a society of individuals where each falcon encodes a possible location and velocity in a specific area. This area is held to be the search area. The proposed method is based on different procedures. The initial procedure utilizes Equation (7), which concerns exploration within the essential process if ($AP < pAP$); otherwise, the secondary procedure uses Equation (8). If $DP < pDP$ then the falcon target one chosen prey (X_{chosen}), and performs its initial movement for hunting, otherwise comparing the BDeu score function of both steps during choosing the most suitable location that is considered under Equations (9, 10). The algorithm above presents the pseudo-code of this procedure. The solution to the FOA's structure appropriates various neighborhoods in the exploration area. The expectation is powerful for updating a solution that is developed for a local search through the group of the fan falcon. The solution area of the Bayesian network structure learning is made for each possible DAG. Each falcon begins a potential solution, which is described as a DAG with empty arcs. A falcon next explores some examination space to obtain the approaching optimal or near-optimal solution, which is essentially recognized as the BDeu score. Applying Equation (4) determines the BDeu score as the goal function of the optimization. Achieving a higher (or the best) BDeu score for the structure of the Bayesian network is the goal of the search process. Each of the initial solutions is provided by iterative processes. Beginning by a clear graph (G_0) with no arcs at the initial states, the arcs are added one after another (provided that they are not covered in the popular graph solution). The process for appending performs if and only if the new solution's score function is more powerful than the current score and the new solution satisfies the DAG constraint. This procedure continue until the number of arcs is equal to the number that is defined in the progress. During the design, the solution begins to select a population for each iteration, including selecting a candidate solution that has a more powerful score function. The falcon proceeds

according to the chosen operative until the method has achieved the highest number of iterations or the BDeu score does not grow anymore. In general, the processes contain four separate operations in the optimization: addition, deletion, movement, and reversion. Addition, deletion, and reversion are simple actions inside this region, including merely replacing an original edge each time from a competitor solution. This enables the inclusion of a relatively small region that is near the solution. On the other hand, the actual edges adjust the set of parents with each movement action, which can make a moderately important change for the current solution. Accordingly, if the solution is not modified after applying simple actions, the move action may update it. Diving is the principal force that uses the preferred procedure within the local optimization, which expands more comprehensively while a falcon approaches the acceptable solution. Diving is a driving force utilizing the same local optimization operator, which grows more widespread as a falcon approaches a desirable solution. Flying directions, the switch with various local optimization operators, which grows extra widespread as a falcon moves continuously from a solution to search for a better one.

Accordingly, the current velocity update by both falcon's best local or best global solutions depend on the values of (DP and AP). The speed of the FOA is updated depending on the current most suitable location of the falcon in the search area. Figure 1 shows Falcon G_0 , which represents a DAG that includes arcs, tries addition, reversion, move, and deletion, and sequentially approaches new solutions G_1 , G_2 , G_3 , and G_4 . Considering that the highest score is in G_3 , it will be chose; then, the falcon will continue to experiment on some comparable process to essentially get $G+3$ as the new solution. If the BDeu score of $G+3$ is more powerful than that of $G+1$, the falcon will proceed to complete a similar operative. The procedures will iterate until the BDeu score stabilizes or the repetition loop equals the maximum. In the full procedure, the falcon chooses among the directions that utilize deletion, movement, reversion, and addition.

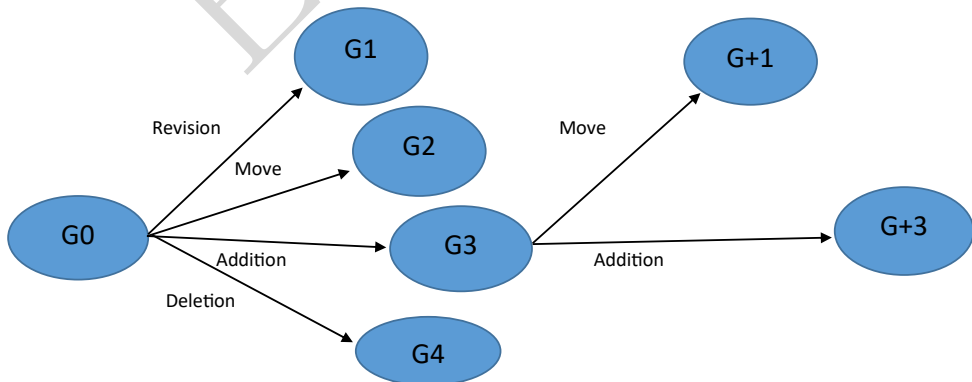


Figure 1. Searching steps for one falcon

5. Experimental evaluation

To assess the FOA's performance, a standard evaluation technique is used by employing probability datasets obtained from the Bayesian network common criteria. The test platform includes a computer that has the following characteristics: Core i3, 2.1GHz CPU, 4GB RAM, Ubuntu 14.04, and using Java to implement the algorithms. The authors studied the characteristics of the proposed method in several static datasets, including Lucap02 (143 variables and 10,000 instances), Andes (223 variables, 338 arcs, and 500 instances), win95pts (76 variables, 112 arcs, and 574 instances), Hepar (70 variables, 123 arcs, and 350 instances), Hailfinder (56 variables, 66 arcs, and 2,656 instances), Alarm (37 variables, 46 arcs, and 10,000 instances), Soybean (35 variables and 307 instances), Hepatitis (35 variables and 137 instances), Static Banjo (33 variables and 320 instances), Water (32 variables, 66 arcs, and 10,083 instances), Epigenetics (30 variable and 72,228 instances), Insurance (27 variables, 52 arcs, and 3,000 instances), Sensors (25 variables and 5,456 instances), Mushroom (23 variables and 1,000 instances), Parkinsons (23 variables and 195 instances), Heart (22 variables and 267 instances), Imports (22 variables and 205 instances), Child (20 variables, 25 arcs, and 230 instances), Letter (17 variables and 20,000 instances), Adult (16 variables and 30,162 instances), Lucas01 (10 variables and 10,000 instances), WDBC (9 variables and 1,000 instances), and Asia (8 variables, 8 arcs, and 3,000 instances) [12].

In the paper, this work is dependent on an assumption of static data, and the learning data sets that the authors examined are stationery sets. Enlarging the FOA to sensor data sets or different forms of online current data sets is a challenging task and could be attempted after assessing its review over stationary data sets.

In this paper, the authors compared the results with pigeon-inspired optimization (PIO) [16], greedy search (GS) methods, and simulated annealing (SA) by using similar metrics for the data sets. Next, to define the parameters of the FOA, they estimated whole algorithms under identical conditions. In the FOA, the following values were employed for the experiments: population size $N = 25$, and $tmax = 1,000$. The constant parameters of the FOA's optimization are $Cc = 2$, $Sc = 3$, $Fc = 4$, $AP = 0.3$, $Vmax = 0.1ub$ (ub is 100, and $Vmax$ is 10), (t) is a random number within a range of $[-1, 1]$, and DP is 0.85. The parameters of the greedy search are as follows: the minimum number of recommended networks after the highest score = 1,000, the recommended minimum networks before reboot = 3,000, the maximum parent count for operations reboot = 5, the maximum recommended networks before reboot = 5,000, and restart by random network = yes. The parameters of the simulated annealing algorithms are as follows: the temperature of re-annealing = 500, the cooling factor = 0.8, and the initial temperature = 1,000. The algorithms were performed for three distinct execution times: 2, 5, and 60 minutes.

Tables 1, 2, and 3 show the score function values for the algorithms in the above-mentioned data sets (including the different times). It can be noted from the results that the proposed method provides more reliable score values than PIO, the simulated

annealing algorithms, and the default greedy search during most comparison states. This means that the FOA obtains the best score with the minimum necessary time.

Table 1

Calculation results of score function values for FOA, simulated annealing, and greedy for 2 minutes of execution time

Dataset	Falcon	PIO	Simulated Annealing	Greedy
Parkinsons	-1,598.91	-1,598.91	-1,601.2968	-1,732.76
Hepatitis	-1,016.34	-1,327.73	-1,330.464	-1,350.16
Imports	-1,773.19	-1,811.99	-1,828.91	-1,994.15
Heart	-2,348.94	-2,423.86	-2,432.18	-2,576.93
Mushroom	-3,345.92	-3,372.51	-3,375.31	-3,734.22
WDBC	-6,668.51	-6,666.04	-6,682.72	-8,089.41
win95pts	-45,978.55	-46,779.58	-47,085.10	-83,749.34
Sensors	-60,343.34	-60,343.3	-60,710.4985	-69,200.39
Hepar	-160,095	-160,095	-161,086.426	-169,497
Letter	-173,090.07	-175,200	-178,562.21	-184,307
Epigenetics	-177,511.65	-176,657	-179,910.33	-225,346
Adult	-205,984.89	-207,809	-211,677.72	-211,844

Table 2

Calculation results of score function values for FOA, simulated annealing, and greedy for 5 minutes of execution time

Dataset	Falcon	PIO	Simulated Annealing	Greedy
Hepatitis	-1,011.88	-1,327.73	-1,330.46	-1,350.16
Parkinsons	-1,598.91	-1,598.91	-1,601.32	-1,721.16
Imports	-1,768.90	-1,811.99	-1,828.91	-2,012.21
Heart	-2,335.44	-2,423.82	-2,423.82	-2,560.43
Mashroom	-3,345.92	-3,372.51	-3,375.31	-3,706.66
WDBC	-6,603.96	-6,666.04	-6,682.72	-7,954.65
win95pts	-43,850.27	-46,779.53	-47,085.15	-83,150.71
Sensors	-59,895.45	-60,343.32	-60,710.54	-69,150
Hepar	-160,082	-160,095	-161,086	-169,881
Letter	-173,090.07	-175,200	-178,562	-184,916
Epigenetics	-176,451.98	-176,657	-179,300	-224,172
Adult	-20,551.48	-207,809	-211,678	-211,781

Table 3

Calculation results for score function values for FOA, simulated annealing, and greedy for 60 minutes of execution time

Dataset	Falcon	PIO	Simulated Annealing	Greedy
Hepatitis	-1,010.02	-1,327.73	-1,330.46	-1,350.16
Parkinsons	-1,598.90+1	-1,598.91	-1,601.32	-1,700.36
Imports	-1,755.31	-1,811.99	-1,828.91	-1,995.76
Heart	-2,325.48	-2,423.82	-2,432.19	-2,527.44
Mashroom	-3,000.96	-3,372.51	-3,375.31	-3,588.69
WDBC	-6,574.21	-6,666.04	-6,682.72	-7,841.35
win95pts	-39,814.78	-46,779.5	-47,085.14	-81,779.52
Sensors	-58291.874	-60343.3	-60710.52	-68,364
Hepar	-160,055	-160,095	-161,086	-16,8871
Letter	-173,090.07	-175,200	-178,562	-184,118
Epigenetics	-176,235.58	-176,657	-179,300	-217,246
Adult	-20,535.93	-207,809	-211,678	-211,762

To assess the success of the discovery of the structure, the confusion matrix was calculated for each data set and its known network structure. The TP , TN , FN , and FP metrics were calculated for each network for each algorithm to additionally obtain the following criteria: sensitivity (SE), F1 score, accuracy (Acc), and AHD, which are defined as follows:

$$sensitivity = \frac{TP}{TP + FN} \quad (11)$$

$$F1_{score} = \frac{2 * TP}{2 * TP + FP + FN} \quad (12)$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (13)$$

$$HD = \frac{FN + FP}{TP + TN + FP + FN} \quad (14)$$

The meanings of these metrics are as follows: TN is an arc (vertex or edge) that is inside neither the learning network nor the natural network. TP is the arc in the correct place inside the learning network. FN is an arc in the natural network but not in the learning network. FP is an arc inside the learning network but not in the natural network. The sensitivity results for simulated annealing, the FOA, PIO, and greedy are shown in Figure 2. The proposed method produces better values than simulated annealing, PIO, and greedy in most datasets.

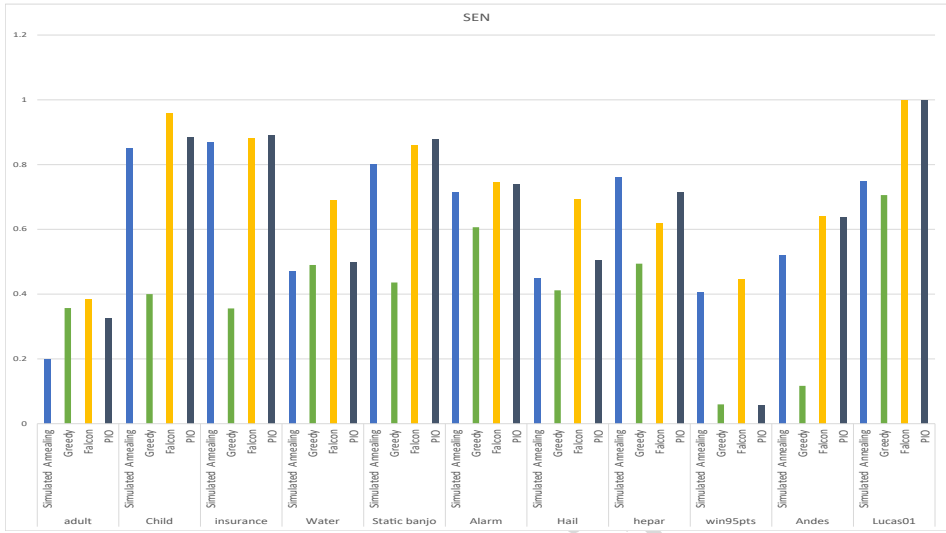


Figure 2. Sensitivity for FOA, PIO, SA, and GS

Furthermore, the proposed method had greater accuracy values than the simulated annealing, PIO, and greedy algorithms in most datasets (as shown in Figure 3).

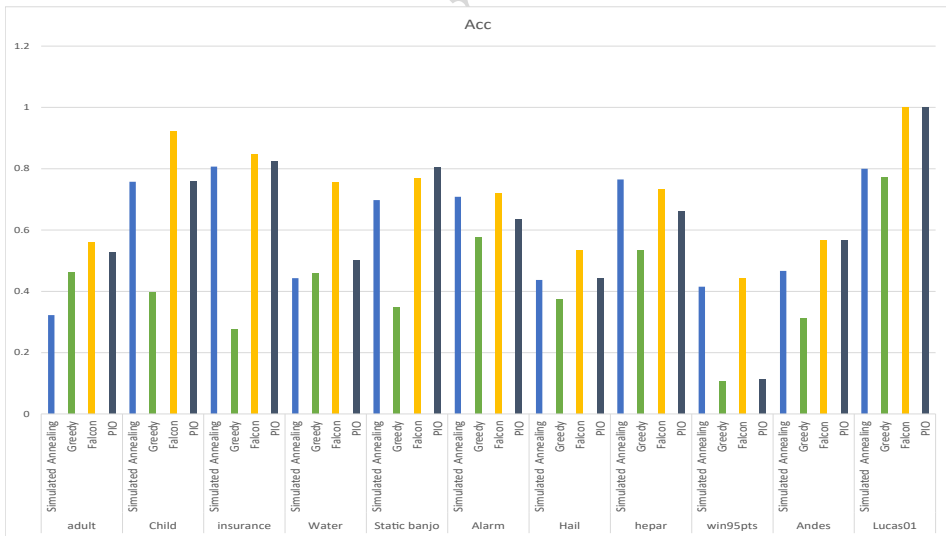


Figure 3. Accuracy for FOA, PIO, SA, and GS

The learning algorithm that was proposed works well in obtaining a suitable structure. As a result, the iterative FOA algorithm is the most suitable algorithm when compared to the other algorithms in most datasets from the point of view of

prediction accuracy. The FOA is also better than the other algorithms from the point of view of construction times. For the performance metrics, we used F1 as a metric of the model’s accuracy in addition to the best score in the Bayesian results.

The F1-score, precision, and recall are used to evaluate the performance of the proposed algorithm. Under these circumstances, precision is the number of directed edges that are found to be correctly divided by the number of all of the edges in the expected BN. recall represents the division of the number of directed edges that are found by the number of edges in the actual BN. It does know that F1 is the harmonic average of accuracy and recall. Figure 4 presents the comparison of the FOA, PIO, simulated annealing, and greedy search.

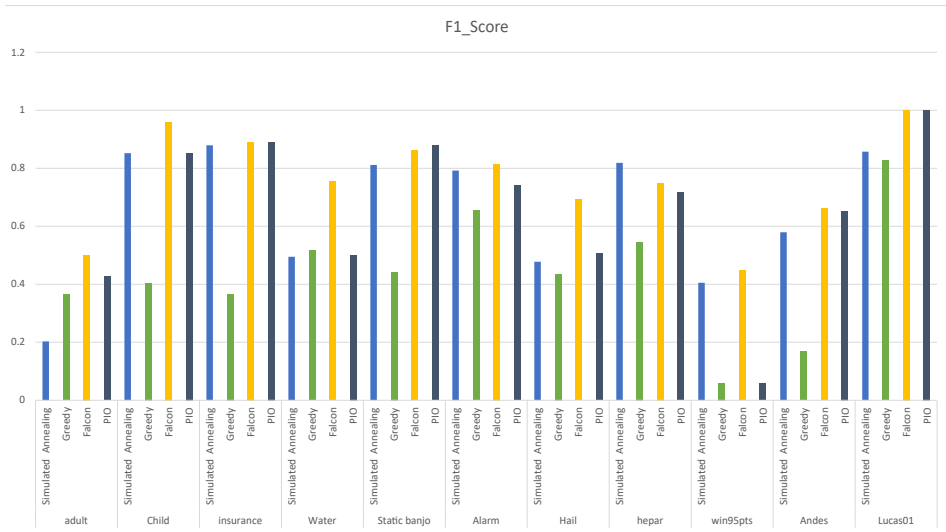


Figure 4. F1 Score for FOA, PIO, SA, and GS

As presented in Figure 4, the proposed methods are more successful than the PIO, greedy search, and simulated annealing methods. Furthermore, the ultimate purpose of the model is to present a convenient representation of the real world, so accuracy is a useful measure of model performance evaluation. The proposed algorithm is also preferable regarding Hamming distances, which are always considerably lower than those that are obtained by using the DAG space.

Hamming distances is one of the most widely used evaluation metrics for BN structure learning, which directly matches the structure of learners and local networks and are also directed entirely toward exploration rather than inference. Figure 5 shows the average Hamming distances for the mentioned algorithms. The results demonstrate that the proposed method produces better performance values than the other methods that we considered.

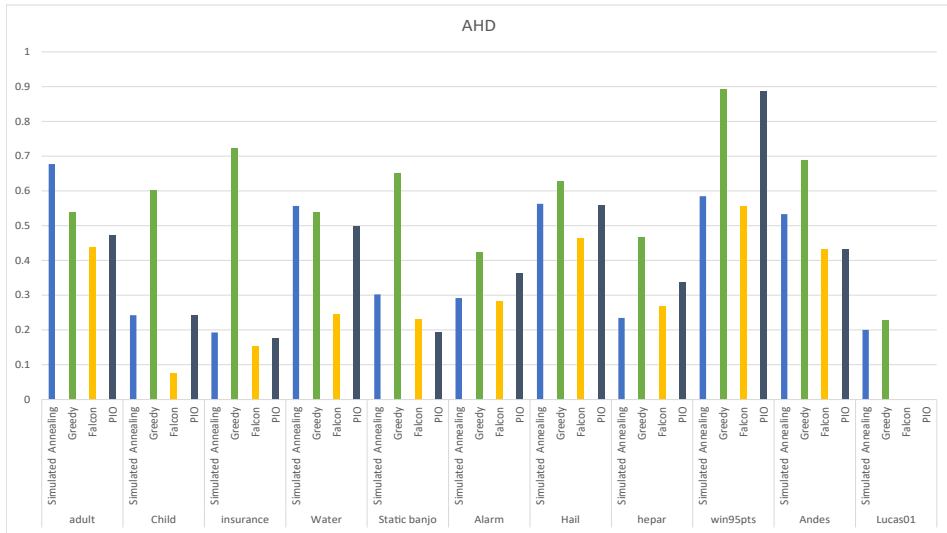


Figure 5. AHD for FOA, PIO, SA, and GS

6. Conclusion

The authors concentrated on the structure learning of a Bayesian network problem and utilized the falcon-inspired optimization procedure for Bayesian network structure learning. The authors applied the score and search technique, appropriating the FOA method as the search and BDeu as the score function. The FOA can be expressed as a stochastic search technique that is dependent on the navigational habits of falcons. In particular, the FOA is a common approach for exploring a discrete solution space; as such, it can be customized to suit any application domain. The concentration control in the FOA presents a quickened concentration to global extremum through providing a falcon to fly following a logarithmic spiral to the shortest useful solution space. The proposed method has a higher ability for searching, which indicates it can discover a more useful structure solution, measure better score function values, and best approximate to a network structure; in addition, the results are accurate. The global search enhances through the steps of the algorithm and immediately drives to global convergence. The authors are planning to further evaluate other essential properties of the FOA, such as resource consumption, analyses of run times, overall performance when utilizing further data sets, and experimental setups.

References

- [1] Askari M.B.A., Ahsae M.G.: Bayesian network structure learning based on cuckoo search algorithm. In: *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, pp. 127–130, 2018.

- [2] Campos de L.M.: A Scoring Function for Learning Bayesian Networks based on Mutual Information and Conditional Independence Tests, *Journal of Machine Learning Research*, vol. 7(77), pp. 2149–2187, 2006.
- [3] Cooper G.F., Herskovits E.: A Bayesian method for the induction of probabilistic networks from data, *Machine Learning*, vol. 9, pp. 309–347, 1992.
- [4] Cowie J., Oteniya L., Coles R.: Particle Swarm Optimisation for learning Bayesian Networks. In: *Proceedings of the World Congress on Engineering 2007, Vol. I*, 2007.
- [5] Djan-Sampson P.O., Sahin F.: Structural Learning; of Bayesian Networks from Complete Data using the Scatter Search Documents. In: *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, vol. 4, pp. 3619–3624, 2004.
- [6] Fan X., Yuan C., Malone B.: Tightening Bounds for Bayesian Network Structure Learning. In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 2439–2445, 2014.
- [7] Fast A.S.: *Learning The Structure of Bayesian Networks with Constraint Satisfaction*, Ph.D. thesis, University of Massachusetts Amherst, Department of Computer Science, 2010. https://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1174&context=open_access_dissertations.
- [8] Fortier N., Sheppard J., Pillai K.G.: Bayesian abductive inference using overlapping swarm intelligence. In: *2013 IEEE Symposium on Swarm Intelligence (SIS)*, pp. 263–270, 2013. doi: 10.1109/SIS.2013.6615188.
- [9] Gadekallu T.R., Khare N.: Cuckoo Search Optimized Reduction and Fuzzy Logic Classifier for Heart Disease and Diabetes Prediction, *International Journal of Fuzzy System Applications (IJFSA)*, vol. 6(2), pp. 25–42, 2017. doi: 10.4018/IJFSA.2017040102.
- [10] Gandomi A.H., Yang X.S., Marand S.T., Alavi A.H.: *Metaheuristic applications in structures and infrastructures*, Elsevier, Oxford, 2013.
- [11] He C.C., Gao X.G.: Structure Learning of Bayesian Networks Based On the LARS-MMPC Ordering Search Method. In: *2018 37th Chinese Control Conference (CCC)*, pp. 9000–9006, 2018. doi: 10.23919/ChiCC.2018.8483049.
- [12] Hedenström A., Rosén M., Åkesson S., Spina F.: Flight performance during hunting excursions in Eleonora’s falcon *Falco eleonora*, *Journal of Experimental Biology*, vol. 202(15), p. 2029–2039, 1999. doi: 10.1242/jeb.202.15.2029.
- [13] Ji J., Wei H., Liu C.: An artificial bee colony algorithm for learning Bayesian networks, *Soft Computing*, vol. 17, pp. 983–994, 2012. doi: 10.1007/s00500-012-0966-6.
- [14] Kareem S.W., Okur M.C.: Evaluation Of Bayesian Network Structure Learning. In: *2nd International Mediterranean Science and Engineering Congress*, pp. 1313–1319, Çukurova University, 2017.

- [15] Kareem S.W., Okur M.C.: Bayesian Network Structure Learning Using Hybrid Bee Optimization and Greedy Search. In: *3rd International Mediterranean Science and Engineering Congress (IMSEC 2018)*, pp. 1–7, Çukurova University, 2018.
- [16] Kareem S.W., Okur M.C.: Bayesian network structure learning based on pigeon inspired optimization, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8(1), pp. 131–137, 2019.
- [17] Kareem S.W., Okur M.C.: Structure Learning of Bayesian Networks Using Elephant Swarm Water Search Algorithm, *International Journal of Swarm Intelligence Research*, vol. 11(2), pp. 19–30, 2020. doi: 10.4018/IJSIR.2020040102.
- [18] Khanteymoori A., Olyae M.H., Abbaszadeh O., Valian M.: A novel method for Bayesian networks structure learning based on Breeding Swarm algorithm, *Soft Computing*, pp. 3049–3060, 2018.
- [19] Larranaga P., Poza M., Yurramendi Y., Murga R., Kuijpers C.: Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18(9), pp. 912–926, 1996. doi: 10.1109/34.537345.
- [20] Li J., Chen J.: A Hybrid Optimization Algorithm for Bayesian Network Structure Learning Based on Database, *Journal of Computers*, vol. 9(12), pp. 2787–2791, 2014.
- [21] Li S., Wang B.: A Method for Hybrid Bayesian Network Structure Learning from Massive Data Using MapReduce. In: *2017 IEEE 3rd International Conference on Big Data Security on Cloud (Bigdatasecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, pp. 272–276, 2017. doi: 10.1109/BigDataSecurity.2017.42.
- [22] Margaritis D.: *Learning Bayesian Network Model Structure from Data*, Ph.D. thesis, Carnegie Mellon University, School of Computer Science, 2003. <https://www.cs.cmu.edu/~dmarg/Papers/PhD-Thesis-Margaritis.pdf>.
- [23] Mirjalili S., Mirjalili S.M., Lewis A.: Grey Wolf Optimizer, *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [24] Orphanou K., Thierens D., Bosman P.A.N.: Learning Bayesian Network Structures with GOMEA. In: *GECCO '18: Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1007–1014, 2018.
- [25] Rahier T., Marié S., Girard S., Forbes F.: Fast Bayesian Network Structure Learning using Quasi-Determinism Screening, 2019. <https://hal.archives-ouvertes.fr/hal-01691217>. Working paper or preprint.
- [26] Sadeghi H.A.: Structure Learning of Bayesian Belief Networks Using Simulated Annealing Algorithm, *Middle-East Journal of Scientific Research*, vol. 18(9), pp. 1343–1348, 2013. doi: 10.5829/idosi.mejsr.2013.18.9.12375.

- [27] Salama K.M., Freitas A.A.: ABC-Miner: An Ant-Based Bayesian Classification Algorithm. In: M. Dorigo, *et al.* (Eds.), *Swarm Intelligence. ANTS 2012, Lecture Notes in Computer Science*, vol. 7461, Springer, 2012.
- [28] Sencer S., Oztemel E., Torkul O., Kubat C., Taskin H., Yildiz G.: Bayesian Structural Learning with Minimum Spanning Tree Algorithm. In: *The World Congress in Computer Science, Computer Engineering and Applied Computing*, 2013.
- [29] Tucker V.A.: Gliding flight: speed and acceleration of ideal falcons during diving and pull out, *Journal of Experimental Biology*, vol. 201(3), p. 403–414, 1998. doi: 10.1242/jeb.201.3.403.
- [30] Tucker V.A.: Gliding flight: drag and torque of a hawk and a falcon with straight and turned heads, and a lower value for the parasite drag coefficient, *Journal of Experimental Biology*, vol. 203(24), p. 3733–3744, 2000. doi: 10.1242/jeb.203.24.3733.
- [31] de Vasconcelos Segundo E.H., Mariani V.C., dos Santos Coelho L.: Design of heat exchangers using Falcon Optimization Algorithm, *Applied Thermal Engineering*, vol. 156, pp. 119–144, 2019. doi: 10.1016/j.applthermaleng.2019.04.038.
- [32] Wang J., Liu S.: Novel binary encoding water cycle algorithm for solving Bayesian network structures learning problem, *Knowledge-Based Systems*, vol. 150(C), pp. 95–110, 2018.
- [33] Yang C., Ji J., Liu J., Liu J., Yin B.: Structural learning of Bayesian networks by bacterial foraging optimization, *International Journal of Approximate Reasoning*, vol. 69, pp. 147–167, 2016.
- [34] Yuan C., Malonean B., Wu X.: Learning Optimal Bayesian Networks Using A* Search. In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 2186–2191, 2011.
- [35] Zhang S.Z., Liu L.: MCMC samples selecting for online bayesian network structure learning. In: *2008 International Conference on Machine Learning and Cybernetics*, vol. 3, pp. 1762–1767, IEEE, 2008. doi: 10.1109/ICMLC.2008.4620690.

Affiliations

Shahab Wahhab Kareem

(1) Erbil Polytechnic University, Department of Technical information Systems Engineering, Erbil Technical Engineering College, Erbil-Iraq, shahab.kareem@epu.edu.iq

(2) Lebanese French University, Department of Information Technology, College of Engineering and Computer Science, Erbil-Iraq

Mehmet Cudi Okur

Yasar University, Software Engineering Department, Faculty of Engineering, Izmir-Turkey, mehmet.okur@yasar.edu.tr

Received: 27.04.2020

Revised: 22.06.2020

Accepted: 05.08.2020