



YAŞAR UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PHD THESIS

**NOVEL SWARM INTELLIGENCE ALGORITHMS FOR
STRUCTURE LEARNING OF BAYESIAN NETWORKS
AND A COMPARATIVE EVALUATION**

SHAHAB WAHHAB KAREEM

THESIS ADVISOR: PROF. DR. MEHMET CUDI OKUR

PHD WITH THESIS IN ENGLISH

PRESENTATION DATE: 07.01.2020

BORNOVA / İZMİR
JAN 2020

We certify that, as the jury, we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of ~~Master of Science~~ / the Doctor of Philosophy.

Jury Members:

Signature:

Prof. Dr. **Mehmet Cudi Okur**
Yaşar University



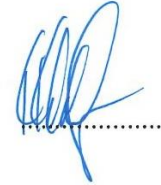
Prof. Dr. **Serdar Korukoğlu**
EGE University



Asst.Prof. Dr. **Mete Eminağaoğlu**
Dokuz Eylül University

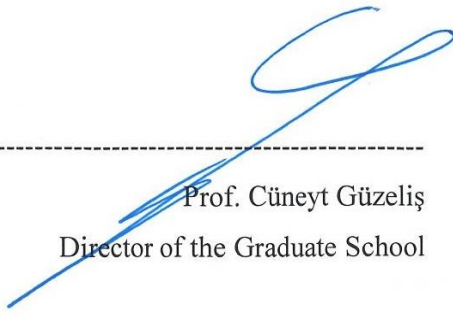


Asst.Prof.Dr. **Korhan Karabulut**
Yaşar University



Asst.Prof. **Dr.İbrahim Zincir**
Yaşar University




Prof. Cüneyt Güzeliş
Director of the Graduate School

ABSTRACT

NOVEL SWARM INTELLIGENCE ALGORITHMS FOR STRUCTURE LEARNING OF BAYESIAN NETWORKS AND A COMPARATIVE EVALUATION

Kareem, Shahab Wahhab

Ph.D, Computer Engineering

Advisor: Prof.Dr.Mehmet Cudi Okur

January 2020

Bayesian networks are useful analytical models for designing the structure of knowledge in machine learning which can represent probabilistic dependency relationships among the variables. A Bayesian network depends on; 1.the parameters of the network and 2.the structure. Parameters represent conditional probabilities while the structure represents dependencies between the random variables. The structure of a Bayesian network is a directed acyclic graph (DAG). Learning the structure of a Bayesian network is NP-hard but still extensive work have been done to optimize approximate solutions. In this thesis, we have conducted research for structure learning to develop algorithms to find a solution to the problem. There are two approaches for learning the structure of Bayesian networks. The first is a constraint-based approach, and the second is a score and a search approach. One common type of method for Bayesian network structure learning is the score-based search. Score-based methods rely on a function to test how well the network model matches the data, and they search for a structure that produces high scores on this function. There are two types of scoring functions: Bayesian score and information-theoretic score. The Bayesian and information-theoretic scores have been implemented in several structure learning methods. In this thesis, we focused on the score based search for testing the structure learning of Bayesian network using heuristic methods for searching and BDeu as a score function. In this thesis we proposed five algorithms for the search part and used BDeu as a score function. We also proposed a sixth method which is also a nature inspired one. The first proposed algorithm used Pigeon Inspired Optimization as a search method and the above mentioned score function. The proposed method has shown a good result when compared with default methods like Simulated Annealing

and greedy search. This algorithm is a novel approach applied for structure learning of Bayesian network. The second proposed algorithm used Bee optimization and Simulated Annealing as a hybrid algorithm, which used Bee optimization as a local search and Simulated Annealing as a global search. The third proposed algorithm also used bee optimization and Simulated Annealing as a hybrid but used Bee optimization as a global search and Simulated Annealing as a local search. The fourth proposed algorithm used Bee optimization and Greedy search as a hybrid algorithm. It used Bee optimization as local search and Greedy as global search. The fifth algorithms also used bee optimization and Greedy as a hybrid algorithm, but it used Bee optimization as a global search and Greedy as a local search. Our last proposed algorithm used Elephant Swarm Water Search Algorithm (ESWSA). The thesis presents the results of extensive evaluations of these algorithms based on common benchmark data sets. Applications of ESWSA in Structure learning of Bayesian Network and comparisons with the Simulated Annealing and Greedy Search, show that this proposed method is better than the default Simulated Annealing and Greedy search methods.

Keywords: Bayesian network, structure learning, Pigeon Inspired Optimization, Bee Optimization, greedy, Simulated Annealing, elephant swarm search, water search, global search, local search, search and score.

ÖZ

BAYES AĞ YAPILARININ ÖĞRENİLMESİ İÇİN YENİ SÜRÜ ZEKASI ALGORİTMALARI VE KARŞILAŞTIRMALI BİR DEĞERLENDİRME

KAREEM, SHAHAB WAHHAB

Doktora Tezi, Bilgisayar Mühendisliği

Danışman: Prof.Dr.Mehmet Cudi Okur

Ocak 2020

Bayes ağları, makina öğrenmesinde değişkenler arasındaki rassal ilişkileri temsil eden bilgi yapısının tasarımında kullanılan yararlı analitik modellerdir. Genel olarak Bayes ağı Ağın Parametreleri ve Ağın yapısına bağlıdır. Parametreler şartlı olasılıkları, yapı ise şans değişkenleri arasındaki bağımlılıkları temsil eder. Bir Bayes ağının yapısı yönlü çevrimsel olmayan bir çizgedir. Bayes ağının yapısını öğrenmek bir NP-zor problem olmasına rağmen, yaklaşık çözümlerin eniyilenmesi için çok sayıda geniş kapsamlı çalışmalar yapılmıştır. Bu tezde yapı öğrenme problemine çözüm bulmayı amaçlayan algoritmalar geliştirmek için araştırmalar yürütülmüştür. Bayes ağların yapısını öğrenmek için iki yaklaşım vardır. Birinci yaklaşım kısıtlamalı diğeri ise skor ve arama temellidir. Skor temelli yaklaşımlar genel yaklaşımlardır. Bu yaklaşımlar ağ modelinin verilere nasıl uyum gösterdiğini ölçen bir fonksiyonu esas alırlar ve bu fonksiyonun değerini daha iyileştirecek yapıyı üretmeye çalışırlar. İki tür skor fonksiyonu vardır : Bayesçi skor ve bilgi teorisi skoru. Her iki skor da yapı öğrenme yöntemlerinde uygulanmıştır. Bu tezde Bayes ağın yapısını öğrenmede skor temelli arama için sezgisel yöntemler kullanılmış ve skor fonksiyonu olarak BDeu metriği kullanılmıştır. Bu amaçla, BDeu yu kullanan altı algoritma önermiştir. Önerilen ilk algoritma güvercinlerin yön bulmasından esinlenen eniyileme algoritmasıdır ve BDeu skorunu kullanmaktadır. Önerilen yöntemin yaygın kullanılan yöntemlerden daha iyi sonuçlar verdiği görülmüştür. Bu algoritma bu alanda ilk defa kullanılmaktadır. İkinci önerilen algoritma arı eniyilemesi algoritmasına ve benzetilmiş tavlama algoritmasına dayanmakta ve ilkini global ikincisini de yerel arama için kullanmaktadır. Üçüncü önerilen yöntem gene önceki ikisini esas almakta fakat bu defa arı eniyilemesi global,

benzetilmiş tavlama algoritması yerel arama için kullanılmıştır. Dördüncü önerilen yöntemde melez bir yöntem olup arı eniyilemesi ve açgözlü amayı esas almakta ve arı eniyilemesini yerel ve açgözlüyü de global arama için kullanılmaktadır. Beşinci yöntem de melezdir ve arı eniyilemesini global, açgözlü yöntemi yerel arama için kullanılmaktadır. Son önerimiz Fil sürülerinin su kaynağı arama algoritmasına dayanmaktadır. Tezde genel kıyaslama veri setleri kullanılarak BDeu metriği ve karışıklık matrislerine dayanan değerlendirmeler tartışılmış, sonuçta güvercin yön bulma ve fil sürüleri su arama yöntemlerine dayanan algoritmaların diğerlerinden daha başarılı olduğu gösterilmiştir.

Anahtar sözcükler: Bayes ağı, yapı öğrenme, Güvercinden Esinlenen Algoritma, Arı Eniyilemesi, açgözlü, Benzetilmiş Tavlama, Fil sürü araması, su araması, global arama, yerel arama, arama ve skor.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor PROF. DR. MEHMET CUDİ OKUR for his guidance and patience during this study.

I would like to express my enduring love to my Wife, Childs, Parents, Sisters and brother who are always supportive, loving and caring to me in every possible way in my life.

Shahab Wahhab Kareem

İzmir, 2020

TEXT OF OATH

I declare and honestly confirm that my study, titled “Novel Swarm Intelligence Algorithms for Structure Learning of Bayesian Networks and a Comparative Evaluation” and presented as a PhD Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Shahab Wahhab Kareem

Signature

.....

January 9, 2020

TABLE OF CONTENTS

ABSTRACT	i
ÖZ	iii
ACKNOWLEDGEMENTS	vi
TEXT OF OATH	vii
TABLE OF CONTENTS	ix
LIST OF FIGURES	xii
LIST OF TABLES	xiv
ABBREVIATIONS	xvii
CHAPTER 1 INTRODUCTION	1
1.1. MOTIVATION.....	1
1.2. THESIS GOALS AND OVERVIEW.....	3
1.3. THESIS ORGANIZATION.....	4
CHAPTER 2 BAYESIAN NETWORK	5
2.1 STATISTICAL MODELLING.....	5
2.1.1 PROBABILITY	6
2.1.2 CONDITIONAL PROBABILITY	6
2.1.3 BAYES RULE	7
2.1.4 INDEPENDENCE	7
2.2 GRAPH THEORY AND BAYESIAN NETWORKS.....	8
2.2.1 GRAPHS, NODES, AND ARCS.....	8
2.2.2 THE STRUCTURE OF A GRAPH.....	9
2.3 PROBABILISTIC GRAPHICAL MODELS.....	10
2.3.1 MARKOV NETWORKS.....	12
2.3.2 BAYESIAN NETWORKS.....	12
2.3.3 SOME PRINCIPLES OF BAYESIAN NETWORK	13
2.3.3.1 D-SEPARATION	13
2.3.3.2 MARKOV EQUIVALENT CLASS	14
2.3.4 QUERYING A DISTRIBUTION.....	15
2.3.4.1 EXACT INFERENCE.....	16
2.3.4.2 APPROXIMATE INFERENCE.....	16
2.4 BAYESIAN NETWORK LEARNING.....	17

2.4.1 LEARNING THE STRUCTURE OF BAYESIAN NETWORKS	20
2.4.1.1 THE SCHEMA FOR LEARNING STRUCTURE.....	22
2.4.1.2 PROCEDURE FOR LEARNING STRUCTURE.....	23
2.4.1.3 THE COMPLEXITY OF STRUCTURE LEARNING.....	25
2.4.1.4 CONSTRAINT BASED METHODS.....	25
2.4.1.5 SCORE-AND-SEARCH BASED METHODS.....	28
2.4.1.6 HYBRID METHOD	36
2.5. EVALUATING STRUCTURAL ACCURACY.....	38
2.5.1 EVALUATION METRICS.....	38
2.5.2 CONFUSION MATRIX.....	39
2.5.2.1 ACCURACY ANT ERROR RATE.....	40
2.5.2.2 SENSITIVITY AND SPECIFICITY.....	40
2.5.2.3 PRECISION, RECALL AND F-SCORE.....	42
CHAPTER 3 PROPOSED ALGORITHMS	44
3.1. PIGEON INSPIRED OPTIMIZATION	44
3.1.1 OVERVIEW OF PIGEON INSPIRED OPTIMIZATION.....	45
3.1.2 MATHEMATICAL MODEL OF PIO.....	46
3.2. SIMULATED ANNEALING.....	49
3.2.1 INTRODUCTION OF SIMULATED ANNEALING.....	49
3.2.2 SIMULATED ANNEALING ALGORITHM.....	50
3.2.3 IMPLEMENTATION OF THE S.A. ALGORITHM.....	52
3.3. GREEDY ALGORITHMS	53
3.3.1 ELEMENTS OF THE GREEDY STRATEGY.....	54
3.3.2 OPTIMAL SUBSTRUCTURE.	56
3.4. BEE ALGORITHMS	56
3.4.1 BEES IN NATURE.....	57
3.4.2 ARTIFICIAL BEES.....	58
3.4.3. BEE ALGORITHM.....	59
3.5. NATURE INSPIRED ELEPHANT SWARM WATER SEARCH ALGORITHM.....	61
3.5.1 ELEPLANT IN NATURE.....	61
3.5.2 SOCIAL BEHAVIOR AND INTERACTION IN ELEPHANTS.....	63
3.5.3.ELEPHANT SWARM WATER SEARCH ALGORITHM	64
3.6. METHODOLOGY.....	67
3.6.1. FIRST PROPOSED METHOD.....	67
3.6.2. SECOND PROPOSED METHOD.....	71

3.6.3. THIRD PROPOSED METHOD.....	75
3.6.4. FOURTH PROPOSED METHOD.....	78
3.6.5. FIFTH PROPOSED METHOD.....	80
3.6.6. SIXTH PROPOSED METHOD.....	83
CHAPTER 4 DATASETS AND EXPERIMENTS	87
4.1. DATASETS	87
4.2. EXPERIMENTAL RESULT OF SCORE FUNCTION	89
4.3. EXPERIMENTAL RESULT OF CONFUSION MATRIX.....	98
CHAPTER 5 CONCLUSIONS AND FUTURE RESEARCH	116
REFERENCES	122



LIST OF FIGURES

Figure 2.1. Directed, undirected, and partially directed	8
Figure 2.2. Parents, neighbors, ancestors, children, and descendants, of a node within a directed graph	10
Figure 2.3. Conditional independence: (a) DAG as an example. (b) Markov random field (MRF) as an example.....	11
Figure 2.4. The Model of Bayesian network; A DAG among parameters and nodes describing the probability distribution.....	13
Figure 2.5. Head-to-tail or Serial relation.....	13
Figure 2.6. Head-to-head or Converging relation.....	14
Figure 2.7. Tail-to-tail or Diverging connection.....	14
Figure 2.8. An example of Markov equivalent class for three variables, , A, B, C, and four DAGs. The (a), (b) and (c) DAGs have a similar independence structure while (d) compares to a different set of independencies.....	15
Figure 3.1 Map and compass operator model of PIO	47
Figure 3.2 Landmark operator model	47
Figure 3.3 Flowchart of the Simulated Annealing algorithm	49
Figure 3. 4 Pseudo-code of a greedy algorithm for a minimization problem	54
Figure 3. 5 Pseudo code of the basic bees algorithm	59
Figure 3. 6 The Bees Algorithm Flowchart.....	60
Figure 3. 7 Group exhibitions in Elephants clan	62
Figure 3. 8 Pseudo Code of The PIO for Structure Learning Bayesian Network	69
Figure 3. 9 Map and compass steps for one Pigeon	70
Figure 3. 10 Pseudo code of BSA hybrid bee local and SA is global search.....	74
Figure 3. 11 Pseudo code SAB (Bee global search and SA is local search)	76
Figure 3. 12 Pseudo code BLGG (Bee local search and Greedy is global search).	79

Figure 3. 13	The construction process of a Bayesian Network	80
Figure 3. 14	Pseudo code BGGL (Bee global search and Greedy is local search)	82
Figure 3. 15	The construction process of a BN.....	83
Figure 3. 16	ESWSA Algorithm for Structure learning Bayesian Network	85
Figure 3. 17	Water searching steps for one Elephant	86
Figure 4.1	Sensitivity of PIO and Simulated Annealing and Greedy	99
Figure 4.2	Accuracy of PIO and Simulated Annealing and Greedy	100
Figure 4.3	F1_Score of PIO and Simulated Annealing and Greedy	100
Figure 4.4	AHD of PIO and Simulated Annealing and Greedy	101
Figure 4.5	PPV for BSA, SAB, and Simulated Annealing	102
Figure 4.6	Sensitivity for BSA, SAB, and Simulated Annealing	103
Figure 4.7	Accuracy for BSA, SAB, and Simulated Annealing	103
Figure 4.8	F1_Score for BSA, SAB, and Simulated Annealing	104
Figure 4.9	AHD for BSA, SAB, and Simulated Annealing.....	104
Figure 4.10	PPV for BLGG, BGGL and Greedy	106
Figure 4.11	Sensitivity for BLGG, BGGL and Greedy	106
Figure 4.12	Accuracy for BLGG, BGGL and Greedy	107
Figure 4.13	F1_Score for BLGG, BGGL and Greedy	107
Figure 4.14	AHD for BLGG, BGGL and Greedy.....	108
Figure 4.15	Sensitivity for ESWSA, Simulated Annealing, and Greedy.....	110
Figure 4.16	Accuracy for ESWSA, Simulated Annealing, and Greedy	110
Figure 4.17	F1 Score for ESWSA, Simulated Annealing, and Greedy	111
Figure 4.18	AHD for ESWSA, Simulated Annealing, and Greedy	112
Figure 4.19	Sensitivity for Proposed Methods.....	114
Figure 4.20	Accuracy for proposed Methods.....	114
Figure 4.21	F1 Score for proposed Methods.....	115
Figure 4.22	AHD for Proposed Methods	115

LIST OF TABLES

Table 2.1. The table presents the amount of various DAGs that can produce several nodes. For example, there are $1.4 \cdot 10^{41}$ different DAGs with 14 nodes	21
Table 2.2. Confusion Matrix	39
Table 2.3. Test result in Confusion matrix	41
Table 2.4. Sensitivity and Specificity in Confusion matrix	42
Table 3.1. Designing the S.A. Algorithm	52
Table 4.1. Calculation results of the best of BDeu Score function for PIO with Simulated Annealing and Greedy in 2 minutes Execution time	90
Table 4.2. Calculation results of the best of BDeu Score function for PIO with Simulate Annealing and Greedy in 5 minutes Execution time	90
Table 4.3. Calculation results of the best of BDeu Score function for PIO with Simulate Annealing and Greedy in 60 minutes Execution time	90
Table 4.4. Calculation results of the best of BDeu Score function for BSA and SAB with Simulate Annealing in 2 minutes Execution time	91
Table 4.5. Calculation results of the best of BDeu Score function for BSA and SAB with Simulate Annealing in 5 minutes Execution time	91
Table 4.6. Calculation results of the best of BDeu Score function for BSA and SAB with Simulate Annealing in 60 minutes Execution time	92
Table 4.7. Calculation results of the best of BDeu Score function for BLGG and BLGG with default Greedy in 2 minutes Execution time	92
Table 4.8. Calculation results of the best of BDeu Score function for BLGG and BLGG with default Greedy in 5 minutes Execution time	93
Table 4.9. Calculation results of the best of BDeu Score function for BLGG and BLGG with default Greedy in 60 minutes Execution time	93
Table 4.10. Calculation results of the best of BDeu Score function for ESWSA, Simulated Annealing, and Greedy in 2 minutes Execution time.....	94

Table 4.11. Calculation results of the best of BDeu Score function for ESWSA, Simulated Annealing, and Greedy in 5 minutes Execution time.....	95
Table 4.12. Calculation results of the best of BDeu Score function for ESWSA, Simulated Annealing, and Greedy in 60 minutes Execution time.....	95
Table 4.13. Calculation results of the best of BDeu Score function for all proposed methos when time is 2M.....	96
Table 4.14. Calculation results of the best of BDeu Score function for all proposed methos when time is 5M.....	97
Table 4.15. Calculation results of the best of BDeu Score function for all proposed methos when time is 60 M.....	97
Table 4.16. Confusion matrix of PIO, Simulated Annealing and Greedy.....	99
Table 4.17. Confusion matrix of BSA, SAB, and Simulated Annealing.	102
Table 4.18. Confusion matrix of BLGG, BGGL, and Greedy.	105
Table 4.19. Confusion matrix of ESWSA, Simulated Annealing, and Greedy.....	109
Table 4.20. Confusion matrix of All proposed methods.....	113

ABBREVIATIONS

ABC	Artificial Bee Colony Optimization
ACO	Ant Colony Optimization
AIC	Akaike information criterion
BA	Bee Algorithm
BD	Bayesian Dirichlet
BDe	Bayesian Dirichlet (“e” for likelihood-equivalence)
BDeu	Bayesian Dirichlet equivalent uniform (“u” for uniform joint distribution)
BFO	Bacterial Foraging Optimization
BGeu	Bayesian Gaussian equivalent uniform
BGGL	Greedy as local search and Bee as global search
BIC	Bayesian Information Criterion
BLGG	Bees as local search and Greedy as global search
BN	Bayesian Network
BP	Backpropagation
BSA	Bees as local search and Simulated Annealing as global search
CF	Cell Formation
CM	Cellular manufacturing
CPT	Conditional Probability Table
DAG	Directed Acyclic Graph.
EF	Employed forager
ER	Recruit unconvertable
ESWSA	Elephant Swarm Water Search Algorithm
ES	Scout begins for the exploration
FN	False Negative
FP	False Positive
GS	Greedy Search
IAMB	Incremental Association Markov blanket algorithm
IC	Inductive Causation
KDD	Knowledge from Data Discovery

LL	Log-likelihood
MAP	Maximum a Posteriori
MAP	Maximum a Posteriori Probability
MB	Markov Blanket
MDL	Minimum Description Length
MIT	Mutual Information Tests
MLP	Multi-layered Perceptron
MMHC	Original Max-Min Hill-Climbing algorithm
MMPC	Max-Min Parent Children
MRF	Markov Random Fields
NML	Normalized Minimum Likelihood
P	Precision
P(V)	Probability Distribution
PC	Parent-Children
PCB	Printed Circuit Board
PDF	Probability Density Function
PGM	Probabilistic graphical models
PIO	Pigeon Inspired Optimization
PMF	Probability Mass Function
PSO	Particle Swarm Optimization
R	Recall
R	Recruit
RCE	Randomized Control Experiment
RF	Reactivated forager
S	Scout
SA	Simulated Annealing
SAB	Simulated Annealing as local search and Bee as global search
SC	Sparse Candidate
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

TP	True Positive
$P(x)$	The probability of event X
$X \cap Y$	The crossing between X and Y
$P(X Y)$	The probability of the event X, given that Y has happened
$u \rightarrow v$	Directed of u to v
θ	The probability distributions of any random variable
Π_{x_i}	Parent of x_i
Γ	Gamma Function



CHAPTER 1

INTRODUCTION

1.1. Motivation

Machine Learning involves techniques for computers programming to learn. Machines used to achieve a universal variety concerning responsibilities, including the development of needed software for most computational tasks. Machine learning approaches deal with several of the related study topics such as the domains of data mining, artificial intelligence and statistics. Data mining explores models within some data which is recognizable by people. Statistics concentrates on explaining the events that are present in experimental or observational data sets [1] [2]. Majority of the researches use data mining methods to train observed data and to extract intelligence rules. With specific rules, it obtains a probabilistic graphics model, statistical models, Bayesian statistics, and machine learning. Graphics models combine probability and graph theory. It's present a simplistic mechanism as dealing including difficulties that arise while coupling engineering and mathematics to reduce ambiguity and complexity. As such, they play a significant role in machine learning algorithms during steps design and analysis. The theory of probability presents methods to analyze how the components joined, guaranteeing that the system remains consistent. The combined results expected to be compatible and present new techniques to propose new interface models for observed data. Some graph-theoretic view of graphical models presents an attractive interface jointly for users that ability to create reactive collections about variables and a data structure that can be used in powerful public-objective algorithms[2]. One of the most important types for probabilistic graphical models is the Bayesian Network [3, 4]. They commonly used in the field of Knowledge from Data Discovery (KDD). A Bayesian network is a directed acyclic graph whose nodes (vertices) describe links and variables (or controlled arcs) show the statistical relationship among variables and a probability distribution defined across those variables. An essential difficulty of the modern study is Bayesian network learning from observed data. The development of principles can be performed both by utilising observed data or expertise. Several kinds of research have been conducted on this subject, deriving on various approaches: Techniques to the development regarding independence structure within data to rebuild for optimizing an actual function of the

graph, namely a score. Optimization techniques aim to the development of a representation of the local structure based on a destination variable to rebuild the network of the global structure. Most researches have restricted their work to static cases for learning the structure of Bayesian networks. The majority of these algorithms use a traditional approach which depended on scores.

In the rest of this thesis, we consider the combination of strategies, namely, global optimization and local search relating the static case. We remarked that the structure learning Bayesian network is a well-researched field. To our knowledge, the researcher about BN structure learning applies the benchmarks to evaluate the procedures. The difficulty in the Bayesian network structure learning instance is the search for discovering the excellent structure. But, this depends on the score and search method, which is computationally NP-hard. Furthermore, causal models can offer enough extra benefits for researchers. It can assist us in experiencing our situation and identifying “laws” of the environment in the Sciences: Chemistry, Biology, Physics, even Genetics. Growing developments under a related thread now can and make for example, scientists to limit the options of the analysis for infections. In that space, the building of patterns automatic or the semi-automatic can be valuable. In the dissertation, we preferred to concentrate on covering structure learning of Bayesian network depending on the score and search method. Different models can describe possible domains—as an example, artificial neural networks, decision trees, Markov networks, blend of essential roles, etc. The researcher in Bayesian Network describes and learn directed causal connections, which is also our final purpose. We attempt to explain the combinatorial difficulty of getting the most significant scoring from data in the Bayesian network structure. This can be view as the challenge of structured learning which is as an inference difficulty. The major combinatorial problem drives from the global constraint that the structure of the graph has to be acyclic. The problem of the structure learning may be called as a linear program covering the polytope described by logical acyclic structures. To decrease the mentioned difficulties through applying a restricted external approach to the polytope which stretch it through exploring the validity constraints. In Case of finding the full solution, it has proven to be the optimal solution of the Bayesian network. Alternative approaches are; Pigeon Inspired Optimization, simulated annealing, the greedy method and Elephant Swarm Water Search algorithms.

1.2. THESIS GOALS AND OVERVIEW

A Bayesian network (BN) involves common useful theoretical principles to describe the possibility of learning from data in artificial intelligence. A graphical model used by Bayesian Network for representing the conditional dependency connections between arbitrary variables and those variables governed by the joint probability distribution [3]. Assume a Bayesian Network and observations for many variables are given, a probabilistic inference can then determine the fitness of the other unobserved variables. Systems accept this standard to design solutions to practical difficulties within various domains, such as biology, medical diagnosis, natural language processing, control, and forecasting [4].

Learning the structure automatically from the data, attracted researchers and several learning algorithms have become available [5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. [15, 16, 17, 18, 19, 20, 21, 22, 23, 24] [25, 26, 27, 28]. Those algorithms choose the score and search, or the dependency analysis approaches. Dependency analysis applies a statistical method for finding dependency and independence connections between variables and whereby constructs a Bayesian Network [10]. Applying a search technique in the score and search approaches to investigate Bayesian Network structures aims to find the highest score value achieved [18]. Both methods have severe disadvantages. Dependency analysis requires dealing with a massive number of cases that are difficult also unpredictable; moreover, it is challenging to guarantee the quality properties of learning. In contrast, Bayesian Network structure learning through the score and search is an NP-hard problem because of the number of variable increments [29]. Once the location of applicant networks grows high, exact search results may be unsuitable for structural learning in Bayesian Network. While some heuristic algorithms, like hill-climbing algorithms [30, 31], K2 [32], repeated local search [33, 34], can mark the difficulty of significant search areas, they grow confined within local optima.

To explore those difficulties, many stochastic algorithms have proposed for the Bayesian Network structure learning during the last years. [35]. We Can classify those algorithms within two classes [33]:

1. Swarm intelligence algorithms, which are nature-inspired optimization procedures that include bacterial foraging optimization (BFO) [5], artificial

Bee colony optimization (ABC) [7], ant colony optimization (ACO) [15, 36], particle swarm optimization (PSO) [16], artificial fish swarm algorithm [37]. They utilize a meta-heuristic search technique in search space of the Bayesian Network and use a scoring function for determining the best of the applied networks.

2. The evolutionary algorithms which represent an inspiration of evolution including common genetics and also genetic programming, genetic algorithm, which involve evolutionary programming, and evolution strategy. Genetic algorithm [38] and evolutionary programming [31] are standard techniques which are useful approaches for Bayesian Network structure learning from data.

1.3. THESIS ORGANIZATION

This study concentrates on the structure learning of Bayesian network. First chapter includes a literature review of structure learning of Bayesian networks based on score and the search approach. The basic principles of Bayesian networks and structure learning of Bayesian networks explained in Chapter 2. The Pigeon Inspired optimization, Elephant Swarm Water Search Algorithm, Bees algorithm, Simulated Annealing, Greedy Search and proposed algorithms for structure learning of Bayesian networks explained in Chapter 3. Chapter 4 concentrates on the results obtained from the implementations of various algorithms that we proposed. Conclusions and recommendations for future studies presented in Chapter 5.

CHAPTER 2

BAYESIAN NETWORK

Knowledge description and thinking of these descriptions have caused the development of several models. Bayesian networks, and Probabilistic graphical models, have been established to be valuable instruments as a description of ambiguous knowledge. Then, many researchers such as [39, 40, 41, 42] proposed a Bayesian probabilistic reasoning formalism for knowledge extraction from incomplete information.

Learning a Bayesian network is composed of two states: parameter learning and structure learning. In this thesis, our focus is on structure learning of Bayesian networks. There are three kinds of techniques in structure learning: techniques depending on a description of conditional independence, techniques depending on optimization like score also hybrid approaches.

To illustrate the advantages of structure learning algorithms, these learning algorithms should be tested using the achieved properties of the corresponding Bayesian networks. Some algorithms use several evaluation metrics in search and identify the network through an application of the score-based methods. Some others concentrate on the application of a source form. In our thesis, we concentrated on specific evaluation procedures utilising a score-based method.

This chapter reviews fundamental descriptions and representations of traditional Bayesian networks, probability and conditional independence.

2.1 STATISTICAL MODELLING

Usually, statistical modelling strategies utilised within several systems to describe complicated multi-parametric structures. The probabilistic form shows an ontological framework; also, it represents the relationships with the model's fundamental entities. Unlike deterministic models, where the links are explained by mathematical equations (either science-based or derived), in statistical models the connections among variables are probabilistic. In the subsequent sections, we present the principle of probability, conditional and marginal probability distributions and their use in the graphical models that underpin the Bayesian structure learning methods.

2.1.1 PROBABILITY

A traditional frequency-based explanation of probability is the following: Probability of a disjoint event is the occurrence frequency of this event compared to the cumulative amount of times the events can happen. Suppose, for instance, the analysis conducted several times, and each time a result is one of three events A, B or C. If the number of their occurrences are n_A, n_B, n_C the probability of event A is then presented by the following Equation [43].

$$P(X) = \frac{n_A}{(n_A + n_B + n_C)} \quad \text{Equation 2-1}$$

Bayesian or most frequent likelihood test based on the three necessary assumptions of probability analysis [44]. First, a probability cannot be larger than one and smaller than zero (Equation 2-2). If that is one, the event will occur; zero means the event will never happen.

$$0 \leq P(X) \leq 1 \quad \text{Equation 2-2}$$

In a unit space S, comprising a measurable number of fundamental events there is a total likelihood that one of the fundamental events will happen

$$P(S) = 1 \quad \text{Equation 2-3}$$

Wherever events are disjoint, the cumulative probability of one or another of the events happening can be obtained by the sum of their specific probabilities

$$P(X \cup Y) = P(X) + P(Y) \quad \text{Equation 2-4}$$

If the events can both happen, the probability of both events happening can be obtained by the Equation:

$$P(X \cup Y) = P(X) + P(Y) - P(X \cap Y) \quad \text{Equation 2-5}$$

Where $X \cap Y$ denotes the intersection between X and Y, which is the event that both X and Y happen [43].

2.1.2 CONDITIONAL PROBABILITY

Conditional probability interprets the occurrence probability of an event, given some other event has already occurred. The probability of the event X, given that Y has happened shown as $P(X|Y)$ and described by:

$$P(X|Y) = P(X \cap Y) / P(Y) \quad \text{Equation 2-6}$$

This is known as the primary rule of conditional probability. Equation 2-6 can express as

$$P(X \cap Y) = P(Y) \cdot P(X|Y) \quad \text{Equation 2-7}$$

The conditional probability definition can extend to cover more joint events as in Equation 2-8.

$$\begin{aligned} P(X \cap Y \cap Z) &= P(X|(Y \cap Z)) \cdot P(Y \cap Z) \\ &= P(X|(Y \cap Z)) \cdot P(Y|Z) \cdot P(Z) \end{aligned} \quad \text{Equation 2-8}$$

The Equation 2-8 is the chain rule and expressed for n joint events in Equation 2-9. This rule is essential for factorizations in probability analysis of Bayesian Networks.

$$P(\cap_{i=1}^n X_i) = \prod_{i=1}^n P(X_i | \cap_{i=1}^{i-1} X_i) \quad \text{Equation 2-9}$$

Bayesian probability declares that every probability is conditional upon specific situations under which determinations performed or operations executed [45].

2.1.3 BAYES RULE

Considering, from the assumptions of probability $A \cap B \equiv B \cap A$, also from the primary rule, the connection between conditional probabilities can express as in Equation 2-10.

$$P(X|Y) = \frac{P(Y|X) \cdot P(X)}{P(Y)} \quad \text{Equation 2-10}$$

The formula, identified as Bayes rule, defined posthumously in a historical form in 1763. It provides a posterior probability, $P(X|Y)$, given any extra information, Y , which is identified as the prior probability, $P(X)$.

2.1.4 INDEPENDENCE

Independence of two events implies that the occurrence of one event is not influenced by the occurrence of another event. Thus the independence of the events X and Y are expressed by Equation 2-11.

$$P(X|Y) = P(X) \quad \text{Equation 2-11}$$

Using the fundamental conditional probability rule, Equation 2-12 follows.

$$P(X \cap Y) = P(X) \cdot P(Y)$$

Equation 2-12

2.2 GRAPH THEORY AND BAYESIAN NETWORKS

2.2.1. GRAPHS, NODES, AND ARCS

A graph $G = (V, A)$ composed from a non-empty collection V of vertices or nodes also a limited (however probably empty) collection A of edges, or links. Each edge $X = (u, v)$ describes essentially a couple of neighboring nodes. The nodes are joined by an arc which represents a weight value. If in (u, v) , order is important, they represent a directed arc or edge. The arc is assumed to direct the link of u to v also generally described by an arrowhead as $(u \rightarrow v)$. It is an additional assumption that the arc moves or are outgoing from u and that it joins or is incoming for v . If (u, v) is unordered, u and v declared to be connected by an undirected edge. Undirected edges represented using a line $(u - v)$.

A graph in which every edge is directed is called a directed graph which includes ordered pairs of vertices. A graph in which every edge is undirected is named the undirected graph. Mixed graph (partially directed) contains together undirected and directed arcs.

Some instances from the mentioned types of graphs shown in Figure 2.1 within the sequence. During the undirected graph, Figure 2.1:

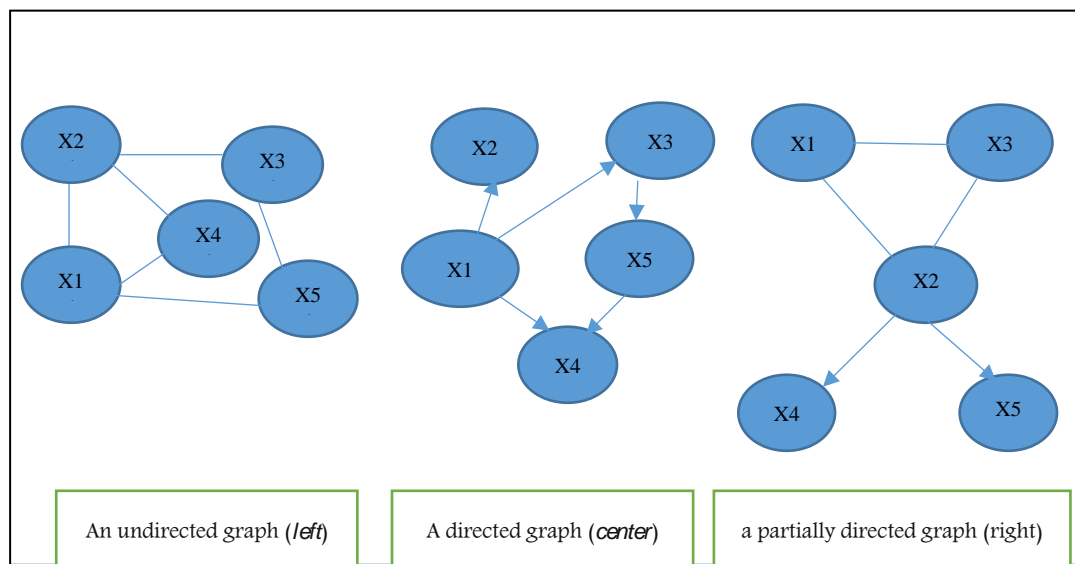


Figure 2.1. Directed, undirected, and partially directed

- The node collection is $V = \{X1, X2, X3, X4, X5\}$ also the edge (link) set is $E = \{(X1-X4), (X1-X2), (X1-X5), (X2-X4), (X3-X5), (X2-X3)\}$.
- Undirected Arcs, so, i.e., $X1-X2$ and $X2-X1$ are similar and represent the same edge.
- In a directed graph, Figure 2.1:
 - The collection of node is $V = \{X1, X2, X3, X4, X5\}$ also identified graph through a set of arc $A = \{(X1 \rightarrow X2), (X1 \rightarrow X4), (X1 \rightarrow X3), (X5 \rightarrow X4), (X3 \rightarrow X5)\}$.
 - It directs Arcs, so, i.e., $X1 \rightarrow X2$ and $X2 \rightarrow X1$ recognized as different arcs. For instance, $X1 \rightarrow X2 \in X1$ and $X2 \rightarrow X1 \notin X1$. Furthermore, it can not present of both arcs in the graph because for each couple of node one arc can be present between the nodes.
 - The mixed graph (partially directed), Figure 2.1, designated through the organization of a set of the edge $E = \{(X1-X2), (X1-X3), (X2-X3)\}$ also an arc set $A = \{(X2 \rightarrow X4), (X2 \rightarrow X5)\}$.

2.2.2. THE STRUCTURE OF A GRAPH

A structure of a graph refers to the configuration of the arcs that appear in a graph. Assumed that the nodes v and u distinguished on each arc and also there is only one arc between them.

The structure of a graph can expose impressive analytical characteristics. A common example is representing and understanding routes. Routes(paths) are a series of edges or arcs joining two nodes, described end-nodes or end-vertices. Routes are represented by a series of vertices $(V1, V2, \dots, Vn)$ that define the series of arcs. The arcs joining the vertices $(V1, V2, \dots, Vn)$ is an individual, which means a route moves over every arc just once. Within directed graphs, this is further appropriated that every arc within a route has the same direction, also the route guides from $V1$ (the end from the initial arc within the route) to Vn (the peak of the latest arc within a route). In mixed also undirected graphs (also within common while applying on a graph although of which set it refers to), arcs in a route can guide in either way or be undirected. Routes in which $V1=Vn$ describe cycles and managed with a special interest in the theory of Bayesian network. If the graph is acyclic, the directed graph structure described it as

an incomplete organization of the nodes, which means when the structure does not include loop or cycle. This organization named a topological or acyclic organization and influenced by the orientation of the arcs: if a node X_i heads X_j , means no arc of X_j to X_i . Depending on this explanation, initial nodes are origin nodes, that should include no incoming arcs; also the leaf nodes are the latest, the leaf node with no outgoing arc, while the incoming at least one arc. If there has route beginning of X_i toward X_j , X_i heads X_j in the index of the organized nodes. During this event, X_i is named the parent of X_j also X_j is called the child of X_i . If the route formed by an individual arc, by similarity v_i is a parent of X_i and X_j is a child of X_i [45].

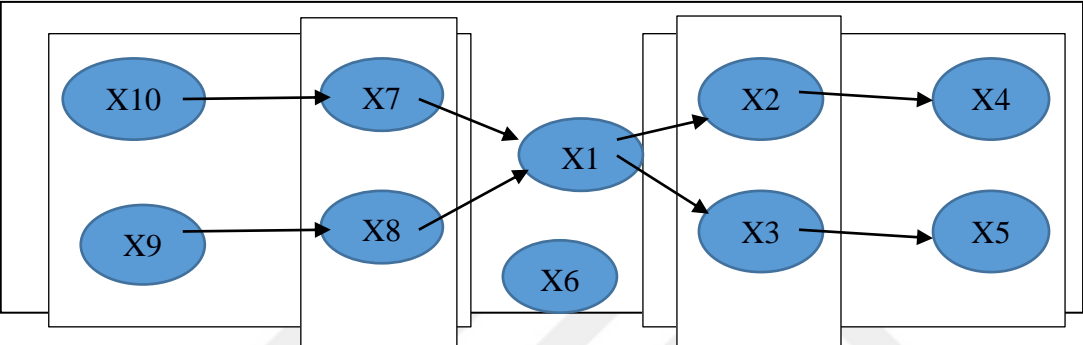


Figure. 2.2 Parents, neighbors, ancestors, children, and descendants, of a node within a directed graph

Suppose, for example, within the DAG that shown in Figure 2.2. The X_1 is a neighbourhood a combination of children with parents; the neighbouring nodes are within one of those pair sections. The nodes are just partly established; for example, they can build no organization with root (head) nodes or leaf (tail) nodes. Since an arrangement, in tradition, they describe the topological organization of a DAG ended over a collection of the unstructured set of nodes, expressed among $X_i = \{X_{i1} \dots, X_{ik}\}$, defining a partition of X .

2.3 PROBABILISTIC GRAPHICAL MODELS

The combination of graph and probability theory produced the probabilistic graphical models(PGM). It presents the mechanism for dealing with a couple of crucial difficulties: complexity plus uncertainty. The combined structure by Graphical models which represent conditional dependence structures between random variables. They play a major role in analysis and designing for machine learning algorithms.

Probabilistic graphical models are diagrams, wherever vertices express arbitrary variables; also links describe dependencies between pairs of variables. Certain forms produce a compressed description of joint probability distributions of random variables. PGMs has two essential types. First, the models of directed graphical, namely Bayesian Networks and second, the models of undirected graphical that identified as Markov Network or Random Fields (MRFs). Figure 2.3 presents those models [46].

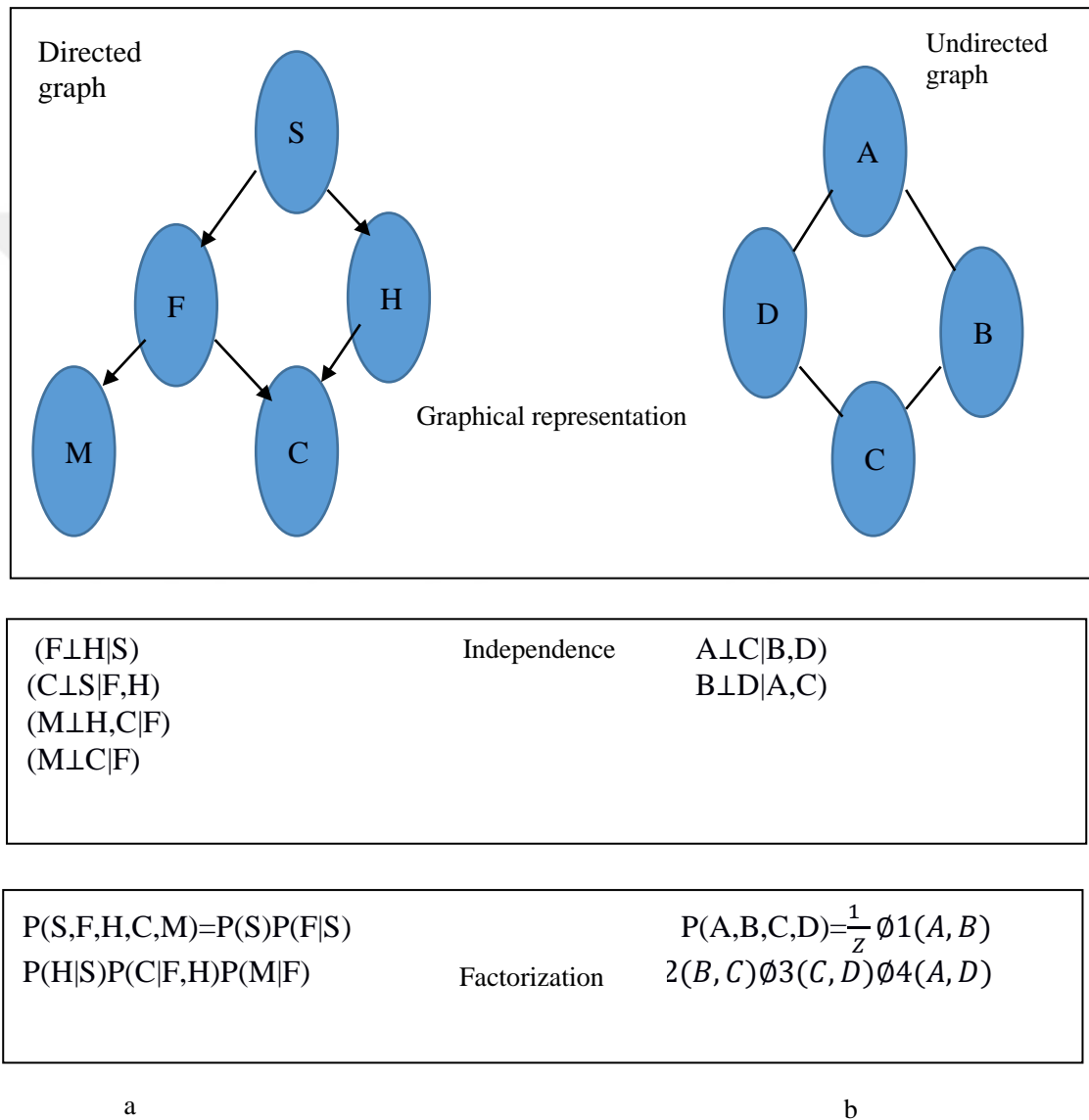


Figure 2.3 Conditional independence: (a) DAG as an example. (b) Markov random field (MRF) as an example [164].

2.3.1 MARKOV NETWORKS

The undirected graphical or Markov network model is this collection of arbitrary variables possessing the characteristic of Markov represented through undirected graphs. The model has the joint probability distribution that can express through factorization based on cliques from a graph (G) as:

$$P(X=x) = \frac{1}{Z} \prod_{C \in d(G)} \phi(C) \quad \text{Equation 2-13}$$

where denoted a normalization factor by Z, the collection of cliques of G by d(G), and the maximal potential of the clique by function $\phi(C)$ [46].

2.3.2 BAYESIAN NETWORKS

Def. 1. (Bayesian network (BN)): The Bayesian network $M = \langle G, \theta \rangle$ is a DAG $G = (V, E)$ and a collection from parameters θ . The set of vertices or nodes V is conformable to a collection of arbitrary variables $\{V_1, V_2, \dots, V_n\}$ also dependencies among these variables represented through the collection of vertices E. A parameters θ express the probability distributions from any arbitrary variable given set of parents i:

$$\theta_i = P(X_i | Pa(X_i)). \quad \text{Equation 2-14}$$

A BN is the compressed description of the joint probability distribution.

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)) \quad \text{Equation 2-15}$$

It needs to validate the Markov condition (definition 2). Fig. 2.4 gives an example of a BN representing the conditional relationships among four variables.

Def. 2. (Markov Condition): The BN of $G = (V, E)$ indicated by M if every element in V is independent of every group of non-descendant elements given its ancestors.

$$(X \perp\!\!\!\perp \text{Non Descendent}(X) | Pa(X)) \quad \text{Equation 2-16}$$

Def. 3. (Faithfulness): P is a probability distribution, and M is the Bayesian network model is dedicated from one another if each of the independence connections correct within P is the needed through the Markov theory upon M [46].

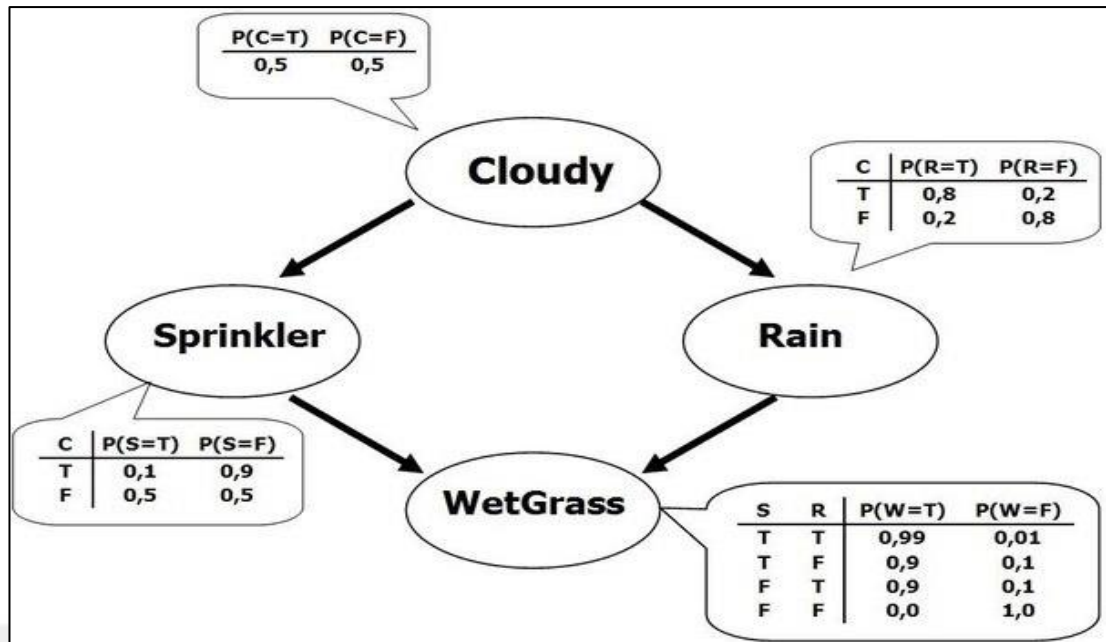


Figure 2.4: The Model of Bayesian network; A DAG among parameters and nodes describing the probability distribution.

2.3.3 SOME PRINCIPLES OF BAYESIAN NETWORKS

2.3.3.1 D-SEPARATION GISRSMMAMMAR@

To understand the flow of probabilistic control in the graph, we have to recognize how information moves from A into B to change the knowledge of C. Suppose three nodes are A, B and C, and also there is a route A—C—B. If the control flows from A to B via C so we can assume that the route A — C — B is active if it's not blocked [47]. It has three modes:

Serial relationship: If C is not detected then the route from A to B shall be active it shall be blocked. Within this situation we have, $A \perp B | C$ and $A \perp B$.

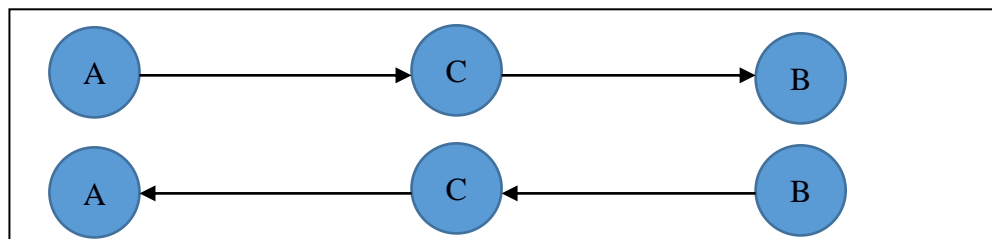


Figure 2.5: Head-to-tail or Serial relation

Converging link: If C is not detected or it should block each descendant of C we have, $A \perp B | C$ and $A \perp B$. This is also called V — structure.

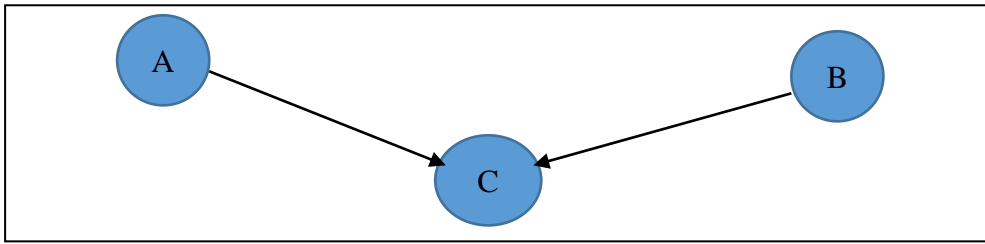


Figure 2.6: Head-to-head or Converging relation

Diverging link: If C not detected then the route on A to B will be open in the other case it would be blocked. So we have $A \perp B \mid C$ and $A \not\perp B$.

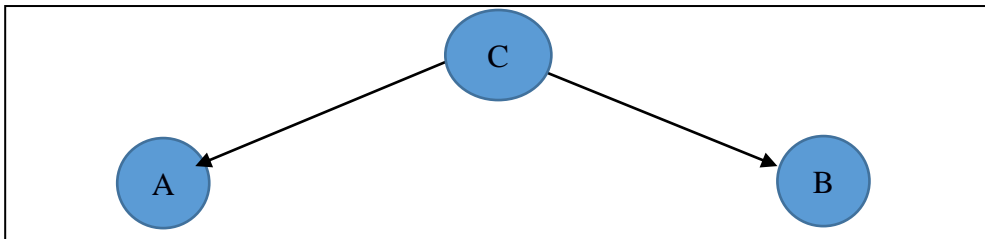


Figure 2.7: Tail-to-tail connection or Diverging

Def. 4. Directional separation (D-separation). Assume A, B are arbitrary variables also C is a collection of arbitrary variables, A plus B is d-separated through C if and only if C blocks each route of A to B [48].

2.3.3.2 MARKOV EQUIVALENT CLASS

G1 & G2 are Two DAGs said to be Markov equivalent if both provide the equivalent conditional independencies. This means that the DAGs which have equal d-separation are Markov equivalent. It means d-separation are Markov equivalent if all DAGs share the same one based on Verma and Pearl's theorem:

Theorem 1. (Verma and Pearl: [49]) A Pair of DAGs (PDAG) are similar if and only if both own the equivalent frame also v-structures (head-to-head joint).

As an instance in Figure 2.8, there are four separate DAGs by the equal number of variables, and owning equal frames. According to Theorem 1, regular Markov equivalent classes are DAGs (a), (b) and (c). However, the v-structure $A \rightarrow C \leftarrow B$ in (d), and it is just a graph under its equivalence class [50].

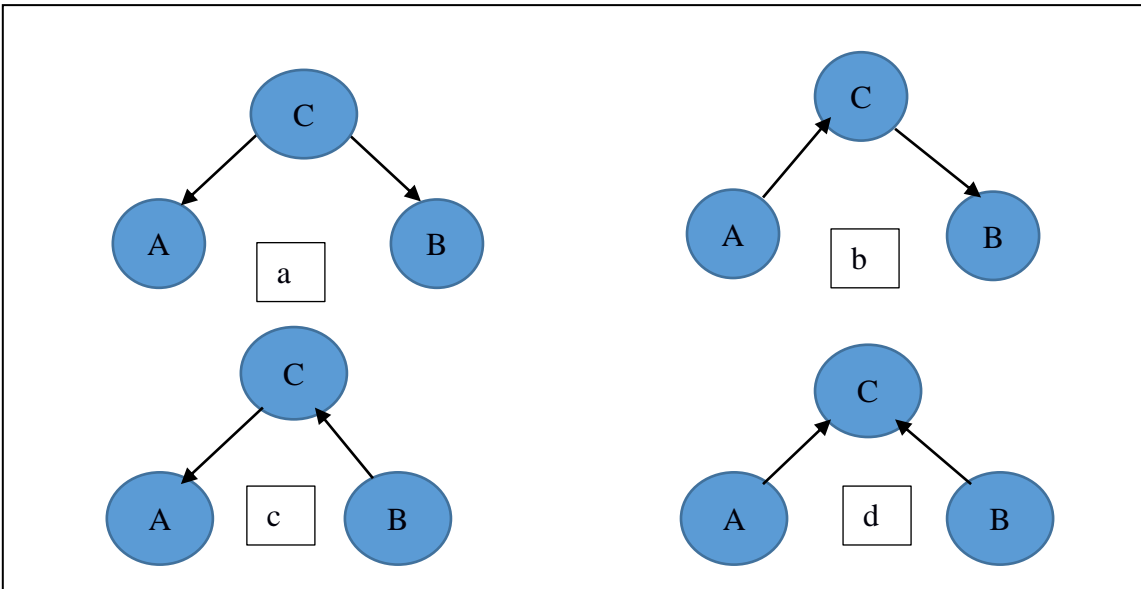


Figure 2.8: An example of Markov equivalent class, A, B, C, and four DAGs. The (a), (b) and (c) DAGs have a similar independence structure while (d) compares to a different set of independencies.

2.3.4 QUERYING A DISTRIBUTION

The $P(V)$ is the Bayesian network standard description of a complete mutual probability distribution. It can be utilised while explaining probabilistic queries regarding the subgroup from unperceived variables while it perceives other variables. A simple query model holds the conditional probability query. In this query model, the query requested for a mutual distribution including the goal is to estimate [47]:

$$P(V|E = e) = \frac{p(V,e)}{P(e)} \quad \text{Equation 2-17}$$

The equation 2-17 holds two parts,

- Variables (V) of the Query, V in the network is a subsection of arbitrary variables.
- Evidence (E), a subgroup from arbitrary variables within a pattern also the instantiation e

An extra query model is the maximum a posteriori probability (MAP). That calculates various responses to each of the variables if its non-evidence. [51]. If V and E are a collection to query variables and evidence in sequence, then.:

It allows calculating the posterior distribution of variables by probabilistic inference [52], also confirmed NP-hard [40]. Several methods introduced within an article in probabilistic inference. It separates these algorithms within exact inference procedures, plus approximate inference procedures — the comprehensive review of inference methods presented in Guo and Hsu [53].

2.3.4.1 EXACT INFERENCE

Pearl proposed a method for Bayesian Network tree-structure called the message propagation inference in [39]. The mentioned technique is a specific inference that also owns polynomial time complexity for all of the vertices. A different modern exact inference holds a joining tree or clique tree [54]. That recognized a clustering algorithm. The difficulty of the size from the biggest clique of the joining tree is exponential. Variable exclusion [55] stands further the Bayesian network exact inference algorithm. That reduces one by one of the variables through clearance out them. The number of mathematical multiplications and numerical summations it effects can adjust its complexity.

2.3.4.2 APPROXIMATE INFERENCE

Approximate inference algorithms are applied for complex structure more commonly than the exact inference. It depends on approximate inference methods of the Monte Carlo approaches. They produce the collection to pick random samples depending on the conditional probability tables within a form, approximate probabilities from query variables through repetitions from representation in the unit. Efficiency is base on the proportion of samples to represent the network structure [53]. A complexity of producing a unit also depends on network size.[56]. However, a problem among these algorithms is associated with a variety of the computed answers.

The primary method which utilises Monte Carlo approaches is logic sampling produced in [57]. Any of the other methods are holding probability weighting [58, 59], self-consequence sampling [59], heuristic interest [59], adaptive consequence sampling [41] etc.

2.4 BAYESIAN NETWORK LEARNING

In BNs, model picking and evaluation identified as learning, a title which acquired from machine learning and artificial intelligence. BN learning implemented as two procedures:

1. Structure learning: learning structure of the DAG;
2. Parameter learning: A local distribution for structured learning of DAG corresponding to the BN, using the data.

Learning can implement in two steps; as unsupervised learning, applying the information presented through a data set, or as supervised learning. Joining both procedures is the normal approach. Usually, the previous information accessible on the network is not suitable for an authority to define a BN. Furthermore, identifying the DAG structure is difficult, if it involves many variables. That is the case, for example, in gene network interpretation.

The following workflow is Bayesian. Assume a data set D and a BN, $B = (G, V)$. If we show the parameters of the global distribution of V with Θ , we can suppose using externally available information that Θ recognizes V in the parametric group of populations for modelling D and write $B = (G, \Theta)$. BN learning can then formalized as

$$\underbrace{\Pr(B | D) = \Pr(G, \Theta | D)}_{\text{learning}} = \underbrace{\Pr(G | D)}_{\text{structure learning}} \cdot \underbrace{\Pr(\Theta | G, D)}_{\text{parameter learning}} \quad \text{Equation 2-19}$$

The breakdown of $\Pr(G, \Theta | D)$ shows the steps described above and holds the logic of the learning procedure. Structure learning can accomplish by searching the DAG, G that maximizes:

$$\Pr(G | D) \propto \Pr(G) \Pr(D | G) = \Pr(G) \int \Pr(D | G, \Theta) \Pr(\Theta | G) d\Theta \quad \text{Equation 2-20}$$

Disintegrate the posterior probability of the DAG by applying the Bayes theorem (i.e., $\Pr(G | D)$) within the result of the previous distribution across the potential DAGs (i.e., $\Pr(G)$) also the possibility of using the data (i.e., $\Pr(D | G)$). Obviously, It is not probable to calculate the latter externally, including determining the parameters Θ of G [60].

The prior distribution $\Pr(G)$ produces an excellent plan to introduce any prior information possible on the conditional independence associations among the variables in V . For example; we want that some arcs should exist within or missing from the DAG, to estimate to the penetrations achieved. It may also have needed that some arcs, if present in the DAG, must locate specifically if this way is the exclusive one that makes sense under the light of the logic defining the appearance of the standing model.

The usual regular selection for $\Pr(G)$ is a non-informative prior to the space of the DAGs, allowing the equivalent possibility to all DAG. It may refuse any DAGs for prior information, as explained before. Furthermore, complex priors (known as structural priors) are more probable, just unusually used in practice for a pair [60]. First, applying a normal probability distribution renders $\Pr(G)$ unnecessary in maximizing $\Pr(G | D)$. It makes it suitable for both computational and algebraic reasons. Second, the number regarding potential DAGs rises the number of nodes exponentially.

Defining a prior distribution across such a huge number of DAGs is a challenging responsibility for regular small problems.

Calculating $\Pr(D|G)$ is similarly uncertain from both an algebraic and computational point of representation. Beginning of the breakdown within local distributions, we can advance by factors of $\Pr(D|G)$ in the following way: [60]

$$\begin{aligned} \Pr(D|G) &= \int \prod_{i=1}^p [\Pr(X_i | \Pi_{X_i}, \Theta_{X_i}) \Pr(\Theta_{X_i} | \Pi_{X_i})] d\Theta \\ &= \prod_{i=1}^p [\int \Pr(X_i | \Pi_{X_i}, \Theta_{X_i}) \Pr(\Theta_{X_i} | \Pi_{X_i}) d\Theta_{X_i}] = \prod_{i=1}^p E_{\Theta_{X_i}} [\Pr(X_i | \Pi_{X_i})] \end{aligned}$$

Equation 2-21

Using this form, $\Pr(D|G)$ can be calculated in a sensible time also for massive datasets. This is reasonable both to the multinomial distribution considered for discrete BNs (via its conjugate Dirichlet posterior) and for the multivariate Gaussian distribution considered for continuous BNs (via its conjugate Inverse Wishart distribution). For discrete BNs, we can determine, $\Pr(D | G)$ in a Bayesian Dirichlet equivalent uniform (BDeu) score from [61]. Because it is the unique fragment of the BDe group of scores in normal usage, it is referred to as BDe.

BDe allows a flat score both over the parameter field of each node and the period of the DAGs:

$$\Pr(G) \propto 1 \text{ and } \Pr(\Theta_{X_i} | \Pi_{X_i}) = \frac{\alpha}{|\Theta_{X_i}|} \quad \text{Equation 2-22}$$

The exclusive parameter of BDe is the perfect representation size α compared among the Dirichlet prior, which concludes how much power it allocates to the prior as the size of an ideal description maintaining it. Following these hypotheses, BDe uses the following form [60]:

$$\text{BDe}(G,D) = \prod_{i=1}^p \text{BDe}(X_i, \Pi_{X_i}) = \prod_{i=1}^p \prod_{j=1}^{q_i} \left\{ \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + n_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk})}{\Gamma(\alpha_{ijk} + n_{ijk})} \right\} \quad \text{Equation 2-23}$$

where:

- p is the number of nodes in G ;
- r_i is the number of classes concerning node X_i ;
- q_i is the number of arrangements from the categories of X_i 's parents;
- n_{ijk} is the number for individuals who have the j th class for node X_i and the k th arrangement for its parents.

It names the similar posterior probability to GBNs Bayesian Gaussian equivalent uniform (BGeu) from [62], which again commonly referred to as BGe. Likewise, to BDe, it implies a noninformative prior to both the parameter range of every node also the range of the DAGs; and its only parameter is the ideal representation size α . Its definition is complicated, and will not be described here.

As a result of the problems described above, two options on the $\Pr(D|G)$ have been defined [60]. The first one is the use of the Bayesian Information Criterion (BIC) as an estimate of $\Pr(D | G)$, as

$$\text{BIC}(G, D) \rightarrow \log \text{BDe}(G, D) \text{ as the sample size } n \rightarrow \infty. \quad \text{Equation 2-24}$$

BIC is analyzable and just based on the probability function,

$$\text{BIC}(G,D) = \sum_{i=1}^p \left[\log \Pr(X_i | \Pi_{X_i}) - \frac{|\Theta_{X_i}|}{2} \log n \right] \quad \text{Equation 2-25}$$

Which, is relatively simple to calculate. The other option is to circumvent the requirement to establish a standard of goodness-of-fit to the DAG also to apply

conditional independence experiments for learning the DAG structure individual arc in time [60].

Once become learned the DAG structure can drive to parameter learning, for which we can determine the parameters for the set of nodes X . Considering that parameters relating to various local distributions are independent, really require determining just the parameters of individual local distributions in time. Following the Bayesian procedure described in Equation (2-19), this would need to get the value of the Θ which maximizes $\Pr(\Theta|G, D)$ by its elements $\Pr(\Theta_{X_i}|X_i, \Pi_{X_i})$. Additional approaches to parameter estimation exist, such as the highest likelihood regularized evaluation.

Local distributions in tradition require just a small number of nodes, i.e., X_i and its parents Π_{X_i} . Their dimension regularly does not balance among the number of nodes in the BN (and often considered being limited by a fixed number of nodes while determining the computational complexity of algorithms), therefore circumventing the nominal curse of dimensionality. That shows every local distribution owns the relatively few numbers from parameters for the test individually and also that estimates are specific in greater proportion within a size from Θ_{X_i} plus a sample size.

2.4.1 LEARNING THE STRUCTURE OF BAYESIAN NETWORKS

Suppose the following circumstances. Any means provide representations of states of a candidate BN (N) across the universe U , and the required building the BN of a problem. It is a common framework for Bayesian networks structural learning. Meanwhile, the actual environment unable to sure that can test the states of the network, but we will consider this case. Also, consider the sample is appropriate, and a set D from states shows a distribution $P_N(U)$ which enable via N .

We think all connections within N are required; for example, if the connection is released, then a network producing unable to express $P(U)$. It can explain as arises: if parents of X are $pa(X)$ are, also Y represents each concerning them, next there are a couple of cases x_1 and x_2 of Y and an arrangement z of the different parents so that

$$P(X|x_1, Z) \neq P(X|x_2, Z).$$

To get an M , near to N in Bayesian network, can be accomplished through operating learning parameter during every potential structure also picking those types to which

PM(U) near to (U). Aforementioned straightforward procedure challenged among three difficulties, that are necessary for Bayesian networks learning. First, the area from every Bayesian network structure is significant. In reality, the amount from various structures, $f(n)$, raises even larger to exponentially during the amount of nodes n it can get (recent case estimations give in Table2.1):

$$f(n) = \sum_{i=1}^n (-1)^{i+1} \frac{n!}{(n-i)! i!} 2^{i(n-i)} f(n-1) \quad \text{Equation 2-26}$$

Second, while seeking within the network structures, we may finish up by some uniformly excellent structures candidate. For over a whole graph in Bayesian network able to describe each configuration covering its universe, we comprehend that it will regularly own some candidates, although a BN over a comprehensive graph will barely be the accurate solution. If so, it is a limited solution.

Table 2.1. The table presents the amount of various DAGs that can produce several nodes. For example, there are $1.4 \cdot 10^{41}$ different DAGs with 14 nodes.

NODES	Number of DAG	NODES	Number of DAG
1	1	8	$7.8 \cdot 10^{11}$
2	3	9	$1.2 \cdot 10^{15}$
3	35	10	$4.2 \cdot 10^{18}$
4	543	11	$3.2 \cdot 10^{22}$
5	29281	12	$5.2 \cdot 10^{26}$
6	$3.8 \cdot 10^6$	13	$1.9 \cdot 10^{31}$
7	$1.1 \cdot 10^9$	14	$1.4 \cdot 10^{41}$

Third, it has a difficulty from over-fitting: a picked model is so familiar to $P_D^\#(U)$ least aberrations of $PN(U)$, over, the comprehensive graph can describe (U) correctly, still, D may have inspected an incompetent system. It has two approaches applied for Bayesian networks structure learning; score-based plus constraint-based. A score-based approach provides a sequence of applicant Bayesian networks, compute a score during all applicant, also declare an applicant of the most significant score. The

constraint-based approaches organize a collection of conditional independence observations, including the data plus apply the set to construct a network among d-separation attributes comparing the conditional restricted independence properties.

To show a centre of structural learning, it should apply these following rule: A Bayesian network $M = (S, \theta S)$ composed of the structured network, S , plus a collection of parameters, θS , where the conditional probabilities of the model defined by parameters. The S is a structure composed of a DAG, $G = (U, E)$, mutually among a designation of the event period for every node per variable within a graph.

2.4.1.1 THE SCHEMA FOR LEARNING STRUCTURE

All DAG that contains the same node can be disjoint in the equivalence classes by Markov equivalence; also all DAG that produced Markov Equivalence class has equal distribution probability. Furthermore, the DAG pattern can build upon a graph called DAG that expresses the entire Markov equivalence class. We will use GP as a stochastic random variable whose potential values are DAG models, (gp). As far as they involve the genuine corresponding frequency distribution, a DAG model case (gp) is the case that (gp) is dedicated to the corresponding frequency distribution. In some circumstances, we may recognize DAG related problems. For instance, if an issue is a causal structure between the variables, then $X1 \rightarrow X2$ expresses the case that $X1$ causes $X2$, while $X2 \rightarrow X1$ describes the different events that $X2$ causes $X1$. But unless declared, only check DAG model issues, including the notation $\rho|G$ confirms the quantity function in the developed Bayesian network including the DAG (G). It seems not to require that the DAG (G) is an issue.

We have the following explanation concerning learning structure:

Definition 6 The following makes up a multinomial Bayesian network structure learning schema:

1. n random variables $X_1, X_2, \dots X_n$ with mutual joint probability distribution P ;
2. an equivalent representation size N ;
3. for each DAG model (gp) including the n variables, a multinomial expanded Bayesian network ($G, F(G), \rho|G$) including equivalent sample size N , where G is any

part of the equivalence group expressed by (gp), such that P is the probability distribution in its secured Bayesian network.

Note that even though a Bayesian network containing the DAG $X1 \rightarrow X2$ can include a configuration in which $X1$ and $X2$ are independent, the case (gp)₁ is the case they conditioned also, does not allow the case they are independent. As usual, we do not immediately select a mutual probability distribution because the number of values in the mutual distribution increases exponentially with the number of variables. Preferably we select dependent distributions from the expanded Bayesian networks such that the probability distributions in all the fixed Bayesian networks are equivalent. For, a presented DAG model (gp), we first discover a DAG G in the equality group it represents. Then in the expanded Bayesian network corresponding to G for all i, j, and

k we set: $a_{ijk} = \frac{N}{r_i q_i}$

where r_i is the number of potential values of X_i in G, and q_i is the number of various instantiations of the parents of X_i [61] presents other techniques for testing priors.

2.4.1.2 PROCEDURE FOR LEARNING STRUCTURE

This part displays how we can learn structure using a multinomial Bayesian network structure learning schema. We begin with this explanation:

Definition 7 The following forms a multinomial Bayesian network structure learning space:

1. a multinomial Bayesian network structure learning schema, including the variables X_1, X_2, \dots, X_n ;
2. A stochastic variable GP whose scope comprises every DAG models including the n variables, and for any value gp of GP a prior probability $P(gp)$;
3. A set $D = \{X^{(1)}, X^{(2)}, \dots, X^{(M)}\}$ of n-dimensional arbitrary vectors such that every $X_i^{(h)}$ has the equivalent space as X_i for any value gp of GP, D is a multinomial Bayesian network sample of size M with parameter $(G, F^{(G)})$, where $(G, F^{(G)})$ is the multinomial expanded Bayesian network comparing to gp in the schema's specification.

A scoring model for a DAG (or DAG model) is a role that selects a meaning to each DAG (or DAG model) depend on consideration based on the data. The formulation in

Equation 2-23 is named as Bayesian scoring criterion score B and applied to score both DAGs and DAG models.

$$\text{score}_B(d, gp) = \text{score}_B(d, G) = P(d|G).$$

Note that in Equation 2-23, we used a DAG pattern to calculate the probability that $D = d$. Therefore, this structure was part of the prior experience learning to evolve in our definition space, also since we did not train on it. Consider, the conditional probability individually explained that is, a basis on a selection of DAGs for $(G, F(G), \rho|G)$. Presented a multinomial Bayesian network structure learning data and space, model collection decomposed of picking and determining the DAG models, including highest probability conditional on the data. The goal of the model collection is to learn a DAG pattern subject to its parameter values (a model) that can apply to decision making and inference. We could enhance a Bayesian network, whose DAG is in the equality group described by $gp1$, to prepare inference including $X1$ and $X2$. Note that we grow the DAG model that is the one including the dependency because in the data the variables are deterministically correlated. Learning from a Mixture of Observational and Experimental Data.

The Bayesian scoring criterion (Equation2-23) regarding every case concerned and whose value corresponds to the equal probability distribution can be used to learn and test the structure just when all the data is observational. That is if no values are available for every variable by conducting a randomized control experiment (RCE). As usual, we can own both observational data also temporary data (data collected of an RCE) for a presented collection of variables. For instance, in the medical area, it involves a large deal of observational data in routinely handled electronic medical records. For specific variables of high clinical importance, we sometimes own data collected from an RCE. Cooper and Yoo enhanced an approach for using Equation 2-23 to score DAGs by using a hybrid of observational, and experimental data [63]. The scoring method presented is applied in several algorithms and investigations ([64], [65]). Cooper and Yoo present managing the situation in which the guidance is stochastic [63]. Cooper represents learning from a composite of observational, experimental, and case-control (biased sample) data [66].

2.4.1.3 THE COMPLEXITY OF STRUCTURE LEARNING

If there are just a few variables, we can exhaustively calculate the probability of all DAG models as produced. We then pick the values of (gp) that maximize $P(d|gp)$ (Note that there should be higher than one maximizing model.) If the number of variables is not few, to get the maximizing DAG models by considering every DAG models is computationally inconvenient. [67] has shown the quantity of DAGs including n nodes provided by the following repetition:

$$f(n) = \sum_{i=1}^n (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i) \quad n > 2 \quad \text{Equation 2-27}$$

$$f(0) = 1$$

$$f(1) = 1.$$

They transmit it as an activity to show $f(2) = 3$, $f(3) = 25$, $f(5) = 29,000$, and $f(10) = 4.2 \times 10^{18}$. There are smaller DAG models than there are DAGs, but this number further is forbiddingly high [68]. Chickering has proven that for certain classes of prior distributions, the difficulty of getting the usual probable DAG patterns is NP-complete [29]. One way to manipulate a problem like this is to improve heuristic search algorithms.

2.4.1.4 CONSTRAINT-BASED METHODS

Constraint-based algorithms depended on the original work of Pearl on maps and its importance to causal graphical patterns. His Inductive Causation (IC) algorithm [69] presents a structure for learning the DAG structure of BNs applying conditional independence tests.

The structure of the IC algorithm is given in Algorithm 2.1. The initial step recognizes which two variables joined through an arc, despite its direction. These variables cannot be independent, given some different variables, because they cannot be d-separated. This action can further view as a backward collection method beginning with the full pattern with a comprehensive graph and pruning depended on analytical tests for conditional independence. The next step deals with including a description from the v-

structures between two non-adjacent nodes A and B with a general neighbor C. By description, v-structures hold just the major joint in which the two non-adjacent nodes are not independent conditional on the third one. If there is a group of nodes that contains C and d-separates A and B, the three nodes are an element for a v-structure joined on C. The condition can be confirmed by conducting a conditional independence analysis for A and B into each potential subgroup of their normal neighbors that covers C.

By a completion to the other round, both the v-structures and the skeleton of the network identifier, so the equality type the BN refers to recognized network. The last round of the IC algorithm recognizes constrained arcs and direction them recursively to get the CPDAG representing the equality type recognized by the previous rounds. An essential problem of the IC algorithm is that they can use the first pair of rounds in the method illustrated in Algorithm 2.1 on several real-world problems because of the exponential number of potentially conditional independence connections.

This has driven into the enhancements to developed algorithms such as

- PC: the primary practical utilization of the IC algorithm [70];
- Grow-Shrink (GS): depended on the Grow-Shrink Markov blanket algorithm [8], an easy forward collection Markov blanket disclosure approach;
- Incremental Association (IAMB): depending on the Incremental Association Markov blanket algorithm [71], a pair-phase pick scheme;

- Fast Incremental Association (Fast-IAMB): a modification to IAMB which applies the uncertain stepwise foremost preference to decrease the number of conditional independence analyses [72];

Algorithm 2.1 Inductive Causation Algorithm

1. For each pair of nodes A and B in V search for set $SAB \subset V$ such that A and B are independent given SAB and $A, B \notin SAB$. If there is no such a set, place an undirected arc between A and B .

2. For each pair of non-adjacent nodes A and B with a common neighbor C , check whether $C \in SAB$. If this is not true, set the direction of the arcs $A-C$ and $C-B$ to $A \rightarrow C$ and $C \leftarrow B$.

3. Set the direction of arcs which are still undirected by applying recursively the following two rules:

(a) if A is adjacent to B and there is a strictly directed path from A to B then set the direction of $A-B$ to $A \rightarrow B$;

(b) if A and B are not adjacent but $A \rightarrow C$ and $C-B$, then change the latter to $C \rightarrow B$.

4. Return the resulting CPDAG.

- Interleaved Incremental Association (Inter-IAMB): extra modification of IAMB, uses the foremost stepwise choice [71] to bypass false positives in the Markov blanket exposure stately [71].

All those algorithms, and PC, learn the Markov blanket of every node. This introductory round considerably analyzes the association to neighbors. It produces a meaningful decrease in the number of conditional independence analyses, and accordingly of the overall computational complexity of the learning algorithm. Potential enhancements are possible by leveraging the equivalence of Markov blankets. While it involves the property of the learned CPDAGs, in general, Inter-IAMB provides some false positives than GS, IAMB or Fast-IAMB, while producing a relative number of false negatives. The PC algorithm as enlarged in [73], [74] and [42] are further aggressive. In case of high dimensional data sets, the guaranteed pick is reasonably the Semi-Interleaved Hiton-PC from [75], which can balance thousands of variables.

Conditional independence analysis is applied to learn discrete BNs are functions of the observed frequencies $\{n_{ijk}, i = 1, \dots, R, j = 1, \dots, C, k = 1, \dots, L\}$ for the random variables X and Y also for every arrangement of the conditioning variables Z .

- The mutual information analysis, an information-theoretic range measure is described as

$$MI(X, Y|Z) = \sum_{i=1}^R \sum_{j=1}^C \sum_{k=1}^L \frac{n_{ijk}}{n} \log \frac{n_{ijk} n_{++k}}{n_{i+k} n_{+jk}} \quad \text{Equation 2-28}$$

and comparable to the log-likelihood proportion analysis G^2 (they differ by a 2^n factor, wherever n is the representation size) [76]

- The standard Pearson's X^2 analysis for contingency tables computes::

$$X^2(X, Y|Z) = \sum_{i=1}^R \sum_{j=1}^C \sum_{k=1}^L \frac{(n_{ijk} - m_{ijk})^2}{m_{ijk}} \quad \text{Equation 2-29}$$

where $m_{ijk} = \frac{n_{i+k} n_{+jk}}{n_{++k}}$

A different possibility denotes the shrinkage estimator for the shared information defined [76] and considered in BNs in [77].

2.4.1.5 SCORE-AND-SEARCH BASED METHODS

Score-based learning algorithms describe the utilization of heuristic optimization procedures to the difficulty of learning the structure of a BN. Every applicant BN has shown a network score following its success of fit, which the algorithm later tries to maximize. Among these algorithms are:

- Greedy search algorithms such as hill-climbing among random restarts or tabu search [78]. Specific algorithms examine a search area beginning of a network structure (normally with no arc) including reversing, adding, and deleting single arc in time till they can update a score (see Algorithm 2.2);
- Genetic algorithms, which mimic real development within a repeated pick from a “most appropriate” types plus the mixing from their properties [79]. In this state, it investigates the search space for the crossover (that joins a structure of pair networks) plus mutation (which includes arbitrary modifications) stochastic executives;

- Simulated annealing [78]. The algorithm implements a stochastic local search via providing adjustments that improve the score of a network plus, concurrently, according to modifications that decrease it, including a probability inversely related to reduce the score.

Algorithm 2.2 Hill-Climbing Algorithm

1. Pick the structure of the network G covering V , normally (however not significantly) empty.
2. Calculate a score of G , expressed as $ScoreG = Score(G)$.
3. Valued $maxscore = ScoreG$.
4. Iterate the next rounds as long as $maxscore$ improvements:
 - (a) during each potential arc reversal, addition, or deletion not happening within the cyclic network:
 - i. calculates the score of the adjusted network G^* , $ScoreG_ = Score(G^*)$:
 - ii. if $ScoreG_ > ScoreG$, set $G = G^*$ also $ScoreG = ScoreG_$
 - (b) update $maxscore$ with the current state from $ScoreG$.
5. Return the DAG G .

A general survey of certain heuristics and complementary methods from artificial intelligence presented in [80]. The exploration for the network that optimizes the BIC score begins, by default, from the clear DAG. The process that improves the BIC score the maximum is, at every step, the expanding of one arc that will show in the final DAG (see Figure 2.9).

Neither (hc) nor (tabu) are capable of learning the true DAG. There are several causes for such a performance. For example, it is possible to both algorithms, to held at a local maximum because of an unsuitable selection at the beginning point of the exploration. The algorithms depended on the scoring function effort for finding the graph that a picked higher score, which normally established mostly standard from fitness among a data plus a graph. All of them apply a scoring function within the organization and an exploration method to estimate the honesty of all examined structures from the area of solutions. They take various learning algorithms based on the exploration procedure applied, and at the descriptions from a scoring function plus a search area. They depend on the scoring functions in many policies, so as the minimum description length [81];

[82]; [83], [78]; [84], information and entropy [85]; [86], or Bayesian approaches ([87]; [88]; [89]; [90]. We will explain later the normal scoring functions in-depth detail. They involve a search, frequently used ones are local search processes [91]; [88]; [87]; [61]) because of the exponentially great size of the there is an increasing concern in different heuristic exploration techniques such as tabu search [92]; simulated annealing [91]), branch and bound [93], [78]), Markov chain Monte Carlo [94], evolutionary programming and genetic algorithms [95]; [96], ant colony optimization [14]), variable neighborhood search [97], estimation of distribution algorithms [98] and greedy randomized adaptive search procedures (GRASP) [14]. Utmost learning algorithms apply various search techniques just an equivalent search area: a DAG area. Potential options are an area regarding the organizations of a variables [99]; [100]; [14]; [97]; [20]), including a subsequent search within a DAG area cooperative among the regulation; an area from primary graphs [69] (further called completed or patterns PDAGs), which partly DAG or PDAGs that canonically describe identity groups regarding DAGs [101]; [51]; [102]; [50]; [103]; also a specific area of RPDAGs (limited PDAGs), which further describe sameness groups of DAGs [104]; [92]). Each learning techniques explore a DAG area among the local search-depended procedure, able to enhance effectiveness if a scoring function applied owns the characteristic of decomposability. The scoring function (g) is decomposable if the mark selected into any structure can represent the whole (within a logarithmic range) of local states which are based just on every node including its parents: [46]

$$g(G: D) = \sum_{X_i \in U_n} g(X_i, Pa_G(X_i): D) \quad \text{Equation 2-30}$$

$$g(X_i, Pa_G(X_i): D) = g(X_i, Pa_G(X_i): N_{X_i, Pa_G(X_i)}^D)$$

where $N_{X_i, Pa_G(X_i)}^D$ is the adequate statistics for each group of variables $\{X_i\} \cup Pa_G(X_i)$ within D , that is a number from situations in D conformable to all potential arrangements of $\{X_i\} \cup Pa_G(X_i)$. For instance, an exploration process that just changes individual arc by any transit can estimate the growth achieved through this exchange. It can reuse the largest from earlier computations also just a statistic to variables they must change whose parent organizations need to recompute. While the process, the deletion or insertion of an arc $X_j \rightarrow X_i$ in a DAG G can estimate by measuring just individual fresh local score, $g(X_i, Pa_G(X_i) \cup \{X_j\}: D)$ or $g(X_i, Pa_G(X_i) \setminus \{X_j\}: D)$, sequentially; the reversal of an arc ($X_j \rightarrow X_i$) challenges the valuation to pair fresh local scores, $g(X_i, Pa_G(X_i) \setminus \{X_j\}: D)$ and $g(X_j, Pa_G(X_j) \cup \{X_i\}: D)$

The different attribute that is especially impressive if the exploration of the learning algorithm in a space of identity classes of DAGs are named the score equivalence: the scoring function g is score equivalent if it selects the corresponding value to each DAGs that is described through the equivalent fundamental graph.

In this way, the outcome regarding estimating the identity group shall be equal for which they pick DAG of the type. Several methods to calculate the consistency of a DAG regarding a data set. They can be classify into two levels: Information and Bayesian criteria.

A- Bayesian Scoring Functions

Beginning with a prior probability distribution for a potential network, the common approach is calculating a posterior probability conditioned on every accessible data D , $p(G|D)$. The greatest network holds an organization which maximizes a posterior probability. That not needed for calculating $p(G|D)$ also during related goals, calculating $p(G, D)$ holds adequate for an expression $p(D)$ is equivalent to each of potential networks. While that was comfortable to operate within a logarithmic range, during tradition, scoring functions practice a value $\log(p(G, D))$ preferably of $p(G, D)$ [87] introduced one from initial scoring functions in Bayesian, named K2. It can represent multinomial distributions, parameter modularity, reduction of missing values, parameter confidence, the regularity of the prior distribution provided in the network structure:

$$g_{K2}(G: D) = \log(p(G)) \sum_{i=1}^n \left[\sum_{j=1}^{q_1} \left[\log \left(\frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \right) + \sum_{k=1}^{r_i} \log(N_{ijk}!) \right] \right] \quad \text{Equation 2-31}$$

where $p(G)$ denotes the prior probability of the DAG G . later, the so-called BD (Bayesian Dirichlet) score introduced by [61] as a popularization of K2:

$$g_{DB}(G: D) = \log(p(G)) + \sum_{i=1}^n \left[\sum_{j=1}^{q_1} \left[\log \left(\frac{\Gamma(\eta_{ij})}{\Gamma(N_{ij} + \eta_{ij})} \right) + \sum_{k=1}^{r_i} \log \left(\frac{\Gamma(N_{ijk} + \eta_{ijk})}{\Gamma(\eta_{ijk})} \right) \right] \right]$$

$$\text{Equation 2-32}$$

where the rates η_{ijk} are the hyper-parameters involving the Dirichlet prior distributions from parameters provided by network structure, also $\eta_{ij} = \sum_{k=1}^{r_i} \eta_{ijk}$. $\Gamma(\cdot)$ is the function Gamma, $\Gamma(c) = \int_0^{\infty} e^{-u} u^{c-1} du$. It should be noted that if c is an integer, $\Gamma(c) = (c-1)!$. If values about every hyper-parameter occur $\eta_{ijk}=1$, reach the K2 score being a particular instance of BD. Within working terms, the designation to a hyper-

parameters η_{ijk} implies hard (but while apply non-informative tasks, like the ones used via K2). In other words, we can edit the BD scores as:

$$S_i() = \sum_{j \in J_i} \left(\log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + n_{ij})} + \sum_{k \in K_{ij}} \log \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})} \right) \quad \text{Equation 2-33}$$

where $J_i \doteq J_i^{\Pi i} \doteq \{1 \leq j \leq r_{\pm} : n_{ij} \neq 0\}$ because $n_{ij} = 0$ shows that all phases cancel each other. Equivalently, $n_{ijk} = 0$ shows that the terms of the regional summation drop out, so let $K_{ij} \doteq K_{ij}^{\Pi ii} \doteq \{1 \leq k \leq r_{\pm} : n_{ijk} \neq 0\}$, be the contents of the classes of X_i such that $n_{ijk} \neq 0$. Let $K_{ij} \doteq \cup K_{ij}^{\Pi ij}$ be a vector among each content comparing to non-zero numbers for J_i (Note that the representation needs regarding as a concatenation of vectors, as we allow $K_i^{\Pi ii}$ to have repetitions). The counts n_{ijk} (and consequently $n_{ij} = \sum_k n_{ijk}$) fully determined if we comprehend the parent collection Π_i .

Rewrite the score:

$$s_i(\Pi_i) = \sum_{j \in J_i} \left(f(K_{ij}, (\forall k) \forall k) + g \left((n_{ijk}) \forall k, (\alpha_{ijk}) \forall k \right) \right) \quad \text{Equation 2-34}$$

With $f(K_{ij}, (\forall k) \forall k) = \log \Gamma(\alpha_{ij}) - \sum_{k \in K_{ij}} \log \Gamma(\alpha_{ijk})$

$$g \left((n_{ijk}) \forall k, (\alpha_{ijk}) \forall k \right) = -\log \Gamma(\alpha_{ij} + n_{ij}) + \sum_{k \in K_{ij}} \log \Gamma(\alpha_{ijk} + n_{ijk})$$

By studying the other hypothesis of likelihood identity [90]; [64], it is probable to designate the hyper-parameters comparatively. While each effect means a scoring function named BDe (and its representation is like on BD one under Equation 2-30), a hyper-parameter can calculate within the due process:

$$\eta_{ijk} = \eta * p(x_{ik}, w_{ij} | G_0) \quad \text{Equation 2-35}$$

where $p(\cdot | G_0)$ describes a probability distribution connected with a prior Bayesian network G_0 and η is a parameter describing the similar representation size. A suitable case of BDe that the prior network selects a legal option to any choice of $\{X_i\} \cup \text{PaG}(X_i)$. It names the resulting score BDeu, which was formally introduced by [88]. This score is just based on an individual parameter, the comparable representation size h , and represented as:

$$g_{\text{BDeu}}(G: D) = \log(p(G)) + \sum_{i=1}^n \left[\sum_{j=1}^{q_i} \left[\log \left(\frac{\Gamma(\frac{\eta}{q_i})}{\Gamma(N_{ij} + \frac{\eta}{q_i})} \right) + \sum_{k=1}^{r_i} \log \left(\frac{\Gamma(N_{ijk} + \frac{\eta}{r_i q_i})}{\Gamma(\frac{\eta}{r_i q_i})} \right) \right] \right]$$

Concerning the expression $\log(p(G))$ that occur within a previous expression, this is simple in imagining the normal distribution (but when own knowledge on the highest advantage from individual structures), so that fits a fixed and able to reject.

B- Scoring Functions based on Information Theory

Certain scoring functions express different alternatives to estimating a level from fitness regarding the DAG on data set plus depending covering information plus codification approaches. Coding tries to decrease as much as several components they require to describe a message (based on its probability). The minimum description length (MDL) principle chooses some coding which needs the tiniest range for describing messages. The different standard formulation from an identical concept proves that to describe the data set with an individual type of special kind; the valid form is one that reduces an amount from description length from a model also a description length from a data given the model. Difficult forms regularly need comprehensive description lengths only decrease a description length of a data given a form. On the other side, pure models need smaller description lengths, just the description length from a data provided model increments. The minimum description length postulate sets a suitable trade-off between precision and complexity. In the definitions, the data set to express holds D , plus a picked group to representations are Bayesian networks. The description length covers the length needed for describing a network and a length specified for describing a data given a network ([83], [78]; [84]; [82]; [81]). To describe the network, we need to collect its probability states, and this needs a period comparable on several free parameters from a factorized mutual probability distribution. This value is named network complexity also expressed as:

$$C(G) = \sum_{i=1}^n (r_i - 1)q_i \quad \text{Equation 2-37}$$

The general proportionality factor is $\frac{1}{2} \log(N)$ [105]. The description length of the network is:

$$\frac{1}{2} C(G) \log(N) \quad \text{Equation 2-38}$$

Concerning a detail from a data showed this model, via utilising Huffman codes it is fieldsets deny is denote the negative from a log-likelihood, a logarithm from the probability function from a data concerning a network. The value mentioned above means a least to the rigid network structure while determining the network parameters

of a data set itself through utilising highest probability. They can display a log-likelihood under the procedure as following [78]:

$$LLD(G) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log\left(\frac{N_{ijk}}{N_{ij}}\right) \quad \text{Equation 2-39}$$

The scoring function (MDL) through improving marks to offer among the maximization difficulty) is:

$$gMDL(G: D) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log\left(\frac{N_{ijk}}{N_{ij}}\right) - \frac{1}{2} C(G) \log(N) \quad \text{Equation 2-40}$$

The different process for estimating the status of a Bayesian network apply criteria depended on information theory, including any from certain compared among a past one. The fundamental approach stands to pick a network structure that strongly matches the data, punished with several parameters that are important to define the mutual distribution. It drives to a popularization from a scoring function within Equation 2-46:

$$g(G:D) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log\left(\frac{N_{ijk}}{N_{ij}}\right) - C(G) f(N) \quad \text{Equation 2-41}$$

where $f(N)$ holds a positive penalization function. When $f(N) = 1$, it depends on a score at the Akaike information criterion (AIC) [106]. If $f(N) = \frac{1}{2} \log(N)$, formerly a score, named BIC, implies depended on a Schwarz information criterion [107], that corresponds among the MDL score. If $f(N) = 0$, it holds the highest probability score, although that does not make beneficial while the valid network applying the principle regularly means a perfect network that incorporates whole a potential arc. It is fascinating to remark that a different way of signifying the log-likelihood under Equation 2-34 is:

$$LLD(G) = -N \sum_{i=1}^n H_D(X_i | Pa_G(X_i)) \quad \text{Equation 2-42}$$

where $H_D(X_i | Pa_G(X_i))$ denotes the dependent entropy of the variable X_i given its parent set $Pa_G(X_i)$, as the probability distribution P_D :

$$H_D(X_i | Pa_G(X_i)) = \sum_{j=1}^{q_i} p_D(w_{ij}) \left(- \sum_{k=1}^{r_i} p_D(x_{ik} | w_{ij}) \log(p_D(x_{ik} | w_{ij})) \right) \quad \text{Equation 2-43}$$

and P_D is the mutual probability distribution compared among the data set D , taken of the data by the highest likelihood. The log-likelihood $LLD(G)$ can additionally express as [78]:

$$LLD(G) = -NH_D(G) \quad \text{Equation 2-44}$$

where $H_D(G)$ expresses the entropy of the mutual probability distribution compared for the graph G if they measure the network parameters of D by highest likelihood:

$$H_D(G) = - \sum_{x_1, \dots, x_n} \left(\left(\prod_{i=1}^n p_D(x_i | Pa_G(X_i)) \right) \log \left(\prod_{i=1}^n P_D(x_i | pa_G(x_i)) \right) \right)$$

Equation 2-45

The different understanding of the scoring functions depended on information is that they try to decrease the conditional entropy of all variables presents its parents, and then they explore the parent collection of all variable that provides as much information as probable on this variable (or which most restricts the distribution). It is essential to append a penalization term considering the smallest conditional entropy captured by calculating a total value for the potential variables given the parent set. Herskovits and Cooper [85] introduced an approach to bypass this over-fitting without applying a penalization formula. They applied the best score, but the method of adding arcs in the network use the averages of a statistical test which determined diversity in entropy between the existing network and that achieved by adding a new arc was statistically meaningful. Regarding the properties of the various scoring functions, each is decomposable and include the exclusion of K2 and BD; they are further score-equivalent [91].

2.4.1.6 HYBRID METHOD

Local search algorithms are hybrid BN structure learning techniques offering with local structure description and global representation optimization constrained for local information various local structure descriptions have introduced. They are applied to discover the applicant Parent-Children (PC) attitude of a destination node such as the Markov Blanket (MB) i.e. children, parents, and spouses, of the destination [71, 108] or the Max-Min Parent Children (MMPC) algorithm [48]. If the global structure description is the final purpose, Parent-Child's description is enough in succession to produce a global undirected graph that can apply as a set of constraints in the global pattern identification. For example, the original Max-Min Hill-Climbing algorithm (MMHC) (algorithm 2.3) introduced by Tsamardinos [109] joins the local association presented by a global greedy search (GS) and Max-Min Parent Children (MMPC) algorithm where the neighborhood of an assigned graph produced by the following executives: append edge assigned to edges in the local search description form (if the edge refers to a collection of constraints also if the resulting is acyclic DAG) (see algorithms 2.3), remove edge and exchange edge (if the resulting is acyclic DAG). They divide the MMPC local structure description, defined in Algorithm 2.4, into a

Algorithm 2.3 MMHC(D)

Require: Data (D)

Ensure: BN structure (DAG)

1: $G_c \leftarrow \emptyset, G \leftarrow \emptyset$

2: $S \leftarrow 0$ % Local identification

3: for all $X \in X$ do

4: $CPC_X = MMPC(X, D)$

5: end for

6: for all $X \in X$ And $Y \in CPC_X$ do

7: $G_c \leftarrow G_c \cup (X, Y)$

8: end for % Greedy search (GS) optimizing score function in DAG space

9: $G \leftarrow GS(G_c)$

10: return the DAG G found

Algorithm 2.4 MMPC(T, D)*Require: target variable (T); Data (D)**Ensure: neighborhood of T (CPC)*1: $ListC = X \setminus \{T\}$ 2: $CPC = MMPC(T, D, ListC)$ % Symmetrical correction3: for all $X \in CPC$ do4: if $T \notin MMPC(X, D, X \setminus \{X\})$ then5: $CPC = CPC \setminus \{X\}$

6: end if

7: end for

Algorithm 2.5 MMPC(T, D, ListC)*Require: target variable (T); Data (D); List of potential candidates (ListC)**Ensure: neighborhood of T (CPC)*1: $CPC = \emptyset$ % Phase I: Forward

2: repeat

3: $\langle F, assocF \rangle = MaxMinHeuristic(T, CPC, ListC)$ 4: if $assocF \neq 0$ then5: $CPC = CPC \cup \{F\}$ 6: $ListC = ListC \setminus \{F\}$

7: end if

8: until CPC has not changed or $assocF = 0$ or $ListC = \emptyset$ % Phase II: Backward9: for all $X \in CPC$ do10: if $\exists S \subseteq CPC$ and $assoc(X; T|S) = 0$ then11: $CPC \setminus \{X\}$

12: end if

13: end for

couple of responsibilities, the neighborhood description itself (MMPC), achieved by a proportional correction (X refers to the neighborhood of T if the reverse is likewise true). The neighborhood association (MMPC), described in Algorithm 2.5, uses the Max-Min Heuristic illustrated in Algorithm 2.6 in sequence repeatedly append (forward phase) in the applicant Parent-Children collection (neighborhood) of a destination variable T the variable the various directly subordinate on T probably to its current neighborhood (line 1 in algorithm 2.6). This method can append any false positives, which later removed in the backward stage.

They estimate dependency with an organization determination function Assoc like X^2 , mutual information or G^2 . The famous examples of this family are the Sparse Candidate algorithm (SC) by Friedman and Nachman [110] and the Max-Min Hill-Climbing (MMHC) algorithm by Tsamardinos, Brown, Constantin, and Aliferis [109]. They base both of these algorithms on a few rounds named maximize and limit.

Algorithm 2.6 MaxMinHeuristic(T,CPC, ListC)

Require: target variable (T); current neighborhood (CPC); List of potential candidates (ListC)

Ensure: the candidate the most directly dependent to T given CPC (F) and its association measurement

(AssocF)

1: $assocF = \max_{X \in ListC} \min_{S \subseteq CPC} Assoc(X; T|S)$

2: $F = \operatorname{argmax}_{X \in ListC} \min_{S \subseteq CPC} Assoc(X; T|S)$

In the initial round, the applicant produced for the parents of any node X_i decreased the entire node set V to a lesser set $C_i \subset V$ of nodes whose operation proved to associate to that of X_i . This results in a less also extra normal search space. The second round explores the network that maximizes a presented score function, directed to the constraints required by the C_i collections. In the Sparse Candidate algorithm, these couple of rounds implemented until there is no replacement in the network or no network upgrades the network score; the selection to the heuristics applied to perform it transmits them to the implementation. On the opposite round, in the MMHC algorithm, limit and maximize performed only once; they apply the Max-Min Parents and Children (MMPC) to learn the applicant sets C_i and a hill-climbing greedy search to get the optimal network.

2.5 EVALUATION OF STRUCTURAL ACCURACY

2.5.1. EVALUATION METRICS

In this part, we introduce techniques that estimate the status of the Bayesian network achieved through the structure learning algorithms. There are two procedures for the estimation of the structure from a learning algorithm:

- Presented a theoretical BN, $B_0 = (G_0, \theta_0)$ and data D produced by the BN, the measures estimate the status from the algorithm through associating the status of the learned graph $B = (G, \theta)$ and that of the original network B_0 including the utility to data.

– The test estimates the quality of the algorithm by analyzing the structure G_0 of the theoretical graph and the structure G of the learned graph. To this point, we mark it would be useful to differentiate the equality classes provided by the learned and primary BN. A BN received from the data identified through its sameness group. They concern the best BN if we discover that the CPDAG of the produced network is similar to the learned BN. So, all estimation metrics need to apply the sameness group to associate with the BNs for vast estimation.

There are many models offered in the literature for the evaluation of structure learning algorithms [111].

2.5.2. CONFUSION MATRIX

In the level of the pair potential during supervised learning, it dased on the four characters for evaluation of the goal by using the classifier for the analysis collection. Predictive analytics, a table of confusion, further identified as a confusion matrix, is a table in its simplest form, having a couple of rows and a couple of columns that provides the estimates of true positives, false positives, false negatives, and true negatives. Every row in the confusion matrix describes a recognized class, every column describes a forecasting class, and all cell includes the number of units in the crossing of those couple of classes. The Confusion matrix structure is presented as follows in Table 2.2.

Table 2.2: Confusion Matrix

	Predicted Class	
	Yes	No
Actual Class	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

The records in the confusion matrix hold integer numbers. The sum of the four records $TP + TN + FP + FN = n$, corresponds to the total quantity of analysis. Classifications that constitute the principal diagonal of the confusion matrix are the accurate classifications i.e., true negatives and true positives. Additional fields mean

classification errors. Several achievement metrics can obtain from the confusion matrix.

2.5.2.1. ACCURACY AND ERROR RATE

Accuracy is the rate of rightly classified cases to all cases in the examination collection, i.e. $(TP + TN) / (TP + TN + FP + FN)$. The error rate is defined as $(1 - \text{Accuracy})$. This metric analyzed by writers as having a weakness to distinguish between classes it considers. Accuracy doesn't display the accurate classifier's appearance below the skewed class population. In actuality, classifiers regularly challenge a higher number of negative cases associated with positive cases. [112]; [113]; [114], [115]; [116]) Accuracy estimates the classifier's achievement including one number to both of the groups also to the individual setting of destination situations. Extra weakness about the accuracy metric is the interchangeable achievement evaluation of a couple of separate situations. The accuracy is the equivalent of both of the situations just, for instance, classifying nearly all positives in the original one includes the faulty classification of nearly every negative. At the opposite side, classifying nearly each positive inside the other situation includes around just half of the false positives.

2.5.2.2. SENSITIVITY AND SPECIFICITY

Sensitivity and specificity are the mathematical models of achievement of binary classification analyses. Sensitivity and specificity expressed as a percentage. In clinical examinations, the sensitivity of medical analysis is the possibility of its producing a 'positive' result if the case is positive and specificity is the possibility of getting a negative outcome if the case is negative. A general optimal forecast effect can produce 100% sensitive (i.e. forecast every case of a diseased population as sick) and 100% specificity (i.e. not forecast any member of the healthy population as sick).

Visualize a scenario, where cases examined for an illness. The examination result may be positive (sick) or negative (healthy), while the real health situation of a case may be different. Four situations may occur:

- The sick case diagnosed sick termed as –“True positive”
- The healthy case classified as sick–“False positive”
- The healthy case recognized as healthy–“True negative”
- The sick case classified as healthy–“False negative”

Of the above circumstances, in two cases, a fault has happened, if a healthy case recognized as sick and the other case where a sick case classified as healthy.

System examination produce analytical conclusions about the distribution on the source of trial data. They further recognize its Statistical Significance Examination. In system testing, there is a “Null hypothesis” which compares to a supposed default “State of reality” (e.g. that a person is available of infection). Comparing to the null system is an “alternative hypothesis” which compares different situations. The purpose is to define if the null hypothesis can reject in approval of the option. The outcome of the analysis may be positive (it may mean infection) or it may be negative (i.e. it appears to no-show infection). If the outcome of the examination seems negative match including the original states of reality, a failure has happened. There are two kinds of error categorized as “Type I and Type II errors” based on which system has recognized as the reality. Type I error identified as “false positive”, or “ α ” error, the error of denying the null system if it is true. A false positive shows that an examination demands something to be positive if that is not the case. For instance, an examination assuming that a woman is pregnant if she is not pregnant. Type II error classified as “error of the second kind” or a “false negative” or “ β ” error, the error of allowing the null system when the choice system is true. Table 2.3 represents the condition:

Table 2.3 Test Result in the Confusion matrix

Test Result		Actual Condition	
		Present	Absent
Test Result	Positive	Condition Present + Positive Result = True Positive	Condition absent + Positive result = False Positive (Type I error)
	Negative	Condition Present + Negative Result = False negative (Type II error)	Condition absent + Negative result = True negative

Sensitivity is defined as:

$$\text{Sensitivity} = \frac{\text{Number of true positives}}{\text{Number of true positives} + \text{Number of false negatives}} \quad \text{Equation 2-46}$$

An individual Sensitivity value seems not to show how good the examination distinguishes the different types (i.e. about negative events). In the binary classification,

this corresponds to the identical specificity examination or equivalently the sensitivity for the different types.

Specificity is the proportion of true negatives to the number of true negatives plus false positives.

$$\text{Specificity} = \frac{\text{Number of true negatives}}{\text{Number of true negatives} + \text{Number of false positives}} \quad \text{Equation 2-47}$$

Sensitivity and specificity are helpful in providing explanations of different treatments in the medical domain being associated with conventional therapy and including different scaling of testing the increase in cases as distinguished upon old, well installed also applied principles [117].

Table 2. 4 Sensitivity and Specificity in the Confusion matrix

		Condition as determined by Gold Standard		
		Positive	Negative	
Test result	Positive	True Positive	False Positive (Type I error)	→ Positive Predictive value
	Negative	False Negative (Type II error)	True Negative	→ Negative Predictive value
		↓ Sensitivity	↓ Specificity	

2.5.2.3 PRECISION, RECALL AND F-SCORE

In this section, we concentrate on three conventional achievement metrics; precision, recall, and F-score. For instance, the experimental result may produce the numbers in a confusion matrix. Of these numbers, one can calculate the precision (p) also recall (r) as follows:

$$p = \frac{Tp}{TP+FP} \quad \text{Equation 2-48}$$

$$r = \frac{Tp}{TP+FN} \quad \text{Equation 2-49}$$

The (weighted) harmonic mean of precision and recall produces the F-score [118]

$$F_{\beta} = (1 + \beta^2) \frac{pr}{r + \beta^2 p} = \frac{(1 + \beta^2) TP}{(1 + \beta^2) TP + \beta^2 FN + FP} \quad \text{Equation 2-50}$$

Both recall and precision become an actual argument in expressions of probability. Precision may display, while the system restores the possibility that a target is essential given that, while the recall is the likelihood they deliver a suitable target.



CHAPTER 3

PROPOSED ALGORITHMS

Optimization performs a pivotal character in engineering and science. In particular, there are such common instances of relationships, including optimization that a record of applicability is limited and incomplete. Even in the highest powerful statements from optimization, there are limitations, i.e. boundaries about the applicable area involving variables and parameters. In particular, planners may ask questions such as “Given that we have access to certain resources (such as a construction material, time, financial resources.), what is the best we can do?” Several difficulties can be expressed since the problem is decreasing (or increasing) a scientific objective, namely, specific objective function [119].

Several approaches describe possibilities for traditional optimization, and they are well-developed for a class of optimization problems related to the increased optimization difficulties, which are described in the next chapter. Traditional optimization approaches further possess weaknesses, including those that not applicable to every optimization problem. There have been many substitutions approaches, during the last few decades and some of them are under improvement [120].

In this chapter, we combine some heuristic searches like Pigeon inspired optimization, Elephant Swarm Water Search, Bee optimization, Simulated Annealing, and Greedy search. Later we explain the proposed algorithms based on the mentioned algorithms.

In the first algorithm, we proposed the Bayesian network structure learning based on pigeon inspired optimization. The second and third algorithms are a hybrid between Bee and Simulated Annealing. The fourth and the fifth algorithm are a hybrid between Bee and greedy search. The final algorithm based on the Elephant swarm water search algorithm.

3.1 PIGEON INSPIRED OPTIMIZATION

In this part, we present the introduction of the pigeon, its behavior and a concept and formulation of the pigeon inspired optimization.

Pigeons are parts of the society Columbiformes which covers doves and pigeons [121]. Pigeons were used for transmitting information by the Egyptians, including transpired during various fighting operations. Homing pigeons can get their places by using three homing mechanisms: magnetic field, sun, and landmarks [122].

3.1.1 OVERVIEW OF PIGEON INSPIRED OPTIMIZATION

Extensive swarm intelligence analyses have shown how some animals, like mammals or fishes, communicate between them in the natural environments in swarm [123]. The range of these swarms in size start form small numbers living in natural places to organized colonies that held huge areas and contain millions of individuals. The skills of the group in swarms show a good robustness and flexibility [124] like preparation of routing [125], construction of nest [126], managing the task [89] and different additional complex behaviors combined in some swarms presented in [127, 128, 129]. The abilities may be very poor for individuals in the swarm, while behaviors of the group can appear in the complete swarm, like a flock of bird migration, exploring in Bee and ant colonies. To complete the task by individuals is hard, while it easily achieved by a swarm of animals. The researcher observed that the smart skills group are materialize by sets of individuals with normal skills through information transmission and swarm intelligence.

In general, swarm intelligence offers among models from some common responses of simple tools combining with themselves, including its circumstances', that drives into every evolution from a logical operative global model[34] [130]. Information passing between representatives is obscure and little. They actualize the interaction between representatives within a dispersed way without a centralized restriction mechanism. In other terms, the whole swarm intelligence form is uncomplicated in real life [131].

The group colony-level performance of the swarm that originate from the communications becomes helpful during performing complicated purposes [132]. Through World Wars (First and Second), pigeons served essentially to the UK, American, German, French and Australian forces. The significant ability of homing by pigeons to utilize the mixing of the magnetic area, the sun, and find their route around in the landmarks. Pigeons reasonably use various navigational tools through various elements presented by Guilford argues [133]. The mathematical model developed by Guilford and his partners for predicting when pigeons order the change of the route

from one to another. If pigeons begin the journey, they rely on extra tools. If during the campaign, they should turn on using landmarks if it needs to reassess the way and perform improvements.

Investigations show that the ability of a pigeon can detect various magnetic area proves that the pigeons' compelling experiences of homing based on little magnetic bits in its beak. The beak of pigeons has iron crystals; they provide a nose of the birds to the north. Investigations explain that this appears to produce a mode within flags of magnetite bits transmitted into the mind through particular trigeminal nerves [134]. Additionally, the pigeon navigation is based also to the sun either entirely or partly, the highest of the sun help the pigeon to recognize the current location and home base [135]. Modern investigations about the behaviour of the pigeon further confirm the ability of the pigeon to recognize any landmarks, principal ways, rivers, and routes to the target straight.

3.1.2 MATHEMATICAL MODEL OF PIO

About pigeons homing, a pair of processes have proposed using several rules [132]:

- A. The operator of the Compass and Map: the ability of pigeons to sense the range of earth by utilizing magneto response to configure a map in their minds. It considers elevation from the sun and a compass for adjusting the path. Since they fly to their target, this operation less depended on the magnetic bit and sun.
- B. The operator of the landmark: if the pigeons fly nearest to the target, they should depend on the landmarks near them. While it's common among landmarks, they must fly into the target straightly. While it's far in the goal also unfamiliar of the landmarks, they must keep track of the closer pigeons among those landmarks.

3.1.2.1 MAP AND COMPASS OPERATOR

Within the operator of the compass and map, the location P_i and the speed V_i of pigeon i are set and updated in D -dimension of search range within every iteration. The following formula can measure the new location P_i and speed V_i of pigeon i in the t^{th} repetition:

$$Vi(t) = Vi(t - 1).e^{-Rt} + \text{rand.}(Pg - Pi(t-1)) \quad \text{Equation 3-1}$$

$$Pi(t) = Pi(t - 1) + Vi(t) \quad \text{Equation 3-2}$$

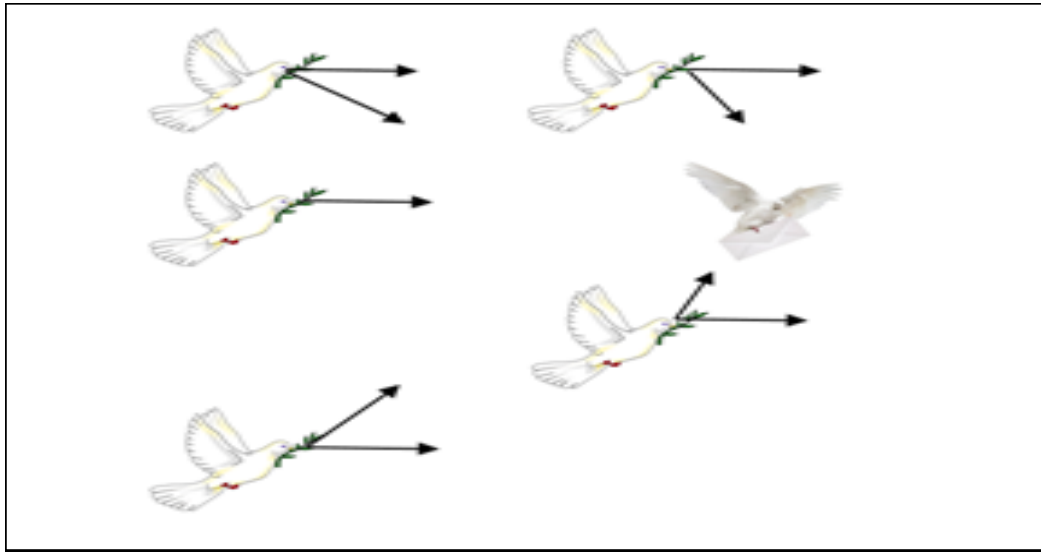


Figure 3.1. Map and compass operator model of PIO [132]

where R is the factor of compass and map, default random number is a rand , in the current location Pg is the best global, and which can achieve by associating each location with every pigeon. Figure 3.1 shows the process of compass and map form of PIO [132].

As presented in Figure 3.2, the better locations of every pigeon produced by using a compass and map. Through analysing all the locations of pigeons, that means the best

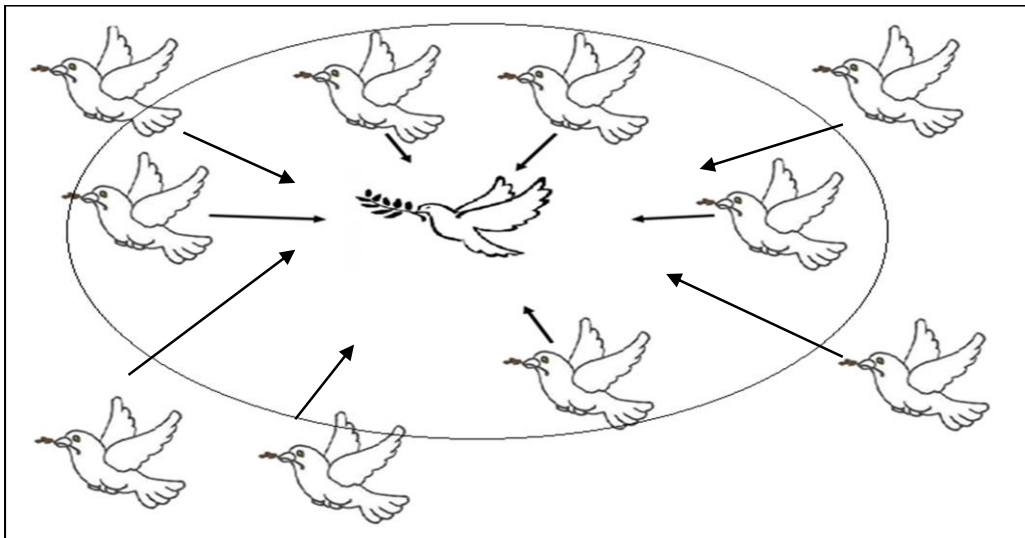


Figure 3.2 Lamndmark operator model [132]

location of the pigeon that is the right-centred. Adjusting the fly direction for every pigeon is through following this particular pigeon's direction, which is based on Equation (3-2).

3.1.2.2 LANDMARK OPERATOR

Within the operator of the landmark, the number of pigeons decreased by half through N_p within each iteration. Still, the target is not near the pigeons; also it's unknown among this landmark. Let the middle at a t^{th} repetition of some pigeon is $P_c(t)$, also assume each pigeon can fly to the target straightly. The location can update the pigeon i at a t^{th} iteration as: [132]

$$N_p = \text{ceil} \frac{N_p(t-1)}{2} \quad \text{Equation 3-3}$$

$$P_i(t) = P_i(t-1) + \text{rand}(P_c(t) - P_i(t-1)) \quad \text{Equation 3-4}$$

where $P_c(t)$ denotes the center position at the t^{th} iteration, defined as

$$P_c(t) = \frac{\sum P_i(t).fitness(P_i(t))}{N_p \sum fitness(P_i(t))} \quad \text{Equation 3-5}$$

To decrease the optimization difficulties into minimum, we select the fitness $P_i(t) = 1/(f_{\min}(P_i(t)) + \epsilon)$. For increasing the optimization into maximum, we select $(P_i(t) = f_{\max}(P_i(t)))$. For any individual pigeon, the optimal location of the N_c^{th} repetition is identified among P_p , and $P_p = \min(P_{i_1}, P_{i_2}, \dots, P_{i_{N_c}})$.

As presented in Figure 3.2, the midpoint from all pigeons (The pigeon in the center of the range) is the goal in every repetition. Half from whole those pigeons (those pigeons outside from the circle) that are far away of the target should keep track of the pigeons near to the goal. The pigeons near the target (the pigeons inside the circle) should be able to fly to the destination fast.

3.2 SIMULATED ANNEALING

3.2.1 INTRODUCTION OF SIMULATED ANNEALING

The simulated annealing methods depend on thermodynamics with an analogy, especially with the form that crystallizes and freezing of liquids, or annealing and cooling of metals. When temperatures are higher, in the liquids, the molecules move from one to another freely. While the liquid is slowly cooled, thermal mobility is lost. A meaning of optimization, SA tries to follow the rule [36]. From a higher temperature, the SA starts wherever the initial values released to expect a large domain of adaptation. They restrict the permitting for mutating input. This process drives the method on a better solution, only the actual process for annealing that provides a structure of a crystal to achieve a better. It shows they should harmonize the algorithm within an organization for maximizing the performance. They describe the SA algorithm with a specific flowchart from Figure 3.3. The principal characteristic of SA experiences for leaving from the local optimum depending on the permission control from a candidate solution [36]. While the popular solution (f_{new}) becomes the actual purpose benefit smaller (assuming minimization) than one from the previous solution (f_{old}), later a solution to they

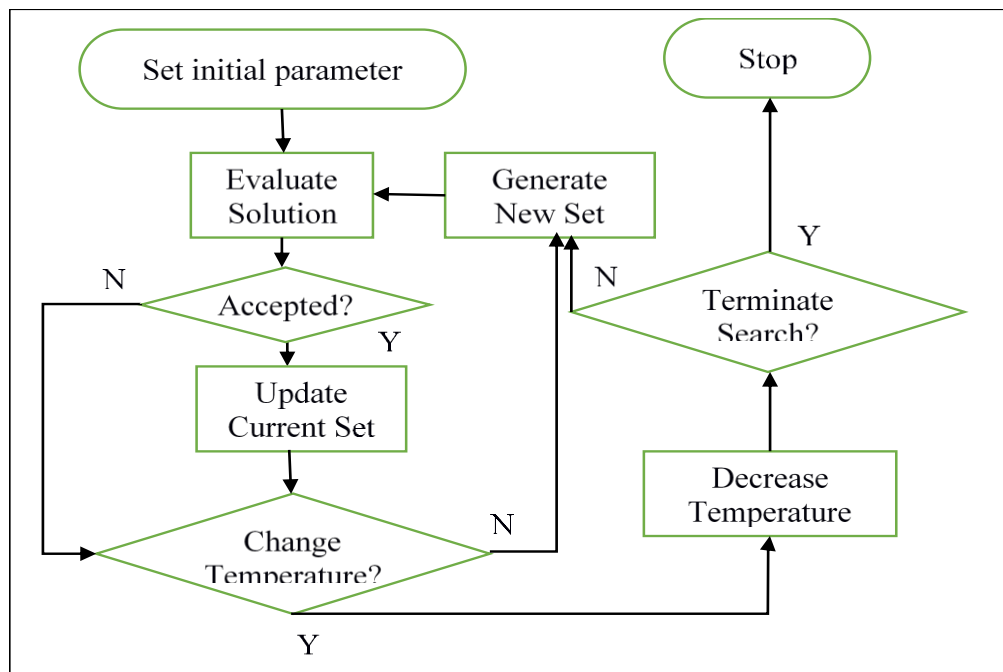


Figure 3.3 Flowchart of the Simulated Annealing algorithm [36].

accept the current state. Unless they may admit the current solution while the value provided through the Boltzmann population:

$$e^{-\frac{f_{new}-f_{old}}{T}} \quad \text{Equation 3-6}$$

is greater than a uniform random number in [0,1], where T is the ‘temperature’ control parameter.

3.2.2 SIMULATED ANNEALING ALGORITHM

In this section, the formulation of the Simulated Annealing procedures and algorithms presented.

3.2.2.1 INITIAL POPULATION

Each iterative process needs a specific description from the beginning of the inference of the parameters’ values. Any algorithms needs utility from different starting solutions. Selecting random initial values of parameters within range is a technique used by SA. Nearest choice of the starting approximation to the global optimum shall make quicker the process of optimization [36].

3.2.2.2 INITIAL TEMPERATURE

The parameter adjusting ‘temperature’ should be justified by the principle described in section (3.2.2.5). T should be very big for enabling the algorithm to leave off from a local minimum while it is not enough to transfer to a global minimum. In the application, the value of T should set within the form depended on approach considering that associated with admitting a size from the values of the aim function. It could establish within the literature [136] any experimental procedures that are able to be useful not only for picking the ‘optimum’ value of T but also at least a suitable start assessment that can harmonize.

3.2.2.3 PERTURBATION MECHANISM

The disturbance technique is a procedure for generating new solutions within a proposed solution. The process for searching the neighborhood from a current solution generating a little modification into the current solution. SA used in combinatorial problems for optimizing the parameters when they are integer numbers. While the change of parameters in any application, the investigation for solutions in a

neighborhood can perform. If (s) is a solution and can be expressed as a vector $s = (X_1, \dots, X_n)$ describing a point within a domain of search. The generated of a new solution from standard deviations by using a vector $\sigma = (\sigma_1, \dots, \sigma_n)$ for creating a disturbance to the current solution. Producing the neighbor solution from the current solution by:

$$x_{i+1} = x_i + N(0, \sigma_i) \quad \text{Equation 3-7}$$

where the standard deviation denoted as σ_i and $N(0, \sigma_i)$ is a random Gaussian number with zero mean.

3.2.2.4 OBJECTIVE FUNCTION

The goal of function or cost is a representation that, within several applications, associates some characteristic (range, cost, etc.) with the parameters that are required to be maximized or minimized. The strategy depends on determining the goals of function which compare the results of simulation experimentally. Then, the algorithm must work for finding the collection to parameters that decrease the error between experimental and simulated results. Using a normalized total of sum the squared errors, the goals of function is expressed by:

$$f_{obj} = \sqrt{\sum_c \sum_i \left(\frac{g_s(x_i) - g_e(x_i)}{g_e(x_i)} \right)^2} \quad \text{Equation 3-8}$$

Where the experimental data indicated as $g_e(x_i)$, simulate data denoted as $g_s(x_i)$, and the number of curves comprising optimization is c [36].

3.2.2.5 COOLING SCHEDULE

The largest schedule of cooling is the change of temperature:

$$T_{i+1} = sT_i \quad \text{Equation 3-9}$$

while ($s < 1$) A new report in [137] show that when the value of s in domain [0.8-0.99] the result is excellent.

The iteration numbers within each temperature is another parameter, which is associated with the range of space in the search or with the neighborhood size. The iteration number may be constant or depended on the process feedback or function of the temperature.

3.2.2.6 TERMINATING CRITERION

There are various techniques for controlling the termination from each algorithm. Standard models are:

- a) the maximum value for iteration;
- b) the minimum value of temperature;
- c) the minimum value of goal functions;
- d) the minimum value of acceptance rate.

3.2.3 IMPLEMENTATION OF THE SA ALGORITHM

The combinatorial optimization problem can be solved through an S.A. algorithm, that can make several decisions. Table 3.1 shows those decisions within two separate organizations. First, the universal elements which are necessary to produce during each implementation of S.A. also designated for that cooling or annealing scheme (process, schedule, etc.). The paper [138] represents a comprehensive program about alternative plans for fixing those values. Second, the difficulty involves special arrangements associated with a real individual challenge to solve. In this relationship, it is recognized that some performance of S.A. based on the plan which created the neighbours. The design from an S.A. algorithm requires both a fitting from all this knowledge (practical and theoretical) possible at the original difficulty including an appropriate planned collection of attempts for finding the suitable group of parameters i.e., the “tuning” of the algorithm.

Table 3.1 Designing the S.A. Algorithm

Decisions	
Generic (Cooling Scheme)	Problems Specific
<ul style="list-style-type: none">• T_o (initial temperature)• L_k (number of iterations)• T_k (temperature function)• Stopping criteria	<ul style="list-style-type: none">• i_o (initial solution)• Neighbour generation• Evaluation of ΔC_{ij}

3.3 GREEDY ALGORITHMS

One of the simple design methods is the greedy method; It presented in this section that can apply to a large category of problems. It describes each small group that operates on those constraints as a feasible solution. The challenge is finding a reasonable solution that minimizes or maximizes a presented goal of the function. It is called the optimal solution that is created by this technique. There a simple process for determining a feasible solution; however, not an optimal solution. The greedy method proposes an algorithm that can operate in steps, regarding the individual input in time. Through every level, they present a decision concerning whether specific information means within the optimal solution. This is prepared arranging inputs into an index through any chosen technique. If a modulation to the input for the next step within the former optimal solution determination appears among the infeasible solutions, it does not append it into a sub-solution. The choice technique itself depends on any optimization criteria. The criteria may not or may be goal function. Many standards of optimization may be probable for a presented problem. However, most of these algorithms produce sub-optimal solutions [139]. During combinatorial optimization, several algorithms produce a solution space, while at every step, a unique space collection component is appended on the partial solution in the construction. They achieve the requested element available also expressed with f the set from each available component each time. For each collection from the candidate components f shall have higher than individual component, the algorithm produced f for building a solution shall have a process for picking the next space regarding on a developed solution F in the construction stage. Between all feasible elements that yet not selected, a greedy algorithm picks the least cost for minimization. Figure 3.4 presents a greedy algorithm pseudo-code. The solution f built by admitting its cost $f(f)$ are beginning from \emptyset and 0 serially, during steps 1 and 2. While step 3, initializes applicant elements between all elements in the space. A development to the solution prepares within the loop of while in steps 4 to 9, finishing if f is empty. In step 5, it picks the least cost form the ground collection component i^* . After that, in steps 6 and 7, each solution in the construction stage, including the cost of updates from the statement as the embodiment of i^* in the solution under construction. In step 8, update the collection of elements in f , observing in i^* , and it's now a member of solution S . In step 10 it returns the cost and solution S . The minimization of the problem presented in Figure

3.4. The argmin within step 5 replaced to argmax in case of maximization, which chooses a candidate component of maximum cost. The following display instances of greedy algorithms during any combinatorial optimization problems.

The greedy method used in optimization problems which include searching through the collection of arrangements for getting an individual which maximizes or minimizes the objective function represented at those arrangements. In progression for solving a presented optimization problem, solution progresses through a series of opportunities. This series begins with any well-defined beginning arrangement and later performs choices which looks the greatest which are achievable by the greedy method not driven on the optimal solution. However, there are various difficulties which it takes the trial to, including so they suppose problems to hold the greedy-pick quality. That is the quality that an optimal global arrangement able to achieve through a list of optimal decisions (that is, decisions which are the greatest from between the opportunities possible to the point), beginning of a well-comprehend arrangement [140].

```

Begin Greedy:
1  $f \leftarrow \emptyset$ ;
2  $f(f) \leftarrow 0$ ;
3  $f \leftarrow \{i \in E: f \cup \{i\} \text{ is not infeasible}\}$ ;
4 while  $f \neq \emptyset$  do
5    $i^* \leftarrow \operatorname{argmin} \{c_i: i \in f\}$ ;
6    $f \leftarrow f \cup \{i^*\}$ ;
7    $f(f) \leftarrow f(f) + c_{i^*}$ ;
8    $f \leftarrow \{i \in f \setminus \{i^*\}: f \cup \{i\} \text{ is not infeasible}\}$ ;
9 end while;
10 return  $f, f(f)$ ;
End Greedy.

```

Figure 3. 4 Pseudo-code of a greedy algorithm for a minimization problem

3.3.1 ELEMENTS OF THE GREEDY STRATEGY

Generating the series of options of the problem that obtains an optimal solution is a technique used by Greedy search. The execution of the selection part at each opportunity picks a better alternative in the current state. The optimal solution is not obtained always by a heuristic approach. In this part, we present several characteristics from greedy methods.

The technique for developing a greedy algorithm can expresse within the following levels:

1. Define some optimal substructure of the problem.
2. Enhance the recursive solution.
3. While presenting the greedy selection, after that only individual sub-problem remains.
4. Show that this selection preserved for making the greedy decision. (Steps 3 and 4 can happen during each series.)
5. Enhance a recursive algorithm which performs the greedy approach.
6. Switch this recursive algorithm into the solution algorithm.

For instance, during the activity-selection problem, first set that S_{ij} is a part of the problem, wherever both i and j are diverse. After having discovered which performed the greedy selection, they could check those parts of difficulties to the act of the frame S_k . Optionally, they could hold the optimal substructure among a greedy range in memory, after that the selection moves just single sub-problem for solving. Later, they should own established that a greedy selection (the initial activity S_1 to the end in S_k), joined among an optimal solution. Further, usually, greedy algorithms approving on the subsequent steps:

1. Calculate the optimization problem within which they get a selection also moved by individual subproblem for solving.
2. Explain that there is permanently the optimal solution for that original problem that produces the greedy selection so that the greedy selection is forever protected.
3. Express optimal sub-structure through explaining that, should have present the greedy selection, anything remains is a subproblem including the characteristic that if they join an optimal solution on the sub-problem including the greedy range that has performed, the report through an optimal solution on the initial problem.

The primary component is the greedy-selection characteristic: that ability to construct a globally optimal solution through producing locally optimal (greedy) selections. In another term, if they regard that selection for creating, the choice which seems best for the current problem, without concerning the effects of sub-problems. Here is where greedy algorithms differ from dynamic programming. In dynamic programming, they

get a selection in every round, just some selection regularly based on the solutions to subproblems. In a greedy algorithm, they execute whatever collection looks greatest at the time also later determine the subproblem which remains. The selection produced through a greedy algorithm shall base on choices so far, just it not able to based on several prospective preferences or at the clarifications to subproblems.

3.3.2 OPTIMAL SUBSTRUCTURE

A problem presents optimal substructure while the optimal solution to the problem includes inside the optimal solutions through subproblems. The characteristic does a principal component during testing the applicability to dynamic programming considering greedy algorithms. While an instance from the optimal substructure, remember what we showed in the previous section which an optimal solution to subproblem S_{ij} involves action a_k , later that should also include optimal solutions on the subproblems S_{ik} and S_{kj} . Presented the optimal substructure, it explained that when they knew which the action that uses as a_k , they could build the optimal solution on S_{ij} through picking a_k including every step into optimal solutions on the subproblems S_{ik} and S_{kj} .

An extra straightforward strategy can be applied concerning optimal substructure if using it on greedy algorithms. While discussed earlier, it has the benefit of considering sub-problems becoming established as the greedy selection in the initial problem. Each requirement shows that an optimal solution on the subproblem joined among the greedy choice executed, results in an optimal solution for the original problem. The design uses inference upon those subproblems to show that presenting the greedy opportunity in each step provides an optimal solution [141].

3.4 BEE ALGORITHMS

Real systems inform us that the individual organisms which elementary systems are ready to execute tasks which are complex through dynamic connections. The Bees' Algorithm (BA) simulates some food foraging presented by colonies from honey Bees. The simulated Bee colony performs partly alike, also partly oppositely from Bee colonies within life. BA used for explaining and simulating deterministic combinatorial and practical optimization problems.

3.4.1 BEES IN NATURE

Behavior of colonies of insects like ants and bees are recognized to be swarm intelligence [142, 143]. This extremely coordinated operation allows those colonies of insects for solving problems exceeding the capability from different fragments through running collectively, including communicating primitively amongst elements of the group. In a honey bee colony, for instance, this model provides bees for investigating the situation in exploration from flower groups (source of food). This exploration includes next designating the specific conditions of food discovered by different bees of the colony. Such a colony described with self-organization, robustness, and adaptiveness [144]. Bees depend on self-organization on comparatively simple habits of a singular insect's role. Continuation for a vast number of various standard insect classes and change within their behavioural models that is reasonable for expressing singular insects' as intelligent from implementing the modification to complicated jobs [122]. The excellent instance holds the nectar operating [145, 143].

3.4.1.1 BEHAVIOUR OF REAL BEES

A colony of honey bees can spread itself across large ranges (larger than 10 km) also within various ways concurrently for appropriating a vast number of food sources [105, 146]. The benefits of the colony by expanding its foragers to desirable areas [130].

The Honey Bee colony picks from the range of search which is useful with several nectar references possible. Earlier investigations should explain that the colony immediately also accurately sets the model for searching within space and time the following development of nectar references. They depend on the bees self-organization on several comparatively easy habits from different insect performance [147]. It is reasonable for supposing a colony, primarily the system of reacting individuals-foraging Bees [148]. During the light, this is reasonable for a first check some important performance from the individuals also later shared this information between those individuals into the organization for achieving general knowledge. "Collective - Swarm intelligence" forms the developing characteristics from the colony about individuals. The information exchange between individuals is the most significant experience during the development of accumulated knowledge. During searching a whole hive, this is reasonable for distinguishing any components which usually exist within every hive. The most significant element from the hive, including regard for

interchanging knowledge, is the dancing period. Contact between Bees associated with the food sources quality happens during the dancing period. It is called the waggle dance [149].

Usually, in a typical insect colony, individuals typically do not accomplish every task. An individual concentrates on a collection of functions according to chance, morphology, or age [121]. An essential component of the whole bee colony is the foragers [149].

3.4.2 ARTIFICIAL BEES

To simulate the communication between the bees, a definition of the performance of the artificial Bees (agents) is needed. In the process above, several scenarios, including several outlines, can be defined for simulation [153]. In social insects, the most activities are about seeking the source of food. It is known that honey Bees "normally spend the last part of their life collecting food" [149]. They "consume a substantial part of their life span knowledge and developing their foraging experiences"[149]. Each Bee colony holds scouts who are the colony's founders [149].

The Bees are searching for food source without any guidance. They are interested in finding different types of the food source. While a consequence of performance, any scouts recognize through the expenses of search and the quality of food source. Infrequently, some scouts may find the food source accidentally, outside food sources. Some scouts trying to solve the challenging combinatorial optimization problems have the task of quick identification from some set of solutions. Any of these solutions on specific challenging combinatorial optimization problems could later confirm to get answers of good quality [153].

The association among the insects reduces the cost of foragers while getting a new source of food. It implies that the association among artificial Bees should further provide quick detection of some solution [153]. The quality of the food source that was found by foragers can be increased by the cooperation of the Bees. This signifies that this help should further assist us in getting better solutions from the hard combinatorial optimization problems.

3.4.3 BEE ALGORITHM

One of the optimization algorithms is a Bee algorithm that relies on the natural behavior of a Bee-inspired population to find the optimal solution [149]. A pseudo-code of Bee algorithm shown in Figure 3.5 in the simple style and Figure 3.9 show the flowchart of the algorithm. The algorithm needs several variables to set, specifically:

- The scout Bees number is (n)
- The picked sites (m) out of the visit site (n)
- The most significant site (e) out of selected sites (m)
- From the most excellent site (e) the number of recruited Bee (nep)
- From the picked site (nsp) the number of Bees that recruited from other ($m-e$) sites.
- Set the initial size of patches (ngh) that include site and its stopping criterion and neighbourhood.

Input: n = scout bee, m = selected sites, e = best of m , nep = bees recruited for e , nsp = bees recruited for $m-e$, ngh = patch size and stop criterion

Output: optimal solution(s)

1. Start a population n with arbitrary solutions.
2. Estimate the fitness of the population n .
3. Loop (stopping criterion not met) //Creating new population.
4. Picked sites to neighborhood exploration m .
5. Recruit bees for picked sites nsp (more bees nep for greatest e sites) and estimate the fitnesses.
6. Pick the appropriate bee from each patch ngh .
7. Allow remaining bees ($n-m$) to explore randomly and estimate their fitnesses.
8. End loop.

Figure 3.5 Pseudo code of the basic bees algorithm

Starting the algorithm by scout Bees (n) located randomly within the search area. The suitability of the positions sensed through scout Bees estimated in step 2. During step 4, Bees that produce the greatest fitnesses accepted necessarily “picked Bees” also site detected by them continue picked during neighbourhood exploration. Next, in steps 5 and 6, the algorithm manages explorations into a neighbourhood from the picked localities, allowing extra Bees to explore neighbourhood on some most significant (e) localities. Alternatively, the fitness advantages used to restrict the possibility of the Bees are picked. It presents explorations into a neighbourhood from the valid e site that describes encouraging solutions further described with raising extra Bees to support them than some other picked Bees. Concurrently among scouting, this differential recruitment is a fundamental process of the Bees Algorithm. In step 6, to any patch, just the Bee among that greatest fitness determination picked to make the subsequent Bee population. In reality, there is never such a limitation. The limitation

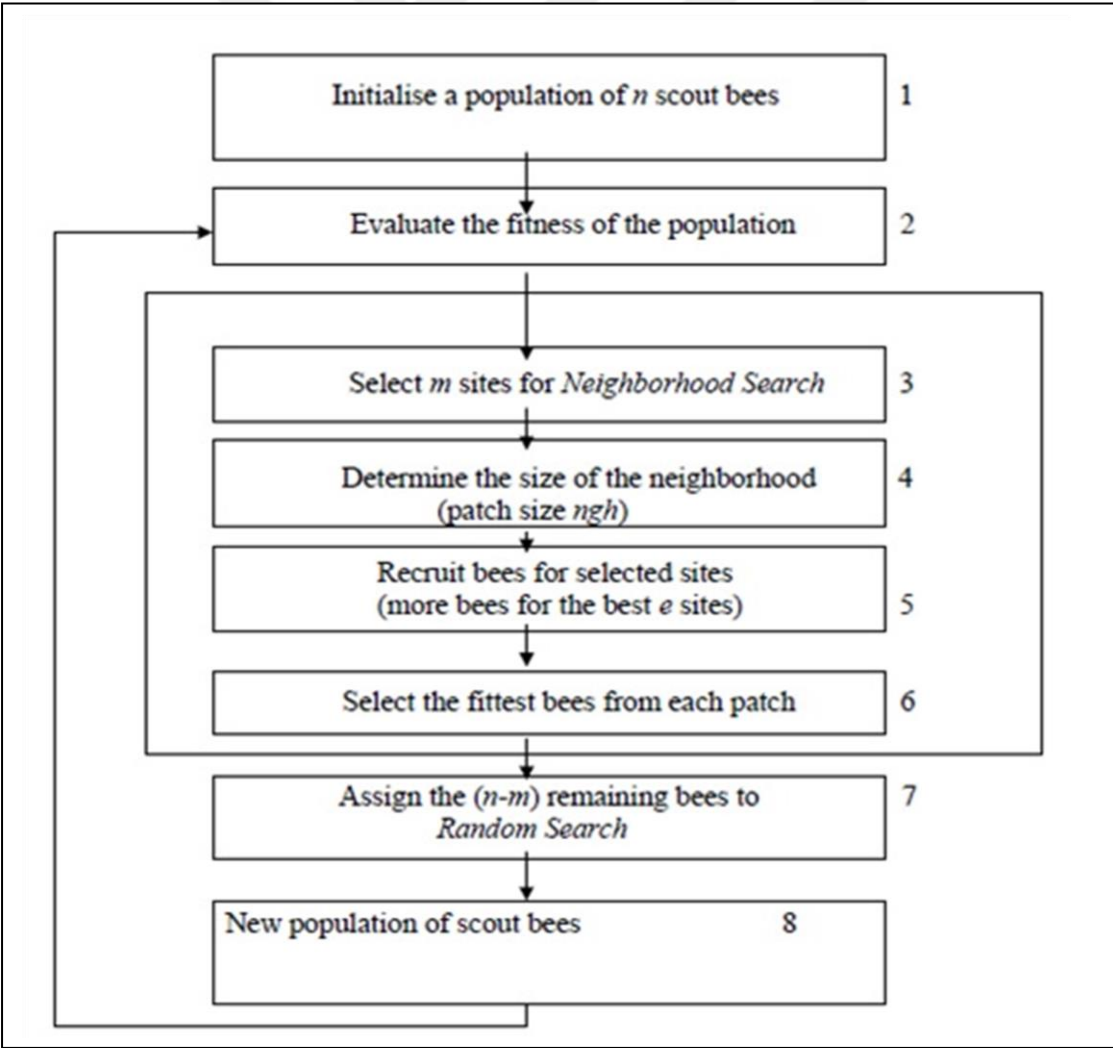


Figure 3.6 The Bees Algorithm Flowchart

means here to decrease some amount of investigated circumstances to happen. In step 7, it selects the Bee remaining within the population randomly nearby the search area scouting to different solutions. It iterates those levels until they satisfy a termination criterion. By the completion from every repetition, the colony mind has a couple of pieces on its new population—delegates of each chosen piece and another scout Bees allowed to transfer arbitrary searches [161]. One method for multi-aim optimization is the Bee algorithm, the purpose of the algorithms to optimize multi-dimensional combinatorial functions. Every function described by a particular collection of parameters that each want to remain encoded under this “Bee” [154].

3.5 NATURE INSPIRED ELEPHANT SWARM WATER SEARCH ALGORITHM

3.5.1 ELEPHANT IN NATURE

We comprehend that elephants signify the biggest mammal at on earth. They continue within a group that really enormously improves also require a high level of communication between the individuals. They live in a community that identifies since "fluid-fission-fusion" community. They need an excellent level for cooperation, within the complicated connections they develop among different individuals. They describe a collective group between the elephants through some closeness and familiarity. A family-unit is one of the common important group that includes two mentioned females at least in the family (See Figure 3.7 below). It does not allow the Males to be a part of the family, except the Male connect to the female of a live individual. The typical family based on 10-50 elephants also the connection between it well-established even coordination habits. The communication incorporates family care, cooperation, support retrieval, and group protection, plus everything involves decision-making that created with the powerful “matriarch”. The matriarch is the principal female lead of the group who is the smartest and the oldest and therefore the multiple qualified. She performs each choice that covers protection, mobility, plus support recovery [155].



Figure 3.7 Group exhibitions in Elephants clan

3.5.1.1 ALTRUISM

Inside the group of families, there is a high level of generosity plus collaboration that associates each family member. Based on the analysis that a member of the family will support another family member for increasing its lifetime, raise the number progeny, and they support for maximizing the member's of the family gene present the generation, however, only they make this in charge of their durability. Elephants sense the right attention on everything in the group about the members. Also, they are raising one emotional, and sensitive response is that combining familial caring [155].

3.5.1.2 INTERACTION

The elephants have several socio styles, one of the essential forms is the interaction, and it's a similar thing through using the skills of the sense. The communication is most famous for it makes a bunch for staying the eye set on protective regions, for managing and understanding their families also establishing their proactive events and make females for going among a young staff previous to weaning [155].

3.5.2 SOCIAL BEHAVIOR AND INTERACTION IN ELEPHANTS

3.5.2.1 ACOUSTIC INTERACTION

It identifies a communication that as audio or acoustic interaction. Elephants release a sound extending a broad area in different intensities organizing an activity, knowing their requirements, bringing the friends, etc.

3.5.2.2 ANGRY ELEPHANTS TRUMPETING & RUMBLING

Some scientists observe that elephants provide the sound from 1 to 20 HZ infrasonic sound which is the out of a range of hearing capacity for human, also able to move across long-distances. Moreover, seismic flags, such as a tiny earthquake, releases every elephant to place themselves in association on their private place [156].

3.5.2.3 CHEMICAL INTERACTION

Some chemical interaction is including one of the significant interactions into the elephant; it is a qualified procedure plus includes the flow of chemical flags that are long-lasting. They provide a universal smell flag, including the smells that are carried through transferring various sources, similar generative region, surface glands, expired air and face.

3.5.2.4 TACTILE INTERACTION

Touching is another interaction that is used by elephants as the form of interaction among them within trunks; also, this acknowledged like physical interaction. It uses the elephant's trunks during different celebrations the same smelling, drinking, tearing vegetation of trees, just for extra physical sense. The trunks used by elephants to study new things and replacing the touches among unknown ones crossing into the forest, in respect on the replication method all communicate through twisting their trunks.

3.5.2.5 MEMORY AND RECOGNITION

In a community under which live elephants, requires a great mind also excellent identification skills. Because they are a regular group and they reach different individuals of the group at a regular basis. So, they need to distinguish the family from

the non-family; two individuals who left during 23 years rejoined after 23 years also observed by Carol Buckley, presented in [156]. Elephants also hold a great mind including an improved cerebral cortex that allows them for achieving a high potential to detect and keep such knowledge to the highest continuance of time, asserted by Granli, Poole in [157].

The essential part of discovery and remembrance within elephants means to stay apart from inbreeding desperation of proceeding. Elephants can change among hereditary kin of non-kin using a phenotype comparable on their smelling sense, while the procedure is yet undiscovered.

Elephants own a great feeling of smelling they only able to stay the record from the group organs through just smelling their urine also it permits them to construct rational plans of their specific situation, explained in [158].

Not just their feeling produces identification also an excellent memory without a group section plus instructs the newborn family in the group. The less family sees their former one's parents and sisters, receive of them how to get food and water. They practice their bodies and recognize it within the land to discover the water.

See the response from the adult also their response concerning several individuals. Through their adulthood, they keep their knowledge as a guide. The list in their youth which explains that they own prepared great minds.

Their communicative group will not be as large as now including interest on their communication if they don't hold an important sensing method of smell, touch, etc. they also receive through communicating among the various individuals and hold the capacity to receive and identify important ideas to a long-time season which continues to their intelligence and communicative group.

3.5.3 ELEPHANT SWARM WATER SEARCH ALGORITHM

Metaheuristics are nature-inspired algorithms for finding approximate solutions to some computationally hard optimization problems. Swarming behaviours of animals including; Firefly-BAT, Cuckoo, ant, pigeon, fish, Bee, ...etc, have been used in metaheuristics [159]. Some properties behind the metaheuristics include; homogeneity, adaptability, illation-free tools plus local optima eschewal ability [160]. An interesting example is the swarm behaviour of the biggest terrestrial mammals, elephants. The

trunk is the typical representative characteristic of an elephant which is multi-objective, like respiration, following things and uplift water [161]. Swarm properties of water search of elephant herds have been utilized to define metaheuristic algorithms [161]. The following four idealizing assumptions employed for describing the proposed method [162]. (i) Elephants walk nearby in exploration for water through dryness within various groups; this act is named elephant swarm. Every group operates concurrently for obtaining water. The leader from every organization (elderly elephant, matriarch) is qualified for using a choice for searching the most significant water source. (ii) While the elephant group discovers any resource of water, the matriarch shares with the nearby groups, the information about the quality and quantity of the resource. Good water level indicates the next valid move. (iii) Elephants hold pretty strong memory. Several elephant groups can retain information about some correct positions of the water supply that existed and recognized through its private group (local best solution). They can also remember the exact location of the best water source, that found out through the entire flock of groups (global best solution). (iv) Local and global water exploration choices represented through a probabilistic constant P . Based on this value, the matriarch opts actions for switching between global and local search options. Because of certain physical and natural factors, water exploration in local may have a higher P value [162]. The elephant can distinguish and learn among several visual also some acoustic signals of discrimination. Several techniques including; acoustic, seismic, and chemical communications are used for communication among elephant groups in long-distance up to 10–12 km away.

The d -dimensional optimization problem can formulate using the location and velocity of the i th elephant group from a swarm (Composed of N members). In t^{th} iteration, the location can represent by $X_{i,d}^t = (X_{i1}, X_{i2}, \dots, X_{id})$. Similarly, the velocity can express using $V_{i,d}^t = (V_{i1}, V_{i2}, \dots, V_{id})$. Based on these, the best local solution for i th elephant group at the current iteration expressed as $P_{best\ i,d}^t = (P_{i1}, P_{i2}, \dots, P_{id})$ and G_{best} expresses the best global solution $G_{best\ i,d}^t = (G_1, G_2, \dots, G_d)$. The starting velocity and position of elephant groups are arbitrarily assigned within the exploration area. During iteration, the positions and velocities of the elephants renewed. Optimal water search decision actions should occur in both global and local scales. While iteration proceeds, the velocities from the members are renewed based on several techniques

during local and global search according to the equations (3-10) and (3-11) below. The value of switching probability p determines the type of search:

$$V_{i,d}^{t+1} = V_{i,d}^t w^t + rand(1, d) \cdot (G_{best,d}^t - X_{i,d}^t) \quad \text{Equation 3-10}$$

If $rand > p$ [global search]

$$V_{i,d}^{t+1} = V_{i,d}^t w^t + rand(1, d) \cdot (P_{best,d}^t - X_{i,d}^t) \quad \text{Equation 3-11}$$

If $rand \leq p$ [local search]

In Equations (3-10 and 3-11), $rand$ is a value that produces a d -dimensional array of random values in $[0,1]$. (.) Expresses element by element multiplication and w^t is the weight of inertia for compromising exploitation and exploration throughout the current iteration. Next, the location of the elephant group is adjusted as specified by the following formula.

$$X_{i,d}^{t+1} = V_{i,d}^{t+1} + X_{i,d}^t \quad \text{Equation 3-12}$$

In Equation 3-12, t_{max} , X_{max} , and X_{min} indicate the maximum iteration number, lower and upper limits regarding positions. A search route is affected by three elements specifically: current velocity ($V_{i,d}^t$), current particle memory commands ($P_{best,d}^t$) and swarm memory commands ($G_{best,d}^t$) [162]. Nevertheless, in ESWSA, the new search route is determined through both current speed and current elephant memory and swarm memory effects. In the global search, the velocity update based on the elephant's best position, and the search continues to obtain the best global solution. In the case of Random Inertia Weight (RIW) [163], the weight of inertia values is chosen randomly, which is extremely valuable for a dynamic system that tries to obtain the optima. The following formula is used to select the weight of inertia in RIW:

$$w^t = 0.5 + (rand * 0.5) \quad \text{Equation 3-13}$$

In Equation 3-13, $rand$ is a uniform random number in $[0,1]$. A successful procedure is the Linearly Decreasing Inertia Weight (LDIW) [161]. This procedure can be used in developing some good tuning properties concerning the optimization. In LDIW, the weight of inertia values depends on the value (w_{max}) and an ultimate small value (w_{min}), according to the following Equation [163]:

$$w^t = w_{max} - \left\{ \frac{w_{max} - w_{min}}{t_{max}} \right\} * t \quad \text{Equation 3-14}$$

where the index of iteration is t , and the maximum number of iterations is t_{max} .

It should note that the PSO approach uses the random repair technique, which involves jumping randomly within the search space, while in ESWSA, the position change based on Equation 3-12.

3.6 METHODOLOGY

The research developed within this thesis concentrates on approaching score and search-based techniques to learn the structure of Bayesian networks from data. The thesis proposes novel algorithms and approaches to establishing Bayesian network structure learning. In this section, we present the Six methods for structure learning Bayesian network:

3.6.1 FIRST PROPOSED METHOD

In this part, we present the novel approach through implementing the Pigeon Inspired Optimization (PIO) for structure learning Bayesian network. The proposed method uses PIO approach as a search method for structural learning of Bayesian networks. The BDeu metric used as a score function for measuring the Bayesian network structure. The PIO algorithm is effectively an iterated procedure that consists of a population of individuals where every pigeon encodes a potential position and velocity in a given space. This space held to be the search space. The proposed method based on two techniques. The first technique uses the map and compass operator model (discussed in section 3.1.3.1) for local search through the necessary process. It uses the first technique map and compass operator model for local search within the specified method. The second technique uses a landmark operator model (discussed in section 3.1.3.2) as a global search. Figure 3.8 shows the pseudo-code of this technique. Once the Pigeons operate, they can use the solutions to their local optimum by utilization of a local search method. PIO algorithm's solution development uses another neighbourhood than a local search. The expectation that local search updates a solution produced by a Pigeon is high. The Bayesian network structure learning solution area is the form for each potential DAGs. A pigeon later examines the exploration area for finding the approximately near-optimal or optimal solution, which is known as the BDeu score metric. Equation (2-47) used to calculate the BDeu score as the goal function of the optimization. The exploration aims for obtaining a higher BDeu score for the network structure. All initial solutions produced through iterative

operations. The detailed implementation procedure of PIO for structure learning Bayesian network can represent as follows:

Step 1: according to environmental modelling, starting with an empty structure.

Step 2: initialize parameters of PIO algorithm, such as solution area dimension D , the population size N_p , map and compass factor R , the number of iteration $N_{c1 \max}$ and $N_{c2 \max}$ for two operators, including $N_{c2 \max} > N_{c1 \max}$.

Step 3: establish all pigeons by a randomized velocity and path. Comparing the fitness (BDeu score function) from every pigeon, also discover the current best position (location).

Step 4: execute a map and compass operator. At the first time, update the velocity and path of every pigeon by using Equations 3-1 and 3-2. Then compare all the pigeons' fitness (BDeu score function) and get the new best position.

Step 5: if $N_c > N_{c1 \max}$, stop the map and compass operator and operate the next operator. Otherwise, go to Step 4.

Step 6: order all pigeons according to their DBeu score values. Half of the pigeons whose fitness is low will follow those pigeons with high fitness according to Equation 3-3 We then find the middle from all pigeons according to Equation 3-5, including this centre does the desired goal. Every pigeon will fly to the target by setting its flying path according to Equation 3-4. Next, put the best solution parameters and the best cost value.

Step 7: if $N_c > N_{c2 \max}$, stop the landmark operator, and output the results. If not, go to Step 6.

Starting with a blank graph (G_0), the arcs are appended one after another, provided that they not included in the current graph solution. If the new solution score function is higher than the current result, the new solution also satisfies the DAG constraint.

This process continues until the quantity of the arcs equals the quantity defined in advance. In the model, the solution starts assigning a population for each operator and picks the solution, which has a higher score function. Pigeon continues according to the selected operator until the process has performed a maximum number of iterations or the BDeu score not increased any more. Typically, the methods hold four separate operations in local optimization: Deletion, Addition, Reversion, Movement. The first three are simple operations within this domain, involve just replacing an individual

Algorithm PIOSB (pigeon inspired optimization for structure learning of Bayesian network)

INPUT: - datasets

N_p : number of individuals in pigeon swarm

D : dimension of the search space

R : the map and compass factor Search range: the borders of the search space

N_{c1max} : the maximum number of generations that the map and compass operation carried out

N_{c2max} : the maximum number of generations that the landmark operation carried out.

OUTPUT: - learning and constructed BN

1. The initialized empty structure and initialize parameters of PIO algorithm (space dimension D , the population size N_p , map and compass factor R , the number of iteration N_{c1max} and N_{c2max} for two operators, and $N_{c2max} > N_{c1max}$).
2. Set each pigeon with a randomized velocity and position. Comparing the BDeu score function of each pigeon, and find the current best position.
3. Operate map and compass operator. Firstly, we update the velocity and position of every pigeon by using Equations (3-1) and (3-2).
4. Compare all the pigeons' fitness and find the new best position, by comparing the BDe score function of each pigeon.
5. If $N_{c2} > N_{c1max}$, stop the map and compass operator and the operate next operator. Otherwise, go to Step 3.
6. Rank all pigeons according to their fitness values. Half of the pigeons whose score function is low will follow those pigeons with a high score according to Equation (3-3).
 - (1) We then find the center of all pigeons according to Equation (3-5), and this center is the desired destination. All pigeons will fly to the destination by adjusting their flying direction according to Equation (3-4).
 - (2) Next, store the best solution parameters and the best score value.
 - (3) If $N_{c1} > N_{c2max}$, stop the landmark operator, and output the results. If not, go to Step 5.
7. Return the maximum BDe score.

Figure 3.8 Pseudo Code of The PIO for Structure Learning Bayesian Network.

edge every time from a competitor solution. It allows the inclusion of a comparatively small area near the solution. With every movement operation, on the other hand, the existing edges change the set of parents which can make a moderately significant modification for the current solution. Therefore, if the solution not changed after applying simple operators, the move operator may improve it. Flying is the primary

force utilizing the chosen operation in local optimization, which grows further widespread while a pigeon approaches the desirable solution. Flying directions, the switch with various local optimization operators, expands extra prevalent as a pigeon flies continuously near a solution through exploration toward a better one. Therefore, the current velocity renewed following either pigeon's best global or best local solution. The velocity of pigeon is regenerated based on the current best position of the pigeon in the local search. On the other hand, the global velocity depends on the best global solution concerning pigeon in a global search, near a global best position.

As shown in Fig.3.9, Pigeon G0, which describes a DAG with arcs, tries reversion, move, addition, and deletion, and reaches new solutions G1, G2, G3, and G4, respectively. Assuming the best score is in G3, it will select, and the pigeon will proceed to examine some similar process to get G+3 as the new solution. If the BDeu score of G+3 is higher than that of G+1, it will continue to perform the corresponding operator. The operations will repeat until the BDeu score stabilizes, or iteration loop reaches the maximum. In the process, they have performed operators, i.e., $m = N_{c2}$ set of starting points in the search space. Addition, deletion, reversion, and move operators are four competitor directions for any pigeon to select the Map and compass operator pigeon attempts to achieve.

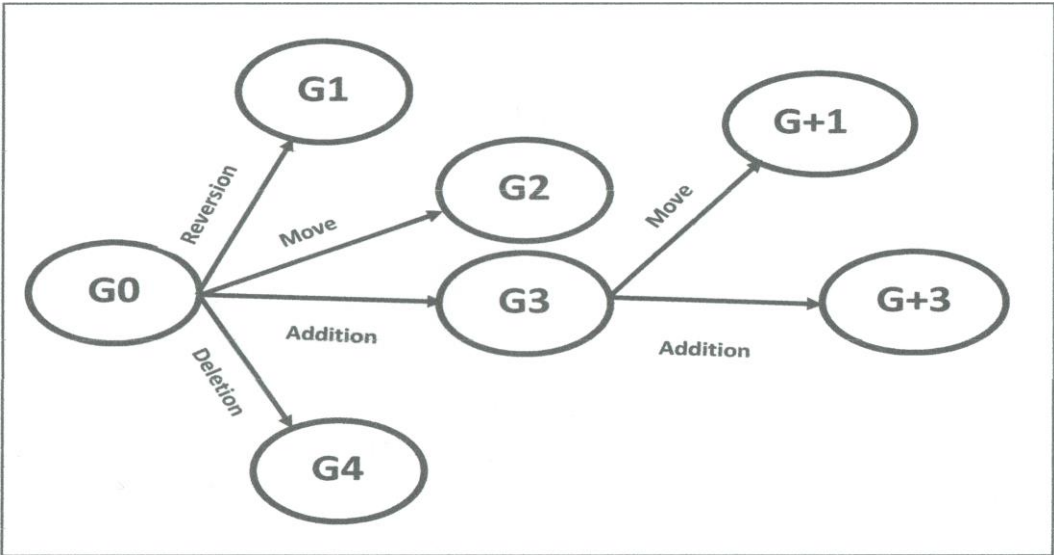


Figure 3.9 Map and compass steps for one Pigeon

3.6.2 SECOND PROPOSED A METHOD

Our second proposed method develops a new algorithm for learning Bayesian network structure based on enhanced Artificial Bee optimization algorithm. All the forms that are required to solve the learning problem using this meta-heuristic described in section (3-4). The Bees algorithm is a repeated procedure that consists of a population of individuals, anyone recognized as a “Bee”. Each Bee encodes a possible solution in a given problem space. This space referred to as the search space, which includes every explication on the problem at hand.

Generally, the Bees algorithm applied to spaces that are too large to be exhaustively searched (such as those in combinatorial optimization). Solutions to a problem encoded in most forms due to certain computational advantages associated with each challenge.

As has been previously addressed, one of the fundamental attributes of Bees algorithm is its ability to search the function space from multiple points in parallel. In this context, parallelism does not refer to the ability to parallelize the implementation of the Bees algorithm; instead, it relates to the ability to represent a vast number of potential solutions in the population of a single generation.

In each production, while some execute from the Bees algorithm, a population of solutions survives. The search of the function area proceeds out of these positions; Bees interpret those circumstances in parallel. It is in the distinction between other searching methods such as hill-climbing, in which it uses an individual element in the function space as the basis of the search. We term the capability to explore various arrangements for each solution as absolute parallelism.

Parallelism is desirable for the population to concentrate on comparable solutions. Once the population has joined, the facility is very limited during arbitrary exploration technique to help in investigating extra pieces of the function space.

Bees algorithm uses local searches through a necessary rule. Once some Bees should realize their solution structure, the solutions can continue on the local optimum through utilization from a regular local search. In Bees algorithm, there is a pair of significant selection settlements during those foraging procedures:

- Selection from a waggle-dancer: the collection executed a purpose of scout-Bees' fitnesses essentially while picking a waggle-dancer knowing a prosperous exploration area to use is complete.
- Selection from representative-Bee during every patch: while preparing a representative- Bee that recognizing either a famous top state from maximization problems space or a lowering into this state of minimization. That achieved the area they should declare which through the preferred waggle-dancer.

The goal from Bees' Simulated Annealing algorithm BSA is to utilize a standard annealing procedure through some representative-Bee preferred settlement to escort an exploration rule towards a higher optimal solution space, to provide to explore condensation to occur.

Through forming the cooling program of simulated annealing, BSA practitioner can apply authority covering the application. Bees algorithm employees no such thought about cooling, also its concentration is not easily controlled.

Local search algorithms experience the difficulty of getting better initial solutions, the artificial Bees provided these solutions. For example, simulated annealing uses a time-consuming method to pause when some cooling action until the distributed balance from states is reached. In particular, simulated annealing does not understand whether the area we have investigated is a section in the arrangement location or whether a field is an excellent site for searching.

Our proposed method improves Bee optimization by the intermingling of search characteristics from BA and SA within unique global principles named Bees' Simulated Annealing (BSA). It supposes the approach to be extra robust and also to offer a good experience as a search algorithm. It uses BDeu as a metric to measure the score function. The Bees algorithm is a repeated process, and it composes a population of individuals, all bees encoding a potential solution inside a presented area. They regard the area primarily as the search area. Usually, the Bees algorithm used in areas that is big to be exhaustively explored (such as those in combinatorial optimization). Solutions to a problem are held encoded in significant patterns for specific computational benefits concerning each problem. The proposed approach uses two levels. A primary level applies Bee's algorithm to local search as the necessary method. Another level uses Simulated Annealing for global search. The Pseudo-code shows

this technique in Figure 3.10. The proposed process needs the parameters from the Bees algorithm, in addition to the parameters of simulated annealing. The parameters required by BSA are:

- The number of scout Bees (n)
- The number of sites selected out of n visited locations (m)
- The number of best sites out of m selected sites (e)
- The number of Bees recruited for best e sites (nep)
- The number of Bees recruited for the other ($m-e$) selected sites (nsp)
- The initial size of patches (ngh) which includes a site and its neighbourhood
- The initial temperature T_0
- A reduction function for decreasing the value of T at every iteration and stopping criterion.

BSA is initialized by an assigned population n , within the starting area, computes every n as scout Bees. Scout Bees are randomly assigned to the search area. The beginning temperature T_0 initialized in step 1. Figure 3.10 shows the BSA algorithm.

The BSA calculates the fitness of the sites (i.e. the achievement from those applicant solutions) sensed through some scout bees in step 2. The m sites that describe encouraging details within a search space are selected as “picked sites” and taken to neighbourhood search in step 4.

In Step 5, BSA manages explorations in the neighbourhood of the picked sites, distributing extra Bees to explore near to the best e sites. The Bees can select immediately according to the fitnesses among the situations they are visiting. During step 6, behind some property of recruited individuals has calculated by using the fitness function, each decision whether to use executives on such individuals and manage it in this population or permit it to be substituted.

BSA offers to balance the exploratory nature of the current Bees algorithm implementation with search intensification by using annealing approach to choose representative-Bee for each patch.

It is preferably allowing just (in the time) locally more significant recruited-Bee through the recruitment procedure. The fresh representative-Bee allowed just while it

passes through a 'fitter' solution space (fitter than the waggle-dancer that recruited), unless, in the state from no fitter Bee determined. This algorithm shall allow the highest number of recruited Bees yet while its development on a few optimal solution spaces (maybe near to a globally better optimal) probabilistically according to a function for the waggle-dancer. The possibility that just local extremum (local top or down) all times is less than 1. In this section, the recommended annealing function will introduce exploration intensification to the Bees Algorithm.

The probability for getting higher marks in the exploration space is higher in the search's starting (foraging) procedure also reduces while reducing a temperature. The outstanding bees in the population distributed randomly throughout the search space scouting for different solutions in step 7. That is the important characteristic of the Bees algorithm to leave local optimum.

Algorithm BSA (*Bee algorithms is local and Simulated Annealing is global search (hybrid bee and simulated annealing algorithms)*)

INPUT: - datasets

OUTPUT: - Learned and constructed BN

- 1-The initial temperature T_0 , Initialize population n with random solutions.
- 2-Evaluate the fitness of the sites (i.e. the performance of the candidate solutions) visited by the scout bees.
- 3- loop until less than stopping conditional: -
 - 3-1 chooses the site solution and evaluates the fitness (Select sites m for neighborhood search., Recruit bees for selected sites (more bees for best e sites) and evaluate fitness's.).
 - 3-2 for loop, compare the best-recruited bee y_j with the bee recruited it x_j If $\text{fitness}(y_j) - \text{fitness}(x_j) < 0$ then $x_j = y_j$
 - 3-3-the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. It is the key feature of Bees algorithm to escape local optimum.
- 4- if $\exp[-(\text{fitness}(y_j) - \text{fitness}(x_j))/T] > \text{random}[0,1]$ then $x_j = y_j$
- 5-the temperature is reduced by a small amount Δt using the decrement functions
- 6-Return the maximum score function for BDeu
- 7- New population with scout-bees and score function.

Figure 3.10 Pseudo code of BSA hybrid bee local and simulated annealing global search.

The colony produces a couple of parts on its current population: representatives of the chosen patches, and scout Bees distributed to transfer random searches at the end of each iteration. Later, the temperature is decreased by a little amount Δt using the decrement function. By harmonizing the decrement amount Δt , the concentration for the search rule is established. The algorithm iterates those steps until they satisfy a check criterion.

3.6.3 THIRD PROPOSED METHOD

In the previous part, we presented a technique as the structure learning Bayesian network using the Bee as a local search and simulated annealing as a global search. In this section, we present another different method which also depends on simulated annealing and Bee, but different from the previous technique. The name the proposed method (SAB) uses Simulated Annealing as local search and Bee as global search further it uses BDeu as score function. Figure 3.11 shows the pseudo-code for the proposed algorithm. As has been earlier discussed, one of the significant properties of the Bees algorithm is its capability to explore the function area of various points in correspondence. Within these circumstances; parallelism seems no regard to the powers to parallelize the implementation for the Bees Algorithm; instead, it relates to experience for representing a vast amount of solutions within one population of a particular generation. In each generation through the execution of the Bees algorithm, a population of solutions survives. The exploration for some function space proceeds to certain circumstances, Bees describe these circumstances in parallel. The proposed method initialized with the empty graph and adds the edges one by one depending on the score function at each iteration that compares the score between the previous step and the selected step. If the score is best, the edge is attached to the graph. Otherwise, they stay with the earliest stage until finding the best score. The procedure continues until iteration equals a threshold, or there is no alternative to get the best score than the previous one. The process for adding, deleting, moving and reversing mentioned in the section (3.5.1) and Figure 3.9. SAB starts with a population n , at the start, it would count all n as scout Bees. Scout Bees are randomly assigned in the search space. The initial temperature T_0 initialized in step 1. Figure 3.11 shows the algorithm. The fitness of the sites (i.e. the performance for the candidate solutions) sensed by the scout Bees are estimated in step 2. Then the temperature is reduced in a small degree for the current position. Next, compare the BDeu score function within the current position and prior position if the score of the current position is close to the prior one, they stay in the current position or the value of $(\exp(-(\text{score of current state}) - (\text{score of the previous state})) > \text{random}(0,1))$ they return to the prior position for selecting another position in step 3. The fitness function used is problem specific. The m sites that realize

assuring points in the search space designated as “selected sites” also accepted for neighbourhood search in step 4.

In Step 5, SAB manages searches in the neighbourhood of the selected sites, distributing extra Bees to explore near to the best e sites. They can arrange the Bees immediately according to the fitnesses compared among the sites that they are visiting. In step 6, the quality of recruited individuals should arrange through using the fitness function. The determination for whether to use drivers on the individuals and whether to hold that within the population allows us to adjust execution.

Generating populations from solutions, rather than a particular solution, is an effort to control the ability to explore deep areas from the exploration space in a parallelization method, as Bees algorithm takes in its previous steps of the exploration. During the earlier phases from the search, there does a tremendous amount of difference within the areas of the function space that is being simultaneously investigated. While the search proceeds, the population serves to concentrate a better solution in the function space. The extensive literature about meta-heuristics reports that a hopeful method for getting high-quality solutions is to pair a local search algorithm with a mechanism to provide initial solutions. Iterated local search algorithm is between the best-performing

Algorithm SAB (*Bee algorithm is global and Simulated Annealing is local search (hybrid bee and simulated annealing algorithms)*)
INPUT: - datasets
OUTPUT: - *Learned and Constructed BN 1*
-The initial temperature T_0 , Initialize population n with random solutions.
2-Evaluate the fitness of the sites (i.e. the performance of the candidate solutions) visited by the scout bees.
3- loop until less than stopping conditional: -
 3-1 the temperature is reduced by a small amount Δt using the decrement functions
 3-2 compare the fitness function (BDeu score function) of the current location and the previous if it's better than previous (set current state is best) or ($\exp - (\text{current score state}) - \exp - (\text{previous score state}) > \text{random} [0,1]$) then selected it difference between the return to the previous location.
4- chooses the site solution and evaluates the fitness (Select sites m for neighborhood search., Recruit bees for selected sites (more bees for best e sites) and evaluate fitness's.).
5- the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. It is the key feature of Bees algorithm to escape local optimum
6-Return the maximum score function for BDeu
7- New population with scout-bees and score function.

Figure 3.11 Pseudo code SAB (Bee global search and Simulated Annealing is local search).

algorithms. They use the local search to original solutions that produced, by presenting modifications on any optimal solutions.

Simulated Annealing algorithm uses local search through the process. Once the Bees should finish their solution development, they can use the solutions on their local optimum with the utilization for a local search routine. Such a coupling of solution development with local search is a hopeful approach. In nature, because the Simulated Annealing algorithm's solution development uses any neighbourhood than local search, the possibility that local search develops a solution invented by a simulated Annealing is excellent. Global search algorithms experience of the problem of getting useful, new solutions, these solutions produce through the artificial Bees. The time-consuming for Simulated Annealing they relinquish within several levels of cooling until the balancing — simulated Annealing has known the limited section of space that should be searching for the right place. For guiding the search, simulated annealing should get any information on the whole area of the effects of previous searches.

It calls an individual approach that combines the Bees algorithm among simulated annealing algorithms to produce the combination Bees' Simulated Annealing. As described before, the Bees algorithm begins with a population of arbitrarily created competitors and 'evolves' towards genuine solutions by implementing local search operatives.

During the standard SA, the algorithm continues iteratively through the beginning for an original point produced via chance, while, preferably of repeating with a solution, BSA seeks for increase a population of solution for iterative neighbourhood operators.

The properties of Bees' simulated annealing are:

1. The algorithm uses a population of solutions from iterating with single solutions, which increases the possibility of leaving from a local optimum and drives to fast concentration to the global solution.
2. BSA can observe a parallel implementation of simulated annealing, which shows its agreement for the parallel processing system.
3. The algorithm is capable of solving complicated problems in huge dimensions that have not explained previously.

3.6.4 FOURTH PROPOSED METHOD

In this section, we present, a different approach for Structure Learning Bayesian Network depended on the improved Bee optimization algorithm. It used the Bee procedure as a search method for learning structural Bayesian network. Apply the BDeu metric as the score function. A hybrid method depended on Bee optimization as a local search and Greedy as global search has applied the search algorithm primarily for solving the optimization problem and uses BDeu as a score function for computing the scoring metric. A proposed method offers improved Bee optimization by intermixing the search characteristics of Greedy and Bee optimization within an individual global principle named Bees' Greedy algorithms (BLGG). Figure 3.12 shows the Pseudo-code. The Bees algorithm is a repeated process that is composed of a population of individuals; every Bee encodes a solution in an assigned problem space. It allocates this space as the exploration space. The Bees algorithm used in areas wide to be exhaustively searched (such as those in combinatorial optimization). It encodes solutions in best patterns expected to specify computational improvements associated with all problems. Once the Bees should finish the structure of the solution, the solutions can hold local optimum through performing the local search routine. In the solution's structure, a Bee' algorithm' uses several neighbourhoods than local search. The probability of developing the structure solution of local searches through a Bee is excellent. BLGG start among an assigned population n , in the beginning, it would include every n scout Bees. Scout Bees randomly assigned within a search space. The beginning population n also picked some arbitrary solutions within step 1. The BDeu score function (the fitness value) computed on the node (they evaluate the representation from the applicant solutions) visited by the scout Bees during step 2. The m sites that describe circumstances within the search space shown as "selected sites" also taken for neighbourhood search during step 4. While Step 5, BLGG manages searches in the neighbourhood from the selected sites, allowing extra Bees expected to seek on the best e sites. The Bees can take immediately regarding the fitnesses associated among the sites that they are visiting. The recruited Bee used to arrange the solution based on the fitness function during step 6. As well as concerning the global search used greedy search to select the random solution from neighborhood of the valid solution and check if the new solution it has a better score than the previous one or not, if it has better solution, they stay in the new position if not the return back to the previous solution in step 7 and 8.

The Bayesian structure learning begins with an empty graph G_0 (arcs-less DAG) and proceeds by combining an arc at a time. The production procedure of a BN presented in Figure 3.13. Wherever a current state G_c of a Bee is a graph that includes all nodes, $X_i \in X_c$ arcs also no directed cycle. Assume there are (z) applicant directed arcs. Under the names from the heuristic information of applicant arcs, some Bee picks the s^{th} arc

Algorithm BLGG (*Bee algorithm local and Greedy is global search (hybrid Bee and Greedy algorithms)*)

INPUT: - datasets

OUTPUT: - Learned and Constructed BN

1. Initialize population n with random solutions.
- 2-Evaluate the fitness of the sites (i.e. the performance of the candidate solutions) visited by the scout bees.
- 3- loop until less than stopping conditional: -
 - 3-1 choose the site solution and evaluate fitness.
 - 3-2 compare the fitness function (*BDeu* score function) of the current location and the previous if it's better than previous (set current state is best) or ($\exp(\text{current state}) - \exp(\text{previous state}) > \text{random} [0,1]$) then selected it the difference between the return to the previous location.
- 4- chooses the site solution and evaluates the fitness (Select sites m for neighborhood search., Recruit Bees for selected sites (more bees for best e sites) and evaluate fitness's.).
- 5- The Remaining Bees in the population are assigned randomly around the search space scouting for new potential solutions.
- 6-Return the maximum score function for *DBeu*.
7. Randomly generate a new network from the current best network and evaluate it.
8. If the newly generated solution in step 7 has a higher score than the current best network, set the new network as the current best network.
9. Repeat 7–8.
10. Stop.

Figure 3.12 Pseudo code BLGG (Bee local search and Greedy is global search).

a_{ij} since a new element of a solution; thus the original position by combining an arc a_{ij} can be expressed as $G_{h+1} = G_c \cup a_{ij}$. Earlier, there is no way to get the score of a BN construction by combining an arc; the construction process is completed, including the Bee preparing its solution G_n . The chosen heuristic is to connect in the diagram the arc performing highest in the selected decomposable metric f .

$$\eta_{ij} = f(x_i, \text{Pa}(x_i) \cup \{x_j\}) - f(x_i, \text{Pa}(x_i)) \quad \text{Equation 3-15}$$

The opportunity for choosing a specific food source based on the score function that used as a metric score (BDeu) after calculating the score function. Subsequently, if the prior score is higher than the new score, hold the old score while they complete the searching for a neighbour by attendant Bees and onlookers. The employee Bee can use an operator like (deletion, addition, reversion, and move) for the arch in the graph, while onlooker picked randomly to manage knowledge for these comfortable operators. For deletion of “ $X_i \rightarrow X_j$ ”, we need to evaluate $\text{Score}(X_j, \Pi X_j \setminus \{X_i\})$; for adding “ $X_i \rightarrow X_j$ ”, we need to evaluate $\text{Score}(X_j, \Pi X_j \cup \{X_i\})$; and for reversion of “ $X_i \rightarrow X_j$ ”, two local scores $\text{Score}(X_j, \Pi X_j \setminus \{X_i\})$ and $\text{Score}(X_i, \Pi X_i \cup \{X_j\})$ are to be updated. Later they search for getting a different solution based on neighbourhood then compare the score metric if it’s greater than prior or the $(\exp(-\text{current}/\text{old}))$ higher than a random number $[1,0]$, the employees transfer to the new location otherwise remain at the same location until finding a new solution with a larger score than prior one. After finishing this operator, they share information about the new solution. The onlooker chooses a suitable solution. If they do not accept the solution of an employee, then the employee will leave the solution and changed to be the scout Bee. The scout Bee may develop a new solution based on the heuristic information and then suits an employee again. After each repetition the Bee performed, the BLGG algorithm shall carry out the updating process, which incorporates local and global updating steps.

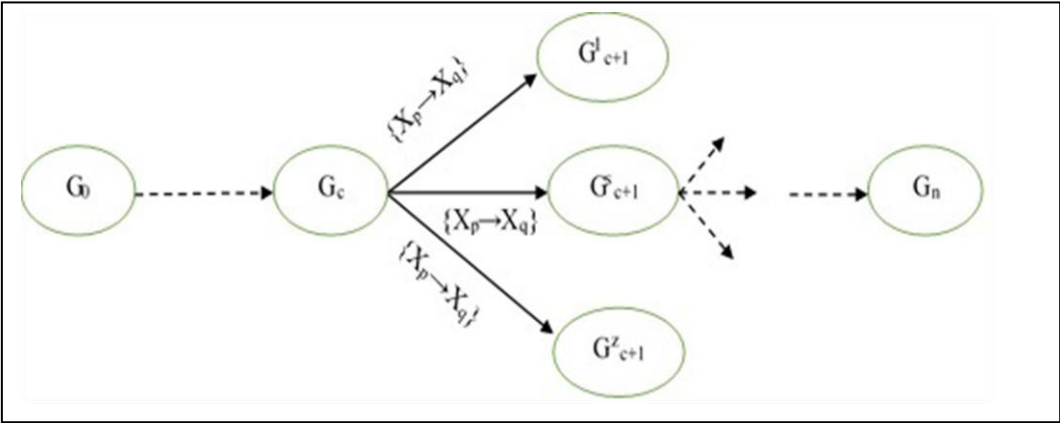


Figure 3.13 The construction process of a Bayesian Network

3.6.5 FIFTH PROPOSED METHOD

In this section, we present the fifth proposed method. The proposed method offers a hybrid method between Greedy search and Bee optimization, used greedy as local search and Bee as global search also used BDeu as score function to structure learning

Bayesian network called (BGGL). The Pseudo-code of BGGL is shown in Figure 3.14. As shown in Figure 3.14, they start randomly choosing a solution in step 1 and test the solution fitness (score function) for the scout Bee that selected the nodes. Iterate a loop until finishing criteria to getting the best score function in the current location by comparing to the neighbour node and test it. After picking the new position they compare the fitness function (BDeu score function) within the current status and prior position. If the BDeu score function in the current score is better than the score from the previous position, they pick the current state, and they remain in the same location until discovering other sites from the current location. If the fitness (BDeu score) of the prior position is better than current or is equal, they cancel the current position and return to the prior position. The recruit bee selected the solution and evaluated it; the other bee selects the solution randomly in the search space to choose the best score function and return the best score at each iteration. The objective for heterogeneous Bees' Greedy algorithm is to utilize a conventional Greedy method after the Representative-Bee has decided to execute the search process towards a higher optimal solution space. One of the significant properties of the Bees algorithm is its facility to search the function space from various points in parallel. Creating a population of solutions, rather than an individual solution, is a trial to check the occurrence to explore large regions of the search space in a parallelized manner, as Bees algorithm does in its earlier stages' of the search. In the earlier stages of the search, there is a vast amount of variety in the regions of the function space, which are concurrently explored. As the search proceeds, the population tends to concentrate nearby the right solution in the function space. They encode solutions in several forms because of certain computational advantages associated with each problem. The most representative solutions involve binary-based encoding, character-based encoding, real-value encoding. In this context, parallelism does not refer to the ability to parallelize the implementation for the Bees' Algorithm; instead, it relates to the ability to represent a massive number of potential solutions in the population of a single generation. In every generation during the execution of the Bees algorithm, a population of solutions exists.

The search of the function space proceeds from these points, Bees represent these points in parallel.

This kind of parallelism allows the members of the population to concentrate on very similar solutions. Once the population has focused, the experience for random search

Algorithm BGGL (*Bee algorithm Global and Greedy is local search (hybrid Bee and Greedy algorithms)*)

INPUT: - datasets

OUTPUT: - Learned and Constructed BN

1. Initialize population n with random solutions.
- 2-Evaluate the fitness of the sites (i.e. the performance of the candidate solutions) visited by the scout bees.
- 3- loop until less than stopping conditional: -
 - 3.1 Randomly generate a new network from the current best network and evaluate it.
 - 3.2 If the newly generated solution in step 3.1 has a higher score than the current best network, set the new network as the current best network
- 4 choose the site solution and evaluate the fitness.
- 5 compare the fitness function (BDeu score function) of the current location and the previous if it's better than previous (set current state is best) or ($\exp(-\text{current state}) - \exp(-\text{previous state}) > \text{random}[0,1]$) then selected it the difference between the return to the previous location.
- 6- chooses the site solution and evaluates the fitness (Select sites m for neighborhood search., Recruit bees for selected sites (more bees for best e sites) and evaluate fitness's.).
- 7- the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions.
- 8-Return the maximum score function for DBeu

Figure 3.14 Pseudo code of BGGL (Bee global search and Greedy is local search)

procedure in Bees' algorithm to help investigate new portions of the function space is hugely limited. The premature concentration of a population may happen if the population grows too homogenous. In the regular Bees algorithm, they should label the problem of the early attention, including the trap of a local optimum through the random search executed after the process. A structure of the Bayesian network holds in four procedures, as shown in Figure 3.13, at every level of performing the algorithm (Addition, deletion, revers, and move). An Addition operator first randomly picks two

nodes X_j and X_i where $i \neq j$, and $X_i \in X \setminus \Pi(X_j)$: If adding an arc $a_{ij} = X_i \rightarrow X_j$ seems not to produce a directed cycle, then $G_{c+1} = G_c \cup \{a_{ij}\}$. A second operator is Deletion, first chooses an arc a_{ij} from nodes X_i to X_j which is already in the G_c , then deletes it from the G_c , a new solution, $G_{c+1} = G_c \setminus \{a_{ij}\}$; is concerned. The third operator is Reversion, randomly selects an arc a_{ij} from A , and then turns the direction for the arc if the inversion of the arc still forms a DAG. Through this operator, a new solution, $G_c \setminus \{a_{ij}\} \cup \{a_{ji}\}$ is constructed. The last operator is Move for two nodes X_i and X_j whose parent sets are not empty, the operator, selects a parent node of these two nodes, $X_k \in \Pi(X_i)$ and $X_l \in \Pi(X_j)$ ($k \neq l$), then changes X_k with X_l if $X_l \in (X \setminus \Pi(X_i) \cup \{X_i\})$; $X_k \in (X \setminus \Pi(X_j) \cup \{X_j\})$ and this move operator still forms a DAG. Namely, the operator simultaneously changes the parent sets of two nodes.

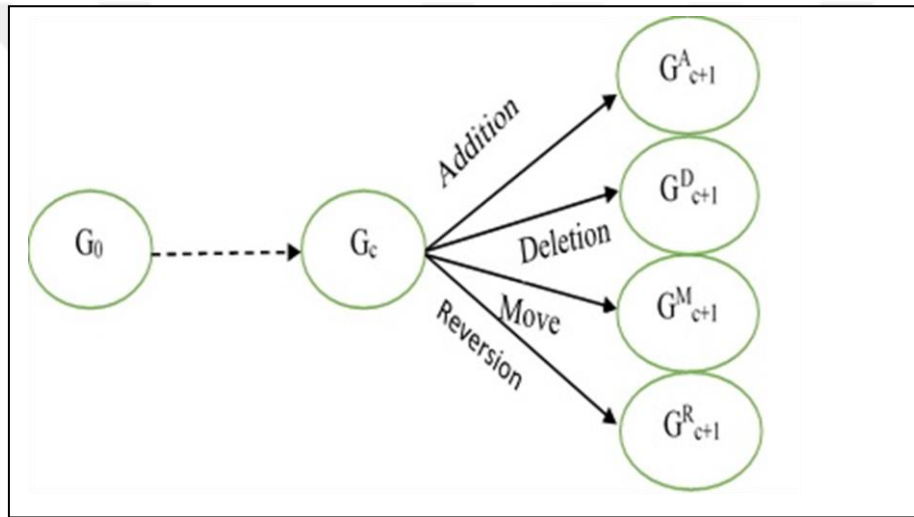


Figure 3.15 The construction process of a BN

3.6.6 SIXTH PROPOSED METHOD

The proposed method uses ESWSA approach as a search method for structural learning of Bayesian networks. The BDeu metric used as a score function for assessing the Bayesian network structure. The ESWSA algorithm is effectively an iterated procedure that consists of a population of individuals where every elephant encodes a potential position and velocity in a given space. This space is held to be the search area. The proposed method depends on two techniques. The first technique uses Equation (3-11) for local search through the essential process if ($\text{rand} \leq p$). The second one uses Equation (3-10) for global search through the necessary process if ($\text{rand} > p$). Figure 3.15 shows the pseudo code of this technique. ESWSA

algorithm's solution construction utilizes different neighbourhood than local search. The expectation is high that the local search updates a solution formed by an elephant group. Structure learning Bayesian network solution area formed for each potential DAG. Every elephant group inside the swarm initiates a possible solution which represents as a DAG having empty arcs. An elephant later examines the exploration area for finding the approximately near-optimal or optimal solution, which is known as the BDeu score. Equation (2-34) is used to calculate the BDeu score as the goal function of the optimization. The exploration aims for obtaining a higher BDeu score for the network structure. All initial solutions produced through iterative operations. Starting with a blank graph (G_0), the arcs are appended one after another, provided that they not included in the current graph solution. The append operation performs if only if the score function of the new solution is higher than the current score and also the new solution satisfies the DAG constraint. This process continues until the quantity of the arcs equals the amount defined in advance. In the model, the solution starts assigning a population for each operator and picks the solution, which has a higher score function. Elephant group continues according to the selected operator until the process has performed a maximum number of iterations or the BDeu score does not increase any more. Typically, the processes hold four separate operations in local optimization: Deletion, Addition, Reversion, Movement. The first three are-simple operations within this domain, involve just replacing an individual edge every time from a competitor solution. It allows the inclusion of a comparatively small area near the solution. With every movement operation, on the other hand, the existing edges change the set of parents, which can make a moderately significant modification for the current solution. Therefore, if the solution not changed after applying simple operators, the move operator may improve it. Walking is the primary force utilizing the chosen operation in local optimization, which becomes further widespread while an elephant approaches the desirable solution. Walking directions, the switch with various local optimization operators, grows extra widespread as an elephant moves continuously from a solution through exploration toward a better one. Therefore, the current velocity renewed by

either elephant's best global or best local solution based on the (p) value. The ESWSA based on the probability value p can switch off from global search into local or from local to global. The velocity of ESWSA is renewed based on the current best position of the elephant in the local search.

On the other hand, the global velocity depends on the best global solution concerning elephants in a global search, near a global best position. As shown in Figure 3.17, an elephant G0, which describes a DAG with arcs, tries reversion, move, addition, and deletion, and reaches new solutions G1, G2, G3, and G4, respectively. Assuming the best score is in G3, it will select, and the elephant will proceed to examine some similar process to get G+3 as the new solution.

Algorithm: Structure Learning of Bayesian Network based on elephant swarm water search algorithm

INPUT: - datasets

NE: number of Elephant swarm

D: search space dimension

P: the switching probability p

Search range: the search space border

t_{max}: maximum number of iteration number; X_{max}: upper boundary, and X_{min}: lower boundary

OUTPUT: - learning Bayesian Network

- (1) *The initialized empty structure and initialize parameters of ESWSA algorithm (dimension space D, size of population NE, the switching probability p, the number of iteration number, upper boundary and lower boundary, (G_{best,i,d}^t), and X_{max} > X_{min}.*
- (2) *Set the velocity and position for all Elephant randomly. Comparing each elephant by BDe score function, and find the best in the current position (P_{best,i,d}^t).*
- (3) *Assign the value of w^t according to the weight update using Equations (3-13) or (3-14).*
- (4) *Find a new best position by comparing the BDeu score function of each elephant.*
- (5) *If rand > p, update elephant velocity (V_{i,d}) using equation (3-10).*
- (6) *else rand ≤ p update elephant velocity using equation (3-11).*
- (7) *Update the position X_{i,d} using equation (3-12).*
- (8) *Evaluate BDeu score function of the new position (X_{i,d}^t)*
- (9) *If current position (X_{i,d}^t) is better than the best position (P_{best,i,d}^t) then update the best position by ((P_{best,i,d}^t) = (X_{i,d}^t))*
- (10) *If ((G_{best,i,d}^t) < current position then update the best solution for global by (G_{best,i,d}^t = (X_{i,d}^t))*
- (11) *The best score value and solution saved.*
- (12) *If X_{min} ≥ X_{max}, stop the iteration process, and the results are present. If not, move into Step 5.*
- (13) *Return the maximum BDe score.*

Figure 3.16 ESWSA Algorithm for Structure learning Bayesian Network.

If the BDeu score of G+3 is higher Than that of G+1, it will continue to perform the corresponding operator. The operations will repeat until the BDeu score stabilizes, or iteration loop reaches the maximum. In the whole process, the elephant selects among the directions using deletion, addition, movement and reversion operations.

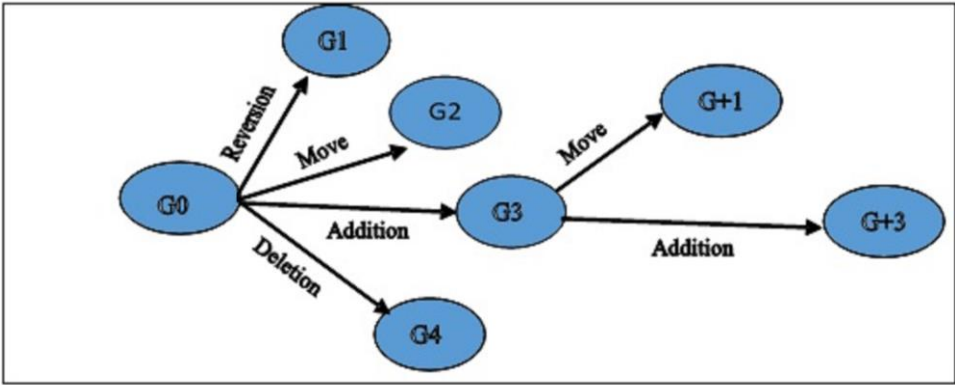


Figure 3.17 Water searching steps for one Elephant [10]



CHAPTER 4

DATASETS AND EXPERIMENTS

4.1 DATASETS

To evaluate algorithms performance, a standard assessment technique utilized by employing probabilistic datasets extracted from popular Bayesian networks benchmarks. The platform for experiments includes a PC having the following properties: Core i3, 2.1GHz CPU, 4GB RAM, Ubuntu 14.04 operating system and utilizes Java to implement the algorithms. For the experiments, we used $p=0.7$. The last ones, $t_{\max}=1000$ and population size $N=50$, are fixed parameters of ESWSA optimization. The parameters of Simulated Annealing algorithms are as follows: Temperature of Reannealing = 500, cooling factor= 0.8, Initial temperature= 1000. Greedy search parameters are as follows: Recommended minimum networks before reboot = 3000, minimum recommended networks after highest score = 1000, maximum recommended networks before reboot = 5000, the maximum parent count for operations Reboot=5, restart by random network = yes. The parameters of Bee algorithms are : Number of Scout Bees $n=200$, Number of Sites m out of n visited sites=30, Number of best site e out of m selected site =7, Number of Bees recruited for best e site $n_2=90$, Number of Bees recruited for the other site $(m-e)$ (n_1)=30, Initial size of patches ng which includes site randomly selected=200, Number of algorithm steps repetitions $imax=10000$. The Pigeon parameters are Pigeons number ($NP=300$), search space dimension ($D=20$), the factor of the map and compass ($P=0.3$), the maximum number of iteration number for the map and compass operation ($Nc1_{\max}=5000$), the maximum number of iteration number for the landmark operation ($Nc2_{\max}=10000$).

The algorithms have been implemented in three different execution times: 2 minutes, 5 minutes and 60 minutes. The datasets that we used in this work are (Alarm, Adult, Epigenetics, Heart, Hepatitis, Imports, Letter, Parkinson's, Sensors, WDBC, Water, win95pts, Andes, Hepar, Hail, static banjo, mushroom, Autos, Soybean,... etc.). The number of nodes, arc, the total number of the instance given below :

- ALARM, It has 37 variables, 46 arcs, Number of parameters 509 and 10000 instances.
- EPIGENETICS, it has 30 variables and no. of instance=72228
- HAILFINDER, it has 56 variables 66 arcs, and 3000 instances.
- ASIA, it has 8 variables, 8 arcs, and 3000 instances.
- INSURANCE, it has 27 variables 52 arcs, and 3000 instances.
- ADULT it has 16 variables and 30162 instance
- CHILD, it has 20 variables, 25 arcs and 230 instance=230.
- PATHFINDER, it has 135 variables, 200 arcs, and 77155 instances.
- HEPATITIS, has 35 variables and 137 instances.
- IMPORTS has 22 variables and 205 instances.
- LETTER, it has 17 variables, and 20000 instances.
- PARKINSONS, it has 23 variables, and 195 instances.
- SENSORS, it has 25 nodes, and 5456 instances.
- WDBC, it has 9 nodes and 1000 instance.
- WATER, it has 32 nodes, arcs 66, and 10083 instance.
- WIN95PTS, it has 76 nodes, no. of arc=112, and 574 instance.
- ANDES, it has 223 variables, 338 arcs, and 500 instance.
- HEPAR2, it has 70 variables, 123 arcs, and 350 instance.
- STATIC BANJO DATASET is the Static Bayesian network with 33 variables and 320 instance.

- LUCAS is modelling a medical application for the diagnosis, prevention, and cure of lung cancer. It has 11 variables and 10000 observations
- HORSE, it has 23 variables and 126 instances.
- FLAG has 29 variables and 194 instances.
- Mushroom, it has 23 variables and 1000 instance.
- SOYBEAN, it has 35 variables and 307 instances.
- SPECT.HEART has 22 variables and 267 instances.
- LUCAP2, it has 143 variables and 10000 instances.

4.2 EXPERIMENTAL RESULTS

4.2.1 FIRST PROPOSED METHOD

In this section, we presented the BDeu score function of the first proposed method (Bayesian Network Structure learning based on Pigeon Inspired Optimization) and compared it to the default Simulated Annealing and Greedy search algorithms using a different dataset. As shown in the tables (4.1, 4.2, and 4.3) the score function of the first proposed method is better than the other mentioned algorithms. We calculate the score function in 3 different times, as shown in the tables. The score produced by the first proposed method in 2 minutes is better than the score provided by Simulated Annealing and Greedy search in 60 minutes. From this table, it can be noted that the proposed method produces better score values than the default Greedy Search plus simulated Annealing Algorithms for all situations. It indicates that the PIO finds the best score with the minimum time required. The BDeu score function of the first proposed method need not implement the program more time as it produced a score function in 2 minutes while other algorithms needed more time to produce a useful score function. So the first proposed method offered a high speed for providing a better BDeu score function.

Table 4.1 Calculation results of the best of BDeu Score function for PIO with Simulated Annealing and Greedy in 2 minutes Execution time

Dataset	PIO	Simulated Annealing	Greedy
Hepatitis	-1327.73	-1330.4645	-1350.16
Parkinson's	-1598.91	-1601.2968	-1732.76
Imports	-1811.99	-1828.9059	-1994.15
Heart	-2423.8	-2432.1878	-2576.93
mushroom	-3372.51	-3375.3104	-3734.22
WDBC	-6666.04	-6682.7161	-8089.41
Water	-13269.5	-13290.8278	-14619.1
win95pts	-46779.5	-47085.0996	-83749.3
Sensors	-60343.3	-60710.4985	-69200.3
Hepar	-160095	-161086.4216	-169497
Letter	-175200	-178562.2167	-184307
Epigenetics	-176657	-179910.3328	-225346
Adult	-207809	-211677.7164	-211844

Table 4.2 Calculation results of the best of BDeu Score function for PIO with Simulated Annealing and Greedy in 5 minutes Execution time

Dataset	PIO	Simulated Annealing	Greedy
Hepatitis	-1327.73	-1330.46	-1350.16
Parkinson's	-1598.91	-1601.3	-1721.16
Imports	-1811.99	-1828.91	-2012.21
Heart	-2423.8	-2423.8	-2560.43
mushroom	-3372.51	-3375.31	-3706.66
WDBC	-6666.04	-6682.72	-7954.65
Water	-13269.5	-13290.8	-14644.7
win95pts	-46779.5	-47085.1	-83150.7
Sensors	-60343.3	-60710.5	-69150
Hepar	-160095	-161086	-169881
Letter	-175200	-178562	-184916
Epigenetics	-176657	-179300	-224172
Adult	-207809	-211678	-211781

Table 4.3 Calculation results of the best of BDeu Score function for PIO with Simulated Annealing and Greedy in 60 minutes Execution time

Dataset	PIO	Simulated Annealing	Greedy
Hepatitis	-1327.73	-1330.46	-1350.16
Parkinson's	-1598.91	-1601.3	-1700.36
Imports	-1811.99	-1828.91	-1995.76
Heart	-2423.8	-2432.19	-2527.44
mushroom	-3372.51	-3375.31	-3588.69
WDBC	-6666.04	-6682.72	-7841.35
Water	-13269.5	-13290.8	-14272
win95pts	-46779.5	-47085.1	-81779.5
Sensors	-60343.3	-60710.5	-68364
Hepar	-160095	-161086	-168871
Letter	-175200	-178562	-184118
Epigenetics	-176657	-179300	-217246
Adult	-207809	-211678	-211762

4.2.2 SECOND AND THIRD PROPOSED METHODS (BSA AND SAB)

In this section, we present BDeu score function for the hybrid Bee and simulated annealing algorithms (Bee algorithm is local and Simulated Annealing is global search (BSA)) as second methods and (Simulated Annealing is a local search and Bee is a global search (SAB)) as third proposed methods. The result compared with default Simulated Annealing as shown in the tables (4.4, 4.5 and 4.6).

Table 4.4 Calculation results of the best of BDeu Score function for BSA and SAB with Simulated Annealing in 2 minutes Execution time

Dataset	Simulated Annealing	BeeLocal SimGlobal	BeeGlobal SimLocal
spect.heart	-2141.4678	-2141.5364	-2140.9118
soybean	-2870.8509	-2859.1344	-2857.2898
Static banjo	-8451.4948	-8449.2862	-8451.8344
Water	-13262.5288	-13262.5288	-13262.5288
Dynamic data	-15935.2861	-15935.2861	-15935.2861
Alarm	-104927.1078	-104927.108	-104927.108
Lucap2	-112260.5067	-111413.333	-111963.759
Hail	-148192.92	-148179.926	-148187.684
hepar	-161051.6944	-161049.602	-161050.961
Andes	-497353.2663	-477461.481	-492382.845

Table 4.5 Calculation results of the best of BDeu Score function for BSA and SAB with Simulated Annealing in 5 minutes Execution time

Dataset	Simulated Annealing	BeeLocal SimGlobal	BeeGlobal SimLocal
spect.heart	-2143.7306	-2141.3482	-2142.5688
soybean	-2857.852	-2847.4824	-2863.8429
Static banjo	-8449.7696	-8445.3556	-8445.411
Water	-13266.0091	-13262.5288	-13262.5288
Dynamic data	-15935.2861	-15935.2861	-15935.2861
Alarm	-104927.1078	-104927.108	-104927.108
Lucap2	-112217.4215	-110142	-110834.219
Hail	-148188.1576	-148179.325	-148178.645
hepar	-161052.5088	-161048.986	-161052.513
Andes	-489795.7252,	-473468.504	-480065.267

Table 4.6 Calculation results of the best of BDeu Score function for BSA and SAB with Simulated Annealing in 60 minutes Execution time

Dataset	Simulated Annealing	BeeLocal SimGlobal	BeeGlobal SimLocal
spect.heart	-2142.2432	-2141.9638	-2141.8104
soybean	-3012.7233	-2984.7118	-2992.9934
Static banjo	-8556.703	-8545.5115	-8552.3736
Water	-13263.7708	-13262.0855	-13262.2007
Dynamic data	-15935.2861	-15935.2861	-15935.2861
Alarm	-105376.7	-105043.762	-105270.67
Lucap2	-150937.567	-149052.6988	-151160.106
Hail	-152298.908	-151671.6704	-151772.555
hepar	-163418.883	-162412.9857	-163230.937
Andes	-586760.471	-578144.03	-587098.489

The Tables (4.4, 4.5, and 4.6) present the score for each algorithm in the mentioned datasets and time values, the results show that the hybrid algorithm produced better scores than the default Simulated Annealing algorithm in the most dataset and equals in some dataset. The results indicate that using Bee as a local search and simulated annealing as a global search(BSA), they produced a better score than the default Simulated Annealing algorithm and SAB Algorithm.

4.2.3 FOURTH AND FIFTH PROPOSED METHODS (BLGG AND BGGL)

In this section, we present the BDeu score function for Fourth (Bee as local search and Greedy as global search(BLGG)), and Fifth (Greedy as local search and Bee as global search(BGGL)) methods. The results are shown in Tables (4.7, 4.8, and 4.9).

Table 4.7 Calculation results of the best of BDeu Score function for BLGG and BGGL with default Greedy search in 2 minutes Execution time

Dataset	Greedy	Bee Local Greedy Global	Bee Global Greedy Local
Dynamic data	-15935.2861	-15935.2861	-15935.2861
spect.heart	-2144.6547	-2144.317	-2141.5364
Water	-13263.7708	-13264.1145	-13262.8093
Static banjo	-8585.2097	-8576.3336	-8570.2096
soybean	-3021.4054	-3025.8652	-3032.1729
Alarm	-105971.754	-106061.1308	-105552.278
Hail	-152649.937	-152099.9767	-152037.997
hepar	-163474.268	-163432.0852	-161050.961
Lucap2	-151215.276	-150907.7339	-151242.738
Andes	-591870.61	-587911.3992	-589927.223

Table 4.8 Calculation results of the best of BDeu Score function for BLGG and BGGL with default Greedy search in 5 minutes Execution time

Dataset	Greedy	BeeLocal Greedy Global	BeeGlobal SimLocal
Dynamic data	-15935.2861	-15935.2861	-15935.2861
spect.heart	-2142.8904	-2143.1913	-2142.7278
Water	-13265.261	-13264.8021	-13264.4597
Static banjo	-8561.9296	-8556.0676	-8448.2838
soybean	-3011.3836	-3009.4569	-2991.8209
Alarm	-106113.938	-105788.8594	-106170.992
Hail	-153436.041	-151710.6892	-151863.228
hepar	-163536.077	-163257.7531	-163374.811
Lucap2	-152092.434	-150308.0311	-151912.804
Andes	-588502.538	-587826.2274	-584604.764

Table 4.9 Calculation results of the best of BDeu Score function for BLGG and BLGG with default Greedy search in 60 minutes Execution time

Dataset	Greedy	BeeLocal Greedy Global	BeeGlobal SimLocal
Dynamic data	-15935.2861	-15935.2861	-15935.2861
spect.heart	-2142.2432	-2141.9638	-2141.8104
Water	-13263.7708	-13262.0855	-13262.2007
Static banjo	-8556.703	-8545.5115	-8552.3736
soybean	-3012.7233	-2984.7118	-2992.9934
Alarm	-105376.7	-105043.762	-105270.67
Hail	-152298.908	-151671.6704	-151772.555
hepar	-163418.883	-162412.9857	-163230.937
Lucap2	-150937.567	-149052.6988	-151160.106
Andes	-586760.471	-578144.03	-587098.489

The results in tables present the score for each algorithm in the mentioned datasets and time values. From this table, it can be noted that the hybrid algorithms Bee and Greedy (BLGG and BGGL) produced better score values than the default Greedy search in most of the datasets as shown in the above table or the score is equal in some datasets.

4.2.4 SIXTH PROPOSED METHODS (ESWSA)

In this section, we present the BDeu score function of the Sixth proposed method (Bayesian Network Structure learning using Elephant Swarm Water Search Algorithm) and compared it to the default Simulated Annealing and Greedy search algorithms using a different dataset. As shown in the Tables (4.10, 4.11, and 4.12), it can be noted that the proposed method produces better score values than the default Greedy Search and Simulated Annealing Algorithms for most situations. It indicates that the ESWSA finds the best score with the minimum time required. We calculate the score function in 3 different times, as shown in the tables. The score produced by the sixth proposed method in 2 minutes is better than the score produced by Simulated Annealing.

Table 4.10 Score function the best of ESWSA, Simulated Annealing, and Greedy in 2 minutes Execution time

Dataset	ESWSA	Simulated Annealing	Greedy
Asia	-54849.9	-56340.27	-56340.3
WDBC	-6660.43	-6682.716	-8089.41
lucas01	-11863.1	-12243.24	-13890.9
Adult	-207809	-211677.7	-211844
Letter	-175200	-178562.2	-184307
Child	-62365.7	-62343.73	-63336.6
Imports	-1811.99	-1828.906	-1994.15
Heart	-2426.42	-2432.188	-2576.93
Parkinson's	-1486.86	-1601.297	-1732.76
Mushroom	-3160.87	-3375.31	-3745.46
Sensors	-60343.3	-60710.5	-69200.3
insurance	-13895.11	-13872.33	-13904.6
Epigenetics	-176636	-179910.3	-225346
Water	-11562.7	-13290.83	-14619.1
Static. banjo	-8409.42	-8451.495	-8585.21
Hepatitis	-1327.73	-1330.465	-1350.16
Hail finder	-75583.9	-148192.9	-153602
Hepar	-160095	-161086.4	-169497
win95pts	-46779.5	-47085.1	-83749.3

Table 4.11 Score function the best of ESWSA, Simulated Annealing, and Greedy in 5 minutes Execution time

Dataset	ESWSA	Simulated Annealing	Greedy
Asia	-54849.9	-56340.27	-56340.3
WDBC	-6660.43	-6682.716	-7954.65
lucas01	-11492.7	-12243.24	-12243.2
Adult	-207258	-211677.7	-211781
Letter	-175200	-178562.2	-184916
Child	-62365.7	-62343.73	-63799.4
Imports	-1811.99	-1828.906	-2012.21
Heart	-2426.42	-2423.804	-2560.43
Parkinson's	-1439.09	-1601.297	-1721.16
Mushroom	-3160.87	-3375.31	-3709.7
Sensors	-60343.3	-60710.5	-69150
insurance	-13895.11	-13872.33	-13904.6
Epigenetics	-176628	-179300.2	-224172
Water	-11562.6	-13290.83	-14644.7
Static. Banjo	-8409.42	-8449.77	-8561.93
Hepatitis	-1327.73	-1330.465	-1350.16
Hail finder	-75583.9	-148188.2	-153075
Hepar	-160095	-161086.4	-169881
win95pts	-46779.5	-47085.1	-83150.7

Table 4.12 Score function the best of ESWSA, Simulated Annealing, and Greedy in 60 minutes Execution time

Dataset	ESWSA	Simulated Annealing	Greedy
Asia	-29791	-56340.27	-56340.3
WDBC	-6660.43	-6682.716	-7841.35
lucas01	-11213.8	-12243.24	-12243.2
Adult	-207258	-211677.7	-211762
Letter	-175200	-178562.2	-184118
Child	-62245.7	-62343.73	-63799.4
Imports	-1811.99	-1828.906	-1995.76
Heart	-2426.42	-2432.188	-2527.44
Parkinson's	-1439.09	-1601.297	-1700.36
Mushroom	-3003.45	3375.31	-3588.69
Sensors	-60343.3	-60710.5	-68364
insurance	-13895.11	-13872.33	-13904.6
Epigenetics	-176628	-179300.2	-217246
Water	-11562.6	-13290.83	-14272
Static. Banjo	-8317.87	-8445.356	-8556.7
Hepatitis	-1327.73	-1330.465	-1350.16
Hail finder	-75583.9	-148182.7	-152299
Hepar	-160095	-161086.4	-168871
win95pts	-46779.5	-47085.1	-83150.7
Lucap2	-105251	-111274.8	-150938
Andes	-469217	-480491.3	-586760

Annealing and Greedy search in 60 minutes. The BDeu score function of the sixth proposed method need implement the program more time to produce a score function in 2 minutes while other algorithms needed more time to produce a useful score function, so the sixth proposed method offers higher speed for producing a better BDeu score function.

4.2.5 COMPARISONS OF THE PROPOSED METHODS.

In this section, we present the comparison of the all proposed method based on the calculation of the score function for all proposed methods in different times (2, 5, and 60 minutes) and applied different dataset as shown in the tables (4.13, 4.14, 4.15). The results of the proposed method for calculating the score function used different datasets has been demonstrated that in most of the situation, the ESWSA better than the other methods.

Table 4.13 Calculation results of the best of BDeu Score function for all proposed methods when time is 2M

Dataset	PIO	ESWSA	Bee Local Sim Global	Bee Global SimLocal	BeeLocal Greedy Global	Bee Global Greedy Local
Adult	-207809	-175200	-211677.716	-211677.716	-211874.6392	-211677.716
Letter	-175200	-175200	-178562.217	-178562.217	-185900.5902	-180657.984
Imports	-1811.99	-1811.99	-1828.9059	-1828.9059	-1999.868	-1898.8428
Heart	-2423.8	-2426.42	-2141.5364	-2140.9118	-2144.317	-2141.5364
Parkinson's	-1598.91	-1486.86	-1601.2968	-1601.2968	-1744.5766	-1661.0025
mushroom	-3372.51	-3160.87	-3375.3104	-3375.3104	-3798.107	-3421.1133
Sensors	-60343.3	-60343.3	-60710.4985	-60710.4985	-69298.6337	-60710.4985
Epigenetics	-176657	-176636	-186661.63	-185485.803	-229270.6243	-212526.244
Water	-13269.5	-11562.7	-13262.5288	-13262.5288	-13264.1145	-13262.8093
Hepatitis	-1327.73	-1327.73	-1330.4645	-1330.4645	-1350.1589	-1330.4645
Hepar	-160095	-160095	-161049.602	-161050.961	-163432.0852	-161050.961
win95pts	-46779.5	-46779.5	-50011.3542	-47153.2753	-85444.2886	-85313.6634

Table 4.14 Calculation results of the best of BDeu Score function for all proposed methods when time is 5M

Dataset	PIO	ESWSA	Bee		BeeLocal Greedy Global	BeeGlobal Greedy Local
			Bee Local Sim Global	Bee Global Sim Local		
Adult	-207809	-207258	-211677.716	-211678	-211915	-211674
Letter	-175200	-175200	-178562.216	-178562	-185521	-180581
Imports	-1811.99	-1811.99	-1907.1782	-1828.91	-2003.22	-1914.8
Heart	-2423.8	-2426.42	-2141.348	-2142.57	-2143.19	-2142.73
Parkinson's	-1598.91	-1439.09	-1601.296	-1601.3	-1738.95	-1633.01
mushroom	-3372.51	-3160.87	-3375.310	-3375.31	-3736.99	-3383.16
Sensors	-60343.3	-60343.3	-60710.4985	-60710.5	-69265.1	-65971.2
Epigenetics	-176657	-176628	-181123.809	-180335	-228900	-208252
Water	-13269.5	-11562.6	-13262.5288	-13262.5	-13264.8	-13264.5
Hepatitis	-1327.73	-1327.73	-1330.4645	-1330.46	-1350.16	-1334.11
Hepar	-160095	-160095	-161048.986	-161053	-163258	-163375
win95pts	-46779.5	-46779.5	-47591.4925	-50011.4	-84426.2	-83033.1

Table 4.15 Calculation results of the best of BDeu Score function for all proposed methods in 60M

Dataset	PIO	ESWSA	Bee		Bee Local Greedy Global	Bee Global Greedy Local
			Bee Local Sim Global	Bee Global Sim Local		
Adult	-207809	-207258	-211677.716	-211677.72	-211720.8765	-211666.444
Letter	-175200	-175200	-178562.217	-178562.22	-183583.4973	-179617.4523
Imports	-1811.99	-1811.99	-1828.9059	-1828.9059	-2000.0022	-1998.973
Heart	-2423.8	-2426.42	-2141.9638	-2141.8104	-2141.9638	-2141.8104
Parkinson's	-1598.91	-1439.09	-1601.2968	-1601.2968	-1715.6506	-1601.2968
mushroom	-3372.51	-3003.45	-3380.2690	-3374.2690	-3650.2127	-3365.7934
Sensors	-60343.3	-60343.3	-60710.4985	-60710.499	-68182.4056	-65358.2679
Epigenetics	-176657	-176628	-179300.215	-179300.21	-213438.6816	-201690.3021
Water	-13269.5	-11562.6	-13262.0855	-13262.201	-13262.0855	-13262.2007
Hepatitis	-1327.73	-1327.73	-1330.4645	-1330.4645	-1350.1589	-1327.9075
Hepar	-160095	-160095	-162412.986	-163230.94	-162412.9857	-163230.937
win95pts	-46779.5	-46779.5	-47085.0996	-50011.354	-79880.8266	-81091.292

4.3 EXPERIMENTAL RESULTS OF CONFUSION MATRICES

4.3.1 FIRST PROPOSED METHOD

To evaluate the success of structure discovery, the confusion matrix is commonly used in the literature [114]. Confusion matrix values can be computed for each algorithm and data set using known network structures. The general idea is to compare the known network structure with the produced network. To calculate the confusion matrix, first, we need to have a set of predictions network so that it can be compared to the actual network. Each row in a confusion matrix represents an actual class, while each column represents a predicted class. To test the success of structure discovery, we have to compute the confusion matrix for each data set and its known network structure. We have calculated the metrics TP, TN, FN, and FP for each network per algorithm and the criteria (Sensitivity (SE), Accuracy (Acc), F1_Score, and AHD). The meanings of these metrics are as follows: A TP is an arc (vertex or edge) in the right position inside the learning network. TN is the arc inside neither the learning network nor the proper network. FP is the arc inside the learning network not in the actual network. The FN is the arc in the actual however, not in the learning network. The result of the confusion matrix for the First proposed method compared with default Simulated Annealing and default Greedy search are shown in Table 4.16. From the table, we can compute the evaluation criteria values. The first one is the sensitivity calculated by using the Equation (2-51) and shown in Figure 4.1. It can be seen that show that the PIO produces better sensitivity values than the Simulated Annealing and Greedy Search in most datasets. Figure 4.2 shows the accuracy of PIO, Simulated Annealing and Greedy search, which are calculated as explained in the section (2.5.2.1). This criterion present demonstrates that the proposed method is better than Simulated Annealing and Greedy search in the most dataset, as shown in Figure 4.2. Similarly, the PIO method in the most dataset has higher accuracy values than the Simulated Annealing and Greedy algorithms, as shown in Figure 4.2. The proposed PIO Learning Algorithm performs well in finding the appropriate structure. As a result, from the point of prediction accuracy, the Iterative PIO algorithm is the best algorithm compared to other algorithms in most datasets, and from the point of construction times also the PIO is better than the other algorithms. The proposed PIO Learning Algorithm performs well in finding the appropriate structure and presented a relatively low time complexity because the global search decreases by half the number of pigeons.

The F1- score, Precision, and Recall are used to evaluate the performance of the proposed algorithm. In these circumstances, Precision is the number of directed edges

Table 4.16 Confusion Matrix of PIO, Simulated Annealing and Greedy

	Algorithm	TP	TN	FN	FP
Water	Simulated Annealing	24	15	27	22
	Greedy	25	15	26	21
	PIO	22	22	22	22
Static banjo	Simulated Annealing	28	2	7	6
	Greedy	17	6	22	21
	PIO	29	4	4	4
Alarm	Simulated Annealing	40	11	16	5
	Greedy	40	11	16	5
	PIO	40	9	14	14
Hail	Simulated Annealing	43	30	53	41
	Greedy	35	19	50	41
	PIO	46	25	45	45
hepar	Simulated Annealing	70	31	22	9
	Greedy	42	38	43	27
	PIO	63	35	25	25
win95pts	Simulated Annealing	81	99	130	130
	Greedy	88	85	109	109
	PIO	8	25	129	129
Andes	Simulated Annealing	204	55	188	108
	Greedy	28	97	212	65
	PIO	285	110	162	141
Lucas01	Simulated Annealing	12	4	4	0
	Greedy	12	5	5	0
	PIO	12	0	0	0

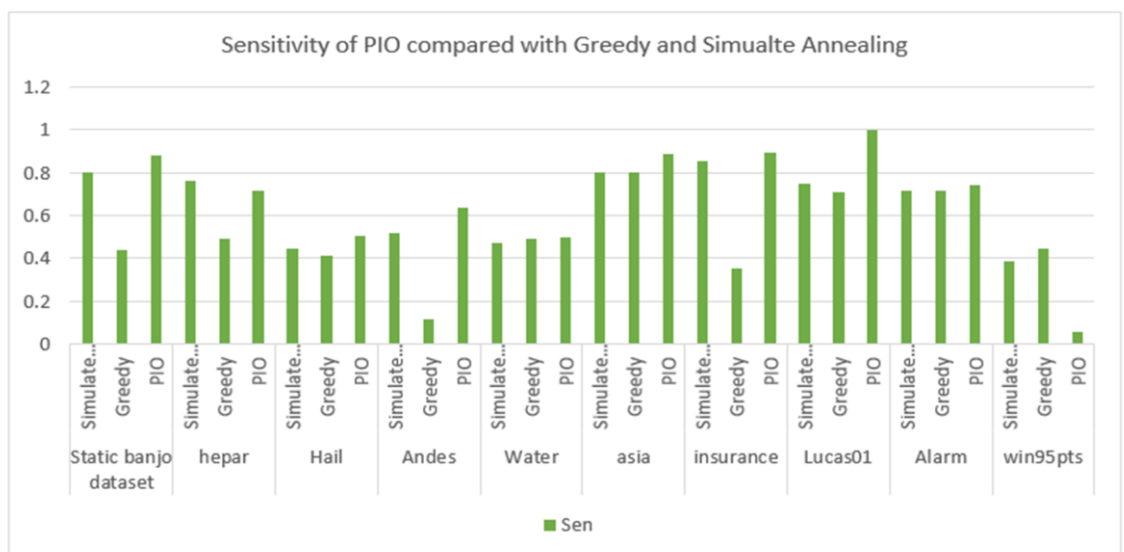


Figure 4.1 Sensitivity of PIO and Simulated Annealing and Greedy

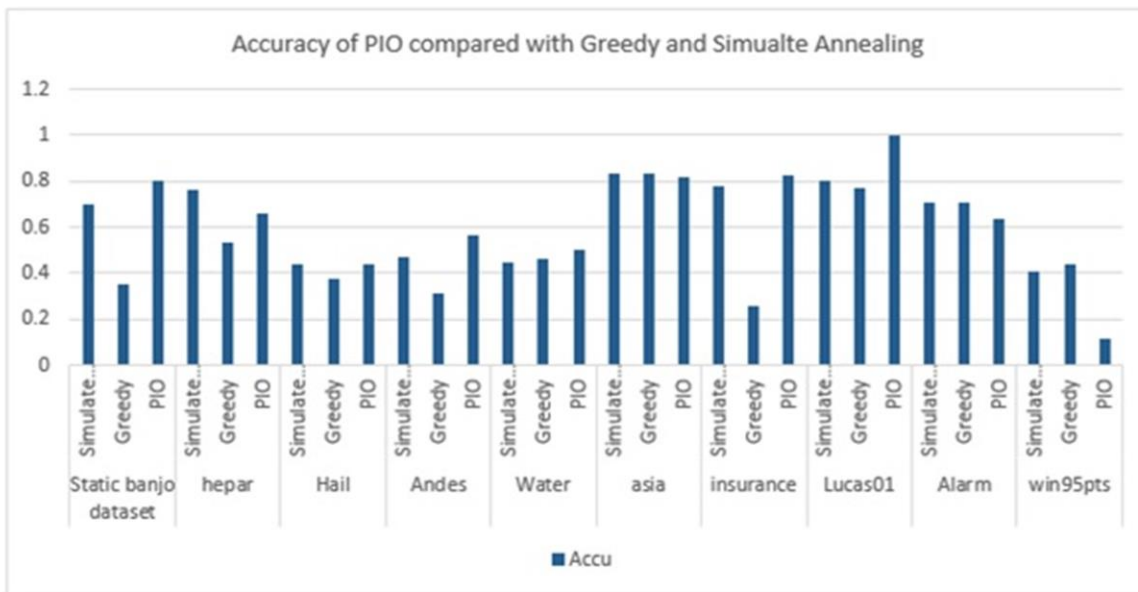


Figure 4.2 Accuracy of PIO and Simulated Annealing and Greedy

that are found correctly divided by the number of all edges in the expected BN. The Recall represents the division of the number of directed edges that are found by the number of edges in the actual BN. F1-score is the harmonic mean of precision and

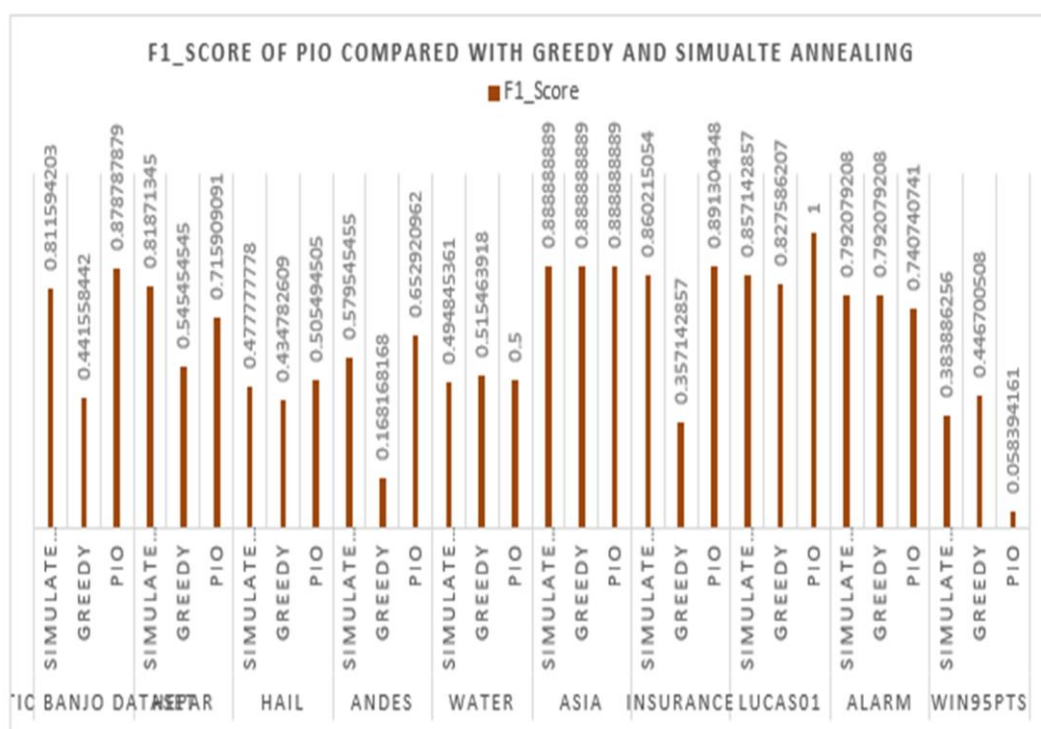


Figure 4.3 F1_Score of PIO and Simulated Annealing and Greedy

recall, which always vary between 0 and 1. An F1 score reaches its best value at 1 and the worst score at 0. Figure 4.3 shows the F1_scores of the PIO compared with

Simulated Annealing, and Greedy Search, which are calculated using the Equation (2-55) Figure 4.3 also shows that the proposed method is better than other mentioned algorithms in most data sets.

Figure 4.4 presents AHD for PIO and Simulated Annealing and Greedy search. The average Hamming distance calculated by

$$AHD = \frac{FP + FN}{TP + TN + FP + FN} \quad \text{Equation 4-1}$$

The proposed algorithm is also preferable based on the Hamming distances, which are always considerably lower than the ones 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 actual networks also they are directed entirely towards exploration rather than inference. Figure 4.4 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 have considered. Hamming distance is also commonly used for error correction.

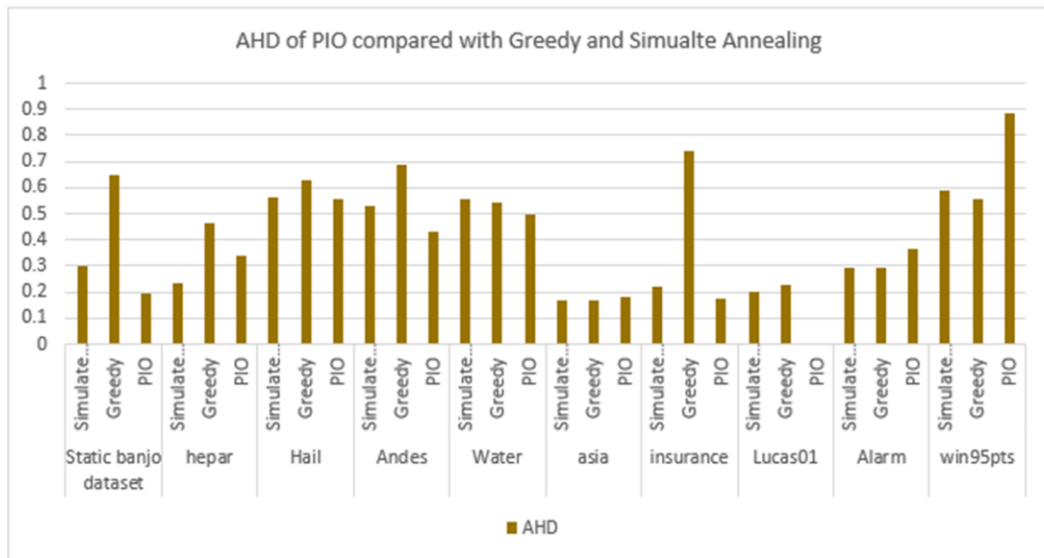


Figure 4.4 AHD of PIO and Simulated Annealing and Greedy

4.3.2 SECOND AND THIRD PROPOSED METHODS (BSA AND SAB)

This section presents the result of the confusion matrix for the Second and Third proposed methods (BSA and SAB) compared with Simulated Annealing. As shown in Table 4.17, the proposed methods are very close or better than simulated Annealing in most datasets.

Table 4.17 Confusion matrix of BSA, SAB, and Simulated Annealing

	Methods	TP	TN	FN	FP
Water	Simulated Annealing	24	14	28	23
	BeeLocal SimGlobal	24	17	25	20
	BeeGlobal SimLocal	24	18	24	18
Static banjo	Simulated Annealing	28	2	7	6
	BeeLocal SimGlobal	30	2	5	4
	BeeGlobal SimLocal	29	2	6	5
Alarm	Simulated Annealing	40	11	16	5
	BeeLocal SimGlobal	40	11	16	5
	BeeGlobal SimLocal	40	11	16	5
Hail	Simulated Annealing	46	32	52	42
	BeeLocal SimGlobal	46	34	54	42
	BeeGlobal SimLocal	45	33	54	43
hepar	Simulated Annealing	69	27	81	124
	BeeLocal SimGlobal	72	33	18	9
	BeeGlobal SimLocal	76	29	18	10
Andes	Simulated Annealing	244	81	174	103
	BeeLocal SimGlobal	220	58	175	91
	BeeGlobal SimLocal	238	53	152	69

From the confusion matrix as shown in the table 4.17, we can calculate the following criteria (Positive Predictive Value(PPV), Sensitivity(Sen), Accuracy(Acc), F1_Score, and Average Hamming Distance (AHD)). The PPV calculated by using the Equation:

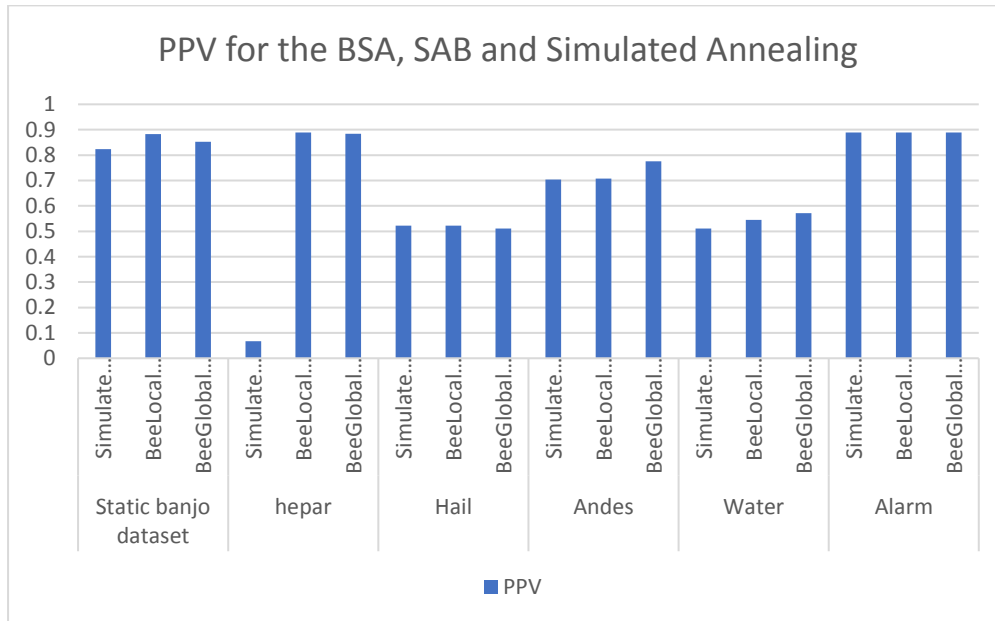


Figure 4.5 PPV for BSA, SAB, and Simulated

$$\text{positive predictive value} = \frac{TP}{TP+FP}$$

Equation 4-2

As the results in Figure 4.5 shows, the proposed methods give better ppv values than Simulated Annealing. The sensitivity values calculated using Equation (2-63) are shown in Figure 4.6. The sensitivity measures the proportion of actual positives that correctly identified. Figure 4.6 demonstrates that the proposed methods (BSA and SAB) are better than the Simulated Annealing. Figure 4.7 shows the Accuracy of the BSA, SAB, and Simulated Annealing; they calculated by using the details of the section (2.5.2.1). The Accuracy result in this figure shows that the BSA and SAB have better values than Simulated Annealing for the most dataset. The F1_score and Average Hamming Distance also calculated using equations (2-55 and 4-1) respectively, the results shown in Figures 4.8 and 4.9. demonstratarate that the BSA and SAB values for most data sets are better than Simulated Annealing.

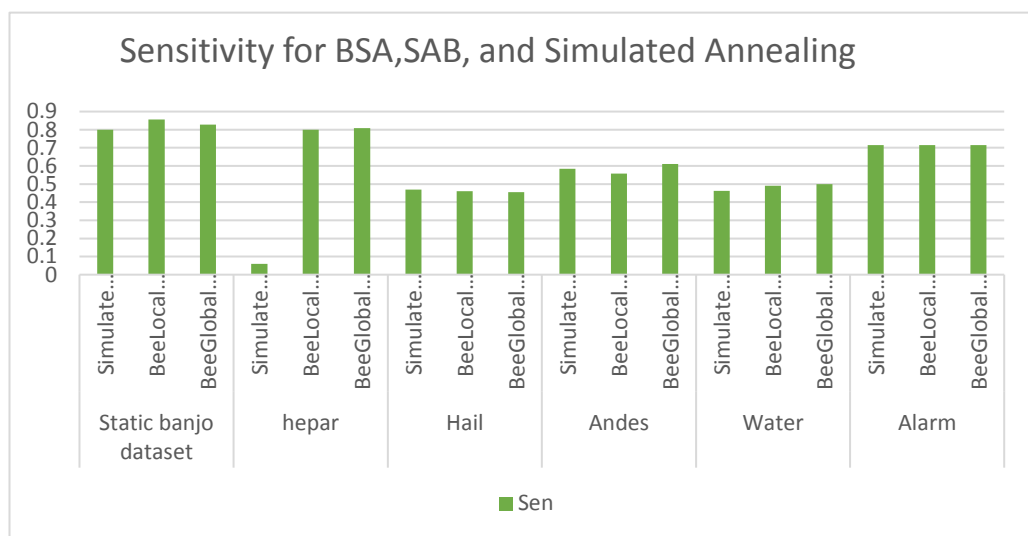


Figure 4.6 Sensitivity for BSA, SAB, and Simulated Annealing

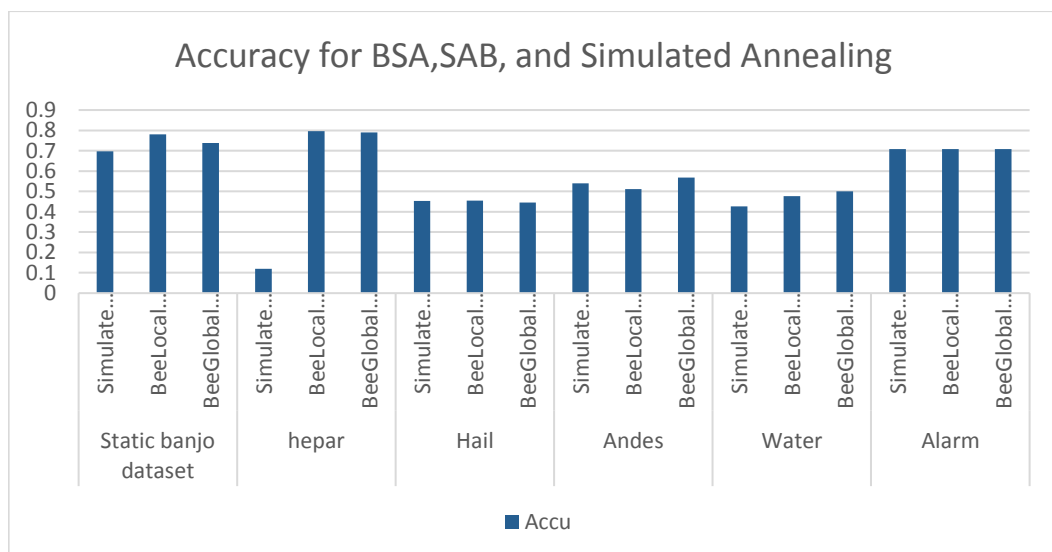


Figure 4.7 Accuracy for BSA, SAB, and Simulated Annealing

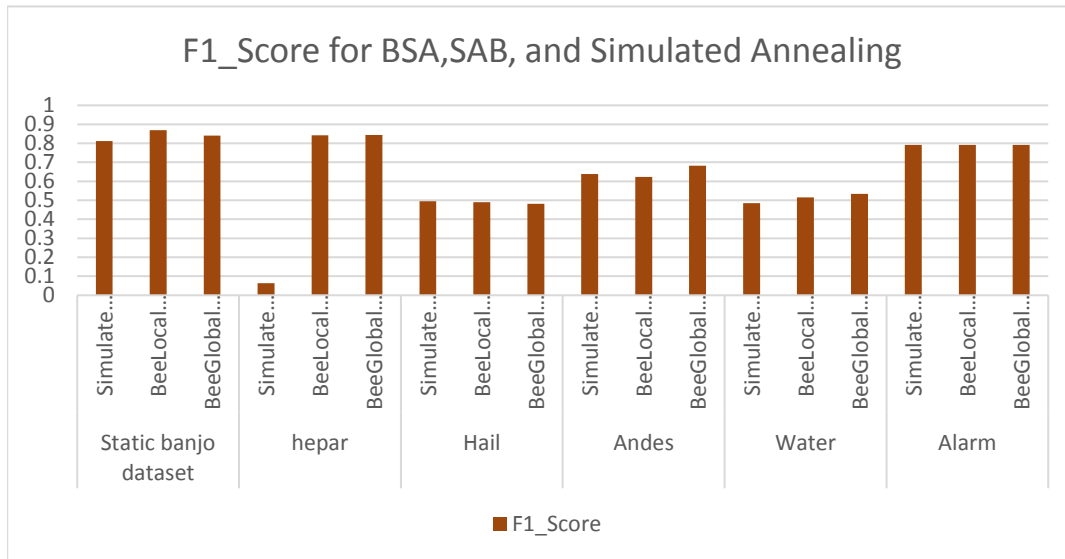


Figure 4.8 F1 Score for BSA, SAB, and Simulated Annealing

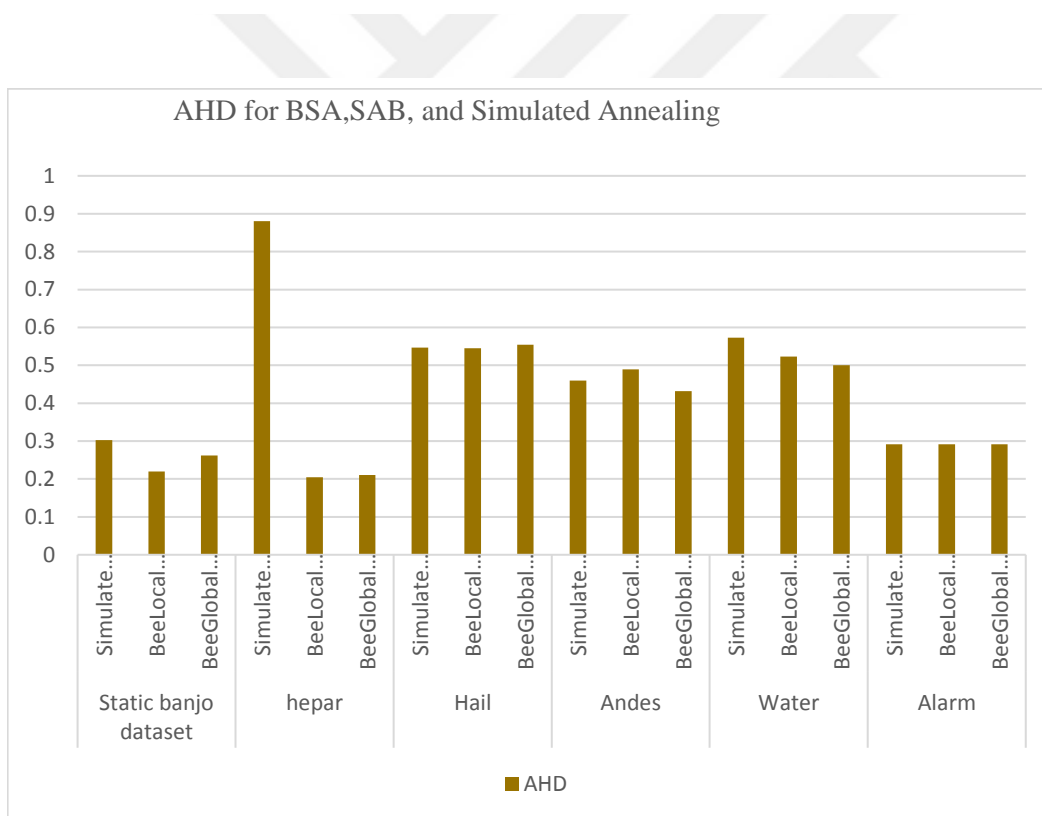


Figure 4.9 AHD for BSA, SAB, and Simulated Annealing

4.3.3 FOURTH AND FIFTH PROPOSED METHODS (BLGG AND BGGL)

In this part, the evaluation of the fourth and fifth (BLGG and BGGL) proposed methods using the confusion matrix calculation are presented, and the results are compared with the default greedy search method. As shown in Table 4.18 that the proposed method is better than the greedy search in most of the datasets.

Table 4.18 Confusion matrix of BLGG, BGGL, and Greedy

dataset	Methods	TP	TN	FN	FP
Water	Greedy	23	17	26	21
	BeeLocal Greedy Global	24	16	26	21
	BeeGlobal Greedy Local	24	18	24	18
Static banjo	Greedy	18	3	18	17
	BeeLocal Greedy Global	19	1	15	14
	BeeGlobal Greedy Local	29	1	5	4
Alarm	Greedy	35	15	25	18
	BeeLocal Greedy Global	37	16	24	16
	BeeGlobal Greedy Local	40	21	26	18
Hail	Greedy	35	20	51	38
	BeeLocal Greedy Global	35	21	52	41
	BeeGlobal Greedy Local	37	18	47	35
hepar	Greedy	45	36	42	28
	BeeLocal Greedy Global	47	37	39	24
	BeeGlobal Greedy Local	69	33	21	8
Andes	Greedy	34	106	197	50
	BeeLocal Greedy Global	39	99	199	52
	BeeGlobal Greedy Local	39	99	199	51

In this part, we present the Positive Predictive Values (PPV) in Figure 4.10, Sensitivity(Sen) values in Figure 4.11, and Accuracy values in Figure 4.12. Figure 4.13 shows the F1_scores for BLGG, BGGL, and greedy search. The section (2.5.2.3)

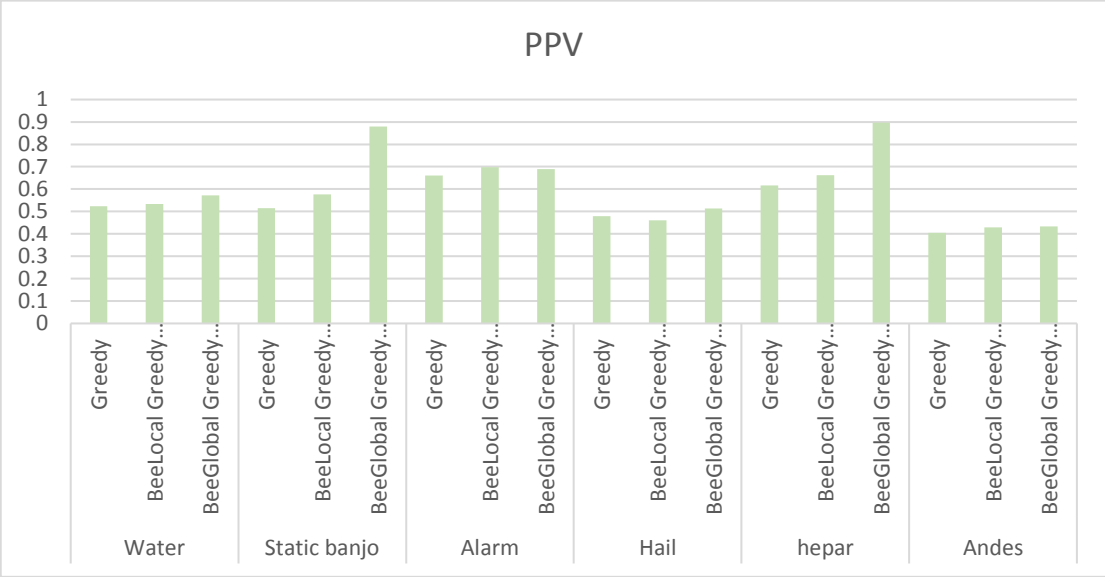


Figure 4.10 PPV for BLGG, BGGL and Greedy

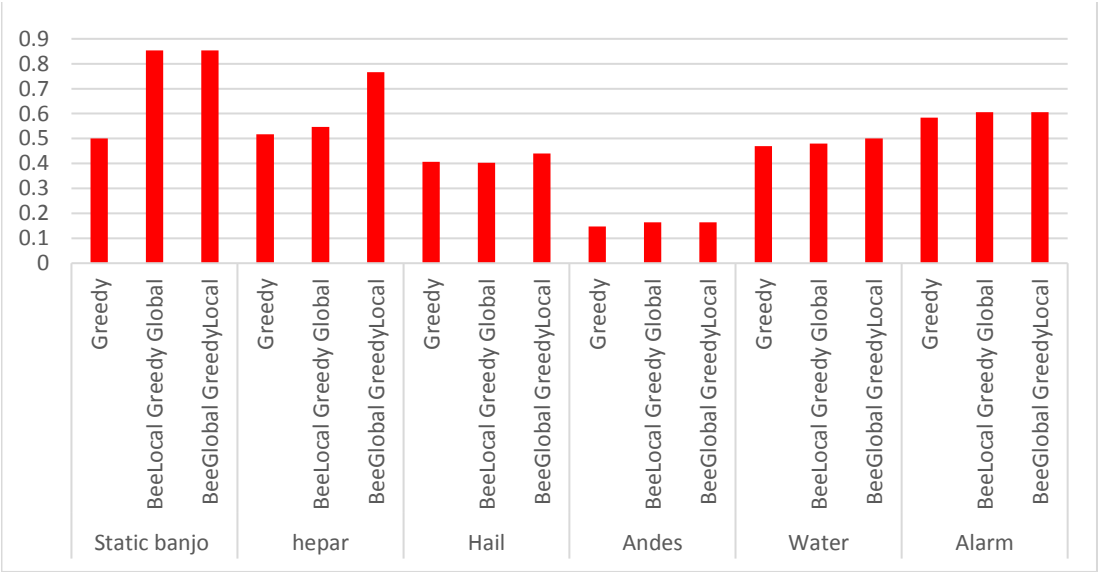


Figure 4.11 Sensitivity for BLGG, BGGL and Greedy

describes in detail the definition and calculation of F1_score. The results of Figure 4.13 show the proposed methods had an excellent F1_score result compared with the default greedy search. The last criterion presented in this section is Average Hamming

Distance (AHD). The number of a different edge between the learned network and the original network is called Hamming Distance.

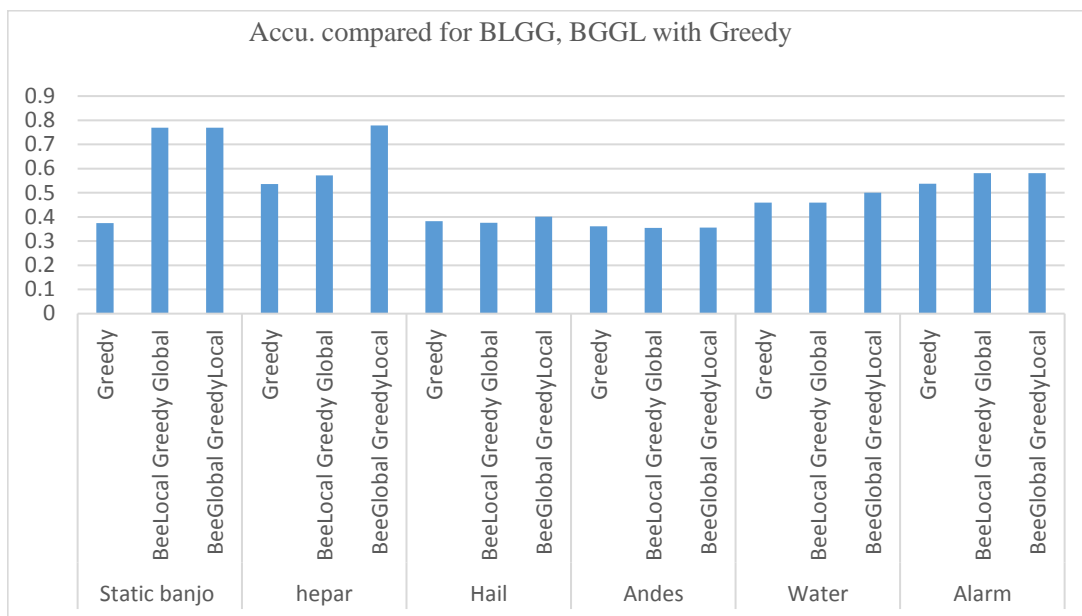


Figure 4.12 Accuracy for BLGG, BGGL and Greedy

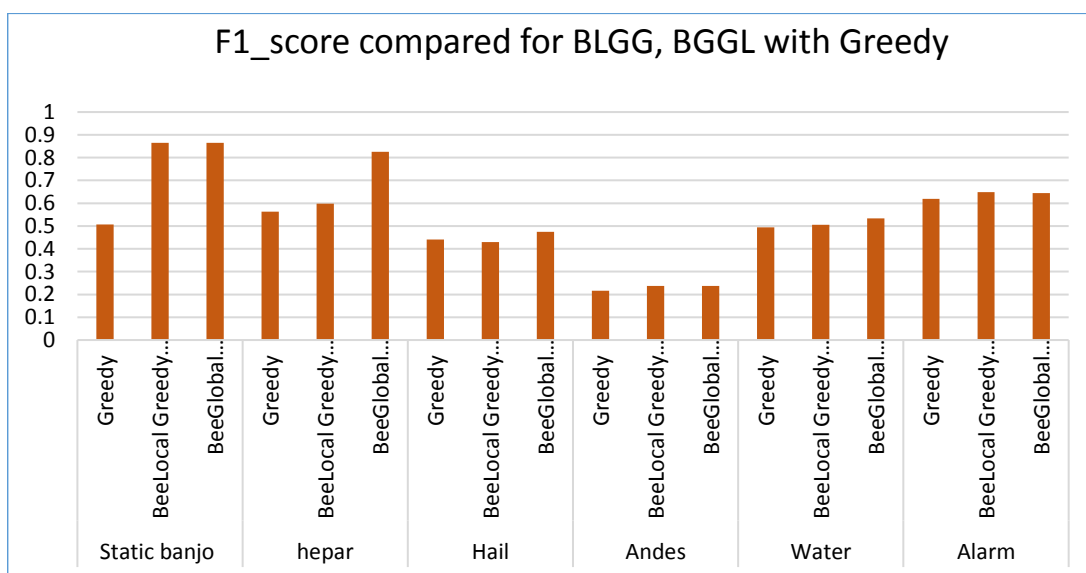


Figure 4.13 F1_Score for BLGG, BGGL and Greedy

Figure 4.14 shows the result of the AHD for BLGG, BGGL and greedy. The result shows the proposed methods decreased the AHD based on the greedy search. Depended on Figure 4.14 by using the proposed method can reduce the AHD.

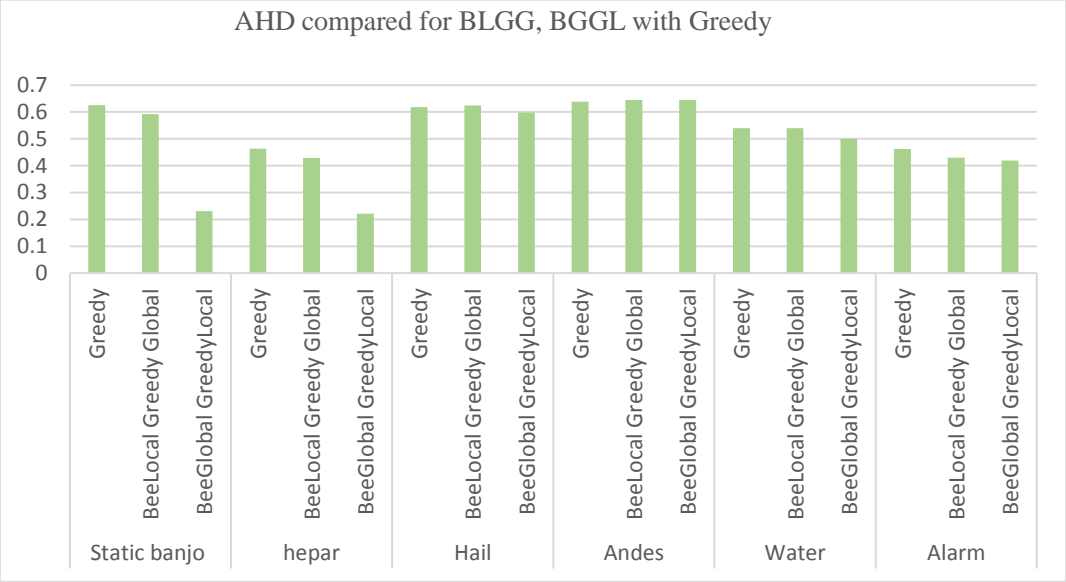


Figure 4.14 AHD for BLGG, BGGL and Greedy

4.3.4 SIXTH PROPOSED METHODS (ESWSA)

To evaluate algorithm performance (ESWSA), a standard assessment technique has been utilized by using probabilistic datasets from popular Bayesian network benchmarks. We investigated the properties of the proposed algorithm in several datasets. We compared the results with Simulated Annealing and Greedy Search methods by using corresponding metrics for the datasets. After defining the parameters of the ESWSA algorithm, local and global search are applied to the datasets. In EWSA, we use three parameters to control exploration performance. We randomly choose the first parameter inertia weight (w^t), which shows the speed of inertia at the current iteration. The second one is switching probability p , which is implemented as a constant parameter i.e., the value remains constant during the entire search. This choice suggests that local and global water exploration can change based on the value of the parameter p . For the experiments, we use $p=0.7$. It also fixes the last ones, t_{max} and population size N parameters of ESWSA optimization. As shown in Table 4.19, the result has shown the proposed method is better than the greedy search in most of the datasets.

To evaluate the success of structure discovery, we have computed the confusion matrix for each data set and its known network structure and calculated the metrics TP, TN, FN, and FP for each network per algorithm and the criteria; Sensitivity (SE), Accuracy (Acc), F1_Score, and AHD.

The Sensitivity results for ESWSA, Simulated Annealing and Greedy, are shown in

Table 4.19 Confusion matrix of ESWSA, Simulated Annealing, and Greedy

dataset	Methods	TP	TN	FN	FP
adult	Simulated Annealing	8	22	32	31
	Greedy	10	20	18	17
	ESWSA	8	21	31	31
Child	Simulated Annealing	23	2	4	4
	Greedy	16	15	24	23
	ESWSA	23	2	4	4
insurance	Simulated Annealing	40	6	6	5
	Greedy	21	7	38	35
	ESWSA	41	6	5	5
Water	Simulated Annealing	24	15	27	22
	Greedy	25	15	26	21
	ESWSA	22	20	24	24
Static banjo	Simulated Annealing	28	2	7	6
	Greedy	17	6	22	21
	ESWSA	29	4	4	4
Alarm	Simulated Annealing	40	11	16	5
	Greedy	37	16	24	15
	ESWSA	40	8	13	13
Hail	Simulated Annealing	43	30	53	41
	Greedy	35	19	50	41
	ESWSA	46	19	39	39
hepar	Simulated Annealing	70	31	22	9
	Greedy	42	38	43	27
	ESWSA	74	35	14	14
win95 pts	Simulated Annealing	81	88	119	119
	Greedy	8	22	126	126
	ESWSA	88	85	109	109
Andes	Simulated Annealing	204	55	188	108
	Greedy	28	97	212	65
	ESWSA	285	88	140	140

Figure 4.15. The proposed method produces better values than the Simulated Annealing and Greedy in the different datasets.

Similarly, the proposed method in most dataset has high accuracy values than the Simulated Annealing and Greedy algorithms, as

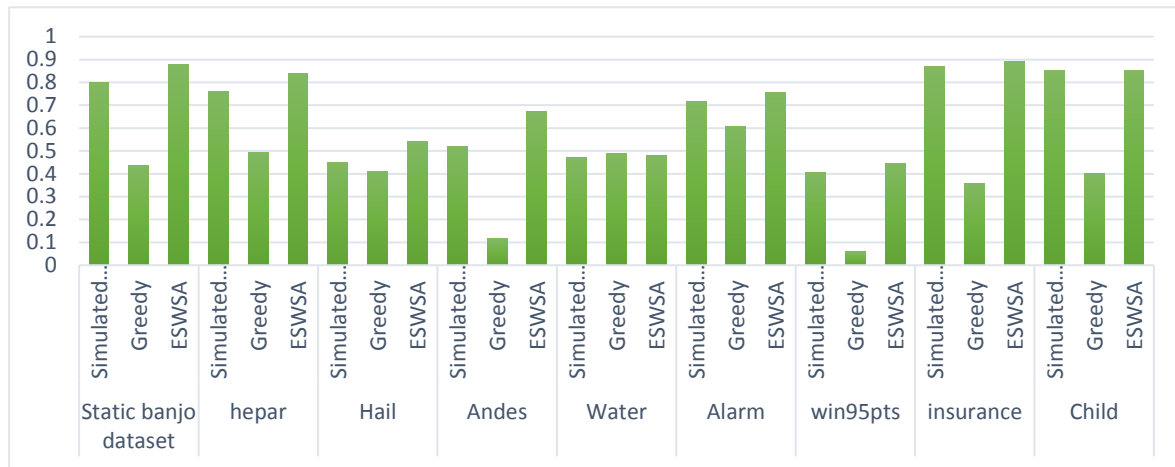


Figure 4.15. Sensitivity for ESWSA, Simulated Annealing and Greedy.

shown in Figure 4.16. The proposed ESWSA Learning Algorithm performs well in finding the appropriate structure. As a result, from the point of prediction accuracy, the Iterative ESWSA algorithm is the best algorithm compared to other algorithms in most datasets, and from the point of construction times also the ESWSA is better than the other algorithms. For performance metrics, in addition to the best score in Bayesian results, we used F1 as a metric of the model's accuracy.

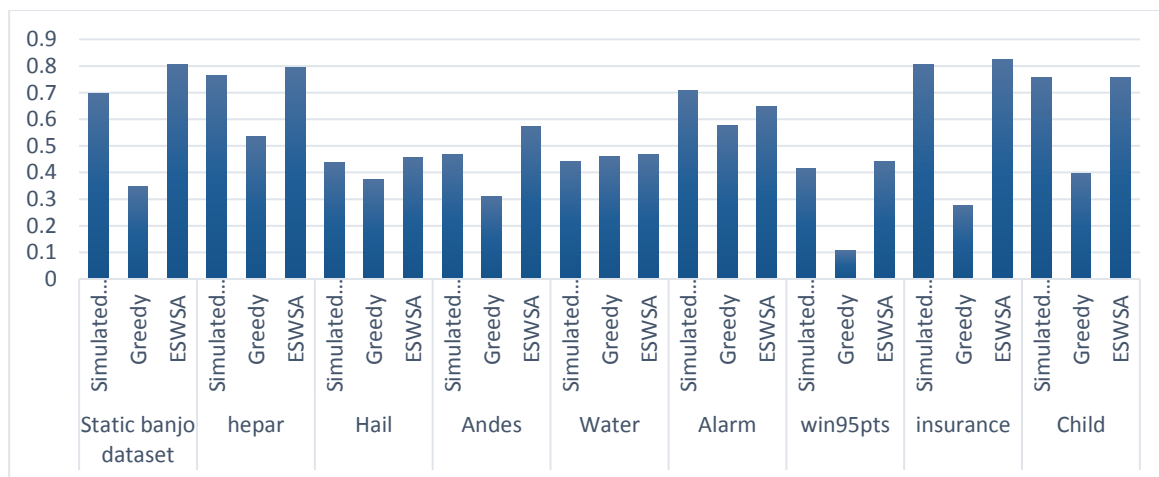


Figure 4.16. Accuracy for ESWSA, Simulated Annealing and Greedy.

The F1- score, Precision, and Recall are used to evaluate the performance of the proposed algorithm. In these circumstances, Precision is the number of directed edges that are found correctly divided by the number of all edges in the expected BN. The Recall represents the division of the number of directed edges that are found by the number of edges in the actual BN. We know that F1 is the harmonic average of accuracy and Recall. Figure 4.17 presents a comparison between the ESWSA, Simulated Annealing, and Greedy search. As shown in Figure 4.17, the proposed methods are successful than the 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 from the Hamming distances, which are always considerably lower than the ones 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 also are directed entirely towards exploration rather than inference.

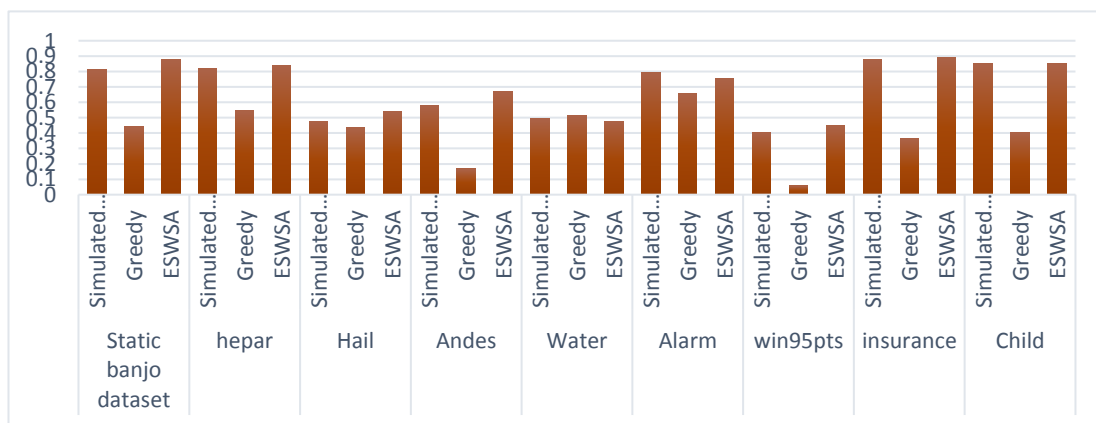


Figure 4.17. F1 Score for ESWSA, Simulated Annealing Greedy.

Figure 4.18 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 have considered.

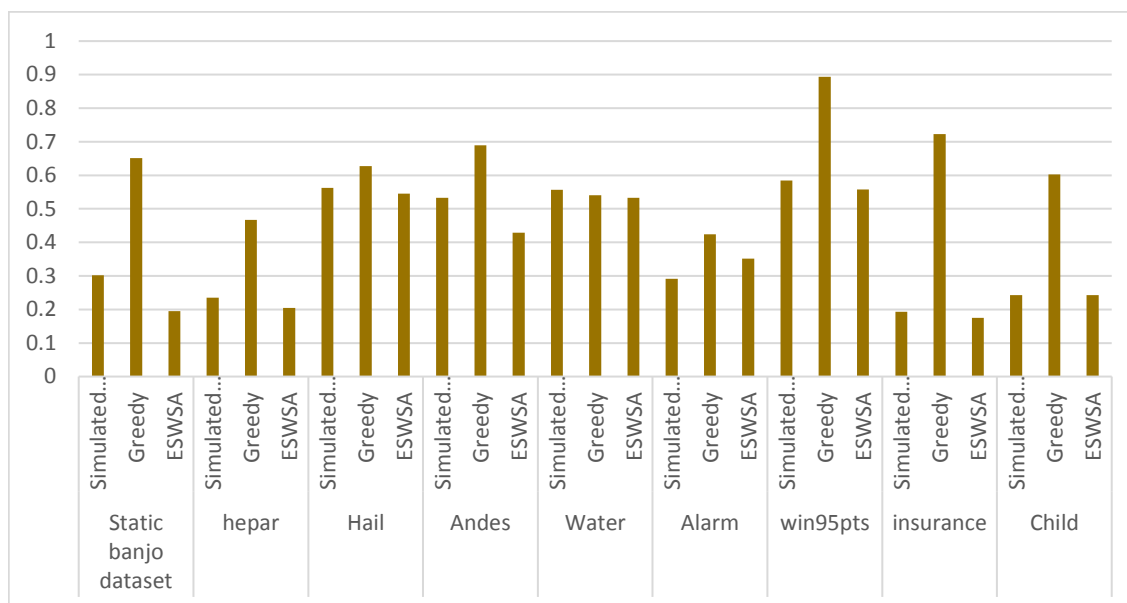


Figure 4.18. AHD for ESWA, Simulated Annealing Greedy.

4.3.5 COMPARISONS AMONG PROPOSED METHODS.

To evaluate the success of structure discovery, the confusion matrix has been computed for each data set and its known network structure. The metrics TP, TN, FN, and FP, have been calculated for each network per algorithm as well as the criteria; Sensitivity (SE), Accuracy (Acc), F1 Score, and AHD. Table 4.20 presents the result of the proposed method using dataset mentioned in the section (4.1). Table 4.20 shows the result of a confusion matrix for the proposed methods.

Figures 4.19 and 4.20 show the Sensitivity and Accuracy of the proposed methods. In most datasets, the ESWA is better than other proposed methods, the second-best is PIO. Figure 4.21 shown the F1_Score of the proposed methods. The result demonstrated that in most datasets, the Hybrid between Bee and Simulated Annealing is better than other proposed methods. Figure 4.22 shown the AHD of the proposed methods. Figure 4.22 shows that the ESWA is better than other proposed methods.

Table 4.20 Confusion matrix of All proposed methods

dataset	Methods	TP	TN	FN	FP
Water	ESWSA	22	20	24	24
	PIO	22	22	22	22
	BeeLocal SimGlobal	24	17	25	20
	BeeGlobal SimLocal	24	18	24	18
	BeeLocal Greedy Global	24	16	26	21
	BeeGlobal Greedy Local	24	18	24	18
Static banjo	ESWSA	29	4	4	4
	PIO	29	4	4	4
	BeeLocal SimGlobal	30	2	5	4
	BeeGlobal SimLocal	29	2	6	5
	BeeLocal Greedy Global	19	1	15	14
	BeeGlobal Greedy Local	29	1	5	4
Alarm	ESWSA	40	8	13	13
	PIO	40	9	14	14
	BeeLocal SimGlobal	40	11	16	5
	BeeGlobal SimLocal	40	11	16	5
	BeeLocal Greedy Global	37	16	24	16
	BeeGlobal Greedy Local	40	21	26	18
Hail	ESWSA	46	19	39	39
	PIO	46	25	45	45
	BeeLocal SimGlobal	46	34	54	42
	BeeGlobal SimLocal	45	33	54	43
	BeeLocal Greedy Global	35	21	52	41
	BeeGlobal Greedy Local	37	18	47	35
hepar	ESWSA	74	35	14	14
	PIO	63	35	25	25
	BeeLocal SimGlobal	72	33	18	9
	BeeGlobal SimLocal	76	29	18	10
	BeeLocal Greedy Global	47	37	39	24
	BeeGlobal Greedy Local	69	33	21	8
Andes	ESWSA	285	88	140	140
	PIO	285	110	162	141
	BeeLocal SimGlobal	220	58	175	91
	BeeGlobal SimLocal	238	53	152	69
	BeeLocal Greedy Global	39	99	199	52
	BeeGlobal Greedy Local	39	99	199	51

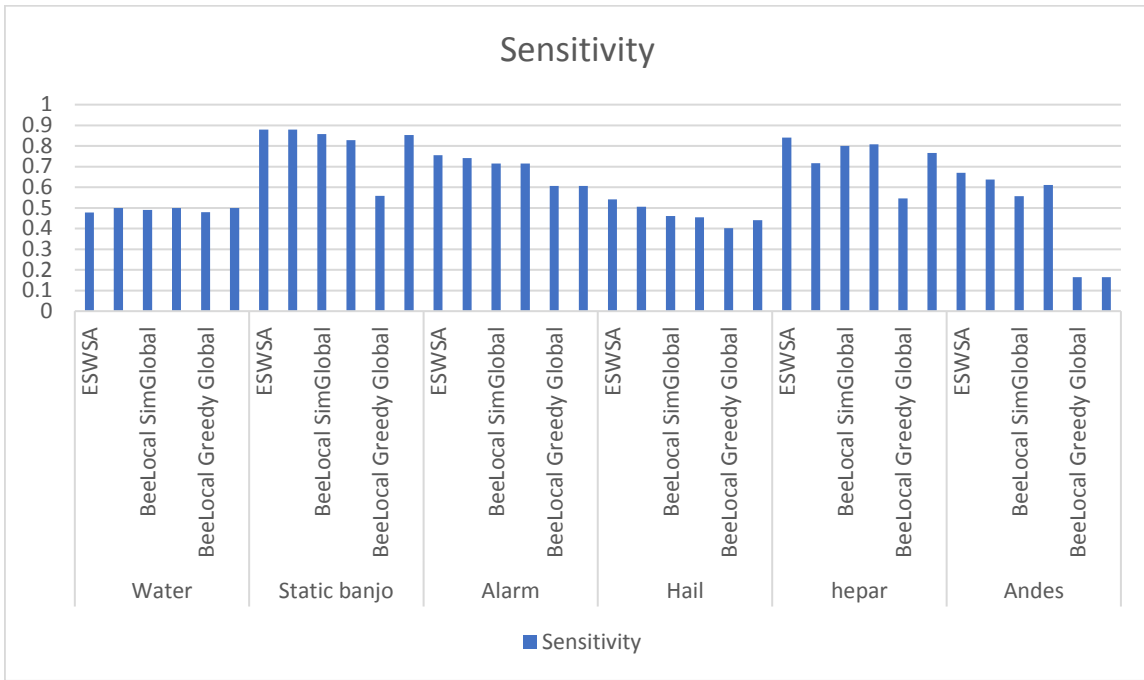


Figure 4.19 Sensitivity of the proposed methods

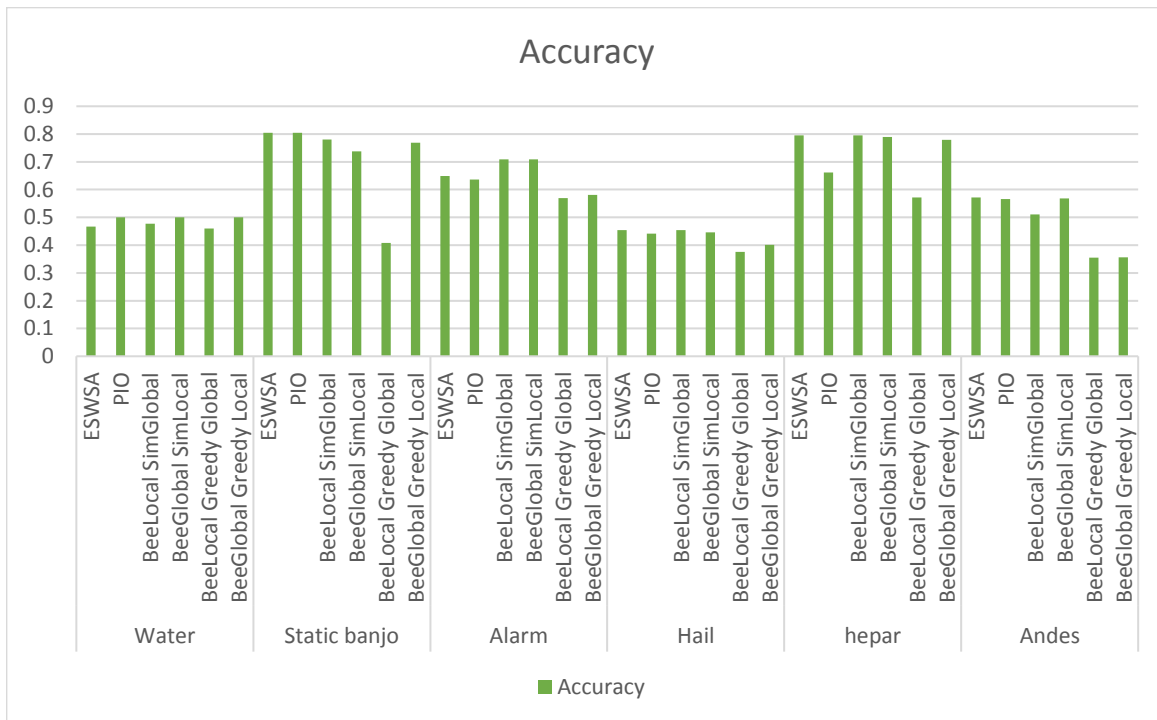


Figure 4.20 Accuracy of the proposed methods

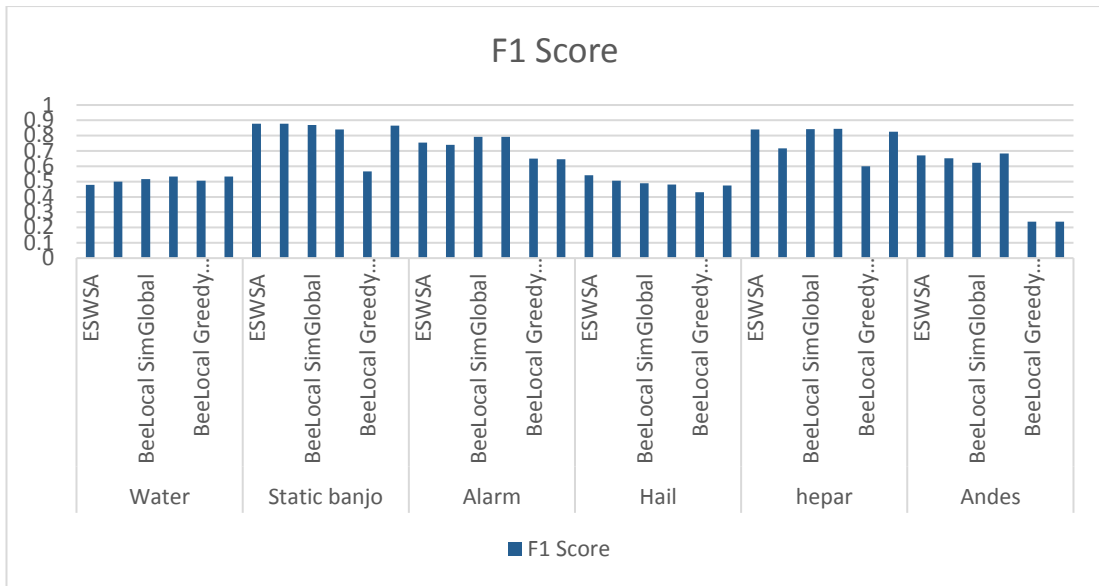


Figure 4.21 F1_Score of the proposed methods

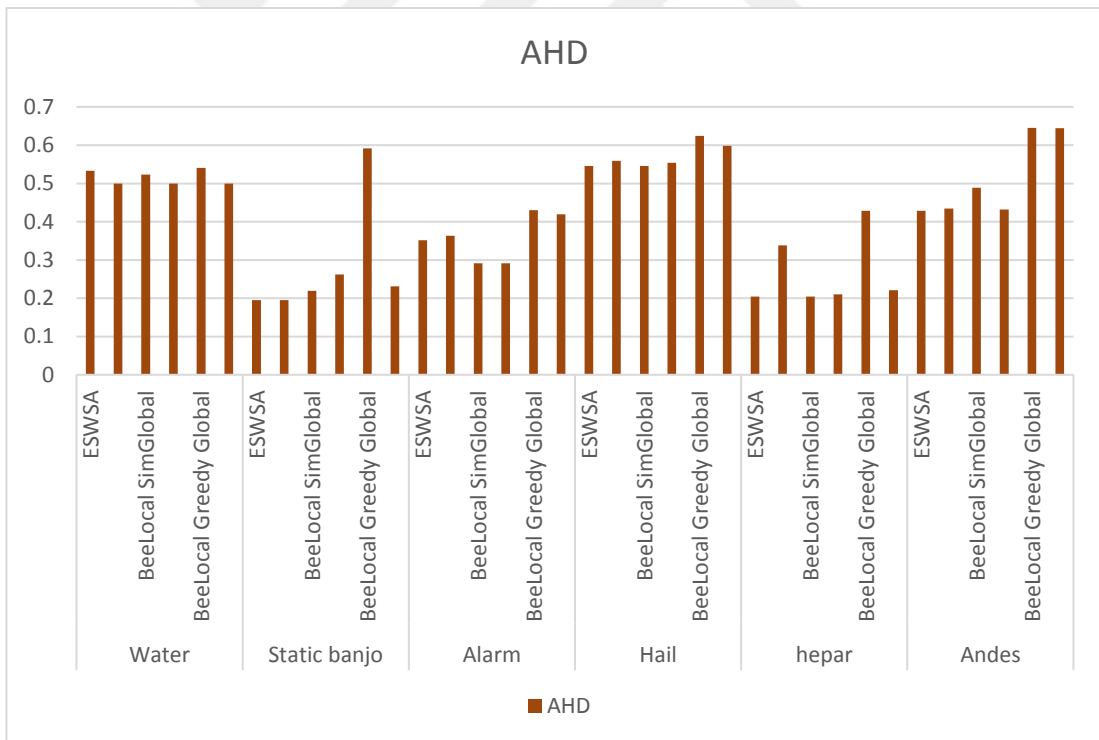


Figure 4.22 AHD of the proposed methods

CHAPTER 5

CONCLUSIONS AND FUTURE RESEARCH

5.1 CONCLUSIONS

The learned structure of Bayesian networks can be used for guiding future action and understanding the causal mechanisms of a system if structure learning algorithms can learn the fundamental structure of the network, and if certain assumptions are met.

In this dissertation, we studied the structure learning of Bayesian networks based on score and search method using the BDeu score function. Swarm intelligence has always been an inspiration for the researcher. We attempt five algorithms for this dissertation based on a meta-heuristic search and also compared the results with the default Simulated Annealing and Greedy search. The Pigeon Inspired Optimization has opened a new horizon for the researchers and it will provide a platform for future research in this field. The PIO has a robust problem-solving potential that can be applied in fields like the travelling salesman problem, Polynomial identity testing, the shortest path problem, and other optimization problems. However, any optimization technique cannot say the best or worst based on one application. For some applications, one may be better than the others.

The Bees Algorithm is a swarm-based algorithm that mimics the natural food foraging behaviour of honey bees. The algorithm involves both random exploration of the solution space and more focused exploitation of promising local search sites. A basic version of the Bees Algorithm has been applied to function optimization problems. It can characterize BA as being a distributed, stochastic search method based on the communications of a colony of ‘artificial Bees’, mediated by ‘artificial waggle dances’. The waggle dance serves as a distributed information used by the Bees to construct solutions to the problem under consideration.

The Pigeon Inspired Optimization (PIO) is the first proposed method to structure learning Bayesian network. The comparative simulation results show that the proposed PIO algorithm is a workable and effective algorithm to structure learning Bayesian network while compared with Simulated Annealing and Greedy search. It provides an alternate approach of problem-solving different from the traditional techniques in use. We can describe PIO as comprising, a stochastic search technique depending on the

information accumulated and shared by pigeons. A PIO is a usual method for searching a discrete solution space. The PIO could miss any promising regions of the search space that the local/global search and switching mechanism operated earlier. A PIO is a common framework that can adjust to suit for any application region. The PIO concentration control to optimal global by allowing to fly in short solution space, the probability in the position to be the right solution space by an extra control through pigeon parameter to leave out the different scale. Our proposed method has more competence for searching, and it can detect good structure solutions, calculate higher score function and excellent approximation to the network. The algorithms improve the global search and lead rapidly to global convergence.

This thesis incorporates enhanced versions of Bees Algorithm(BA), first by implementing simulated annealing approach to the selection of local search sites, and then BA for global search, also, BA as local and Simulated Annealing as global search as the second and third proposed method. We propose BA with a Greedy search in fourth and fifth proposed methods, by search intensification through controlling of the random search by BA after the local search phase. This thesis presents the results obtained from NP-hard problems to show the robustness and speeding up the ability of the Bees Algorithm variants.

There is a vast literature to solve NP-hard problems, but there are some problems associated with these methods:

1. Some algorithms use fixed-length problem representation; this limitation will affect the problem solutions when those problems get larger dimensions.
2. Most algorithms require a large population to attain an optimal solution due to the inconsistency in using inappropriate problem-specific local search mechanisms.

An exchange neighbour operator is used with structure learning the Bayesian network, because of its simplicity and robustness. It uses a modification to the stepping stone method with the classical transportation problem to establish an appropriate local search operator which has the strength required to hold the problem constraints.

From our work, several conclusions can be drawn for applying Bee in structure learning Bayesian networks:

1. All the BA variants include strong exploitation of the best solutions found during the search. The most successful ones add specific features to avoid premature stagnation of the search. The main differences between the various BA extensions comprise the technique used to control the search process. Experimental results show that for structure learning Bayesian network, these variants make a better and faster performance gains than classical BA.

2. BA is a general method for searching discrete solution space in a way like Bees foraging process. BA could miss some promising areas of the search space if the local/global search switching mechanism performed earlier than it's supposed to be. SBA presents an advantage over BA by equipping with a switching strategy that allows the systematic determination of the transition for the local to global search, this avoids computationally expensive earlier transition in advance and makes up a major benefit of the proposed methods. It is earlier switching in BA required more iteration and time to get the algorithm back in a track to the promising optimal or near-optimal search space if it gets back at all.

3. BSA and SAB constitute a general framework that can change to suit any application area. The simulated annealing method suffers from slow convergence for its random nature of movements. Simulated annealing also suffers from the difficulty in getting some required accuracy, although it may approach the neighbourhood of the global minimum. By manipulating the cooling schedule of simulated annealing, BSA and SAB practitioners can exercise control over convergence. Bees algorithm employees no such concept of cooling, and its convergence is not controlled. Convergence control in BSA and SAB provides rapid convergence to global extremum by allowing Bees to move to less profitable solution space probabilistically to get nearer to more profitable solution space that provides the speedier version of SA and more controlled BA since the extra control provided by introducing a temperature allows separating problems on different scales.

4. BLGG and BGGL combine the advantages of both BA and Greedy search into a hybrid algorithm which applies and has a better searching ability and power to reach a near-optimal solution by achieving an appropriate balance between the exploitation of the search experience gathered so far, and the exploration for unvisited or unexplored search space regions. It leads to the development of a fast convergence controlled method to solve complicated types of optimization problems.

5. When using local search, it suffices to apply a small constant number of Bees to achieve high performance. Experimental results suggest that in this case, the role played by heuristic information becomes much less important. Besides the choice of the right parameters made, that shows the usage of the smaller population achieves faster convergence and more time reduction.

6. Combinatorial optimization problems arise in many practical and theoretical problems. Often, these problems are very hard to solve to optimality. Structure learning Bayesian network was the combinatorial optimization problem to attack by BA and its variants. Under low conditions (small-sized instance), all the algorithms tested have similar performance. Here, it is hard to assess if an algorithm is significantly better than another.

7. The Bee Algorithm is a swarm-based algorithm that imitates the natural food foraging behaviour of honey bees. The algorithm includes both random explorations of the solution space and more focused exploitation of promising local search sites. A basic version of the Bees Algorithm is used in optimization problems. It can characterize BLGG as being a divided, stochastic search method based on the communication of a colony of 'artificial Bees', arbitrated by 'artificial waggle dances. The waggle dance works as a distributed information used by the Bees to construct solutions to the problem under consideration. BLGG is a common method for searching discrete solution space in a way related to Bees foraging process. BLGG can drop any promising areas of the search space if they do earlier the local/global search switching mechanism than it's assumed to be. GLBG is a common core that can adjust to suit any application area. By managing the neighbourhood schedule of greedy, the GLBG practitioner can apply control over convergence. Bees algorithm employes no such idea of the neighbourhood and its concentration is not checked. Concentration check in GLBG presents speedy concentration to the global extreme by providing Bees to move to tiny beneficial solution space to get nearer to extra valuable solution space that presents a quicker version of greedy and more controlled Bee algorithm. The proposed method has a higher performance for searching; this means it can get great structure solution, calculate higher score function and examine the network proposed. Develops the solution for local search to the global search and drive to the global convergence. The proposed approach can be viewed as the parallel implementation of Greedy, which shows the stability for parallel processing.

8. Distributed computing is a promising approach to meet the ever-increasing computational requirements. Scheduling is the most important issue in the distributed system because the effectiveness directly corresponds to the parallelization obtained. With inappropriate scheduling, mechanisms can fail to exploit the true potential of the distributed system. The schedule has the dual responsibility of minimizing the execution time of the resulting schedule and balancing the load among the processor. BA and all of its variants handle this problem by finding optimal and near-optimal schedules in a reasonable amount of time.

9. The Elephant Swarm Water Search Algorithm is an optimization technique that has adopted. For implementing an elephant, swarm behavior was a challenging task, therefore in status for the implement that response in real-time protocol improvement.

ESWSA is a method for searching a discrete solution space and it can be adjusted to suit for any application area. Concentration control in ESWSA presents quickened concentration to the global extremum by allowing the elephant to move to the shortest useful solution space. The proposed method has a higher ability for searching, which shows it can detect better structure solution, calculate higher score function and excellent approximation to the network structure and the results are more accurate. The algorithms improve the global search and lead rapidly to global convergence.

Considering the performance of elephant swarm water search in nature, a novel swarm-based heuristic search approach, called ESWSA proposed ESWSA to solving optimization score and search technique for structure learning Bayesian network. At the initial phase, the position and speed for each elephant will generate randomly. At the new stage of ESWSA, each elephant in the group has renewed the position also speed through using group information by updating the operator.

The Elephant Swarm Optimization technique presented during the research facilitates as an optimized search method to get an increasing enhanced system performance.

5.2 SUGGESTION FOR FUTURE WORK

- The algorithms presented in this thesis are still being developed; the next step would test them on a greater variety of problems with different parameters, stopping criteria, and problem-specific neighbourhood operators.
- Keep in mind that care must be taken in applications as much as implementation since different choices such as the selection for local search operator and other problem parameters determine the actual efficiency of any procedure (algorithm).
- It is controlling the randomization of the initial population by using seeds in the initial population to improve the BA and its variants further.
- Using another heuristic search to structure learning Bayesian network based hybrids between PIO and Bee algorithm, Bat Algorithm, hybrid PIO and Bat Algorithm.
- Using PIO for optimizing other problems like 4 mapping colour and job schedules.
- Elephant herding optimization algorithm (EHO) is one of the recent swarms' intelligence algorithms, which can be used for structure learning Bayesian networks.
- Another possibility is using ESWSA for the optimization algorithm for support vector machine parameter tuning.
- Apply ESWSA approach for the energy-based positioning problem
- Compare the elephant swarm optimization technique with other evolutionary computing based optimization techniques like Modified Interactive based Evolutionary Computing (MIEC) techniques with the behaviour of elephant herding idealize into clan updating operator and separating the operator.

REFERENCES

- [1] T. G. Dietterich, Machine learning In Nature Encyclopedia of Cognitive Science, London: Macmillan, 2003.
- [2] N. Friedman, K. Murphy, and S. Russell, "Learning the structure of dynamic probabilistic networks. Proceedings," in *14th Conference on Uncertainty in Artificial Intelligence (UAI-98)* , San Francisco, 1998.
- [3] Timo Koski and John M. Noble,, Bayesian Networks- An Introduction, Wiley series in probability and statistics,, 2009.
- [4] Olivier Pourret and Patrick Naim, Bayesian networks : a practical guide to applications, England: John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, 2008.
- [5] Cuicui Yang , Junzhong Ji , Jiming Liu , Jinduo Liu and Baocai Yin, "Structural learning of Bayesian networks by bacterial foraging optimization," *International Journal of Approximate Reasoning*, p. 69, 2016.
- [6] X. L. Li, "A particle swarm optimization and immune theory-based algorithm for structure learning of Bayesian networks," *international Journal of Database Theory and Application*, pp. 61-70, 2010.
- [7] Junzhong Ji, Hongkai Wei , Chunlian Liu, "An artificial bee colony algorithm for learning Bayesian networks," *Springer-Verlag Berlin Heidelberg*, December 2012.
- [8] D. Margaritis, "Learning Bayesian Network Model Structure from Data," Carnegie-Mellon University, Pittsburgh, PA. Available as Technical Report CMU-, 2003.
- [9] B. Yet, "Bayesian Networks for Evidence Based Clinical Decision Support," University of London, Queen Mary, London, 2013.
- [10] A. S. Fast, LEARNING THE STRUCTURE OF BAYESIAN NETWORKS WITH CONSTRAINT SATISFACTION, Massachusetts: PHD Thesis , Department of Computer Science, University of Massachusetts, February 2010, 2010.
- [11] Nathan Fortier, John Sheppard and Karthik Ganesan Pillai, "Abductive Inference using Overlapping Swarm Intelligence," in *IEEE Symposium on Swarm Intelligence*, 2013.
- [12] Wang Chun-Feng, Liu Ku, "A Novel Hybrid Method for Learning Bayesian Network," *Journal of Computers*, 2015.
- [13] J. Cowie, L. Oteniya, R. Coles , "Particle Swarm Optimisation for learning Bayesian Networks," Engineering and Physical Sciences Research Council., 2007.
- [14] L. M. de Campos and J. F. Huete., "Stochastic algorithms for searching causal orderings in Bayesian networks," in *Technologies for Constructing Intelligent Systems 2 - Tools*,, 2002.
- [15] Khalid M. Salama and Alex A. Freitas, "ABC-Miner: An Ant-Based Bayesian Classification Algorithm," 2012.
- [16] Junyi Li and Jingyu Chen, "A Hybrid Optimization Algorithm for Bayesian Network Structure Learning Based on Database," *Journal of Computers*, VOL.

9, 2014.

- [17] JI Jun-Zhong, ZHANG Hong-Xun, HU Ren-Bing and LIU Chun-Nian, "A Bayesian Network Learning Algorithm Based on Independence Test and Ant Colony Optimization," *ACTA AUTOMATICA SINICA.*, 2009.
- [18] Emmanuel S.A, George K.A, Francis T.O, "SAGA:A hybrid search algorithm for Bayesian Network structure learning of transcriptional regulatory networks," *Elsevier, Journal of Biomedical Informatics* 53, p. 27–35, 2015.
- [19] A. S. Hesar, "Structure Learning of Bayesian Belief Networks Using Simulated Annealing Algorithm," *Middle-East Journal of Scientific Research* , vol. 18, pp. 1343-1348, 2013.
- [20] P. Larraiaaga , M. Poza, "Structure Learning of Bayesian Networks by Genetic Algorithms," *Springer-Verlag Berlin Heidelberg GmbH*, 1996.
- [21] Changhe Yuan, Brandon Malone and Xiaojian Wu , "Learning Optimal Bayesian Networks Using A* Search," in *NSF grants IIS-0953723 and EPS-0903787, 21 IJCAI*, Barcelona, 2011.
- [22] Patrick O. Djan-Sampson and Ferat Sahin, "Structural Learning; of Bayesian Networks from Complete Data using the Scatter Search Documents," in *IEEE International Conference on Systems, Man and Cybernetics*, 2004.
- [23] Alireza Sadeghi Hesar, Hamid Tabatabaee and Mehrdad Jalali , "Structure Learning of Bayesian Networks Using Heuristic Methods," in *Int. Conference on Information and Knowledge Management*, 2012.
- [24] Changhe Yuan and Brandon Malone , "An Improved Admissible Heuristic for Learning Optimal Bayesian Networks," in *UAI Workshop on Causal Structure Learning*, 2012.
- [25] Xiannian Fan, Changhe Yuan Brandon Malone , "Tightening Bounds for Bayesian Network Structure Learning," in *Association for the Advancement of Artificial Intelligence*, 2014.
- [26] Safiye Sencer, Ercan Oztemel, Harun Taskin and Orhan Torkul, "Bayesian Structural Learning with Minimum Spanning Tree Algorithm," in *The World Congress in Computer Science, Computer Engineering, and Applied Computing*, 2013.
- [27] M. Koivisto and K. Sood,, "Exact Bayesian structure discovery in Bayesian networks," *J. Machine Learning. Research*, pp. 549-573, 2004.
- [28] Ajit Singh and Andrew W. Moore , "Finding optimal Bayesian networks by dynamic programming," CMU-CALD-05-106, Carnegie Mellon University, 2005.
- [29] D. Chickering, "Learning Bayesian Networks is NP-Complete," *Springer-Verlag*, 1996..
- [30] Kennedy, J., Eberhart, R.C., *Swarm Intelligence*, San Francisco: Morgan Kaufmann Publisher, 2001.
- [31] M.L. Wong and K.S. Leung, "An efficient data mining method for learning Bayesian network using an evolutionary algorithm-based hybrid approach," in *IEEE Trans. Evol. Comput.* 8(4), 2004.
- [32] Zhang, B., Duan, H., "Predator-prey pigeon-inspired optimization for UAV three-dimensional path planning," *Adv. Swarm Intell.* 8795, p. 96–105, 2014.
- [33] Y. Shi, "an optimization algorithm based on brainstorming process," *Int. J. Swarm Intel. Re.* pp. 35-62, 2011.

- [34] Bai, H., Zhao, B, "A Survey on Application of Swarm Intelligence Computation to Electric Power System.," in *6th World Congress on Intelligent Control and Automation*, 2006.
- [35] Saoussen Aouay; Salma Jamoussi; Yassine Ben Ayed, "Particle Swarm Optimization based method for Bayesian Network Structure," in *5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*, UAE, 2013.
- [36] R. Chibante, *Simulated Annealing Theory with Applications*, Croatia: Sciyo., 2010..
- [37] Chun-Feng Wang, and Kui Liu, " Learning Bayesian Network Classifier Based on Artificial Fish Swarm Algorithm," *IAENG International Journal of Computer Science*, Nov. 2015..
- [38] D.Y. Liu, F. Wang, Y.N. Lu, W.X. Xue and S.X. Wang., "Research on learning Bayesian network structure based on genetic algorithms," *Comput. Sci. Res. Dev.*, pp. 916-922, 2001.
- [39] Pearl, Judea, "Bayesian networks: A model of self-activated memory for evidential reasoning," in *7th Conference of the Cognitive Science Society*, California., 1985.
- [40] G. F. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," *Artificial intelligence*, p. 393–405, 1990.
- [41] Jian Cheng and Marek J. Druzdzel, "AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks.," *Journal of Artificial Intelligence Research*, pp. 155-188, 2000..
- [42] Buhlmann, P., Kalisch, M., and Maathuis, M. H, "Variable Selection in High-Dimensional Linear Models: Partially Faithful Distributions and the PC-Simple Algorithm," *Biometrika*, p. 261–278., 2010.
- [43] P. A. Leicester, *The development of object oriented Bayesian networks to evaluate the social, economic and environmental impacts of solar PV*, Loughborough University: Loughborough University, 2015.
- [44] A. Kolmogorov, "Foundations of the Theory of Probability," *Springer, Berlin*, 1950.
- [45] K. Koch, *Introduction to Bayesian statistics*, Berlin: Springer, 2007.
- [46] A. Yasin, *Incremental Bayesian network structure learning from data streams*. Machine Learning, England: Université de Nantes., 2013.
- [47] Daphne Koller and Nir Friedman, "Probabilistic graphical models: principles and techniques," MIT press, 2009.
- [48] Pearl, Judea, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference," *Morgan Kaufmann*, p. 39, 1988.
- [49] Thomas S. Verma and Judea Pearl, "Equivalence and synthesis of causal models," in *Sixth Conference on Uncertainty in Artificial Intelligence*, San Francisco, 1991.
- [50] D. M. Chickering, "Learning equivalence classes of Bayesian-network structures.," *Journal of Machine Learning Research*, p. 445–498, 2002.
- [51] S. Andersson, D. Madigan, and M. Perlman., "A Characterization of Markov equivalence classes for acyclic digraphs," *Annals of Statistics*, p. 505–541, 1997.
- [52] J. Ding, "Probabilistic inferences in Bayesian networks," *arXiv preprint*, p. 41,

2010.

- [53] Haipeng Guo and William Hsu., "A survey of algorithms for real-time Bayesian network inference," in *AAAI/KDD/UAI02 Joint Workshop on Real-Time Decision Support and Diagnosis Systems*, Canada,, 2002.
- [54] Steffen L. Lauritzen and David J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *Royal Statistical Society*, p. 157–224, 1981.
- [55] Nevin Zhang and David Poole, "A simple approach to Bayesian network computations," in *Tenth Canadian Conference on Artificial Intelligence*, Canada, 1994.
- [56] Frank Van Harmelen, Vladimir Lifschitz, and Bruce Porter, *Handbook of knowledge representation*, Elsevier, 2008.
- [57] M. Henrion, "Propagating uncertainty in Bayesian networks by probabilistic logic sampling," *Uncertainty in Artificial Intelligence*, pp. 317-324, 1988.
- [58] Robert M. Fung and Kuo-Chu Chang, "Weighing and Integrating Evidence for Stochastic Simulation in Bayesian Networks," in *Fifth Annual Conference on Uncertainty in Artificial Intelligence, UAI '89*, Amsterdam,, 1990.
- [59] Ross D. Shachter and Mark Alan Peot, "Simulation Approaches to General Probabilistic Inference on Belief Networks," in *Fifth Annual Conference on Uncertainty in Artificial Intelligence, UAI '89*, Amsterdam,, 1990.
- [60] Marco Scutari and Jean-Baptiste Denis, *Bayesian Networks with Examples in R*, Taylor & Francis Group, 2015.
- [61] D. Heckerman, D. Geiger, and D. M. Chickering., "Learning Bayesian networks: the combination of knowledge and statistical data," *Machine Learning*, pp. 197-243, 1995.
- [62] Geiger, D. and Heckerman, D., "Learning Gaussian Networks," Microsoft Research, Redmond, Washington., 1994.
- [63] Cooper, G.F., and C. Yoo, "Causal Discovery from a Mixture of Experimental and Observational Data," in *Fifteenth Conference, Uncertainty in Artificial Intelligence*, California,, 1999.
- [64] Tong, S., and D. Koller, "Active Learning for Structure in Bayesian Networks," in *eventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, Washington,, 2001.
- [65] Pe'er, D., A. Regev, G. Elidan and N. Friedman, "Inferring Subnetworks from Perturbed Expression Profiles," in *Ninth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, Copenhagen, Denmark, 2001..
- [66] G. Cooper, "A Bayesian Method for Causal Modeling and Discovery Under Selection," in *Sixteenth Conference, Uncertainty in Artificial Intelligence*, California,, 2000.
- [67] R. Robinson, "Counting Unlabeled Acyclic Digraphs," *Springer- Verlag*, p. 622, 1977.
- [68] Gillispie, S.B., and M.D. Pearlman, "Enumerating Markov Equivalence Classes of Acyclic Digraph Models," in *Seventeenth Conference, Uncertainty in Artificial Intelligence*, California, 2001.
- [69] Verma, T. S. and Pearl, J., "Equivalence and Synthesis of causal Models," *Uncertainty in Artificial Intelligence*, p. 255–268, 1991.

- [70] Spirtes, P., Glymour, C., and Scheines, R., "Causation, Prediction, and Search," *MIT Press*, 2000.
- [71] Tsamardinos I., Aliferis C. F., and Statnikov A., "Algorithms for Large Scale Markov Blanket Discovery," in *16th International Florida Artificial Intelligence Research Society Conference*, Florida , 2003.
- [72] Yaramakala, S. and Margaritis, D., "Speculative Markov Blanket Discovery for Optimal Feature Selection," in *5th IEEE International Conference on Data Mining*, 2005.
- [73] Kalisch, M. and Buhlmann, P., "Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm," *Journal of Machine Learning Research*, p. 13–636, 2007.
- [74] Kalisch, M. and Buhlmann, P., "Robustification of the PC-Algorithm for Directed Acyclic Graphs," *Journal of Computational and Graphical Statistics*, p. 773–789, 2008.
- [75] Aliferis, C. F., Statnikov, A., Tsamardinos, I., Mani, S., and Xenofon, X. D., "Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part I: Algorithms and Empirical Evaluation," *Journal of Machine Learning Research*, p. 171–234, 2010.
- [76] Hausser, J. and Strimmer, K., "Entropy Inference and the James-Stein Estimator, with Application to Nonlinear Gene Association Networks," *Journal of Machine Learning Research*, p. 1469–1484, 2009.
- [77] Scutari, M. and Brogini, A., "Bayesian Network Structure Learning with Permutation Tests," *Communications in Statistics – Theory and Methods*, p. 3233–3243, 2012.
- [78] B. RR, "Bayesian belief networks: from construction to inference," Utrecht University, Netherlands., 1995.
- [79] Larrañaga P, Sierra B, Gallego MJ, Michelena MJ, Picaza JM, "Learning Bayesian networks by genetic algorithms: a case study in the prediction of survival in malignant skin melanoma," in *6th conference on artificial intelligence in medicine in Europe (AIME '97)*, 1997.
- [80] Russell SJ, Norvig P, *Artificial intelligence: a modern approach*, Englewood Cliffs: Prentice Hall, 2009.
- [81] J. Suzuki, "A construction of Bayesian networks from databases based on the MDL principle," in *Ninth Conference on Uncertainty in Artificial Intelligence*, 1993.
- [82] W. Lam and F. Bacchus., "Learning Bayesian belief networks. An approach based on the MDL principle," *Computational Intelligence*, p. 269–293, 1994.
- [83] R. R. Bouckaert., "Belief networks construction using the minimum description length principle," *Computer Science*, 1993.
- [84] N. Friedman and M. Goldszmidt, "Learning Bayesian networks with local structure," in *Twelfth Conference on Uncertainty in Artificial Intelligence*, 1996.
- [85] E. Herskovits and G. F. Cooper. , "An entropy-driven system for the construction of probabilistic expert systems from databases," in *Sixth Conference on Uncertainty in Artificial Intelligence*, 1990.
- [86] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, p. 462–467,

1968..

- [87] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," in *Machine Learning*, 9, 1992.
- [88] W. Buntine, "Theory refinement of Bayesian networks," in *Seventh Conference on Uncertainty in Artificial Intelligence*, 1991.
- [89] Menzel, R., & Giurfa, M., "Cognitive architecture of a mini-brain: The honeybee.," in *Trends in Cognitive Sciences*, 2001.
- [90] Heckerman, D., Geiger, D., and Chickering, D., "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, vol. 20, pp. 197-243, 1995.
- [91] D. M. Chickering, "A transformational characterization of equivalent Bayesian network structures," in *Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995.
- [92] S. Acid and L. M. de Campos, "Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs," *Journal of Artificial Intelligence Research*, p. 445–490, 2003.
- [93] J. Tian, "A branch-and-bound algorithm for MDL learning Bayesian networks," in *Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000.
- [94] T. Kocka and R. Castelo, "Improved learning of Bayesian networks," in *Seventeenth Conference on Uncertainty in Artificial Intelligence*, 2001.
- [95] P. Larrañaga, M. Poza, Y. Yurramendi, R. Murga, and C. Kuijpers., "Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 912–926., 1996.
- [96] M. L. Wong, W. Lam, and K. S. Leung, "Using evolutionary computation and minimum description length principle for data mining of probabilistic knowledge," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999.
- [97] L. M. de Campos and J. M. Puerta, "Stochastic local and distributed search algorithms for learning belief networks.," in *III International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Model*, 2001.
- [98] R. Blanco, I. Inza, and P. Larrañaga, "Learning Bayesian networks in the space of structures by estimation of distribution algorithms.," *International Journal of Intelligent Systems.*, p. 205–220, 2003.
- [99] L. M. de Campos, J. A. Gámez, and J. M. Puerta, "Learning Bayesian networks by ant colony optimization: Searching in two different spaces," *Mathware and Soft Computing*, p. 251–268, 2002.
- [100] N. Friedman and D. Koller, "Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks," *Machine Learning*, p. 95–126, 2003.
- [101] P. Spirtes and C. Meek, "Learning Bayesian networks with discrete variables from data," in *First International Conference on Knowledge Discovery and Data Mining.*, 1995.
- [102] D. Dash and M. Druzdzel, "hybrid anytime algorithm for the construction of causal models from sparse data," in *Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999.

- [103] D. Madigan, S. A. Andersson, M. D. Perlman, and C. T. Volinsky, "Bayesian model averaging and model selection for Markov equivalence classes of acyclic digraphs.," *Communications in Statistics – Theory and Methods*, p. 2493–2520, 1996.
- [104] S. Acid, L. M. de Campos, and J. G. Castellano., "Learning Bayesian network classifiers: searching in a space of partially directed acyclic graphs," *Machine Learning*, pp. 213-235, 2005.
- [105] S. T.D., "The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies," *Harvard University Press*, 1995.
- [106] V. F. K, *Bees: Their Vision, Chemical Senses and Language*, N.Y., Ithaca,: Cornell University Press, 1976.
- [107] Bonabeau, E., Dorigo, M., & Theraulaz, G., "From natural to artificial swarm intelligence.," *Oxford University Press*, 1999.
- [108] S. Rodrigues de Morais and A. Aussem, "A novel scalable and data efficient feature subset selection algorithm.," in *European conference on Machine Learning and Knowledge Discovery in Databases*, Berlin., 2008.
- [109] I. Tsamardinos, L. E. Brown, and C. F. Constantin, F. Aliferis., "The max-min hill-climbing Bayesian network structure learning algorithm.," *Machine learning* , pp. 31–78., 2006.
- [110] Friedman N, Pe'er D, Nachman I, "Learning Bayesian network structure from massive datasets: the "Sparse Candidate algorithm.," in *uncertainty in artificial intelligence (UAI)*,, 1999.
- [111] H. Nguyen, *Reseaux bayesiens et apprentissage ensembliste pour letude differentielle de reseaux de regulation genetique*, France: Universite Nantes, 2010.
- [112] N. A. OBUCHOWSKI, "Receiver operating characteristic curves and their use in radiology," *Radiology*,, p. 3–8, 2003.
- [113] T. FAWCETT, "ROC Graphs: Notes and Practical Considerations for Researchers.," HP Labs Tech Report HPL, 2004.
- [114] PROVOST, F. and FAWCETT, T., "Robust Classification for Imprecise Environments.," *Machine Learning Journal*,, p. 203–231, 2001.
- [115] PROVOST, F., FAWCETT, T., , "Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions," in *Third International Conference on Knowledge Discovery and Data Mining*, 1997.
- [116] PROVOST, F., FAWCETT, T., KOHAVI, R., "The Case Against Accuracy Estimation for Comparing Induction Algorithms," in *Fifteenth International Conference on Machine Learning*, 1998.
- [117] Devashish Sharma, U.B. Yadav, Pulak Sharma, "THE CONCEPT OF SENSITIVITY AND SPECIFICITY IN RELATION TO TWO TYPES OF ERRORS AND ITS APPLICATION IN MEDICAL RESEARCH," *Journal of Reliability and Statistical Studies*, pp. 53-58, 2009.
- [118] C. van Rijsbergen, *Information Retrieval*, Butterworth, 1979.
- [119] M. Wahde, "Biologically Inspired Optimization Methods an Introduction," *WIT Press*, 2008..
- [120] Anand Jayant Kulkarni, Ganesh Krishnasamy, and Ajith Abraham, "Cohort Intelligence: A Socio-inspired Optimization Method," *Springer*, 2017.

- [121] Bonabeau, Dorigo M., Theraulaz G, "Swarm Intelligence: From Natural to Artificial Systems," 1999.
- [122] S. Camazine, "Self-organization in biological systems.," *Princeton University Press*, 2003.
- [123] Vittori, K., Talbot, G., Gautrais, J., Fourcassie, V., Araujo, A. F., & Theraulaz, G., "Path efficiency of ant foraging trails in an artificial network.," *Journal of Theoretical Biology*, p. 507–515, 2006.
- [124] Theraulaz, G., Gautrais, J., Camazine, S., & Deneubourg, J. L., "The formation of spatial patterns in social insects: from simple behaviours to complex structures.," *Philosophical Transactions of the Royal Society of London*, pp. 1263-1282., 2003.
- [125] Beshers, S. N., & Fewell, J. H., "Models of division of labor in social insects.," *Annual Review of Entomology*, p. 413–440, 2001.
- [126] Barnes, L. E., Fields, M. A., & Valavanis, K. P., "Swarm formation control utilizing elliptical surfaces and limiting functions.," in *IEEE Transactions on Systems, Man, and Cybernetics*, 2009.
- [127] Thorup, K., Alerstam, T., Hake, M., & Kjellen, N., "Bird orientation: Compensation for wind drift in migrating raptors is age dependent," in *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 2003.
- [128] Venayagamoorthy, G.K., Harley, R.G, "Swarm Intelligence for Transmission System Control," in *IEEE Power Engineering Society General Meeting*, 2007.
- [129] Kennedy, J., Eberhart, R.C., *Swarm Intelligence*, San Francisco: Morgan Kaufmann Publisher, 2001.
- [130] Bonabeau E., Dorigo M. and Theraulaz, G. “, *Swarm Intelligence*, Oxford University Press, 1999.
- [131] B. Doneley, "PIGEON MEDICINE AND SURGERY," in *The North American Veterinary Conference*, American, 2006,.
- [132] Haibin Duan and Peixin Qiao, "Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning," in *International Journal of Intelligent Computing and Cybernetics*, 2014.
- [133] Guilford, T., Roberts, S. and Biro, D, "Positional entropy during pigeon homing II: navigational interpretation of Bayesian latent state models," *Journal of Theoretical Biology*, pp. 25-38, 2004.
- [134] Mora, C., Davison, C., Wild, J. and Walker, M., "Magnetoreception and its trigeminal mediation in the homing pigeon," *Nature*,, pp. 508-511, 2004.
- [135] A. Whiten, "Operant study of sun altitude and pigeon navigation," *Nature*,, pp. 405-406., 1972.
- [136] Pham, D.T, Karaboga, D, "Intelligent optimisation techniques: genetic algorithms, tabu search, simulated annealing and neural networks," *Springer*,, 2000.
- [137] Fouskakis, D. & Draper, D., "Stochastic optimization: a review," *International Statistical Review*, pp. 315-349., 2002.
- [138] N. Collins, "Simulated annealing- an annotated bibliography," *American J.of Mathematic and Management Science*,, pp. 209-307, 1988.
- [139] ELLIS HOROWITZ, SARTAJ SAHNI, *FUNDAMENTALS OF COMPUTER ALGORITHMS*, United States of America: Computer Science Press, Inc.,,

- 1978.
- [140] Mauricio G.C. Resende • Celso C. Ribeiro, "Optimization by GRASP Greedy Randomized Adaptive Search Procedures," *Springer Science+Business Media*, 2016.
- [141] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, Third Edition, Massachusetts London, England: Massachusetts Institute of Technology, 2009.
- [142] Nakrani, S. and Tovey C, "On honey bees and dynamic allocation in an internet server colony," *Adaptive Behavior*, pp. 223-240, 2004.
- [143] Teodorović Dušan, Orco Mauro Dell, "Bee Colony Optimization-A Cooperative Learning Approach to Complex Transportation Problems," *Advanced OR and AI Methods in Transportation*, pp. 51-60, 2005.
- [144] Chong Chin Soon, Low Malcolm Yoke Hean, Sivakumar Appa Iyer, Gay Kheng Leng, "A Bee Colony Optimization Algorithm to Job Shop Scheduling," in *Winter Simulation Conference*, N.J., USA. WSC., 2006..
- [145] J., Beni G. and Wang, "Swarm intelligence in cellular robotics systems," *NATO Advanced Workshop on Robots and Biological System*,, 1989..
- [146] V. F. K., "Die Tänze der Bienen," *Österr Zool*, pp. 1-48, 1946.
- [147] G. J.L., "Landmark Learning by Honey Bees," *Animal Behaviour*, vol. 35, pp. 26-34, 1987.
- [148] Camazine S. and Sneyd. J, "A Model of Collective Nectar Source by Honey Bees: Self- organization Through Simple Rules," *Journal of Theoretical Biology*, pp. 547-571, 1991.
- [149] Lučić Panta, Teodorović Dušan, Modeling Transportation Problems Using Concepts of Swarm Intelligence and Soft Computing, USA,: Virginia Polytechnic Institute and State University, 2002.
- [150] W. B.G., "Aristotle and the Dance of Bees," *The Classical Review*, 1958.
- [151] Landgraf Tim and Rojas Raúl, "Tracking honey bee dances from sparse optical flow fields," Freie Universität Berlin, Berlin, 2007.
- [152] L. M., "Schwarmbienen auf Wohnungssuche," *Z. vergl. Physiol*, pp. 263-324, 1955.
- [153] Lučić Panta, Teodorović Dušan, "Computing with Bees" Attacking Complex Transportation Engineering Problems," *International Journal on Artificial Intelligence tools*, pp. 375-394, 2003..
- [154] Pham D.T., Ghanbarzadeh A., Koç E., Otri S., Rahim S, Zaidi M., "The Bees Algorithm – A Novel Tool for Complex Optimisation Problems," in *International Conference on Intelligent Production Machines and Systems (IPROMS 2006)*, Cardiff, 2006.
- [155] H.CHANDRAMOULI, "Elephant Swarm Optimization In Wireless Sensor Network To Enhance Network Lifetime," *Shri Jagdishprasad Jhabarmal Tibrewala university*.
- [156] B. C, "Not So Dumbo - Elephant Intelligence," 2003. [Online].
- [157] Granli P, Poole J, "Why and How Elephants Communicate - Elephant Voices," 2006. [Online]. Available: http://www.elephantvoices.org/index.php?topic=about_sevp..
- [158] M. P, "Elephants Use Mental Maps to Track Family Members," 2007. [Online].

Available: <http://www.newscientist.com/article/dn12998.htm..>

- [159] Amir Hossein Gandomi, Xin-She Yang, Siamak Talatahari Marand, and Amir Hossein Alavi, "Metaheuristic applications in structures and infrastructures," *USA: Elsevier*, 2013.
- [160] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis, "A grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.
- [161] Gai-Ge Wang, Suash Deb, and Leandro dos S. Coelho, "Elephant Herding Optimization," in *3rd International Symposium on Computational and Business Intelligence* , Bali, Indonesia, 2015.
- [162] S. MANDAL, "Elephant swarm water search algorithm for global optimization," *Indian Academy of Sciences*, 2018.
- [163] Xin J, Chen G and Hai Y., "A particle swarm optimizer with multistage linearly-decreasing inertia weight," 2009.
- [164] S. Ammar, Probabilistic graphical models for density estimation in high dimensional spaces: application of the Perturb & Combine principle with tree mixtures, England: Université de Nantes, 2010.