Comparative Analysis of Search and Score Metaheuristics for Bayesian Network Structure Learning using Node Juxtaposition Distributions

Yanghui Wu[1], John McCall[1], and David Corne[2]

[1] IDEAS Research Institute, Robert Gordon University, Aberdeen, UK
{y.wu3,j.mccall}@rgu.ac.uk
[2] School of Mathematics and Computer Science, Heriot-Watt University, Edinburgh,UK
dwcorne@macs.hw.ac.uk

Abstract. Learning Bayesian networks from data is an NP-hard problem with important practical applications. Metaheuristic search on the space of node orderings combined with deterministic construction and scoring of a network is a well-established approach. The comparative performance of different search and score algorithms is highly problem-dependent and so it is of interest to analyze, for benchmark problems with known structures, the relationship between problem features and algorithm performance. In this paper, we investigate four combinations of search (Genetic Algorithms or Ant Colony Optimization) with scoring (K2 or Chain). We relate node juxtaposition distributions over a number of runs to the known problem structure, the algorithm performance and the detailed algorithmic processes. We observe that, for different reasons, ACO and Chain both focus the search on a narrower range of orderings. This works well when the underlying structure is compatible but poorly otherwise. We conclude by suggesting future directions for research.

Keywords: Ant Colony Optimization, Genetic Algorithm, Bayesian Network Structure Learning, Node Ordering, Chain Model, K2

1 Introduction

Bayesian networks (BNs) are probabilistic graphical models which are used to represent knowledge about uncertain domain. The network consists of a directed acyclic graph (DAG) whose nodes represent random variables, and whose edges represent the direct dependencies between these variables, and a joint probability distribution (JPD) over the random variables. The JPD factorises according to the DAG structure. In many domains, the BN structure and parameters must be learned from data. Learning BN structure is a NP hard problem. It is known that the number of possible structures grows super-exponentially with the number of nodes [1], and so evaluating all possible structures is infeasible in most practical domains, where the number of variables is typically large. The process of finding cheaper approaches for learning the structure of BNs from large datasets is now a very research active area.

A well-established approach to learning BN structure uses metaheuristic search on the space of node orderings combined with deterministic construction and scoring of a network. The comparative performance of different search and score algorithms is highly problem-dependent and so it is of interest to analyze, for benchmark problems with known structures, the relationship between problem features and algorithm performance. In this paper, we investigate combinations of two metaheuristic search techniques, Genetic Algorithms (GA) and Ant Colony Optimisation (ACO) with two scoring approaches, K2 and Chain. All are previously published algorithms for which empirical trade-offs between computational expense and structural accuracy with a high degree of problem dependency have been observed [2–4].

In this paper, we attempt to understand this problem dependency. We explore the distributions of nodes juxtapositions in the best solutions found over a number of runs and relate this to the known problem structure, the algorithm performance and the detailed algorithmic processes.

The remainder of this paper is organized as follows: in section 2, we briefly describe search and score approaches for BN structuring learning. In section 3 we describe experiments used to generated node juxtaposition distributions. Results are discussed in section 4, and conclusions presented in section 5.

2 Background

2.1 Bayesian Network Structure Learning using Search and Score

Search and score approaches attempt to search for the BN structure which best fits the data according to a scoring function. A range of well-known search techniques have been applied in search and score, including Hill Climbing [5], Genetic Algorithms [6], Simulated Annealing [7], Particle Swarm Optimization (PSO) [8], and Ant Colony Optimization (ACO) [9, 10]. The most common scoring functions used in these algorithms include the K2-CH metric [11], BDeu [12], BIC [13], and Minimum Description Length (MDL) [14].

2.2 K2 Algorithm and K2-based Search and Score

K2 is a well-known greedy algorithm that constructs and evaluates a BN from a database of cases [11]. K2 assumes that an ordering on the variables is available and that, a priori, all structures are equally likely. Moreover, it assumes a maximum number of parents a node can have. It starts by assuming that all nodes in the DAG are without parents (i.e. no edges). At each step, edges are added where doing so increases the joint probability of the resulting structure. K2 stops when no further edges can be added.K2 can thus be used as a scoring approach by applying it to an ordering selected by a metaheuristic search. Several K2-based search and score algorithms have been proposed. Here, we briefly introduce two relevant to this paper: K2GA and K2ACO. K2GA [2] uses a GA to search the space of node orderings. The fitness of each ordering is evaluated by running the K2 search algorithm on each ordering evaluated and returning the score of the network structure found. Standard ordering based operators are applied. Similarly, in K2ACO [4], node orderings are generated by a colony of ants. Evaluations from K2 are then used for pheromone update.

2.3 Chain Based Search and Score

Chain scoring for BN structure learning is first proposed in [3]. It is based on the hypothesis that an initial search phase of evaluating fixed chain structures imposed on orderings provides a sufficiently good scoring function to locate high scoring regions of the space of node orderings. A second phase then follows where K2 is applied directly to the best orderings found. Given a node orderingX_1,X_2,...,X_n, we define the chain structure by adding edges between successive nodes. Thus X_i is the sole parent of X_{i+1}. E_i is the edge from X_i to X_{i+1} Figure 1.

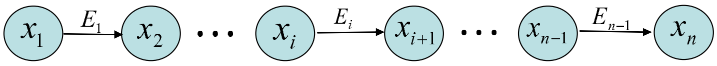

Fig. 1. Chain structure on an ordering

In our previous work, GA (ChainGA) [3] and ACO (ChainACO) [4] are developed as Chain-based search heuristics. At each evaluation step, a chain structure of the given ordering is constructed and evaluated using the K2-CH [11] score metric evaluations. At the end of evaluation, the ordering corresponding to the best fitness score is then produced for K2 algorithm to construct the BN structure. This is a relatively cheap evaluation in terms of the number of K2-CH factor evaluations needed. Our previous results have shown that the Chain structure model can get a significant reduction in computational cost for large data sets. The pseudocode for ChainACO and ChainGA is given in Tables 1 and 2 respectively.

3 Experiments

The aim of our experiments is to investigate the behaviour of GA and ACO metaheuristics searching the space of node orderings using the Chain and K2 evaluation methods. We try to explore the relationship between arcs derived from node juxtapositions in the best orderings found (Figure 1) and arcs in the original structure. We therefore make runs of each metaheuristic with each evaluation method, which we denote ChainACO, ChainGA, K2ACO and K2GA. Note that in these experiments, we only require to run the first phase of ChainACO and ChainGA algorithms to obtain the best node orderings, as the search of ordering

Table 1. Pseudocode of ChainACO Algorithm

```
Initialize pheromone
Initialize heuristic information
Loop
   Each ant is positioned on a starting node
   Loop
     Each ant applies a state transition rule to incrementally
     build a solution and a local pheromone updating rule
   Until all ants have built a complete solution
   A global pheromone updating rule is applied
Until termination criterion is met
Implement K2 Algorithm on best solution to learn the best
structure.
```

Table 2. Pseudocode of ChainGA Algorithm

```
Initialize population
Repeat
   Select best-ranking individuals to reproduce
   Apply crossover operator
   Apply mutation operator
Until termination criterion is met
Implement K2 Algorithm on best solution to learn the best
structure.
```

space ends at this point - the second phase is deterministic. However we are not comparing the computational efficiency or the scores of final networks produced as that has been covered in our earlier work [3, 4].

Four well known benchmark problems have been selected in our research: Asia, Car, Insurance and Alarm. The Asia network is a simple network with 8 binary nodes and 8 edges. It is a diagnostic demonstrative Bayesian network [15]. The Car Diagnostic Network consists of 18 nodes and 17 edges. It can be applied to diagnose malfunctioning of self-propelling vehicles [3] . The Insurance network contains 27 nodes and 52 arcs, is a network for evaluating car insurance risks [16] . The Alarm network is a medical diagnostic system for intensive care patient monitoring consisting of 37 nodes and 46 edges [17]. All the data cases are sampled using the Netica tool [18]. In this paper, the dataset sizes for Asia, Car, Insurance and Alarm are 5000, 10000, 5000 and 3000 cases respectively.

In all cases, the scoring metric used to evaluate the node ordering is the K2-CH metric. For ChainACO and ChainGA we carry out 200 experimental runs each. For K2ACO and K2GA we carry out only 50 runs each due to time complexity. The parameters used for ACO and GA based algorithms in this paper are the same as those used in [3, 4].

4 Results and Discussion

Table 3 presents the distribution of node juxtapositions recorded from the best ordering found in each of 200 experimental runs of ChainACO on the Asia network. The row index indicates the first node in a juxtaposition, the column index indicates the second. For example, the Table shows that in 65 of the runs, the node juxtaposition 1-2 appeared 65 times, and the node juxtaposition 2-1 appeared 135 times. This means that, in all runs of ChainACO on Asia, nodes 1 and 2 were juxtaposed in the best ordering found, with a 135:65 preference for node 2 preceding node 1. In all of these cases, the ordering will have been evaluated using a chain structure inserting a directed edge between these nodes. It is not of course necessary that any particular juxtaposition will appear in all experimental runs. Each ordering found will contain $n - 1$ juxtapositions where n is the number of nodes. The sum of entries in the ordering distributions table will therefore in general be $r \cdot (n-1)$, where r is the number of runs. In this case, the entries sum to $1400 = 200 \times 7$.

Table 3. Node Juxtaposition Distribution for 200 runs of ChainACO on Asia

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0	65	0	135	0	0	0	0
2	135	0	19	21	9	7	0	0
3	10	76	0	3	12	4	0	4
4	44	0	4	0	0	145	0	0
5	0	9	65	0	0	0	2	31
6	0	3	0	37	0	0	158	2
7	0	0	0	0	0	42	0	158
8	0	2	3	0	155	0	40	0

It is noticeable from Table 3 that the distribution of node juxtapositions is concentrated on a relatively small subset of possible node juxtapositions. This indicates that ChainACO is highly consistent in the node orderings it produces and suggests a strong convergence property of the search. In Figure 2, we present a visual representation of the node distributions produced by all four algorithms on Asia and Car. Here, the instance counts have been replaced by a normalized grayscale representation running from white (juxtaposition occurs on 0% of runs) through to black (juxtaposition occurs on 100% of runs). It is easy to observe from Figure 2a that there is a marked difference in distribution between ChainACO (top-left) and K2GA (bottom right). ChainACO produces a high contrast image consisting of mostly very dark or very light pixels whereas the K2GA image is much more diffuse. It is hard to visually detect much of a difference in contrast between K2ACO and ChainGA other than that they lie somewhere in between the other two. Moreover the dark areas for ChainACO do not particularly coincide with those for K2GA. However results in [4] show that each algorithm reliably reproduces the Asia network.

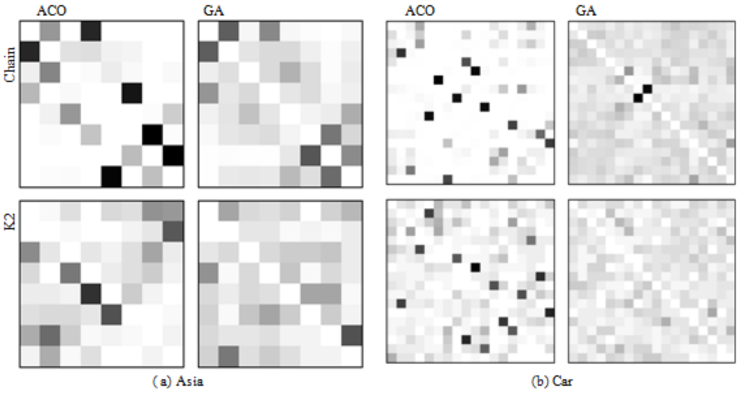

Fig. 2. Grayscale Grids of the Edges Occurrences in Asia and Car Networks within the Four Algorithms.

In Figure 2b, there are more possible node juxtapositions and the visual contrast is more marked. In order from highest to lowest visual contrast, the images are ordered ChainACO, K2ACO, ChainGA and K2GA. This ordering is consistent with a hypothesis that both the Chain scoring approach and the ACO metaheuristic result in more concentrated distributions than the K2 scoring approach and the GA metaheuristic respectively. Finally, the equivalent diagrams for Insurance and Alarm are shown in Figure 3. As these networks have many more possible node juxtapositions the diagrams have a finer granularity but the same effects are observable.

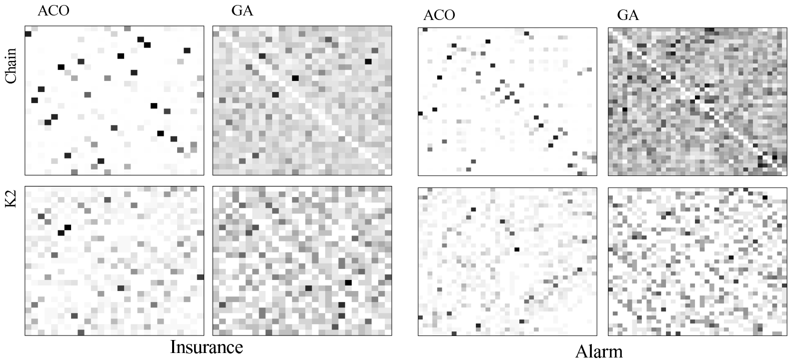

Fig. 3. Grayscale Grids of the Edges Occurrences in Insurance and Alarm Networks within the Four Algorithms.

We present node juxtaposition frequencies for all four algorithms for the Insurance and Alarm problems as Box plots in Figure 4. These essentially show the same information. Here the effect of the Chain approach manifests as a low median frequency for most possible juxtapositions with a small number of high frequency outliers. This is particularly noticeable for ChainACO in both Figures. Conversely, for K2 approaches, and K2GA in particular, there is a higher median node distribution frequency and a large distribution of frequencies in the interquartile range, corresponding to the more diffuse visual pattern observed earlier.

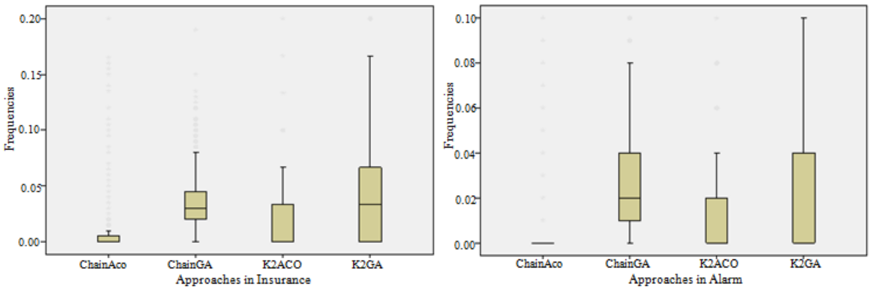

Fig. 4. Comparison of Frequencies of Each Edges Found in Insurance and Alarm Networks within four Algorithms.

Overall, there appears to be a small reduction in variability of the final ordering produced deriving from the use of ACO rather than GA, but the dominant difference in behaviour derives from the choice of scoring approach. We observe that the Chain scoring approach concentrates the search on a smaller set of node juxtapositions, and hence node orderings than the K2 scoring approach does. This is because, for any particular ordering, Chain only inserts edges between juxtaposed nodes whereas K2 may insert an edge between any two nodes. Thus it is possible to discover valuable interactions from a wider range of orderings with K2 than with Chain. Conversely, it takes longer to evaluate orderings with K2 because a large number of possible edges have to be considered in turn for each ordering. Therefore, the relative merits of Chain and K2 for any particular problem lie in how amenable the dependencies in the data are to discovery using the Chain approach.

Figure 5 and Figure 6 are diagrams of the known true structures for Asia and Car respectively, annotated with the best ordering found by ChainACO. For each node juxtaposition occurring in the best ordering that corresponds with an edge in the true structure, an arrow is added in the middle of the edge in the direction of the node juxtaposition. If there is no edge in the true structure corresponding to the node juxtaposition, a dotted arrow is added to the diagram. Solid directed edges with no central arrow therefore represent edges that occur in the true structure but are not represented by node juxtapositions in the best ordering found. We also annotate each node juxtaposition with the overall percentage of runs in which it appeared in the best ordering for that run.

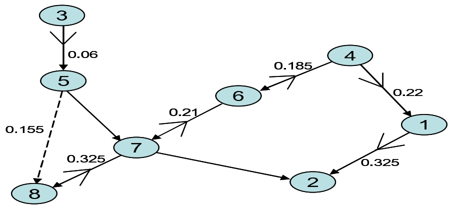

Fig. 5. Asia - annotated by best ChainACO ordering 3-5-8-7-6-4-1-2.

Figure 5 shows that all but one of the node juxtapositions in the best ordering found by ChainACO for Asia coincide with true arcs. Of these three out of six are correct and three reversed. The solution adds one node juxtaposition corresponding to a spurious arc and omits two arcs. However the nodes corresponding to the two omitted arcs are correctly ordered and so could be discovered by K2 in the second phase of the algorithm. This analysis shows that it is possible to create chains closely aligned to the structure. This explains why ChainACO and ChainGA perform well on Asia.

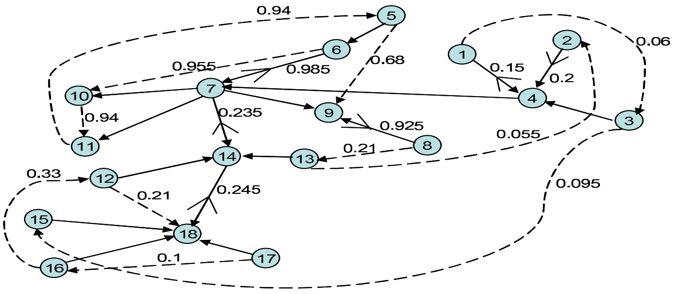

Fig. 6. Car - annotated by best ChainACO ordering 17-16-12-18-14-7-6-10-11-5-9-8-13-2-4-1-3-15.

Figure 6 shows that only six of the seventeen node juxtapositions in the best ordering found by ChainACO for Car coincide with true arcs. Of these only one out of six is correct and five are reversed. The solution adds eleven node juxtapositions corresponding to spurious arcs and omits ten arcs. It is noticeable that the true Car structure contains nodes such as 4, 7, 14 and 18 each of which is a hub for a cluster of tightly-bound nodes. The binding between these clusters is loose. Such a topology is not amenable to the construction of chains where many node juxtapositions correspond to a true arc independent parents. For example, only one node can be positioned before node 18, which immediately excludes at least three true arcs in any ordering. Therefore in this case, the Chain approach finds spurious arcs that can coexist as node juxtapositions in a single ordering and give a better score than orderings that include correct node juxtapositions.

This inherent difficulty in aligning chains of node juxtapositions with the true structure explains why ChainACO and ChainGA perform poorly on Car.

Table 4 contains for each problem and for the best node ordering found over all runs by ChainACO, the numbers: C, of node juxtapositions corresponding to correct arcs; R to reversed arcs; A to additional, spurious, arcs; and O is the number of arcs in the true structure to which no node juxtapositions correspond; T represents the total arcs in true structure. The table shows that C and R dominate for those problems (Asia and Alarm) where ChainACO performs well. On the other hand, A and O dominate for those problems (Car and Insurance) where ChainACO performs poorly.

Table 4. Alignment of best ChainACO ordering with true structures

	C	**R**	**A**	**O**	**T**
Asia	3	3	1	2	8
Car	1	5	11	11	17
Insurance	10	8	8	34	52
Alarm	14	6	15	26	46

5 Conclusions

In this paper we have conducted experiments to investigate the behaviour of GA and ACO metaheuristics searching the space of node orderings using the Chain and K2 evaluation methods. We have explained problem-dependent performance trade-offs between cost and structure quality in terms of the relationship between the Chain scoring mechanism, which relies on ordering node juxtapositions and true arcs in the original structure. In all problems investigated, the Chain scoring approach focused the search on a narrower range of orderings than did the K2 scoring approach. A lesser effect was also observed in that ACO-based methods appeared to concentrate search more than GA-based methods.

The major conclusion of our analysis of node juxtaposition statistics is that in problems where the true structure of the data is amenable to alignment of node juxtapositions in a single ordering, the Chain scoring approach is able to yield high quality solutions with significantly less computational effort than the K2 scoring approach. In problems where such alignment is not possible, the Chain scoring approach is likely to be unsuccessful in producing high quality structures and so the relative benefit of reduced computational time is lost.

Finally our results suggest a possible direction for future work. A generalization of the Chain scoring approach that could detect shorter series of well-aligned node juxtapositions combined with a coarser-grain version of the K2 algorithm could potentially assemble high quality structures at reduced cost, even in situations where a single ordering would not admit a set of well-aligned arcs. ACO is a promising approach for more generalized construction approaches.

References

1. R. W. Robinson. Counting labeled acyclic digraphs. In F. Harary editor, New Directions in the Theory of Graphs, Academic Press, New York, pp. 239–273,(1973).
2. P. Larraaga, C. Kuijpers, and R. Murga. Learning Bayesian network structures by searching for the best ordering with genetic algorithms. IEEE Transactions on System, Man and Cybernetics 26, pp. 487–493, (1996).
3. R. Kabli, F. Herrmann, J. McCall. A chain-model genetic algorithm for Bayesian network structure learning, Proceedings of the 9th annual conference on Genetic and evolutionary computation. In IEEE World Congress on Computational Intelligence 2007, London, England, July 07–11.
4. Y. Wu, J. McCall, and D. Corne. Two Novel Ant Colony Optimization Approaches for Bayesian Network Structure Learning. Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2010), (2010).
5. I. Tsamardinos, L. E. Brown. Constantin F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm, Machine Learning, 65(1), pp. 31–78, 2006.
6. P. Larraaga, R. Murga, M. Poza, C. Kuijpers. Structure learning of Bayesian networks by hybrid genetic algorithms. In: Preliminary Papers of the Fifth International Workshop on Artificial Intelligence and Statistics, pp. 310–316, 1995.
7. T. Wang, J. Touchman, and G. Xue. Applying two- level simulated annealing on Bayesian structure learning to infer genetic networks. In Proceedings of the IEEE Computational Systems Bioinformatics Conference, pp. 647–648, 2004.
8. J. Cowie, L. Oteniya, R. Coles. Particle Swarm Optimization for learning Bayesian Networks. Proceedings of the World Congress on Engineering 2007 Vol. I WCE 2007, July 2–4, 2007, London, U.K.
9. L.M. de Campos and J.F. Huete. A new approach for learning Bayesian networks using independence criteria. International Journal of Approximate Reasoning. vol. 24, pp. 11–37, 2000.
10. L.M. de Campos, J.A. Gmez, and J.M. Puerta. Learning Bayesian network by ant colony optimization: Searching in two different spaces. Mathware and Soft Computing IX (2-3), pp.251–268, 2002.
11. G. F. Cooper, E. Herskovits. A Bayesian Method for the Induction of Probabilistic Network from Data, Machine Learning, 9(4), pp.309–347, 1992.
12. W. Buntine. Theory refinement on Bayesian networks. UAI-17, Morgan Kaufmann, pp.52–60, 1991.
13. G. Schwartz. Estimating the dimensions of a model. Ann. Statist, 6(2), pp. 461–464.1978.
14. D. Heckerman, D. Geiger, and D. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. In KDD Workshop, pp. 85-96, 1994.
15. S.L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society, 50(2), pp. 157–224, 1988.
16. J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. Machine Learning, 29(2), 213–244, 1997.
17. I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. Proceedings of the Second European Conference on Artificial Intelligence in Medical Care, Springer-Verlag, Berlin. pp. 247–256, 1989.
18. Netica. Netica Bayesian network software from Norsys. `Http://www.norsys.com.`