International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:7, 2007

Choosing Search Algorithms in Bayesian Optimization Algorithm

Hao Wu, and Jonathan L. Shapiro

***Abstract*—**The Bayesian Optimization Algorithm (BOA) is an algorithm based on the estimation of distributions. It uses techniques from modeling data by Bayesian networks to estimating the joint distribution of promising solutions. To obtain the structure of Bayesian network, different search algorithms can be used. The key point that BOA addresses is whether the constructed Bayesian network could generate new and useful solutions (strings), which could lead the algorithm in the right direction to solve the problem. Undoubtedly, this ability is a crucial factor of the efficiency of BOA. Varied search algorithms can be used in BOA, but their performances are different. For choosing better ones, certain suitable method to present their ability difference is needed. In this paper, a greedy search algorithm and a stochastic search algorithm are used in BOA to solve certain optimization problem. A method using Kullback-Leibler (KL) Divergence to reflect their difference is described.

***Keywords*—**Bayesian Optimization Algorithm, Greedy Search, KL Divergence, Stochastic Search.

I. Introduction

THERE has been much recent work about optimization algorithms that build probability models based on good solutions found so far and use the constructed models to guide the further search. This class of algorithms is called Estimation of distribution algorithms (EDAs). The general scheme of EDAs works as follows [8],

(1) Develop a probability distribution model by using initial population

(2) Sampling step: generate a data set by sampling from the probability model

(3) Testing step: test the data as solutions to the problem

(4) Selection step: create an improved data set by selecting the better solutions and removing the worse ones

(5) Learning step: create a new probability model from the old model and the improved data set

(6) If the termination criteria are not met, go to (2)

The Bayesian Optimization Algorithm (BOA) [2][3], is an algorithm based on the estimation of distributions. It uses Bayesian networks to build the probability model. To learn the structure of Bayesian network, various search algorithms can be used in the BOA. Generally, a greedy search algorithm is used. In this paper, the greedy search algorithm and a stochastic search algorithm are used in BOA, and the performance comparison between the two search algorithms in following sections shows the stochastic search algorithm solves certain optimization problem more efficiently according to the greedy search algorithm. To find out the reason, we tried several methods, and found out a method using Kullback-Leibler (KL) Divergence to reflect the ability difference between the two search algorithms, which is described in the experiment section.

II. Bayesian Optimization Algorithm

Bayesian Optimization Algorithm (BOA), an algorithm based on the estimation of distributions, uses Bayesian networks to model promising solutions and biases the sampling of new candidate solutions.

A. BOA Procedure

The procedure of the BOA follows:

(1) Set t=0, randomly generate an initial population P (0)

(2) Select a set of promising strings S (t) from P (t)

(3) Construct a Bayesian network B using a chosen metric and constraints

(4) Generate a set of new strings O (t) according to the joint distribution encoded by B

(5) Create a new population P (t+1) by replacing some strings from P (t) with O (t), set t=t+1

(6) If the termination criteria are not met, go to (2)

B. Bayesian Network

Bayesian networks [4] are often used for modeling multinomial data with both discrete and continuous variables. A Bayesian network encodes the relationships between the variables contained in the modeled data. It represents the structure of a problem. Bayesian networks can be used to describe the data as well as to generate new instances of the variables with similar properties as those of given data. In the network, every node corresponds to one variable. An edge between two corresponding nodes is used to represent the relationship between two variables.

Mathematically, an acyclic Bayesian network with directed edges encodes a joint probability distribution. This can be

Manuscript received July 15, 2005.

Hao Wu is a PhD student, under Jonathan L. Shapiro's supervision, of School of Computer Science, University of Manchester, United Kingdom (phone: +44 161 2756205; e-mail: wuh@cs.man.ac.uk).

Jonathan L. Shapiro is with School of Computer Science, University of Manchester, United Kingdom (e-mail: jls@cs.man.ac.uk).

written as

$$P(X)=\prod_{i=1}^{n}P(X_i \mid Pa(X_i)) \tag{1}$$

where X is a vector of variables, $Pa(X_i)$ is the set of parents of X_i in the network (the set of nodes from which there exists an edge to X_i) and $P(X_i / Pa(X_i))$ is the conditional probability of X_i conditioned on the variables $Pa(X_i)$.

For a successful application of Bayesian Optimization Algorithm (BOA), it is necessary to learn a Bayesian network structure and parameters (conditional probabilities) which reflect the dependencies and independencies that decompose the problem properly. Learning the parameters for a given structure is easy, because the value of each variable in the population of promising solutions is specified. But learning the structure is a much more difficult problem. The algorithm for structure learning generally contains two important components:

(1) A scoring metric for measuring the quality of Bayesian network structures;

(2) A search procedure for searching the space of all possible network structures to find the best one with respect to a given scoring metric.

In this paper, Bayesian Information Criterion (BIC)[7] is selected as the scoring metric.

The number of all possible structures for a Bayesian network with n nodes can be calculated by using a recursive formula [5] as follows:

$$r(n)=\sum_{i=1}^{n}(-1)^{i+1}\binom{n}{i}2^{i(n-i)}r(n-i) \tag{2}$$

This equation gives $r(3)=25$, $r(5)=29281$, $r(10)\approx 4.2\times 10^{18}$. Since Equation 2 is super-exponential, it is impractical to perform an exhaustive search to find the best structure.

III. Search Algorithms

A. Greedy Search

In BOA, normally a greedy search algorithm given a scoring metric is used as follows:

(1) Initialize the network (e.g., to an empty network)

(2) Collect all possible simple graph operations (e.g., edge addition) that can be performed on the network without violating the constraints (e.g., introduce cycle?)

(3) Pick the operation that increases the score of the network the most

(4) Perform the operation picked in the previous step

(5) If the network can no longer be improved under given constraints on its complexity or a maximal number of interactions have been reached, finish

(6) If the network still can be improved, go to (2)

There are three elementary operations that could be chosen: edge addition, edge removal and edge reversal. However, according to empirical results, using edge removal and reversal with edge addition does not significantly improve the learning comparing with just using edge addition [2][3][6]. So people generally only choose edge addition as the operation.

B. Stochastic Search

A stochastic search algorithm performs series of elementary operations (edge addition, edge removal or edge reversal) between randomly chosen pair of nodes, which improve the quality of the current network the most until no more improvement can be obtained in a specific number of continuous iterations. The basic procedure works as follows:

(1) Initialize the network

(2) Randomly choose two different nodes *Xi* and *Xj*

(3) If no edge between them,

Collect edge additions (from *Xi to Xj* or from *Xj to Xi*) that can be performed on the network without violating the constraints (e.g., introduce cycle?);

Pick the operation that increases the score of the network the most;

Perform the operation picked in the previous step

(4) If an edge exists,

Collect edge removal and edge reversal that can be performed on the network without violating the constraints;

Pick the operation that increases the score of the network the most;

Perform the operation picked in the previous step

(5) If no improvement has been obtained in a specific number of continuous iterations, finish

(6) If any improvement obtained, go to (2)

IV. Experiments

The key objective of BOA is that the constructed Bayesian network could generate new and useful solutions, which could lead the algorithm in the right direction to solve the optimization problem efficiently, which means that this ability of the search algorithm is crucial. For search algorithms selection, how to present their ability difference needs to be solved.

The experiments are divided into two parts: Part I is to make performance comparison between the greedy search algorithm and the stochastic search algorithm for one optimization problem; Part II is to introduce a KL Divergence method to reflect their ability difference.

A. Part I

Firstly, a search optimization problem is needed. Since the results that these search algorithms obtain are Bayesian networks, an optimization problem highly relative with Bayesian networks is set as follows:

Given a randomly generated Bayesian network *B* including a structure of *10* binary nodes with *20* edges and relative conditional probabilities, find out the vector X_k that could maximize the joint probability distribution $P(X)$ of *B*.

So this optimization problem is that the best one is expected to be found among 2^{10} strings.

One example structure of *B* is shown as Fig.1:

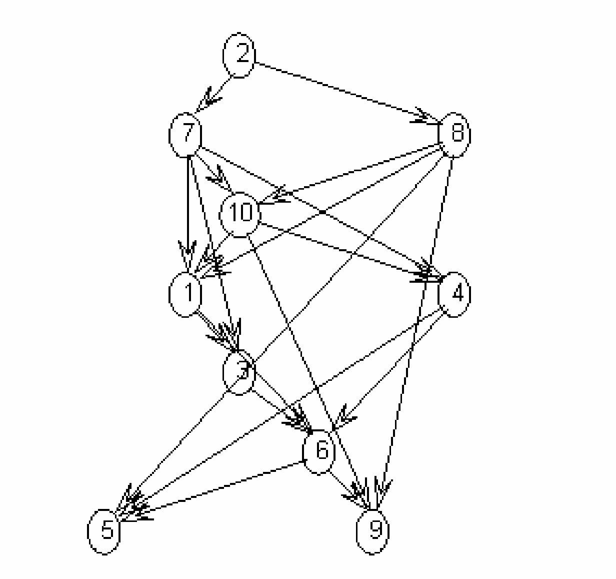

Fig. 1 An example structure of Bayesian Network

Following the procedure of BOA, the Bayesian Information Criterion (BIC)[7] is selected as the scoring metric, no different individuals existing in the population is set as the termination criterion, and no less than ten percent of new population equal to the optimum is set as the mark that the problem has been solved. We varied the data size (the number of samples) by 50, 100 and 200, and made tables to record 100 runs results by using the stochastic search algorithm and the greedy search algorithm.

(1) Data Size: 50

TABLE I
100 RUNS RESULTS OF TWO ALGORITHMS GIVEN RANDOM BAYESIAN NETWORK OF 10 NODES WITH 20 EDGES (SS=STOCHASTIC SEARCH, GS=GREEDY SEARCH, #=THE NUMBER OF)

# Bayesian Network *B*	100
# SS converged at optimum, GS didn't	30
# GS converged at optimum, SS didn't	13
# Neither converged at optimum	33
# Both did, but SS took fewer generation times	16
# Both did, but GS took fewer generation times	3
# Both took same generation times to optimum	5

As the table above shows, given 100 randomly generated Bayesian networks, the stochastic search algorithm could find out the best solution for 54 of them, while the greedy search algorithm could solve 37 of them. 16 of 19 the stochastic search algorithm needed fewer generation times than the greedy search algorithm to reach the global optimum.

(2) Data Size: 100

TABLE II
100 RUNS RESULTS OF TWO ALGORITHMS GIVEN RANDOM BAYESIAN NETWORK OF 10 NODES WITH 20 EDGES (SS=STOCHASTIC SEARCH, GS=GREEDY SEARCH, #=THE NUMBER OF)

# Bayesian Network *B*	100
# SS converged at optimum, GS didn't	34
# GS converged at optimum, SS didn't	3
# Neither converged at optimum	20
# Both did, but SS took fewer generation times	33
# Both did, but GS took fewer generation times	3
# Both took same generation times to optimum	7

As the table above shows, given 100 randomly generated Bayesian networks, the stochastic search algorithm could find out the best solution for 77 of them, while the greedy search algorithm could solve 46 of them. 33 of 36 the stochastic search algorithm needed fewer generation times than the greedy search algorithm to reach the global optimum.

(3) Data Size: 200

TABLE III
100 RUNS RESULTS OF TWO ALGORITHMS GIVEN RANDOM BAYESIAN NETWORK OF 10 NODES WITH 20 EDGES (SS=STOCHASTIC SEARCH, GS=GREEDY SEARCH, #=THE NUMBER OF)

# Bayesian Network *B*	100
# SS converged at optimum, GS didn't	21
# GS converged at optimum, SS didn't	0
# Neither converged at optimum	16
# Both did, but SS took fewer generation times	39
#Both did, but GS took fewer generation times	13
#Both took same generation times to optimum	11

As the table above shows, given 100 randomly generated Bayesian networks, the stochastic search algorithm could find out the best solution for 84 of them, while the greedy search algorithm could solve 63 of them. 39 of 52 the stochastic search algorithm needed fewer generation times than the greedy search algorithm.

B. Part II

The statistic results of Part I show that the stochastic search algorithm outperformed the greedy search algorithm for this optimization problem under given condition. It seems that the former owns stronger ability to generate new and useful strings than the latter. Thus, a suitable method is needed to present this difference between them.

Firstly, we tried to make a comparison of the entropy of the Bayesian networks obtained by these two search algorithms. The entropy *H* [1] is calculated using following equation:

$$H = -\sum_i p_i \log p_i \tag{3}$$

where p_i is the probability of i. And when p_i equals to 0, *H* equals to 0.

However, these two search algorithms got Bayesian networks with similar entropy. The results given data size 100 is could be shown in Fig. 2. (* for Stochastic Search, o for Greedy Search)

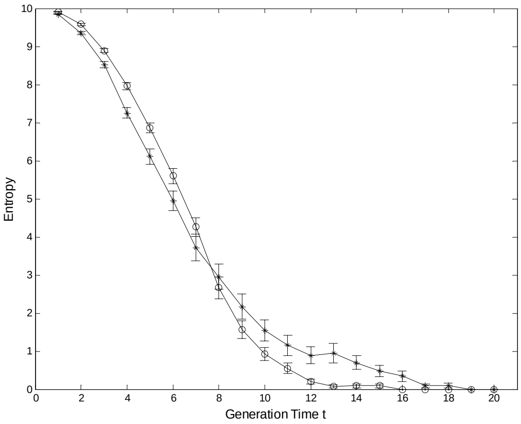

Fig. 2 Entropy of 100 runs

Then we tried another method using Kullback-Leibler (KL) Divergence [1], which is a natural distance function from one probability distribution to the other one. It could measure how much different information contained between them.

The method we introduced is to calculate and record all KL Divergences between the Bayesian Network *B(t)* learned at time *t* and the one *B(t+1)* learned at time *t*+1, then calculate average and use error-bar to show chosen results together in a same figure. The KL Divergence is calculated using following equation:

$$KL-D(B(t),B(t+1))=\sum_{x}B(t)\log\frac{B(t)}{B(t+1)} \qquad (4)$$

where *x* denotes any possible instance of the Bayesian network.

The following figures present the KL Divergence difference between the two search algorithms when data size equals to 100.

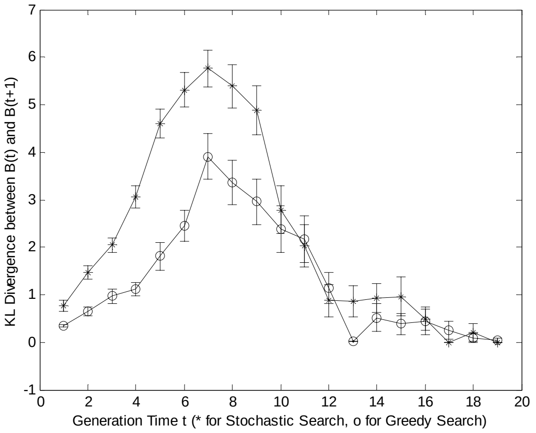

Fig. 3 KL Divergence of 100 runs

Fig. 3 shows the average KL Divergence of 100 runs between the constructed Bayesian network at time *t* and the one at time *t+1* for two search algorithms.

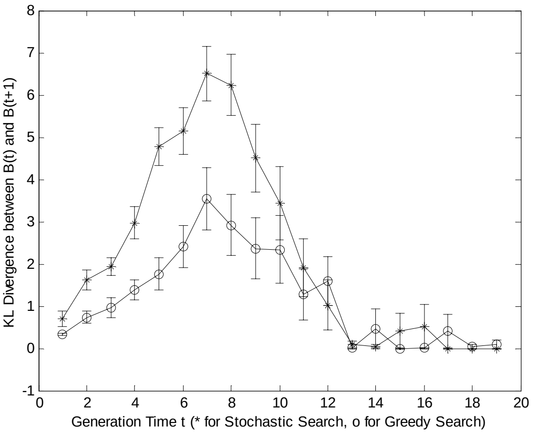

Fig. 4 KL Divergence of results that SS found Optimum but GS did not

Fig. 4 shows the average KL Divergence of results that SS found Optimum but GS did not, between the constructed Bayesian network at time *t* and the one at time *t+1* for two search algorithms.

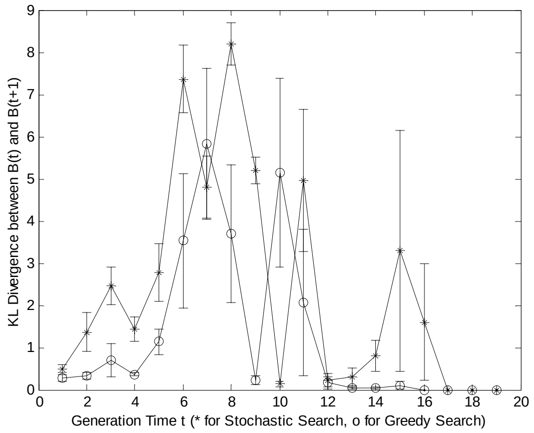

Fig. 5 KL Divergence of results that GS found Optimum but SS did not

Fig. 5 shows the average KL Divergence of results that GS found Optimum but SS did not, between the constructed Bayesian network at time *t* and the one at time *t+1* for two search algorithms.

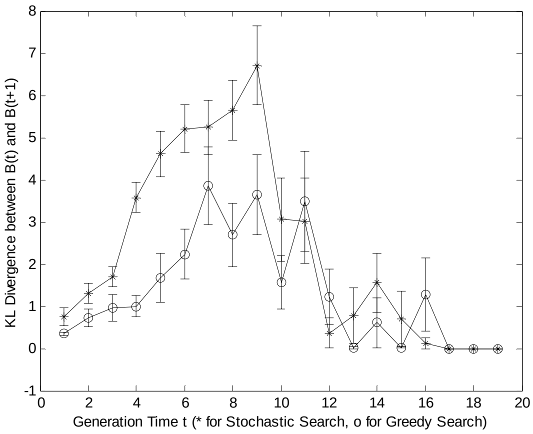

Fig. 6 KL Divergence of results that neither found Optimum

Fig. 6 shows the average KL Divergence of results that neither found Optimum, between the constructed Bayesian network at time *t* and the one at time *t+1* for two search algorithms.

As the series of KL Divergence figures present, the distances between the two curves for search algorithms shows that SS got larger KL Divergence at each generation than GS did, which means that the former obtains more different Bayesian network than the latter.

V. Discussion

From the experiment Part I, statistic results show that the stochastic search algorithm outperformed the greedy search algorithm for this optimization problem under given condition. This means that the SS owns stronger ability to generate new and useful strings than the GS. To explain this difference, a method using KL Divergence is introduced in the experiment Part II. The figures obtained using this method show that the stochastic search algorithm got larger KL Divergence between generations than the greedy search algorithm did. It means that, in each generation, the former got more different Bayesian network than the latter, which probably increased the chance to generate new and useful strings, and finally made the Stochastic Search outperformed the Greed Search in our experiment. The method using KL Divergence could reflect but not yet fully measure this ability of search algorithms. Further research work about this is needed.

References

[1] T. M. Cover, *Elements of information theory,* Wiley series in telecommunications, New York, 1991.

[2] Martin Pelikan, David E. Goldberg and Erick Cantu-Paz, "BOA: The Bayesian Optimization Algorithm," IlliGAL Report No. 99003. Illinois Genetic Algorithms Laboratory 1999.

[3] Martin Pelikan, David E. Goldberg, Kumara Sastry, *Bayesian* "Optimization Algorithm, Decision Graphs and Occam's Razor," IlliGAL Report No.2000020 Illinois Genetic Algorithms Laboratory, 2000.

[4] Judea Pearl, *Probabilistic Reasoning in Intelligence Systems*, Morgan Kaufmann, San Mateo, CA, 1988.

[5] R. W. Robinson, *Counting Unlabeled Acyclic digraphs*. In C. H. C. Little, Ed., Combinatorial Mathematics V, volume 622 of Lecture Notes in Mathematics, Berlin, 1977.

[6] Martin Pelikan, David E. Goldberg, Erick Cantu-Paz, "Linkage Problem, Distribution Estimation and Bayesian Networks," Evolutionary Computation (2000) 8(3): 311-340.

[7] Peter Grunwald, "A Tutorial Introduction to Minimum Description Length Principle," Centrum voor Wiskunde en Informatica, 2004

[8] Pedro Larranaga, Jose A. Lozano, *Estimation of Distribution Algorithms*. University of the Basque Country, 2002.