Blocking Gibbs Sampling in Very Large Probabilistic Expert Systems

Claus Skaanning Jensen
Aalborg University
Fredrik Bajers Vej 7E
DK-9220 Aalborg Ø
Denmark
claus@iesd.auc.dk

Augustine Kong
University of Chicago
1118E 58th St.
Chicago IL 60637
kong@galton.uchicago.edu

Uffe Kjærulff
Aalborg University
Fredrik Bajers Vej 7E
DK-9220 Aalborg Ø
Denmark
uk@iesd.auc.dk

25 August 1994

Abstract

We introduce a methodology for performing approximate computations in very complex probabilistic systems (e.g. huge pedigrees). Our approach, called *blocking Gibbs*, combines exact local computations with Gibbs sampling in a way that complements the strengths of both. The methodology is illustrated on a real-world problem involving a heavily inbred pedigree containing 20,000 individuals. We present results showing that blocking-Gibbs sampling converges much faster than plain Gibbs sampling for very complex problems.

Keywords: probabilistic expert system, graphical model, Bayesian network, junction tree, pedigree analysis, Monte Carlo.

1 Introduction

Over the last decade or so, fast and exact methods have been developed for computation in graphical models (Bayesian networks) of complex stochastic systems (Cannings, Thompson & Skolnick 1976, Cannings, Thompson & Skolnick 1978, Lauritzen & Spiegelhalter 1988, Shenoy & Shafer 1990, Jensen, Lauritzen & Olesen 1990, Dawid 1992, Lauritzen 1992, Spiegelhalter, Dawid, Lauritzen & Cowell 1993). The success of the exact methods has become a reality despite the fact that computation in Bayesian networks is generally NP-hard (Cooper 1990), i.e., there is often an exponential relationship between the number of variables and the complexity of computation. Thus, for a large class of real-world problems, exact computation is prohibitive.

Stochastic simulation techniques (Monte-Carlo methods) have thus become increasingly popular alternatives to exact methods, since they are flexible, easy to implement, and their computational complexity tends to scale manageably with the size of the networks under consideration (Gelfand & Smith 1990, Thomas, Spiegelhalter & Gilks 1992, Gelman & Rubin 1992, Geyer

1992, Smith & Roberts 1993). Their main disadvantages are associated with difficulty in deciding whether the desired precision has been reached and the fact that even moderately sized problems compute slowly. Their reliance on pseudorandom number generators may also to some extent be a problem (Ripley 1987). Using these simple Monte-Carlo methods, computation time often exceeds any acceptable level when considering very large networks (e.g. pedigrees of thousands of individuals).

The present paper suggests and evaluates a variant of Gibbs sampling (Geman & Geman 1984) involving simultaneous sampling of sets of variables using the junction tree architecture for exact local computations (Jensen 1988, Jensen et al. 1990, Dawid 1992). Since the method simulates sets (or blocks) of variables, we shall henceforth refer to it as *blocking Gibbs*. Further, Gibbs sampling involving blocks of size one shall be referred to as *plain Gibbs*. The evaluation of our blocking-Gibbs method is conducted as an empirical comparison study of the convergence properties of plain and blocking Gibbs for different size networks all of which are subnetworks of a real-world pedigree containing 20,000 breeding pigs. The pedigree is heavily inbred and several animals have an enormous amount of offspring which, in particular, causes serious problems for the convergence properties of plain Gibbs. Based on the outcome of this study, we present rules of thumb and general guidelines to obtain an optimal compromise between complexity and rate of convergence.

Section 2 reviews the exact local computation scheme for Bayesian networks which is used in our blocking-Gibbs method. The Monte-Carlo methods forward sampling and Gibbs sampling are also briefly reviewed. Section 3 describes the blocking-Gibbs method, and Section 4 reports the results of our empirical study of blocking Gibbs. Section 5 raises some yet unresolved issues and discusses some perspectives of the suggested method.

2 Well-known Methods for Belief Updating

Current methods for computing conditional marginal distributions given (categorical) evidence (i.e., information about the states of one or more variables) — often denoted *belief updating* — are based either on exact local computations or on various stochastic simulation techniques. Since the blocking-Gibbs methodology is based on elements from both categories, we briefly review the methods that will be employed in Section 3.

2.1 Exact Local Computations

Shachter, Andersen & Szolovits (1994) have shown that all exact methods for belief updating in Bayesian networks can be viewed as variations on a single, general algorithm involving clustering of variables, thus in effect creating a secondary structure, which is often called a *junction tree* (Jensen 1988). The clusters are the nodes of a junction tree and the cliques of a triangulated graph obtained by adding edges to the moral graph (Lauritzen & Spiegelhalter 1988) of the Bayesian network. A triangulated graph is an undirected graph with no cycles of length greater than 3 without a chord. The moral graph of a Bayesian network is obtained by adding undirected edges between all pairs of disconnected nodes with common children and by subsequent replacement of directed edges by undirected ones.

In the present paper, we shall refer to an object-oriented version of the exact method of Lauritzen & Spiegelhalter (1988) which has been described by Jensen et al. (1990) and implemented in the expert-system shell Hugin (Andersen, Olesen, Jensen & Jensen 1989). In this scheme, belief updating is implemented through inward/outward message passing in a junction tree, where the intersections between neighbouring clusters are called *separators*. The set of clusters of a junction tree shall henceforth be denoted by $\mathcal{C}$ and the separator set by $\mathcal{S}$. The complexity of computation imposed by a junction tree is proportional to the complexity of the tree given as the sum of the complexities of the elements in $\mathcal{C} \cup \mathcal{S}$. If $\mathrm{Sp}(v)$ denotes the state space of variable v, then the complexity of a set $A \in \mathcal{C} \cup \mathcal{S}$ is given as

$$c(A) = \prod_{v \in A} |\mathrm{Sp}(v)|.$$

The junction-tree structure can be used as the basis for other kinds of computations. Dawid (1992) has described ways to compute, for example, the most probable instantiation of unobserved variables (i.e., the set of instantiations which, in the light of evidence, has maximal joint probability) and a random joint sample of the unobserved variables given evidence. The latter plays an important role in the blocking-Gibbs method.

2.2 Monte-Carlo Methods

Exact computation may be prohibitive due to either excessive computational complexity or inability to provide analytical solutions to a given inference problem, for example when computation of a marginal belief involves functions for which exact integration is impossible. Various Monte-Carlo methods often provide successful approximate solutions in these cases. We shall briefly review two such methods, namely forward sampling and (plain) Gibbs sampling both of which have complexity proportional to the complexity of the moral graph of the network under consideration.

2.2.1 Forward Sampling

Let p be a probability function on Sp(V), where V is a set of (discrete) variables, such that it factorizes according to a directed, acyclic graph, $\mathcal{G} = (V, E)$. That is, if $\mathrm{pa}(v)$ denotes the set of parents of $v \in V$, then

$$p(x) = \prod_{v \in V} p(x_v \,|\, x_{\mathrm{pa}(v)})$$

implying that p is Markov with respect to $\mathcal{G}$; see e.g. Lauritzen, Dawid, Larsen & Leimer (1990). A sample from p can be obtained as follows. Let $V_0 \subseteq V$ such that $\mathrm{pa}(v) = \emptyset$ for all $v \in V_0$. Since p is Markov with respect to $\mathcal{G}$, the variables in V_0 can be sampled independently. Let $V_1 \subseteq V \setminus V_0$ such that $\mathrm{pa}(v) \setminus V_0 = \emptyset$ for all $v \in V_1$. Now, since V_1 is a set of conditionally independent variables given V_0, they can be sampled independently. Continuing in this fashion until all variables have been sampled, we obtain a sample of the joint distribution. Thus, an approximation, $\hat{p}$, of p can be obtained by creating n samples, $x^1, \ldots, x^n$, from p and letting $\hat{p}(x_v) = \left|\{x^i : x^i_v = x_v\}\right| / \, n$. This sampling procedure is known as *forward sampling*.

This basic forward-sampling procedure works well even in the case where evidence, x_A, is available for a subset $A \subseteq V$ such that $\mathrm{pa}(v) \subset A$ for all $v \in A$. But if there is at least one $v \in A$ for which $\mathrm{pa}(v) \not\subset A$, then only the samples for which the A-marginal is identical to x_A should be taken into account. This variant of forward sampling, which has been investigated by Henrion (1988) under the name of *logic sampling*, gets increasingly inefficient as $p(x_A)$ decreases. A more general forward-sampling procedure, which is capable of handling non-categorical evidence is known under the name of *importance sampling* (see e.g. Yiannoutsos & Gelfand (1994)).

2.2.2 Gibbs Sampling

When evidence is available, importance sampling (of which logic sampling is a special case) is very inefficient when the probability of the evidence is small with respect to the sample density. In that case, the sampling technique known as *Gibbs sampling* (Geman & Geman 1984) may be a good alternative. The basic idea in (plain) Gibbs sampling can be explained as follows. Given the Markov blanket of a variable $v \in V$ (i.e., its parents, its children and the parents of its children), v is independent of the remaining variables, and can hence be sampled independently. Thus, given a legal instantiation of all unobserved variables, that is, a sample x_V such that $p(x_V) > 0$, the next sample can be obtained by visiting all unobserved variables in any order and drawing from their conditional distributions given their Markov blankets. Since the Markov blanket of $v \in V$ is identical to its neighbours in the moral graph of $\mathcal{G}$, Gibbs sampling operates on the moral graph as opposed to the directed, acyclic graph.

A legal initial configuration (sample) compatible with the set of observed variables, A, can be found using forward sampling. Still, finding one can be problematic when $p(x_A)$ approaches zero, since the expected number of iterations needed is $p(x_A)^{-1}$. However, in the case of diallelic pedigrees, a simple, fast and non-iterative method is available, see Appendix A.

Each iteration of the Gibbs sampler defines a component of a Markov chain whose equilibrium distribution is the joint distribution from which we want to simulate. Therefore, after many iterations, the samples can be considered a draw from the desired distribution. Since the initial sample might not be 'sufficiently representative' of the equilibrium distribution, the first k samples are discarded. The choice of k is based on rules of thumb like e.g. '5-10 % of the samples', and these k iterations are often denoted the *burn-in*.

3 Blocking-Gibbs Sampling

The blocking-Gibbs sampler, which is a generalization of the plain Gibbs sampler, will be described in detail in this section.

The rate of convergence of any sample estimate based on stochastic simulation depends heavily on the degree to which the samples are independent. Gibbs samples are obviously *dependent*, whereas forward samples are *independent*. Therefore the rate of convergence of the Gibbs sampler is less than or equal to the rate of the forward sampler, whenever evidence is absent. The degree of dependence among the samples is often expressed through the notion of *mixing rate*. A sampling scheme is said to mix slowly if the degree of dependence among consecutive samples is high, and, conversely, to mix quickly if the samples are independent or almost independent.

An obvious improvement of plain Gibbs sampling to obtain faster mixing can be achieved through simultaneous sampling of sets of variables. The cost of such a generalization lies in the increased complexity implied by computing the necessary joint probabilities. Our job is therefore to make an optimal choice of scheme in the range given by the following two extremes:

(1) Sampling one variable conditional on its Markov blanket is computationally simple but leaves the samples 'rather dependent' (slow mixing).

(2) Sampling all variables simultaneously makes the samples independent but is generally computationally intractable (fast mixing).

A set of variables which are sampled simultaneously shall be referred to as a *component*. The union of the components must naturally comprise all variables, but they need not necessarily be disjoint. Assume that we select k components, $E_1, \ldots, E_k$, such that the variables in E_i, $i = 1, \ldots, k$, can be sampled simultaneously conditional on $V \setminus E_i$, and such that $\bigcup_i E_i = V$. For each E_i, we define $A_i = V \setminus E_i$. The components shall also be denoted the *E-sets*, and their complementary sets the *A-sets*.

The blocking-Gibbs algorithm now visits the components sequentially. When a component, E_i, is visited, it is simulated (i.e., a sample of it is drawn) conditional on the current configuration of A_i. When all components have been visited, one iteration of blocking Gibbs has been completed. As the union of the components is V, we see that all variables must have been sampled at least once. It is possible, however, that some variables have been sampled several times.

The parameters determining the efficiency of blocking Gibbs is the size of the A-sets and the number, k, of A-sets. The lower bound on the size of the A-sets is given by the maximum size of the components. We want each component to be as large as possible to maximize the rate of convergence. On the other hand, the complexity of simulating a component increases with increasing size of the component.

The approach applied to simulate a component, E_i, is based on the random propagation variant of exact local computations in a junction tree (Ploughman & Boehnke 1989, Ott 1989, Kong 1991, Dawid 1992). The clusters of the junction tree are subsets of variables in E_i. We construct this junction tree by first constructing a junction tree for the entire Bayesian network (i.e., we moralise and triangulate the directed, acyclic graph of the network). This large junction tree (which is much too large for exact computations) is then reduced by removing a sufficiently large set of variables from the junction tree to allow exact computations. This set of variables is A_i.

The removal of an observed variable v from a set $A \in \mathcal{A} = \mathcal{C} \cup \mathcal{S}$ (a cluster or a separator) imposes a reduction of complexity given by

$$\delta_{\mathcal{A}}(A, v) = \begin{cases} 0 & \text{if } v \notin A \\ c(A) & \text{if } \exists A^* \in \mathcal{A} \setminus \{A\} : A \setminus \{v\} \subset A^* \\ c(A)\left(1 - c(v)^{-1}\right) & \text{otherwise,} \end{cases}$$

where a reduction of $c(A)$ is obtained when the cluster (separator) with v removed is a proper subset of another cluster (separator), in which case A disappears from the junction tree. Thus, the total reduction of complexity for a junction tree $(\mathcal{C}, S)$ is then

$$\Delta_{\mathcal{C},S}(v) = \sum_{C \in \mathcal{C}} \delta_{\mathcal{C}}(C, v) + \sum_{S \in \mathcal{S}} \delta_{\mathcal{S}}(S, v).$$

In the sequel, we shall use Δ instead of $\Delta_{\mathcal{C},S}$, as the relevant junction tree shall be understood. Further, we shall use $\Delta(A)$ as shorthand for $\sum_{v \in A} \Delta(v)$.

Therefore, to maximize the size of component E_1, the variables in the corresponding A-set, A_1, should be those which provide maximal reduction of the complexity of the junction tree. The first variable, v_1, to be included in A_1 is the one which maximizes Δ. This variable is then removed from all of the clusters and separators of which it is a member. As mentioned above, that might cause some clusters (and their associated separators) to become proper subsets of other clusters, and hence the junction tree should be restructured. Similarly, the next variable, v_2, to be included in A_1 is selected from the new reduced (and possibly restructured) junction tree. This process continues until the complexity of the junction tree gets sufficiently low to allow exact computations. Let $(v_1, v_2, \ldots, v_{|V|})$ be a total ordering of the variables such that $\Delta(v_i) \geq \Delta(v_{i+1})$ for all $i = 1, \ldots, |V| - 1$, and let $A_1 = \{v_1, \ldots, v_n\}$.

Provided simultaneous sampling of V is prohibitive, the number of components must obviously be at least two. Therefore, since the above procedure selects an optimal A-set (A_1), we are forced to let the remaining A-sets, $A_2, \ldots, A_k$, be sub-optimal in the sense that $|A_i| \geq |A_1|$. However, using A_1 as a base set (i.e., $A_2, \ldots, A_k$ are composed by substituting some of the variables in A_1 with variables from $\{v_{n+1}, \ldots, v_{|V|}\}$), the remaining A-sets can be as small as possible. In composing the remaining A-sets, we must make sure that $\bigcap_{i=1}^k A_i = \emptyset$. One way of composing the remaining $k-1$ A-sets based on A_1 could be to define

$$A_i \cap A_1 = \{v_j \mid (j - i + 1) \bmod (k-1) \neq 0\},\ 1 < i \leq k, \tag{1}$$

and let $r_i > 1$ be the integer such that

$$\Delta(\{v_{n+1}, \ldots, v_{n+r_i}\}) \geq \Delta(A_1 \setminus A_i). \tag{2}$$

Then

$$A_i \setminus A_1 = \{v_{n+1}, \ldots, v_{n+r_i}\}. \tag{3}$$

Three parameters determine the composition of the A-sets, namely the number, k, of A-sets, the size, n, of the initial A-set, A_1, and the algorithm for composing the A-sets. In Section 4.4, an empirical investigation is conducted to determine the optimal values of these parameters.

4 Empirical Investigation

In this section we will perform a thorough empirical investigation of blocking Gibbs. First, however, we present the real-world problem used as the basis for the experiments, and we describe the prerequisites of the experiments. Two investigations are performed: a comparison of blocking and plain Gibbs, and a sensitivity analysis of blocking Gibbs with respect to choice of parameter values.

4.1 A Real-World Problem

The experiments are based on an extremely complex real-world problem, namely estimation of genotype probabilities for individuals in a heavily inbred pedigree containing approximately 20,000 breeding pigs. The maximum number of generations from top to bottom of the pedigree is 13. Each individual may have a hereditary trait, PSE, which causes the meat to be unfit for human consumption. This trait is caused by a simple genetic effect. Only two alleles are important, N and n, yielding three genotypes, NN, Nn and nn. The PSE disease is present if the genotype is nn.

The genetic model is Mendelian which simply means that the offspring have probability 0.5 of receiving a particular gene from one of its parents. If the frequency of N in the population is p, and the frequency of n is $q = 1 - p$, and Hardy-Weinberg proportions are assumed (see e.g. Thompson (1986)), the prior probability distribution of the genotypes of any founder, A, is $P(A) = (p^2, 2pq, q^2)$.

If the individuals A and B have offspring C, the conditional probabilities for the genotypes of C given the genotypes of A and B, are as shown in Table 1. In the experiments we have assumed no mutation, i.e., when a gene segregates from parent to offspring, it never changes into another gene, due to e.g. environmental effects.

A		NN			Nn			nn		
B		NN	Nn	nn	NN	Nn	nn	NN	Nn	nn
	NN	1	0.5	0	0.5	0.25	0	0	0	0
C	Nn	0	0.5	1	0.5	0.5	0.5	1	0.5	0
	nn	0	0	0	0	0.25	0.5	0	0.5	1

Table 1: The conditional probability distribution of the genotypes of offspring C given the genotypes of parents A and B.

4.2 Prerequisites of Comparison

Before describing the actual comparison study we shall describe the assumptions and conditions applied in the comparisons.

The comparisons were carried out on three subsets of the pedigree, referred to as Pedigree A (455 variables), Pedigree B (704 variables), and Pedigree C (1894 variables). We use these relatively small networks, since, for statistical purposes, a large number of full computations are performed on each network. Pedigree B was constructed from Pedigree A by adding a suitable amount of parents, offspring, and parents of offspring of the individuals in Pedigree A. Similarly, Pedigree B is contained in Pedigree C. We chose to avoid evidence to ease the comparison task.

We compute the convergence rate of a sampling scheme by comparing the resulting approximated distributions with the correct ones, i.e., the equilibrium distribution of the associated Markov chain. Since our investigations are based on absence of evidence, the 'correct' distributions can be obtained by means of forward sampling with a sample size of e.g. $n = 10^6$. Given n and an estimate $\hat{p}_{ij}$ of p_{ij} (the probability of variable i being in state j) obtained by forward sampling, a confidence interval can be computed for p_{ij} by utilizing the asymptotic behaviour of the distribution function for $\hat{p}$.

The metric used for calculating the accuracy of a result is the average mean square error,

$$\mathrm{mse} = \frac{1}{|V|}\sum_{i=1}^{|V|}\frac{1}{|\mathrm{Sp}(v_i)|}\sum_{j=1}^{|\mathrm{Sp}(v_i)|}\left(\hat{p}_{ij} - p_{ij}\right)^2.$$

Empirically, the values of the average mean square errors computed from the simulations are fitted to the model

$$\mathrm{mse} = \beta \cdot t^{\alpha} + \mathrm{noise}, \tag{4}$$

which specifies a linear relationship between $\log(\mathrm{mse})$ and $\log(t)$. In theory, for large enough t, which depends on the mixing rate of the Gibbs sampler, this model is supposed to hold approximately with $\alpha = -1$. If the fitted value of α is substantially bigger than -1, it is an indication that the particular Gibbs sampling scheme is mixing very slowly.

The comparison between blocking and plain Gibbs is performed with suboptimal parameter values for blocking Gibbs. In Section 4.4, we shall conduct a sensitivity analysis to reveal the impact of suboptimal choice of parameter values. The A-sets were constructed according to the following two methods. Unless stated otherwise, Method 1 has been applied.

4.2.1 Construction Method 1

Using Method 1, $A_1 = \{v_1, \ldots, v_n\}$ is constructed as described in Section 3, with the exception that the cost reduction imposed by a variable v is given by the number of clusters of which it is a member. The A-sets $A_2, \ldots, A_k$ are constructed as described by (1)–(3), with the exception that $r_2 = \cdots = r_k = n/(k-1) + 1$ (which appeared to be sufficient to allow exact computation for all E-sets, $E_2, \ldots, E_k$).

This method has the advantage that A_1 can be rather small, as most of the complexity-reducing variables are contained in A_1. On the other hand, $A_2, \ldots, A_k$ have to be larger than A_1.

4.2.2 Construction Method 2

Alternatively, the A-sets could be constructed such that they are all equally sized.

To construct k A-sets of size n, we select the set, A, of $n + r$ $(r > 0)$ variables appearing in the largest number of clusters. The A-sets are now given by

$$A \cap A_i = \{v_j \mid (j - i) \bmod k \neq 0\},\ 1 \leq i \leq k,$$

whereby we make sure that $\bigcap_{i=1}^{k} A_i = \emptyset$.

4.3 Comparison of Blocking and Plain Gibbs

We will now compare the rates of convergence for blocking and plain Gibbs by measuring the average mean square errors for various sample sizes. Below we show and discuss the results of the comparisons for Pedigrees A–C.

4.3.1 Pedigree A

Using the above terminology, the blocking-Gibbs parameters are $k = 5$ and $n = 50$ for Pedigree A. The results presented in Figure 1 depicts the average mean square error of the estimates obtained for various sample sizes.[1] From this figure we observe that blocking Gibbs converges much faster than plain Gibbs. However, if the Gibbs sampling were limited in time to, say, 20 minutes, plain Gibbs would possibly provide the most accurate estimates. After this point, the precision of blocking Gibbs gets increasingly better than that of plain Gibbs.

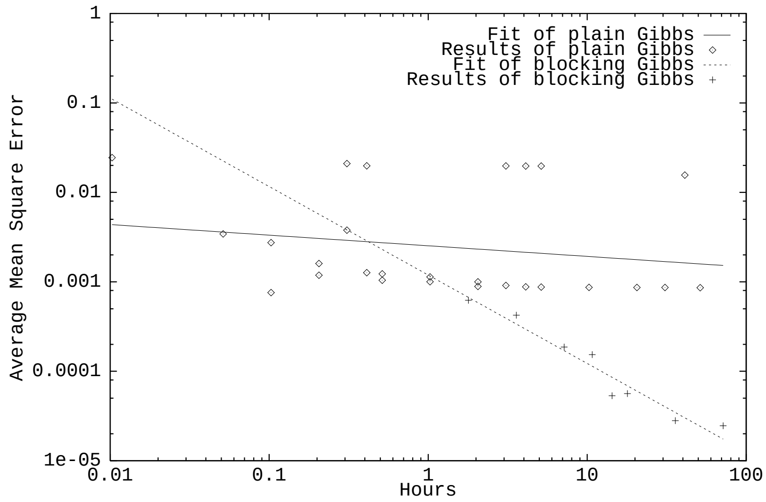

Figure 1: Blocking Gibbs vs. plain Gibbs for Pedigree A.

We also observe that the measurements of blocking Gibbs can be fitted very nicely to a straight line, as opposed to those of plain Gibbs, which seem to reside in two or more 'modes'.[2] We believe that this behaviour is caused by the presence of a single individual with many offspring. When such an individual is present, it will be very difficult to switch from one subset to another, thus making the Markov chain get stuck in one of the modes.[3]

4.3.2 Pedigree B

The blocking-Gibbs parameters are $k = 5$ and $n = 100$ for Pedigree B. The results are presented in Figure 2. Again, blocking Gibbs converges faster than plain Gibbs, and better precision can be obtained with blocking Gibbs except for very short runs. Here, the measurements of plain Gibbs do not indicate two or more distinct 'modes' of the Markov chain, probably due to the fact that more than one individual of Pedigree B have many offspring, yielding a larger number of modes.

[1] These and all subsequent results were obtained on a Sun 4-40 workstation.
[2] The mse measurements have been fitted to straight lines (cf. (4)) using linear regression.
[3] We will elaborate on this issue later on.

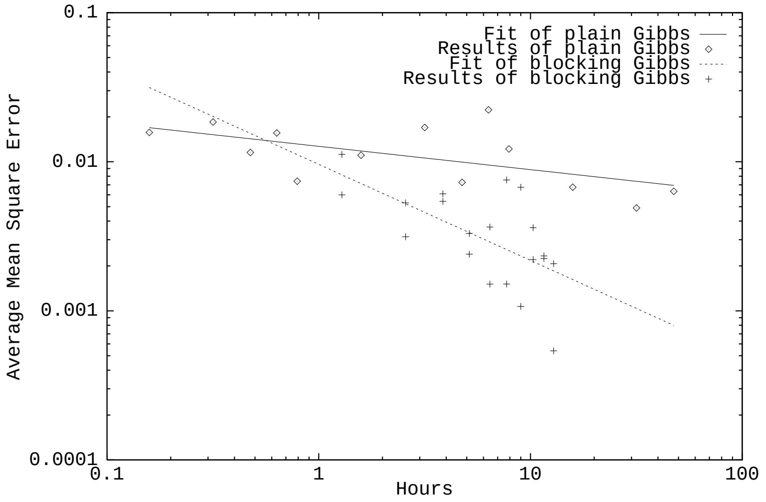

Figure 2: Blocking Gibbs vs. plain Gibbs for Pedigree B.

4.3.3 Pedigree C

The blocking-Gibbs parameters are $k = 5$ and $n = 200$ for Pedigree C. The results are shown in Figure 3. As in the previous cases, blocking Gibbs converges faster than plain Gibbs, but here, it is possible that better results can be obtained with plain Gibbs for runs shorter than 10 hours.

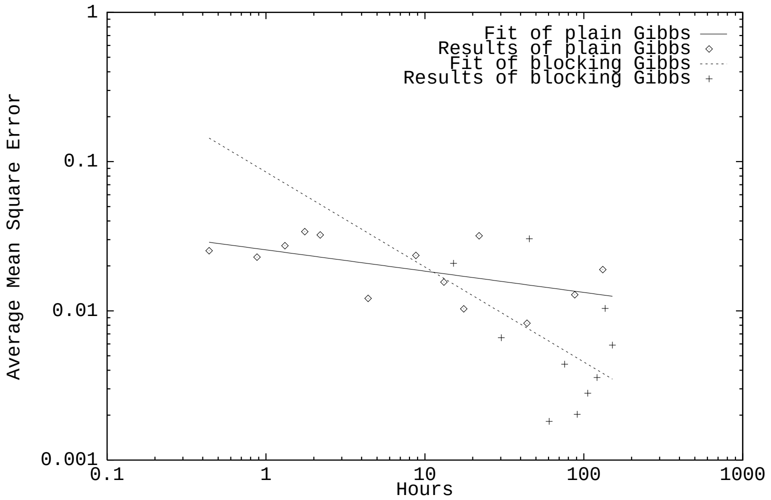

Figure 3: Blocking Gibbs vs. plain Gibbs for Pedigree C.

4.3.4 Summary

In general, the above results show that blocking Gibbs converges faster than plain Gibbs for large, complex pedigrees. But better results can often be obtained by plain Gibbs when an upper bound on the time complexity is imposed.

All the above comparisons were based on suboptimal parameters for blocking Gibbs. Thus, the superiority of blocking Gibbs over plain Gibbs can be expected to be even more pronounced (cf. Section 4.5). Optimization of the blocking-Gibbs parameters is the issue of the following section.

4.4 Adjusting Parameter Values for Blocking Gibbs

We now conduct an empirical sensitivity analysis of the rate of convergence of blocking Gibbs with respect to choice of parameter values. As mentioned previously, the parameters are the size, n, of A_1, the number, k, of A-sets, and the method applied to construct the A-sets. The search for optimal parameter values is conducted by varying n, k, and the construction method (Method 1 or Method 2; see Sections 4.2.1–4.2.2). We select a sample size of 1000 for each configuration of parameter values investigated. The comparison of the different configurations shall be based on the performance measure

$$\mathrm{perf} = \lambda \cdot \log(\mathrm{mse}) + \log(t).$$

This performance measure expresses the fact that two configurations yielding, respectively,

(1) mse $= 0.001$ in $t = 100$ seconds, and

(2) mse $= 0.01$ in $t = 10$ seconds,

are usually not equally good. If configuration 2 had been run for 100 seconds, we would not necessarily have obtained an mse of 0.001, but probably a higher value (lower precision). The coefficient λ is chosen such that two points on a line following (4) have identical performance measures.

We have performed the sensitivity analysis for Pedigree B only.

4.4.1 Size of A-sets

We present three figures showing respectively mse, time and performance as a function of the size of A-sets, n. In all three figures, results are presented for both Method 1 and Method 2. The results are as follows.

Average mean square error. See Figure 4. The results for Method 1 are denoted 'mse (1)', and likewise for Method 2. For both methods the mse seems to decrease as n decreases. This behaviour is anticipated, since in the limit, where the A-sets are empty, the samples become independent, and the larger the A-sets, the more blocking Gibbs resembles plain Gibbs (i.e., the samples become 'maximally dependent').

Iteration time. See Figure 5. As expected, the iteration time increases enormously as n decreases for small A-sets. That is, the size of the E-sets increases, resulting in large clusters of the junction trees, which slows down the computations.

Performance. See Figure 6. For both methods, optimal performance is obtained for n ranging from 27 to 40.

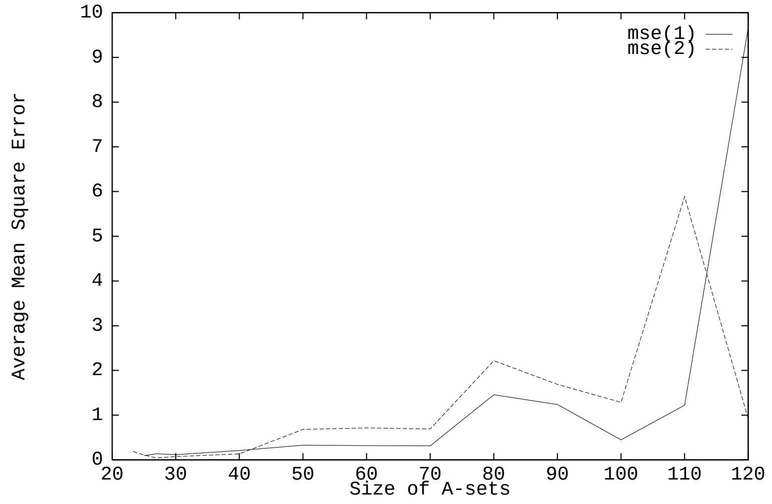

Figure 4: Precision of blocking Gibbs as a function of the size of A-sets using Method 1 and Method 2.

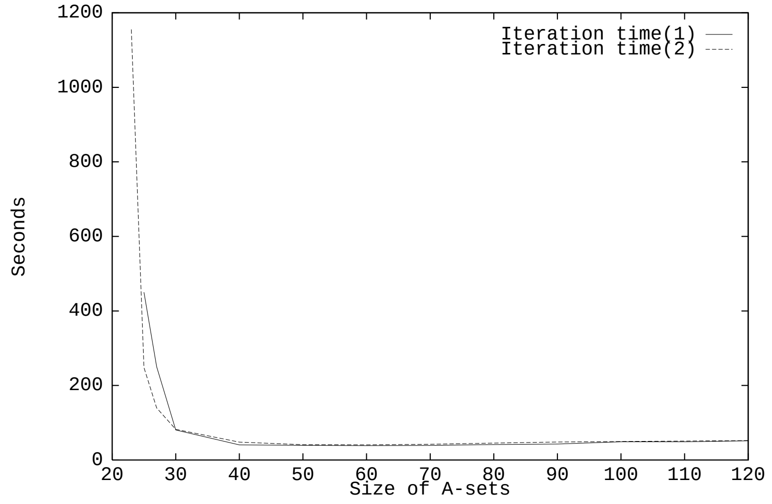

Figure 5: Iteration time of blocking Gibbs as a function of the size of A-sets using Method 1 and Method 2.

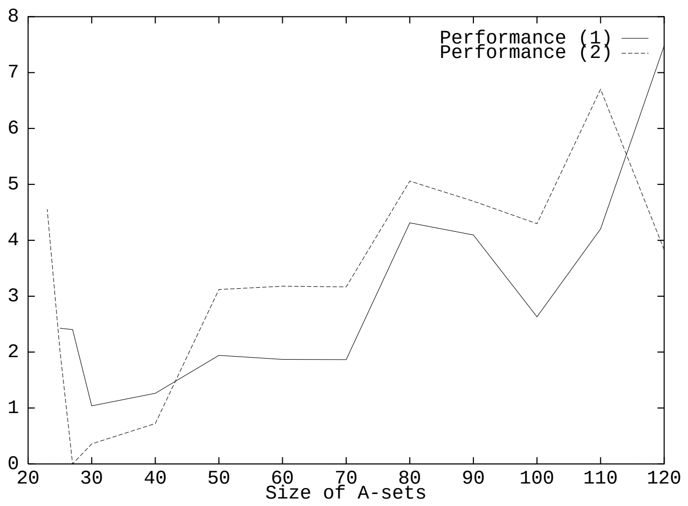

Figure 6: Performance of blocking Gibbs as a function of the size of A-sets using Method 1 and Method 2.

Notice two conflicting tendencies. As n increases, the iteration time decreases, but the mse increases. To find the optimal configuration, some compromise must be established. A rule of thumb may be to choose the size of the A-sets as small as possible while not increasing the iteration time significantly.

4.4.2 Number of A-sets

Again, three figures show mse, iteration time and performance as functions of the number of A-sets, k. The results are as follows.

Average mean square error. See Figure 7. The results reveal no obvious pattern, though it is clear that Method 2 is superior in all cases.

Iteration time. See Figure 8. It is clear that the iteration time increases when k is less than 4 and greater than 6. The optimal k-value seems to be 4, 5 or 6.

Performance. See Figure 9. Again, no obvious pattern can be observed, except for the fact that Method 2 is superior.

The results are not as clear as for the size of the A-sets. It seems that the mse does not depend on the number of A-sets. In this case the best choice may be the number of A-sets that yields the smallest iteration time. However, further investigations should be conducted.

4.4.3 Construction of A-sets

The performance of the two construction methods may be evaluated through further analysis of previous results (relative to the size and number of A-sets). The results are as follows.

Size of A-sets. See Figures 4, 5 and 6. Method 2 has the best overall performance for sizes in the range 27–40.

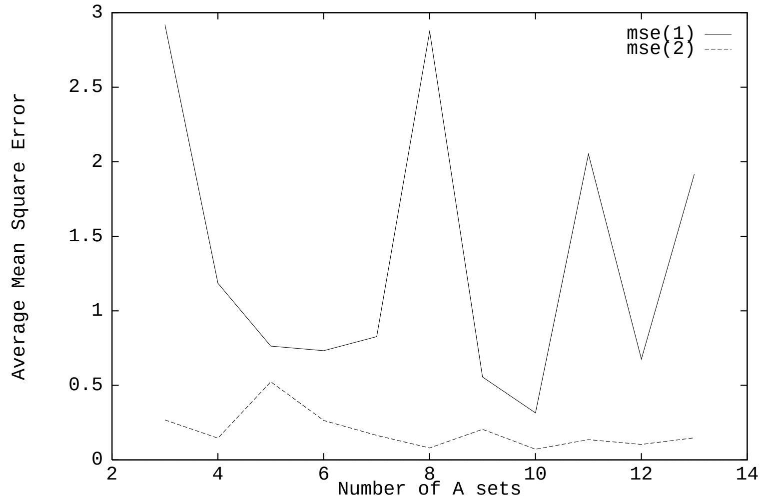

Figure 7: Precision of blocking Gibbs as a function of the number of A-sets using Method 1 and Method 2.

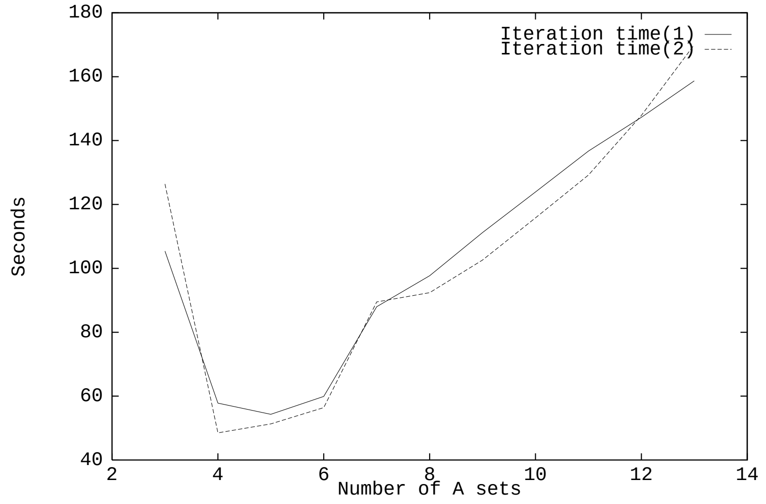

Figure 8: Iteration time of blocking Gibbs as a function of the number of A-sets using Method 1 and Method 2.

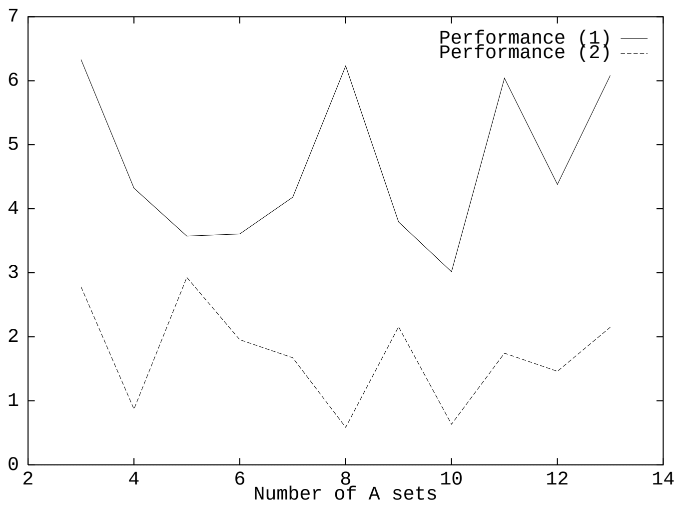

Figure 9: Performance of blocking Gibbs as a function of the number of A-sets using Method 1 and Method 2.

Number of A-sets. See Figures 7, 8 and 9. Method 2 is best in all cases.

The result is obvious: Method 2 should always be used.

4.4.4 Summary

From the results in the previous sections we can list a few rules of thumb for selecting optimal parameter values for blocking Gibbs.

(1) The size of A-sets should be as small as possible without increasing the iteration time significantly.

(2) The number of A-sets should be selected such that the iteration time becomes minimal.

(3) Method 2 should be used for construction of A-sets with the possible modification that the optimal cost reduction metric described in Section 3 should be applied.

Obviously, these guidelines are not as clear as we might wish, especially Rule 2 could probably be clarified by further investigations.

The investigations were performed without inclusion of any evidence. However, the presence of evidence does not affect the results obtained here.

Although the above results were derived from a heavily inbred pedigree of breeding pigs, it seems likely that they will also apply to other areas of interest, for example, human pedigrees which differ from pig pedigrees in that they are not as inbred and usually much more evidence is available.

4.5 Impact of Parameter Adjustment

The impact of employing optimal parameter values of blocking Gibbs will now be investigated for Pedigree B. The results of applying the following suboptimal and optimal parameter values are compared.

Suboptimal. Five A-sets, 100 variables in the initial A-set, Method 1 (used in Section 4.3).

Optimal. Six A-sets, 30 variables in the initial A-set, Method 2.

The result of this comparison can be seen in Figure 10 which displays the average mean square error (mse) as function of iteration time of blocking Gibbs with suboptimal and optimal parameters.

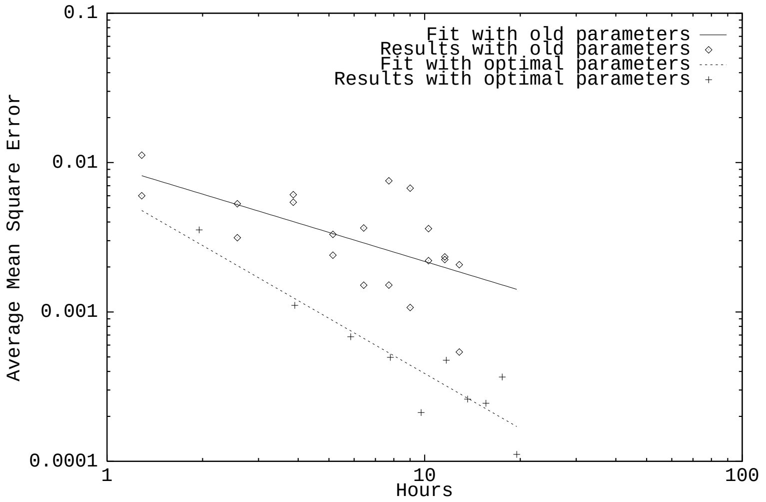

Figure 10: Blocking Gibbs with suboptimal parameters vs. blocking Gibbs with optimal parameters, Pedigree B.

It appears from Figure 10 that much faster convergence is obtained with optimal parameter values. Thus, it seems likely that the choice of parameter values has a great impact on the rate of convergence of blocking Gibbs.

5 Discussion

Using an application in genetic pedigree analysis, we have demonstrated how probability propagation (a method for performing exact computations) can be combined with Gibbs sampling (a Monte-Carlo method) to effectively solve problems with large complex network systems. The strengths of the two methods complement each other. While a rather detailed case study has been performed, some general questions about blocking Gibbs have to be addressed. For example, potential users need to understand in what type of situations will blocking Gibbs perform better than plain Gibbs after adjusting for the extra time needed to perform one iteration. Also, how to choose the blocks and how to effectively utilize the generated samples are important practical issues.

In general, plain Gibbs tends to perform very well if the network is not too large and the unobserved variables are not too highly dependent on each other. The empirical results in previous sections show that the performance of plain Gibbs gets worse as the size of the pedigree increases. The following example illustrates the reason for the deterioration and the rate of deterioration.

> *Example 1:* Consider a family consisting of a father, a mother and k offspring. Suppose the two alleles N and n are equally likely and there are no observed data on any member of the family. In that case, there is probability 1/16 that all members of the family have genotype NN and, by symmetry, there is probability 1/16 that all members of the family have genotype nn. Suppose plain Gibbs is applied with an initial configuration of all NN genotypes. It is clear that any changes have to start with one of the parents. Conditioned on the offspring being all NN, the genotype of the father (or the mother) can either be NN or Nn. The conditional odds of NN to Nn can be easily calculated to be $2^{k-1} : 1$. Hence the probability of changing from NN to Nn in an iteration decreases exponentially as k increases. As a consequence, the expected number of iterations needed for this change to take place increases exponentially. Furthermore, note that this is only one small step towards moving to the configuration of all nn.

While the above example demonstrates how quickly the performance of plain Gibbs can deteriorate, with only two alleles, it is at least true that the correct answer can be obtained if enough iterations are performed (see Sheehan & Thomas (1993)). This is not necessarily the case with three or more alleles.

> *Example 2:* Suppose a gene of interest has three alleles labeled as 1, 2 and 3. Consider a family consisting of a father, a mother and two offspring. Suppose there are no direct data on the parents, but the two offspring are observed to have genotypes 11 and 23. It is clear that the genotypes of the father and mother can either be 12 and 13 respectively, or 13 and 12 respectively. However, if plain Gibbs is applied, no matter how many iterations are performed, the genotypes of the two parents will stay at the initial configuration and never change. Obviously, if the two parents are treated as a block, the two configurations will be sampled with equal frequency.

In general, plain Gibbs will fail entirely if the induced Markov chain is *not irreducible*, which means that some of the configurations cannot *communicate* with each other. As demonstrated, this can happen for a very simple family structure in pedigree analysis and, indeed, can also easily happen to other expert systems. The problem occurs when the unobserved variables, e.g. the genotypes of the two parents in the above example, are too highly dependent on each other. This example not only highlights the potential superiority of blocking Gibbs, but also points out the importance of the choice of A sets, or equivalently, the choice of E-sets. If the simple family described above is part of a bigger pedigree, the Markov chain induced by blocking Gibbs will not be irreducible if none of the E-sets contains both the father and mother. Hence, finding an algorithm for choosing the A-sets which will at least guarantee irreducibility is a challenging and important problem.

Literature on the theoretical properties of (plain) Gibbs sampling has grown quickly in the last few years, but most of the results either do not apply to blocking Gibbs, or do not address the practical problems. For example, the only theoretical work, as far as we know, which studies the effect of blocking is Liu, Wong & Kong (1992) and it only considers the case where the blocks (the E-sets) do not overlap. In situations where the blocks do overlap, some of the variables are in more than one block and so there are multiple samples per iteration cycle. As we have done in the last two sections, a natural way of utilizing these multiple samples is to take a simple average of all of them. It is however unclear whether some sort of weighted average may not be superior. In general, more theoretical work on blocking Gibbs is necessary to guide the practical user.

Acknowledgements

This research was supported by the PIFT programme of the Danish Research Councils and the National Institutes of Health grant no. R01-GM46800. We are grateful to Lars Brenk and Henrik Wendt for their efforts in the original project that formed the basis for the present research. We also wish to thank Søren Andersen of Danske Slagterier for providing the pedigree data.

A Finding a Legal Instantiation in Diallelic Pedigrees

As pointed out in Section 2.2.2, it can be difficult to find a legal initial configuration for Gibbs sampling, i.e., an instantiation of all unobserved variables that is compatible with evidence. For diallelic pedigrees, however, it is very simple to find such an instantiation.

Theorem 1 *Let $V = \{v_1, \ldots, v_n\}$ be the set of variables of a diallelic pedigree with genotypes NN, Nn and nn, and joint probability function p complying with the assumptions of Section 4.1. Let $V_0 = \{v_1, \ldots, v_j\}$, $j < n$, be the set of variables the genotypes of which can be inferred from evidence either by direct observation or by logical implications from evidence. Then $p(x_{V_0}, x_{V \setminus V_0}) > 0$ when $x_v = Nn$ for all $v \in V \setminus V_0$.*

Proof: Let $V_i = V_{i-1} \cup \{v_{j+i}\}$ for $0 < i \leq n - j$. The proof is by induction on the number of instantiated variables. Assume that $p(x_{V_k}, x_{V \setminus V_k}) > 0$ and let $u = v_{j+k+1}$. Now, writing $p(x_{V_{k+1}}, x_{V \setminus V_{k+1}})$ as

$$p(X_u = \mathrm{Nn} \,|\, x_{\mathrm{pa}(u)}) \prod_{v \in \mathrm{ch}(u)} p(x_v \,|\, x_{\mathrm{pa}(v) \setminus \{u\}}, X_u = \mathrm{Nn}) \prod_{v \in V \setminus (\mathrm{ch}(u) \cup \{u\})} p(x_v \,|\, x_{\mathrm{pa}(v)}),$$

we see, by the induction hypothesis, that the last term is positive. (The set $\mathrm{ch}(u)$ is the set of children of u.)

By inspection of Table 1 we conclude that

$$p(x_v \,|\, x_{\mathrm{pa}(v)}) > 0 \;\Rightarrow\; p(x_v \,|\, x_{\mathrm{pa}(v) \setminus \{u\}}, X_u = \mathrm{Nn}) > 0.$$

To see that the first term is positive, assume that it is zero. From Table 1 it follows that the genotypes of $\mathrm{pa}(u)$ must be either both NN or both nn, implying that the genotype of u must be either NN or nn, and that $\mathrm{pa}(u) \cup \{u\} \subseteq V_0$. Contradiction! $\square$

References

Andersen, S. K., Olesen, K. G., Jensen, F. V. & Jensen, F. (1989). HUGIN — A shell for building Bayesian belief universes for expert systems, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 1080–1085.

Cannings, C., Thompson, E. A. & Skolnick, H. H. (1976). The recursive derivation of likelihoods on complex pedigrees, *Advances in Applied Probability* **8**: 622–625.

Cannings, C., Thompson, E. A. & Skolnick, H. H. (1978). Probability functions on complex pedigrees, *Advances in Applied Probability* **10**: 26–61.

Cooper, G. F. (1990). The computational complexity of probabilistic inference using Bayesian belief networks, *Artificial Intelligence* **42**: 393–405.

Dawid, A. P. (1992). Applications of a general propagation algorithm for probabilistic expert systems, *Statistics and Computing* **2**: 25–36.

Gelfand, A. E. & Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities, *Journal of the American Statistical Association* **85**: 398–409.

Gelman, A. & Rubin, D. B. (1992). Inference from iterative simulation using single and multiple sequences (with discussion), *Statistical Science* **7**: 457–511.

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(6): 721–741.

Geyer, C. J. (1992). Practical Markov Chain Monte Carlo (with discussion), *Statistical Science* **7**: 473–511.

Henrion, M. (1988). Propagating uncertainty in Bayesian networks by probabilistic logic sampling, *in* J. F. Lemmer & L. M. Kanal (eds), *Uncertainty in Artificial Intelligence 2*, Elsevier Science Publishers B. V. (North-Holland), Amsterdam, pp. 149–163.

Jensen, F. V. (1988). Junction trees and decomposable hypergraphs, *Research report*, Judex Datasystemer A/S, Aalborg, Denmark.

Jensen, F. V., Lauritzen, S. L. & Olesen, K. G. (1990). Bayesian updating in causal probabilistic networks by local computations, *Computational Statistics Quarterly* **4**: 269–282.

Kong, A. (1991). Efficient methods for computing linkage likelihoods of recessive diseases in inbred pedigrees, *Genetic Epidemiology* **8**: 81–103.

Lauritzen, S. L. (1992). Propagation of probabilities, means and variances in mixed graphical association models, *Journal of the American Statistical Association* **87**(420): 1098–1108.

Lauritzen, S. L., Dawid, A. P., Larsen, B. N. & Leimer, H.-G. (1990). Independence properties of directed Markov fields, *Networks* **20**(5): 491–505. Special Issue on Influence Diagrams.

Lauritzen, S. L. & Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems, *Journal of the Royal Statistical Society, Series B* **50**(2): 157–224.

Liu, J., Wong, W. H. & Kong, A. (1992). Correlation structure and convergence rate of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes, *Technical Report 299*, Department of Statistics, University of Chicago. Also in *Biometrika*.

Ott, J. (1989). Computer-simulation methods in linkage analysis, *Proceeding of the National Academy of Science, USA 86*, pp. 4175–4178.

Ploughman, L. & Boehnke, M. (1989). Estimating the power of a proposed linkage study for a complex genetic trait, *American Journal of Human Genetics* **44**: 543–551.

Ripley, B. D. (1987). *Stochastic Simulation*, Wiley & Sons.

Shachter, R. D., Andersen, S. K. & Szolovits, P. (1994). Global conditioning for probabilistic inference in belief networks, *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, California, pp. 514–522.

Sheehan, N. & Thomas, A. (1993). On the irreducibility of a Markov chain defined on a space of genotype configurations by a sampling scheme, *Biometrics* **49**: 163–175.

Shenoy, P. P. & Shafer, G. R. (1990). Axioms for probability and belief-function propagation, *in* R. D. Shachter, T. S. Levitt, L. N. Kanal & J. F. Lemmer (eds), *Uncertainty in Artificial Intelligence 4*, Elsevier Science Publishers B. V. (North-Holland), Amsterdam, pp. 169–198.

Smith, A. F. M. & Roberts, G. O. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods, *Journal of the Royal Statistical Society, Series B* **55**(1): 5–23.

Spiegelhalter, D. J., Dawid, A. P., Lauritzen, S. L. & Cowell, R. G. (1993). Bayesian analysis in expert systems (with discussion), *Statistical Science* **8**: 219–247 and 247–283.

Thomas, A., Spiegelhalter, D. J. & Gilks, W. R. (1992). BUGS: A program to perform Bayesian inference using Gibbs sampling, *in* J. M. Bernardo, J. O. Berger, A. P. Dawid & A. F. M. Smith (eds), *Bayesian Statistics 4*, Oxford University Press, Oxford, UK, pp. 837–842.

Thompson, E. A. (1986). *Pedigree Analysis in Human Genetics*, John Hopkins University Press.

Yiannoutsos, C. T. & Gelfand, A. E. (1994). Simulation approaches for calculations in directed graphical models, *in* S. Gupta & J. Berger (eds), *Statistical Decision Theory and Related Topics*, Vol. V, Springer-Verlag, N.Y., pp. 441–452.

A		NN			Nn			nn		
B		NN	Nn	nn	NN	Nn	nn	NN	Nn	nn
	NN	1	0.5	0	0.5	0.25	0	0	0	0
C	Nn	0	0.5	1	0.5	0.5	0.5	1	0.5	0
	nn	0	0	0	0	0.25	0.5	0	0.5	1

Table 1: The conditional probability distribution of the genotypes of offspring C given the genotypes of parents A and B.

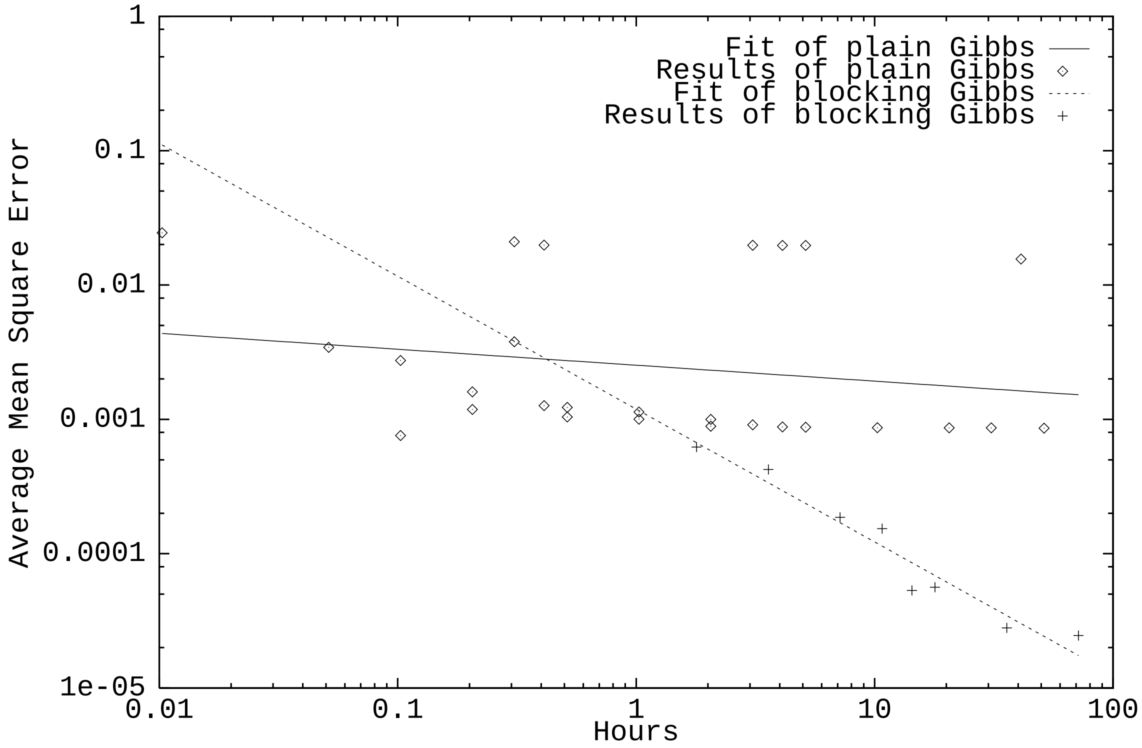

Figure 1: Blocking Gibbs vs. plain Gibbs for Pedigree A.

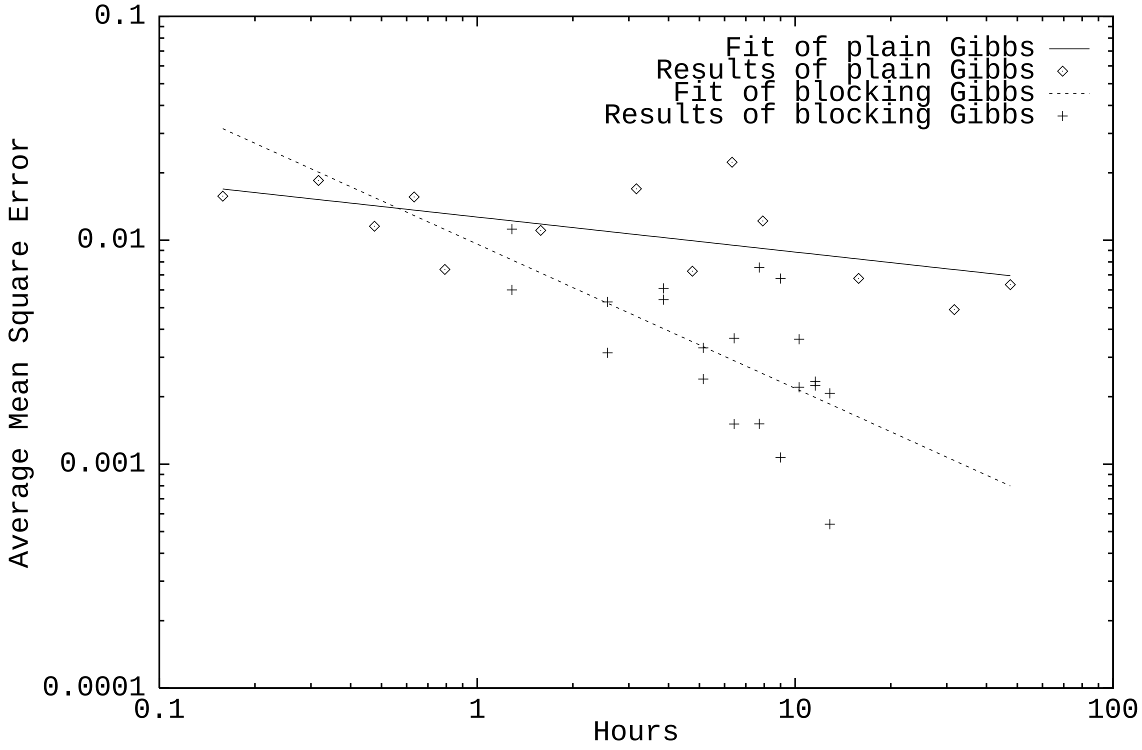

Figure 2: Blocking Gibbs vs. plain Gibbs for Pedigree B.

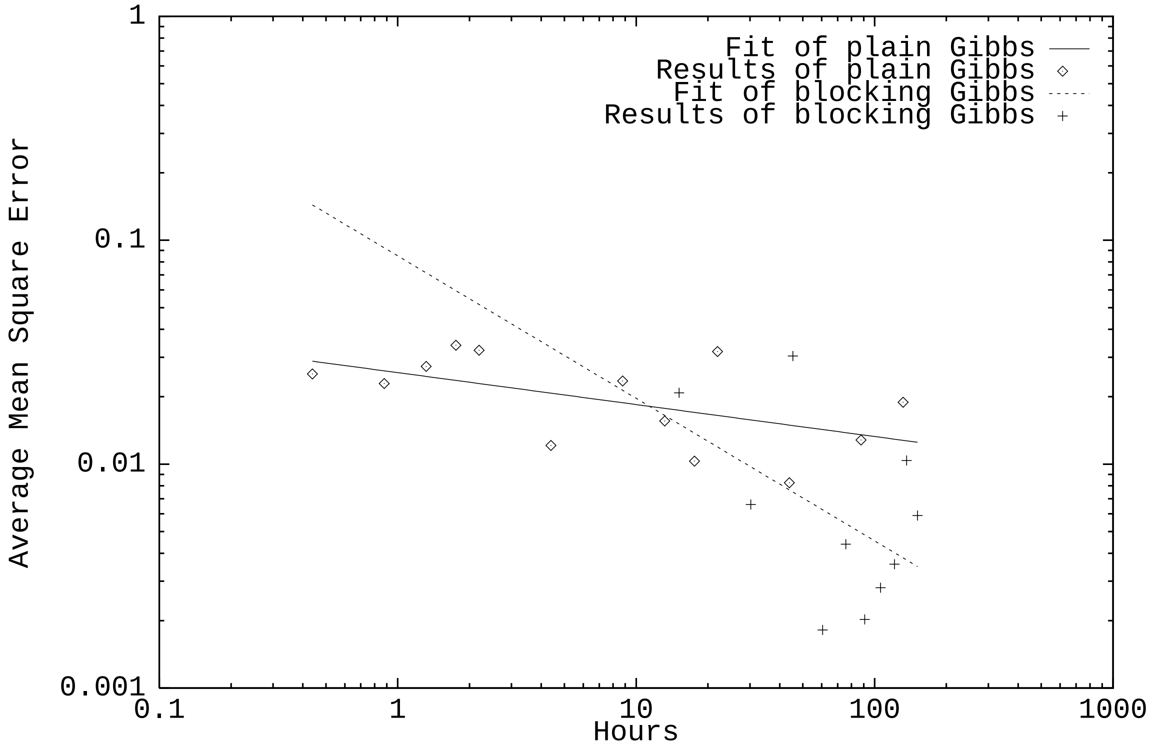

Figure 3: Blocking Gibbs vs. plain Gibbs for Pedigree C.

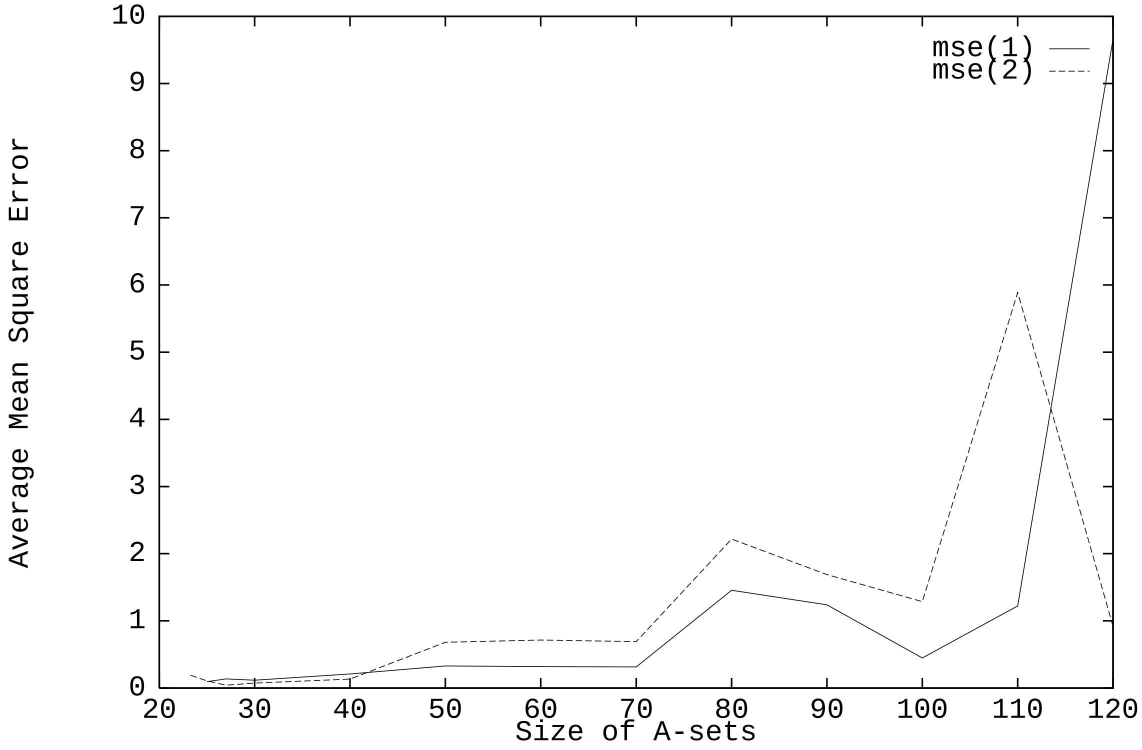

Figure 4: Precision of blocking Gibbs as a function of the size of A-sets using Method 1 and Method 2.

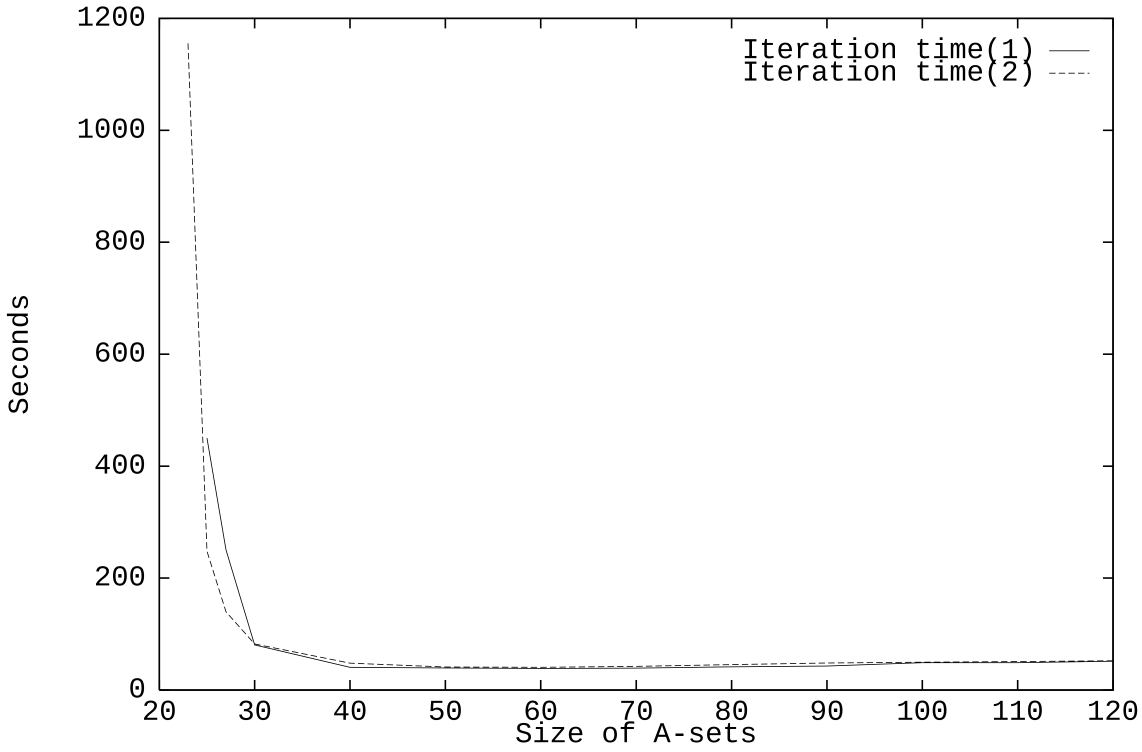

Figure 5: Iteration time of blocking Gibbs as a function of the size of A-sets using Method 1 and Method 2.

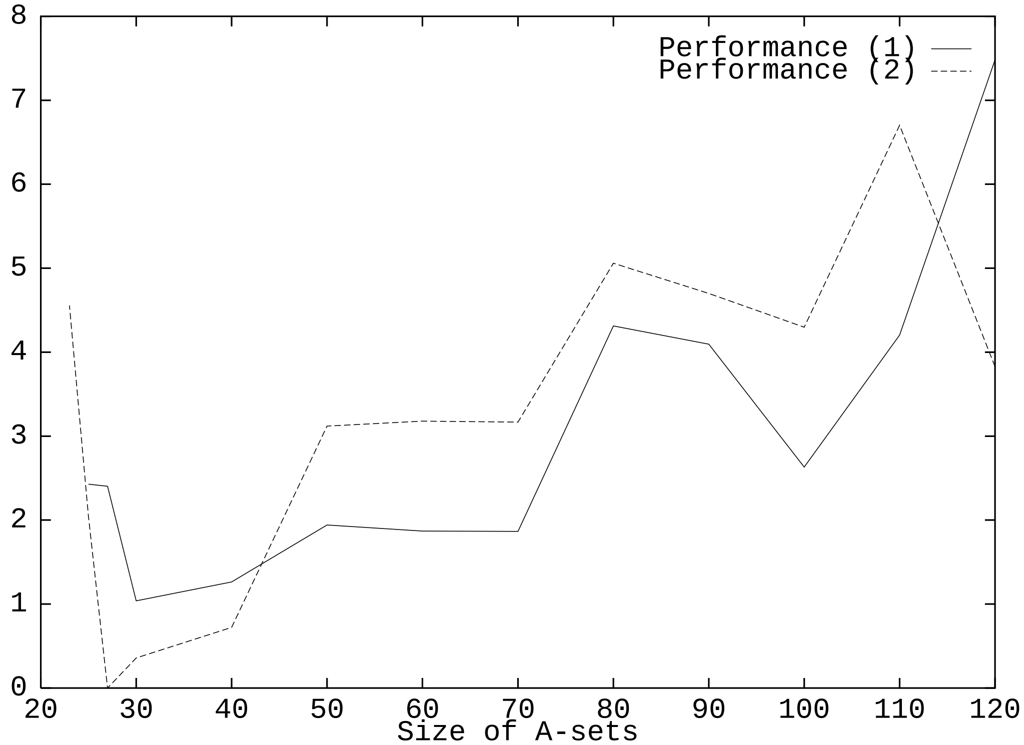

Figure 6: Performance of blocking Gibbs as a function of the size of A-sets using Method 1 and Method 2.

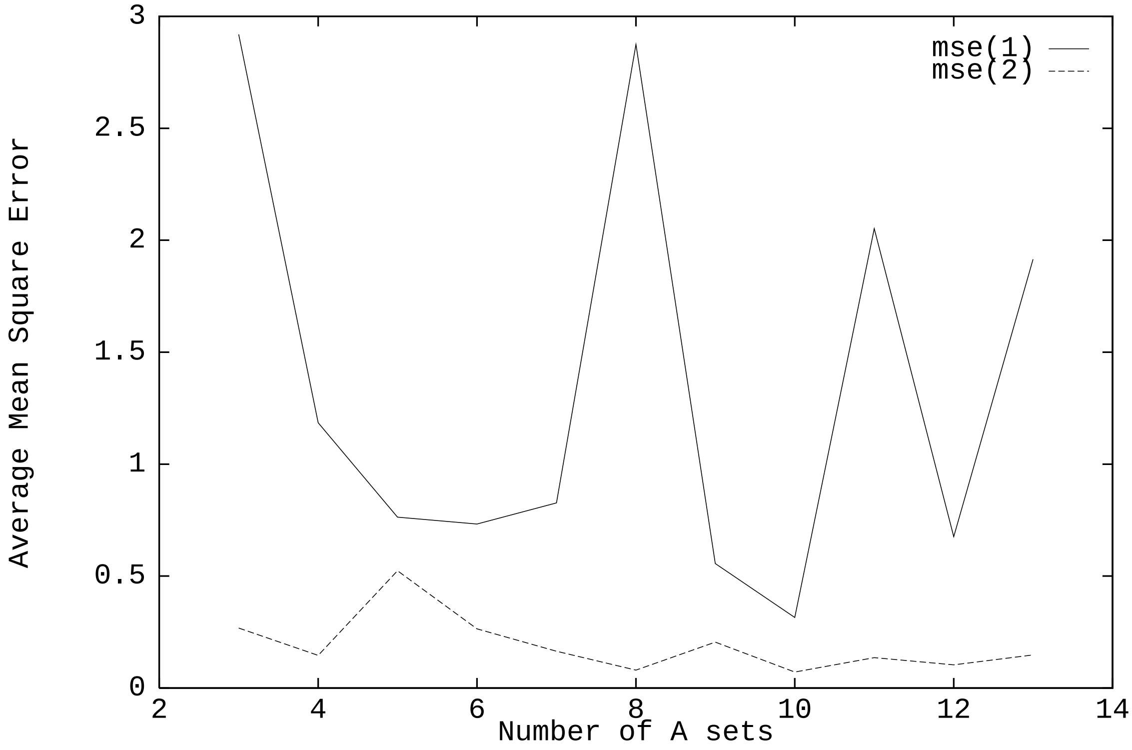

Figure 7: Precision of blocking Gibbs as a function of the number of A-sets using Method 1 and Method 2.

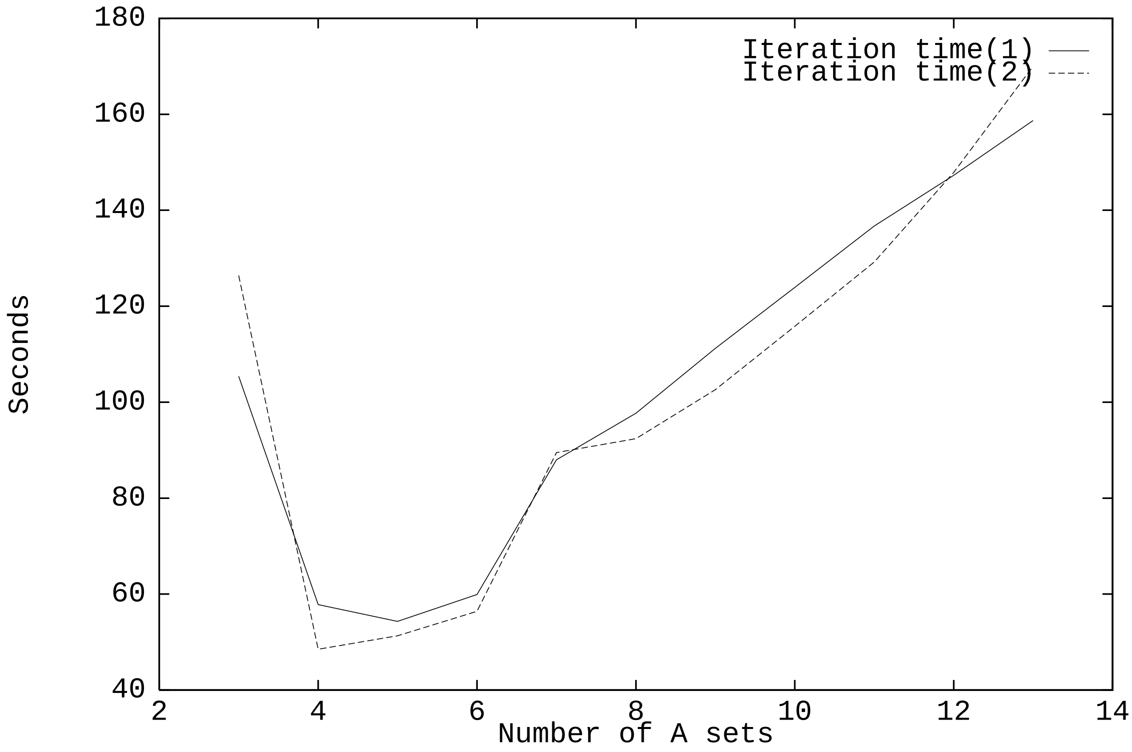

Figure 8: Iteration time of blocking Gibbs as a function of the number of A-sets using Method 1 and Method 2.

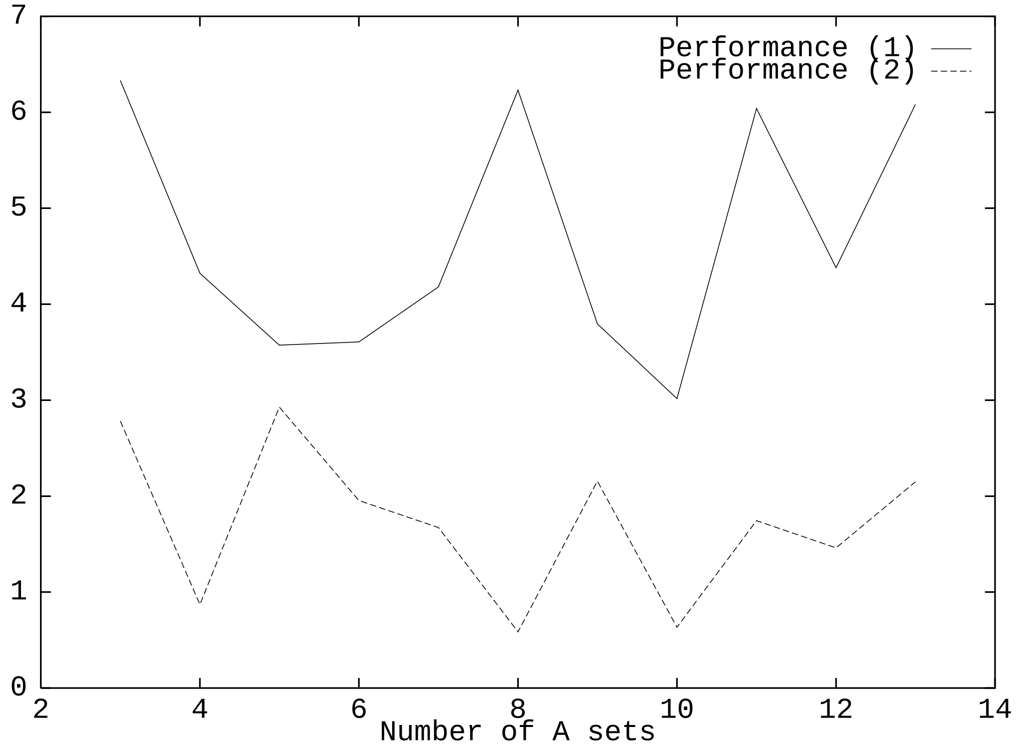

Figure 9: Performance of blocking Gibbs as a function of the number of A-sets using Method 1 and Method 2.

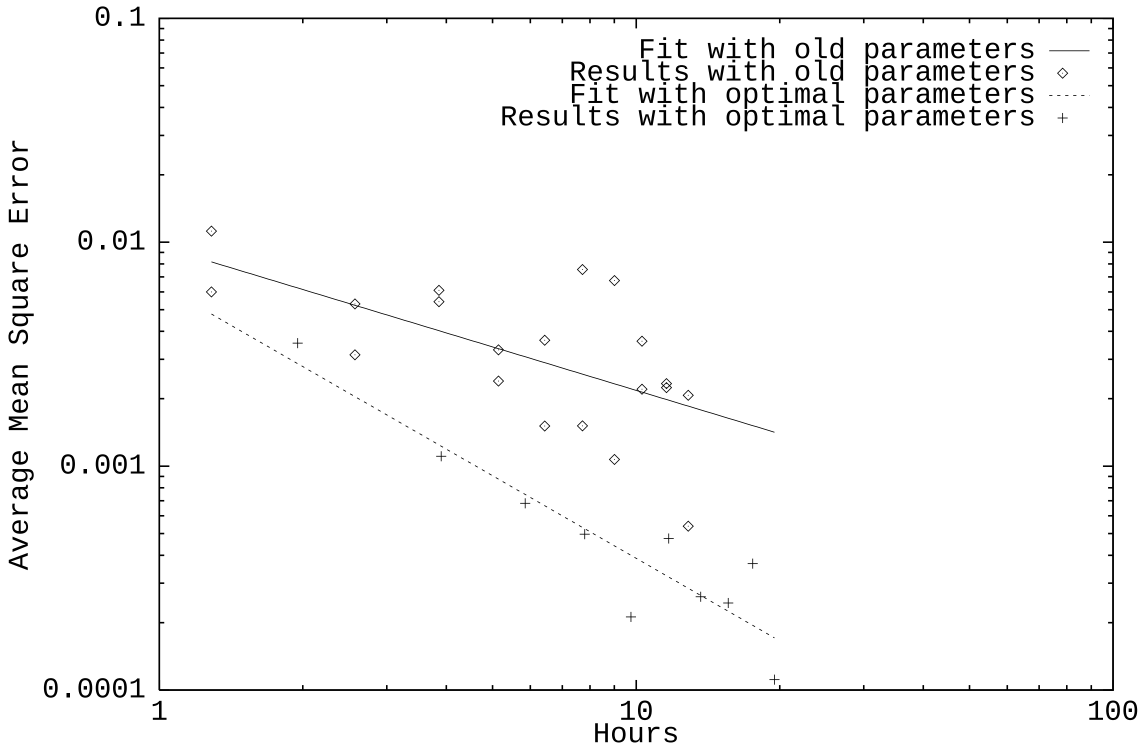

Figure 10: Blocking Gibbs with suboptimal parameters vs. blocking Gibbs with optimal parameters, Pedigree B.

Footnotes

(1) These and all subsequent results were obtained on a Sun 4-40 workstation.

(2) The mse measurements have been fitted to straight lines (cf. (4)) using linear regression.

(3) We will elaborate on this issue later on.

Postal address for the despatch of proofs and offprints

Uffe Kjærulff
Aalborg University
Fredrik Bajers Vej 7E
DK-9220 Aalborg Ø
Denmark