

Solving the Ising Spin Glass Problem using a Bivariate EDA based on Markov Random Fields

Siddhartha K. Shakya, John A.W. McCall, Deryck F. Brown
 School of Computing, The Robert Gordon University
 St. Andrew Street, Aberdeen, AB25 1HG, Scotland, UK
 {ss,jm,db}@comp.rgu.ac.uk

Abstract—Markov Random Field (MRF) modelling techniques have been recently proposed as a novel approach to probabilistic modelling for Estimation of Distribution Algorithms (EDAs). An EDA using this technique was called Distribution Estimation using Markov Random Fields (DEUM). DEUM was later extended to DEUM_d. DEUM and DEUM_d use a univariate model of probability distribution, and have been shown to perform better than other univariate EDAs for a range of optimization problems. This paper extends DEUM to use a bivariate model and applies it to the Ising spin glass problems. We propose two variants of DEUM that use different sampling techniques. Our experimental result show a noticeable gain in performance.

I. INTRODUCTION

Estimation of Distribution Algorithms (EDAs) [14] is a well-established topic in the field of evolutionary algorithms. EDAs are motivated by the idea of discovering and exploiting the interaction between variables in the solution. An EDA maintains the *selection* and *variation* concepts of evolution. However, it replaces the crossover and mutation approach to variation in a traditional GA by estimating and sampling a probabilistic model of promising solutions. EDAs are classified as univariate, bivariate or multivariate [7] according to the type of interaction between variables that can be represented by the probabilistic model.

In [25] an algorithm using a Markov Random Field (MRF) (also known as an undirected graphical model or a Markov Network) approach [16], [8], [5], [15] to probabilistic modelling has been proposed. This was called Distribution Estimation Using Markov Random Field (DEUM). DEUM was later extended to DEUM_d [23], [24]. DEUM and DEUM_d were presented as novel univariate EDAs and were shown to perform better than other EDAs of their type over a wide range of optimization problems [25], [23], [24].

This paper extends DEUM to bivariate problems and applies it to a well known Ising spin glass problem [6]. Spin glasses have a range of practical applications in both statistical physics and artificial intelligence. Due to their interesting properties, such as symmetry and a large number of plateaus, they have also been widely studied by the GA (and EDA) community [18], [17], [21], [22].

The outline of the paper is as follows. Section II describes the Ising spin glass problem and gives some background on previous applications of EDAs to it. Section III presents our approach to probabilistic modelling and defines the structure and parameters of our model for Ising spin glass problem. In

section IV, we describe how to estimate the model parameters from the population of solutions. Section V presents the results with Is-DEUM using a Metropolis sampling method. Section VI presents results with Is-DEUM using a Gibbs sampling method. Section VII presents some discussion of the results and concludes the paper.

II. ISING SPIN GLASS PROBLEM AND EDA

The general Ising spin glass problem can be described by an energy function, H , defined over a set of spin variables $\sigma = \{\sigma_1, \sigma_2, \dots, \sigma_l\}$ and a set of coupling constants h and J as

$$H(\sigma) = - \sum_{i \in L} h_i \sigma_i - \sum_{i < j \in L} J_{ij} \sigma_i \sigma_j \quad (1)$$

Here, each coupling constant $h_i \in h$ and $J_{ij} \in J$ relate to a single spin σ_i and a pair of spins σ_i and σ_j respectively. Each spin variable σ_i can be either +1 or -1. One specific choice of value for the spin variable is called a configuration. L is a lattice of n sites.

Given coupling constants h_i and J_{ij} , the task in the Ising spin glass problem is to find the value for each σ_i that minimises the energy, H . For the purpose of this paper, we only consider the coupling constants relating pairs of spin variables and therefore set $h_i = 0, \forall i \in L$. Additionally, we restrict J_{ij} to take only two values $J_{ij} \in \{+1, -1\}$. Here, we

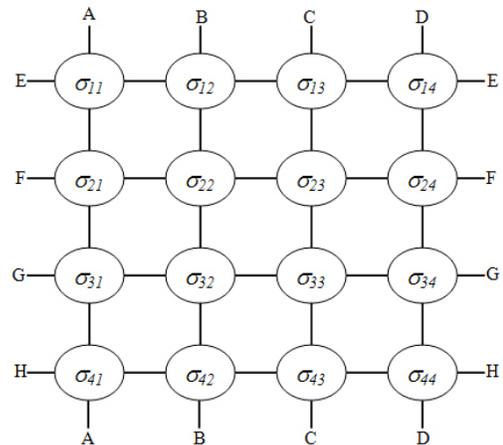


Fig. 1. A structure showing the interaction between spins for a two dimensional Ising Spin Glass system with 4×4 spins

consider the spin glass system on a two dimensional lattice consist of $n = l \times l$ sites, where each spin variable interacts only with its nearest neighbouring variables on a *toroidal* lattice (Figure. 1). The Hamiltonian specifying the energy for this system can be written as

$$H(\sigma) = - \sum_{i=1}^l \sum_{j=1}^l (J_{ij,(i+1)j} \sigma_{ij} \sigma_{(i+1)j} + J_{ij,i(j+1)} \sigma_{ij} \sigma_{i(j+1)}) \quad (2)$$

where, $i + 1 = 1$ if $i = l$ and $j + 1 = 1$ if $j = l$.

Here, each $J_{ij,i'j'}$ is the coupling constant in two dimensional lattice relating to spin σ_{ij} and $\sigma_{i'j'}$. For convenience, we reformulate this as a maximization problem and so seek to maximise

$$-H(\sigma) = \sum_{i=1}^l \sum_{j=1}^l (J_{ij,(i+1)j} \sigma_{ij} \sigma_{(i+1)j} + J_{ij,i(j+1)} \sigma_{ij} \sigma_{i(j+1)}) \quad (3)$$

In the context of EDAs, spin glass systems on a two dimensional lattice have been of particular interest to researchers. In particular, [18], [17], [20] showed that hierarchical Bayesian Optimization Algorithm (hBOA) could efficiently solve these problems outperforming other algorithms. [21] used the Ising spin glass problem as a test problem for the two algorithms Markov Network Estimation of Distribution Algorithm (MN-EDA) and Markov Network Factorised Distribution Algorithm (MN-FDA) and showed that their performance is better than that of other EDAs based on Bayesian networks. Also [13] stated that, although the two dimensional Ising spin glass problem is in the class of *Additively Decomposable Functions* (ADF), it cannot be efficiently represented as a *Junction tree* [9]. This is because, the junction tree based EDA has a *triangular structure* of dependency and therefore requires interaction between variables of order at least 3. However, the two dimensional Ising spin glass problem has a bivariate structure and therefore has a maximum clique order of 2. [22] argues that the Kikuchi approximation approach to estimate the distribution used by MN-EDA can represent the bivariate dependency as an exact factorisation, and therefore has an advantage over junction tree based EDAs. This argument applies to the DEUM algorithms as well, as they also can represent the exact factorization from the structure in the form of potential functions. This is described in detail in next section.

III. MARKOV RANDOM FIELD APPROACH TO MODELLING THE ISING SPIN GLASS PROBLEM

Markov Random Fields (MRF) is a branch of probabilistic theory. Also known as Markov Network or Undirected Graphical Models, MRF belongs to the general class of *Probabilistic Graphical Models* (PGM) [16], [27], [8], [5]. Another form of PGM is Bayesian Networks (also known as Directed Graphical Models), which has been widely exploited to model the distribution in EDAs [2], [7], [19], [12].

MRF regards a solution $x = \{x_1, x_2, \dots, x_n\}$ as a set of values taken by the set of random variables $X = \{X_1, X_2, \dots, X_n\}$. Given an undirected graph, G , where each node represents a variable $X_i \in X$ and each edge represents an interaction between variables in X , the joint probability distribution (jpd), $p(X = x)$ (or simply $p(x)$), can be expressed as a Gibbs distribution

$$p(x) = \frac{e^{-U(x)/T}}{\sum_y e^{-U(y)/T}} \quad (4)$$

using the Hammersley-Clifford theorem [4]. Here $U(x)$ (or more precisely $U(X = x)$) is an energy function defined as

$$U(x) = \sum_i^m \psi_i(c_i) \quad (5)$$

where each $\psi_i(c_i)$ is a function defined over a clique¹ c_i in G known as clique potential function. m is the total number of cliques in G . T is a temperature coefficient.

In [1], MRF theory was used to provide a formulation of the jpd, $p(x)$, that relates solution fitness, $f(X = x)$ (or simply $f(x)$), to the energy $U(x)$. To be precise:

$$p(x) = \frac{f(x)}{\sum_y f(y)} \equiv \frac{e^{-U(x)/T}}{\sum_y e^{-U(y)/T}} \quad (6)$$

from which an equation for each solution, x , can be derived

$$-\ln(f(x)) = U(x) \quad (7)$$

For simplicity, temperature, T , from (6) is set to 1.

$U(x)$ gives a full specification of the jpd (4), so it can be regarded as a probabilistic model of the fitness function. In particular, minimising $U(x)$ is equivalent to maximising $f(x)$. We refer to (7) as the **MRF Fitness Model** (MFM).

In [25], a *Univariate MFM* was used that assumes no interactions between variables in X . The graph G for such X is an edge-less graph, where there are n singleton cliques $C_i = \{X_i\}$. For each C_i the clique potential function was defined as $\alpha_i x_i$, where the α_i are parameters associated with X_i known as the *MRF parameters*. Therefore, $U(x)$ in terms of clique potential functions was

$$U(x) = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n \quad (8)$$

Being the unknown part of $U(x)$, the MRF parameters, α_i , completely determine the probability distribution.

In this paper, we define $U(x)$ for the two dimensional Ising spin glass problem (figure 1). Here, each spin variable, $\sigma_{ij} \in \sigma$, can be seen as a random variable, X_{ij} , in a set, X . Therefore, each solution $X = x$ can be seen as the string representation of the matrix

$$x = \begin{Bmatrix} x_{11}, x_{12}, \dots, x_{1l}, \\ x_{21}, x_{22}, \dots, x_{2l}, \\ \vdots \\ x_{l1}, x_{l2}, \dots, x_{ll} \end{Bmatrix} \quad (9)$$

¹A clique is a set of mutually neighbouring nodes in graph G

Here, the total number of variables in X is $n = l^2$. For such x , the fitness function to be maximised is

$$f(x) = \sum_{i=1}^l \sum_{j=1}^l (J_{ij,(i+1)j} x_{ij} x_{(i+1)j} + J_{ij,i(j+1)} x_{ij} x_{i(j+1)}) \quad (10)$$

Each variable $X_{ij} \in X$ interacts with four of its immediate neighbours. Figure (1) can be seen as an undirected graphical structure, G , for X . There are total of $2n$ order 2 cliques in G . For each clique $\{X_{ij}, X_{i'j'}\}$, we assign a potential function $\beta_{ij,i'j'} x_{ij} x_{i'j'}$ and therefore the energy, $U(x)$ in MFM (7) for such X will be

$$U(x) = \sum_{i=1}^l \sum_{j=1}^l (\beta_{ij,(i+1)j} x_{ij} x_{(i+1)j} + \beta_{ij,i(j+1)} x_{ij} x_{i(j+1)}) \quad (11)$$

Here, each $\beta_{ij,i'j'}$ is the MRF parameter associated with bivariate clique $\{X_{ij}, X_{i'j'}\}$. It is important to distinguish between $\beta_{ij,i'j'}$ in $U(x)$ with $J_{ij,i'j'}$ in $f(x)$. $\beta_{ij,i'j'}$ is a real valued parameter of the model and will be estimated from a set of solutions. This contrasts with the coupling constants $J_{ij,i'j'} \in \{-1, 1\}$. We use β to denote the set of all $2n$ bivariate MRF parameters $\beta_{ij,i'j'}$.

Depending upon the number and order of cliques considered, we may construct different MFMs from a single graph G . For example, in addition to potential functions $\beta_{ij,i'j'} x_{ij} x_{i'j'}$ for order 2 cliques $\{X_{ij}, X_{i'j'}\}$, we can also assign a potential function, $\alpha_{ij} x_{ij}$, to each singleton clique $\{X_{ij}\}$. The energy for the resulting MFM can be written as

$$U(x) = \sum_{i=1}^l \sum_{j=1}^l (\alpha_{ij} x_{ij} + \beta_{ij,(i+1)j} x_{ij} x_{(i+1)j} + \beta_{ij,i(j+1)} x_{ij} x_{i(j+1)}) \quad (12)$$

We use α to denote the set of all n univariate parameters α_{ij} .

We refer to (11) as the *minimal MFM* and (12) as the *complete MFM* for the two dimensional Ising spin glass problem. We use θ to denote the full set of parameters for either MFM.

IV. USING MFM TO LEARN THE MRF PARAMETERS

The basic idea here is to use a set of solutions D to approximate the parameters, θ , of the MRF. Let us explain this in more detail.

Each solution in a given population provides an equation satisfying the MFM (7). Selecting a set of solutions D consisting of N promising solutions from a population P therefore allows us to estimate the distribution by solving the system of equations:

$$F = A\theta^T \quad (13)$$

Here, F is the vector containing $-\ln(f(x))$ of all solutions in D , θ , the unknown part of the equation, is the vector of

all MRF parameters and A is the matrix of solution values in D .

For the minimal MFM, F will be an N dimensional vector containing $-\ln(f(x))$ of the solutions in D , θ will be a $2n$ dimensional vector of all MRF parameters β and A will be an $N \times 2n$ dimensional matrix, where each element a_{rs} of A is the product of the alleles from r^{th} solution associated with s^{th} parameter of the model. For the complete MFM, $\theta = \{\alpha, \beta\}$ will be a vector of $3n$ MRF parameters, as there will be $2n$ parameters in set β and n parameters in set α , and A will be an $N \times 3n$ dimensional matrix accordingly.

Depending on the relationship between N and the length of θ , the system will be under-, over-, or precisely-specified. A standard fitting algorithm can be used to give a maximum likelihood estimation of the θ . For mathematical reasons, $\{-1, 1\}$ are used as the values of x_i in $U(x)$, rather than $\{0, 1\}$. This is the standard practice in MRF modelling [10].

V. USING A METROPOLIS METHOD TO SAMPLE MRF

So far we have shown how to construct a MFM for the Ising spin glass problem and use it to approximate the MRF parameters. Once we get the parameters of the model, the jpd, $p(x)$, is completely specified. Therefore, the next step is to sample $p(x)$. In this section we develop a *zero temperature Metropolis method* for this purpose.

A. Zero Temperature Metropolis method

Metropolis methods are a class of *Markov Chain Monte Carlo* (MCMC) algorithms [11] that have been widely used to sample from a probability distribution. It tries to minimise the energy of the Gibbs distribution. In our case, it results in maximisation of fitness (7). Here we present a variant which we call *Bitwise Zero-Temperature Metropolis method* (BZTM). Given a set of MRF parameters, θ , calculated from a set of solutions D , it is then possible to sample a new solution, $x^o = \{x_1^o, x_2^o, \dots, x_n^o\}$ using the BZTM as shown in (Figure 2). For the complete MFM presented in (12), ΔU

Bitwise Zero-Temperature Metropolis method (BZTM)

- 1) Generate a solution $x^o = \{x_1^o, x_2^o, \dots, x_n^o\}$ at random.
 - 2) Repeat:
 - a) Set $x^{tmp} = x^o$.
 - b) For $i = 1$ to n
 - i) Mutate variable x_i^o to obtain the mutated solution $x^{o'}$.
 - ii) Set $\Delta U = U(x^{o'}) - U(x^o)$.
 - iii) if $\Delta U < 0$ set $x^o = x^{o'}$.
 - :Until $x^{tmp} = x^o$.
 - 3) Terminate with answer x^o .
-

Fig. 2. The pseudo-code of the Bitwise Zero-Temperature Metropolis method

can be determined explicitly from the following formula:

$$\Delta U = \left(x_{ij}^{o'} - x_{ij}^o \right) \left(\alpha_i + \beta_{(i-1)j,ij} x_{(i-1)j}^o + \beta_{ij,(i+1)j} x_{(i+1)j}^o + \beta_{i(j-1),ij} x_{i(j-1)}^o + \beta_{i(j+1),ij} x_{i(j+1)}^o \right) \quad (14)$$

Similarly, for the minimum MFM presented in (11), ΔU can be determined explicitly from the following formula:

$$\Delta U = \left(x_{ij}^{o'} - x_{ij}^o \right) \left(\beta_{(i-1)j,ij} x_{(i-1)j}^o + \beta_{ij,(i+1)j} x_{(i+1)j}^o + \beta_{i(j-1),ij} x_{i(j-1)}^o + \beta_{i(j+1),ij} x_{i(j+1)}^o \right) \quad (15)$$

This significantly reduces the cost of calculating ΔU .

B. DEUM with the Metropolis method

Now that we know how to sample the MRF parameters, we can formulate DEUM for the Ising spin glass problem (Is-DEUM). As the Is-DEUM described here implements the Metropolis methods as the sampling technique, we denote it as Is-DEUM_m. (Figure 3) shows the workflow of Is-DEUM_m. Notice that the Is-DEUM_m only has a single

Is-DEUM with Metropolis sampler (Is-DEUM_m)

- 1) Generate a population, P , of size M at random.
 - 2) Select a set D consisting of N fittest solutions from P , where $N \leq M$.
 - 3) Calculate the MRF parameters θ by fitting MFM to D .
 - 4) Repeat:
 - Sample $x^o = \{x_1^o, x_2^o, \dots, x_n^o\}$ using BZTM
 - :Until R iteration completes Or $f(x^o)$ is optimal/good enough
 - 5) Terminate with answer x^o .
-

Fig. 3. The pseudo-code of the DEUM for Ising model with Metropolis sampling method

generation. Also in step 4 of the algorithm, we repeatedly use BZTM to sample different x^o . We found that by repeatedly sampling the x^o with different random start, the optimum solution was found in first generation (as we shall show in the next section)². This therefore eliminates the necessity of creating a child population³.

C. Experiments and Results

Experiments were performed with three different sizes of two dimensional Ising Spin Glass system. They were 4×4 ($n = 16$), 6×6 ($n = 36$) and 8×8 ($n = 64$). Four random instances of each problem size were used for the experiment.

²This has also been illustrated in [26] for the Onemax problem, where a Zero-Temperature Metropolis algorithm was able to find the solution in single generation

³Though it is straightforward to form a child population once we know how to sample from the MRFs (see [26] for an example)

Each instance was generated by randomly sampling the coupling constant $J_{ij} \in \{+1, -1\}$. The optimum solution for each instance was verified by using the Spin Glass Ground server, provided by the group of Prof. Michael Juenger⁴. The parameters for Is-DEUM_m were chosen empirically.

These experiments are divided into two parts:

- 1) A performance comparison with other EDAs
- 2) A performance comparison between complete and minimal MFM

1) *Experiments on the performance comparison with other EDAs:* The aim of this experiment is to compare the performance of Is-DEUM with that of other EDAs. Mainly, the comparison is made with the results presented in [22], where the performance of five different EDAs, both using MRF and Bayesian networks, have been presented for similar instances of Ising spin glass problem. Namely, they were MN-EDA (using Kikuchi approximation approach), MN-EDA^f (using Kikuchi approximation with fixed structure as Is-DEUM), MN-FDA (using junction graph approach), EBNA_{k2} (using Bayesian network with $k2$ metrics) and MT-FDA (using a mixture of tree model).

We ran 100 independent runs of Is-DEUM_m for each of the 12 instances and recorded the number of fitness evaluations needed to find the optimum. The minimal MFM (11) was used in Is-DEUM_m and the whole population was selected for estimation of MRF parameters, i.e. we take $D = P$. Therefore, the selection size N was equal to the population size M . To determine M , we started with the minimum number of M needed to make the system of linear equation specified (in case of (11) the minimum M is $2n$). Then we gradually increased it, until a success rate of over 95% was achieved (in other words, until more than 95 out of 100 runs found the optimum). The resulting M was taken as the population size for that particular instance. The maximum number of allowed repetitions, R , for BZTM was set to 3000. Is-DEUM_m was terminated if the optimum was found or R repetitions of BZTM were done. As, at the end of each BZTM, the fitness evaluation was done in order to calculate $f(x^o)$, the number of fitness evaluations was calculated as the sum of population size and the total repetitions of the BZTM needed before finding the optimum.

Table (I) shows the experimental results on the performance of Is-DEUM_m on all 12 instances of the Ising problems. The first column shows the problem instances (PI). The second and third column shows the average number of fitness evaluation (FE) and the corresponding standard deviation (SD) over the 100 runs, the fourth column shows the success rate (SR) and the fifth column shows the population size (PS) used for the corresponding instances.

The performance of Is-DEUM_m was significantly better than that of other EDAs presented in [22], both in terms of success rate and the number of fitness evaluation needed to find the optimum. In particular, the best EDA reported in [22] was MN-FDA^f with average fitness evaluation and

⁴<http://www.informatik.uni-koeln.de/ls.juenger/research/sgs/sgs.html>

TABLE I
PERFORMANCE OF IS-DEUM_m WITH MINIMAL MFM FOR 12
INSTANCES OF ISING SPIN GLASS PROBLEM

PI	FE	SD	SR	PS
I-16-1	41.53	1.14	100	40
I-16-2	60.44	13.96	100	50
I-16-3	52.57	1.96	100	50
I-16-4	41.66	1.38	100	40
I-32-1	126.19	48.87	100	90
I-32-2	107.45	21.07	100	90
I-32-3	98.59	9.98	100	90
I-32-4	115.66	28.82	100	90
I-64-1	231.66	35.28	100	200
I-64-2	361.87	170.24	100	200
I-64-3	362.6	177.75	100	200
I-64-4	275.66	92.15	100	200

success rate of 220.17 and 98.5% respectively for $n = 16$, 1586.02 and 95.25% respectively for $n = 36$ and, 6110.8 and 95% respectively for $n = 64$. Whereas, for Is-DEUM, they were 49.05 and 100% respectively for $n = 16$, 111.96 and 100% respectively for $n = 36$ and 296.69 and 100% respectively for $n = 64$. This is a significant improvement in the performance.

2) *Experiment on the performance comparison between complete and minimal MFM*: The aim of this experiment is to show that, for the Ising spin glass problem, the use of minimal MFM instead of complete MFM does not decrease the quality of the solution, but does reduce the computational cost needed to find the solution.

Table (II) shows the experimental results on the performance of Is-DEUM_m using the complete MFM (12) on all 12 instances of the Ising spin glass problems. The experimental

TABLE II
PERFORMANCE OF IS-DEUM_M WITH COMPLETE MFM FOR 12
INSTANCES OF ISING SPIN GLASS PROBLEM

PI	FE	SD	SR	PS	tr
I-16-1	61.45	1.27	100	60	1.11
I-16-2	67.43	9.20	100	60	1.02
I-16-3	62.64	3.58	100	60	1.02
I-16-4	61.44	1.38	100	60	1.07
I-32-1	161.52	47.22	100	130	1.33
I-32-2	146.52	17.06	100	130	1.27
I-32-3	137.61	8.22	100	130	1.29
I-32-4	148.64	19.79	100	130	1.34
I-64-1	284.80	49.06	100	250	2.10
I-64-2	376.45	132.26	99	250	1.76
I-64-3	334.99	96.76	100	250	1.55
I-64-4	319.86	115.15	100	250	1.87

setups were similar to that of Is-DEUM_m using the minimal MFM (11) described in previous sub-section. The minimum number of population size (PS) needed to make the system of linear equation specified for (12) was $M = 3n$, as compared to $M = 2n$ for (11). Therefore as we can see from Table (II), the optimum population size needed for all 12 instance for Is-DEUM_m with the complete MFM was greater than that needed by Is-DEUM_m with the minimal MFM (shown in Table (I)). As a result, the number of fitness evaluations

for Is-DEUM_m with the complete MFM was greater than that of Is-DEUM_m with the minimal MFM. Also in sixth column of Table(II), the ratio of extra time (t-ratio) needed by complete MFM in comparison to minimal MFM is shown for each instance. For each instance, the t-ratio is equal to the average time taken by Is-DEUM_m with complete MFM divided by the average time taken by minimal MFM. For $n = 16$ the difference in time is fairly small, however as n grows the t-ratio grows and for $n = 64$ the time taken by complete MFM is almost double to the time taken by minimal MFM. This result is expected, as the computational time to calculate the MRF parameters grows polynomially with the size of the matrix A in the system of linear equations (13). Matrix A grows as selection size $N = M$ grows and N grows as the number of MRF parameters in the MFM grows.

This result shows that, without losing the quality of the solutions, using the minimal MFM instead of the complete MFM, results in reduced computational cost. We will use the minimal MFM for the rest of the experiments presented in this paper.

VI. USING A GIBBS SAMPLER TO SAMPLE MRF

So far we have shown that by sampling the MRF using a Metropolis method, Is-DEUM was able to solve Ising spin glass systems of size $n = 16$, $n = 32$ and $n = 64$. However, for problem sizes of $n = 100$ and higher, Is-DEUM with the Metropolis method was not able to find the optimum solution. In this section we describe another sampling method known as *Gibbs Sampler* (GS) and incorporate it in Is-DEUM. The aim here is to solve Ising spin glass problems of larger size. We also present experimental results on the performance of this version of Is-DEUM.

A. Gibbs sampler

As with the Metropolis method, the Gibbs sampler (GS) [3] is a class of MCMC algorithm that has been widely used to sample probability distributions. In order to explain GS, we first need to define the formulation of marginal probability $p(x_{ij})$ for each variable, x_{ij} , from the jpd $p(x)$.

We use x^+ to denote x having a particular $x_{ij} = +1$, similarly, we use x^- to denote x having $x_{ij} = -1$. We denote the probability that the variable in position ij is equal to 1 by $p(x_{ij} = 1)$. Clearly, $p(x_{ij} = -1) = 1 - p(x_{ij} = 1)$. We can now write

$$p(x_{ij} = 1) = \frac{p(x^+)}{p(x^+) + p(x^-)} \quad (16)$$

Substituting $p(x)$ from (4) and cancelling the Z , we get

$$p(x_{ij} = 1) = \frac{e^{-U(x^+)/T}}{e^{-U(x^+)/T} + e^{-U(x^-)/T}} \quad (17)$$

or,

$$p(x_{ij} = 1) = \frac{1}{1 + e^{(U(x^+) - U(x^-))/T}} \quad (18)$$

As $U(x^+)$ and $U(x^-)$ agree in all terms other than those containing x_{ij} , the common terms in both $U(x^+)$ and $U(x^-)$

drop out and we get the following expression as the estimate of the marginal probability for $x_{ij} = 1$.

$$p(x_{ij} = 1) = \frac{1}{1 + e^{2W_{ij}/T}} \quad (19)$$

where, W_{ij} for (11) is

$$W_{ij} = \beta_{ij,(i+1)j}x_{(i+1)j} + \beta_{ij,i(j+1)}x_{i(j+1)} + \beta_{(i-1)j,ij}x_{(i-1)j} + \beta_{i(j-1),ij}x_{i(j-1)} \quad (20)$$

Note that, as $T \rightarrow 0$, the value of $p(x_{ij} = 1)$ tends to a limit depending on the W_{ij} . If $W_{ij} > 0$, then $p(x_{ij} = 1) \rightarrow 1$ as $T \rightarrow 0$. Conversely, if $W_{ij} < 0$, then $p(x_{ij} = 1) \rightarrow 0$ as $T \rightarrow 0$. If $W_{ij} = 0$, then $p(x_{ij} = 1) = 0.5$ regardless of the value of T . Therefore, the W_{ij} are indicators of whether the x_{ij} at the position ij should be 1 or -1 . This indication becomes stronger as the temperature is cooled towards zero.

Now let us describe a variant of GS, which we call the *Bitwise Gibbs Sampler* (BGS). Pseudo code for BGS is shown in (Figure 4). It starts by randomly generating a solution, then calculates $p(x_{ij})$ for a chosen x_{ij} and replaces it by sampling $p(x_{ij})$. This continues until a termination criterion is satisfied. The temperature coefficient, T , in GS can be used to control the convergence of $p(x_{ij})$. Here, we start with high temperature, T , then at each iteration, gradually decrease it using a cooling schedule so as to gradually converge $p(x_{ij})$ to its limit. The DEUM_d algorithm described in [23], [24] also uses temperature to control the convergence of the marginals.

Bitwise Gibbs Sampler (BGS)

- 1) Generate a solution $x^o = \{x_1^o, x_2^o, \dots, x_n^o\}$ at random.
 - 2) set $r = 0$ and also set the initial value for T .
 - 3) Repeat:
 - a) Set $x^{tmp} = x^o$.
 - b) For $i = 1$ to n
 - i) Increase r by 1
 - ii) Decrease T
 - iii) Set $x_i^o = 1$ with probability $p(x_i^o = 1)$
 :Until $x^{tmp} = x^o$.
 - 4) Terminate with answer x^o .
-

Fig. 4. The pseudo-code of the Bitwise Gibbs Sampler

B. DEUM with Gibbs Sampler

Now that we know how to sample the MFM using a Gibbs Sampler, we can incorporate it in DEUM. Figure (5) shows the workflow of Is-DEUM with a Gibbs Sampler (Is-DEUM_g). Notice that, as with Is-DEUM_m (shown in Figure (3)), Is-DEUM_g only has a single generation. As shown in [26], we could easily incorporate a multiple generation scheme in Is-DEUM_g. We found that, for the Ising spin

Is-DEUM with Gibbs Sampler (Is-DEUM_g)

- 1) Generate a population, P , of size M
 - 2) Select the set D consisting of N fittest solutions from P , where $N \leq M$.
 - 3) Calculate the MRF parameters θ by fitting MFM to D .
 - 4) Repeat:
 - Generate $x^o = \{x_1^o, x_2^o, \dots, x_n^o\}$ using BGS
 - :Until R iteration completes Or $f(x^o)$ is optimal/good enough
 - 5) Terminate with answer x^o .
-

Fig. 5. The pseudo-code of the DEUM for Ising model with Gibbs Sampler

glass problem, by repeatedly sampling the x^o with different random starts, Is-DEUM_g was consistently able to find the optimum solution in the first generation.

C. Experiments and Results

Experiments were conducted with three different sizes of Ising Spin Glass problem: 10×10 ($n = 100$), 16×16 ($n = 256$) and 20×20 ($n = 400$). Four random instances of each problem size were used for the experiment. Each instance was generated by randomly sampling the coupling constant $J_{ij} \in \{+1, -1\}$. The optimum solution for each instance was verified by using Spin Glass Ground server, provided by the group of Prof. Michael Juenger⁵. The parameters for each algorithm were chosen empirically.

We divide our experiments into two parts:

- 1) A performance comparison with other EDAs
- 2) A performance comparison with Repeated Bitwise Gibbs Sampler (RBGS)

1) *Experiment on the performance comparison with other EDA*: We made 30 independent runs of Is-DEUM_g for each of the 12 instances of the Ising spin glass problem and recorded the number of fitness evaluations needed to find the optimum. The minimal MFM (11) was used to estimate the energy of the Gibbs distribution. The population size and selection size for Is-DEUM_g were 100 and 250 respectively for $n=100$, 3000 and 700 respectively for $n=256$ and 8000 and 1000 respectively for $n=400$. The temperature T for the BGS was set to $T = 1/0.0005r$, where r is the current number of x_i^o samplings done in BGS (see Figure (4)). As r increases, T decreases and the solution x^o will converge to a particular value for each x_i^o . The maximum number of allowed repetitions, R , for BGS was set to 500. Is-DEUM_g was terminated if the optimum was found or R repetitions of BGS were done. As, at the end of each BGS, a fitness evaluation was done in order to calculate $f(x^o)$, the number of fitness evaluations was calculated as the sum of population size and the total repetitions of the BGS needed before finding the optimum.

⁵<http://www.informatik.uni-koeln.de/ls.juenger/research/sgs/sgs.html>

Table (III) shows the experimental results on the performance of Is-DEUM_g on all 12 instances of the Ising spin glass problems. The first column shows the problem

TABLE III
PERFORMANCE OF IS-DEUM_g ON ALL 12 INSTANCES OF ISING
PROBLEM

PI	FE	SD	FE-PS	IT	SR
I-100-1	1008.90	7.96	8.90	639×10^3	100
I-100-2	1002.73	1.82	2.73	149×10^3	100
I-100-3	1010.97	10.71	10.97	723×10^3	100
I-100-4	1003.20	2.32	3.20	156×10^3	100
I-256-1	3015.27	17.45	15.27	115×10^5	100
I-256-2	3003.23	2.96	3.23	213×10^4	100
I-256-3	3054.28	66.17	54.28	351×10^5	97
I-256-4	3007.50	6.05	7.50	639×10^4	100
I-400-1	8093.62	113.80	93.62	152×10^6	97
I-400-2	8036.47	34.87	36.47	632×10^5	100
I-400-3	8058.77	54.14	58.77	103×10^6	100
I-400-4	8047.07	42.25	47.07	755×10^5	100

instances (PI). The second and third column shows the average number of fitness evaluation (FE) and the corresponding standard deviation (SD) over the 100 runs. The fourth column shows the average number of fitness evaluation without counting the number of evaluation needed to evaluate the population (FE-PS). The fifth column shows the average *internal calculation* (IT) of Is-DEUM_g before finding the solution. Internal calculation is the total number of marginal probability calculations done over all repetitions of BGS. The sixth column shows the success rate (SR) of finding the optimum over 100 runs.

Previously, BOA and hBOA have been applied to Ising problem instances of size $n = 100$, $n = 256$ and $n = 400$ [18], [17], [20]. Their performances were reported to be comparable and sometime better than state of the art algorithms for solving Ising spin glass problems.

Our results show that the number of fitness evaluations needed to find the solution for Is-DEUM_g was significantly less than that reported for BOA and hBOA. For example for $n = 400$, the average fitness evaluation for hBOA (with a hill climber) was about 10^5 (BOA was not able to find the solution for $n = 400$). Whereas, for Is-DEUM_g this was only about 8000.

However, for larger instances of Ising spin glass problem, such as $n = 400$, the computational time for sampling x^o using BGS gets higher and therefore the repeated sampling of x^o dominates the computational time taken by the rest of the process in Is-DEUM_g. It even dominates the time taken to solve the system of equations. Therefore, the performance of Is-DEUM_g for Ising spin glass problem should be evaluated in terms of the number of internal calculations done by the BGS, rather than by the number of fitness evaluations. As BOA and hBOA do not have such an internal calculation process, it is difficult to compare them with Is-DEUM_g.

2) *Experiment on the performance comparison with Repeated Bitwise Gibbs Sampler (RBGS)*: In Is-DEUM_g, we have used GS to sample the MRF which we estimate from the

population of solution. Here we show how we can directly apply GS to the fitness function and sample x^o .

Given the fitness function (10), the marginal probability $p(x_{ij} = 1)$ for any $x_{ij} = 1$ can also be estimated directly as

$$p(x_{ij} = 1) = \frac{1}{1 + e^{2\Gamma_{ij}/T}} \quad (21)$$

Where,

$$\Gamma_{ij} = J_{(i-1)j,ij}x_{(i-1)j} + J_{ij,(i+1)j}x_{(i+1)j} + J_{i(j-1),ij}x_{i(j-1)} + J_{i(j+1),ij}x_{i(j+1)} \quad (22)$$

We can use (21) in BGS (Figure (4)) as formulation of

Repeated Bitwise Gibbs Sampler algorithm (RBGS)

- 1) Repeat:
 - Sample a solution $x^o = \{x_1^o, x_2^o, \dots, x_n^o\}$ using BGS
 - :Until R iteration completes Or $f(x^o)$ is optimal/good enough
 - 2) Terminate with answer x^o .
-

Fig. 6. The pseudo-code of the Repeated Bitwise Gibbs Sampler algorithm

$p(x_{ij})$ and sample x^o . As the performance of BGS heavily depends on the initial solution, we repeatedly ran the BGS with different random starts. Figure (6) presents the workflow of the *repeated BGS algorithm* (RBGS). The temperature T

TABLE IV
PERFORMANCE OF IS-DEUM AND RBGS FOR ISING SPIN GLASS
PROBLEM OF SIZE $n = 100$, $n = 256$, AND $n = 400$

PI	Is-DEUM _g			RBGS		
	FE-PS	IT	SR	FE	IT	SR
I-100	6.45	417×10^3	100	6.53	330×10^3	100
I-256	20.07	136×10^5	99	37.19	284×10^5	99
I-400	58.98	984×10^5	99	94.67	114×10^6	87

for the BGS was set to $T = 1/0.005r$, where r is the current number of x_i^o samplings done in BGS (see Figure (4)). Note that, the constant associated with r here is 0.005 which is greater than that used in Is-DEUM_g. This was simply because, with 0.005 the performance of RBGS was better than that with 0.0005. The maximum number of allowed repetitions R was set to 500.

As at the end of each repetition of BGS, a fitness evaluation (FE) was done to check the quality of sampled x^o , we record the total number of BGS repetitions as the number of fitness evaluations for RBGS. However, In contrast to RBGS, Is-DEUM_g had to evaluate a population of solutions, which, although taking a negligible amount of time in comparison to the time taken for a sampling using BGS, hugely contributed to the number of fitness evaluations for Is-DEUM_g. Therefore, for Is-DEUM_g, we use the fitness evaluation without counting the evaluation of the population

(FE-PS) to compare with FE of RBGS. On the other hand, the internal calculation of BGS (IT), is the most dominant factor for the computational cost in both algorithms and therefore we also record the IT for RBGS and compare it with that of Is-DEUM_g.

Table (IV), shows the average of fitness evaluations (FE) (FE-PS for Is-DEUM_g), internal calculation (IT) and success rate (SR) for three different sizes of Ising spin glass problem. The result for Is-DEUM_g is the average taken from Table (III) for each problem size.

Our results show that in terms of both fitness evaluations and internal calculation taken to find the optimum, the performance of Is-DEUM_g was better than that of RBGS. Also note that the success rate for Is-DEUM_g was 99% in comparison to 87% for RBGS for $n = 400$. These results show that sampling from the MRF estimated from the population instead of the actual fitness function results in better performance of the algorithm.

Let us explain the above results. The fitness landscape of the Ising spin glass problem contains large number of plateaus. This is because the values of coupling constants, J , relating two spin variables are restricted to -1 and +1. As RBGS uses J to estimate $p(x_{ij})$, it takes time to overcome all the plateau. However, Is-DEUM_g uses real-valued MRF parameters β to estimate $p(x_{ij})$. β therefore alters the fitness landscape by introducing some variation to the plateau. The result is more efficient searching of the fitness landscape by the sampling algorithm.

VII. CONCLUSION

Is-DEUM differs from other EDAs in two ways, 1) Use of a model of the fitness function (MFM) to estimate the model parameters by fitting it to a set of solution D and 2) Use of MCMC algorithms to sample the MRF. Our experimental results suggest estimating and sampling the distribution in this way may significantly improve the performance of an EDA.

The Is-DEUM presented here (and also other DEUM algorithms developed so far) assumes the structure of the MRF to be fixed. It will be interesting to extend DEUM to incorporate a structure learning algorithm. This will allow DEUM to be applied to a wider range of multivariate optimization problems.

REFERENCES

- [1] D. F. Brown, A. B. Garmendia-Doval, and J. A. W. McCall. Markov Random Field Modelling of Royal Road Genetic Algorithms. *Lecture Notes in Computer Science*, 2310:65–78, January 2002.
- [2] R. Etxeberria and P. Larrañaga. optimization with bayesian networks. In *Proceedings of the Second Symposium on Artificial Intelligence. Adaptive Systems. CIMA 99.*, page 332339, Cuba, 1999.
- [3] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. In M. A. Fischler and O. Firschein, editors, *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, pages 564–584. Kaufmann, Los Altos, CA., 1987.
- [4] J. M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. *Unpublished*, 1971.
- [5] M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.
- [6] R. Kindermann and J. L. Snell. *Markov Random Fields and Their Applications*. AMS, 1980.
- [7] P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.
- [8] S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.
- [9] 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 B*, 50:157–224, 1988.
- [10] S. Z. Li. *Markov Random Field modeling in computer vision*. Springer-Verlag, 1995.
- [11] N. Metropolis. Equations of state calculations by fast computational machine. *Journal of Chemical Physics*, 21:1087–1091, 1953.
- [12] H. Mühlenbein and T. Mahnig. FDA - A scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(4):353–376, 1999.
- [13] H. Mühlenbein, T. Mahnig, and A. R. Ochoa. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):215–247, 1999.
- [14] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions: I. binary parameters. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature – PPSN IV*, pages 178–187, Berlin, 1996. Springer.
- [15] I. Murray and Z. Ghahramani. Bayesian Learning in Undirected Graphical Models: Approximate MCMC algorithms. In *Twentieth Conference on Uncertainty in Artificial Intelligence (UAI 2004)*, Banff, Canada, 8-11 July 2004.
- [16] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufman Publishers, Palo Alto, CA, 1988.
- [17] M. Pelikan. *Bayesian optimization algorithm: From single level to hierarchy*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2002. Also IlliGAL Report No. 2002023.
- [18] M. Pelikan and D. E. Goldberg. Hierarchical BOA solves Ising spin glasses and MAXSAT. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, pages 1271–1282, 2003. Also IlliGAL Report No. 2003001.
- [19] M. Pelikan and H. Mühlenbein. The bivariate marginal distribution algorithm. In R. Roy, T. Furuhashi, and P. K. Chawdhry, editors, *Advances in Soft Computing - Engineering Design and Manufacturing*, pages 521–535, London, 1999. Springer-Verlag.
- [20] M. Pelikan, J. Ocenasek, S. Trebst, M. Troyer, and F. Alet. Computational complexity and simulation of rare events of ising spin glasses. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, pages 36–47, 2004.
- [21] R. Santana. *Probabilistic modeling based on undirected graphs in Estimation Distribution Algorithms*. PhD thesis, Institute of Cybernetics, Mathematics and Physics, Havana, Cuba, 2003.
- [22] R. Santana. Estimation of Distribution Algorithms with Kikuchi Approximation. *Evolutionary Computation*, 13:67–98, 2005.
- [23] S. Shakya, J. McCall, and D. Brown. Estimating the distribution in an EDA. In B. Ribeiro, R. F. Albrechet, A. Dobnikar, D. W. Pearson, and N. C. Steele, editors, *In proceedings of the International Conference on Adaptive and Natural computing Algorithms (ICANNGA 2005)*, pages 202–205, Coimbra, Portugal, 2005. Springer-Verlag, Wien.
- [24] S. Shakya, J. McCall, and D. Brown. Using a Markov Network Model in a Univariate EDA: An Empirical Cost-Benefit Analysis. In *proceedings of Genetic and Evolutionary Computation Conference (GECCO 2005)*, Washington, D.C., USA, 2005. ACM.
- [25] S. K. Shakya, J. A. W. McCall, and D. F. Brown. Updating the probability vector using MRF technique for a Univariate EDA. In E. Onaindia and S. Staab, editors, *Proceedings of the Second Starting AI Researchers' Symposium, volume 109 of Frontiers in artificial Intelligence and Applications*, pages 15–25, Valencia, Spain, August 2004. IOS press.
- [26] S. K. Shakya, J. A. W. McCall, and D. F. Brown. Incorporating a metropolis method in a distribution estimation using markov random field algorithm. In *proceedings of IEEE Congress on Evolutionary Computation (IEEE CEC 2005)*, volume 3, Edinburgh, UK, 2005. IEEE press.
- [27] J. Whittaker. *Graphical Models in applied multivariate Statistics*. John Wiley, 1990.