Optimisation by Estimation of Distribution with DEUM framework based on Markov Random Fields

Siddhartha Shakya and John McCall

***Abstract*—In this paper, we present a Markov Random Field (MRF) approach to estimating and sampling the probability distribution in populations of solutions. The approach is used to define a class of algorithms under the general heading Distribution Estimation Using Markov Random Fields (DEUM). DEUM is a subclass of Estimation of Distribution Algorithms (EDAs) where interaction between solution variables is represented as an undirected graph and the joint probability of a solution is factorised as a Gibbs distribution derived from the structure of the graph. The focus of this paper will be on describing the three main characteristics of DEUM framework, which distinguishes it from the traditional EDA. They are: 1) use of MRF models, 2) fitness modelling approach to estimating the parameter of the model and 3) Monte Carlo approach to sampling from the model.**

***Index Terms*—Estimation of Distribution Algorithms, Evolutionary algorithms, Fitness aodelling, Markov Random Fields, Gibbs distribution.**

I. Introduction

In nature, improved organisms are evolved by means of natural selection and random variation. Evolutionary Algorithms (EA) adopt this approach and simulate natural selection and variation to evolve a better solution to a problem. Each of the classical branches of EA, Genetic Algorithms (GA) [1], Evolution Strategies (ES) [2] and Evolutionary Programming (EP) [3], encapsulates selection and variation in some form. Selection and variation have well-understood roles in EA. Selection puts pressure on the evolution of high quality solutions by selecting fitter solutions from a population. Variation produces a set of successor solutions based on the selected solutions, exploiting knowledge gained so far while continuing to explore novel solutions.

In a GA, variation is achieved using two genetic operators, *crossover* and *mutation*. Crossover forms the new population by exchanging some parts between the selected solutions. Mutation slightly modifies some parts of the newly-formed solutions to introduce some genetic variation in the new population. In recent years, a probabilistic approach to variation has been proposed which replaces crossover and mutation with *distribution estimation* and *sampling*. Distribution estimation derives a probability distribution from a population of solutions. Sampling generates a new population with statistical properties determined by the distribution. Algorithms using this approach to variation are called Estimation of Distribution Algorithms (EDAs) [4], [5]. EDAs have been recognised as a powerful technique for optimisation, comparing well with classical GAs on a range of benchmark problems [6], [7], [8].

Much research in EDAs focuses on different approaches to distribution estimation and sampling and their relative effectiveness. In particular, directed graphical models (Bayesian networks) have been widely studied and are well-established as a useful approach in EDAs. In this paper, we introduce a framework of an EDA based on undirected graphical models (Markov Random Fields) called Distribution Estimation Using Markov Random Fields (DEUM) [1]. One of the distinct characteristics of this framework is that, it builds a model of the fitness function as opposed to a model of good solutions. This characteristics also distinguishes it from a recently proposed MRF based EDA called Markov Network EDA (MN-EDA) [11] that builds model based on *Kikuchi approximation* [12]. The focus of this paper will be on describing how the three distinct components of an EDA, 1) estimating the model, 2) estimating the parameters and 3) sampling from the model, are incorporated within the DEUM framework and how they interact together to perform optimisation within different instances of this framework.

The paper is structured as follows. Section 2 describes the general framework of DEUM algorithms. Section 3 describes Markov Random Fields (MRF), a class of Probabilistic Graphical Model (PGM) that DEUM uses as its model of distribution. It also explains the general motivation behind using PGM in EDAs and distinguishes Bayesian Networks with MRFs. Section 4 describes the parameter estimation technique used in DEUM. In particular, it describes how to build a model of fitness function and used it to estimate the MRF parameters. Section 5 presents the sampling technique used in DEUM. Section 6 describes two instances of DEUM, a univariate DEUM that directly samples from the Gibbs distribution and a bivariate DEUM that uses a Monte Carlo sampling technique to sample from the model. Finally, section 7 presents the summary of the work and concludes the paper.

Estimation of Distribution Algorithm

1) Generate initial (parent) population P
2) Select a set of solutions D from P
3) Estimate the probability distribution of solutions, $p(x)$, from D
4) Sample $p(x)$ to generate offspring, and replace parent
5) Go to step 2 until termination criteria are meet

Fig. 1. The workflow of the general Estimation of Distribution Algorithm

[1] DEUM has been initially used to denote a MRF based univariate EDA that maintained a probability vector for sampling [9]. This algorithm, being an instance of general DEUM framework, has been later named as DEUM$_{pv}$ [10]

II. DEUM: A GENERAL FRAMEWORK

An EDA regards a solution, $x = \{x_1, x_2, .., x_n\}$, as a set of values taken by a set of variables, $X = \{X_1, X_2, ..., X_n\}$. As shown in Figure 1, all EDAs begin by initialising a population of solutions, P. A set of promising solutions D is then selected from P, and is used to estimate a probabilistic model of X, $p(X = x)$ or simply $p(x)$. $p(x)$ is then sampled to generate the next population.

The general framework of DEUM is very similar to that of any EDAs, which is shown in (Figure 2).

Distribution Estimation using MRF (DEUM)

```
1) Generate parent population $P$
2) Select a set of solutions $D$ from $P$
3) Estimate an MRF from $D$, i.e, estimate the probability
   distribution $p(x)$ from $D$ assuming undirected dependency
   between variables in $x$. This involves:
     a) Estimating structure of the MRF
     b) Estimating Parameter of the MRF using fitness
        modelling approach
4) Sample MRF to generate new solutions
5) Go to step 2 until termination criteria are meet
```

Fig. 2. The pseudo-code of the Distribution Estimation Using MRF (DEUM) algorithm

There are, however, several noticeable characteristics specific to DEUM. They are

1) It uses MRF as its probabilistic models
2) It builds a model of fitness function and uses it to estimate the parameters of the MRF
3) It then samples from the MRF. The process of sampling from MRF is different than that of other typical EDAs

Next three sections describe each of these topics in more detail.

III. MARKOV RANDOM FIELDS AND DEUM

In order to motivate the use of MRF in DEUM, it is important to understand the notion of Probabilistic Graphical Models (PGM) in EDAs. In this section we first describe the PGM in context of EDAs and then distinguish Bayesian Networks with the MRF. We then describe several properties of MRF that are exploited by the DEUM framework.

A. Probabilistic Graphical Models

The performance of an EDA heavily depends on how successfully it estimates the joint probability distribution $p(x) = p(x_1, x_2, ..., x_n)$. In general, the computation of $p(x)$ for a bit string variable encoding, $x_i \in \{0, 1\}$, involves the computation of probabilities for all 2^n configurations of x. This is not computationally feasible in most problems of interest. However, in many cases, a good approximation to $p(x)$ can be obtained by factorising the distribution in terms of the marginal probabilities of combinations of variables X_i, thus reducing the costs of distribution estimation and sampling. The simplest factorisation of $p(x)$ is in terms of the marginal probabilities of individual variables (1).

$$p(x) = \prod_{i=1}^{n} p(x_i) \tag{1}$$

This model assumes that each $X_i \in X$ is independent and does not interact with other variables in X . As interaction between solution variables is introduced, the terms in the factorisation of $p(x)$ become complex, involving conditional probabilities between two or more variables [2]. This is where PGM comes into effect.

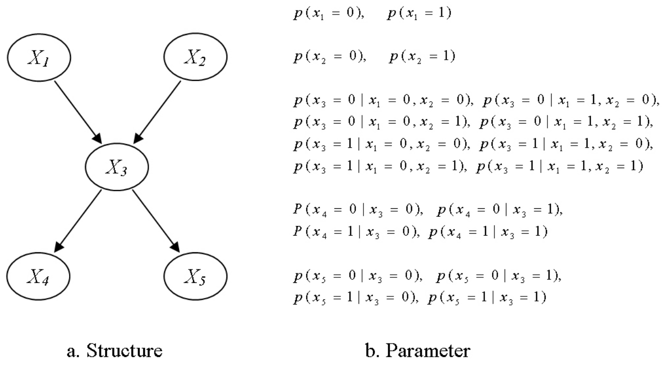
PGM provides an efficient and effective tool to represent the factorisation of the joint probability distribution (**jpd**), $p(x)$, and therefore have an important role in EDAs. They can be seen as a merger of two disciplines, probability theory and graph theory [14]. They are mainly categorised into two groups.

1) Directed models (Bayesian networks)
2) Undirected models (Markov Random Fields/Markov networks)

Let us give the formulation of jpd for each of them.

B. Bayesian networks

A Bayesian network can be regarded as a pair (B, Θ), where B is the structure of the model and the Θ is a set of parameters of the model. The structure B is a *Directed Acyclic Graph (DAG)* [3], where each node corresponds to a variable in the modelled data set and each edge corresponds to a conditional dependency. A set of nodes Π_i is said to be the parent of X_i if there are edges from each variable in Π_i pointing to X_i.


X_1
X_2
X_3
X_4
X_5
$p(x_1 = 0)$, $p(x_1 = 1)$
$p(x_2 = 0)$, $p(x_2 = 1)$
$p(x_3 = 0 \mid x_1 = 0, x_2 = 0)$, $p(x_3 = 0 \mid x_1 = 1, x_2 = 0)$,
$p(x_3 = 0 \mid x_1 = 0, x_2 = 1)$, $p(x_3 = 0 \mid x_1 = 1, x_2 = 1)$,
$p(x_3 = 1 \mid x_1 = 0, x_2 = 0)$, $p(x_3 = 1 \mid x_1 = 1, x_2 = 0)$,
$p(x_3 = 1 \mid x_1 = 0, x_2 = 1)$, $p(x_3 = 1 \mid x_1 = 1, x_2 = 1)$
$P(x_4 = 0 \mid x_3 = 0)$, $p(x_4 = 0 \mid x_3 = 1)$,
$P(x_4 = 1 \mid x_3 = 0)$, $p(x_4 = 1 \mid x_3 = 1)$
$p(x_5 = 0 \mid x_3 = 0)$, $p(x_5 = 0 \mid x_3 = 1)$,
$p(x_5 = 1 \mid x_3 = 0)$, $p(x_5 = 1 \mid x_3 = 1)$

a. Structure

b. Parameter

Fig. 3. A Bayesian network on 5 binary random variables

The parameter $\Theta = \{p(x_1|\Pi_1), p(x_2|\Pi_2), ..., p(x_n|\Pi_n)\}$ of the model is the set of conditional probabilities, where

[2] Depending on the level of complexity, factorisations are categorised into three groups: univariate, bivariate and multivariate factorisation. An excellent review of this can be found in [5], [13]

[3] A DAG is a graph where each edge joining two nodes is a *directed edge*, and also there is *no cycle* in the graph i.e. it is not possible to start from a node and travelling towards the correct direction return back to the starting node

each $p(x_i|\Pi_i)$ is the set of probabilities associated with a variable $X_i = x_i$ given the different configuration of it's parent variables Π_i.

Figure 3 shows the structure and the parameters of a Bayesian network, where each variable X_i is binary. i.e. $x_i \in \{0, 1\}$. For this structure the joint probability distribution can be factorised as:

$$p(x_1, x_2, x_3, x_4, x_5) =$$

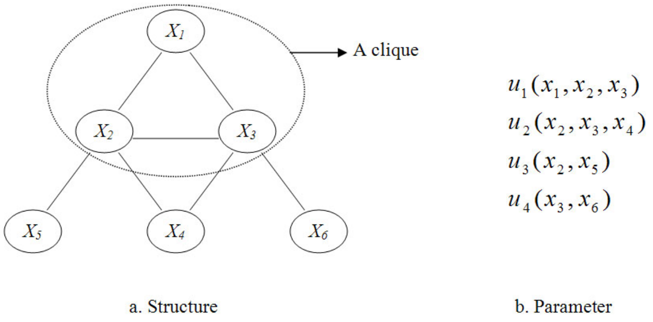
$$p(x_1)p(x_2)p(x_3|x_1, x_2)p(x_4|x_3)p(x_5|x_3) \qquad (2)$$

In general, given a set of variables $X = \{X_1, X_2, .., X_n\}$ a joint probability distribution for any Bayesian network is

$$p(x) = \prod_{i=1}^{n} p(x_i|\Pi_i) \qquad (3)$$

C. Markov Random Fields

A Markov Random Field is a pair (G, Ψ), where G is the structure and the Ψ is the parameter set of the network. G is an undirected graph where each node corresponds to a random variable in the modelled data set and each edge corresponds to conditional dependencies between variables. However, unlike Bayesian networks, the edges in Markov Random Fields are undirected. Here, the relationship between two nodes should be seen as a *neighbourhood relationship*, rather than a parenthood relationship. We use $N = \{N_1, N_2, ..., N_n\}$ to define a *neighbourhood system* on G, where each N_i is the set of nodes neighbouring to a node X_i [4]. The parameter set Ψ is a set of potential functions in terms of *cliques* in the structure G. These will be described next.



a. Structure

b. Parameter

Fig. 4. A Markov Random Field on 6 random variables

A MRF is characterised by its *local Markov property* known as *Markovianity* [15], [16] which states that a node X_i can be completely defined by knowing only its neighbouring nodes N_i. N_i is sometimes referred to as *Markov Blanket* for X_i [17]. In terms of probability it can be written as

$$p(x_i|x - \{x_i\}) = p(x_i|N_i) \qquad (4)$$

Local Markov property, however, does not provide the formulation for the joint probability distribution $p(x)$. Fortunately, Hammersley and Clifford [18] have provided a theorem that formulates the joint probability distribution for an MRF in terms of the *Gibbs distribution*. Let us explain this in detail.

Definition 3.1 (Clique): Given an undirected graph G, a clique is a fully connected subset of the nodes.

Definition 3.2 (Sub Clique): Given an undirected graph G, a sub clique of a clique is a fully connected subset of nodes within that clique.

Definition 3.3 (Maximal Clique): A clique is called *maximal*, if it is not a sub clique of any other clique.

Definition 3.4 (Singleton Clique): A clique is called *singleton* if it consist of a single node from G.

Gibbs distribution: A Gibbs distribution over a set of random variables X has the following form

$$p(x) = \frac{e^{-U(x)/T}}{Z} \qquad (5)$$

where,

$$Z = \sum_{y \in \Omega} e^{-U(y)/T} \qquad (6)$$

is a normalising constant, Ω is the set of all possible solutions, T is a parameter of the distribution known as the *temperature* and $U(x)$ (or more precisely $U(X = x)$) is known as the *energy* of the distribution.

Given an undirected graph, G, on X, energy, $U(x)$, is defined as a sum of *potential functions* over the cliques, C_i, in G.

$$U(x) = \sum_{i=1}^{m} u_i(c_i) \qquad (7)$$

Here, $u_i(c_i)$ (or more precisely $u_i(C_i = c_i)$) is a potential function defined over a clique C_i which reflects the neighbourhood relationship between nodes in C_i. Equation (5), in terms of clique potential function, can also be written as

$$p(x) = \frac{e^{-\sum_{i=1}^{m} u_i(c_i)/T}}{Z} \qquad (8)$$

We use $C = \{C_1, C_2, .., C_m\}$ to denote the set of all considered clique in $U(x)$. The set C can consist of all possible cliques in G, i.e., all maximal cliques, their sub cliques including singleton cliques. However, it is always possible to consider only the maximal cliques in C and define the energy $U(x)$. For example, the structure shown in Figure 4 has four maximal cliques

$$C_1 = \{X_1, X_2, X_3\},\ C_2 = \{X_2, X_3, X_4\}$$

$$C_3 = \{X_2, X_5\},\ C_4 = \{X_3, X_6\}$$

The formulation of Gibbs distribution for this structure can be written as

$$p(x) = \frac{e^{-\sum_{i=1}^{4} u_i(c_i)/T}}{Z} \qquad (9)$$

Temperature, T, has a very important role in Gibbs distribution. It controls the *sharpness* of the jpd. i.e. when the temperature is high, all configurations of X tends to be equally distributed. Conversely, near the zero temperature, the jpd concentrates around the *global energy minima*.

[4] In literatures, MRF is also defined in terms of sites and neighbourhood system [15], [16], where a site corresponds to a node X_i

MRF-Gibbs equivalence: A jpd on set of random variables, X, obeys following three conditions:

1) $p(x)->(0,1)$, Probability of each x lies between 0 and 1
2) $p(x)>0$, Positivity condition
3) $\sum_x p(x)=1$ Sum over probability of all possible x is 1

In addition, if X is an MRF, it also follows the local Markov property (4).

We can also define a *Gibbs Random Field* over X, which is characterised by its global property: the Gibbs distribution.

Definition 3.5 (Gibbs Random field): A set of random variables X with neighbourhood system N is said to be Gibbs Random Field (GRF), if and only if they obey a Gibbs distribution.

The Hammersley-Clifford theorem [18] then establishes the equivalence between the local Markov property of MRF and global Gibbs property of GRF.

Theorem 3.1 (The Hammersley-Clifford theorem): Any set of random variables X with a neighbourhood system N is an MRF if and only if X is also a GRF.

Proof: can be found in [15]. ■

In another words, Hammersley-Clifford theorem states that a jpd for any MRF can be equivalently specified as a Gibbs distribution (5). The practical value of the theorem is that, the behaviour of a system using Gibbs distribution completely depends on the chosen form of the potential functions, $u_i(c_i)$, and the temperature, T. These parameters can be varied in order to achieve desired system behaviour. We exploit this property of Gibbs distribution to estimate and sample the MRF in DEUM framework.

IV. Estimating the MRFs: a fitness modelling approach

In previous section we have formulated the joint probability of an MRF as a Gibbs distribution. In this section, we describe the way in which DEUM estimates its parameters.

In a typical EDA, the process of estimating $p(x)$ uses a selection method to identify a set of good solutions in a population. This set is then used to empirically determine the distribution of the terms in the factorisation. For example in UMDA, (1) is used as the model of distribution, where marginal probabilities for each x_i is calculated as follows,

$$p(x_i=1)=\frac{1}{N}\sum_{x\in D, x_i=1} x_i \tag{10}$$

A noticeable feature of this approach is that all selected solutions are given equal weight in determining the probabilistic model, even though they may vary greatly in fitness. This raises the question as to whether the fitness of individual solutions could be more accurately represented in the model and whether this would be beneficial in terms of algorithm performance.

We can relate fitness to probability more precisely by considering the mass distribution of fitness over solution space. This can be regarded as a probability distribution. We obtain a factorisation of this distribution using Markov Random Field (MRF) theory. Let us describe this in detail.

A. Using fitness to model the energy for the Gibbs distribution

Assuming that the probability of a solution is proportional to its fitness, the jpd, $p(x)$, can be modelled in terms of fitness as

$$p(x)=\frac{f(x)}{Z} \tag{11}$$

Where, $Z=\sum_{y\in\Omega} f(y)$ is the partition function and Ω is the set of all possible solutions.

For such $p(x)$, we have

1) $p(x)->[0,1]$, Probability of each x lies between 0 and 1
2) $p(x)>0$, Positivity condition: assumes $f(x)>0$. This can be maintained by mapping $f(x)$.
3) $\sum_{x\in\Omega} p(x)=1$, Sum over probability of all solution is 1

Now, from (5) and (11), we can deduce following equivalence of jpd for MRF in terms of fitness function.

$$p(x)=\frac{e^{-U(x)/T}}{\sum_{y\in\Omega} e^{-U(y)/T}} \equiv \frac{f(x)}{\sum_{y\in\Omega} f(y)} \tag{12}$$

From which, following relationship between fitness and the energy can be deduced [19].

$$-ln(f(x))=U(x) \tag{13}$$

For simplicity, here we assume T from (12) to be 1. In other words, (13) defines the equivalence shown in (12). We refer to (13) as **MRF Fitness Model (MFM)**. From (7), MFM can also be written in terms of potential functions as:

$$-ln(f(x))=\sum_{i=1}^{m} u_i(c_i) \tag{14}$$

Energy, $U(x)$, in MFM (13) gives the full specification of the jpd (5), so MFM can be regarded as a probabilistic model of the fitness function. Also notice that, minimising $U(x)$ here is equivalent to maximising $f(x)$.

At this point, it is important to notice that the log-linear form of MFM (14) is the result of our assumption of jpd as a mass distribution of fitness over solution space, as shown in (11). We could easily get different relationship between $f(x)$ and $U(x)$ by making diferent assumption about mass distribution of fitness function. For example, assuming $p(x)=\frac{e^{-f(x)}}{\sum_{y\in\Omega} e^{-f(y)}}$, we would get a linear MFM as $f(x)=\sum_{i=1}^{m} u_i(c_i)$.

In subsequent sections, we show how MFM (14) is used to estimate the parameters for the MRF.

B. Defining energy in terms of potential functions

In general, the form of energy, $U(x)$ in MFM, will model the different order of interaction between variables in X. The form of energy, however, will depend on our chosen potential functions over the cliques in the structure G. Here we define energy for a univariate and a bivariate structure. In subsequent sections we will formulate EDAs based on these two structures.

Univariate structure

Univariate structure assumes each variables $X_i \in X$ to be independent. The graph G for such structure will be an edge less graph. Therefore, the set of maximal cliques, C, in G would consist of n singleton cliques $C_i = \{X_i\}$. For each clique, $\{X_i\}$, we associate a potential function as follows:

$$u_i(x_i) = \alpha_i x_i \tag{15}$$

From (13) the MFM can then be written as:

$$-ln(f(x)) = U(x) = \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n \tag{16}$$

In terms of jpd (5), it can also be written as

$$p(x) = \frac{e^{-\sum_{i=1}^{n} \alpha_i x_i}}{Z} \tag{17}$$

where,

$$Z = \sum_{x \in \Omega} e^{-\sum_{i=1}^{n} \alpha_i x_i} \tag{18}$$

Here, α_i are the parameters associated with each clique $\{X_i\}$. α_i being the only unknown parameters of the potential function (15), completely specifies the $U(x)$ and therefore completely specifies the Gibbs distribution (17). Therefore, they are also known as *MRF parameters* [16]. We use θ to refer to vector of all MRF parameters in the model. For univariate case, the vector $\theta = \alpha = \{\alpha_1, \alpha_2, ..., \alpha_n\}$. In terms of MFM, (13), an MRF parameter measures the effect that the interaction between variables in a clique have on the fitness of the solution, $f(x)$. Obviously, in univariate case (16), α_i measures the effect of a single variable, X_i, on fitness.

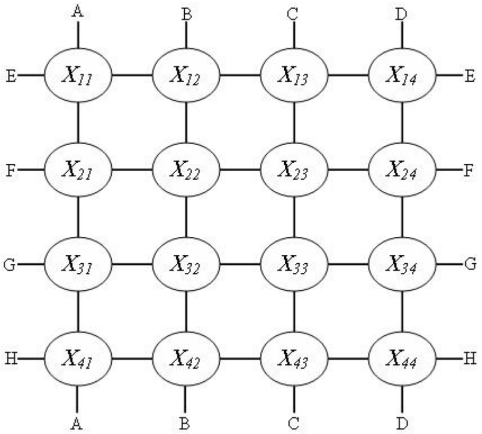


Fig. 5. A structure showing the bivariate interaction between variables in a two dimensional lattice

Bivariate structure: A bivariate structure represents the pair-wise interaction between variables. Here we consider a bivariate structure on two dimensional lattice with *toroidal* neighbourhood. For example, in Figure 5, a bivariate structure on a two dimensional lattice with $n = 4 \times 4$ variables is shown, where each variable $X_{ij} \in X$ interacts with four of its immediate neighbours. This structure can also be seen as an instance of the *Ising model* on two dimensional lattice [20]. The set of maximal cliques, C, in this case, contains 2×4^2 bivariate cliques $C_{ij,i'j'} = \{X_{ij}, X_{i'j'}\}$. This structure can be generalised to the $n = l \times l$ variables, where C will contain $m = 2l^2$ bivariate cliques. For each clique $\{X_{ij}, X_{i'j'}\}$, we assign a potential function $\beta_{ij,i'j'} x_{ij} x_{i'j'}$, where, each $\beta_{ij,i'j'}$ is the MRF parameter associated with bivariate clique $\{X_{ij}, X_{i'j'}\}$. The energy, $U(x)$ in MFM (13) for such X will therefore be

$$-ln(f(x)) = U(x) =$$

$$\sum_{i=1}^{l}\sum_{j=1}^{l}\left(\beta_{ij,(i+1)j} x_{ij} x_{(i+1)j} + \beta_{ij,i(j+1)} x_{ij} x_{i(j+1)}\right) \tag{19}$$

In terms of Gibbs distribution it can also be written as

$$p(x) = \frac{e^{-\sum_{i=1}^{l}\sum_{j=1}^{l}\left(\beta_{ij,(i+1)j} x_{ij} x_{(i+1)j} + \beta_{ij,i(j+1)} x_{ij} x_{i(j+1)}\right)/T}}{Z} \tag{20}$$

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. Let us define two types of MFM.

Definition 4.1 (Minimal MFM): We define a Minimal MFM as the MFM where the potential functions in $U(x)$ are defined on all the maximal cliques and not on any of their sub-cliques.

Definition 4.2 (Complete MFM): We define Complete MFM as MFM where the potential functions in $U(x)$ are defined on all the maximal cliques, their sub-cliques including singleton cliques.

Equation (19) is an example of a minimal MFM for the structure shown in Figure 5. We can also build a complete MFM for this structure. For this, we assign a potential function, $\alpha_{ij} x_{ij}$, to each singleton clique $\{X_{ij}\}$ in addition to potential functions $\beta_{ij,i'j'} x_{ij} x_{i'j'}$ for order 2 cliques $\{X_{ij}, X_{i'j'}\}$. The energy for the resulting MFM can be written as

$$-ln(f(x)) = U(x) =$$

$$\sum_{i=1}^{l}\sum_{j=1}^{l}\left(\alpha_{ij} x_{ij} + \beta_{ij,(i+1)j} x_{ij} x_{(i+1)j} + \beta_{ij,i(j+1)} x_{ij} x_{i(j+1)}\right) \tag{21}$$

We use α to denote the set of all n univariate parameters α_{ij}. Set of all MRF parameters θ will then contain both α and β.

C. Estimating the parameters of MRF

Once we define the potential function for the given structure of MRF and build a MFM, next step is to estimate the parameters of the MRF, θ. In DEUM, we do so by fitting the derived MFM to a dataset (i.e. set of solution), D.

Each solution in a given population provides an equation satisfying the MFM, where MRF parameters will be the unknown part. Applying this to a set D consisting of N solutions therefore allow us to estimate θ by solving the system of equations:

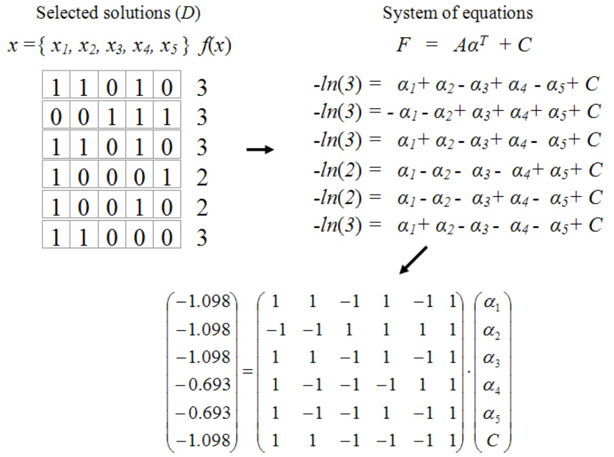
$$F = A\theta^T + C \tag{22}$$

Here, F is the column vector containing $-\ln(f(x))$ of all solutions in D, θ, is the vector of all MRF parameters [5], A is the matrix of values in D and C is a constant known as intercept of the system of equation [6].

For example, for univariate structure, a solution, x, in the set D will provide an equation satisfying (16). Where, left hand side of the equation will be the $-ln(f(x))$ and the right hand side will be the sum over the product of each $x_i \in x$ with the MRF parameter α_i. Here α_i is the unknown part of the equation. Note that, for mathematical reason, {-1,1} should be used as the values for x_i rather than $\{0, 1\}$. This ensures the arithmetical symmetry between possible values of x_i and is a standard practice in MRF modelling techniques. Applying (16) to the whole set, D, therefore allows us to estimate MRF parameters, α, by solving the system of equations:

$$F = A\alpha^T + C \tag{23}$$

Here, F is the N-dimensional column vector containing $-\ln(f(x))$ for the set of solutions in D. A is the $N \times n$ dimensional matrix of allele values in the set D, $\theta = \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ is the vector of MRF parameters. Figure 6 shows an example of a set of solutions, D, and the corresponding set of linear equations.



Selected solutions (D)

$x = \{x_1, x_2, x_3, x_4, x_5\}$ $f(x)$

1	1	0	1	0	3
0	0	1	1	1	3
1	1	0	1	0	3
1	0	0	0	1	2
1	0	0	1	0	2
1	1	0	0	0	3

System of equations

$$F = A\alpha^T + C$$

$$-ln(3) = \alpha_1 + \alpha_2 - \alpha_3 + \alpha_4 - \alpha_5 + C$$
$$-ln(3) = -\alpha_1 - \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + C$$
$$-ln(3) = \alpha_1 + \alpha_2 - \alpha_3 + \alpha_4 - \alpha_5 + C$$
$$-ln(2) = \alpha_1 - \alpha_2 - \alpha_3 - \alpha_4 + \alpha_5 + C$$
$$-ln(2) = \alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 - \alpha_5 + C$$
$$-ln(3) = \alpha_1 + \alpha_2 - \alpha_3 - \alpha_4 - \alpha_5 + C$$

$$\begin{pmatrix} -1.098 \\ -1.098 \\ -1.098 \\ -0.693 \\ -0.693 \\ -1.098 \end{pmatrix} = \begin{pmatrix} 1 & 1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ C \end{pmatrix}$$

Fig. 6. A set of solutions D and the corresponding set of linear equations including the constant C for univariate MFM

Depending on the relationship between N and n, the system will be under-, over-, or precisely-specified. A standard least square fitting algorithm can be used to give a estimation of the α_i (and the constant C). We state one of the most stable algorithm for this purpose known as Singular Value Decomposition (SVD) [21]. SVD can give useful results even when the system of linear equations is under-specified or over-specified.

This approach can be similarly extended for bivariate (or any multivariate) MFM. The size of the matrix A will depend on the number of MRF parameters in θ and the size of the set D. For example, if we consider a complete MFM for the bivariate structure (Equation 21), the size of the matrix will be $N \times s$, where s, the length of θ, is $3n$ as θ will contain both n parameters in α and $2n$ parameters in β.

V. Sampling the MRF

Once the estimation of an MRF is completed, next step is to sample from the model. One of the way to sample from MRF is by using its local Markov property (4), i.e. by estimating the marginal probability, $p(x_i|N_i)$, of each variable X_i conditional upon set of its neighbouring variables N_i. $p(x_i|N_i)$ can then be used to sample further x_i. Note that, for univariate model (16), $p(x_i|N_i)$ generalises to $p(x_i)$. In general, $p(x_i|N_i)$ could be directly estimated from the population by means of frequency counting, as done in other Bayesian Networks based EDAs. In DEUM, however, we estimate it from the jpd, (5).

A. *Finding marginals from the Gibbs distribution*

Let us use x^+ to denote a solution x having a particular $x_i = +1$ and x^- to denote a soltuion x having $x_i = -1$. The probability that the value of the variable in position i is equal to 1 given its neighbours, $p(x_i = 1|N_i)$, can then be written as

$$p(x_i = 1|N_i) = \frac{p(x^+)}{p(x^+) + p(x^-)} \tag{24}$$

Substituting $p(x)$ from (5) and cancelling the Z, we get

$$p(x_i = 1|N_i) = \frac{e^{-U(x^+)/T}}{e^{-U(x^+)/T} + e^{-U(x^-)/T}} \tag{25}$$

or,

$$p(x_i = 1) = \frac{1}{1 + e^{(U(x^+) - U(x^-))/T}} \tag{26}$$

Since, $U(x^+)$ and $U(x^-)$ agree in all terms other than those containing x_i, 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 $x_i = 1$ conditional upon N_i:

$$p(x_i = 1|N_i) = \frac{1}{1 + e^{2W_i/T}} \tag{27}$$

Similarly, we can get following expression as the estimate of the marginal probability for $x_i = -1$:

$$p(x_i = -1|N_i) = \frac{1}{1 + e^{-2W_i/T}} \tag{28}$$

Here, W_i is the difference in two energies, $U(x^+)$ and $U(x^-)$, after substituting the x_i to 1 for all the remaining terms in $U(x^+)$ and to -1 for all remaining terms in $U(x^-)$. For example, W_i for the univariate MFM (16) simplifies to

$$W_i = \alpha_i \tag{29}$$

and therefore the marginal probability of $x_i = 1$ simplifies to

$$p(x_i = 1) = \frac{1}{1 + e^{2\alpha_i/T}} \tag{30}$$

Similarly, W_i (or more precisely W_{ij}) for bivariate MFM (19) simplifies to

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

[5] θ^T is the transpose of vector θ to make it a column vector

[6] C can be seen as the parameter associated with an empty clique in structure G

$$\beta_{(i-1)j,ij}x_{(i-1)j} + \beta_{i(j-1),ij}x_{i(j-1)} \tag{31}$$

and therefore the marginal probability of $x_{ij} = 1$ simplifies to

$$p(x_{ij} = 1|N_{ij}) = \frac{1}{1 + e^{2\left(\substack{\beta_{ij,(i+1)j}x_{(i+1)j}+\beta_{ij,i(j+1)}x_{i(j+1)}+ \\ \beta_{(i-1)j,ij}x_{(i-1)j}+\beta_{i(j-1),ij}x_{i(j-1)}}\right)/T}} \tag{32}$$

B. Role of temperature in sampling a Gibbs distribution

As we said earlier, temperature, T, has a very important role in Gibbs distribution. It controls the *convergence* of the distribution. In equation (27), as $T \to 0$, the value of $p(x_i = 1|N_i)$ tends to a limit depending on the W_i. If $W_i > 0$, then $p(x_i = 1|N_i) \to 0$ as $T \to 0$. Conversely, if $W_i < 0$, then $p(x_i = 1|N_i) \to 1$ as $T \to 0$. If $W_i = 0$, then $p(x_i = 1|N_i) = 0.5$ regardless of the value of T. Therefore, the W_i are indicators of whether the x_i at the position i should be 1 or -1. This indication becomes stronger as the temperature is cooled towards zero. In subsequent sections we present instances of DEUM, that uses temperature to control the convergence of the probability distribution, and therefore control the convergence of the algorithm.

VI. Instances of DEUM

In this section we describe two instances of DEUM algorithms. We also highlight some of the experimental results on their performance. The aim here is to provide an overview of some of the working example of DEUM. More complete description of these and other DEUM algorithms together with detail experimental results can be found in [9], [22], [23], [24], [25], [10].

A. DEUM$_d$: A univariate DEUM with direct sampling from Gibbs distribution

In this section, we describe a DEUM, which estimates univariate model of probability distribution (17) and samples from it. We call it a DEUM instance with direct sampling from Gibbs distribution (DEUM$_d$). DEUM$_d$ begins by initialising a population of solution P. The N best solution is then selected from P. MRF parameters, α, are then calculated by fitting the univariate MFM, (16), on the selected set of solution. This is achieved by solving the system of linear equations, (23). The $p(x_i = 1)$ is then calculated from equation (27) and sampled to generate the child population. The child then replaces the parent, P, and this process continues until termination criteria are satisfied.

The five-step workflow for DEUM$_d$ is shown in Figure (7).

As described earlier, κ, i.e. inverse of T, has a direct effect on the convergence of Gibbs distribution and therefore on the convergence of the DEUM$_d$. As the number of iterations, g, grows, the marginal probability, $p(x_i)$, gradually cools to either 0 or 1. However, depending upon the type of problem, different cooling rates may be required. In particular, there is a trade-off between convergence speed of the algorithm and the exploration of the search space. Therefore, the cooling rate parameter, τ, has been introduced. τ gives explicit control over the convergence speed of DEUM$_d$. Decreasing τ slows the cooling, resulting in better exploration of the search space. However, it also slows the convergence of the algorithm. Increasing τ, on the other hand, makes the algorithm converge faster. However, the exploration of the search space will be reduced.

Distribution Estimation using MRF with direct sampling (DEUM$_d$)

```
1) Generate an initial population, $P$, of size $M$.
2) Select set $D$ from $P$ consisting of $N$ fittest solutions,
   where $N \leq M$.
3) Calculate the MRF parameters $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ by
   fitting univariate MFM to $D$.
4) Generate $M$ new solutions using the following distribution:
       $p(x) = \frac{e^{-\sum_{i=1}^{n} \alpha_i x_i / T}}{Z}$
   where, $p(x_i = 1) = \frac{1}{1+e^{\kappa\alpha_i}}$ and $p(x_i = -1) = \frac{1}{1+e^{-\kappa\alpha_i}}$. Here, $\kappa$, the inverse of temperature $T$, is
   defined as $\kappa = g\tau$ where, $g$ is the number of the current
   iteration and $\tau > 0$ is a cooling rate parameter chosen
   by the user.
5) Replace $P$ by the new population, and go to Step 2 until
   the termination criterion is satisfied.
```

Fig. 7. The pseudo-code of the Distribution Estimation Using MRF with direct sampling (DEUM$_d$) algorithm

B. Is-DEUM$_g$: A bivariate DEUM with a Monte Carlo approach to sampling

Since there are no dependency between variables in univariate case, marginal probability for each variable, $p(x_i|N_i)$ simplifies to the univariate marginal probability, $p(x_i)$, computation of which only involves the value for the variable itself. However, in bivariate (or multivariate) case, it is essential to know the value of N_i, which is then used to estimate W_i in (31). Since, we are trying to optimise both X_i and its neighbours N_i at the same time, and the value for one effects the value for another, it is difficult to decide on the value of N_i to be used for estimating $p(x_i|N_i)$. This, therefore, does not allow us to directly estimate and sample the marginal probabilities as done in DEUM$_d$.

In order to extend DEUM to bivariate (and multivariate) case, we need to resolve this situation, for which we propose to use an iterative sampling technique, known as Monte Carlo samplers [26]. Specifically, we are interested in Gibbs sampler (GS) [27]: a well known instance of the Monte Carlo sampler.

In this section we describe a version of DEUM that use previously defined bivariate model, (20), as its model of distribution and the Gibbs sampler as sampling technique. Since, the structure for (20) can be seen as a variant of Ising model, the name Is-DEUM$_g$ has been used. The symbol g stands Gibbs sampler.

Gibbs Sampler (GS)

```
1) Generate a solution $x^o = \{x_1^o, x_2^o, .., x_n^o\}$
2) Set the initial value for $T$.
3) Repeat:
    a) Select a variable $x_i^o$ from $x^o$ and set $x_i^o = 1$ with
       probability $p(x_i^o = 1|N_i^o)$
    b) Decrease $T$
   :Until termination criteria is satisfied
4) Terminate with answer $x^o$.
```

Fig. 8. The pseudo-code for a Gibbs Sampler

Figure 8 shows the workflow of a Gibbs sampler.

The general idea of a Gibbs sampler is to repeatedly sample variables in x^o until a termination criteria is satisfied, such that a (locally) optimal x^o is produced. Different termination criteria could be could be used for this purpose, for example, to terminate after a fixed number of iteration is performed, or to terminate if no further improvement in energy $U(x^o)$ could be found. The temperature coefficient, T, in Gibbs sampler can be used to control the convergence of $p(x_i^o|N_i^o)$. In each iteration, T is decreased. This gradually converges $p(x_i^o|N_i^o)$ to its limit. This iterative process would produce a x^o that, depending upon the allowed iteration, would be closer to a optima pointed by current set of MRF parameters θ.

Figure 9 shows the workflow of Is-DEUM$_g$, which incorporates Gibbs sampler as its sampling method.

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 $\theta$ by fitting (20) to $D$.
4) Run Gibbs sampler $M$ times to generate new population.
5) If termination criteria is not satisfied, replace parent with
   new population and go to step 2
```

Fig. 9. The pseudo-code of the DEUM with Gibbs Sampler

As we said earlier, the convergence of GS depends on two factors: a) how fast we decrease the temperature T and b) how many iteration we allow in the GS. This, therefore, also effects the performance of the Is-DEUM$_g$. Decreasing T quickly may result in premature convergence of x^o. Conversely, decreasing T slowly may result in high computation cost. Similarly, allowing GS to iterate for large number of runs would converge x^o to some local optima pointed by the current set of MRF parameters θ. This, therefore, would result in increasing number of similar solutions being present in the new population that are converged to some local optima. If the current optima pointed by θ is not the global optima for the problem, the result would be a quick loss of diversity in the population, even straight after the initial generation. Therefore, setting the correct rate of change for temperature and setting the allowed number of iteration is crucial in the performance Is-DEUM$_g$.

C. Results

Number of experimental analysis with wide range of different optimisation problems has been done to test the performance of these (and other) DEUM instances. Detail description of them, as stated in previous section, can be found in the publications elseware [9], [22], [23], [24], [25], [10]. In this section we quickly go through some of these interesting results.

Univariate problem

Since DEUM$_d$ is a univariate EDA, obvious problem to test it is with the univariate problem. The experimental results with the OneMax problem, a typical univariate problem, showed that the performance of DEUM$_d$ is significantly better than that of Univariate Marginal Distibution Algorithm (UMDA) [4], a univariate EDA, and a Simple GA with uniform crossover (Figure 10). More specifically, it showed that, by setting the very low initial temperature, close to zero, (i.e. vary high value for τ), the DEUM$_d$ was able to find the solution in a first generation requiring only about $1.5n$ fitness evaluation. The explanation to this result is that the low temperature tends to converge the distribution to an extreme, in the very first generation. The value of $p(x_i)$ taking one of the extrema of either 0 or 1 then depends only on the value of estimated MRF parameters α_i, as can be seen from (30). For OneMax problem, fitness modelling approach to estimating MRF parameters gives a very accurate estimation of α in the initial generation, and therefore the first solution sampled from the converged marginal probability is optimal. The number of fitness evaluation, $1.5n$, is therefore the size of the population.

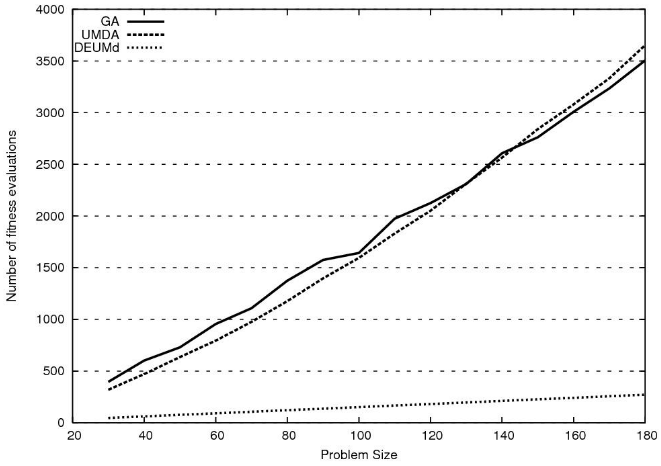


Fig. 10. Scalability of DEUM$_d$ for onemax problem

Deceptive problem

Univariate EDAs do not scale well with deceptive problem. This is mainly because local improvement in fitness, while solving such problems, misleads the algorithm away from the global optima. This together with the fact that the univariate EDA do not take into account any interaction between variables makes them a poor performer in deceptive problems. Interestingly, experimental results on the performance of DEUM$_d$ for a 60 bit Trap function of order 5 [28], a difficult variant of deceptive problem, showed that by slowing the cooling schedule, DEUM$_d$ was consistently able to find the solution for this problem. In contrast, other univariate EDAs, such as Population Based Incremental Learning (PBIL) [29] and Univariate Marginal Distribution Algorithm (UMDA) [4], was not able to solve this problem, even with a very high population size and with extremely large number of fitness evaluation. Figure 11 plots the Run Length Distribution (RLD) curve [30] for DEUM$_d$, and a GA [7], on 60 bit trap function. RLD shows the cumulative percentage of successful runs that terminated within certain number of fitness evaluation.

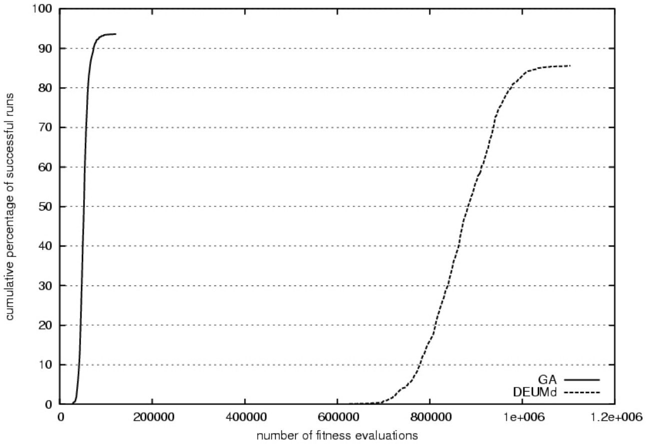


Fig. 11. Performance of DEUM$_d$ on 60 bit Trap function of order 5

Ising problem

Ising spin glass problem has been introduced in early 1920s to model the spin glass system. They have range of practical applications in both statistical physics and AI. 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 [6], [28], [31], [11]. The experimental results on the performance of Is-DEUM$_g$ for this problem showed that, in terms of number of fitness evaluations needed to find the solution, it significantly outperformed other EDAs previously applied to this problem. In particular, it has been found that the MRF parameters θ estimated from the initial population contained enough information to correctly predict the global optimum, i.e., by slowly decreasing the temperature and by allowing the high number of iteration in the GS, the optimum solution was found within first few x^o sampling for the new population. This highly reduced the number of fitness evaluation required by Is-DEUM$_g$.

VII. Conclusion

In this paper we presented DEUM as a general framework of an EDA based on MRF. We described three main component of DEUM, 1) MRF models, 2) Fitness modelling for parameter estimation and 3) Sampling from MRF, and showed how these components interact together to perform optimisation in the two different instances of DEUM. We also briefly described some of the interesting results on the performance of these algorithms.

There are two main explanations for the success of DEUM algorithms. Firstly, DEUM builds a model of fitness function to approximate the MRF. This contrasts with other EDAs that build a model of selected solutions, where each selected solution has equal contribution to the probability distribution. Fitness modelling allows DEUM to use fitness in variation part of the evolution by regulating the contribution of a solution to the estimation of probability distribution. Secondly, DEUM exploits the temperature coefficient in the Gibbs distribution to regulate the behaviour of the algorithm. In particular, with higher temperature, the distribution is closer to being uniform, and with lower temperature, it concentrates near some optima. This gives DEUM an explicit control over the convergence of the algorithm. These two factors, put together, makes DEUM a promising framework for optimisation of the real world problems.

[7] As expected, a GA with onepoint crossover was also able to find the solution

References

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[2] I. Rechenberg, *Evolutionstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Formman-Holzboog, 1973.

[3] L. J. Fogel, “Autonomous automata,” *Industrial Research*, vol. 4, pp. 14–19, 1962.

[4] H. Mühlenbein and G. Paaß, “From recombination of genes to the estimation of distributions: I. binary parameters,” in *Parallel Problem Solving from Nature – PPSN IV*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Berlin: Springer, 1996, pp. 178–187. [Online]. Available: citeseer.nj.nec.com/uehlenbein96from.html

[5] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2002.

[6] M. Pelikan and D. E. Goldberg, “Hierarchical BOA solves Ising spin glasses and MAXSAT,” *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, pp. 1271–1282, 2003, also IlliGAL Report No. 2003001.

[7] J. A. Lozano, R. Sagarna, and P. Larrañaga, “Solving job scheduling with Estimation of Distribution Algorithms,” in *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, P. Larrañaga and J. A. Lozano, Eds. Kluwer Academis Publishers, 2001, pp. 231–242.

[8] Q. Zhang, J. Sun, and E. Tsang, “Eda+ga: Evolutionary algorithm with guided mutation for the maximum clique problem,” *IEEE Trans. on Evolutionary Computation*, vol. 9, pp. 192–200, 2005.

[9] S. K. Shakya, J. A. W. McCall, and D. F. Brown, “Updating the probability vector using MRF technique for a Univariate EDA,” in *Proceedings of the Second Starting AI Researchers’ Symposium, volume 109 of Frontiers in artificial Intelligence and Applications*, E. Onaindia and S. Staab, Eds. Valencia, Spain: IOS press, August 2004, pp. 15–25.

[10] S. Shakya, "Deum: A framework for an estimation of distribution algorithm based on markov random fields," Ph.D. dissertation, The Robert Gordon University, Aberdeen, UK, April 2006.

[11] R. Santana, "Estimation of Distribution Algorithms with Kikuchi Approximation," *Evolutonary Computation*, vol. 13, pp. 67–98, 2005.

[12] R. Kikuchi, "A Theory of Cooperative Phenomena," *Physical Review*, vol. 81, pp. 988–1003, Mar. 1951.

[13] P. Larrañaga, R. Etxeberria, J. A. Lozano, B. Sierra, I. Inza, and J. M. Peña, "A review of the cooperation between evolutionary computation and probabilistic graphical models," in *Second Symposium on Artificial Intelligence. Adaptive Systems. CIMAF 99*, 1999, pp. 314–324, la Habana.

[14] M. I. Jordan, Ed., *Learning in Graphical Models*, ser. NATO Science Series. Dordrecht: Kluwer Academic Publishers, 1998.

[15] J. Besag, "Spatial interaction and the statistical analysis of lattice systems (with discussions)," *Journal of the Royal Statistical Society*, vol. 36, pp. 192–236, 1974.

[16] S. Z. Li, *Markov Random Field modeling in computer vision*. Springer-Verlag, 1995.

[17] K. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, 2002.

[18] J. M. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," *Unpublished*, 1971.

[19] 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*, vol. 2310, pp. 65–78, January 2002.

[20] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*. AMS, 1980.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, UK: Cambridge University Press, 1993.

[22] S. Shakya, J. McCall, and D. Brown, "Estimating the distribution in an EDA," in *In proceedings of the International Conference on Adaptive and Natural computiNG Algorithms (ICANNGA 2005)*, B. Ribeiro, R. F. Albrechet, A. Dobnikar, D. W. Pearson, and N. C. Steele, Eds. Coimbra, Portugal: Springer-Verlag, Wien, 2005, pp. 202–205.

[23] ——, "Using a Markov Network Model in a Univariate EDA: An Emperical Cost-Benefit Analysis," in *proceedings of Genetic and Evolutionary Computation COnference (GECCO2005)*. Washington, D.C., USA: ACM, 2005, pp. 727–734.

[24] 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)*, vol. 3. Edinburgh, UK: IEEE press, 2005, pp. 2576–2583.

[25] ——, "Solving the ising spin glass problem using a bivariate eda based on markov random fields," in *proceedings of IEEE Congress on Evolutionary Computation (IEEE CEC 2006)*. Vancouver, Canada: IEEE press, 2006.

[26] N. Metropolis, "Equations of state calculations by fast computational machine," *Journal of Chemical Physics*, vol. 21, pp. 1087–1091, 1953.

[27] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions and the bayesian restoration of images," in *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, M. A. Fischler and O. Firschein, Eds. Los Altos, CA.: Kaufmann, 1987, pp. 564–584.

[28] M. Pelikan, "Bayesian optimization algorithm: From single level to hierarchy," Ph.D. dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, 2002, also IlliGAL Report No. 2002023.

[29] S. Baluja, "Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning,," Pittsburgh, PA, Tech. Rep. CMU-CS-94-163, 1994. [Online]. Available: citeseer.nj.nec.com/baluja94population.html

[30] H. H. Hoos and T. Stutzle, "Towards a characterisation of the behaviour of stochastic local search algorithms for SAT," *Artificial Intelligence*, vol. 112, no. 1-2, pp. 213–232, 1999. [Online]. Available: citeseer.nj.nec.com/hoos99towards.html

[31] R. Santana, "Probabilistic modeling based on undirected graphs in estimation distribution algorithms," Ph.D. dissertation, Institute of Cybernetics, Mathematics and Physics, Havana, Cuba, 2003.