# Solving the MAXSAT Problem using a Multivariate EDA based on Markov Networks

Alexander Brownlee
School of Computing
The Robert Gordon University
St Andrew St, Aberdeen, UK
+44 1224 262472
sb@comp.rgu.ac.uk

John McCall
School of Computing
The Robert Gordon University
St Andrew St, Aberdeen, UK
+44 1224 262780
jm@comp.rgu.ac.uk

Deryck Brown
School of Computing
The Robert Gordon University
St Andrew St, Aberdeen, UK
+44 1224 262729
db@comp.rgu.ac.uk

## ABSTRACT
Markov Networks (also known as Markov Random Fields) have been proposed as a new approach to probabilistic modelling in Estimation of Distribution Algorithms (EDAs). An EDA employing this approach called Distribution Estimation Using Markov Networks (DEUM) has been proposed and shown to work well on a variety of problems, using a unique fitness modelling approach. Previously DEUM has only been demonstrated on univariate and bivariate complexity problems. Here we show that it can be extended to a difficult multivariate problem and is capable of accurately modelling a fitness function and locating an optimum with a very small number of function evaluations.

## Categories and Subject Descriptors
I.2.8 **[Artificial Intelligence]**: Problem Solving, Control Methods, and Search;
G.3 **[Probability and statistics]**: Probabilistic algorithms, Stochastic processes

## General Terms
Algorithms, Performance, Theory

## Keywords
Estimation of Distribution Algorithms, Evolutionary Algorithms, Probabilistic Modelling

## 1. INTRODUCTION
Estimation of Distribution Algorithms (EDAs) [6] is a well-established topic in the field of evolutionary algorithms. EDAs develop the concept of evolution found in a GA, continuing the principals of selection and variation. They differ by replacing the traditional genetic reproduction operators of crossover and mutation with the construction and sampling of a probabilistic model. EDAs are typically categorised by the complexity of their probabilistic model structure as univariate, bivariate or multivariate [8].

DEUM is a framework for constructing EDAs using Markov Networks to model the fitness function. The initial theory was published in [1] and DEUM was first presented in [11]. Background on DEUM can be found in [12].

In [13] DEUM was extended to bivariate problems, using the Ising Spin Glass problem [4] as a test case. The experiments described in that paper used an implementation of DEUM with a fixed bivariate model structure. Here we propose an extension using a fixed multivariate structure; specifically that of the MAXSAT problem. The performance of the algorithm is measured, using the hierarchical Bayesian Optimisation Algorithm (hBOA) and WalkSAT algorithm presented in [9] as benchmarks. The ability of the algorithm to model fitness is demonstrated by using it to predict the fitness of randomly generated individuals.

The overall purpose of this paper is to demonstrate that DEUM can successfully construct an accurate model of the fitness function and that this can be extended to higher complexity problems than in previous publications.

The remainder of this paper is structured as follows. Section 2 looks at the MAXSAT problem in more detail. Section 3 goes on to describe DEUM, the fitness modelling approach it uses and how it is sampled. Section 4 describes and presents results of an experiment comparing the performance of DEUM on MAXSAT with that of other algorithms. An analysis of the results is then presented. In Section 5 we describe an experiment with demonstrates the fitness modelling capability of DEUM with results and analysis. Section 6 concludes the paper and summarises our intended future work on this topic.

## 2. MAXSAT AND EDAS
The Maximum Satisfiability or MAXSAT Problems are described in [3]. A MAXSAT problem attempts to find a set of values which maximises the number of satisfied clauses of a fixed predicate logic formula expressed in conjunctive normal form. Many real-world problems can be mapped on to MAXSAT, including the well-known graph colouring problem. It is known to be NP-complete in its general form. The SATLIB resource [2] provides a collection of a large number of sample MAXSAT problems.

MAXSAT is particularly useful for experiments in modelling high order interactions as each instance of the problem uses a known predefined structure. It has already been used in benchmarking an EDA [9].

Encoding MAXSAT for evolutionary algorithms is a straightforward task. The candidate solutions are bitstrings in which each bit encodes a predicate variable in the formula. An individual's


*GECCO'07*, July 7–11, 2007, London, England, United Kingdom.

fitness is equal to the number of satisfied clauses given the predicate values in it.

# 3. BACKGROUND

## 3.1 Distribution Estimation Using Markov Networks

DEUM uses a Markov Network to model fitness as an energy distribution over the solution space. A Markov Network is an undirected graphical model, in contrast to the directed graphical models such as Bayesian Networks used by many EDAs.

A Markov Network models a set of random variables as nodes on a graph, and interactions between those variables as edges. It is characterised by a property known as Markovianity, which states that the distribution of any node can be completely defined by the values of its neighbouring nodes. The Markov Network may be viewed as a set of cliques, a clique simply being any fully connected subgraph of the graphical model. This allows a joint probability distribution for the Markov Network to be defined in terms of the Gibbs distribution:

$$p(x) = \frac{f(x)}{\sum_y f(y)} \equiv \frac{e^{-U(x)/T}}{\sum_y e^{-U(y)/T}} \quad (1)$$

U(x) is a sum of *clique potential functions*, each of which models the neighbourhood relationship between variables in one particular clique on the graph. The summations are over all possible solutions *y*. *T* is a temperature coefficient, which remains set to 1 in all of our current experiments. For some problems slowly reducing this temperature could prevent premature convergence on a poor model.

As with many evolutionary algorithms DEUM models a set of individuals. Each individual $x = \{x_1, x_2, ..., x_n\}$ is a particular set of values which can be applied to the set of variables $X = \{X_1, X_2, ..., X_n\}$ in a problem – here these are the MAXSAT predicate variables. Each individual is assigned a fitness to denote the quality of solution it represents.

In [1] it was shown that an equation for each individual in a population may be derived from the joint probability distribution shown in (1). This relates solution fitness to an energy function calculated from the values taken by variables in a set of individuals:

$$-\ln(f(x)) = U(x) \quad (2)$$

Here, *f(x)* is the fitness of an individual *x* and *U(x)* is the energy function derived from alleles. *U(x)* fully specifies the joint probability distribution, so can be regarded as a probabilistic model of the fitness function. Minimising *U(x)* is equivalent to maximising *f(x)*.

In early versions of DEUM, the Markov Network used a univariate structure, and there was only one clique for each variable. Ising-DEUM [13] introduced extra cliques for the bivariate interactions. Here we expand this further to incorporate terms for interactions between up to 3 variables, which we call trivariate terms. The sets of interactions are derived directly from the structure of the given MAXSAT problem instance. This gives us an energy function for each individual which can be expressed as:

$$U(x) = \begin{array}{l} c + \sum_{i=0}^{n} \alpha_i x_i + \sum_{i=0}^{n} \sum_{j=0}^{n} \alpha_{ij} x_i x_j \\ + \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=0}^{n} \alpha_{ijk} x_i x_j x_k \end{array} \quad (3)$$

where each $\alpha$ is a parameter associated with a clique on the Markov Network, $c$ is a constant representing the *zero-clique* of background energy in the Markov Network, $n$ is the number of variables in each individual and $x_i$ represents the value of variable $i$ in the solution $x$. Here {-1,1} are used as the values of $x_i$ in place of {0,1} to ensure arithmetical symmetry between values. The set of $\alpha$ values completely model the distribution.

An example is helpful to illustrate this. A simple MAXSAT problem has the set of predicates in (4).

$$(x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee x_3) \wedge (\bar{x}_4 \vee x_2) \quad (4)$$

The negations may be ignored when considering the relationships between predicate variables giving us the undirected graphical structure shown in Figure 1.

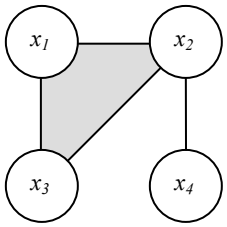


**Figure 1: Relationships Between Predicate Variables**

In the general energy function (5) for this problem, we have a constant, a term for each of the predicate variables $x_i$, a term for the bivariate interactions shown as edges on the graph and a term for the trivariate interaction shown by the shaded area.

$$U(x) = \begin{array}{l} c + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_4 + \alpha_{12} x_1 x_2 \\ + \alpha_{13} x_1 x_3 + \alpha_{23} x_2 x_3 + \alpha_{24} x_2 x_4 + \alpha_{123} x_1 x_2 x_3 \end{array} \quad (5)$$

An individual $x = \{0011\}$, with fitness $f(x) = 2$ would thus have the energy function shown in (6).

$$-\ln(2) = c - \alpha_1 - \alpha_2 + \alpha_3 + \alpha_4 + \alpha_{12} - \alpha_{13} - \alpha_{23} - \alpha_{24} + \alpha_{123} \quad (6)$$

To determine the Markov Network parameters, a random population is formed in the normal manner for an evolutionary algorithm. The energy function for each individual is formed resulting in a set of equations relating $\alpha$ values, energy (derived from fitness) and alleles. Singular value decomposition (SVD) [10] is used to solve the system of simultaneous equations and determine the unknown $\alpha$ values. The entire population is used in this process – selective pressure comes from energy minimisation in the model rather than traditional selection operators. The model is then sampled to generate a new population for the next generation.

## 3.2 Gibbs Sampler

To generate a new population DEUM can employ a number of sampling techniques. In [13] it was found that the Gibbs sampler performed well on more complex models, and that is the technique employed here. The Gibbs sampler repeatedly samples marginal probabilities for individual variables. For each variable $x_i$ the marginal probability of that variable taking the value 1 is given by:

$$p(x_i = 1) = \frac{1}{1 + e^{2Wi/T}} \quad (6)$$

where $T$ is a temperature constant and $W_i$ is an energy function for all the cliques which contain $x_i$:

$$W_i = \alpha_i + \sum_{\substack{j=0 \\ j \neq i}}^{n} \alpha_{ij} x_j + \sum_{\substack{j=0 \\ j \neq i}}^{n} \sum_{\substack{k=0 \\ k \neq i}}^{n} \alpha_{ijk} x_j x_k \quad (7)$$

The temperature $T$ falls over the run of the Gibbs sampler according to a cooling scheme.

We have made methodological changes to the implementation of the algorithm used in [13], both of which we have found empirically to yield better results. Firstly, we now adopt the exponential cooling scheme proposed by Kirkpatrick in [5]. The scheme starts with an initial temperature $t_o$, and at generation g the temperature is given by:

$$t_g = k t_{g-1} \quad (8)$$

where $k$ is a constant in the range $0 < k < 1$.

The second modification to the Bitwise Gibbs Sampler used in [13] is that bits are now sampled at random rather than in a raster scan. The sampler runs until no further improvement to the current individual or 10000 iterations have completed.

The sampler runs as follows:

Repeat for each individual $x^o$ in the previous population:

1. Set $g = 0$ and set initial value for $T$
2. Repeat:
   - 2.1 Set $x^{tmp} = x^o$
   - 2.2 Pick a variable $x^o_i$ at random
   - 2.3 Compute marginal probability distribution for $x^o_i$ according to (7)
   - 2.4 Sample distribution to obtain new value for $x^o_i$
   - 2.5 Increase $g$ by 1

   Until $x^{tmp} = x^o$ or $g = 10000$

Terminate with answer $x^o$

## 3.3 DEUM with Gibbs Sampler

Incorporating the Gibbs sampler into DEUM is a straightforward task, giving us the following algorithm:

1. Generate an initial population, $P$, of size $M$ with uniform distribution.

2. Calculate the Markov Network parameters by making a maximum likelihood estimation from the initial population.

3. Use Gibbs sampler to sample Markov Network, until $M$ individuals or an optimal individual is generated

4. Terminate with the fittest solution found in step 4.

Notice that there is only a single generation to this algorithm. It may be adapted to run for multiple generations by repeating steps 2-3 using the population generated in 3 to build a new model. We have not done this here as we found that once the model has been generated using an appropriately sized population, repeatedly sampling the model with different random starts will normally yield an optimum.

# 4. SOLVING THE MAXSAT PROBLEM

## 4.1 Aims

The aim of this experiment was to draw a comparison between the extended DEUM algorithm and existing algorithms applied to MAXSAT. Chosen for comparison are a multivariate EDA (hBOA) and a MAXSAT-specific algorithm (WalkSAT).

## 4.2 Method

The algorithm was run on the set of 3-CNF benchmark problems obtained from SATLIB [2]. The problem sizes were 20, 50, 75, 100, 125 and 150 (as used in [9]) On each size of problem, the algorithm was repeated on 20 different instances selected at random from the set held by SATLIB. Each instance tested belongs to the phase transition region, the point at which the problems tip from generally solvable instances to generally unsolvable. This occurs where the number of clauses is equal to the number of predicates multiplied by 4.3. Each instance tested is from the set of those proven to be solvable.

In the first experiment single generation DEUM was run to find an optimum. The problem is considerably more complex than the Ising problem described in [13]. Consequently, the Gibbs sampler must run for considerably longer and with a much slower cooling rate than used on that problem: the cooling rate parameter was 0.995 for problems up to size 100 and 0.999 for the larger problems. The sampler ran for a maximum of 10000 iterations, in contrast to 500 iterations on the Ising problem. The population size used in each case was the number of terms in the energy function multiplied by 1.1.

The number of fitness evaluations reported is the number used in evaluating the initial population plus the number used in evaluating the individuals generated by the Gibbs sampler. Also reported is the number of internal probability calculations used by the Gibbs sampler.

## 4.3 Results

Table 1 shows the experimental results of the performance of DEUM on differently sized instances of the MAXSAT problem. The first column (PS) is the problem size. Next is the number of fitness evaluations (FE) required by the algorithm to find an optimum, including evaluation of the initial population and the population produced by the Gibbs sampler, averaged over the 20 runs. The corresponding standard deviation (FE-SD) for this data is also included. Next is the average number of iterations (IT) of the Gibbs sampler followed by its standard deviation (IT-SD) – each iteration contains one marginal probability calculation. The final column shows the success rate (SR) of finding the optimum over the 20 runs. The number of function evaluations required to find an optimum is also represented graphically in Figure 2, where the error bars represent one standard deviation in the set of results.

**Table 1: Performance of DEUM on MAXSAT Problem**

| PS | FE | FE-SD | IT | IT-SD | SR |
|---|---|---|---|---|---|
| 20 | 533 | 22 | $5.27 \times 10^6$ | $6.87 \times 10^4$ | 100 |
| 50 | 1626 | 82 | $1.58 \times 10^7$ | $1.87 \times 10^5$ | 100 |
| 75 | 2980 | 608 | $2.49 \times 10^7$ | $1.85 \times 10^5$ | 100 |
| 100 | 3667 | 507 | $3.40 \times 10^7$ | $1.17 \times 10^5$ | 100 |
| 125 | 5151 | 826 | $4.32 \times 10^7$ | $1.99 \times 10^5$ | 65 |
| 150 | 6853 | 1510 | $5.25 \times 10^7$ | $1.51 \times 10^5$ | 70 |

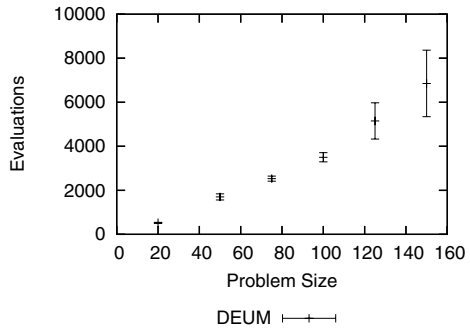


**Figure 2: Performance of DEUM on MAXSAT Problem**

## 4.4 Analysis

### *4.4.1 Comparison with hBOA and WalkSAT*

Previously hBOA had been tested on the same set of problems and had been reported to perform comparably with the MAXSAT-specific solver WalkSAT, when hBOA was used in a hybrid with the deterministic hillclimber GSAT. From Table 1 we can see that DEUM requires significantly fewer fitness evaluations than reported for hBOA and without the use of a hybrid approach. For example, to solve instances of the problem at size 100 bits, DEUM requires an average of 3500 evaluations. This compares to $10^5$ evaluations for hBOA + GSAT. DEUM also compares favourably with WalkSAT, which was reported to require $10^4$ evaluations. The decreasing success rate with larger problems is most likely caused by the Gibbs sampler, which is partly dependent on the random start and is highly dependent on an optimal cooling rate and run time.

### *4.4.2 Overhead*

We previously reported that for Ising problem the computational complexity of the algorithm was dominated by the Gibbs sampler. Here, again the Gibbs sampler presents a significant computational expense, requiring greater than $10^7$ iterations for all but the smallest instances of MAXSAT. On this problem however the computational time required to calculate the Markov Network parameters dominates the run time of the algorithm. The system of equations forms a matrix of considerable size: for a typical 100 bit MAXSAT problem there are 100 univariate terms, around 1100 bivariate terms and 430 trivariate terms. With an unknown alpha value for each term the system of equations has over 1600 unknowns. We have found by experimental analysis that the model is poor enough that it is extremely unlikely to yield an optimum unless there are slightly more equations than unknowns. The number of equations is the same as the population size, so in the experiments here we used a population of the number of terms multiplied by 1.1 – resulting in n approximating to $\sqrt{3\times10^6}$ . SVD is known to have a computational complexity of $O(n^3)$ for an nxn matrix.

# 5. FITNESS PREDICTION

## 5.1 Aims

Given the algorithm overheads we have described, it would appear that our initial approach is unlikely to be the best use of fitness modelling. In our previous publications, the advantage gained by DEUM over other EDAs has been its use of fitness modelling. The aim of this experiment is to demonstrate the fitness modelling capability of the Markov Network as a basis for further work.

## 5.2 Fitness Modelling and Prediction

The unique advantage which is gained by using this approach is that rather than building a distribution of highly fit individuals, DEUM models the fitness function directly. In addition to sampling model to find an optimum we can also use it to predict the fitness of individuals.

Predicting the fitness of an individual is a simple task given a previously constructed model. The bitstring is encoded as before so that for each $x_i$, 0 is coded as -1 and 1 remains unchanged. These values are substituted into the energy function (3) to give a predicted energy *U(x)* for the individual. The predicted fitness can then be calculated thus:

$$f(x) = e^{-U(x)} \quad (7)$$

Here, we use this as a measure of quality of fitness models being constructed. This concept also has a number of possible applications; for problems where the fitness function is expensive to compute the model could then be used in place of calls to the fitness function.

## 5.3 Method

The procedure for this experiment is identical to the first part of the previous experiment. A random population is generated and used to build a model. Subsequently, a second random population equal in size to the starting population was generated and evaluated. The model was then used to predict the fitnesses for this population, and the product moment correlation coefficient [7] between the predicted and true sets of fitnesses was calculated. We call this figure the *fitness prediction correlation.*

## 5.4 Results

The results in Table 2 are again sorted by MAXSAT problem size (PS) in the first column. The following two columns are the average number of fitness evaluations (FE) and accompanying standard deviation (FE-SD) required in evaluating the first population. That is, the number of evaluations required to build the model. The final two columns are the average fitness prediction correlation (FPC) and the corresponding standard deviation (FPC-SD).

Correlation coefficient values run from -1 (perfect negative correlation) to 1 (perfect positive correlation). Values greater than 0.9 typically indicates a strong linear relationship between two sets of data [7]. It can be seen from Table 2 that the correlation between true fitness values and those predicted using the model is considerably higher than this.

**Table 2: Fitness Prediction Capability of Model**

| PS | FE | FE-SD | FPC | FPC-SD |
|---|---|---|---|---|
| 20 | 284 | 5 | 0.9970 | 0.0011 |
| 50 | 849 | 11 | 0.9984 | 0.0004 |
| 75 | 1341 | 10 | 0.9988 | 0.0002 |
| 100 | 1834 | 10 | 0.9992 | 0.0001 |
| 125 | 2336 | 12 | 0.9993 | 0.0001 |
| 150 | 2829 | 13 | 0.9994 | 0.0001 |

## 5.5 Analysis

The results show that the algorithm is able to accurately predict the fitnesses of completely random individuals. This indicates that

DEUM is able to closely model the fitness function after a relatively small number of samples of it.

Firstly this reinforces our thought in section 4.4.1 that the poor algorithm success rate was caused by the nature of the Gibbs sampler. Given the accuracy of the fitness models being constructed it is feasible that by altering the sampling scheme to run for considerably longer with a considerably slower cooling rate an optimum could be found in every case. However, the run time of the sampling algorithm is likely to render this approach impractical.

This suggests a new approach to consider in future work. Fitness prediction could be used to bias traditional genetic operators such as crossover and mutation. Some work has already been done in this area using different probabilistic models – an example is the guided mutation operator described in [14].

# 6. CONCLUSIONS AND FUTURE WORK

This study has shown that the DEUM fitness modelling approach can be extended to more complex problems than the bivariate and univariate problems previously investigated. A Markov Network can be built and sampled to produce an optimal individual. It can also be used to predict the fitness of individuals instead of calling the real fitness function. The experiments here demonstrate that this can be achieved with a small number of function evaluations relative to other algorithms.

However, this is achieved at a high cost in terms of algorithm overhead. In earlier experiments, DEUM gained a significant advantage over other algorithms through its use of fitness modelling. With increasing problem complexity the number of terms in the energy function increases which leads to an $O(n^3)$ increase in model build time. The time required by the sampler also increases considerably with as problem complexity grows. Further investigation is required into the effect of using models which do not perfectly match the problem structure, which could have a significantly smaller number of terms in their energy function. Limiting the model structure to a particular degree of complexity may be one way to achieve this. Indeed, this approach has been taken in a number of algorithms employing Bayesian networks and shown to be successful [6]. This will also require some degree of structure learning or approximation to be effective. We will need to investigate the trade-off between maximum fitness modelling accuracy and efficient model construction.

While the model is still able to model fitness – even to a less accurate degree than that described in this paper – it may be used to guide traditional genetic operators. Our immediate consideration for future work is to investigate the use of hybrid operators. This could well lead to similar or better problem solving ability with reduced computational cost.

# 7. REFERENCES


[1] BROWN, D.F., GARMENDIA-DOVAL, A.B. AND MCCALL, J.A.W. 2002. Markov Random Field Modelling of Royal Road Genetic Algorithms. In *Selected Papers from the 5th European Conference on Artificial Evolution*, Springer-Verlag, 65-76.

[2] HOOS, H.H. AND STÜTZLE, T. 2000. SATLIB: An Online Resource for Research on SAT. In *SAT 2000*, I.P. GENT, H.V. MAAREN AND T. WALSH, Eds. IOS Press, 283-292.

[3] JOHNSON, D.S. 1973. Approximation algorithms for combinatorial problems. In *STOC '73: Proceedings of the fifth annual ACM symposium on Theory of computing*, Austin, Texas, United States, ACM Press, New York, NY, USA, 38-49.

[4] KINDERMANN, R. AND SNELL, J.L. 1980. *Markov Random Fields and their Applications*. AMS Press, Providence, RI, USA.

[5] KIRKPATRICK, S., GERLATT, C.D. AND VECCHI, M.P. 1983. Optimization by Simulated Annealing. *Science* 220, 671-680.

[6] LARRAÑAGA, P. AND LOZANO, J.A. 2002. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston.

[7] LUCEY, T. 1984. *Quantatitive Techniques: An Instructional Manual*. D. P. Publications, Eastleigh, Hampshire, UK.

[8] PELIKAN, M., GOLDBERG, D.E. AND LOBO, F. 1999. *A Survey of Optimization by Building and Using Probabilistic Models*. Technical Report IlliGAL Report 99018, University of Illinois at Urbana-Champaing

[9] PELIKAN, M. AND GOLDBERG, D. 2003. Hierarchical BOA solves Ising spin glasses and MAXSAT. In *Proceedings of the Genetic and Evolutionary Computation Conference 2003* (GECCO-2003), 2003, Springer-Verlag, 1271-1282.

[10] PRESS, W.H., FLANNERY, B.P., TEUKOLSKY, S.A. AND VETTERLING, W.T. 1986. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK.

[11] SHAKYA, S., MCCALL, J. AND BROWN, D. 2004. Updating the Probability Vector using MRF Technique for a Univariate EDA. In *Proceedings of STAIRS 2004*, IOS Press, 15-25.

[12] SHAKYA, S.K. 2006. *DEUM: A framework for an Estimation of Distribution Algorithm based on Markov Random Fields*. PhD Thesis, The Robert Gordon University, Aberdeen, UK.

[13] SHAKYA, S.K., MCCALL, J.A.W. AND BROWN, D.F. 2006. 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, 16-21 July 2006, IEEE Press,

[14] ZHANG, Q., SUN, J. AND TSANG, E. 2005. An evolutionary algorithm with guided mutation for the maximum clique problem. *IEEE Trans. Evolutionary Computation*, vol 9, issue 2. Pp 192-200.