Markov Random Field Modelling of Genetic Algorithms

Evaluation of Research

Siddhartha K Shakya
School of Computing, St Andrew Street

February 16, 2004

Abstract

The Proposed project *Markov Random Field Modelling of Genetic Algorithm* aims to introduce **MOA**: ***M**arkov Random Field **O**ptimization **A**lgorithm.* The idea is based on the use of Markov Random Field models as a probabilistic model capturing the interdependency between variables in the GA chromosome for better evolution of a solution. This report is a self evaluation of our research to date. We start by giving an introduction to our research area, and justifying the initial research question. Next, we evaluate the literature review undertaken during the research study. The justification of the adopted research methodology, and the data analysis technique are also discussed. Finally, an evaluation of the findings to date is presented and a plan of the future work is shown.

Contents

List of Figures

Chapter 1

Introduction

A general **optimization problem**[29] is to find the optimum or near optimum solution from a specified set of possible candidate solutions using some measure for evaluating each individual solution. Algorithms to solve such problems are called **optimization algorithms**. Depending upon the type of applicable problem domains, optimization algorithms can be divided into two categories. First are the classes of algorithms that are customized to work with specific types of problems, and are called specialized optimization algorithms. Second are the classes of optimization algorithm that can work in a wide variety of optimization problem and are known as general optimization algorithms. Our research will focus on a class of general optimization algorithms known as **Genetic algorithm** (**GA**) [13].

Genetic Algorithms (GAs) are a class of optimization algorithm capable of evolving the solution by using the techniques of natural *selection* and *variation* [28, 29]. A GA encodes solutions as a string of symbols or chromosome, and selects and recombines a population of chromosomes to evolve better solutions. Both the Selection and Variation (also known as Recombination) operators play a very important role in evolution. However, the traditional crossover and mutation approach of variation in GAs was soon found to be limited for many real-life problems, particularly on those where, important partial solutions (also known as Building Blocks) are loosely distributed in the chromosome. Most of the research in the GA community was focused on the modification of the variation operators to improve the GA performance.

1.1 The Knowledge Gap

In recent years, an alternative method of variation was proposed where the traditional crossover and mutation operators are replaced by a two-stage process. 1) Build a probabilistic model encapsulating the interdependency between genes in a chromosome. 2) Use this model to estimate the probability distribution of genes in the population, and sample them to generate a new population. Algorithms using this approach are called **Estimation of Distribution Algorithms** (**EDAs**) [25]. EDAs are found to perform better in problems where a traditional GA fails to give satisfactory performance. However, the success and failure of an EDA solely depend upon the effectiveness of probabilistic model used.

Independently from the above work, in [4][1], the use of **Markov Random Field** (**MRF**)[19] theory for modelling chromosome fitness in GAs has been proposed. Markov Random Field (MRF) modelling technique has been widely used in image analysis and computer vision fora wide variety of detection problems. In [4], a GA has been transformed into the well known labelling problem (in Image analysis), and by using MRF theory, a relationship between chromosome fitness and gene interaction has been derived. The probabilistic model encapsulating this relationship is known as a Markov Random Field Model of the Fitness Function [4] or simply a MRF Model. So far MRF models have been used to predict child fitness. However, no attempt has been made to design "GA-alike" algorithms incorporating MRF models.

1.2 Justification of Research Hypothesis/Question

We notice that MRF models could be used as probabilistic models for EDAs. We hypothesise that the use of a MRF model will improve the evolution process resulting in better performance of EDAs. An intuitive justification of this hypothesis is as follows. Let us recall the selection and variation in evolution. Traditionally, in the evolution process, the involvement of chromosome fitness has been limited to selection operators, and was excluded from the variation operators. Both GAs and EDAs used chromosome fitness to select better chromosomes. However, neither crossover/mutation in GAs nor probabilistic models in EDAs use the information contained in chromosome fitness in their operation. Designing an *EDA-like* algorithm using MRF models will utilise this extra information for variation. Thus it may improve the overall performance of the

---

[1] Research work has been conducted at School of Computing in The Robert Gordon University

evolution of a high quality solution.

1.3 The Research Project

Our Research Project is called the *Markov Random Field Modelling of Genetic Algorithm*, and is motivated from the previous research work done in the school of computing [4]. We aim to introduce MOA: The ***M**arkov Random Field **O**ptimization **A**lgorithm.* MOA also uses a probabilistic approach to variation, and can be seen as a novel EDA. MOA aims to introduce a Markov Random Field (MRF) model as a novel approach of building probabilistic models which uses both gene interaction in a chromosome, and its fitness to give a probabilistic model of the fitness function. We hypothesize that the estimation of the probability distribution using a MRF model becomes more accurate, and results in better optimization.

Chapter 2

Evaluation of Literature Review

The method of validating the literature review in theoretical computer science is to reconstruct the experiments done by other researchers, and validate the results on your own. The aim is to test and validate the algorithms proposed by previous researchers i.e reproduce their results. Our research study bears a strong theoretical nature. As the core of our research question comes from the field of GAs, our initial literature review was on the latest progress in this area. However, as the process progressed we substantially narrowed the wide area of GAs to the probabilistic model building GAs. Thus, we identified the two main areas relevant to our research question.

- Estimation of Distribution Algorithms (EDA) together with Bayesian networks and Search heuristics;
- Markov Random Field (MRF) modelling techniques which includes MRF theory, MRF in optimization, least square fit techniques and Maximum likelihood estimation techniques.

The literature was evaluated according to its relevance to the theme of our project, particularly with its relevance to the two areas named above. In addition to this, the reliability of the source of the literature and authority was also taken into account. Furthermore, extra attention was given to the quality of the experiments and results published. Finally a higher priority was given to works published more recently, but earlier impportant works could not be ignored.

2.1 The process

EDAs are a novel concept in the GA field with less than 12 years of history. Since their emergence, an increasing amount of research has been done in this area. As the literature review progressed, I came to know about the leading researchers and research institutions involved in EDAs and also about the future direction of this research.

First proposed by Baluja [1], the simplest form of this class is called Population Based Incremental Learning (PBIL), and several different variants of EDAs have been proposed to date. Most of the early EDAs were focused on the binary representation of a solution vector, i.e focused on the *discrete* problem domain. Most of them have been later modified to work on the *continuous* domain, and published with different names. Probabilistic models in the continuous domain are more complicated, since the joint probability density function (jpdf) is calculated using more complex mathematical methods when compared to the simple marginal or conditional probability used in discrete EDAs. For the initial stage of our research, we focused our literature review on discrete EDAs. We adopted the method used in [28] and categorise EDAs in to three different groups: **Univariate**, **Bivariate** and **Multivariate** according to the probability model they use.

While our initial goal for the research study was to establish the Markov Random Field Optimization algorithm (MOA) covering all three variants of the probabilistic models, we started to realise that the first step towards MOA is to establish the univariate case being the simplest. Univariate EDAs do not consider the dependencies between variable in an individual i.e. considers building blocks of order one. So the joint probability distribution becomes simply the product of the univariate marginal probabilities of all of the variables in an individual. Due to its simplicity, the algorithm in this category are computationally very efficient, and perform excellently on linear problem such as function optimization where the variables are not significantly interdependent. However, these algorithms fail on complex problems where the variables interact with each other. Population based Incremental Learning (PBIL) [1], Univariate Marginal Distribution Algorithm (UMDA) [25] and Compact Genetic Algorithm (cGA) [11] all use a univariate model of probability distribution.

We further our literature review to Bivariate EDAs. They consider pair-wise dependencies between variables in a chromosome, i.e., they consider the building blocks of order two. Similarly the probability model becomes more complex than that of the univariate model, and takes the form of a probabilistic network between variables. This class of algorithm performs better in problems with pair-wise interaction between the variables. However, it fails on problems with multiple variable interactions. Mutual Information Maximization for input clustering (MIMIC) [6], Combining Optimizers with Mutual Information Trees (COMIT) [2], and Bivariate Marginal

Distribution Algorithm (BMDA)[32] all use bivariate model of the probability distribution.

EDAs that consider the interdependency between variables of order more than two can be seen as Multivariate EDAs. The probability network representing the interdependency of variables obviously becomes more complex, and the computation time to construct such a network hugely increases, making it almost impossible to search through all possible models. Due to its simplicity, most of the algorithms in this class uses *Greedy heuristics* to search for a good model, however the greedy heuristic does not always guarantee accuracy. Some other complex search algorithms have also been successfully used for this purpose, and lots of current EDA research is focused on finding good heuristics. Extended Compact Genetic Algorithm (ECGA) [10], Factorised Distribution Algorithm (FDA)[21, 22, 23], Bayesian Optimization algorithm (BOA) [26, 27, 29, 30, 31], Learning Factorised Distribution Algorithm (LFDA) [23, 24], Estimation of Bayesian Network (EBNA)[9, 16, 17, 18] all use a multivariate model of probability distribution.

The literature review on EDAs gave us an up-to-date knowledge of on-going research, and the future direction it is heading in. As our research also deals with probabilistic modelling using MRFs, we further our literature review on MRF modelling techniques. MRFs [19] are a generalisation of Markov Chains. They have been applied for many years to the analysis of images, particularly in the detection of visual patterns or textures. The comprehensive collections of works on different theoretical and application area of MRF theory can be found in [3]. Most of the literature on the application of MRF theory is focused on image analysis. However, MRF are also used in the optimization process and incorporated in an optimization algorithm (mostly customized). Less evidence has been found on the use of MRF theory in GA optimization. The work presented in [4] is the closest relating GAs with MRF, and is an initial motivation for our research question. They present an interesting method of probabilistic modelling of chromosome fitness which they call MRF modelling of Fitness function. They provide a theory where any GA fitness function can be modelled in terms of an MRF model.

2.2 Purified Research Question

From our survey on EDAs we note that the EDAs for the discrete domain proposed so far solely adopt marginal or conditional probability to construct the probabilistic model to estimate the distribution of genes in the population. However, no evidence on the investigation of alternative methods for constructing probabilistic models has been found. Rather, most of the research on

EDAs is focused on finding a good way to build the dependency network, i.e, on finding a good search heuristic.

The outcome of the survey done on EDAs concludes that the gap in the research on more sophisticated ways to build a probabilistic model clearly needs attention. Again, from the survey done with MRF modelling techniques, we note the fact that very little or no effort has been made to use a MRF model as a probabilistic model for EDA. Also no effort has been made to represent the multivariate dependency between variables (in the context of a GA) as a MRF model.

The literature review process was very beneficial for a deep understanding of our initial research question. At the end of which, we were forced to decompose our initial research question into three separate parts:

1. How can a MRF model be incorporated into an EDA?
2. Will the use of a MRF model as the probabilistic model improve the performance of EDA?
3. How can bivariate and Multivariate dependencies between variables be effectively represented as a MRF model? (Representing univariate dependencies is straightforward.)

Furthermore, our literature review gave us a clear picture of the future road map to tackle the above questions. As an outcome of literature review, we were able to further divide our primary objectives of establishing Markov Random Field Optimization Algorithm (MOA) into three parts:

1. Design and implement an algorithm using a simple univariate model of the fitness function, which we name the ***U****nivariate* ***M****arkov Random Field* ***O****ptimization* ***A****lgorithm* (UMOA).
    - Find a way to transform the MRF parameters into a probability distribution for the direct sampling of child population.
    - Perform a workflow analysis of UMOA.
    - Apply UMOA to different optimization problems such as the Travelling Salesman Problem (TSP)[8], and the Graph Coloring Problem (GCP)[7].
2. Extend UMOA to work with bivariate and multivariate MRF models, and establish a general framework for the Markov Random Field Optimization algorithm (MOA).
    - Perform a detailed study on the existing techniques for finding the dependency network.

    - Research different ways of representing multivariate interaction as MRF models.

3. Perform a workflow analysis of MOA.

Chapter 3

Evaluation of Research Methodology

In this chapter we discuss the research methodology used for our project. Although the project is in its initial stage, some experimental analysis has already been performed. To explain the adopted research methodology more clearly, we follow the approach taken by Joseph McGrath in [20], and use a strategy circumplex [Figure 3.1] to represent the available research methodologies.

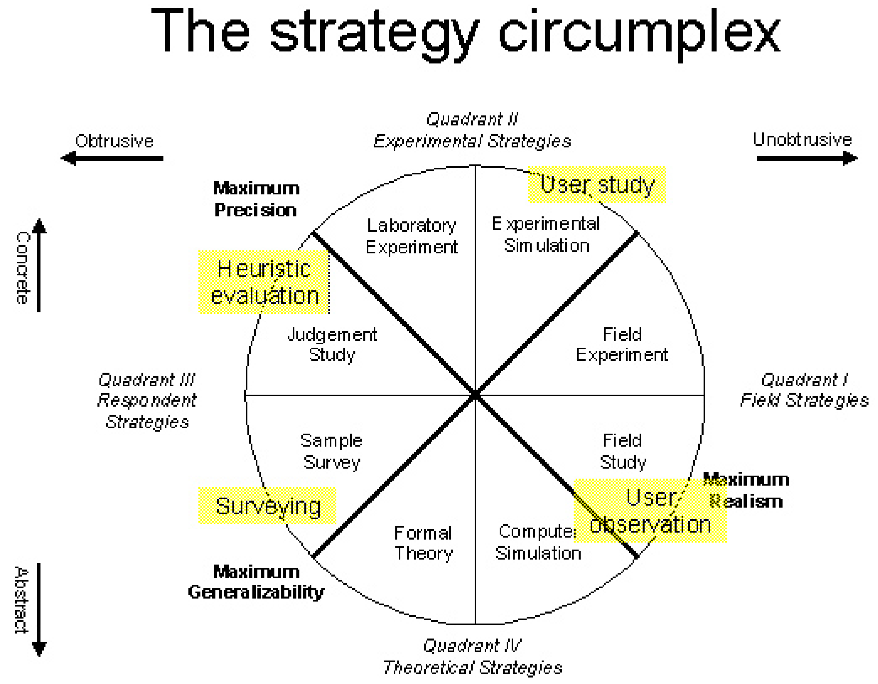

Figure 3.1: The strategy circumplex showing how the different data gathering methods work.

We now briefly describe each of the data gathering methods.

Laboratory Experiment: This is a very well-controlled method, i.e., what you will give to the user, and what you are expecting from them is very well-defined. Users are mainly in the lab and given the task or the system, and their response is recorded.

Experimental Simulation: This is less controlled than the laboratory experiment method. It is mostly conducted outside the lab, where the real-life environment is simulated. Users are given the certain task and their response is observed.

Field Experiment: This is an even less controlled method than Experimental Simulation. The users are given a certain task in the real-life environment, for example, in their work-place and their response is observed. Usually results are in the form of filled Question Answer sheets, and the data obtained are of a quantitative nature.

Field Study: This type of method is used where strict user observation without interruption is needed. The method is applied in the real-life environment, and the task is the usual day-to-day work of the user. The monitoring or their working style is evaluated.

Computer Simulation: This is a method where no human is involved, and is totally conducted using computers. This type of data collection method is useful for comparing the performance of two computational processes or algorithms. The data are collected from computer simulation only, and is usually quantitative in nature.

Formal Theory: This is a theoretical approach to the data collection. Data collection is totally dependent upon existing theories, for example, Designing an algorithm. No experiment is conducted for data collection purpose. Example data: bench mark problems that are already proven to be the fundamental data for conducting an experiment to evaluate a particular type of algorithms.

Sample Survey: This is a survey or the collective feedback strategy of data collection. For example, a list of questions with possible answers is prepared and distributed to the population of users, and later their answers are collected. This is a typical survey of the attitude of users from different environments.

Judgement Study: This is a more qualitative data collection method. Data will be in the form of report/text from a user or system. The response of an expert can also be placed in this class. This type of method is usually seen as the evaluation of a user's opinion on a specific topic.

3.1 Justification of Research methodology

The experiments in our research study are usually based on testing and analysing the performance of algorithms. The algorithms being tested are of pure simulative nature, and no human computer interaction is needed. Therefore the testing of each algorithm is achieved entirely by computer simulations, an so our research methodology has a strong theoretical nature, and falls in the Fourth quadrant of the strategy circumplex [Figure 3.1]. To explain it more precisely, let us recall our research objective and put the following scenario.

To address the first objective set out in the previous chapter, we design an algorithm using the Univariate MRF model, which we call Univariate MRF Optimization Algorithm (UMOA). We choose different benchmark problems of order one dependency, like **OneMax** and simple **Function Optimization**, to test UMOA. These problems are proven to be ideal for univariate EDAs[29], and widely adopted in the GA literature. The experimental input data are generated by computer simulation, i.e, a random number generator is used to get the initial population of chromosomes. The numerical performance is then observed by running UMOA on these data sets for a specific problem.

Random data generation, used in above scenario, is one of the most highly-used research methodologies in the theoretical computer science field. It avoids the algorithm being focused on a single situation and tests it against a wide range of random situations. As UMOA is designed to work in the discrete domain, the data generated will be of a discrete type, more precisely the binary numbers used by UMOA fall in the category of *Nominal data* rather than *Ordinal data*. An analytical scenario similar to the one above will be used to address our remaining objectives and sub-objectives. Therefore, our research methodology will be of a purely quantitative type.

To date some experiments have been done to address our first objective, and encouraging results have been obtained. To summarise, this section we note that our research is solely of a quantitative nature, and the data are of a quantitative type. Further, all of the experiments and data collection is done by the use of computers. The human involvement in data collection is excluded so the research strategy adopted for my area falls in the fourth quadrant of strategy circumplex, i.e, the theoretical strategy will be adopted as the research methodology.

Chapter 4

Evaluation of data analysis techniques

There exist many different data analysis techniques, such as chi square test, regression analysis, mean, median, etc. Apart from these traditional techniques, research in theoretical computing tends to use different *comparative analysis* techniques which mainly focus on performance of their proposed algorithm in comparison to the existing state of the art algorithms. Although the comparative analysis does say that the performance of one algorithm is better than another, most of the time it does not really explain why it is so. Alternatively in [14] a different analysis technique called the *scientific technique* is proposed. Scientific technique as described by the author is more controlled in the sense that it covers the effect of different situational/parameter changes on the performance of an algorithm and is focused on building a better model for testing. Nonetheless, our research still uses a comparative analysis, however we must stress the fact that running different comparative analysis for the same data set will partially satisfy the effect of different conditional changes on the performance of the algorithm, and therefore satisfy the concept of scientific data analysis techniques. Furthermore it appears to be the most widely accepted data analysis technique in the GA and EDA communities.

The rest of the sections will look at some of the comparative data analysis techniques we have used so far.

4.1 Speed vs Problem Size

The observation of speed vs the problem size is one of the most highly used data analysis techniques in algorithm performance testing. Even more frequent use of this can be seen in different GA and

EDA literatures [12, 29, 32]. This type of analysis helps to identify the change in performance of an algorithm as the problem size changes. The performance is usually measured in terms of time taken to find the solution. In the case of GAs and EDAs, the time is usually measured in terms of the number of function evaluations, i.e, the number of chromosome fitness calculations. Due to the stochastic nature of GAs and EDAs, the experiment is usually carried out for N number of different random problem instances for each problem size, and the average of all the runs is record for plotting. N varies for different situations but it is usually greater than 30 and less than 1000.

Figure 4.1 is an example of speed vs problem size analysis that compares the performance of UMOA with the existing Univariate EDAs and GAs . The average number of fitness evaluations to find optimum is plotted against 30 to 180 sized OneMax problems for four different algorithms.

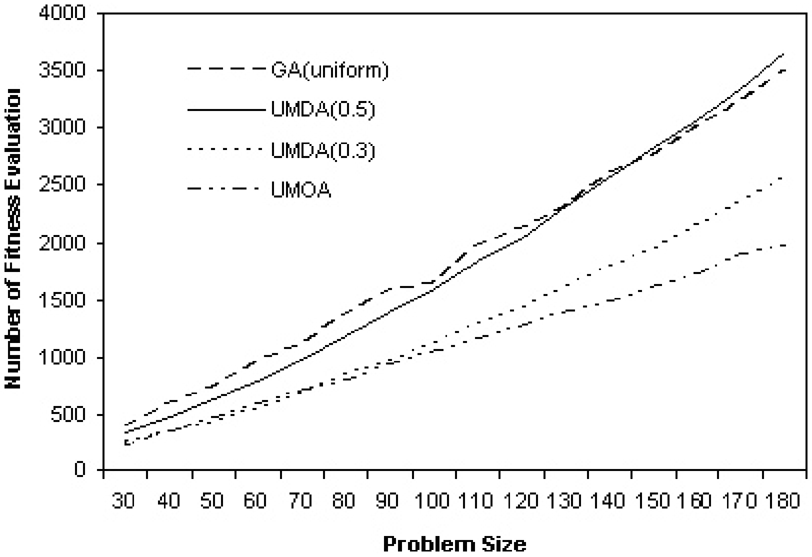

Figure 4.1: Average number of fitness evaluations taken by four different algorithms for 30 to 180 sized OneMax problems.

4.2 Run Length Distributions

Run Length Distribution (RLD) [15] is another type of data analysis technique that is used more often to study the behaviour of a stochastic algorithm on a specific problem instance. RLDs are a very powerful technique used to understand the dynamic behaviour that is usually observed in stochastic algorithms like GAs and EDAs. It helps us to analyse the reliability of the algorithm on a particular instance of a problem, and figure out the range of time needed to achieve a specific percentage of successful runs.

Figure 4.2 shows an RLD for four different algorithms, i.e, the cumulative percentage of successful runs is plotted against the number of function evaluations needed to achieve that level of success. We can see that apart from UMDA the rest of the algorithms are very reliable for the tested problem. Furthermore, plotting them together gives us a comparison on their performance in terms of the time they take to find the solution. For example, with UMOA, 50 percent of runs found the optimum within 1700 function evaluations, in comparison to 2200 function evaluations of PBIL.

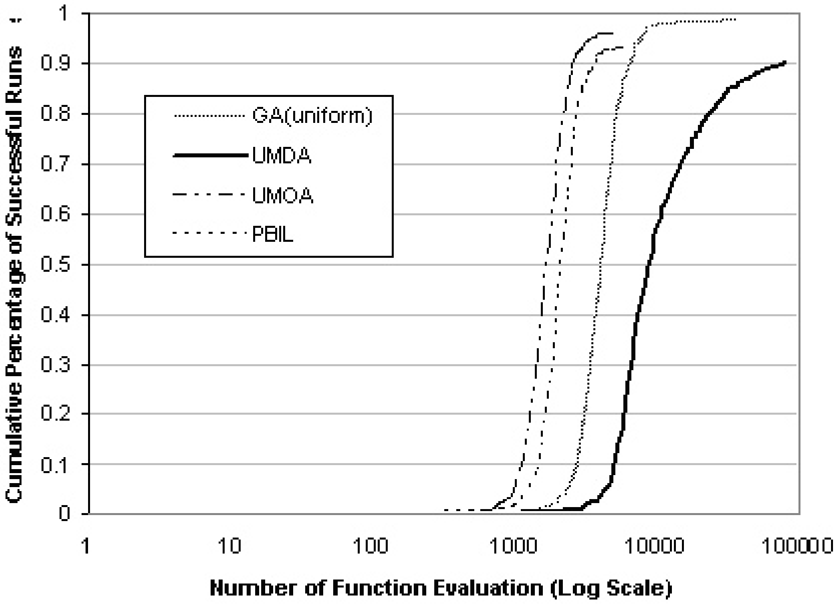

Figure 4.2: Experimental results showing, for each algorithm running on the 20 bit Schaffer f6 function, the cumulative percentage of successful runs that terminated within a certain number of function evaluations.

Chapter 5

Evaluation of Findings

Although our research is in its initial stages, we have so far conducted a few sets of experiments to address the first objective stated in Chapter 2. As we stated earlier, UMOA has been designed and tested on different problems. The results are very encouraging and more experiments are planned in near future. This section will discuss the experimental results found to date and evaluate them.

5.1 The results and justification

UMOA uses a univariate MRF model and thus falls in the class of Univariate EDAs. We compare UMOA with other state of the art univariate EDAs, and GAs on three different problems. These are:

1. OneMax;
2. Schaffer f6 Function Optimization;
3. Composed Trap function of order 5.

Each of the problems has a different structure. The OneMax function is a simple linear problem that is decomposable of order one, and therefore is an ideal problem for univariate EDAs. It has been shown that UMDA works very well on this problem, even with a very small selection size. Figure 4.1 in the previous chapter shows the experimental results comparing UMOA with three other algorithms on the OneMax problem. It illustrates that the performance of UMOA is comparable with other univariate EDAs for small problems. However, as problem size grows, UMOA significantly outperforms the rest of the algorithms.

The f6 function proposed by Schaffer et al [5] is an interesting function, which has been frequently used to evaluate the performance of GAs. A simplified version of it was used to analyse the performance of UMOA. The interesting features of this function are that it has lots of local sub-optimal solutions, and a single global optimal solution. So the hill climber algorithm will rapidly become trapped in one of the sub-optimal solutions. Figure 4.2 in the previous chapter shows the experimental results comparing UMOA with three other algorithms on Schaffer f6 function optimization problem. Again UMOA outperformed the rest of the algorithms, further providing evidence that using a MRF models as a probabilistic model improves evolution.

A composed trap function of order 5 falls in the class of deceptive problems. Its important feature is that a block of similar bits less than 5 in a chromosome misleads the algorithm to a local optimum. The purpose of this experiment is to show that, for trap functions, UMOA also has the same problem as other univariate EDAs and GA with uniform crossover, which is to converge to a local optimum. Our experiment shows that neither the GA(uniform), UMDA, PBIL nor UMOA could find the optimum, even with a population size of 15000. It is obvious that, to tackle this problem with a probabilistic model, we should consider the dependencies between the variables of order 5, and thus we can not solve it by using algorithms that assumes order one dependency. This experiment suggests the need for an algorithm using a multivariate MRF model for this type of problem.

Chapter 6

Future Work

At this stage of our research, we are close to achieving our first objective, which is to design and implement the UMOA. Our research so far is very promising, and is subsequently building a clear pathway for further research. This section lists the further work that we aim to finish within the time frame of our research study.

- Find a way to transform the MRF parameters into a probability distribution for directly sampling a child population.
- Perform a workflow analysis of UMOA.
- Apply UMOA to different optimization problems like TSP[8] and GCP[7].
- Extend UMOA to work with a multivariate MRF model, and establish the general framework for the Markov Random Field Optimization algorithm (MOA).
    1. Perform a detailed study of the existing techniques for finding dependency networks.
    2. Research different ways of representing multivariate interactions as MRF models.
- Perform a workflow analysis of MOA

To conclude this report we state that the results obtained from our research to date are very encouraging. We aim to further our research by designing extended versions of UMOA, and similar promising results are expected in near future.

References

[1] Shumeet Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning,. Technical Report CMU-CS-94-163, Pittsburgh, PA, 1994.

[2] Shumeet Baluja and Scott Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In *Proceedings of the 1997 International Conference on Machine Learning*, 1997.

[3] Charles A. Bouman. Markov random fields and stochastic image models, 1995.

[4] J. Mccall D. Brown, A. Garmendia-Doval. Markov random field modelling of royal road genetic algorithms. *Evolution Artificielle 2001*, page 12, 2001.

[5] Lawrence Davis, editor. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York, 1991.

[6] Jeremy S. de Bonet, Charles L. Isbell, Jr., and Paul Viola. MIMIC: Finding optima by estimating probability densities. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 424. The MIT Press, 1997.

[7] G. Marble D.W. Matula and J. D. Isaacson. Graph coloring algorithms, 1972.

[8] A. Rinnooy-Kan E. L. Lawler, J. K. Lenstra and D. Shmoys. The traveling salesman problem: A guided tour of combinatorial optimization, 1985.

[9] R. Etxeberria and P. Larrañaga. optimization with bayesian networks. In *Proceedings of the Second Symposium on Artificial Intelligence. Adaptive Systems. CIMAF 99.*, page 332339, Cuba, 1999.

[10] G. Harik. Linkage learning via probabilistic modeling in the ecga, 1999.

[11] G. R. Harik, F. G. Lobo, and D. E. Goldberg. The compact genetic algorithm. *IEEE-EC*, 3(4):287, November 1999.

[12] Tomoyuki Hiroyasu, Mitsunori Miki, Masaki Sano, Hisashi Shimosaka, Shigeyoshi Tsutsui, and Jack Dongarra. Distributed probabilistic model-building genetic algorithm. In *Genetic and Evolutionary Computation – GECCO-2003*, volume 2723 of *LNCS*, pages 1015–1028, Chicago, 2003. Springer-Verlag.

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

[14] J. Hooker. Testing heuristics: We have it all wrong, 1996.

[15] Holger H. Hoos and Thomas Stutzle. Towards a characterisation of the behaviour of stochastic local search algorithms for SAT. *Artificial Intelligence*, 112(1-2):213–232, 1999.

[16] P. Larrañaga, R. Etxeberria, J. Lozano, and J. na. Optimization by learning and simulation of bayesian and gaussian networks, 1999.

[17] P. Larrañaga, R. Etxeberria, J. Lozano, and J. na. Combinatorial optimization by learning and simulation of bayesian networks, 2000.

[18] Pedro Larrañaga and Jose A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation.* Kluwer Academic Publishers, 2001.

[19] S. Z. Li. *Markov random field modeling in computer vision.* Springer-Verlag, 1995.

[20] Joseph E. McGrath. Methodology matters: doing research in the behavioral and social sciences. pages 152–169, 1995.

[21] Heinz Mühlenbein. Evolutionary algorithms and the boltzmann distribution. In *Foundations of Genetic Algorithms (FOGA2002).*

[22] Heinz Mühlenbein and Thilo Mahnig. Convergence theory and application of the factorized distribution algorithm. *Journal of Computing and Information Technology*, 7(1):19–32, 1999.

[23] Heinz Mühlenbein and Thilo Mahnig. FDA - A scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(4):353–376, 1999.

[24] Heinz Mühlenbein and Thilo Mahnig. Mathematical analysis of evolutionary algorithms for optimization. In A. Rodriguez, M. Ortiz, and R. Santana, editors, *Proceedings of the Third Internatinal Symposium on Adaptive Systems*, pages 166–185, Cuba, 2001. Institute of Cybernetics, Mathematics and Physics (ICIMAF).

[25] Heinz Mühlenbein and Gerhard Paass. From recombination of genes to the estimation of distributions: I. binary parameters. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature – PPSN IV*, pages 178–187, Berlin, 1996. Springer.

[26] M. Pelikan and D. E. Goldberg. Research on the Bayesian Optimization Algorithm. Technical Report 2000010, Illinois Genetic Algorithms Lab, UIUC, Urbana, IL, 2000.

[27] M. Pelikan, D. E. Goldberg, and E. Cant'u-Paz. BOA: The Bayesian Optimization Algorithm. In W. Banzhaf et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO99*, volume I, pages 525–532, San Fransisco, CA, 1999. Morgan Kaufmann Publishers.

[28] M. Pelikan, D. E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. Technical Report 99018, Illinois Genetic Algorithms Lab, UIUC, Urbana, IL, 1999.

[29] Martin 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.

[30] Martin Pelikan and David E. Goldberg. Hierarchical problem solving by the bayesian optimization algorithm. IlliGAL Report No. 2000002, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, 2000.

[31] Martin Pelikan and David 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.

[32] Martin Pelikan and Heinz 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.