Bio-control in Mushroom Farming using a Markov Network EDA

Yanghui Wu, John McCall, Paul Godley, Alexander Brownlee, David Cairns and Julie Cowie

Abstract— **In this paper we present an application of an Estimation of Distribution Algorithm (EDA) that uses a Markov network probabilistic model. The application is to the problem of bio-control in mushroom farming, a domain which admits bang-bang-control solutions. The problem is multi-objective and uses a weighted fitness function. Previous work on this problem has applied genetic algorithms (GA) with directed intervention crossover schemes aimed at effective biocontrol at an efficient level of intervention. Here we compare these approaches with the EDA Distribution Estimation Using Markov networks ($DEUM_d$). $DEUM_d$ constructs a probabilistic model using Markov networks. Our experiments compare the quality of solutions produced by $DEUM_d$ with the GA approaches and also reveal interesting differences in the search dynamics that have implications for algorithm design.**

I. INTRODUCTION

ESTIMATION of Distribution Algorithms (EDA) are a sub-class of Evolutionary Algorithms that use information gained from a population of solutions to construct a probabilistic model. The model is sampled to produce successor populations, replacing the genetic operators, such as crossover and mutation, used in traditional genetic algorithms. The process of constructing and sampling a model can be more computationally intensive than using genetic operators and so must be justified by a more efficient or effective search. Another potential gain is that the models resulting from the search might contain valuable information about the search space, thus justifying additional computational expense.

In this paper we present an application of an EDA to bio-control in mushroom farming. The work is motivated by a general interest in control problems where the control is implemented as a time series of discrete interventions. An underlying system responds to the control and the search goal is to find a sequence of interventions that optimizes the system response. In this application, and in our previous work on chemotherapy optimization [6], [7], [9], each intervention carries a significant cost, and so there is a need to minimize the number of interventions.

In previous work on the bio-control application [2], [3], [4], we developed genetic algorithms (GA) with variable length encodings that used directed intervention crossover operators to control the number and distribution of intervention points. Here we apply the EDA Distribution Estimation Using Markov networks ($DEUM_d$) to the same problem and compare the two approaches.

In Section II, we explain and formulate the bio-control problem. Section III briefly describes previous work on this problem using directed intervention crossover GAs and explains the $DEUM_d$ algorithm and how itcan be applied to this problem. In Section IV we describe some comparative experiments the results of which are analyzed in Section V. Finally, we present our conclusions in Section VI.

II. BIO-CONTROL IN MUSHROOM FARMING

A. Background

When mushrooms are produced in commercial quantities, the quality and yield of the mushroom crop can be seriously damaged through infestation by sciarid flies. Sciarid fly larvae are known to feed on the mycelium in the casing layer of mushroom causing crop production to significantly decline. An important weapon in combatting sciarid fly is the use of the nematode worm, *Steinernema feltiae*, which feeds on sciarid larvae thus reducing the problem. Nematode worms are sold commercially for bio-control of sciarid flies on mushroom farms. The nematodes are applied to the crop by periodic spraying.

In [1] a dynamic mathematical model is developed that expresses the life cycle of Sciarid larvae in the presence of periodic spraying with nematode worms. In this model, nematodes are injected into the system at a series of discrete interventions points. The model consists of a series of coupled differential equations and can be implemented numerically using standard techniques.

The model consists of a series of populations, consisting of the 4 stages of the sciarid life cycle (egg, larva, pupa and adult) and the nematode worm. Each population is governed by a differential equation. There are transitions between sciarid populations representing the growth stages of the flies and there are interaction terms representing fatal interactions with nematode worms. Sciarids are damaging to the mushroom crop only in the larval stage (though adults lay eggs which can then develop into larvae) and so the main focus of bio-control is to minimise the number of larvae at any time.

Fenton et. al. developed optimisation objectives based on the number, $L(t)$, of sciarid flies in the damaging larval stage

Manuscript received December 15, 2007. This work was supported in part by the Chinese Government Scholarship Fund.

Y. Wu is with the Northwest A&F University, Yangling, China (e-mail: sxxawyh@hotmail.com).

J. McCall, is with the Robert Gordon University, Aberdeen, Scotland. (Corresponding author: telephone +44 (0) 1224 262700 e-mail: jm@comp.rgu.ac.uk).

P. Godley is with the University of Stirling, Stirling, Scotland (email: pgo@cs.stir.ac.uk).

A. Brownlee is with the Robert Gordon University, Aberdeen, Scotland (e-mail: sb@comp.rgu.ac.uk).

D. Cairns is with the University of Stirling, Stirling, Scotland (email: dec@cs.stir.ac.uk).

J. Cowie is with the University of Stirling, Stirling, Scotland (email: jco@cs.stir.ac.uk).

978-1-4244-1823-7/08/$25.00©2008 IEEE

of development present in the crop at time t. The problem admits a *bang-bang control* strategy, that is a fixed dose of nematodes is either applied or not applied at each of a series of potential intervention points during the period of mushroom cultivation. Spraying the crop with nematodes is both costly and time consuming and so there is a second objective that aims to spray efficiently, i.e. to minimise the number of sprayings required. This is clearly in conflict with the objective of reducing the larva population.

From [1], [2], a weighted objective function has been formulated as follows:

$$F(x) = \sum_{t=0}^{T} L(t) + NP \qquad (1)$$

Here x represents a control strategy, somehow encoding a series of intervention / non-intervention decisions at a series of evenly-spaced time steps $t =$ 0,1,…,T. A control strategy evaluates as $F(x)$ which sums the number of sciarid larvae present at each time step and finally adds on a penalty P for each of N interventions specified by x. The values of $L(t)$ at each time step are calculated by running the bio-control model with the specified control and recording the values of larva population produced.

Of particular interest here is the penalty applied for each intervention. This is due to the cost and inconvenience to the farmer of spraying mushrooms with nematodes. However costs related to application of controls are a more general feature of other applications. In particular we have encountered similar features in our previous work on chemotherapy design. Here fewer interventions are desirable because of toxic side-effects of therapeutic drugs and in clinical practice, to ease the difficulty of administering the treatment.

III. Evolutionary Algorithms for Bio-Control

Solutions to the mushroom bio-control problem will seek to minimise interventions whilst maximising the effect of the control on the sciarid larva population. Algorithms that can detect time points where the system response is particularly sensitive to intervention or a combination of interventions, will be able to use these to maximise the effectiveness of interventions and so reduce the overall number of interventions required.

In previous work, Godley et al. developed genetic algorithms for the bio-control problem that used directed intervention crossover operators [2], [3], [4]. The purpose of these operators is to explicitly seek to control the number and distribution of intervention points. The two operators developed, CalEB (Calculated Expanding Bin) and TInSSel (Targeted Intervention with Stochastic Selection), favour the reduction of intervention points and attempt to focus the location of interventions at time points where they seem to have most effect. CalEB and TInSSel operate on variable length encodings that contain interventions labeled as taking place at specific time points. Both operators use the number of interventions present in the parents to calculate the number required in the children, with CalEB utilising a "binning" approach to select the genetic material from the parents, whereas TInSSel contains an element of stochastic selection.

Godley et al.'s experiments compared CalEB, TInSSel and a standard Uniform Crossover (UC) [2]. These experiments used initialization schemes that controlled the maximum number of interventions allowed in randomly generated solutions in the starting population. This limit was increased through a series of runs to evaluate the effect each of the crossover approaches had on finding a solution. At the extreme, solutions were allowed with no limit on the number of interventions beyond the total number of possible intervention points. This corresponds to the realistic situation where little is known about the correct number and distribution of interventions. The experiments showed that directed intervention crossovers were robust in limiting the number of interventions used whereas a GA using UC was unsuccessful in focusing interventions.

Our motivation in this paper is to explore whether an EDA can be used to achieve a similar performance. The probabilistic model used by an EDA is sampled to produce good solutions and so should reflect the objective. We apply the univariate EDA Distribution Estimation Using Markov networks DEUM$_d$ [5], [8], which has previously been successfully applied to identifying optimized dosing schedules in chemotherapy [6], [7], [9]. This algorithm operates on bitstring solutions and so we adopt a natural encoding for the bio-control problem with one variable for each time step. A variable value of '1' represents an intervention at that time step and a variable value of '0' indicates no intervention.

DEUM$_d$ uses a fully-factorised Markov network distribution which models the fitness function according to the following relation:

$$-\ln F(x) = \left(\alpha_0 V(x_0) + \ldots + a_T V(x_T)\right)/T \qquad (2)$$

Here $F(x)$ is the evaluation of a solution x as described in (1) and $x_0,\ldots,x_T$ are the decision variables. $V(x_i)$ evaluates to $(-1)^{1-x_i}$. The coefficients α_i are the Markov network parameters and are determined by best fit to a set of selected solutions.

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 α 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 α values.

These determine the probabilistic model and new solutions can be sampled according to the marginal probabilities:

$$p(x_i = 1) = \frac{1}{1 + e^{2\alpha_i / T}} \qquad (3)$$

T is a temperature coefficient which is varied according to

a cooling scheme as the algorithm progresses through a series of generations.

The $DEUM_d$ algorithm proceeds as follows:

1. Generate initial population P
2. Select a set of solutions D from P
3. Estimate MRF parameters α from D
   - using system of equations (2)
4. Sample MRF using (3) to generate new solutions
5. Go to step 2 until termination criteria are meet

IV. EXPERIMENTS

We aim to investigate the effectiveness of $DEUM_d$ in comparison with genetic algorithms using directed intervention crossover operators. We make comparisons according to two criteria: best (i.e. minimum) fitness achieved; and efficient use of interventions. An algorithm will use interventions efficiently if it is able to find competitive solutions that are parsimonious in their use of interventions.

A. Initialisation Parameter, M

In the real-world situation, there is a practical limit to the number of sprayings that will be possible as this will be done in addition to other work required on the farm. One potential approach to generating solutions that are parsimonious in intervention use is to seed the initial population with solutions that are limited to a certain number of interventions. This should hopefully bias the search in favour of such solutions.

In [2], control of the initialisation conditions was used to explore the quality of solutions that could be obtained when solutions in the initial population were limited to certain levels of intervention use. We will adopt this procedure in the present work.

As in [2], we consider a cultivation period of 50 days and allow intervention points with a granularity of 1 day. Therefore our encoding consists of 50 decision variables, each representing intervention or non-intervention on a particular day in the cultivation period. We will run a series of experiments where the number of interventions in the initial population is controlled by an upper limit, M, which is incremented over the range 1 to 50. For example, when $M = 10$, each member of the starting population has between 1 and 10 interventions scattered across the 50 day cultivation period. After initialisation at a given value of M, the number of interventions used is allowed to increase up to the limit of 50 as solutions evolve.

B. Other Parameter Settings

To compare the effectiveness of different approaches as closely as possible, we set up parameters, such as population size, to be identical wherever possible. All parameters used are detailed in Table 1. All of the approaches use Tournament Selection. The genetic algorithms using UC, CalEB and TInSSel all adopt the same two-stage mutation operator as detailed in [2]. $DEUM_d$ of course replaces these operators with modelling and sampling.

Table 1: Parameters

Parameter	*Value*
Population size	50
Maximum generations	200
Crossover probability	1
Mutation probability	0.05
Cooling rate ($DEUM_d$ only)	0.01
Intervention penalty, P	50

C. Experimental Procedure

Each algorithm was executed for 200 runs, for each value of the initialisation control, M. Results were taken after 50, 100 and 200 generations on each run. For each algorithm we recorded the best fitness score achieved after 50, 100 and 200 generations on each run. These are graphed against the value of M in Figures 1, 3 and 5. We also recorded the number of interventions used by the best solution found on each run after 50, 100 and 200 generations. These are also graphed against M in Figures 2, 4 and 6. Tables 2-4 display, for each algorithm at generations 50, 100 and 200, the mean best fitness, its standard deviation, the standard error and 95% and 99% confidence intervals when $M = 50$. This corresponds to the situation where no prior knowledge exists about the number or distribution of interventions needed for good solutions. Table 5 shows the results of a significance test comparing the performance of $DEUM_d$ with the genetic algorithms after 100 and 200 generations.

V. ANALYSIS OF RESULTS

We present and analyse the results sequentially after 50, 100 and 200 generations in order to observe the dynamics of the different approaches. Running the algorithms beyond this point revealed no further significant observations. Recall that this is a minimization problem and so low values of fitness are to be preferred.

A. Performance after 50 Generations

Figure 1 shows that, irrespective of the initialization conditions, $DEUM_d$ is not finding as good solutions as the genetic algorithms. They all perform best when the number of interventions in the initial population is limited to between 5 and 10, indicating that solutions containing between 5 and 10 interventions are the best for this problem. As M increases beyond 10, the quality of solution found by the genetic algorithms deteriorates but the quality of solutions found by $DEUM_d$ improves. This indicates a very different search dynamic. When $M = 50$, $DEUM_d$ is finding solutions almost as good as those found by the UC genetic algorithm but still not competitive with those found by the genetic algorithms using directed intervention crossover operators.

Figure 2 shows that $DEUM_d$ utilises many more interventions in its best solution found after 50 generations than do the other approaches, around 17 out of 50 possible interventions. This is unlikely to be practical for mushroom

farmers. At this stage in the evolution, TInSSel and CalEB are finding solutions using around 8 interventions regardless of the number of initial intervention beyond 20. This was a finding of [2] and shows the success of the directed intervention approach in using interventions efficiently. UC GA uses the second highest number of interventions and its performance deteriorates as the control limit M is increased. By the time M reaches 50, the UC GA is finding slightly worse solutions than $DEUM_d$ at 50 generations. Table 2 shows mean best solution values, standard deviations, standard error on the mean and 95% and 99% confidence intervals for each algorithm at generation 50 when M is 50 (i.e. no initialisation control).

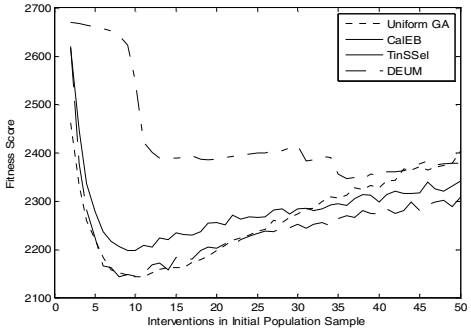

Fig.1 Fitness of best solution after 50 generations for varying values of initialisation control limit M.

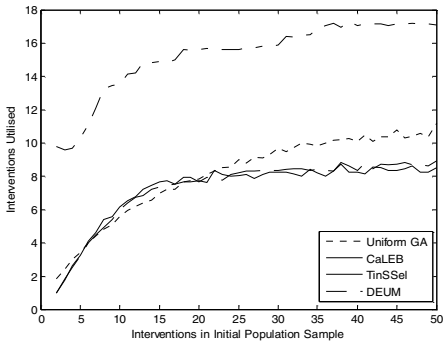

Fig.2 Intervention usage of best solution after 50 generations for varying values of initialisation control limit M

Table 2: Mean fitnesses at 50 generations

	UC GA	CalEB	TInSSel	$DEUM_d$
Mean	2403.7	2341.5	2307.2	2377.5
Std. Dev.	144.8	152.8	153.0	162.2
Std. Err.	6.48	6.84	6.84	7.25
95% conf.	12.73	13.43	13.44	14.21
99% conf.	16.75	17.68	17.69	18.71

B. Performance after 100 Generations

We now allow the evolution to continue and compare the performance of the algorithms after 100 generations. All algorithms have found better solutions by this stage as may be observed from Figure 3. However, we now observe a significant improvement in the relative performance of $DEUM_d$. The difference in search dynamics is again clearly observable. As the limit M on number of interventions is increased up to between 15 and 20, the performance of $DEUM_d$ improves. This is in contrast to the other algorithms where performance declines slightly or even markedly as in the case of the UC GA. For values of M above 15, $DEUM_d$ finds as good or better solutions than the other approaches by generation 100. All of the algorithms show a decline in performance as M increases beyond 20, but this is particularly marked for the UC GA. Figure 4. shows that the number of interventions used by the best solution found by $DEUM_d$ remains impractically high, even though the solutions it is finding are now better than solutions found by the other approaches. It is also noticeable from Figure 4 that the number of interventions used by the UC GA has started to increase to an impractical number. The two directed intervention crossover GAs are successful in keeping the number of interventions down, a performance which is consistent with the results observed in [2].

Table 3 shows mean best solution values, standard deviations, standard error on the mean and 95% and 99% confidence intervals for each algorithm at generation 100 when M is 50 (i.e. no initialisation control).

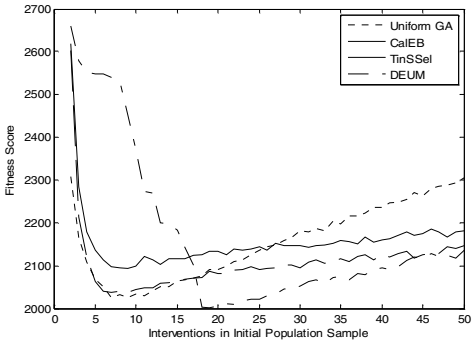

Fig.3 Fitness of best solution after 100 generations for varying values of initialisation control limit M.

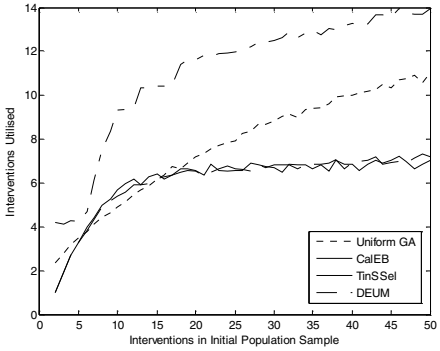

Fig.4 Intervention usage of best solution after 100 generations for varying values of initialisation control limit M

Table 3: Mean fitnesses at 100 generations

	UC GA	CalEB	TInSSel	$DEUM_d$
Mean	2305.3	2180.1	2145.9	2136.7
Std. Dev.	133.2	124.9	120.2	141.0
Std. Err.	5.95	5.59	5.37	6.30
95% conf.	11.70	10.98	10.56	12.35
99% conf.	15.40	14.45	13.89	16.25

C. Performance after 200 Generations

We now allow the evolution to continue until all algorithms have converged. This was observed to happen consistently by 200 generations. Figure 5 shows that, once all algorithms have converged, $DEUM_d$ finds the lowest fitness solutions overall, provided the initialisation control M exceeds around 12. For higher values of M**,** the best fitness achieved remains stable status close to the best result**.** This performance is now better than that of the directed intervention crossover GAs, TInSSel and CalEB, though they still maintain a better performance if the initial number of interventions is limited to below 12.

Figure 6 shows that the number of interventions utilised by $DEUM_d$ remains higher than that used by TinSSel and CalEB. However at on average 6.5 interventions over 50 days, the solutions found by $DEUM_d$ are practical. By comparison TInSSel and CalEB find solutions containing around 4.5 interventions on average. The results show that there is a significant gain in larva reduction to be made in compensation for two extra interventions while staying within the limits of practicality. It is also noticeable that $DEUM_d$ is able to resist the proliferation of interventions even when the initialisation control M is removed. The UC GA continues to use an excessive number of interventions and tends to find poorer solutions, increasingly so as the initialisation control is removed.

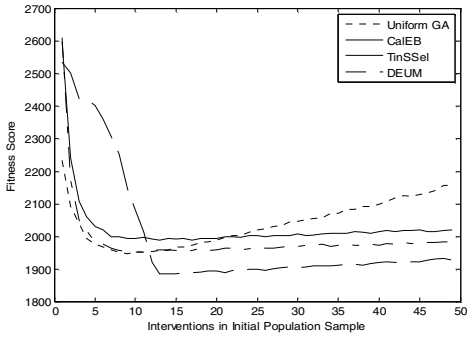

Fig.5 Fitness of best solution after 200 generations for varying values of initialisation control limit M.

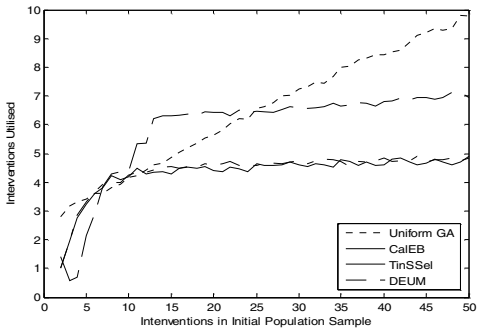

Fig.6 Intervention usage of best solution after 200 generations for varying values of initialisation control limit M

Table 4: Mean fitnesses at 200 generations

	UC GA	CalEB	TInSSel	$DEUM_d$
Mean	2156.9	2018.5	1983.0	1927.2
Std. Dev.	117.3	70.4	72.9	69.9
Std. Err.	5.24	3.15	3.26	3.13
95% conf.	10.30	6.18	6.40	6.13
99% conf.	13.56	8.14	8.43	8.09

Table 4 shows mean best solution values, standard deviations, standard error on the mean and 95% and 99% confidence intervals for each algorithm at generation 200 (convergence) when M is 50 (i.e. no initialisation control). In Table 5 we make a statistical comparison of performance at generations 100 and 200 using a Wilcoxon rank-sum test. At generation 100, there is no significant difference in the mean fitness of the best solutions found by $DEUM_d$ and TlnSSel, at a 95% confidence level. $DEUM_d$ does however outperform CalEB and UC GA in terms of fitness. However, the schedules generated by TInSSel at this stage are more practical. At convergence (generation 200), with no initialisation control, $DEUM_d$ significantly outperforms all of the other approaches.

Table 5: Results of the Wilcoxon rank-sum test comparison of fitness scores for each approach at generations 100 and 200

	$DEUM_d$ vs. Uniform	*$DEUM_d$ vs. CalEB*	*$DEUM_d$ vs. TlnSSel*
Gene=100	19.441	5.157	1.115
Gene=200	37.620	20.578	12.353

VI. CONCLUSION

In this paper we have applied an Estimation of Distribution Algorithm, $DEUM_d$, to a bio-control problem in mushroom farming. The problem has some interesting features that allow us to compare the relative merits of using an EDA against GAs that use specialised operators, in this case directed intervention crossover.

The bio-control application requires a discrete set of interventions, each of which is costly to implement. There are therefore two conflicting objectives that are combined in a single weighted objective function. The use of an

initialisation control was explored to limit the number of interventions in the initial population of solutions. This might reflect a pragmatic decision to encourage algorithms to generate only practicable solutions.

The results show that, while such a control is useful for UC GA, and fairly neutral for TInSSel and CalEB, it actually impedes $DEUM_d$. This reflects the different dynamics of the search. For the UC GA, as the initialisation control is relaxed, more and more interventions creep into solutions and the weighted penalty is ineffective at removing them. Thus the quality of solutions that can be found deteriorates. For the GAs using the directed intervention crossovers, TInSSel and CalEB, the relaxation of initialisation control has little effect because the crossovers are designed to direct interventions to points in the schedule where they have best effect rather than to blindly accumulate them. For $DEUM_d$, which relies on model building the restriction on the initial population restricts the ability of the algorithm to sample a range of possible intervention points. It is clear from the analyses in Section V that, in the early stages of the evolution, $DEUM_d$ operates on solutions that use infeasible numbers of interventions. However by convergence at 200 generations, this has resulted in more accurate targeting of interventions that more successfully balances the two objectives, giving significantly better solutions. In this application, TInSSel and CalEB are processing fewer intervention points at any time, and therefore may miss relatively small but significant improvements in fitness.

Here we have shown that the probabilistic modelling approach in $DEUM_d$ does provide additional benefit over genetic operators in targeting interventions effectively. In future work we intend to explore the potential of EDAs to discover structure in the problem that may provide better understanding of the system dynamics as a whole.

REFERENCES

[1] A. Fenton, R.L. Gwynn, A. Gupta, R. Norman, J.P. Fairbairn, and P.J. Hudson. Optimal application strategies for entomopathogenic nematodes: integrating theoretical and empirical approaches. Journal of Applied Ecology 2002 39,481-492.

[2] P. M. Godley, D. E. Cairns, and J. Cowie. Directed intervention crossover applied to bio-control scheduling. In *IEEE CEC 2007: Proceedings of the IEEE Congress On Evolutionary Computation*, 2007.

[3] P. M. Godley, D. E. Cairns, and J. Cowie. Maximising the efficiency of bio-control application utilising genetic algorithms. In *EFITA / WCCA 2007: Proceedings of the 6th Biennial Conference of European Federation of IT in Agriculture*, Glasgow, Scotland, UK, 2007. Glasgow Caledonian University.

[4] Paul M. Godley and Julie Cowie and David E. Cairns, Novel Genetic Algorithm Approaches for Time-Series Problems, Doctoral Symposium on Engineering Stochastic Local Search Algorithms pp47-51, IRIDIA, 2007 ISSN: 1781-3794

[5] Siddhartha Shakya, John McCall. Optimization by Estimation of Distribution with DEUM Framework Based on Markov Random Fields. International Journal of Automation and Computing.04(3),July 2007,262-272.

[6] Andrei Petrovski, Siddhartha Shakya, John McCall. Optimisimg Cancer Chemotherapy Using an Estimation of Distribution Algorithm and Genetic Algorithms. GECCO'06, July 08-12, 2006, Seattle, WA, USA.

[7] A Petrovski, J. McCall, Multi-objective Optimization of Cancer Chemotherapy Using Evolutionary Algorithms, in E. Zitzler , K. Deb, L. Thiele, C.A. Coello Coello, D.Corne, (eds) Proc. EMO Conference, Zurich. Springer-Verlag, pp 531-545, Berlin, 2001

[8] Siddhartha Shakya, John McCall and Deryck Brown. Using a Markov network model in a univariate EDA: an empirical cost-benefit analysis. in GECCO 2005, 2005, Washington,DC,USA

[9] John McCall, Andrei Petrovski, Siddhartha Shakya, Evolutionary Algorithms for Cancer Chemotherapy Optimisation, in G. B. Fogel , D. W. Corne , Y. Pan (Eds) *Computational Intelligence in Bioinformatics*. Chapter 12 pp265-294, Wiley-IEEE Press, New York, NY, USA, 2007.