

# Optimising Cancer Chemotherapy Using an Estimation of Distribution Algorithm and Genetic Algorithms

Andrei Petrovski  
School of Computing  
The Robert Gordon University  
Aberdeen, UK  
ap@comp.rgu.ac.uk

Siddhartha Shakya  
School of Computing  
The Robert Gordon University  
Aberdeen, UK  
ss@comp.rgu.ac.uk

John McCall  
School of Computing  
The Robert Gordon University  
Aberdeen, UK  
jm@comp.rgu.ac.uk

## ABSTRACT

This paper presents a methodology for using heuristic search methods to optimise cancer chemotherapy. Specifically, two evolutionary algorithms - Population Based Incremental Learning (PBIL), which is an Estimation of Distribution Algorithm (EDA), and Genetic Algorithms (GAs) have been applied to the problem of finding effective chemotherapeutic treatments. To our knowledge, EDAs have been applied to fewer real world problems compared to GAs, and the aim of the present paper is to expand the application domain of this technique.

We compare and analyse the performance of both algorithms and draw a conclusion as to which approach to cancer chemotherapy optimisation is more efficient and helpful in the decision-making activity led by the oncologists.

## Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search

; G.3 [Probability and statistics]: Probabilistic algorithms, Stochastic processes

; J.3 [Life and Medical Sciences]: Health

## General Terms

Algorithms, Performance, Theory

## Keywords

Estimation of Distribution Algorithms, Evolutionary Computation, Probabilistic Modelling

## 1. INTRODUCTION

Many decision-making activities involve searching through a large space of possible solutions. In the chemotherapy problem we have studied [17], [19], [20], the size of the solution space increases exponentially with the number of de-

cision variables, the values of which need to satisfy certain feasibility criteria.

The requirements imposed on decision variables often make the structure of a solution space quite intricate - regions of feasible solutions are scattered irregularly throughout the solution space, and only one of these regions contains the optimal solution. To find the optimal solution in such situations becomes a difficult task for conventional optimisation methods (gradient-based or simple heuristics). Similarly, the methods of mathematical programming cannot easily deal with multiplicity of feasible regions in the solution space or complementary constraints. Although a number of advances have been made in the area of solving non-linear optimisation problems using numerical methods [6], they cannot guarantee finding a feasible solution - let alone the optimal one.

It has been found [14], [19], [25], [26] that Genetic Algorithms show a good and robust performance on a class of non-linear, multi-constrained chemotherapy design problems. However, the field of evolutionary computation is growing, and alternative techniques of computational optimisation are being developed. The advent of probabilistic models for evolutionary computation has led to the development of more efficient algorithms that are able to resolve some of the limitations exhibited by conventional Genetic Algorithms [8]. One such technique that uses probabilistic models is the Estimation of Distribution Algorithm (EDA) [15], which replaces the traditional crossover and mutation operators used by GAs with the estimation and sampling from probabilistic models of each generation [21]. In EDAs, the important patterns or building blocks are identified from the population of promising solutions. A model of the probability distribution is used to preserve those patterns and is explicitly sampled to generate a better population. EDAs are reported to solve problems that are known to be GA-hard [16].

Various implementations of EDAs have been proposed in the literature. They can be categorised into three groups: a) univariate b) bivariate and c) multivariate (see [12] for more information). For the purpose of this paper, we will concentrate on an algorithm called Population Based Incremental Learning (PBIL), proposed by Baluja et al. [2], [3]. PBIL is a univariate EDA that uses binary representation of the chromosome. Other univariate EDA include Univariate Marginal Distribution Algorithm (UMDA) [15], compact Genetic Algorithm (cGA) [9], and a recently proposed EDA called Distribution Estimation using Markov Random Fields (DEUM) [24, 23]. Amongst other EDAs, PBIL is one of the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'06, July 8-12, 2006, Seattle, Washington, USA.

Copyright 2006 ACM 1-59593-186-4/06/0007 ...\$5.00.

most frequently-occurring algorithm used for solving real life optimisation problems [7, 11].

Although studied extensively in theory, only a few real-world applications of EDAs have been reported [1, 22]. The purpose of this paper is to ascertain the applicability of the PBIL to the problem of finding effective chemotherapeutic treatments and to compare its performance with that of the known method, in particular with that of GAs.

The paper is structured as follows. Section 2 introduces the problem of cancer chemotherapy. Section 3 describes the methodology of solving this problem by transforming a chemotherapeutic treatment into a GA/PBIL chromosome. Section 4 presents the experimental results; the final Section 5 discusses the results and concludes the paper.

## 2. OPTIMIZATION PROBLEM OF CANCER CHEMOTHERAPY

Amongst the modalities of cancer treatment, chemotherapy is often considered as inherently the most complex [27]. As a consequence of this, it is extremely difficult to find effective chemotherapy treatments without a systematic approach. In order to realise such an approach, we need to take into account the medical aspects of cancer treatment.

### 2.1 Medical aspects of Chemotherapy

Drugs used in cancer chemotherapy all have narrow therapeutic indices. This means that the dose levels at which these drugs significantly affect a tumour are close to those levels at which unacceptable toxic side-effects occur. Therefore, more effective treatments result from balancing the beneficial and adverse effects of a combination of different drugs, administered at various dosages over a treatment period [19].

The beneficial effects of cancer chemotherapy correspond to treatment objectives which oncologists want to achieve by means of administering anti-cancer drugs. A cancer chemotherapy treatment may be either curative or palliative. Curative treatments attempt to eradicate the tumour; palliative treatments, on the other hand, are applied only when a tumour is deemed to be incurable with the objective to maintain a reasonable quality of life for as long as possible.

The adverse effects of cancer chemotherapy stem from the systemic nature of this treatment: drugs are delivered via the bloodstream and therefore affect all body tissues. Since most anti-cancer drugs are highly toxic, they inevitably cause damage to sensitive tissues elsewhere in the body. In order to limit this damage, toxicity constraints need to be placed on the amount of drug applied at any time interval, on the cumulative drug dosage over the treatment period, and on the damage caused to various sensitive tissues [27]. In addition to toxicity constraints, the tumour size (i.e. the number of cancerous cells) must be maintained below a lethal level during the whole treatment period for obvious reasons.

The goal of cancer chemotherapy therefore is to achieve the beneficial effects of treatment objectives without violating any of the above mentioned constraints.

### 2.2 Problem formulation

In order to solve the optimisation problem of cancer chemotherapy, we need to find a set of treatment schedules, which satisfies toxicity and tumour size constraints yielding at the same time acceptable values of treatment objectives. This

set will allow the oncologist to make a decision on which treatment schedule to use, given his/her preferences or certain priorities. In the remainder of this section we will define the decision vectors and the search space for the cancer chemotherapy optimisation problem, specify the constraints, and particularise the optimisation objectives.

Anti-cancer drugs are usually delivered according to a discrete dosage program in which there are  $s$  doses given at times  $t_1, t_2, \dots, t_s$  [13]. In the case of multi-drug chemotherapy, each dose is a cocktail of  $d$  drugs characterised by the concentration levels  $C_{ij}, i \in \overline{1, s}, j \in \overline{1, d}$  of anti-cancer drugs in the bloodplasma. Optimisation of chemotherapeutic treatment is achieved by modification of these variables. Therefore, the solution space  $\Omega$  of the chemotherapy optimisation problem is the set of control vectors  $\mathbf{c} = (C_{ij})$  representing the drug concentration profiles.

However, not all of these profiles will be feasible as chemotherapy treatment must be constrained in a number of ways. Although the constraint sets of chemotherapeutic treatment vary from drug to drug as well as with cancer type, they have the following general form.

1. Maximum instantaneous dose  $C_{max}$  for each drug acting as a single agent:

$$g_1(\mathbf{c}) = \left\{ C_{maxj} - C_{ij} \geq 0 : \forall i \in \overline{1, s}, \forall j \in \overline{1, d} \right\} \quad (1)$$

2. Maximum cumulative  $C_{cum}$  dose for drug acting as a single agent:

$$g_2(\mathbf{c}) = \left\{ C_{cumj} - \sum_{i=1}^s C_{ij} \geq 0 : \forall j \in \overline{1, d} \right\} \quad (2)$$

3. Maximum permissible size of the tumour:

$$g_3(\mathbf{c}) = \left\{ N_{max} - N(t_i) \geq 0 : \forall i \in \overline{1, s} \right\} \quad (3)$$

4. Restriction on the toxic side-effects of multi-drug chemotherapy:

$$g_4(\mathbf{c}) = \left\{ C_{s-effk} - \sum_{j=1}^d \eta_{kj} C_{ij} \geq 0 : \forall i \in \overline{1, s}, \forall k \in \overline{1, m} \right\} \quad (4)$$

The factors  $\eta_{kj}$  in the last constraint represent the risk of damaging the  $k^{th}$  organ or tissue (such as heart, bone marrow, lung etc.) by administering the  $j^{th}$  drug. Estimates of these factors for the drugs most commonly used in treatment of breast cancer, as well as the values of maximum instantaneous and cumulative doses, can be found in [4], [5].

Regarding the objectives of cancer chemotherapy, we focus our study on the primary objective of cancer treatment - tumour eradication. We define eradication to mean a reduction of the tumour from an initial size of around  $10^9$  cells (minimum detectable tumour size) to below  $10^3$  cells.

In order to simulate the response of a tumour to chemotherapy, a number of mathematical models can be used [13]. The most popular is the Gompertz growth model with a linear cell-loss effect [27]:

$$\frac{dN}{dt} = N(t) \cdot \left[ \lambda \ln \left( \frac{\Theta}{N(t)} \right) - \right]$$

$$\left[ \sum_{j=1}^d K_j \sum_{i=1}^s C_{ij} \{H(t - t_i) - H(t - t_{i+1})\} \right] \quad (5)$$

where  $N(t)$  represents the number of tumour cells at time  $t$ ;  $\lambda, \Theta$  are the parameters of tumour growth,  $H(t)$  is the Heaviside step function;  $k_j$  are the quantities representing the efficacy of anti-cancer drugs, and  $C_{ij}$  denote the concentration levels of these drugs. One advantage of the Gompertz model from the computational optimisation point of view is that the equation (5) yields an analytical solution after the substitution  $u(t) = \ln(\Theta/N(t))$  [13]. Since  $u(t)$  increases when  $N(t)$  decreases, the primary optimisation objective of tumour eradication can be formulated as follows [17]:

$$\underset{\mathbf{c}}{\text{minimise}} F(\mathbf{c}) = \sum_{i=1}^s N(t_i) \quad (6)$$

subject to the state equation (5) and the constraints (1)-(4).

### 3. METHODOLOGY

In this section we are going to explain how the optimisation problem of cancer chemotherapy can be addressed by applying GA and PBIL. In order to make the comparison fair, the binary representation of solutions for both algorithms was used. It has been reported that integer encoding of GA solutions can improve the algorithm's performance [18], but for the purpose of the present study we have implemented Genetic Algorithms using binary representation to make the analysis of comparative results valid.

#### 3.1 Binary string representation of the problem

Multi-drug chemotherapy schedules, represented by decision vectors  $\mathbf{c} = (C_{ij}), i \in \overline{1, s}, j \in \overline{1, d}$ , are encoded as binary strings known as *chromosomes*. The representation space  $\mathbf{I}$  (a discretized version of  $\Omega$ ) can then be expressed as a Cartesian product

$$\mathbf{I} = A_1^1 \times A_1^2 \times \dots \times A_1^d \times A_2^1 \times A_2^2 \times \dots \times A_2^d \times \dots \times A_s^1 \times A_s^2 \times \dots \times A_s^d$$

of allele sets  $A_i^j$ . Each allele set uses a 4-bit representation scheme

$$A_i^j = \left\{ x_1 x_2 x_3 x_4 : x_k \in \{0, 1\} \forall k \in \overline{1, 4} \right\}$$

so that each concentration level  $C_{ij}$  takes an integer value in the range of 0 to 15 concentration units [19]. In general, with  $s$  treatment intervals and up to  $2^p$  concentration levels for  $d$  drugs, there are up to  $2^{spd}$  individual elements. Henceforth we assume that  $s = 10$  and that the number of available drugs is restricted to ten [17]. These drugs are delivered sequentially - one after another - to form a multi-drug dose, which is administered periodically over the treatment period that consists of up to  $s$  cycles. The values  $s = 10$  and  $d = 10$  result in the individual (search) space of power  $|\mathbf{I}| = 2^{400}$  individuals, referred to as chromosomes.

Thus, a chromosome  $x \in \mathbf{I}$  can be expressed as

$$x = \left\{ x_1 x_2 x_3 \dots x_{4sd} : x_k \in \{0, 1\} \forall k \in \overline{1, 4sd} \right\}$$

and the mapping function  $m : \mathbf{I} \rightarrow \mathbf{C}$  between the individual  $\mathbf{I}$  and the decision vector  $\mathbf{C}$  spaces can be defined as

$$C_{ij} = \Delta C_j \sum_{k=1}^4 2^{4-k} x_{4d(i-1)+4(j-1)+k}, \quad \forall i \in \overline{1, s}, j \in \overline{1, d}$$

where  $\Delta C_j$  represents the concentration unit for drug  $j$ . This function symbolizes the decoding algorithm to derive a decision vector from a chromosome  $x$ . Applying the evaluation function  $F$  to  $\mathbf{c}$  yields the value of the fitness function for both algorithms.

$$F(\mathbf{c}) = \sum_{p=1}^n \sum_{j=1}^d \kappa_j \sum_{i=1}^p C_{ij} e^{\lambda(t_{i-1}-t_p)} - \sum_{s=1}^4 P_s d_s \quad (7)$$

where  $d_s$  are the distance measures specifying how seriously the constraints (1)-(4) are violated, and  $P_s$  are the corresponding penalty coefficients. If all constraints are satisfied (i.e. a treatment regimen is feasible), then the second term in (7) will be zero, significantly increasing the value of the fitness function.

#### 3.2 Genetic Algorithms

A general GA starts by randomly generating  $M$  number of chromosomes to form a population  $P$  (also known as parent population). As stated above, each chromosome  $x = \{x_1, x_2, \dots, x_n\}$ , where  $n = 4sd$ , is a binary string representing a chemotherapy schedule.  $N$  chromosomes are then selected from  $P$  according to the quality/fitness of those chromosomes. Various selection mechanisms can be applied for this purpose. In our case we use *fitness proportionate* selection [8].

Other GA parameters have been chosen on the basis of the previous work, presented in [18], aimed at tuning the GA factors in order to improve the algorithm's performance.

#### 3.3 Population Based Incremental Learning

Similarly to a binary GA, PBIL solutions are also bit-string chromosomes. So, in our case, a chromosome  $x = \{x_1, x_2, \dots, x_n\}$  represents a candidate chemotherapeutic treatment, where  $n = 4sd$  is the chromosome length, and each  $x_i$  is either 0 or 1.

PBIL starts by initializing a probability vector  $p = \{p_1, p_2, \dots, p_n\}$  where, each  $p_i = 0.5$ . Each  $p_i$  represents, the probability of 1 being presented in  $i^{th}$  position of a chromosome.  $p$  is then sampled  $M$  times to create a population  $P$  of chromosomes.  $N$  chromosomes are then selected from  $P$  according to the quality/fitness of that chromosome. As with GA, a number of selection mechanisms can be applied for this purpose. In our case we use truncation selection [12] which is to select the best  $N$  solutions from  $P$ . After selection, the marginal probability  $\rho_i$  for each  $i^{th}$  allele position is calculated from the selected  $N$  solutions. ( $\rho_i$  can be simply calculated by dividing the frequency of 1 in  $i^{th}$  position of allele in selected set by  $N$ ).  $\rho_i$  is then used to update the probability vector  $p$  (see workflow of PBIL for more detail). This updated probability vector replaces the initial probability vector. This process continues until termination criteria are satisfied.

Workflow of PBIL is shown below:

1. Initialise a probability vector  $p = \{p_1, p_2, \dots, p_n\}$  where, each  $p_i = 0.5$
2. Sample  $p$  to generate an initial population,  $P$ , of size  $M$ .
3. Select the  $N$  fittest solutions from  $P$ , where  $N \leq M$ .
4. For each allele  $x_i$ , calculate the marginal probability  $\rho_i$  from selected  $N$  solutions.
5. Update  $p$  using following updating rule:

for  $i = 1$  to  $n$  do

$$p_i = p_i * (1 - \lambda) + \rho_i * \lambda$$

where,  $0 \leq \lambda \leq 1$ .  $\lambda$  is known as *learning rate parameter* [2] chosen by the user.

6. Go to Step 2 until the termination criterion is satisfied.

## 4. EXPERIMENTAL RESULTS

We compare the performance of GA and PBIL on the problem of multi-drug cancer chemotherapy optimization addressed in [14], [19]. Two different measures were adopted to evaluate the performance of each algorithm:

1. **Efficiency / Speed:** The efficiency of an algorithm can be measured in terms of computational time it takes to find a solution. In the case of evolutionary algorithms, the calculation of the fitness of a particular chromosome (fitness evaluation) is the most computationally expensive task. Therefore, we measure the efficiency in terms of number of fitness evaluations taken by the algorithm to find a solution. In our case, finding a solution means to find at least one feasible solution (i.e. satisfying all the constraints (1)-(4)).

2. **The quality of solution:** A solution's quality in our case is the value of the fitness function (7) that indicates how successful a particular chemotherapy schedule is in reducing the overall tumour burden. Therefore, the quality of solution is measured in terms of the best fitness, i.e the best chromosome (best schedule that minimises the tumour burden) found by the algorithms.

### 4.1 Efficiency comparison

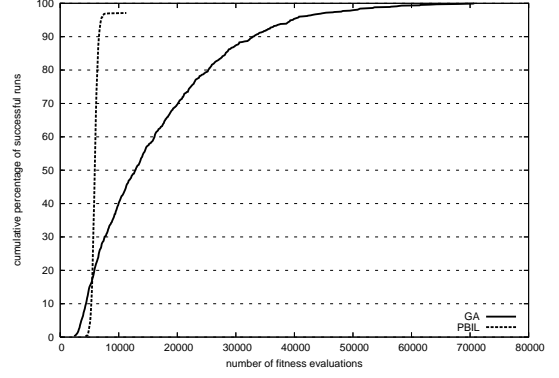
In the efficiency comparison experiment, each algorithm was run 1000 times and for each run the number of fitness evaluations taken to find at least one feasible solution (i.e. satisfying all the constraints (1)-(4)) was recorded. Run Length Distribution (RLD)[10] curves were plotted to measure the performance (Figure 1). RLD shows, for each algorithm, the cumulative percentage of successful runs that terminated within a certain number of function evaluations.

The parameters for each of the algorithms were chosen either on the basis of the previous work [18] (for Genetic Algorithms) or empirically (for PBIL), taking into consideration the fairness of the comparison.

In accordance with the best GA parameter settings found in [18] for the cancer chemotherapy problem, we used a GA population of 100 chromosomes, one-point crossover with the probability of 0.614, uniform mutation with probability of 0.075, and the elitism of two best chromosomes. The algorithm terminates when it finds a feasible solution.

Similarly, PBIL population consisted of 100 chromosomes, the selection size  $N$  was 20, and the learning rate  $\lambda$  was 0.3.

Figure 1 shows that the performance of PBIL was better than the performance of GA. We can see that, for PBIL 90%



**Figure 1: Experimental results in the form of RLD showing, for each algorithm running on chemotherapy optimization problem, the cumulative percentage of successful runs that terminated within a certain number of function evaluations.**

of the runs found feasible solutions within 6,700 function evaluation in comparison with 32,900 function evaluations for GA <sup>1</sup>. Also, the steepness of the RLD curve for PBIL shows that this algorithm is very reliable and finds solutions within a predictable number of function evaluations. At the same time, the gradual slope of the RLD curve for Genetic Algorithms shows that the number of GA fitness evaluations required to find a feasible solution varies for almost every individual run and cannot be predicted accurately.

**Table 1: Mean and stdev of the number of fitness evaluations**

|      | mean     | stdev    |
|------|----------|----------|
| GA   | 16208.10 | 12045.81 |
| PBIL | 5953.45  | 522.40   |

The mean value and the standard deviation of the number of function evaluations taken by the algorithms over 1000 runs is presented in the Table 1. These results are analyzed by applying a one-tailed t-test that yields the p-value of  $< 0.05$  (Table 2), indicating statistical significance of the difference in speed for PBIL and GA.

**Table 2: T-test analysis of PBIL and GA mean values**

| Diff. in means | Std. error Mean | t-test | p-value    |
|----------------|-----------------|--------|------------|
| 10254.65       | 381.29          | 26.89  | $< 0.0001$ |

### 4.2 Comparison of Treatment Quality

The goal of the second experiment was to find the best chemotherapeutic regimen within a specified number of function evaluations, which was set to 200,000. In order to better achieve this goal, the population size of both GA and PBIL was increased to 200 chromosomes. Also, to encourage

<sup>1</sup>A smaller number of function evaluations indicates the ability of an algorithm to find a feasible solution faster.

more thorough exploration of the solution space, the mutation probability of GA was adjusted in accordance with the strategies suggested in [18]. Also, the learning rate of PBIL was decreased to 0.05.

The results of the treatment quality experiment are presented in Table 3, which shows the mean and the standard deviation of the fitness values found by the algorithms.

**Table 3: mean and stdev of fitness**

|      | mean   | stdev  |
|------|--------|--------|
| GA   | 0.4520 | 0.0354 |
| PBIL | 0.4917 | 0.0005 |

It is evident that the PBIL solution is better than that of GA <sup>2</sup>. This conclusion is supported by a one-tailed t-test applied to the fitness values found by the algorithms and presented in Table 4.

**Table 4: Significance test of results on quality of fitness**

| Diff. in means | Std. error Mean | t-test  | p-value  |
|----------------|-----------------|---------|----------|
| -0.0397        | 0.003           | 15.8676 | < 0.0001 |

The p-value in Table 4 indicates that the difference in the best fitness values for PBIL and GA is highly significant. This means that PBIL outperforms GA on the chemotherapy optimisation problem not only with respect to the computational speed of finding a feasible treatment, but with respect to the quality of the found solution as well.

## 5. CONCLUSION

In this paper we have examined the problem of designing efficient chemotherapeutic treatments that satisfy toxicity and tumour size constraints. This problem has been formulated in terms of exploring a vast search space of possible treatment schedules using two computational heuristic algorithms - Estimation of Distribution Algorithm (in the form of PBIL) and Genetic Algorithm.

The heuristic algorithms proposed have been successfully applied to the chemotherapy optimisation problem and were able to find feasible solutions to this problem in a reasonable computational time. Our experimental results show that PBIL outperforms Genetic Algorithms in both the speed of finding a feasible treatment schedule and in the quality of the final solution.

This result can have practical implications. The better quality of the drug schedules found by PBIL can lead to more effective chemotherapeutic treatments. On the other hand, the ability of PBIL to explore the solution space of possible treatments more efficiently, enables us to provide a more helpful decision-support tool using OWCH - the online decision support system described in [14].

Also, a successful application of PBIL in the present context opens the possibility of using multivariate EDAs for designing chemotherapeutic treatments. Multivariate EDAs

have been shown to perform better on a wider range of optimization problems as they are able to capture the interactions between variables in their probabilistic model. This is particular important in the context of cancer chemotherapy optimisation, where interactions between drugs play a vital role in the overall success of a treatment regimen [5]. We intend to address this issue in the future.

## 6. REFERENCES

- [1] J. Aickelin and J. Li. An Estimation of Distribution Algorithm for Nurse Scheduling. *Annals of Operational Research*, in print, 2006.
- [2] S. 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.
- [3] S. Baluja and S. 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.
- [4] J. Cassidy and H. McLeod. Is it possible to design a logical development plan for an anti-cancer drug? *Pharmaceutical Medicine*, 9:95–103, 1995.
- [5] D. Dearnaly, I. Judson, and T. Root, editors. *Handbook of adult cancer chemotherapy schedules*. The Medicine Group (Education) Ltd., Oxfordshire, 1995.
- [6] R. Fletcher and S. Leyffer. Numerical experience with solving mpecs as nlps. Technical Report NA/210, Department of Mathematics, University of Dundee, UK, 2002.
- [7] E. Galić and M. Höhfeld. Improving the generalization performance of multi-layer-perceptrons with population-based incremental learning. In *Parallel Problem Solving from Nature. PPSN-IV*, pages 740–750, 1996.
- [8] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [9] G. R. Harik, F. G. Lobo, and D. E. Goldberg. The compact genetic algorithm. *IEEE-EC*, 3(4):287, November 1999.
- [10] H. H. Hoos and T. Stutzle. Towards a characterisation of the behaviour of stochastic local search algorithms for SAT. *Artificial Intelligence*, 112(1-2):213–232, 1999.
- [11] I. Inza, M. Merino, P. Larrañaga, J. Quiroga, B. Sierra, and M. Giralá. Feature subset selection by population-based incremental learning. A case study in the survival of cirrhotic patients with TIPS. *Artificial Intelligence in Medicine*, In press, 2001.
- [12] P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.
- [13] R. Martin and K. Teo. Optimal control of drug administration in cancer chemotherapy. *World Scientific*, 1994.
- [14] J. McCall and A. Petrovski. A decision support system for cancer chemotherapy using genetic algorithms. In *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*, pages 65–70. IOS Press, 1999.

<sup>2</sup>In this experiment we are trying to find a solution with the largest value of the fitness function - such a solution represents the best feasible treatment schedule, and thus preferable.

- [15] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions: I. binary parameters. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature – PPSN IV*, pages 178–187, Berlin, 1996. Springer.
- [16] M. Pelikan and D. E. Goldberg. Hierarchical BOA solves Ising spin glasses and MAXSAT. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, pages 1271–1282, 2003. Also IlliGAL Report No. 2003001.
- [17] A. Petrovski. *An Application of Genetic Algorithms to Chemotherapy Treatment*. PhD thesis, The Robert Gordon University, Aberdeen, UK, 1999.
- [18] A. Petrovski, A. Brownlee, and J. McCall. Statistical optimisation and tuning of GA factors. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages Vol. 1, 758–764, Edinburgh, Scotland, 2005. IEEE Press.
- [19] A. Petrovski and J. McCall. Multi-objective optimisation of cancer chemotherapy using evolutionary algorithms. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimisation*, Zurich, Switzerland, 2001.
- [20] A. Petrovski, B. Sudha, and J. McCall. Optimising cancer chemotherapy using particle swarm optimisation and genetic algorithms. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature*, pages 633–641, Birmingham, UK, 2004. Springer: Lecture Notes in Computer Science 3224.
- [21] R. Sanatana, P. Larranaga, and J. Lozano. Interactions and dependencies in estimation of distribution algorithms. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1418–1425, Edinburgh, UK, 2005. IEEE press.
- [22] R. Sanatana, P. Larranaga, and J. Lozano. Protein structure prediction in simplified models with estimation of distribution algorithms. In *Proceedings of the IV Congreso Espanol sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB-2005)*, pages 245–252, Granada, Spain, 2005.
- [23] S. Shakya, J. McCall, and D. Brown. Using a Markov Network Model in a Univariate EDA: An Emperical Cost-Benefit Analysis. In *In proceedings of Genetic and Evolutionary Computation Conference (GECCO 2005)*, Washington, D.C., USA, 2005. ACM.
- [24] S. K. Shakya, J. A. W. McCall, and D. F. Brown. Updating the probability vector using MRF technique for a Univariate EDA. In E. Onaindia and S. Staab, editors, *Proceedings of the Second Starting AI Researchers’ Symposium*, pages 15–25, Valencia, Spain, 2004. IOS press.
- [25] K. C. Tan, E. F. Khor, J. Cai, C. M. Heng, and T. H. Lee. Automating the drug scheduling of cancer chemotherapy via evolutionary computation. *Artificial Intelligence in Medicine*, pages 169–185, 2002.
- [26] M. Villasana and G. Ochoa. Heuristic design of cancer chemotherapies. *IEEE Transactions on Evolutionary Computation*, 8(6):513–521, December 2004.
- [27] T. Wheldon. *Mathematical models in cancer research*. Adam Hilger, Bristol, Philadelphia, 1988.