The Effects of Mutation and Directed Intervention Crossover when applied to Scheduling Chemotherapy

Paul M. Godley
Department of Computing
Science and Mathematics
University of Stirling
Stirling, Scotland
pgo@cs.stir.ac.uk

David E. Cairns
Department of Computing
Science and Mathematics
University of Stirling
Stirling, Scotland
dec@cs.stir.ac.uk

Julie Cowie
Department of Computing
Science and Mathematics
University of Stirling
Stirling, Scotland
jco@cs.stir.ac.uk

John McCall
The School of Computing,
Faculty of Design and
Technology
The Robert Gordon University
Aberdeen, Scotland
jm@comp.rgu.ac.uk

Kevin M. Swingler
Department of Computing
Science and Mathematics
University of Stirling
Stirling, Scotland
kms@cs.stir.ac.uk

ABSTRACT

This paper discusses the effects of mutation and directed intervention crossover approaches when applied to the derivation of cancer chemotherapy treatment schedules. Unlike traditional Uniform Crossover (UC), the directed intervention techniques actively choose the intervention level based on the fitness of the parents selected for crossover. This work describes how directed intervention crossover principles are more robust to mutation and lead to significant improvement over UC when applied to cancer chemotherapy treatment scheduling.

Categories and Subject Descriptors

F.2.2 [**Problem Solving, Control Methods, and Search**]: Sequencing and scheduling; G.3 [**Probability and Statistics**]: Time series analysis; I.2.8 [**Problem Solving, Control Methods, and Search**]: Scheduling

General Terms

Algorithms

Keywords

Genetic Algorithms, Time Series, Chemotherapy, Crossover

1. INTRODUCTION

Genetic Algorithms (GAs) are frequently used to find effective solutions in a large solution space and may be applied to both static and time dependent problem domains

[1]. Previous work into GA crossover approaches produced a directed intervention crossover method, Targeted Intervention with Stochastic Selection (TInSSel), suitable for application to time series problems [4]. This crossover approach was used to derive optimisation schedules for the application of nematode worms as a bio-control agent for combating sciarid flies in the production of mushroom crops [5, 6].

TInSSel was shown to outperform traditional Uniform Crossover (UC) when deriving schedules of bio-control application over a range of initial intervention spreads. This work introduces two alternative crossover approaches to TInSSel – Fitness Directed Search (FDS) and Directed Uniform Crossover (DUC). In contrast to UC, TInSSel, FDS and DUC operate on the principle of calculating the required number of interventions to use in offspring based on the relationships between parents fitness values and intervention levels.

In order to test these techniques, a cancer chemotherapy treatment problem was chosen since GAs have already been successfully used in chemotherapy design problems [15]. This has a similar form to the previous scheduling work as anti-cancer drugs are generally applied according to a schedule where s doses are given at times $t_1, t_2, \ldots, t_s$ [8].

In addition, the search effectiveness of a GA is significantly affected by mutation [18]. With respect to intervention processes such as chemotherapy scheduling, it is therefore important to investigate the performance of an algorithm with respect to different levels of mutation in order to understand their robustness and ability to exploit diversity.

This paper is structured as follows: Section 2 outlines the approaches under consideration, Section 3 details the cancer chemotherapy scheduling problem and Section 3.1 details the relevant equations. Section 4 presents the experiments and their results, and Section 5 summarises the conclusions derived from this work.

2. PRELIMINARIES

2.1 Gene Representation

The TInSSel approach was a result of studying GA crossover approaches in a variable length chromosome environment. Due to this variable length facet of the work, the gene encoding utilised is slightly different from conventional GA approaches. Each gene in the chromosome represents an intervention in the chemotherapy schedule and has a schedule location associated with it, while the omission of an intervention represents a non intervention on that day. In the nematode work, each location represented a day of the application schedule, thus if a chromosome had two genes, e.g. (2,16), this represents a schedule whereby the mushroom farmer applies the bio-control agent to their crops on the second and sixteenth day of the schedule. Section 3.1 defines a cancer problem whereby each gene does not represent a potential daily intervention, but groups of 4 genes are used to calculate the strength of dose for a particular day.

2.2 Uniform Crossover

GAs commonly use Uniform Crossover (UC) as a method of achieving recombination. UC is a popular method of gene selection as it avoids the destructive tendencies that Single Point Crossover (SPC) demonstrates in later generations [3]. In addition, UC is effective at protecting common genes and recombining non-common genes [7]. In order to gauge the effectiveness of the directed crossover approaches, we compare their performance to the established UC approach. As defined in Section 2.1, as there is a variable length environment, the gene representations are modified. In this work, UC crossover operates by automatically picking those genes that are present in both parents (i.e. they have a selection probability of 1.0) then each of the remaining genes (those not present in both parents) with a selection probability of 0.5.

2.3 TInSSel

The majority of crossover approaches have the gene selection process separated from the process of generating offspring, however, alternatives have been investigated [11]. TInSSel is a variant length encoding crossover approach that produces children which have an intervention schedule derived from the fitter of the two parents selected for breeding. It was designed to test the principle that targeted intervention selection would improve performance in time-series problems.

2.3.1 TInSSel Algorithm

The first step of the TInSSel process is to select the number of intervention points to be present in each offspring. The fittest parent in the recombination pool is found and the number of intervention points utilized by this parent is noted as (I_F). Although the size of the fitter parent is known, exploration is encouraged in this process by adjusting the number of interventions in the offspring such that they vary around that of I_F. In order to calculate the limits of this variance, we first calculate the absolute difference in the number of interventions, (D_i), between parents as:

$$D_i = |I_1 - I_2| \qquad (1)$$

where I_1 and I_2 are the number of interventions for parents one and two respectively.

For a given offspring, a stochastic element is introduced such that the actual number of target interventions to use I_T, is calculated as:

$$I_T = I_F - \frac{D_i}{2} + rnd(D_i) \qquad (2)$$

I_T is a natural number constrained by the minimum number of interventions I_{min}, which must be applied (usually 1) and a maximum number of interventions I_{max}. I_{max} is limited to the size of the set of interventions present in both parents. The function $rnd(x)$ returns a random real value between 0 and x. I_F acts as the centre point for the mean target intervention level with bounds determined by the difference between the two parent intervention levels.

Having determined the number of interventions a child will have, the next step is to calculate when the interventions will occur. TInSSel ensures that intervention points present in all the parents selected for crossover are passed on to the offspring before interventions present in only some of the parents. Interventions present in all parents selected for breeding are placed in the set of duplicates S_{dup}, of size I_D and these intervention points will have priority in being passed to the offspring (as is the case in UC). Interventions from S_{dup} will be added once only, at random, until I_T is reached or no common intervention points remain. Note that if I_T is less than the size of I_D then not all elements of S_{dup} will be included. Having selected interventions common to both parents, the number of additional interventions required (I_B) is therefore I_T - I_D. I_B will be a value between 0 and I_T since it is possible that there are no duplicate interventions across all parents. To determine the remaining interventions, selection is carried out exactly as UC for genes with 0.5 probability of selection (Section 2.2), until the required number is reached.

2.4 Directed Uniform Crossover

TInSSel has previously been shown to outperform UC when applied to the scheduling of bio-control applications [4, 5, 6]. The TInSSel approach described in Section 2.3 produces a window of potential intervention schedule sizes based on the number of interventions present in the fitter parent and the difference in intervention numbers between both parents selected for crossover. Although TInSSel has proven that directed intervention can be effective, the question remains as to whether this is due to the sizing window approach of TInSSel, or indeed purely through providing a target number of interventions for crossover to aim for? In order to investigate this question, Directed Uniform Crossover (DUC) is presented as a simple technique that provides a target number of interventions to be picked during the crossover process. Rather than calculating a window of potential intervention numbers as with TInSSel, DUC simply finds the fittest parent in the recombination pool and the number of intervention points utilized by this parent is noted as (I_T). As with the UC approach detailed in Section 2.2, all genes present in both parents are added to the offspring and if more material is still required, it is picked at random from those interventions present in only one of the parents until I_T is reached.

2.5 Fitness Directed Search

Although TInSSel uses the difference in intervention numbers between parents, it does not take into account the fitness difference. The Fitness Directed Search approach aims to utilise both these properties to calculate the number of interventions to use in offspring. The FDS approach places

an emphasis on selecting intervention sizes that are close to the size of the fitter parent, while shifting in the direction that appears to offer the best improvement, based on the intervention and fitness gradients between the two parents. For example, if the fitter parent has more interventions than the less fit parent, then an intervention size will be returned that is greater than the fittest parent with a margin that is proportional to this fitness difference.

The FDS algorithm is presented with two parents for selection. F_1 is the normalised fitness associated with parent one and I_1 is the number of interventions utilised by parent one, with F_2 and I_2 being the respective values for parent two.

The normalised fitness score, F_{norm} is calculated by finding the maximum (F_{max}) and minimum (F_{min}) fitness scores contained in the current population and applying Equation 3, where F is the score being normalised.

$$F_{norm} = \frac{F - F_{min}}{F_{max} - F_{min}} \qquad (3)$$

The number of interventions used by the fitter of the two parents is recorded as I_F and the normalised optimal fitness score is recorded as T, where $T=0$ for a minimisation problem and $T=1$ for a maximisation problem.

The number of interventions to select in the offspring I_T, is calculated as shown in Equation 4.

$$I_T = I_F + (2T - 1)(I_1 - I_2)(F_1 - F_2) \qquad (4)$$

I_T incorporates both the fitness and size difference between parents and therefore provides an intervention estimate that captures the dynamics of the relationship between the parents fitness and intervention values. Note that if there is no difference between the number of interventions in the parents or if the fitness score for parents are the same, the calculation of I_T is equivalent to the DUC approach.

2.6 Mutation

Mutation is a key component of GAs and prevents stagnation in the population by introducing random diversity [18]. We have investigated 3 different levels of mutation in this work in order to understand the ability of the algorithms to exploit this diversity.

A relatively standard form of mutation is used where the bit string of interventions is first padded out with zeros up to the maximum number of interventions. This expanded string, representing the chromosome, is traversed and bits are randomly flipped according to a probability m (in this study values of m are 0, 0.001 and 0.01). If this process is repeated on a bit string with no external factors, the odds of an individual bit being set will therefore tend toward 0.5, resulting in a string with half the bits set to 1 and the other half set to 0. This form of mutation introduces a tendency to add genes when the number of interventions is low with respect to $L/2$ and conversely, to remove genes when the interventions is greater than $L/2$, where L is the bit length of a chromosome.

3. CANCER CHEMOTHERAPY PROBLEM

Chemotherapy is frequently used to combat cancer whereby chemotherapy drugs interfere with the division of cancerous cells in order to cause the cancer to recede [12]. Construct-

ing an effective chemotherapy treatment schedule is a non-trivial task; indeed chemotherapy is often considered one of the most complex cancer treatments [20]. This complexity is due to the range of drugs available and other factors such as toxicity levels in the patient caused by treatments. Through the use of deterministic mathematical models, valuable efforts in the analysis of cancer chemotherapy have been made [9]. However, the problem can be regarded as analytically intractable due to both its multi-constraint nature and the non-linearity of the optimisation functions [13]. This complexity makes this problem an ideal test to assess the abilities of DUC in searching a multi-constraint, extensive search space where GAs have been previously successful in calculating treatment schedules [14].

3.1 Problem Formulation

We define the problem formulation of the single cancer chemotherapy model from [10]. There exists a solution space S of chemotherapeutic regimens where the representation space I of these solutions is defined as a Cartesian product, as shown in Equation 5.

$$I = A_1 \times A_2 \times \ldots \times A_n \qquad (5)$$

Where $A_i, i = \overline{1, n}$ are the allele sets which encode the concentration levels $C_i, i = \overline{1, n}$ of the anti-cancer drug in the blood plasma. A 4-bit representation for each concentration C_i is used, meaning that all allele sets $A_i, i = \overline{1, n}$ consist of 16 elements as defined in Equation 6.

$$A_i = \{0000, 0001, 0010, 0011, \ldots, 1110, 1111\} \qquad (6)$$

This means that each concentration C_i has an integer value in the range 0 to 15 and any treatment schedule $c = (C_i), i = \overline{1, n}$ can be represented as a $4n$-bit chromosome $x \in I$ as detailed in Equation 7.

$$x = \{a_1 a_2 a_3 \ldots a_{4n} : a_k \in \{0, 1\} \forall k = \overline{1, 4n}\} \qquad (7)$$

The target for these equations is to minimise the final tumour size $N(T_{final})$ after a fixed treatment period $[T_0, T_{final}]$. Thus the objective of the search is to find the treatment schedule $c = (C_i | i = \overline{1, n})$ that minimises $N(T_{final})$. The tumour growth is defined by the Gompertz model as shown in Equation 8. This shows how the population of tumour cells of size N at time t will grow at a rate λ and the proximity of the current size N to an absolute limiting size θ.

$$\frac{dN}{dt} = N(t)\lambda ln\frac{\theta}{N(t)} \qquad (8)$$

The concentration of anti-cancer drugs relates to their ability to kill cells, as shown in Equation 9 where N is the number of tumour cells at time t, $C(t)$ is the drug concentration at time t and κ is the toxicity of the drug.

$$\frac{dN}{dt} = -\kappa C(t)N(t) \qquad (9)$$

Equations 8 and 9 are combined to form the differential equation shown in Equation 10, which represents the tumour response to chemotherapy. Each treatment is evaluated through passing its encoded dose schedule to a simulation of the response based on Equation 10, with the fitness

Parameter	Value
λ	0.1
C_{cum}	120
κ	0.045
θ	100.484

Table 1: Chemotherapy Model Parameters

score being updated during the simulation run. The final score at the end of the run forms the fitness function for the experiments undertaken in this work. The parameters used in these expressions for the experiments carried out in this work are shown in Table 1.

$$\dot{N}(t) = N(t)\lambda ln\left(\frac{\theta}{N(t)}\right) - \kappa C(t)N(t) \qquad (10)$$

This model is subject to the constraints detailed in equations 11 - 13 whereby equations 11 and 13 are for all $i = \overline{1,n}$.

$$g_1(c) = C_{max} - C_i \geq 0 \qquad (11)$$

$$g_2(c) = C_{cum} - \sum_{i=1}^{n} C_i \geq 0 \qquad (12)$$

$$g_1(c) = N_{max} - N(t_i) \geq 0 \qquad (13)$$

Equation 11 details the maximum instantaneous dose C_{max} for the drug, Equation 12 the maximum cumulative C_{cum} dose of the drug and Equation 13 the maximum permissible size N_{max} of the tumour. The goal of cancer chemotherapy is to achieve the beneficial effects of treatment without violating the above constraints [16].

4. EXPERIMENTS & RESULTS

Tournament selection was used to select parents for breeding as it has been shown to provide better or equivalent convergence and computational properties when compared to alternative approaches [2]. There were 2 potential parents in each tournament and the GA maintained a steady state population update. This allows the offspring to immediately be used as part of the breeding pool and has been shown to make a shift towards the optimal solution in an early part of the optimisation process [19]. The other parameters for the GA are shown in Table 2. The cancer chemotherapy schedules were 400 bits long, and as described in section 3.1, there are 4 bits for each chemotherapy dose which is equivalent to a 100 dose schedule. The experiments used the same parameters with the only exception being the crossover approach used and mutation probability.

As one cannot assume that data of this type are normally distributed, a nonparametric statistical test is required. We have used the Kruskal-Wallis (KW) one-way analysis of variance by ranks test to discern whether the samples are from different populations.

A KW probability less than 0.05 indicates that at least one of the groups is different from at least one of the others. For the results presented in this paper, the KW probabilities for all studies were at most 0.00006, indicating that a statistical difference was present between at least one group and one other group.

Parameter	Value
Population Size	100
Fitness Function Limit	10,000
Crossover Probability	1.0
Mutation Probability m	0, 0.001, 0.01
Chromosome Length	400

Table 2: GA Parameters

This test does not identify which groups are different, thus in order to distinguish this information, the formula detailed in Equation 14 was used [17]. This allows us to test the hypothesis $H_0 : \theta_u = \theta_v$ against $H_0 : \theta_u \neq \theta_v$ between two groups u and v.

$$|\bar{R}_u - \bar{R}_v| \geq Z_{\alpha/k(k-1)}\sqrt{\frac{N(N+1)}{12}(\frac{1}{n_u} + \frac{1}{n_v})} \qquad (14)$$

$k = $ *number of samples or groups*
$n_j = $ *number of cases in the jth sample*
$N = $ *number of cases in the combined sample*
$R_j = $ *sum of the ranks of the jth sample*
$\bar{R}_j = $ *average of the ranks of the jth sample*
$\bar{R} = $ *average of the ranks in the combined sample*

For each evaluation, as there are k = 4 groups, we have 4(4-1)/2 = 6 comparisons. An $\alpha = 0.05$ was used resulting in a critical value of z = 2.63. N is 800, with n_u and n_v both 200. Thus when two samples are compared, if the resulting score is greater than 60.97, $H_0 : \theta_u \neq \theta_v$ can be accepted, showing a statistically significant difference between samples.

In this study, 200 runs of each crossover approach were conducted with the 3 different mutation probabilities. Each run was recorded up to 10,000 Fitness Function Evaluations (FFE) in order to gauge the performance of the crossover approaches over time.

The results for each of these experiments are shown in Figures 1 to 3. Each figure shows the median performance of the relevant crossover approach at a given point in time (as measured by FFEs) with a set level of mutation, where a high score indicates an effective treatment schedule. Since data of this type are rarely normally distributed, all the graphs plot the median values from the runs with the error bars showing the first and third quartiles.

For each experiment, mean rank tables are provided for each group (Tables 3 to 5), sampled at 1000 FFE intervals. Comparison of the mean rank differences in these tables indicates a significant difference between approaches when the mean rank difference is greater than 60.97.

4.1 No mutation

For the case where no mutation is present (Figure 1), all four crossover approaches reach a point where no further improvements can be made. It is interesting to note that the relatively simple DUC approach produces the best results in this case since it just uses the best parent's intervention level as the guide for offspring. Since UC is unguided, it has no means of adjusting its current intervention position and, as will shown in the next section, relies purely on mutation to find the best intervention level to use. The TInSSel and FDS approaches attempt to move the intervention level

based upon the relative differences in the parent intervention and fitness scores. At the start of the evolutionary process, there is some variance in the population and progress is made, however this variance quickly disappears and the population stagnates. With no variance to exploit, TInSSel and FDS also cease to improve. Note also that since both of these approaches tend to vary around the current fittest parent's intervention level, they are slower to get 'off the mark' than DUC.
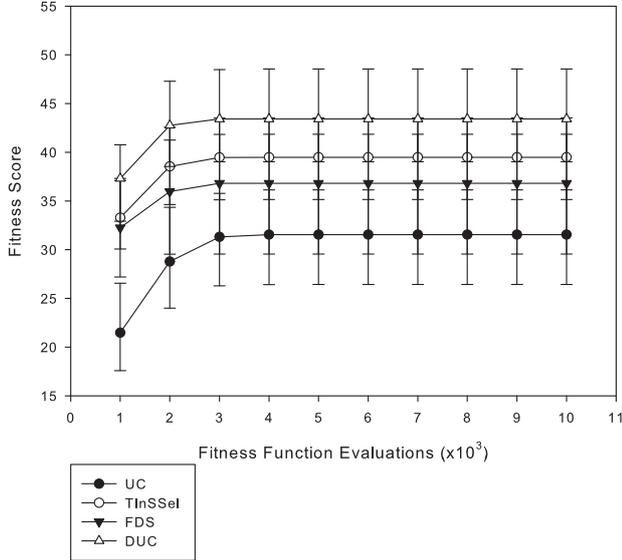
Figure 1: Fitness scores with no mutation

FFEs	UC	TInSSel	FDS	DUC
1000	151.54	468.20	415.03	567.24
2000	208.45	457.79	371.19	564.57
3000	231.00	455.31	357.99	557.71
4000	235.31	453.97	356.27	556.46
5000	235.39	453.96	356.22	556.44
6000	235.39	453.96	356.22	556.44
7000	235.39	453.96	356.22	556.44
8000	235.39	453.96	356.22	556.44
9000	235.39	453.96	356.22	556.44
10000	235.39	453.96	356.22	556.44

Table 3: Mean ranks with no mutation

4.2 Mutation of 0.001

Figure 2 shows the results for a mutation level of 0.001. All four crossover approaches are seen to perform significantly better than in the previous case and all tend toward a near optimal performance given sufficient time. The main advantage of the directed approaches appears to be their relative efficiency at finding a good solution in a given time. UC is now able to use the variance introduced by this level of mutation to return better solutions when compared to its previous performance where no mutation was present. FDS and TInSSel are initially outperformed by DUC but are able to regain this ground in the later stages and maintain a slight advantage. It is also worth noting that even

when FFEs equals 10,000, all the directed approaches still statistically outperform UC (Table 4).
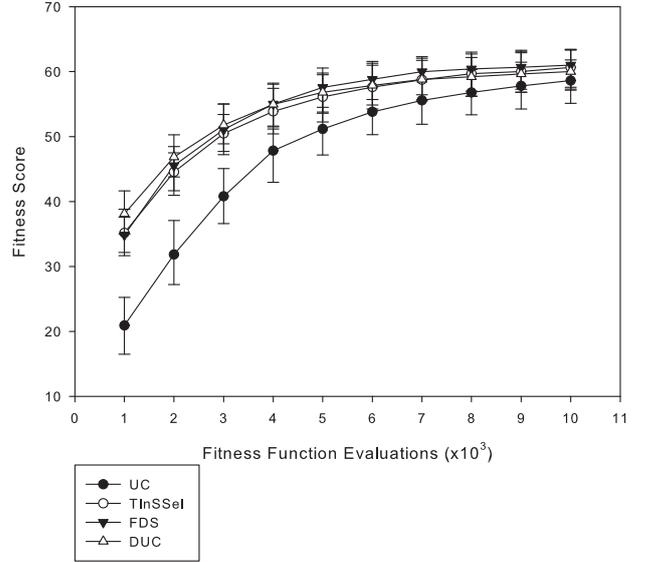
Figure 2: Fitness scores for mutation of 0.001

FFEs	UC	TInSSel	FDS	DUC
1000	113.72	457.24	465.19	565.86
2000	136.73	444.30	475.23	545.75
3000	160.07	446.74	483.00	512.20
4000	200.62	435.77	480.46	485.16
5000	230.85	427.64	483.91	459.61
6000	261.10	425.52	478.00	437.39
7000	281.83	423.84	469.01	427.32
8000	304.41	421.34	457.62	418.64
9000	322.20	420.55	445.74	413.52
10000	336.10	423.76	434.87	407.28

Table 4: Mean ranks with mutation of 0.001

4.3 Mutation of 0.01

Figure 3 shows the results for a mutation level of 0.01. The performance of the four crossover approaches actually deteriorate in this case, with UC performing significantly worse than the directed approaches. Since the level of noise in a given chromosome is likely to be relatively high, the DUC approach will be targeting intervention levels that are potentially erroneous. In contrast, the FDS approach is able to use the fitness and intervention gradient between the two parents to calculate a suitable level of interventions to aim for.

In response to the question raised in section 2.4, we can see that from 2,000 FFEs onward, there is a statistically significant difference between the searching approaches of TInSSel and FDS versus DUC. TInSSel stochastically searches around the point used by DUC and FDS works out the correct direction to move from the value used by DUC. These results would seem to indicate that it is important to vary intervention levels around the currently observed best intervention level if the system is to remain robust to noise.
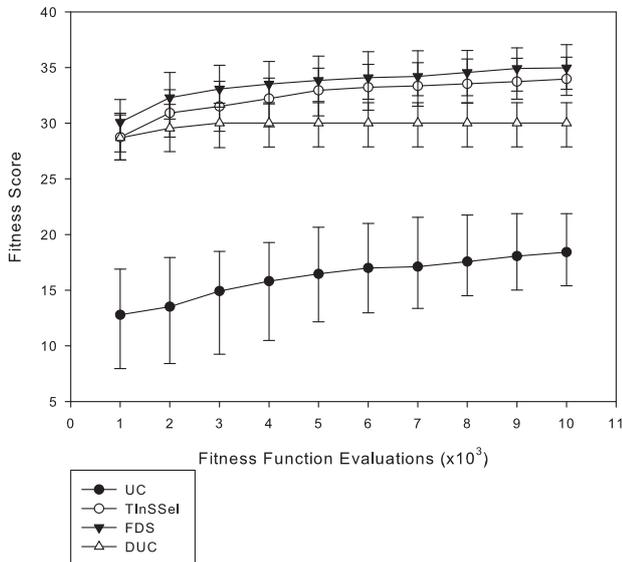
Figure 3: Fitness scores for mutation of 0.01

FFEs	UC	TInSSel	FDS	DUC
1000	104.17	482.90	536.33	478.61
2000	103.70	493.27	577.02	428.01
3000	103.19	501.20	587.77	409.86
4000	104.47	513.67	587.42	396.45
5000	105.59	526.19	586.22	384.01
6000	105.76	534.19	587.18	374.88
7000	105.57	538.53	590.59	367.32
8000	105.45	545.96	589.33	361.28
9000	105.37	546.97	595.69	353.98
10000	105.25	550.93	597.55	348.28

Table 5: Mean ranks with mutation of 0.01

5. CONCLUSION

Section 4 reports our results comparing UC, TInSSel, FDS and DUC at deriving cancer chemotherapy treatment schedules. The experiments show a statistical advantage in using a directed intervention technique over traditional UC regardless of the fitness function evaluation point being observed. The directed approaches are quicker than UC at finding good scores and appear to be able to more effectively exploit the diversity introduced by mutation. Further investigation will focus on time-series problems which offer a finer granularity of intervention time in order to assess the potential benefits of directed crossover techniques in a more complex environment.

6. ACKNOWLEDGMENTS

The authors would like to thank Catherine Howie of the University of Stirling for her statistical advice.

7. REFERENCES

[1] P. Collard, C. Escazut, and A. Gaspar. An evolutionary approach for time dependant optimization. In *Int. Conf. on Tools for Artificial Intelligence 96*, pages 2–9. IEEE Computer Society Press, 1996.

[2] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[3] L. J. Eshelman, R. A. Caruana, and J. D. Schaffer. Biases in the crossover landscape. In *Proceedings of the third international conference on Genetic algorithms*, pages 10–19, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[4] 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*. IEEE press, 2007.

[5] 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.

[6] P. M. Godley, J. Cowie, and D. E. Cairns. Novel genetic algorithm approaches for time-series problems. In *Doctoral Symposium on Engineering Stochastic Local Search Algorithms*, pages 47–51, Brussels, Belgium, 2007. IRIDIA.

[7] X. Hu and E. A. D. Paolo. An efficient genetic algorithm with uniform crossover for the multi-objective airport gate assignment problem. In *IEEE CEC 2007: Proceedings of the IEEE Congress On Evolutionary Computation*. IEEE press, 2007.

[8] R. Martin and K. L. Teo. *Optimal control of drug administration in cancer chemotherapy*. World Scientific Pub Co Inc, Singapore, 1994.

[9] A. S. Matveev and A. V. Savkin. Optimal control applied to drug administration in cancer chemotherapy: the case of several toxicity constraints.

[10] J. McCall, A. Petrovski, and S. Shakya. Evolutionary algorithms for cancer chemotherapy optimization. In

G. B. Fogel, D. W. Corne, and Y. Pan, editors, *Computational Intelligence in Bioinformatics*, chapter 12, pages 265–296. Wiley IEEE Press, 2008.

[11] C. K. Mohan. Selective crossover: towards fitter offspring. In *SAC '98: Proceedings of the 1998 ACM symposium on Applied Computing*, pages 374–378, New York, NY, USA, 1998. ACM.

[12] G. Ochoa, M. Villasana, and E. K. Burke. An evolutionary approach to cancer chemotherapy scheduling. *Genetic Programming and Evolvable Machines*, 8:301–318, December 2007.

[13] A. Petrovski. *An Application of Genetic Algorithms to Chemotherapy Treatments*. PhD thesis, The Robert Gordon University, Aberdeen, Scotland, December 1998.

[14] A. Petrovski and J. McCall. Computational optimisation of cancer chemotherapies using genetic algorithms. In *Proceedings of Workshop on Recent Advances in Soft Computing, Soft Computing Techniques and Applications*, pages 117–122, 1999.

[15] A. Petrovski, S. Shakya, and J. McCall. Optimising cancer chemotherapy using an estimation of distribution algorithm and genetic algorithms. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 413–418, New York, NY, USA, 2006. ACM.

[16] A. Petrovski, B. Sudha, and J. McCall. Optimising cancer chemotherapy using particle swarm optimisation and genetic algorithms. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 633–641, Heidelberg, Germany, 2004. Springer.

[17] S. Siegel and N. J. C. Jr. *Nonparametric Statistics for The Behavioral Sciences*. McGraw-Hill, New York, NY, USA, 1988.

[18] W. M. Spears. Crossover or mutation? In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 221–237. Morgan Kaufmann, San Mateo, CA, 1993.

[19] F. Vavak and T. C. Fogarty. A comparative study of steady state and generational genetic algorithms for use in nonstationary environments. In *Proc. of the Society for the Study of Artificial Intelligence and Simulation of Behaviour Workshop on Evolutionry Computing 96*, pages 301–307, 1996.

[20] T. Wheldon. *Mathematical models in cancer research*. IOP Publishing Ltd., Bristol, 1988.