Directed Intervention Crossover applied to Bio-Control Scheduling

Paul M. Godley, David E. Cairns, and Julie Cowie

Abstract—**This paper describes two directed intervention crossover approaches that are applied to a bio-control dynamic system. Unlike traditional uniform crossover, both the Calculated Expanding Bin (CalEB) method and Targeted Intervention with Stochastic Selection (TInSSel) approach actively choose an intervention level and spread based on the fitness of the parents selected for crossover. Results indicate that these approaches lead to significant improvements over Uniform Crossover (UC) when a penalty is introduced for each intervention point used by the crossover algorithm.**

I. Introduction

In optimization of intervention schedules for models of dynamic systems, Genetic Algorithms (GAs) commonly use Uniform Crossover as a method of achieving recombination [1]. A sample solution for such a system is encoded as a bit string that spans the time period of the process, where a '1' indicates that an intervention should take place at a given time interval and a '0' indicates no action should be taken. Although the standard UC approach produces good solutions, it can take time to find the initial parameters that lead to these solutions. In particular, although we may know that it is preferable to minimize the number of interventions in a process due to side effects or costs associated with an intervention, it is not initially known how few interventions will achieve the best overall result.

A typical approach to solving this type of problem is to seed a GA with a broad spread of initial random solutions, where the number of interventions in the initial population varies widely across the intervention period. A penalty for each intervention used is then introduced to a fitness function, where the fitness function is driving solutions towards using minimal interventions. This approach does eventually lead to a good solution but can require an excessive number of generations for a GA to converge, since it is difficult for the algorithm to determine which of the intervention points are valid and how many are still required for an effective solution.

Our research proposes two novel crossover approaches that actively limit the number of intervention points to use when producing offspring via recombination. The number of interventions is directed by the fitness of the parents used in the selection process.

The approaches are tested using a dynamic model of the interaction of the nematode worm *Steinernema feltiae* on a population of sciarid flies infecting a mushroom crop. The nematode worm is used as a bio-control agent to eliminate sciarid fly larvae and thus reduce damage to a mushroom crop. The challenge is to find the optimal number of nematode worm doses required to adequately suppress sciarid fly levels. The model of a bio-control agent is used to test our approaches since the authors are familiar with its dynamics. However, we believe our techniques to be applicable across many domains. In view of this, current research is focused on evaluating the use of the approach for cancer chemotherapy scheduling [2].

In Section II, we begin by describing the background to the nematode bio-control agent problem and the equations used in the modeling of this domain. In Section III the proposed novel crossover approaches and the specifics of their implementation are presented. Section IV details the experiments undertaken and Section V reports our results to date. Finally, the conclusions and direction of future research are discussed in Section VI.

II. Problem Background

In the field of mushroom farming, one of the principal constraints to the quantity of mushrooms produced is the presence of sciarid flies. Sciarid larvae are known to feed on the mycelium in the casing layer of mushrooms causing crop production to significantly decline. One control agent known to be lethal to sciarid flies is the nematode worm *Steinernema feltiae.* As such, application of the nematodes to mushroom crops acts as an excellent defence mechanism. The principal aim of the farmer is to maximize the profit of crop production. This would appear to be achieved through maximizing the use of the bio-control agent. There is however a financial constraint to the use of the nematodes, therefore, the number of actual intervention treatments needs to be kept to a minimum. It is the optimization of this intervention schedule that forms the fitness function for the experiments undertaken in this work.

A. *Problem Formulation*

In order to successfully resolve this optimization problem, treatment schedules need to be derived that minimize the sciarid larvae population, whilst also maintaining an acceptably low level of intervention points and nematodes used. These potential schedules can then be reviewed by the

Manuscript received March 15, 2007.
P. M. Godley is with the Department of Computing Science and Mathematics, University of Stirling, UK (phone: +44 1786-467441; fax: +44 1786-467551; e-mail: pgo@ cs.stir.ac.uk).
D. E. Cairns and J. Cowie, are with the Department of Computing Science and Mathematics, University of Stirling, UK (e-mail: {dec,jco}@ cs.stir.ac.uk).

farmer to make an informed decision, based on their priorities and preferences, regarding which approach to implement.

In [3], a generalized model for the lifecycle of sciarid flies is specified which includes potential infection from *Steinernema feltiae*. In this model, there is a set of discrete intervention points ($t_1, t_2, \ldots, t_n$) where nematodes can be applied to the system (Equations 1-12). The key optimization question is at which points in the model should the nematodes be applied to maximise their effect on the sciarid fly population. We have used the equations specified in [3] to model the effects of proposed intervention schedules. Table I defines the parameters and constants used in the equations of our model.

TABLE I. LIST AND DEFINITION OF MODEL PARAMETERS

Parameter	Description	Constant Value
E,L,P,A	Egg/larval/pupal/adult host density	Variable
N	Free living infective nematode density	Variable
$T_{E,L,P,A}$	Duration of egg/larval/pupal/adult stage	5, 10, 5, 8 (days)
$\delta_{E,L,P,A}$	Daily mortality rate of egg/larvae/pupae/adult	0.35, 0.125, 0.1, 0.275
ρ	Viable eggs laid per adult pest	150
β	Infection rate of nematodes	0.000095
Λ	Nematodes produced per infected host	2000
μ	Mortality rate of nematodes	0.7
T_I	Delay between infection and lysis of cadavers	12 days
A0	Initial density of adult pests	4.5
T_a	Time of adult invasion	7 days

The implemented model assumes that the infection process is about to start and the adults are ready to lay eggs. Eggs die at a rate of δ_E with those eggs that survive developing into larvae after T_E days. Larvae die at a rate of δ_L and those remaining after T_L days pupate. The pupae die at rate δ_P and after T_P days the remaining pupae subsequently turn into adults and lay more eggs for the duration of their T_A lifespan. Thus the full life cycle of the sciarid flies takes place over $T_E + T_L + T_P + T_A$ days. A diagrammatical overview of this lifecycle is shown in Figure 1.

Equations (1 - 12) are used to model the dynamics of the nematode / sciarid populations. Equations (1 - 5) model the change in eggs, larvae, pupae, adults and nematodes respectively from one time step to the next. Equation (6) defines the probability of surviving the egg stage, (7) the probability of surviving the pupae stage and (8), the probability of surviving the adult stage. Equation (9) defines the transfer rate into the egg stage. Equation (10) calculates the duration of the adult host stage, (11) the maturation rate from the larval stage and (12) the probability of surviving the larval stage.

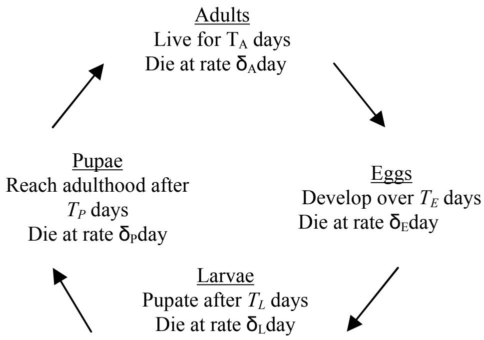

Fig. 1. The sciarid fly lifecycle. The full life cycle of the sciarid fly takes place over $T_E + T_L + T_P + T_A$ days.

$$\frac{dE(t)}{dt} = R(t) - R(t - T_E)\sigma_E - \delta_E E(t) \quad (1)$$

$$\frac{dL(t)}{dt} = R(t - T_E)\sigma_E - M(t) - \delta_L L(t) - \beta N(t) L(t) \quad (2)$$

$$\frac{dP(t)}{dt} = M(t) - \delta_P P(t) - M(t - T_P)\sigma_P \quad (3)$$

$$\frac{dA(t)}{dt} = M(t - T_P)\sigma_P - M(t - T_P - T_A)\sigma_P \sigma_A - \delta_A A(t) \quad (4)$$

$$\frac{dN(t)}{dt} = \Lambda \beta N(t - T_I) L(t - T_I) - \mu N(t) - \beta N(t) L(t) \quad (5)$$

$$\sigma_E = e^{-\delta_E T_E} \quad (6)$$

$$\sigma_P = e^{-\delta_P T_P} \quad (7)$$

$$\sigma_A = e^{-\delta_A T_A} \quad (8)$$

$$R(t) = \frac{\rho}{T_{AVE}} A(t) \quad (9)$$

$$T_{AVE} = [1 - \sigma_A] / \delta_A \quad (10)$$

$$M(t) = R(t - T_E - T_L)\sigma_{L(t)}\sigma_E \quad (11)$$

$$\sigma_{L(t)} = e^{-\int_{t - T_L}^{t} \beta N(x) + \delta_L \, dx} \quad (12)$$

The fitness calculation for the model is detailed in Equation 13. The fitness score is a count of all the larvae present in the system throughout the duration of the modelling process plus the penalty of intervening multiplied by the number of interventions. Since the aim of the intervention schedule is to reduce the total number of sciarid fly larvae in the crop, the optimal treatment schedule will be that which returns a fitness score closest to zero. The nematode control agent is only effective against sciarid flies when they are in their larval stage, thus the nematode intervention schedule needs to maximize the impact on this particular stage of the fly's development.

Fitness function $$F = \sum_{t=0}^{T} L(t) + NP \qquad (13)$$

T =Number of time steps
L(t) = larvae in existence at time t
N=Number of interventions used
P=Penalty per intervention

III. CROSSOVER APPROACHES

The aim of this research is to investigate whether incorporating the number of interventions used by parents with relatively high levels of fitness could be used to effectively drive the crossover process. From this perspective, we derived two crossover approaches that we refer to as Calculated Expanding Bin (CalEB) and Targeted Intervention with Stochastic Selection (TInSSel) respectively. CalEB and TInSSel both provide mechanisms for crossover of variable and fixed length chromosome schedule encodings. They differ, however, in their method of selecting interventions for recombination.

A. Gene Encoding

Many chromosome encodings of scheduling problems utilize one gene per available intervention time, such as allowing one gene for each day of an intervention schedule [4]. An intervention gene is simply an integer value defining the time in the schedule to perform an intervention. The only other information required by a GA is the strength of dose to apply per intervention. In order to compare the CalEB and TInSSel algorithms with standard uniform crossover, this dosage strength was fixed for all methods.

B. The CalEB Crossover Process

CalEB is a novel crossover approach that as with other GA crossover techniques, produces children from the genetic information contained within the parent chromosomes that are selected for breeding. The majority of evolutionary algorithms have the selection process completely separated from the process of generating offspring [5], CalEB aims to extend upon this by using the number of interventions in the fitter parent to derive an optimal number of interventions for the offspring.

The first step of the CalEB 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 utlized by this parent is used to derive the target intervention level (I_T). Although the size of the fitter parent is known, exploration is encouraged in this process by adjusting the number of interventions in the offspring towards that of the fitter parent, while still permitting some degree of variance around this number.

In order to calculate I_T, the absolute difference in number of interventions between the parents D_i is calculated:

$$D_i = |I1 - I2| \qquad (14)$$

where I_1 and I_2 are the number of interventions for parents one and two respectively. In addition, the actual number of interventions used by the fitter of the two parents is recorded as I_F. I_T is calculated as:

$$I_T = I_F - \frac{D_i}{2} + rnd(D_i) \qquad (15)$$

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. Note that when I_F is odd, it is always rounded up to the nearest integer value.

To illustrate the approach, we provide the following minimization example, hereafter referred to as example A:

Parent 1 (P_1): Interventions *(1,3,5,7,9,12),* I_1=6, Score=0.8
Parent 2 (P_2): Interventions *(1,5,9,10),* I_1 = 4, Score=0.6

In example *A* above, P_2 is the fitter parent with a score of 0.6. This information, combined with D_i will produce offspring with between 2 and 6 interventions.

Having determined the number of interventions a child will have, the next step is to calculate when the interventions will occur. CalEB 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. In example A, both P_1 and P_2 share a set of common intervention points $S_{dup,}$ where S_{dup} contains (1, 5, 9) and is of size $I_{D,}$ where I_D=3. As such, these intervention points will have priority in being passed to the offspring (as is the case in uniform crossover). 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 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, we amalgamate all remaining interventions from all parents into an ordered list. The ordered list will have I_A interventions as defined in Equation 16:

$$I_A = \sum_{p=1}^{P} I_p - (NI_D) \quad (16)$$

P=Population size
N = Number of parents in the recombination pool
I_p = Interventions in parent p

We now use I_B to split the ordered list into a set of bins where each bin will contain I_A/I_B entries. The aim of using a set of bins from which to pick further interventions is to ensure interventions are evenly selected from the set of remaining alternatives. We require I_B further interventions and have therefore divided the number of interventions we have left by I_B. We proceed to visit each bin containing these intervention points and select one intervention point from the bin based on its prevalence in the bin. Each intervention has a probability of selection associated with it, which is equal to the number of times it occurred in the parents, divided by the number of parents.

Using example A, we would have a set of remaining candidate intervention points (S_C) containing (3,7,10,12). If $I_T = 5$, we would have added all the members of S_{dup} to the set of interventions for the offspring (S_O) and I_B would be equal to 2. We therefore would divide S_C into 2 bins since 2 interventions are still required, and pick a gene at random from each bin. Given this scenario, possible outcomes for S_O could be (1,5,9,3,12) or (1,5,9,7,10).

C. The TInSSel Crossover Process

As detailed in Section III B, CalEB is different from the UC approach in two ways: It uses parent information to direct the size of children and it selects relevant genetic information from an evenly spread set of bins. In order to assess the contribution of these factors to any results obtained, a second approach, TInSSel, was developed. The TInSSel algorithm is a hybrid approach. It collates intervention information and uses parent sizes to determine children sizes identically to the CalEB approach. However, a different process of selecting appropriate genetic material from the parents is used. As with CalEB, all material present in both parents is incorporated into the child, however, rather than binning the remaining interventions and then selecting from each bin, a random set of interventions is made from those remaining in the parent intervention pool, until the required number is reached. Developing this new approach allows us to differentiate between any advantages gained from incorporating parent intervention levels (used by CalEB and TInSSel but not UC) and those benefits derived from the binning process (used by CalEB but not TInSSel or UC).

Using example A, we would have a set of remaining candidate intervention points (S_C) containing (3,7,10,12). If $I_T = 5$, we would have added all the members of S_{dup} to the set of interventions for the offspring (S_O) and I_B would be equal to 2. Two interventions would therefore be picked at random from S_C. Given this scenario we may produce the same outcomes for S_O as CalEB however it is possible that the selected interventions may not be as evenly spread. For example, the following outcomes are permissible: (1,5,9,3,7) or (1,5,9,10,12) .

IV. Experiments

A. Uniform Crossover

In order to gauge the effectiveness of CalEB and TInSSel, we compare their performance against the established UC approach. In UC, for each gene required for the child, a parent is chosen at random from the parent set and the relevant gene is copied. UC is a popular recombination method as it avoids the destructive tendencies that Single Point Crossover (SPC) demonstrates in later generations [6]. The UC algorithm can be summarized as follows: For gene 1 of a child, a parent is chosen at random from the parent pool and its gene 1 value copied to the child's gene 1 value. For gene 2 of a child, another randomly selected parent is chosen from the parent pool and its gene 2 value copied to the child's gene 2 value. This process is repeated until the child contains the required number of genes.

B. GA Parameters

All three crossover approaches were tested with the same GA parameters and architecture, the only difference between the tests being the form of crossover used. This facilitated a fair comparison of the different crossover techniques. The GA parameters are detailed in Table II. The values for the sciarid fly model are based on the best fit parameters defined in [3], as they were empirically proven as robust and accurate when compared to independent field trials (shown in Table I).

TABLE II. Parameters for GA

Parameter	Value	Parameter	Value
Population size	50	Crossover probability	1
Number of parents	2	Mutation probability	0.05
Number of children	2	Days in nematode schedule	50
Generations to perform	50	Max nematodes/ intervention	1000

Tournament selection was chosen as the parent selection strategy as it has been shown to provide better or equivalent convergence and computational complexity properties when compared to alternative approaches [8]. In order to allow for maximum interchange of genetic material between solutions, a crossover rate of 1.0 was used (i.e. crossover will always occur). The effect of mutation in this system is outlined in Section IV E.

C. Penalty Factors

Each intervention in the schedule represents the mushroom farmer having to spend time and resources in spraying the crops. This information is incorporated in the model as a penalty value associated with each intervention. This encourages the system to use as few interventions as possible while also deriving as efficient a treatment as possible. This value was varied depending upon the required experimental design. Its value is given for each experiment.

D. Initial Population Spread

The starting populations for the experiments are initialized with between 1 and M interventions at random. In the models used, M was incremented per run from 1–50 for 50 day schedules. For example, when M=10, each member of the starting population has between 1 and 10 interventions scattered across the 50 day intervention period. The aim of the experiments was to determine how each of the three crossover methods used this initial population spread to search for a solution. This reflects a scenario where the decision maker is trying to determine the optimal number of interventions to use, and is unsure of the sample population to use. It also demonstrates how the initial variance in the population affects the robustness of each approach and its use of the population gene pool to find a good solution over a given number of generations.

E. Mutation

Mutation is a two-stage process. In the first stage, mutation is evaluated on a gene by gene basis. A random number between 0 and 1 is generated and if it is less than or equal to the mutation probability, mutation is applied. The intervention value of the gene to be mutated is randomly changed to a new, unique value.

For the second stage, another random number between 0 and 1 is generated. If this random value is less than the probability of adding a gene (0.05), there is a 0.5 probability of a gene being added as long as the current number of genes plus 1 remains less than the maximum genes allowed. If a gene is not added, an intervention point is picked at random and removed from the child, as long as this does not reduce the chromosome to containing zero genes.

V. Results

The following section outlines the experiments performed. For each experiment, two figures and a table are provided. The first figure shows the effect on the best fitness score achieved (where the optimal score would be 0) when the initial number of interventions is between 1 and the value defined on the x axis (see Section IV D). The second figure shows the number of interventions used to gain the given fitness score for the same set of results. The table provides the mean best score, standard deviation, standard error and both the 95% and 99% confidence intervals for the case of M=50; the rightmost point in each graph.

For each initial intervention spread, 500 runs were completed in order to provide an adequate sample of performance to account for the stochastic nature of each individual run. These results were then averaged to provide a clearer overview of the underlying statistics. The results for these experiments are shown in Figures 2-11, where each sample point on the lines in these graphs is the average of 500 runs for the given settings. The starting population spread, selection and mutation process for these experiments is constant for all crossover approaches, with the only difference between runs being the crossover technique used. These graphs show how varying the initial intervention level allows us to gain an understanding of the search profile carried out by each approach. Tables III–VII show the statistics for each experiment when the maximum intervention parameter M, is set to 50.

A. Zero Intervention Penalty

In Section IV C, we described the penalty factor associated with interventions. For the first set of experiments, no penalty factor was applied in order to check the performance of the three approaches on a problem where CalEB and TInSSel should not benefit from their intervention selection process. In principle, all three approaches should perform in a similar manner. In the absence of a penalty, CalEB and TInSSel are not able to actively search for an optimal intervention size. In this experiment, the penalty was 0, and the GA ran for 50 generations.

Figure 2 shows that all three approaches follow a similar trend. Figure 3 shows that when no penalty is associated with the number of interventions used, all three approaches use a large number of interventions, with UC using slightly fewer interventions than the other approaches. Table III shows that when run with the maximum permissible intervention spread of 50, all three approaches produce comparable results.

B. 50 Point Intervention Penalty

A penalty cost of 50 points per intervention was then introduced to the system and a second set of experiments undertaken. The number of GA generations remained at 50. Figure 4 shows how all three approaches struggle to find good solutions when the problem complexity increases beyond an initial intervention spread of around 25. While initially maintaining good performance, UC deviates away from the other two approaches. This steady decline in fitness score can be attributed to the increase in interventions used by UC, as shown in Figure 5. TInSSel and CalEB settle around 8 interventions regardless of the number of initial interventions when the intervention spread is greater than 20 whereas UC requires 10 or more interventions for a worse score. Table IV shows that when M is 50, UC performs statistically worse than both TInSSel and CalEB.

C. 50 Point Intervention Penalty, 100 Generations

The next set of experiments was to examine the behaviour of the three approaches when left for a greater number of generations. A 50 point per intervention penalty still remained, but now each run lasts for 100 generations. Figure 6 shows a similar trend to the 50 generation experiments, but as one would expect, the scores are of a higher quality than the 50 generation runs. Both CalEB and TInSSel used on average 6 interventions compared to UC using more than 10 interventions when *M* is greater than 35. Table V shows that with the extra 50 generations, the mean score of UC has improved by approximately 100, whereas both TInSSel and CalEB have improved by more than 150.

D. 50 Point Intervention Penalty, 200 Generations

In order to ascertain the effect of the three approaches when provided with a significant number of generations with which to search, the number of generations was doubled to 200. The penalty value was kept constant at 50. Figure 8 shows that the fitness scores for CalEB and TInSSel plateau close to 2000 when the number of interventions in the initial spread *M*, is greater than 6. The best fitness score for UC continues to climb when the number of interventions in the initial spread is greater than 15. This is highlighted further in Figure 9, which shows that although TInSSel and CalEB settle on average of 4.5 interventions for solutions, UC continues to use an increasing number of interventions. Table VI shows the clear difference in quality of solution between UC and the other approaches, and that TInSSel appears to produce significantly more optimal results than CalEB.

E. Inverted Fitness Function

To test the performance of the crossover approaches for a maxization problem, the gene structure was reversed such that having a gene set to 1 indicated a non-intervention and 0 an intervention for the given day. The fitness scores for this inverted problem are shown in Figure 10 and the intervention usage in Figure 11. The penalty per intervention remains the same as above at 50 and each experiment was run for 200 generation. This is therefore the maximization equivalent of experiment in discussed in Section V D. As expected, a clear inverse correlation was found between these results and the minimization equivalents shown in Figures 8 and 9. Table VII shows that TInSSel and CalEB produce comparable results that again outperform UC.

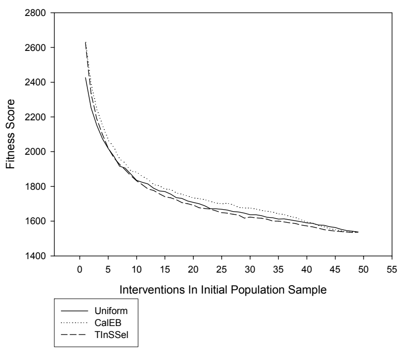

Fig. 2. Fitness score for 50 generations with 0 penalty per intervention. It is interesting to note that all approaches follow a similar trend.

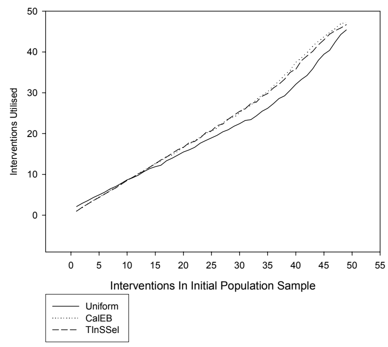

Fig. 3. Intervention usage for 50 generations with 0 penalty. Although a similar trend is shown between approaches, UC uses fewer interventions than both CalEB and TInSSel.

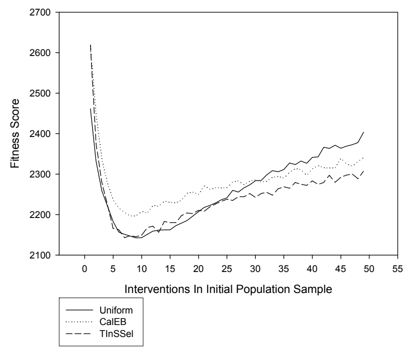

Fig. 4. Fitness score for 50 generations with a 50 penalty per intervention. Although similar trends are shown across all 3 approaches, as *M* increases, the UC fitness score increases faster than the other approaches.

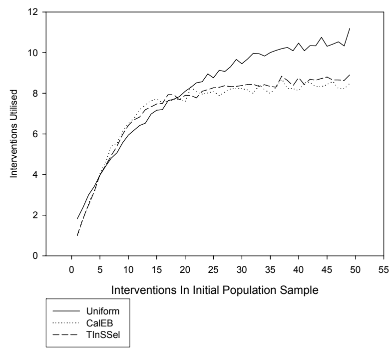

Fig. 5. Intervention usage for 50 generations with a 50 penalty per intervention. As *M* increases past 20, UC continues to use more interventions when compared to both CalEB and TInSSel.

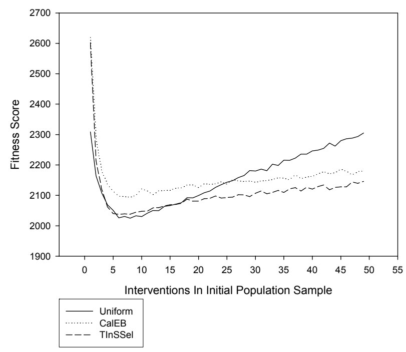

Fig. 6. Fitness score for 100 generations with a penalty of 50 per intervention. All approaches have twice as many generations to find the

best solution compared to previous experiments, the UC scores continue to climb when *M* is greater than 10.

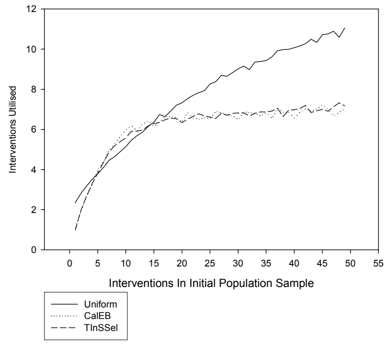

Fig. 7. Intervention usage for 100 generations with a penalty of 50 per intervention. As with the trend demonstrated in Fig. 5, UC continues to use more interventions when *M* is greater than 25.

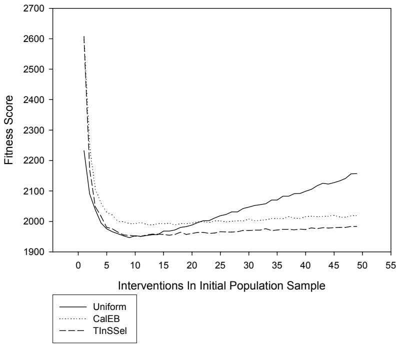

Fig. 8. Fitness score for 200 generations with a penalty of 50 per intervention. Both CalEB and TInSSel settle on constant solutions when *M* is greater than 7. Even with 200 generations to find a solution, once *M* exceeds 15, the quality of solutions found by UC deteriorates.

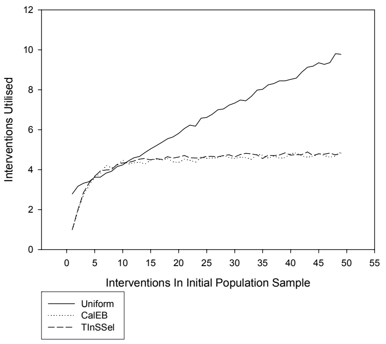

Fig. 9. Intervention usage for 200 generations with a penalty of 50 per intervention. CalEB and TInSSel settle on solutions with approximately 4 interventions and maintain this regardless of the value of *M*. This is not the case for UC.

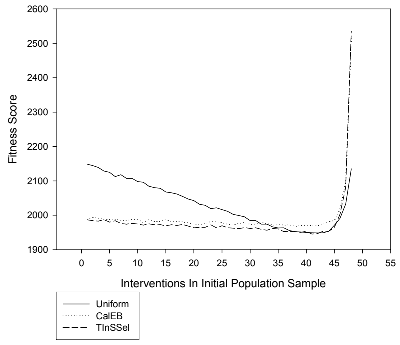

Fig. 10. Fitness score for the inverted fitness function over 200 generations with a penalty of 50 per gene. Although a slight variation in scores exist between this and the normal fitness function (Fig. 8), there is a clear reverse image of trends.

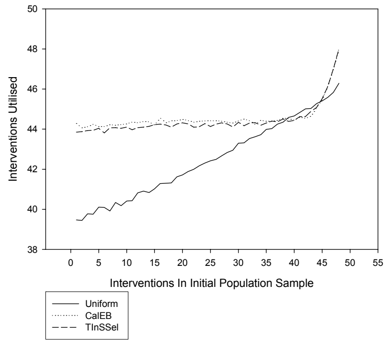

Fig. 11. Intervention usage for the inverted fitness function over 200 generations with a penalty of 50 per gene. As with Fig. 8 and 10, there is a clear correlation between the intervention usage here and the non-inverted case (Fig. 9).

TABLE III.
50 GENERATION, 0 PENALTY

	Uniform	CalEB	TInSSel
Mean	1537.80	1537.23	1536.37
Std. Dev.	16.16	5.67	8.65
Std. Err.	0.72	0.25	0.39
95% Conf.	1.42	0.50	0.76
99% Conf.	1.87	0.66	1.00

TABLE IV.
50 GENERATION, 50 PENALTY

	Uniform	CalEB	TInSSel
Mean	2403.71	2341.50	2307.18
Std. Dev.	144.85	152.84	152.96
Std. Err.	6.48	6.84	6.84
95% Conf.	12.73	13.43	13.44
99% Conf.	16.75	17.68	17.69

TABLE V.
100 GENERATION, 50 PENALTY

	Uniform	CalEB	TInSSel
Mean	2305.29	2180.13	2145.93
Std. Dev.	133.15	124.92	120.15
Std. Err.	5.95	5.59	5.37
95% Conf.	11.70	10.98	10.56
99% Conf.	15.40	14.45	13.89

TABLE VI.
200 GENERATION, 50 PENALTY

	Uniform	CalEB	TInSSel
Mean	2156.92	2018.48	1983.01
Std. Dev.	117.28	70.35	72.89
Std. Err.	5.24	3.15	3.26
95% Conf.	10.30	6.18	6.40
99% Conf.	13.56	8.14	8.43

TABLE VII.
INVERTED FITNESS FUNCTION, 200 GENERATION, 50 PENALTY

	Uniform	CalEB	TInSSel
Mean	2148.77	1987.33	1986.85
Std. Dev.	108.58	61.56	68.88
Std. Err.	4.86	2.75	3.08
95% Conf.	9.54	5.41	6.05
99% Conf.	12.56	7.12	7.97

VI. CONCLUSIONS

In Section V, we reported our experimental results charting the behaviour of both the CalEB and TInSSel algorithms when compared to UC. It can be seen that when no intervention penalty is applied, the approaches perform equally well. Once an intervention penalty is introduced, both CalEB and TInSSel perform better than UC when *M* increases. This result holds true even when the experiment is allowed to run for an increasing number of generations, or indeed when the fitness function is inverted to form a maximize function.

From the experiments carried out in Section V, we conclude that when there is no intervention penalty applied, the choice of crossover technique is arbitrary. When an intervention penalty is introduced to the fitness function, the ability of both CalEB and TInSSel to utilize intervention information makes these approaches more suitable for problems of this type.

Given that both CalEB and TInSSel share the same target intervention calculation process and only differ in their selection of interventions, it would seem that TInSSel is a more efficient choice as it produces better results while being computationally more efficient. We propose that the performance gain achieved by CalEB and TInSSel over UC can be attributed to the utlization of parent intervention data. Our results also confirm that if more interventions produce better scores, larger offspring will be produced.

Given CalEB and TInSSel's ability to derive good solutions when presented with a broad initial spread of interventions, the GA decision maker need not be unduly concerned with setting up the initial population spread. Regardless of the initial spread, CalEB and TInSSel will on average return a good solution. In comparison, UC appears to struggle when presented with an initial population containing multiple interventions.

Current research is focused on continued evaluation of CalEB and TInSSel's effectiveness. In addition, we are investigating their applicability to other domains, and are currently in the process of applying these methods to the scheduling of cancer chemotherapy treatments [9].

REFERENCES

[1] A. Petrovski, A. Brownlee, J. McCall, "Statistical optimisation and tuning of GA factors," in Proc. IEEE CEC, Edinburgh, IEEE Press, pp 758-764, Sept 2005.

[2] A. Petrovski, J. McCall, E. Forrest, "An Application of Genetic Algorithms to Optimisation of Cancer Chemotherapy," Int. J. of Mathematical Education in Science and Technology, 29, 3, pp 377-388, 1998.

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

[4] S. Tse, Y. Liang, K. Leung, K. Lee, S. Mok, "Multi-Drug Cancer Chemotherapy Scheduling by a New Memetic Optimization Algorithm," in Proc. IEEE CEC, Edinburgh, IEEE Press, pp 699–706, Sept 2005.

[5] C. K. Mohan, "Selective Crossover: Towards Fitter Offspring. Selected Areas in Cryptography," in Proc. Symposium on Applied Computing (SAC'98), Atlanta (GA), pp 374 – 378, 1998.

[6] L. Eshelman, L, R. A. Caruana, J. D. Schaffer, "Biases in the Crossover Landscape," in D. Schaffer (ed.) Proc. of 3rd Int. Conf. on Genetic Algorithms, pp 61 – 69, 1989.

[7] R. C. Linton, "Adapting Binary Fitness Functions in Genetic Algorithms," in Y. Seong-Moo, L.H. Etzkorn (eds.) Proc. ACMSE Alabama. ACM Press, pp 391–395, 2004.

[8] K. Deb, "Multi-Objective Optimisation Using Evolutionary Algorithms," J. Wiley & Sons, New York, 2001.

[9] A. Petrovski., J. McCall, "Multi-objective Optimisation 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.