Department of Computing Science and Mathematics
University of Stirling

Directed Intervention Crossover Approaches in Genetic Algorithms with Application to Optimal Control Problems

Paul M. Godley

Technical Report CSM-181

ISSN 1460-9673

May 2009

Department of Computing Science and Mathematics
University of Stirling

Directed Intervention Crossover Approaches in Genetic Algorithms with Application to Optimal Control Problems

Paul M. Godley

Department of Computing Science and Mathematics
University of Stirling
Stirling FK9 4LA, Scotland

Telephone +44 1786 467 421, Facsimile +44 1786 464 551
Email pgo@cs.stir.ac.uk

Technical Report CSM-181

ISSN 1460-9673

May 2009

Abstract

Genetic Algorithms (GAs) are a search heuristic technique modelled on the processes of evolution. They have been used to solve optimisation problems in a wide variety of fields. When applied to the optimisation of intervention schedules for optimal control problems, such as cancer chemotherapy treatment scheduling, GAs have been shown to require more fitness function evaluations than other search heuristics to find fit solutions. This thesis presents extensions to the GA crossover process, termed directed intervention crossover techniques, that greatly reduce the number of fitness function evaluations required to find fit solutions, thus increasing the effectiveness of GAs for problems of this type.

The directed intervention crossover techniques use intervention scheduling information from parent solutions to direct the offspring produced in the GA crossover process towards more promising areas of a search space. By counting the number of interventions present in parents and adjusting the number of interventions for offspring schedules around it, this allows for highly fit solutions to be found in less fitness function evaluations.

The validity of these novel approaches is illustrated through comparison with conventional GA crossover approaches for optimisation of intervention schedules of bio-control application in mushroom farming and cancer chemotherapy treatment. These involve optimally scheduling the application of a bio-control agent to combat pests in mushroom farming and optimising the timing and dosage strength of cancer chemotherapy treatments to maximise their effectiveness.

This work demonstrates that significant advantages are gained in terms of both fitness function evaluations required and fitness scores found using the proposed approaches when compared with traditional GA crossover approaches for the production of optimal control schedules.

Acknowledgements

Many people have helped to make this project possible and I would like to take this opportunity to thank them. First I would like to thank God for the health, wisdom and favour that allowed me to undertake this task.

My beautiful wife Sharon has given me the support, love and encouragement to keep going, even when I felt like stopping. My Mum and Dad have always encouraged me and have always been there for me and I love and respect them dearly. My other Mum and Dad across the water have also been a great help and encouragement, and I greatly love and appreciate them. Pastors Scott and Jon have always helped with their encouragement and wisdom.

My supervisory team of David Cairns and Julie Cowie has made this project a pleasure. They have offered insightful feedback, comments and criticism which has made this project what it is. I would also like to thank John McCall for his time and input through the duration of this project. He has provided me with much food for thought and his encouragement and optimism has helped improve the project significantly.

Many thanks to Carron Shankland, who has given me feedback and advice which has greatly helped. I am also indebted to Catherine Howie who has single handedly made statistics much more appealing and understandable. To everyone who offered comments, criticism and feedback at the conferences, seminars and lectures I have given, thank you.

My friends have always been there to show me that there is more to life than work. Thank you Fraser and Jon, let's hope that the next 16 years of friendship are as fun as the last!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Genetic algorithms (GAs) are search procedures inspired by the mechanisms of natural adaptation. They were defined by Holland [1] in 1975, and have been extensively studied and used in real-world applications.

GAs have been used to find good solutions from large search spaces across a wide range of application areas. One such area where they have proven successful is for deriving solutions to optimal control problems. Optimal control theory is concerned with finding the best control schedule for a dynamic system which optimise a performance criterion in some way.

Optimal control problems arise in a wide range of fields of engineering and sciences [2], however, the task of designing and implementing algorithms for solving optimal control problems can be a difficult one [3]. GAs have been readily applied to this type of problem as they require little knowledge of the problem domain itself to search for solutions.

GAs work on a set of solutions, termed a population. Each member of the population is referred to as a chromosome and encodes a solution to the problem in hand. For many optimal control problems each chromosome encodes a set of interventions across a time period. Each chromosome has an associated fitness score describing how well the encoded solution solves the problem in hand. Following a survival of the fittest metaphor, fit solutions are assigned a higher probability of producing offspring than less fit solutions. Offspring are created through the evolutionary inspired processes of crossover and mutation, where crossover splices genetic material from parents into offspring, and mutation introduces diversity into the solutions. The crossover process is commonly regarded as more innovative than the mutation operator [4]. Conventional GA crossover approaches exchange the scheduling information contained in parent solutions at random to produce offspring.

Although GAs have been applied successfully to a range of optimal control problems, they have been found to be slower when compared to other search heuristics [5; 6; 7]. This leads to the question of how GAs could be made more effective for this type of problem.

The hypothesis put forward in this work is that the performance of GAs can be improved with respect to intervention based optimal control problems through using the number of interventions present in parent schedules to direct the offspring to promising areas of the search space. As crossover is key to the GA process and is considered more innovative than mutation an extension to enhance crossover could therefore provide a more effective search for problems of this type and improve the GA performance. In order to prove this position, novel techniques have been created, termed directed intervention crossover approaches. Through comparison with traditional GA crossover approaches for optimisation of bio-control and cancer chemotherapy schedules, the validity of the novel crossover techniques will be demonstrated.

The remainder of this thesis is structured as follows:

Chapter 2 contains a review of genetic algorithms, including details of their operation and theory. The main components of GAs are discussed, such as the encoding of solutions, how fitness is assigned to solutions, how parents are selected to produce offspring and how offspring are subsequently constructed. Other aspects of GA operation are considered, such as how the population is initialised and when the GA operation terminates.

Optimal control problems are the application domain for the techniques discussed in this work. Chapter 3 provides an introduction to these types of problems and describes some heuristic techniques for solving them. The limitations of using GAs for optimal control problems are discussed and an extension for their improvement is outlined.

Chapter 4 discusses techniques developed to test the validity of these extensions. This chapter introduces the directed intervention crossover techniques of Targeted Intervention with Stochastic Selection (TInSSel), Calculated Expanding Bin (CalEB) and Directed Uniform Crossover (DUC). The rationale and algorithmic processes for each of these techniques are described.

In order to assess the novel crossover approaches detailed in Chapter 4, suitable test problems are required. Chapter 5 details the test functions used for analysis in this work, namely bio-control scheduling for mushroom farming and cancer chemotherapy drug scheduling. This chapter describes the background for the problems as well as their mathematical formulation.

Chapter 6 details the experimental method that will be used for the analysis of the crossover approaches. This chapter details the rationale behind the selection of GA parameters used for experimentation as well as the statistical tests and methods used to ensure accurate analysis of data.

Chapter 7 reviews the strengths of traditional crossover approaches for optimisation of the problems detailed in Chapter 5. This chapter reviews single point crossover (SPC), two point crossover (2PC) and uniform crossover (UC) for both bio-control and cancer chemotherapy treatment scheduling.

Chapter 8 details the results of applying the novel crossover techniques discussed in Chapter 4, to the problems described in Chapter 5. These techniques are compared to the best performing traditional crossover approach found in Chapter 7 in order to determine the benefits of the novel techniques.

From analysis of the findings in Chapter 8, an extension to the novel crossover techniques is proposed. Chapter 9 details an implementation of this extension called Fitness Directed Crossover (FDC). This chapter compares the FDC technique with the best performing novel and traditional crossover techniques for optimisation of the test problems.

The final chapter summarises the contributions made by the novel crossover techniques introduced in this work and reflects on the associated strengths and limitations. Further extensions to the directed intervention crossover approaches are described as well as potential areas for further analysis of the work.

Finally, Appendices A through to J contain the detailed experimental analysis from Chapters 7, 8 and 9. This shows the empirical results of applying the techniques to the test problems over a range of experimental parameters, as well as the associated tables of statistical analysis.

Chapter 2

Genetic algorithms as a method of evolutionary search and optimisation

Genetic algorithms (GAs) are search procedures inspired by the mechanisms of natural adaptation. They simulate the evolutionary process, and have a simple operation. They were defined by Holland [1] in 1975, and have been extensively studied and used in real-world applications. GAs are considered attractive in many fields due to their robustness, simplicity and the variety of solutions they find [8].

GAs start with a population of random individuals, where each individual encodes a solution to the problem in hand. Through procedures that model survival of the fittest concepts, solutions with higher utility will have preference in producing offspring. Over time this tends to lead to better individuals typically being found.

This chapter is a review of these search procedures and starts with an overview of their basic operation and methodology. It discusses the concepts relating to how a population of solutions is encoded, initialised, scored for fitness and selected for producing offspring. Other areas reviewed are how offspring are produced and reintroduced into the population and how and when stopping criteria are determined. Following this, the rationale and underlying theory of the GA will be examined. Current GA practice is then described, outlining the common methods for determining parameter settings, including both population size and crossover and mutation rates.

2.1 Genetic algorithm operation

The GA process is shown in Figure 2.1.

In general, GAs are initialised with a random population of solutions, where each solution represents an encoding for the problem in hand. Following this, solutions are typically selected to produce offspring based on their fitness, with greater opportunity offered to fitter solutions, thus facilitating good solutions to pass their material on to the subsequent generations. Children are then bred from the parents using the evolutionary inspired methods of crossover and mutation. The population is subsequently updated to include these children. If the stopping criteria has not been reached, the selection, breeding and replacement process repeats, otherwise the GA process is complete. Each of these components will be described in more detail in the following sections.

In order to apply GAs to a problem, an internal representation of the search space and an evaluation function are required, and both of these components are critical to the successful application of the GAs to the problem of interest [9]. The following sections will explore these concepts in greater depth.

2.1.1 Encoding

When applying a GA to a problem domain, the first step is to determine an appropriate encoding. This is seen as one of the primary decisions concerning a GA [10].

The encoding allows each possible solution to encapsulate the relevant variables for the problem. The set of all possible solutions is termed the search space for the problem. Many different encodings have been used, with a string representation of a binary structure being the traditional approach, and other encodings including real, permutation and tree encodings. In a binary encoding, with a bit string of length n, each individual represents one point in a space of size 2^n.

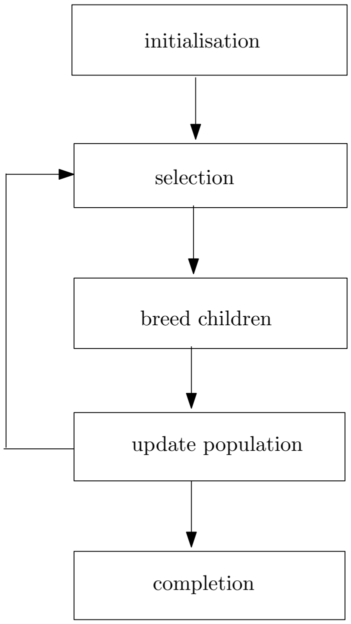


Figure 2.1: The GA process

As GAs are inspired by the theory of evolution, various terms are used from this domain. Each solution is referred to as a chromosome, which is made up of decision variables, termed genes. Each gene has a value and this is termed the allele, such as a 0 or a 1 in a binary representation. The position of the gene within the chromosome is known as the locus. For example, given the following binary encoded chromosome:

1	0	0	0

The gene at locus one has an allele of 1, with the remaining three genes at loci two, three and four respectively, having alleles of 0.

Deciding an appropriate encoding for the GA is a key decision and is considered a central factor in the success of a genetic algorithm [11]. Once an encoding for solutions is found, an evaluation function, more commonly referred to as a fitness function, is required to provide a score of a solution's utility.

2.1.2 Fitness function

In GA terminology, fitness is defined as the means of profit, utility or goodness that is to be optimised [12]. The fitness function is the only information the GA requires of the problem [13]. By evaluating a chromosome with regard to the fitness function, a score will be returned pertaining to how well the chromosome solves the problem in hand. Continuing the biological metaphor, the fitness score reflects how well the individual is adapted to the environment [14].

A simple fitness function known as the Onemax problem, is shown in Equation 2.1. This problem takes a binary string x of length l and seeks to maximise the number of 1's present. The greater the number of 1's in a solution, the higher the score the fitness function returns.

$$\sum_{t=1}^{l} x_t, x_t \in \{0,1\}^l \tag{2.1}$$

If the chromosome described in Section 2.1.1 was passed through this fitness function, it would return a score of 1. The optimal solution to this fitness function would be for each gene of a chromosome to contain a 1.

Once an appropriate fitness function has been decided upon, the initialisation method and stopping criteria have to be defined. Each of these concepts will now be discussed.

2.1.3 Initialisation method

Genetic Algorithms work on a population of chromosomes. This has a significant advantage in that a high number of portions of the solution space are explored with great efficiency, concentrating samples in the most promising regions [15].

Traditional GA methodology uses a random population initialisation, where each chromosome is created by picking random values for each gene. This is of benefit since no knowledge is required of the structure of solutions.

As with most aspects of GA methodology, alternative initialisation methods exist. In a domain where the GA user has a knowledge of what represents good structures, the population can be initialised including some 'seeded' good values. This benefits the GA as salient characteristics from these solutions can spread through the population allowing for highly fit solutions to be ascertained in less time than with purely random initialisation. Other techniques include using the best solution found from previous runs of the GA as a partial solution from which to generate the population for the next run [16], and for some problems, case-based initialisation of populations [17].

2.1.4 Stopping criteria

There are various options open to the GA practitioner for deciding when the search process should stop. Common approaches involve the GA stopping after a fixed number of fitness function evaluations, when the population has converged or when a solution of a particular fitness has been found. If the optimal fitness score is known, the search process could be stopped on discovery of a solution with this fitness, however as many optimisation problems have complicated objective functions or contain difficult constraint structures, in practice, any near-optimal solution would be desirable if obtained with a reasonable effort [18].

Theoretical guidelines for the number of fitness function evaluations required to stop the search process have been reviewed and include bounds for an optimal solution, for certain classes of problems [19] and bounds for convergence [20].

The following sections review how solutions are selected for breeding, how the breeding process produces children through the methods of crossover and mutation, and subsequently, how these child solutions are introduced back into the population.

2.1.5 Selection

The selection mechanism chooses the chromosomes in the population for reproduction and its main function is to emphasise better solutions in the population [21]. Generally, selection is based on the fitness of the chromosome, thus the fitter the solution the more times it is likely to be chosen to reproduce. Selection is seen to provide the driving force in evolutionary algorithms, and selection pressure is seen as a critical parameter [22]. If the selection pressure is too high, there is a high probability of the population prematurely converging to a sub-optimal solution. When the selection pressure is too low, there is not enough drive in the system, thus optimal solutions may still be found, but the process may be much slower than necessary. To combat this, techniques such as Boltzmann selection are used which varies the selection pressure in a similar manner to the temperature control of simulated annealing, according to a preset schedule. This provides a low pressure at the start of the run which facilitates reproduction from less fit individuals and thus increases diversity. As the run continues, the selection pressure grows thus giving more reproductive opportunity to higher fitness solutions.

In Holland's original work [1], fitness-proportionate selection was used. Fitness-proportionate selection uses the fitness of a solution relative to the average fitness of the population to determine the likelihood of selection to produce offspring. This is commonly implemented as roulette wheel sampling whereby each solution gets space on the 'roulette wheel' proportional to the fitness of the solution compared to the population. The higher the fitness of the solution compared to the average fitness of the population, the greater the space on the wheel allocated to that solution. When the wheel is spun (i.e. when selection occurs), there exists a higher probability of the selection marker landing on a fitter solution.

Although widely used, fitness-proportionate selection of this form has various drawbacks related to scaling. If the population is created randomly, there will probably be a wide spread of fitness values. As this form of selection greatly increase the likelihood of selection relative to the population average, numerous copies of these fitter solutions will be chosen for reproduction. This can introduce the problem of super individuals. This is caused through fitter than average solutions dominating the selection process. This may lead the GA to prematurely converge on sub-optimal solutions due to a lack of genetic diversity [23]. Techniques such as winnowing and rank based selection [24] have been designed to allow fitness-proportionate selection to overcome these problems.

Winnowing subtracts the worst fitness score from the previous w generations from each of the current fitness scores to produce a more constant selection pressure.

Rank based selection was introduced by Baker [24]. In this approach, the members of a population are ranked according to their fitness and the number of expected offspring for each member is calculated based on this rank, rather than on the direct fitness value of the solution. There are varying forms of rank based selection commonly in use, such as linear and exponential.

Both rank based selection and fitness-proportionate selection can be quite expensive processes. Rank based has to sort the entire population and fitness-proportionate has to undertake two passes of the population, one to calculate the mean population fitness and another to determine the relative fitness value of each solution. A popular alternative to these approaches is tournament selection. Tournament selection has been shown to provide better or equivalent convergence and computational properties when compared to alternative approaches [25].

Tournament selection has been used in a number of works and follows a simple premise. First, n individuals are chosen at random to compete, then the fitter of these individuals is chosen for breeding. Binary selection, where $n = 2$, is commonly used [26]. Larger tourneys can also be used, and the larger the tournament size, the higher the selection pressure. As described in [27], a key strength of tournament selection is the ease of parallel implementation.

Comparative analysis of selection schemes have been undertaken, such as [28]. When compared to truncation or rank based alternatives, tournament selection has higher selection variance and smaller loss of diversity for the same selection intensity [29]. However, as with all approaches, tournament selection has drawbacks. As each tournament is carried out separately, tournament selection can suffer from sampling errors [22].

Truncation selection was first used in genetic algorithms in [30] and incorporates some threshold T, where only a fraction of T best individuals can be selected. Each of the individuals in this threshold have similar probabilities of selection.

Once individuals have been chosen to produce offspring, the next step is to create offspring. This is achieved through the processes of crossover and mutation.

2.1.6 Crossover

Crossover and mutation are termed the genetic operators and provide the main search functionality of a GA. Mutation creates random diversity in the population while crossover promotes emergent behaviour from existing genetic components [31]. Crossover is commonly regarded as more innovative than the mutation operator [4] and the search power of a crossover operator is commonly considered as a measure of how flexible the operator is in creating an arbitrary point in the search space [32].

Once parent solutions have been selected for breeding, the next step is to apply crossover and mutation. As with its biological equivalent, the GA crossover process creates offspring representations from parts of the parent representations. This allows salient characteristics from the parent solutions to be combined to produce children that are potentially fitter than either parent alone. Indeed, without the crossover process, GAs would amount to little more than random search [33]. Crossover is applied subject to a predetermined probability p_c. Typically a value n is generated between 0 and 1 and if $n \leq p_c$ crossover will occur. If n is not less than p_c, one of the parent chromosomes are copied at random, into the next generation. Commonly used crossover techniques include single point crossover, two point crossover and uniform crossover [34]. Each of these will be reviewed in turn.

Single point crossover

Single point crossover (SPC) is the simplest form of crossover. A crossover point in the chromosome is randomly determined and the parts of two parents after this point are exchanged, thus producing new offspring. Consider the following two parents selected to produce offspring:

1	1	1	1

0	0	0	0

If crossover was determined to occur after the second gene, this would produce the following two children:

1	1	0	0

0	0	1	1

Two point crossover

Two point crossover (2PC) is an extension to SPC which selects two crossover points at random and swaps contents of the parent chromosomes between those points to form offspring. Using the two parents described before, if crossover points 2 and 4 were chosen, this would make the first child comprise of gene 1 from the first parent, genes 2 and 3 from the second parent and gene 4 from the first parent. Conversely, the second child will have gene 1 from the second parent, genes 2 and 3 from the first parent and gene 4 from the second parent. This produces the following two children:

1	0	0	1

0	1	1	0

Uniform crossover

Uniform crossover was studied by Ackley [35] and subsequently popularized by Syswerda [36] and is a popular alternative to the one and two point crossover approaches. Uniform crossover (UC) can perform more genetic exchange per crossover process than both single and two point crossover and works on a probability of taking a particular gene value from one of the parents. Typically, this involves generating a random number, n, between 0 and 1 for each child gene position, and if $n \leq 0.5$ then the gene value becomes the same as the first parent at that locus. If $n > 0.5$ then the offsprings value for the gene in question is copied from parent two. Usually the swapping probability s_p is taken to be 0.5 [37], however, parameterized versions of UC, where different values of s_p are applied, have been used and found to be useful in certain cases [38].

Using the two parents selected for breeding detailed previously, if the random values 0.1, 0.6, 0.2 and 0.9 were generated for n for the first child, this would translate to taking gene one from parent one, gene two from parent two, gene three from parent one and gene four from parent two. This would create the following child:

1	0	1	0

If for the second child the values 0.8, 0.8, 0.9, 0.1 were generated, this would create the following child:

0	0	0	1

Using UC, if both parents contain the same gene value at a particular locus, this value will always be copied to the child.

UC is probably the most commonly used crossover operators due to its ability to re-combine non-common genes and find and protect common genes [39]. However, some research indicates that UC outperforms two point crossover for smaller populations, while for larger populations, two point crossover is the superior approach [40]. There is no consensus on the 'correct' crossover approach to use on a particular type of problem.

The crossover stage is key to the GA process. Once this crossover process is complete, the next step is to apply mutation to the offspring.

2.1.7 Mutation

After crossover has produced offspring, mutation is then applied. Mutation diversifies the search directions and avoids convergence to a local optimum [41]. Mutation introduces new gene values into the population and through this, allows exploration of other areas of the search space. For example, if a population consisted of binary solutions which all had a '1' as the last gene of the chromosome, no amount of crossover could ever change this as regardless of crossover strategy or crossing point, each solution will continue to have a '1' in the last gene position. Mutation, could change this value (or indeed any other gene in the chromosome) and thus provide evaluation of a different area of the search space. The simplest form of mutation is 'bit flipping'. Here, for a binary string, if the current gene value is a '0', mutation produces a '1' and conversely, for a gene value of '1', mutation produces the value '0'.

Mutation has a probability associated with it, p_m, which is generally quite low, usually in the region of 0.01 to 0.005. For each gene, a random value is generated and if this is $\leq p_m$, then the value of that gene is subject to change. For example, given the following binary chromosome:

1	0	1	0

If the random values generated to test for mutation were 0.800, 0.900, 0.200 and 0.001, where p_m is 0.005, this would mean leaving genes one, two and three as they are but flipping the value of gene four, thus producing:

1	0	1	1

Once the processes of crossover and mutation are complete, the offspring are evaluated using the fitness function as described in Section 2.1.2. They are then ready to become part of the next generation of solutions. There are various approaches to replacing the old solutions with the new and these will now be discussed.

2.1.8 Replacement

Alternatives exist regarding when to add the offspring to the current population and which members of the current population should be removed to make space for these new members.

A common technique for incorporating offspring into a new population is the generational method. This means that for a population of size N, parents will be selected from the current population to produce offspring until some predetermined fraction of N offspring have been produced. This fraction of new individuals for each generation is called the generation gap [42]. A generation gap of 0 would mean that none of the previous population is replaced, with a value of 1 specifying that the entire population is replaced at each generation. Once produced, the current population and the offspring are merged to form the next generation, of size N. This process continues till the stopping criteria is met.

To ensure that good solutions are not lost from generation to generation, elitism is commonly employed. With elitism, the best chromosome in the current population is guaranteed to be added to the next generation, i.e. it will not be overwritten by new offspring from one generation to the next. This ensures that the gene values which currently provide the best fitness will not be lost to the subsequent generation.

The generational replacement technique is not without some potential drawbacks [21]. One is that even with an elitist strategy, many good solutions may not reproduce and thus their genes are lost. One solution to this problem is that of the steady state replacement strategy.

Steady state is an alternative replacement approach which places a small number of offspring into the population at each generation. Various studies have compared this and the generational approach [43] and while the generational approach may be more commonly used, techniques such as GENITOR [44] use a steady state replacement strategy as a key part of their algorithmic operation. Although there is no ‘correct’ choice for the replacement strategy, for some non-stationary environments, steady state has been shown to outperform generational replacement. This was attributed to it allowing offspring to be immediately used as part of the mating pool, making a shift towards the optimal solution possible in a relatively early part of the optimization process [45].

The generational approach tends towards a high generation gap value, thus replacing most of the current population, while the steady state replaces only a few individuals and thus has a smaller value. As mentioned previously, techniques such as elitism can be used to protect good solutions with a common representation of this ensuring that the best solution in the population is copied forward to the subsequent generation.

The Chapter thus far has reviewed the representation and operation of a genetic algorithm. The following section reviews the theory which underlies this process.

2.2 Genetic algorithm theory

Conventional understanding of GA operation centers around the theories developed by Holland [1]. This is based on the notion that useful parts of different solutions can be combined to form good solutions. Holland introduces a schema as a similarity template to represent a set of solutions from the search space. Holland’s work focussed on bit strings, and each schema is represented by the alphabet $\{0, 1, *\}$, where * represents a wildcard at a particular position.

Using this ternary alphabet, 1*01 represents a schema which requires the first bit to be a 1, the second bit can be either a 0 or a 1, the third bit as a 0 and the fourth bit a 1. If a bit string obeys a schema s pattern, it is said to be an instance of s. Both 1101 and 1001 are instances of the previous schema. Each 1 or 0 in a schema are known as defined bits and the order of the schema is the number of defined bits. The distance between the two outermost defined bits is known as the defining length. The previous schema therefore has order 3 and a defining length of 4. The operation of GAs can thus be regarded as “a search for schemas of high average fitness, carried out by sampling individuals in a population and biasing future samples towards schemas that are estimated to have above-average fitness” [46].

Holland used these concepts to perform analysis on the effects of the GA operators on a schema H. This showed the dynamics of the increase and decrease of schema instances and was formalized as the Schema Theorem, as shown in Equation 2.2.

$$m(H, t+1)) \geq m(H, t)\frac{f(H)}{\bar{f}}\left[1 - p_c\frac{\delta(H)}{l-1} - o(H)p_m\right] \tag{2.2}$$

where:
$m(H, t)$ is the number of instances of schema H at time t,
$f(H)$ is the average fitness of strings representing schema H at time t,
$\bar{f}$ is the average fitness of the entire population,
p_c is the probability of crossover,
$\delta(H)$ is the defining length of schema H,
l is the length of binary string,
$o(H)$ is the order of schema H,
p_m is the probability of mutation.

The schema theorem assumes fitness-proportionate selection, one point crossover for a population of binary strings, using non-overlapping generational replacement but has been extended to consider various other selection schemes and operators [47; 48].

This theorem says that while selection emphasizes fit schemata (a template that identifies a subset of strings with similarities at certain string positions), the variation operators destroy some and "advocates the idea that GAs achieve success through the juxtaposition of short, low-order, high performance schemas"[49]. These short, low-order, high performance schemas are known as building blocks and are sampled, recombined and resampled to form chromosomes of potentially higher fitness, also known as the building block hypothesis [12].

Holland's analysis of the schema theorem also showed that when a GA is evaluating an individual, it is actually evaluating partial solutions as well. Consider the following chromosome:

1	0	1	1

when this is evaluated by the GA, it is also evaluating sub solutions of 1011, such as 1***, 1**1, *0** and many more. This evaluation of partial solutions is called implicit parallelism and is considered one of the reasons why GAs are so powerful [14]. This means that when a population of N solutions are evaluated, a number of partial solutions are also evaluated, at a cost of only N fitness function evaluations.

Mitchell believes that with regards to GA theory, there is more open questions than answered ones [11]. This has not stopped GAs being widely applied and the following section outlines some of the current practice in the use of these techniques.

2.3 Current genetic algorithm practice

The previous sections reviewed both the components that make up a GA and their underlying theory. This section builds on this by considering how parameter values for these techniques are derived.

2.3.1 Parameter settings

As discussed previously, when applying a GA to a problem, there are various choices to be made including the selection type and crossover type. These may in turn have associated parameters, such as tournament size for tournament selection or probabilities for crossover and mutation. Other parameter settings required for a GA run include determining the population size.

In his 1975 thesis, De Jong [42] carried out a systematic study on the effects of parameters on the optimisation of a set of test functions. There were five test functions, namely the sphere model, generalised Rosenbrock's function, step function, quartic function with noise and Shekel's foxhole. Experiments on these functions found that population sizes of 50 - 100, crossover rates of between 0.6 and 0.9 and mutation rates of 0.01 and 0.001 were effective.

Although the parameter settings were effective for the five test functions, these settings have been used as a template to a wide variety of solutions but theoretical results have proven this to be a mistake [14]. It is therefore essential for the GA practitioner to test problems over a range of parameter settings, rather than opting for standard parameters which proved effective for a different type of problem.

There is strong empirical evidence showing that population size is one of the most important parameters, playing a significant role in the performance of genetic algorithms [50]. In principle, small populations are at risk of under representing the search space and thus converging to a poor solution and large populations allow exploration of fewer generations per unit of computational effort and thus may preclude convergence [51].

Parameters such as population size have added complexity in that they are not independent, and have interacting roles with other aspects of the GA. The interacting role between population size and crossover are of great interest to the genetic algorithm community [33]. Despite the operational simplicity of GAs, they are complex systems to analyse [52].

As deciding upon a relevant crossover or mutation level is a factor in the successful application of a GA, various techniques have been used to try and automate these decisions. Grefenstette [53] used a GA to self-adapt the optimal parameter settings and this has led to a variety of approaches to take the onus of parameter setting away from the GA user. As crossover is traditionally regarded as the main search mechanism in genetic algorithms, most efforts to self-adapt the characteristics of this operator stem from this area [54]. Many different techniques exist in this field including the Self Adaptive Genetic Algorithm (SAGA) [55] and more recently the parameterless GA [14; 56]. These aim to give the users more time to focus on other aspects of experimentation and will allow non-expert users to harness the power of GAs [57].

There are a variety of choices in methods and parameters for configuring a GA. Values for parameters can greatly affect the ability of the GA to efficiently find a near-optimum solution [58]. It is therefore of key importance that fundamental components of the GA, such as crossover methods, be robust to changes in these parameters, such as population size and mutation rate or the replacement strategy used.

2.4 Summary

This chapter has introduced genetic algorithms as a method of evolutionary search and optimisation. The components of a GA were described and the theory underpinning their operation has been outlined. Through this chapter, the crossover process has been described as the main search proponent of GAs and some common crossover techniques have been outlined. The importance of parameter settings has also been described. Due to the difficulty in determining parameter values, it is important that GA techniques are robust to variability in these parameters. The next chapter will review a particular type of problem to which GAs can be applied: optimal control problems.

Chapter 3

Optimal control problems

The previous chapter introduced Genetic Algorithms as an evolutionary inspired search technique. GAs have been used to find good solutions from large search spaces across a wide array of application areas. One such area where GAs have proven successful is for deriving solutions to optimal control problems. Although optimal control problems arise in a wide range of fields of engineering and sciences [2], the task of designing and implementing algorithms for solving optimal control problems can be a difficult one [3]. However, as GAs require little knowledge of the problem itself to search for solutions, they can be applied readily to problems of this type.

This chapter discusses optimal control problems within the context of evolutionary computation and is organised as follows. Section 3.1 will discuss optimal control problems and Section 3.2 will review how GAs and indeed other Evolutionary inspired approaches have been applied to solving optimal control problems. Finally, Section 3.3 concludes this chapter, describing a novel extension for the GA crossover process in order to improve GA performance when applied to optimal control problems. The experimentation and analysis of these novel crossover techniques form the main thesis of this work.

3.1 Introduction to Optimal Control

Optimal control theory is concerned with finding the controls for a dynamic system which optimise a performance criterion in some best way. These controls are usually not a single input but may be multiple inputs of different values over time. An example of this may be to find the times to press the accelerator on a car to drive it around a track in the minimum amount of time, or alternatively, to minimise the fuel usage of the car for a lap of the track.

The essential elements of the control problem are defined in [59] as:

1. A mathematical model (system) to be ‘controlled’
2. A desired output of the system
3. A set of admissible inputs or ‘controls’
4. A performance or cost functional which measures the effectiveness of a given ‘control action’

In the above car examples, the mathematical model would represent the car, the desired output would be to complete the lap of the track, and the controls would be timings of when to press the accelerator and for how long to keep it depressed and by how much. The cost functional for the first example would be the time taken to complete a lap, with the fuel usage during a lap forming the cost for the second example.

A vector of variables represents the state of the system and a control set of variables transforms the system to another state. The problem in optimal control is to find the control vector which transfers the system to a state which maximises or minimises some performance criterion, whilst adhering to the set of equality and inequality constraints that may exist. In the car example, constraints may be in place for how far the accelerator can be depressed or for the cumulative fuel usage throughout the lap which must be less than the fuel limit for the car. It should be noted that both the control vector and the state vectors can be discrete or continuous.

Mathematical techniques such as dynamic programming can be used to solve problems of this type, however, this technique has been shown to break down on problems of moderate size and complexity [60]. In the field of chemotherapy scheduling for example it was observed that traditional algorithms for optimal control can become intractable as constraints are added, thus for more detailed analysis, alternative approaches are required [61]. A range of these alternative approaches will now be discussed.

3.2 Heuristic Approaches to Optimal Control

Optimal control is one of the areas which has received the most attention in the field of evolutionary computation techniques such as genetic algorithms [62]. Although genetic algorithms were first developed in the seventies, it was not until 1990 that they were first applied to optimal control problems [3]. This initial work successfully applied GAs to two discrete-time optimal control problems. This work was further extended in [63] and used a modified version of the standard GA, which included a form of arithmetical crossover to tackle three optimal control problems. Arithmetic crossover forms children through linear combinations of their parents gene values, using a parameter a, which is either a constant (uniform arithmetical crossover) or depends on the age of the population (non-uniform arithmetical crossover). In [63] the GA was successfully applied to a linear-quadratic problem, the harvest problem and the push-cart problem. For all three of these problems, the solutions found by the GA were close to those found by a search-based computational package. The conclusion from these initial works was that GAs could be successfully applied to optimal control problems. Since then, evolutionary approaches such as GAs have been successfully applied to a variety of optimal control problems [64].

One such problem where GAs have successfully derived optimal control schedules is that of the optimisation of cancer chemotherapy treatment schedules [61; 65; 49; 66; 67]. This work shows the effectiveness of GAs in finding solutions in a highly complex search space: the composition of an optimal schedule for anti-cancer chemotherapy treatments is a non-linear optimal control problem which is subject to contradictory constraints [68]. These constraints include the maximum instantaneous dose for each application of the drug, the maximum cumulative dose for the drug and the maximum permissible size of the tumour.

Variations of GAs, such as the Co-operative Co-evolutionary Genetic Algorithm (CCGA), have also been used to solve optimal control problems [69]. The CCGA approach was introduced by Potter and De Jong [70] and involves a population with a number of sub-populations or species, where each species represents a part of the problem to be optimised. The fitness of members of the species are evaluated and individuals from each species are combined to form a complete solution to the problem in hand, and the evolutionary process proceeds in the usual GA fashion. In [69], both the standard GA and the CCGA use a binary encoding, standard uniform crossover and bit-flip mutation to find solutions which outperform the dynamic programming technique for the optimal control problems under review.

An alternative GA technique, Adaptive Elitist population based Genetic Algorithm (AEGA) was found efficient at deriving drug scheduling policies for optimal control of cancer chemotherapy [71]. This approach uses an elitist crossover operator which actively tries to reduce the population's redundancy.

GAs fall under the banner of Evolutionary Algorithms (EA). GAs are one of the 3 main streams of EAs, along with Evolutionary Programming (EP) and Evolution Strategies (ES) [72]. Both EP and ES have a number of similarities and were developed at the same time, with EP being developed in America and ES in Germany. Both of these other approaches have also been applied to problems of this type.

Evolutionary programming was first presented by Fogel in the 1960s [73] to evolve finite state machines, and this technique has been successfully applied to optimal control problems [74; 75]. Unlike GAs, EP has no selection for breeding as each member of the population generates one offspring and EP does not use crossover and instead relies purely on mutation to manipulate genetic material. Each subsequent generation includes a percentage of the fittest individuals in the current generation and the EP replacement scheme is always elitist [76].

Evolution Strategies were first developed in [77] and as with EP, use mutation to create and modify genetic material. Originally the population was represented by a single individual being represented as a real valued vector [76], however, more recent ES algorithms, such as those described in [78], are population based.

Although both EP and ES have many common features, such as real value representation of individuals and the use of mutation as the main search operator, they are not the same. Variations in the recombination operators and selection mechanism lead to differences between approaches [72].

Villasana and Ochoa [5] compared the performance of GAs, Evolution Strategies and Simulated Annealing (SA) for deriving bang bang optimal controls for cancer chemotherapy. A bang bang controller is one which switches abruptly between two states. Simulated annealing was first introduced in [79] and as the name suggests, builds an analogy with the way metals cool and anneal in thermodynamics, where the optimisation process is represented as the process of cooling. A solution to the problem is perturbed and will be accepted, even if the solution fitness score is less than that of the current state as long as the temperature is high. As the temperature decreases, only smaller more efficient changes will be accepted, and this allows for a more local search of the space. This work showed the effectiveness of ES over both Simulated Annealing and GAs for problems of this type, with GAs showing slower convergence than the other approaches. This has led to further work by these authors using ES as opposed to GAs for problems of this type [80]. Simulated annealing has also been successfully used by others, such as Agur et al [81] for deriving optimal controls for chemotherapy scheduling.

Although GAs have been used effectively, as stated above they have been shown as slower to converge to good solutions than both ES and SA. This is also the case when GAs were compared to Particle Swarm Optimisation (PSO).

PSO is a search heuristic developed by Kennedy and Eberhart in 1995 and was inspired by the social behaviour displayed by bird flocking or fish schooling [82]. Each solution is encoded as a particle. Particles traverse the search space, based on the best solution that particle has found so far, as well as direction from the best solution found globally by the swarm. In this comparison, GAs were once again outpaced in finding optimal controls for complex models [6].

GAs have been compared with estimation of distribution algorithms (EDAs) for the task of producing cancer chemotherapy schedules [7]. EDAs are a byproduct of GA work, first detailed in [83]. They use a probabilistic sampling of the population rather than mutation or crossover and use the information gained from this sampling to produce the next set of population members. GAs were again outperformed for deriving optimal control schedules . EDAs proved more effective than GAs with regard to both the speed at which a feasible treatment could be found and the fitness of solution found.

The efficiency of the GA for problems of this type can be improved through various means. Tuning of those factors which affect the optimisation process [68] can increase the speed of the GA as can changing the encoding for chromosomes to an integer representation as opposed to the traditional binary approach [84]. The GA search process for optimal control problems can also be improved through incorporation of problem specific heuristic arguments into the GA process [85].

As stated previously, GAs have been successfully applied to optimal control problems. This section has described that when compared to more sophisticated approaches, GAs are typically slower to find solutions. GAs are however a computationally cheap approach to problems of this type.

This raises the question as to whether there is there another way of enhancing the GA to quickly find good solutions to optimal control problems? Is there any information contained in good solutions that can be harnessed to direct offspring to promising areas of the search space in a quick and efficient manner? If the search process could be directed through observing the information contained in highly fit solutions, this could speed the discovery of good solutions. As stated in Section 2.1.6, crossover is key to the GA process and is considered more innovative than mutation. An extension to enhance crossover could therefore provide a more effective search for problems of this type and improve the GA performance.

One such technique, which reviews the structure of parent chromosomes to better create offspring is introduced in the following section.

3.3 Possible extensions to conventional crossover techniques

The previous section discussed optimisation problems which require scheduling of interventions of some form across a time period to provide maximum utility. Most evolutionary approaches to optimal control problems use a coding system which would set the control value to be used at a definite time in the interval [ti:tf], for instance [2;6] [64]. Representations of these interventions could take many forms, from simple alternatives such as a binary string encoding of a bang bang problem, where a ‘1’ represents an intervention at that point and a ‘0’ represents no intervention, to more complex representations, such as an integer encoding defining the strength of intervention dosage or action to perform.

As detailed in Section 2.1.6, crossover is key to the successful operation of a GA. For this reason, it appears logical to first review modification of this part of the GA process to suit problems of this type, as this will directly affect the key element in the GA process.

Conventional GA crossover approaches, such as those discussed in Section 2.1.6 have been used to produce solutions to these types of problems. However, traditional techniques such as UC and SPC do not consider the number of interventions contained in parent solutions and instead just blindly crossover genetic material.

A simple example of this issue can be represented by a situation where an optimisation function requires interventions to be scheduled over a ten day period, with one potential intervention per day. This could be easily represented as a 10 bit binary string, where a ‘0’ represented a non intervention and a ‘1’ an intervention.

If two solutions are selected for crossover, one which is averagely fit, with three interventions:

0	0	1	1	0	0	0	1	0	0

and a much fitter chromosome with seven interventions:

1	0	1	1	0	1	1	0	1	1

Standard crossover approaches could be used to produce new solutions. However, as the solution with more interventions is considerably fitter than the solution containing less interventions, and both have been selected to produce offspring, is there some useful information as to the beneficial number of interventions for offspring contained in this? SPC would randomly pick a crossover point at which to recombine material and UC would work probabilistically along the solutions, picking genes from each parent and would, assuming a conventional value for the swapping probability $n = 0.5$, take roughly half the genes from parent 1 and half the genes from parent 2. Could there be any advantage in ensuring that offspring have a number of interventions which is close to that of the fitter parent?

Although a simple example which ignores the non-linear landscape of many complex problems, it does however introduce the question as to whether it may be beneficial for conventional techniques such as UC to consider the number of interventions which appear to offer more utility? However, even if considering the number of interventions for simple binary bang bang style representations proved effective, how would these considerations work for gene representations which include a range of allowable values for an intervention? For example, in a situation where different dosage strengths can be represented for each intervention. In order to answer these questions various novel crossover approaches have been constructed and these are detailed in the next chapter.

3.4 Summary

This chapter has described optimal control problems. This began with an introduction to this type of problem followed by a description of GA applications to optimal control problems. This showed that although GAs can be used to solve problems of this type, they have been outperformed by other search heuristics.

In order to improve the effectiveness of GAs for problems of this type, an extension to the GA crossover approach was proposed. This extension reviewed the number of interventions contained in chromosomes selected for offspring production, and uses this information to calculate offspring intervention levels. Chapter 4 details various implementations of this extension.

Chapter 4

Directed intervention crossover approaches

The previous chapter discussed problems which require interventions to be made at points across some time period. Chapter 3 concluded by asking whether GA crossover techniques could be made more effective for these types of problems through reviewing the number of interventions present in fit parents and using this to determine the optimal number of interventions for offspring.

To answer this question, techniques need to be constructed both for deciding upon intervention levels for offspring and also how to select the appropriate interventions from parents. The creation and investigation of such techniques form the main focus of this work and due to the processes involved, these approaches are termed directed intervention crossover techniques.

To increase clarity in the following descriptions, some definitions are necessary. In this work, each gene represents a possible intervention of some action at a specified time and if a gene is set to 0, this represents no intervention at that point in time and any other value represents an intervention. Thus all genes with a non-zero value will be considered interventions. The intervention level of a solution is simply a count of all these non-zero genes contained in a chromosome.

The next section reviews how intervention levels for offspring are calculated and the following sections outline techniques for selecting appropriate numbers of interventions from parents.

4.1 Calculating target interventions for offspring

Various alternatives could be used for calculating the number of interventions for offspring based on intervention levels in parents. One could simply use the intervention levels of the fitter parent as the number of interventions which children should contain; however, as variance in the population is vital to successful interrogation of the search space, this may preclude a thorough analysis of the search space and prompt premature convergence. Another alternative could be to vary the offspring intervention levels around that of the fitter parent, while allowing some variance in the sizing. This would allow offspring to further explore the search space around that of the fitter parent.

The latter of these alternatives formed the starting point for this work, although the hypothesis of premature convergence for techniques using just the fitter parent intervention level is tested in the Directed Uniform Crossover approach as described in Section 4.2.3.

This work focusses on the traditional GA approach where two parents are utilised to produce two offspring, although Section 10.3.2 identifies possible extensions to this. This approach raises the question as to how to vary the intervention levels for offspring around that of the fitter parent? As this is a new area of research, there are currently no techniques for performing these calculations. Many alternative approaches could be constructed, but the one chosen for initial investigation, which also takes into account the fitness level of the less fit parent, was to use the difference in intervention levels between parents as bounds for exploration around the fitter parents intervention level. This presented a logical starting point for experimentation and works as follows:

The first step of the sizing 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 utilised 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:

$$D_i = |I_1 - I_2| \tag{4.1}$$

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:

$$I_T = I_F - \frac{D_i}{2} + rand(D_i) \tag{4.2}$$

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 $rand(x)$ returns a random real value between 0 and x. If D_i is even, the window of variance is simply calculated as $D_i/2$. However in order to always have an even window, if D_i is odd, one is added to D_i before I_T is calculated. In order to enforce variety in the population to avoid stagnation, if $D_i < 2$, for example when both parents are of the same size, D_i is set to 2 before I_T is calculated, thus always giving a window of variability of at least one either side of the fitter parent size.

This approach results in I_F acting as the centre point for the mean target intervention level with bounds determined by the difference between the two parent intervention levels. The rationale behind this method is that the fitter parent should act as the guide for a new set of interventions but that exploration around this point should be encouraged.

To clarify this process, consider the following two parent chromosomes representing intervention schedules across time. They use a binary representation where a 1 represents an intervention and a 0 represents no intervention at the given point in time. Let us assume that the first parent chromosome has a weaker fitness than the second parent chromosome.

0	0	1	1	0	0	0	1	0	0

1	0	1	1	0	1	1	0	1	1

Thus I_F = 7 and $D_i = 4$, since $I_1 = 3$, and $I_2 = 7$. Given the stochastic nature of Equation 4.2, the target intervention level I_T, will therefore be a value in the range 5 to 9.

A new value of I_T is calculated for each child separately, thus producing more variety in the offspring and consequently, the population.

It is interesting to note that if either UC or SPC were presented with these parents for producing offspring, the children must contain interventions at locus 3 and 4, and with the exception of mutation, have no other way to remove these from offspring. Using the sizing technique described above, this allows for the crossover process to potentially shed some of these interventions which could facilitate a broader search of the solution space.

This section has described sizing calculations for determining the number of interventions to be present in offspring. These calculations include variance producing a range of possible intervention sizes which centre around that of the number of interventions contained in the fitter parent. This variance is scaled according to the difference in intervention levels between parents, thus when two parents are close in intervention size, the window of potential variance around the fitter parent intervention size will be smaller and conversely, when there is a large difference in intervention sizes between parents, a larger window of potential interventions will be used. Constraints are in place to ensure that the window is always able to be sized around I_F, and also that the number of interventions for offspring will never reduce below 0 or be greater than the interventions present in the parents. The ability of directed intervention crossover techniques to shed interventions even when they are present in both parents has been noted.

Now that calculations are in place for determining the size of offspring, the next step is to decide which of the interventions should be selected for inclusion in the offspring.

4.2 Intervention selection

When deciding upon which interventions to pass onto offspring, one concern is whether to pick genetic material via a uniform distribution over time, such as conventional treatment protocol for application of cancer treatments

[86], or whether to pick the genetic information stochastically, irrespective of the distribution over time. In order to investigate this issue, crossover techniques to demonstrate each of these options were created. Targeted Intervention with Stochastic Selection (TInSSel) and Calculated Expanding Bin (CalEB) provided techniques for initial inspection of directed intervention crossover concepts. Each of these techniques will be reviewed in turn.

4.2.1 TInSSel

Targeted Intervention with Stochastic Selection (TInSSel) is a directed intervention crossover technique which, as the name describes, picks the genetic information in a stochastic manner.

Consider the two parent chromosomes described earlier. They use a binary representation where a 1 represents an intervention and a 0 represents no intervention at the given point in time.

0	0	1	1	0	0	0	1	0	0

1	0	1	1	0	1	1	0	1	1

Having determined the number of interventions a child will have using the sizing calculation from the previous section, 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. The locus of interventions present in all parents selected for breeding are placed in the set of duplicates S_{dup}, and these intervention points will have priority in being passed to the offspring. It would appear prudent to ensure that priority is given to these genes when passing information onto offspring over genes present in only one of the parents. This is similar to the operation of UC, as if a '0' or '1' exists at a particular locus for both parents, regardless of the parent chosen to select the gene from, the duplicate value will be chosen.

The locus of interventions present in one parent but not the other are placed in the set of interventions $S\ _{single}$. From the parents described above, $S_{dup} = \{3, 4\}$ and $S_{single} = \{1, 6, 7, 8, 9, 10\}$.

Interventions from S_{dup} will be added the child, at random, until I_T is reached or no common intervention points remain. Note that if I_T is less than the size of S_{dup} then not all elements of S_{dup} will be included.

Having selected interventions common to both parents, the number of additional interventions required is therefore I_T minus the size of S_{dup}. This 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 points that will be used, interventions are randomly selected from the set S_{single} until a total of I_T points have been picked from both S_{dup} and S_{single}.

From the earlier example, all offspring for that situation would require between 5 and 9 interventions. As this is more interventions that those present in S_{dup}, other interventions will be required. In other circumstances however, two alternatives exist. If the target number of interventions $I\ _T$ was less than the number of elements contained in S_{dup}, the required number of interventions would be picked once only, at random, from $S\ _{dup}$ until I_T was reached. The other possibility would be that I_T was equal to the size of S_{dup}, in this instance offspring would simply contain all the interventions contained in $S\ _{dup}$.

From the above example, both offspring will contain interventions at locus 3 and 4. This means that dependant on the result of the $rand()$ function, between 3 and 7 other interventions will be required for offspring. For clarity, the number of remaining interventions will be termed $I\ _R$.

As TInSSel does not enforce a uniform distribution for selection of the remaining genes over time, it can simply pick from the set S_{single} until I_R is reached. Each gene can only be picked once from $S\ _{single}$. If the $rand()$ function had returned a 1, this means that I_T would be 6. As discussed, interventions from S_{dup} will be added first, thus the child will contain interventions at locus 3 and 4, and $I\ _R$ will be I_T - 2 = 4. Thus 4 interventions will be picked at random from the set $S_{single} = \{1, 6, 7, 8, 9, 10\}$. When these are added to the material from $S_{dup} = \{3, 4\}$, this allows possible offspring to include the following combinations {1,3,4,6,7,8}, {3,4,7,8,9,10}, {1,3,4,8,9,10}, among others.

4.2.2 CalEB

The Calculated Expanding Bin (CalEB) approach was created to ascertain the effectiveness of picking genetic material distributed over time. Many problems, especially of a medical nature, require interventions distributed throughout a time period. One such example is that of cancer chemotherapy treatments, where conventional treatment protocol administers cancer drugs according to such a distribution as it is easier to organise patients to be present for treatment at fixed times [86].

CalEB uses the same initial process as TInSSel for selecting interventions from S_{dup} and is identical to TInSSel when the number of interventions required for offspring, I_T, is less than or equal to the number of interventions present in S_{dup}. When I_T is greater than S_{dup}, CalEB produces a list of possible interventions, which contains I_R copies of each gene in S_{single} ordered according to their ascending locus position. This list is broken into I_R bins, and one intervention is picked from each bin at random for addition to the offspring. If an intervention has already been added to the offspring, an alternative intervention is picked, and this process is continued until an offspring of size I_T have been created.

Thus, continuing the above example, the offspring will contain interventions at locus 3 and 4 and $I_R = 4$. There would therefore be 4 copies of each intervention from S_{single}, taking the form:

$$[1, 1, 1, 1, 6, 6, 6, 6, 7, 7, 7, 7, 8, 8, 8, 8, 9, 9, 9, 9, 10, 10, 10, 10]$$

This is partitioned into I_R bins, each containing 6 interventions, producing the following:

$$[1, 1, 1, 1, 6, 6]$$

$$[6, 6, 7, 7, 7, 7]$$

$$[8, 8, 8, 8, 9, 9]$$

$$[9, 9, 10, 10, 10, 10]$$

Each bin is visited in order. If an intervention with the value ‘1’ was picked from the first bin, then either ‘6’ or ‘7’ would be valid choices from the second bin. Alternatively, if an intervention with the value ‘6’ was picked from the first bin, only an intervention with the value ‘7’ would be able to be picked from the second bin. Once a valid intervention has been picked from each bin, these values twinned with the interventions already added from S_{dup} will then be used to set the appropriate intervention values for the offspring. Thus if {1,6,8,9} were picked from the four bins, these would be added to the {3,4} from S_{dup}, producing {1,3,4,6,8,9}. In a binary environment, this would therefore produce the following bit string:

1	0	1	1	0	1	0	1	1	0

The above chromosome shows that the genes at the locus contained in the list of genes to set are all set to the value of ‘1’, while all other gene values are left with a ‘no-intervention’ value of ‘0’.

4.2.3 Directed Uniform Crossover

In Section 4.1 an alternative for calculating the target number of interventions for children was introduced whereby the target is set to the number of interventions contained in the fitter of the parents selected for crossover. This in effect would set I_T to be equivalent to I_F. This was dismissed as the basis for the directed intervention crossover techniques as it was felt that this greedy style of intervention calculation, where the target is the current best, may preclude a thorough analysis of the search space and prompt premature convergence. In order to ascertain if these assumptions are correct, a technique based on the concept of using the number of interventions contained in the fitter parent as an absolute intervention limit for offspring was created. This approach is termed Directed Uniform Crossover (DUC).

Using two of the parents defined earlier, whereby:

Parent 1 is of average fitness and contains three interventions:

0	0	1	1	0	0	0	1	0	0

Parent 2 is considerably fitter and contains seven interventions:

1	0	1	1	0	1	1	0	1	1

as $I_F = 7$, $I_T = 7$.

The required number of interventions, I_T are then selected from the parents in the same stochastic method as described for the TInSSel technique.

4.3 Summary

This chapter has introduced three novel GA crossover techniques, TInSSel, CalEB and DUC for the optimisation of optimal control problems. Each of these approaches are in the category of directed intervention crossover, whereby the number of interventions for offspring are calculated in some way from the parent intervention numbers.

Both TInSSel and CalEB use a window of search around the number of interventions in the fitter parent, facilitating search around promising areas of the solution space. The intervention target calculation for DUC simply uses the number of interventions contained in the fitter parent for the offspring intervention level. Once the intervention target is calculated, each of the approaches select interventions present in both parents before those interventions only present in one. If more interventions are still required, both TInSSel and DUC pick in a stochastic manner from the remaining set of interventions, whereas CalEB picks interventions in a uniform distribution over time.

In order to evaluate the directed intervention crossover techniques, suitable optimal control test problems were required. A bio-control scheduling problem and a chemotherapy scheduling problem were chosen for test environments, and the following chapter will review each of these problems.

Chapter 5

Test problems

In order to ascertain the effectiveness of the directed intervention techniques introduced in Chapter 4, suitable test problems are required. These problems need to be optimal control problems which require discovery of highly fit intervention schedules. This chapter introduces the test problems under review, that of bio-control optimisation and cancer chemotherapy treatment scheduling.

In the following section (Section 5.1), an introduction to the biological aspects of the bio-control problem will be outlined, followed by the mathematical representation of the model dynamics. This will be followed by Section 5.4, which details similar information for the cancer chemotherapy treatment scheduling problem.

5.1 Bio-control problem

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. Both chemical and biological controls have been used against this pest. Biological controls are beneficial as it avoids the build up of chemical controls which can be detrimental to the environment and can lead to pesticide resistance [87].

As constraints about the use of chemical insecticides have increased, entomopathogenic nematodes of the genera Steinernema and Heterorhabditis have been shown as an environmentally safe alternative [88]. These nematodes are isolated from a wide variety of ecosystems ranging from sub-Arctic to arid and tropical climates [89]. Entomopathogenic nematodes of the genera Heterorhabditis and Steinernema are commercially used to control pest insects [90] with Steinernema feltiae proving the most effective nematode for controlling sciarid species [91]. Scientific evidence supports the conclusion that using nematode worm Steinernema feltiae control is safe to the environment, application personnel, the general public and the consumers of agricultural products treated with them [92].

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. 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 farmer to make an informed decision, based on their priorities and preferences, regarding which approach to implement. The mathematical formulation of this problem follows.

5.2 Problem formulation

In [93], a generalized model for the lifecycle of sciarid flies is specified which includes potential infection from Steinernema feltiae. The lifecycle for sciarid flies is shown in Figure 5.1. This shows that the adults lay eggs which turn into larvae. These larvae pupate and become adults which subsequently lay more eggs.

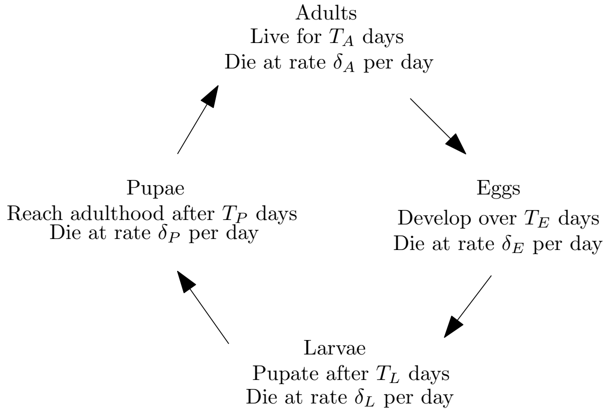

Adults
Live for T_A days
Die at rate δ_A per day
Eggs
Develop over T_E days
Die at rate δ_E per day
Larvae
Pupate after T_L days
Die at rate δ_L per day
Pupae
Reach adulthood after T_P days
Die at rate δ_P per day

Figure 5.1: The Sciarid Fly Lifecycle

Parameter	Description	Constant Value
E,L,P,A	Egg/larval/pupal/adult host density	Variable
N	Free living infective nematode density	Variable
T_E, T_L, T_P, T_A	Duration of egg/larval/pupal/adult stage	5, 10, 5, 8 (days)
$\delta_E, \delta_L, \delta_P, \delta_A$	Daily mortality rate of egg/larvae/pupae/adult	0.35, 0.125, 0.1, 0.275
ρ	Viable eggs laid per adult pest	75
β	Infection rate of nematodes	0.000095
λ	Nematodes produced per infected host	4000
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

Table 5.1: List and Definition of Model Parameters

In this model, there is a set of discrete intervention points $(t_1, t_2, ..., t_n)$ where nematodes can be applied to the system. 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. As the nematode control agent is only effective against sciarid flies when they are in their larval stage, the nematode intervention schedule needs to maximize the impact on this particular stage of the fly's development. The equations specified in [93] are used to model the effects of intervention schedules. Table 5.1 defines the parameters and constants used in the equations of this model.

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.

Equations 5.1 - 5.12 are used to model the dynamics of the nematode / sciarid populations. Equations 5.1 - 5.5 model the change in eggs, larvae, pupae, adults and nematodes respectively from one time step to the next. Equation 5.6 defines the probability of surviving the egg stage, 5.7 the probability of surviving the pupae stage and 5.8, the probability of surviving the adult stage. Equation 5.9 defines the transfer rate into the egg stage. Equation 5.10 calculates the duration of the adult host stage, 5.11 the maturation rate from the larval stage and 5.12 the probability of surviving the larval stage.

$$\frac{dE(t)}{dt} = R(t) - R(t - T_E)\sigma_E - \delta_E^{E(t)} \tag{5.1}$$

$$\frac{dL(t)}{dt} = R(t - T_E)\sigma_E - M(t) - \delta_L^L(t) - \beta N(t)L(t) \tag{5.2}$$

$$\frac{dP(t)}{dt} = M(t) - \delta_P P(t) - M(t - T_P)\sigma_P \tag{5.3}$$

$$\frac{dA(t)}{dt} = M(t - T_P)\sigma_P - M(t - T_P - T_A)\sigma_P\sigma_A - \delta_A^A(t) \tag{5.4}$$

$$\frac{dN(t)}{dt} = \lambda\beta N(t - T_I)L(t - T_I) - \mu N(t) - \beta N(t)L(t) \tag{5.5}$$

$$\sigma_E = e^{-\delta_E T_E} \tag{5.6}$$

$$\sigma_P = e^{-\delta_P T_P} \tag{5.7}$$

$$\sigma_A = e^{-\delta_A T_A} \tag{5.8}$$

$$R(t) = \frac{\rho}{T_{AVE}} A(t) \tag{5.9}$$

$$T_{AVE} = [1 - \sigma_A]/\delta_A \tag{5.10}$$

$$M(t) = R(t - T_E - T_L)\sigma_{L(t)}\sigma_E \tag{5.11}$$

$$\sigma_{L(t)} = e^{\int_{t-T_L}^{t} \beta N(x) + \delta_L dx} \tag{5.12}$$

The fitness calculation for the model is detailed in 5.13. As detailed in Section 5.1, it is the sciarid larvae which cause damage to the crop. Thus 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.

$$F = \sum_{t=0}^{T} L(t) + NP \tag{5.13}$$

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

The model is over a 50 day period, where the farmer can either spray a crop on a day or not. Due to the nature of the control, in that it can either be on or off, this means that this problem admits a bang bang control strategy [94]. In order to encode schedules for optimisation by the GA, a 50 bit binary chromosome was used. This means that each day is represented by the bit at that particular locus in the chromosome, i.e. day 1 is represented by locus 1 and day 50 by locus 50. As the bio-control agent is either applied on a day or not, a binary representation is sufficient as a 0 represents no intervention on that day and a 1 represents application of the bio-control agent on that day. If the following chromosome represented days 1 - 4 of a schedule, this represents application of the bio-control agent on days 1, 3 and 4.

1	0	1	1

The dynamics of the sciarid larvae population in the absence of bio-control agent are shown in Figure 5.2. This shows the number of sciarid larvae present on each day of the 50 day period. The number of sciarid larvae increase, decrease and subsequently increase in a cyclical manner, as was shown in the work of Fenton et al [93]. Over a 50 day period, there are two distinct peaks in the number of sciarid larvae. At day 10, there are approximately 38 larvae before the population decreases and at day 34 there are 80 larvae present.

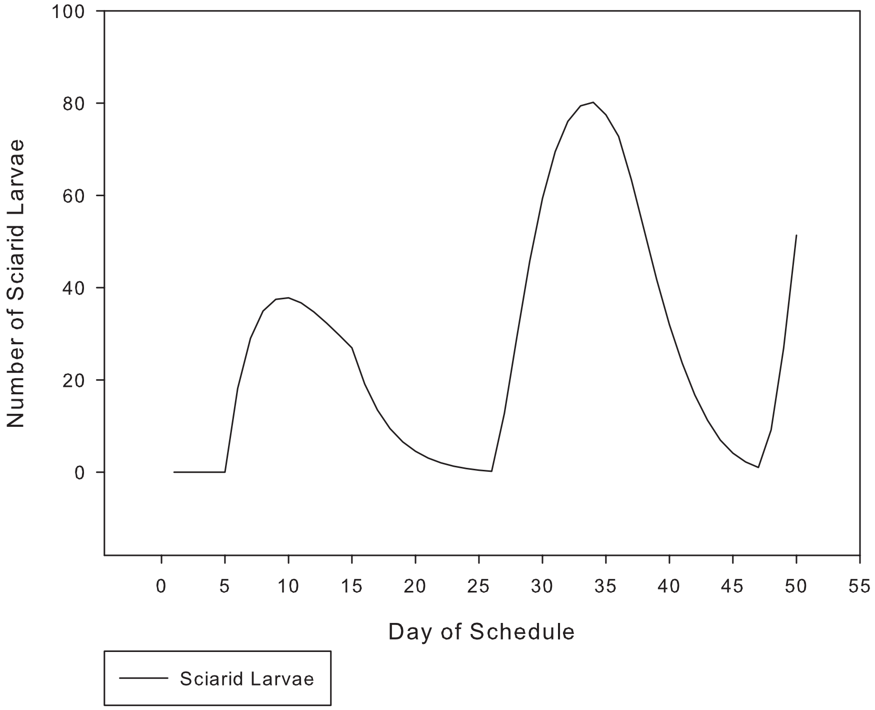


Figure 5.2: Sciarid Larvae Population

5.3 Experiment parameters

The experiment parameters for the bio-control scheduling problem will now be reviewed. Section 5.3.1 describes the limit for the number of fitness function evaluations undertaken per experiment. This is followed by Section 5.3.2, which describes the penalty values associated with each application of the bio-control treatment.

5.3.1 Fitness function evaluations

Through analysis of preliminary runs of the model for each crossover approach, it was found that 5,000 fitness function evaluations (FFEs) was sufficient to ensure convergence for all crossover approaches under review. The statistical output of the runs are therefore analysed in 100 FFEs increments for each approach, and therefore are from 100 - 5,000 FFEs.

5.3.2 Intervention penalties

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, P. This encourages the system to use as few interventions as possible while also deriving as effective a treatment as possible. A major difficulty in handling constraints using penalty function methods in GAs has been to set appropriate values for penalty parameters and this often requires users to experiment with different values of penalty parameters [95]. With too large a penalty value per intervention, the omission of interventions may become as important as the placing of key interventions. In order to fairly compare crossover techniques for the scheduling of interventions, a range of penalty values were used. Through empirical analysis, P values of 0, 5, 20, 35 and 50 per intervention were shown to represent many aspects of the search space. When P is 0 per intervention, this means that the best solution is to apply the bio-control agent at nearly every opportunity as there is no cost associated with intervening. When P is increased to 50 per intervention, the best solution appears to be application of two well placed interventions. By increasing P above 50, initial tests found that the best solution became one or zero interventions, as the penalty per intervention became higher than the benefit received from the intervention. For this reason, P was capped at 50. As the increased value of P makes placement of interventions more important, this should allow for a fair comparison of crossover techniques, without any bias introduced by any one intervention penalty.

As well as the bio-control scheduling problem, the novel crossover approaches will also be applied to the task of cancer chemotherapy treatment scheduling. Section 5.4 outlines the cancer chemotherapy scheduling problem followed by Section 5.4.1 which describes the problem formulation. Section 5.5 describes the parameter settings for the experiments.

5.4 Chemotherapy scheduling problem

Cancer is a class of diseases whereby cells display uncontrolled growth. According to the World Health Organization, it is the leading cause of death worldwide and accounted for 7.9 million deaths (around 13% of all deaths) in 2007 [96]. Chemotherapy is commonly used to combat cancer and treats the disease through using chemicals which kill cells. Composing an effective chemotherapy treatment schedule is a non-trivial task; indeed chemotherapy is often considered one of the most complex cancer treatments [97].

Through the use of deterministic mathematical models, valuable efforts in the analysis of cancer chemotherapy have been made [98]. However, the problem can be regarded as analytically intractable due to both its multi-constraint nature and the non-linearity of the optimisation functions [49]. This complexity makes this problem an ideal test to assess the abilities of the directed intervention crossover approaches in searching a multi-constraint, extensive search space where GAs have been previously successful for calculating treatment schedules [99].

As mentioned in Section 3.2, a variety of mathematical models for the optimal control of cancer treatments exist. Over recent years, a body of work by Petrovski and McCall has applied GAs to models for single objective optimisation of cancer chemotherapy [66; 67; 99; 49] as well as for multi-objective optimisation [100]. GAs and other search approaches have been analysed using these models, with alternative techniques applied including particle swarm optimisation [6], and estimation of distribution algorithms [7].

It is the single objective form of chemotherapy that is used as the fitness function for this Chapter. The multi-objective function forms part of future work for these techniques, as detailed in Section 10.3. Although the single objective model can allow for a number of different drugs to be scheduled, for transparency and initial testing, the focus will be on single drug schedule optimisation. The single objective chemotherapy model provides a well-used test function for ascertaining the effectiveness of the directed intervention crossover approaches for chemotherapy scheduling.

When encoding solutions to this problem, a binary representation is commonly used to encode dosages [66; 99; 49]. A binary representation uses 4 bits per dosage to represent dosage strengths between 0 and 15. Analytical work by Petrovski et al found a clear advantage in using an integer encoding for dosages compared to binary encoding. For this reason, integer encoding of dosage schedules will be used in these tests.

An integer encoding of solutions is more appropriate for the idea behind the directed intervention crossover approaches. As the directed intervention techniques use the number of interventions contained in parent schedules to define offspring schedules, a binary encoding where 4 bits are used to represent a single intervention would not elegantly fit with this model. For instance, with 4 bits per dosage, a dosage of strength 15 would be represented by the binary string 1111, and a dosage of strength 1 would be represented as 0001. Both of these dosages represent one dosage in the schedule, however, the directed intervention technique would count the dosage strength of 15 as 4 interventions for offspring calculation and the dosage of strength 1 as 1 intervention. In a integer encoding, whether the gene has a value of 1 or 15, as long as the dosage strength is greater than 0, it will be counted as one intervention for offspring intervention level calculation. Each non-zero gene will be counted as an intervention, and this will provide a more logical representation of interventions for directed intervention crossover.

An integer representation of the single objective chemotherapy scheduling problem provides a well tested, non-binary optimal control problem with which to test the directed intervention crossover approaches. Section 5.4.1 will now discuss the formula used for this function.

5.4.1 Problem Formulation

As mentioned in Section 5.4, an integer encoding is used to define the concentration level of the anti-cancer drugs. In this encoding, each concentration has an integer value in the range 0 to 15. A 0 represents no intervention of the drugs at that time, and 15 defines application of the maximum dose. This concentration range of 0 - 15 was considered sensible by the clinical oncologists with whom McCall et al have collaborated [86].

Parameter	Value
λ	0.7
C_{cum}	120
κ	0.045
θ	100

Table 5.2: Chemotherapy Model Parameters

If the following chromosome represented days 1 - 4 of a schedule, this represents no application of the cancer drugs on day 1 or 2, with the maximum allowable dose on day 3 and the smallest allowable dose on day 4.

0	0	15	1

The aim for the chemotherapy 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 a treatment schedule that minimises $N(T_{final})$. In the model by Petrovski and McCall, the tumour growth is defined by the Gompertz model as shown in Equation 5.14. Work by Petrovski and McCall compared a range of tumour growth models and found that the underlying tumour model does not affect the ability of the GA in the objectives of treatment [99].

Equation 5.14 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.

$$\frac{dN}{dt} = N(t)\lambda ln\frac{\theta}{N(t)} \tag{5.14}$$

A relationship exists between the concentration of anti-cancer drug and its ability to kill cells. This relation is shown in Equation 5.15 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.

$$\frac{dN}{dt} = -\kappa C(t)N(t) \tag{5.15}$$

Equations 5.14 and 5.15 are combined to form the differential equation shown in Equation 5.16, which represents the tumour response to chemotherapy.

$$\frac{dN}{dt} = N(t)\lambda ln\left(\frac{\theta}{N(t)}\right) - \kappa C(t)N(t) \tag{5.16}$$

Each treatment is evaluated through passing its encoded dose schedule, where Ci represents the dosage strength for day i and permitted dosages are in the range $0 - C_{max}$, to a simulation of the response based on Equation 5.16. The score returned by the fitness function is calculated using $ln\left(\frac{\theta}{N(T_{final})}\right)$, where N is minimised when this score is maximised. A description of the complete mathematical model can be found in [65]

The parameters used in these expressions for the experiments carried out in this work are shown in Table 5.2. These parameter settings were obtained from McCall to match those of the work of Petrovski and McCall. This allows for analysis of the directed intervention crossover techniques on a robust, well tested model. The tumour growth rate is defined by λ and the maximum cumulative dose of drug by C_{cum}. The toxicity of the drug is defined by κ and θ represents the absolute limiting size of the tumour.

As with many complex models, this model is subject to a number of constraints. These constraints are detailed in equations 5.17 - 5.19.

$$g_1(c) = C_{max} - C_i \geq 0 \tag{5.17}$$

$$g_2(c) = C_{cum} - \sum_{i=1}^{n} C_i \geq 0 \tag{5.18}$$

$$g_3(c) = N_{max} - N(t_i) \geq 0 \tag{5.19}$$

Equation 5.17 details the maximum instantaneous dose, C_{max}, for the drug, Equation 5.18 the maximum cumulative dose of the drug, C_{cum}, and Equation 5.19 the maximum permissible size of the tumour, N_{max}. The goal of cancer chemotherapy is to achieve the beneficial effects of treatment without violating the above constraints [6].

5.5 Experiment parameters

As with the bio-control problem discussed previously, each experiment reviewing cancer chemotherapy optimisation will run for a fixed number of FFEs. Section 5.5.1 details the rationale behind the chosen limit.

5.5.1 Fitness function evaluations

Although 5,000 FFEs was sufficient for analysis of the crossover approaches for bio-control scheduling, preliminary analysis of the chemotherapy problem showed a requirement for a larger number of FFEs. Through analysis of preliminary runs of the model for each crossover approach, it was found that 40,000 fitness function evaluations (FFEs) was enough to ascertain the trends and abilities of each crossover approach for this problem. The statistical output of the runs are therefore analysed in 1,000 FFEs increments for each approach, and are from 1,000 - 40,000 FFEs.

5.6 Summary

This chapter has introduced two optimal control problems with which to test the directed intervention crossover approaches described previously. Chapter 6 details the experimental method which will be used for analysis of the crossover approaches at optimising both of these problems.

Chapter 6

Experimental method

6.1 Methodology

Chapter 4 outlined novel GA crossover approaches for application to optimal control problems. In order to evaluate the effectiveness of these techniques, empirical analysis is necessary. This will allow investigation into the hypothesis detailed in Chapter 1 that the performance of GAs can be improved through using the number of interventions present in parent schedules to direct the offspring to promising areas of the search space.

Correct parameter settings for GAs are crucial to their effective traversal of the search space and it is clear that a good choice of GA parameters can lead to improved performance in most practical problems [101]. However, choosing parameter settings for a GA usually requires experimentation and there is no easy way to set them well for an arbitrary problem [14]. It is therefore important to evaluate approaches over a range of parameter values in order to gauge their performance in a fair and balanced manner.

In order to address this aspect, a structured approach was taken to study algorithm performance. This chapter described this approach and the rationale behind it.

6.2 Parameter Selection

Chapter 2 outlined the various considerations present when using a GA. Mitchell has indicated that although individual optimisation of parameters such as population size and crossover or mutation rate appears logical, these parameters tend to have nonlinear relationships, and cannot be optimised individually [11]. It is therefore important that a spread of parameter combinations are evaluated for determining the robustness and effectiveness of the crossover approaches under review. Typically a range of population sizes, crossover and mutation rates are empirically evaluated to decide on settings [14].

In addition to settings for the population size, crossover rate and mutation rate, other concerns for the GA practitioner include deciding upon an appropriate selection mechanism, crossover approaches and replacement strategies. The rationale for settings for each of these parameters will now be reviewed.

6.3 Population Size

When using a GA, the decision maker has to decide on the number of solutions to be present in the population. As described in Section 2.3.1, the effective performance of the GA method depends upon this parameter and it has a major influence on the successful convergence of the GA to the optimum solution [102]. Goldberg observed that if the population size is too small the genetic algorithm will converge too quickly whereas if the population size is too large it will take a relatively longer time for significant improvement [103]. As the population size can have such an large effect on the performance of the algorithms, it appears prudent to experiment with a range of values. Each of the experiments are therefore reviewed for population sizes of 50, 100 and 150. Although smaller population sizes have been successfully used elsewhere, they may require the selection of an initial population in a systematic way rather than a more conventional random initialisation approach [51].

Alander conducted experimental work which suggests that for some problems classes, a population size of between n and $2n$ should be considered, where n is the bitstring length [104]. This finding, however, cannot be generalized to cover all problems for GA applications. As described in Section 5.2, the bio-control problem is encoded as a 50 gene bitstring and using Alander's analysis, this would point towards populations of between 50 and 100 for this problem being sufficient. Petrovski et al found that the optimal population size for the chemotherapy optimisation problem is approximately 75 [84]. This means that the population sizes, N, of 50 - 150 should be an appropriate range for both of these problems.

6.4 Crossover Rate

Although crossover probabilities of 1.0 are common in the literature[105], it is prudent to evaluate the approaches over a range of p_c values. For this reason, the experiments will review the novel techniques over crossover rates of 1.0, 0.9 and 0.8.

When $p_c = 0.9$, this means that, on average, 1 in 10 of children inserted into the next generation are purely replicas of one or the other of the parents selected for crossover. Similarly, when $p_c = 0.8$, on average, 2 in 10 of the children being added to the next generation of solutions are simply replicas of the parent genes. Although lower crossover rates, such as $p_c = 0.3$, could have been undertaken, as values in this range are quite uncommon in literature, p_c values in the range 0.8, 0.9 and 1.0 offered a more standard parameter range for experimentation.

6.5 Mutation Rate

In the absence of mutation the evolutionary process stagnates after several generations and cannot proceed [106]. This stagnation is because without mutation, the only variation in the population is that introduced when the population is first initialised. Although crossover can recombine genes, it cannot introduce any new genetic material into the population, hence stagnation occurs.

Conversely, if the mutation rate is set too high, the problem becomes almost random due to the abundance of new material for the algorithms to sort through.

In order to evaluate the approaches over a range of situations, experiments will consider three levels of mutation, 0, 0.005 and 0.05. This will show the ability of the algorithms to cope with no mutation, a median level of mutation and a high level of mutation one order of magnitude greater than the median level. Through reviewing these experiments, this will show how the techniques deal with instances where no new material is being added to the population compared with the case where large quantities are introduced, as well as a balance between these ranges.

6.6 Selection Mechanism

Various alternative selection mechanisms can be used and there is currently no theory as to which selection approach should be used over others for all problems. Indeed, it has been shown that all selection algorithms suffer from loss of diversity for some reason or another [107].

In this work, binary tournament selection has been used for all experimentation. This form of tournament selection has a low selection pressure and allows for less fit solutions to contribute to offspring. Furthermore, this form of selection offers no advantage to any particular crossover approach under review and thus allows for a fair and unbiased analysis of each of the crossover approaches. As described in Section 2.1.5, tournament selection has been shown to provide better or equivalent convergence and computational properties when compared to alternative approaches [25].

6.7 Crossover Approaches

Chapter 4 introduced novel directed intervention crossover operators for application to optimal control problems. In order to determine the effectiveness of these approaches it is necessary for them to be compared to standard GA crossover techniques.

Rather than comparing all standard approaches with the directed techniques for all experiments, it would appear prudent for evaluative purposes to first compare the standard techniques only. If one of the standard techniques is consistently as good as the other techniques or better, this would allow the directed techniques to be compared with this, facilitating reduced computational overhead and statistical comparisons. For this reason, for both optimal control models under consideration, the first set of experiments will compare uniform crossover, single point crossover and two point crossover to determine if a single approach is better than others for these optimal control problems. This approach will then be compared with CalEB, TInSSel and DUC to allow for clearer comparisons as to the efficiency of the directed techniques.

6.8 Replacement Strategy

Section 2.1.8 described two common approaches to replacing offspring in a population, namely the generational and the steady state approach. When configuring a GA, there is no defining rationale with regards to which of these approaches should be used for a particular problem. In order to ensure that the crossover techniques under evaluation are not receiving any unfair advantage through the replacement strategy used, both generational and steady state replacement experiments are examined.

Rather than duplicating all graphs and statistical analysis by reviewing each experiment setting for both generational and steady-state, a reduced approach was undertaken. As generational is the more common approach to replacing offspring into the population, each experiment will be reviewed in full for this replacement strategy. Steady-state replacement will be considered for the median case for each experiment, thus from the population size described in Section 6.3, this will be for a population of 100 and from the crossover rate described in Section 6.4, p_c of 0.9 will be used, for p_m of 0, 0.005 and 0.05.

To ensure that the best member of the population is not lost due to replacement, an elitist strategy is used for both generational and steady state replacement strategies. This ensures that the single best member of the population will not be replaced from one generation to the succeeding generation.

6.9 Statistical Analysis

In order to determine the effect of the novel crossover approaches to optimal control problems, statistical analysis of the results is necessary. By conducting the appropriate statistical tests, information can be derived as to the efficiency, robustness and differences between approaches. If the data points are normally distributed, then techniques such as t-tests and the plotting of mean values with standard deviations and standard error would be appropriate. However, if the data points are not normally distributed, then non-parametric techniques such as the Kruskal-Wallis one-way analysis of variance, and plotting of median and inter-quartile range values would more accurately represent the data and its divergences.

In order to check the distribution of the data points, twenty samples were selected at random from the underlying data set across all runs. Each sample represents the best fitness score found for each of the 100 runs of the parameter configuration. By analysing these samples through the use of the Shapiro-Wilk test and a histogram, information concerning the normality of the data can be obtained. The Shapiro-Wilk test is one of a number of techniques that check for the normality of the underlying population. It evaluates the null hypothesis that a sample came from a normally distributed sample. The larger the test statistic, the more probable the sample came from a normally distributed sample and thus the null hypothesis can be rejected if the test statistic becomes too small (< 0.05).

Chapter 5 introduced both of the problems under review. There were two classes of problem from which data could be analysed, the bio-control scheduling problem and the cancer chemotherapy scheduling problem. As described in Section 5.3.1, the bio-control scheduling data is sampled over 5,000 FFEs, in increments of 100, from 100 - 5,000. The bio-control problem is also sub-divided into penalty values (P) of 0, 5, 20, 35 and 50, as described in Section 5.3.2.

The chemotherapy scheduling data spans 40,000 fitness function evaluations (FFEs) and is sampled in increments of 1,000 from 1,000 - 40,000 FFEs as outlined in Section 5.5.1.

Both the problems are then divided into experiments by population sizes, N, of 50, 100 and 150, crossover rate, p_c, of 0.8, 0.9 and 1.0 and mutation rates, p_m, of 0, 0.005 and 0.05. In each of these experiments there are 6 crossover techniques used, SPC, 2PC, UC, CalEB, TInSSel and DUC. The following 20 data points were chosen at random to ascertain whether data points are normally distributed.

1. Bio-control, 1,300 FFEs, $P = 50$, $N = 100$, $p_c = 0.8$, $p_m = 0$, SPC
2. Bio-control, 700 FFEs, $P = 20$, $N = 100$, $p_c = 1.0$, $p_m = 0$, 2PC

3. Bio-control, 4,000 FFEs, $P = 20$, $N = 50$, $p_c = 0.8$, $p_m = 0$,UC
4. Bio-control, 3,200 FFEs, $P = 0$, $N = 50$, $p_c = 1.0$, $p_m = 0.05$, 2PC
5. Bio-control, 200 FFEs, $P = 35$, $N = 150$, $p_c = 0.9$, $p_m = 0.005$, 2PC
6. Bio-control, 4,500 FFEs, $P = 50$, $N = 100$, $p_c = 0.8$, $p_m = 0.005$, UC
7. Bio-control, 4,000 FFEs, $P = 20$, $N = 150$, $p_c = 0.9$, $p_m = 0$, UC
8. Bio-control, 2,300 FFEs, $P = 35$, $N = 100$, $p_c = 0.8$, $p_m = 0.05$,TInSSel
9. Bio-control, 1,200 FFEs, $P = 50$, $N = 150$, $p_c = 0.8$, $p_m = 0.05$,DUC
10. Bio-control, 700 FFEs, $P = 20$, $N = 100$, $p_c = 0.9$, $p_m = 0$, 2PC
11. Bio-control, 3,400 FFEs, $P = 50$, $N = 150$, $p_c = 1.0$, $p_m = 0.005$, UC
12. Chemotherapy, 29,000 FFEs, $N = 50$, $p_c = 1.0$, $p_m = 0.005$,TInSSel
13. Chemotherapy, 11,000 FFEs, $N = 50$, $p_c = 1.0$, $p_m = 0$, SPC
14. Chemotherapy, 4,000 FFEs, $N = 50$, $p_c = 0.8$, $p_m = 0.005$, SPC
15. Chemotherapy, 24,000 FFEs, $N = 150$, $p_c = 0.8$, $p_m = 0$, 2PC
16. Chemotherapy, 36,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$,TInSSel
17. Chemotherapy, 11,000 FFEs, $N = 150$, $p_c = 1.0$, $p_m = 0.05$, UC
18. Chemotherapy, 32,000 FFEs, $N = 50$, $p_c = 0.8$, $p_m = 0$, UC
19. Chemotherapy, 23,000 FFEs, $N = 50$, $p_c = 1.0$, $p_m = 0$,CalEB
20. Chemotherapy, 36,000 FFEs, $N = 100$, $p_c = 0.8$, $p_m = 0.05$,TInSSel

Figures 6.1 - 6.4 show the histogram plot for each of the 20 test cases. The label for each figure shows the associated Shapiro-Wilk test statistic.

Through analysing these tests, it is clear that normality cannot be assumed in this data. Test cases 3, 4, 5, 7, 8, 11, 14, 15, 16 and 19 all have a test statistic of less than 0.05 and thus these data points are not from a normally distributed sample. The test statistic could not be computed for test case 6 as the data had all converged to one value. As 11 of the 20 data points are not normally distributed, it is prudent to use non-parametric statistical techniques for the analysis of results. These non-parametric techniques will now be described.

6.9.1 Non-parametric Statistics

As shown in Section 6.9, normality cannot be assumed for the data in this work. As there are more than 2 samples for comparison the Kolmogorov-Smirnov cannot be used, thus all statistical analysis will use a Kruskal-Wallis one-way analysis of variance. The Kruskal-Wallis is an extension of the Mann-Whitney U test for 3 or more groups. This test will return a p-value which will show the probability that samples are from different populations. Due to the number of statistical tests per experiment, results will be evaluated at the 99% confidence level. Therefore when a significant p-value (< 0.01) indicates that at least one of the groups is different from at least one of the others. This test does not however identify which groups are different, thus in order to distinguish this information, the formula detailed in Equation 6.1 is used [108]. This allows us to test the hypothesis $H_0 : \theta_u = \theta_v$ against $H_1 : \theta_u \neq \theta_v$ for the two groups *u* and *v*.

$$|\bar{R}_u - \bar{R}_v| \geq Z_{\alpha/k(k-1)} \sqrt{\frac{N(N+1)}{12} (\frac{1}{n_u} + \frac{1}{n_v})} \tag{6.1}$$

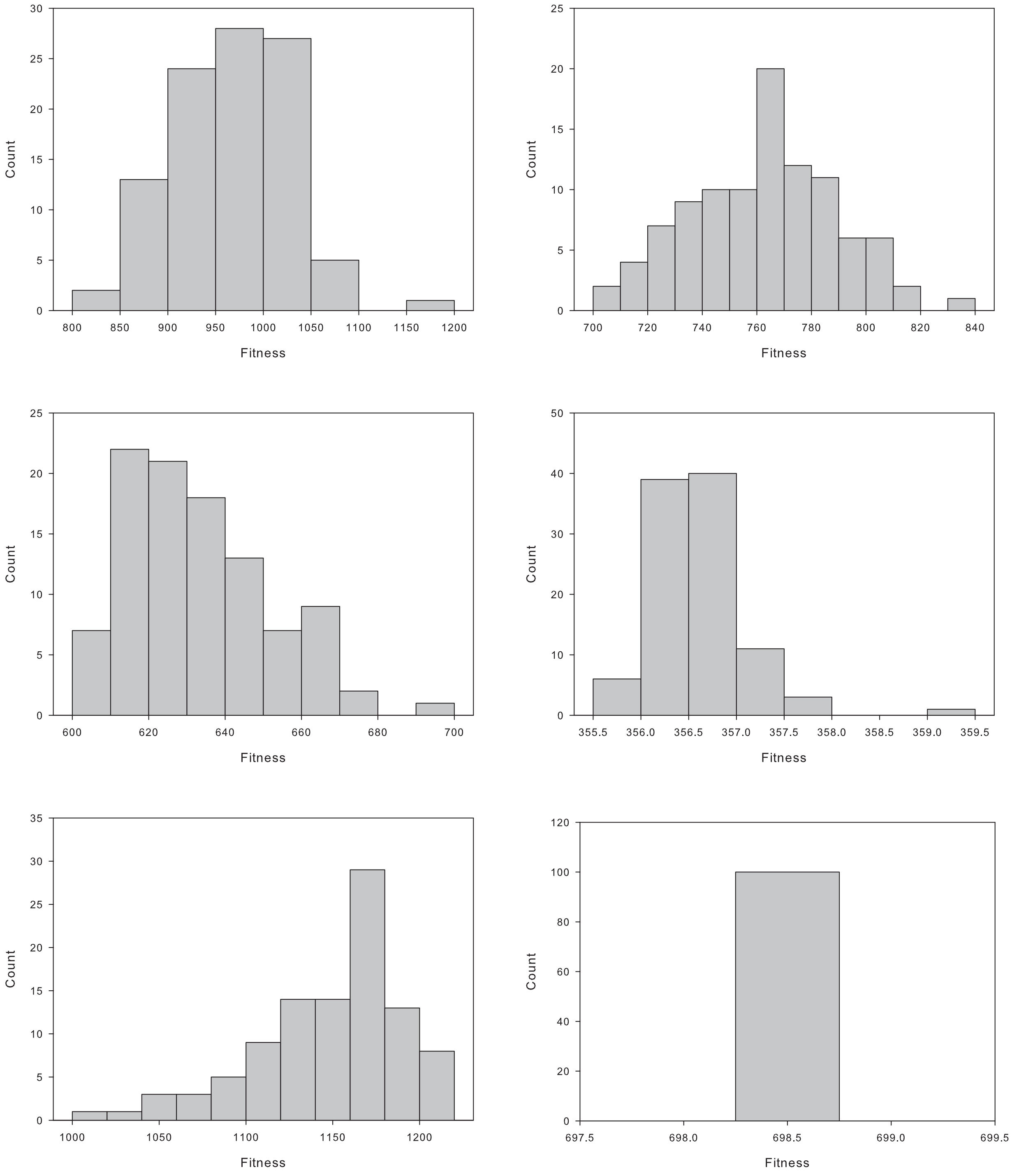


Figure 6.1: Top row shows samples 1 and 2, middle row samples 3 and 4, and bottom row samples 5 and 6 with Shapiro-Wilk p-values of 0.4741, 0.6480, 0.0060, 2.9500E-07, 0.0002 and uncomputable respectively

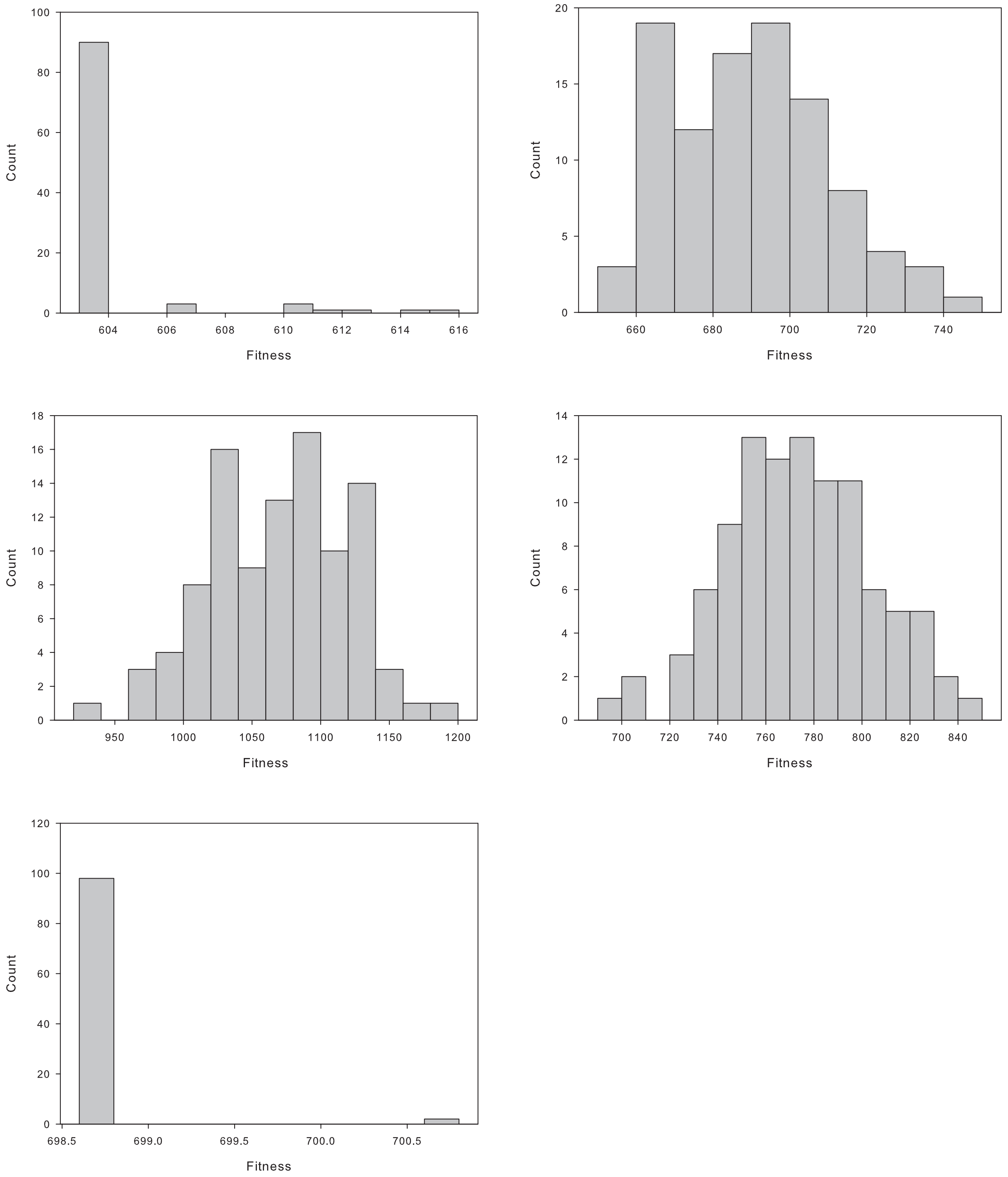


Figure 6.2: Top row shows samples 7 and 8, middle row samples 9 and 10, and bottom row sample 11 with Shapiro-Wilk p-values of 3.0434E-19, 0.0050, 0.7690, 0.9436 and 1.0429E-21 respectively

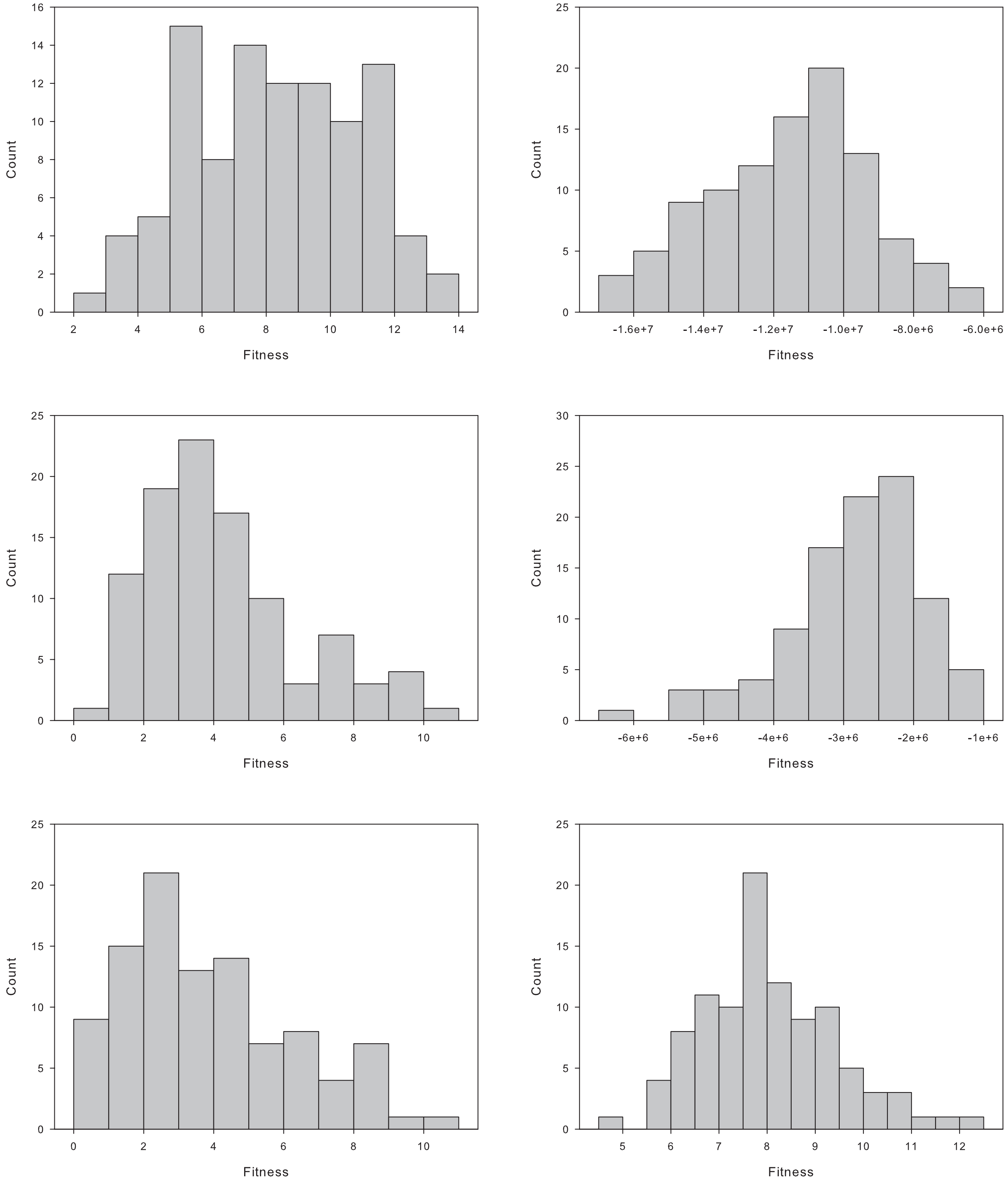


Figure 6.3: Top row shows samples 12 and 13, middle row samples 14 and 15, and bottom row samples 16 and 17 with Shapiro-Wilk p-values of 0.0523, 0.4190, 4.2309E-05, 0.0004, 0.0002 and 0.1453 respectively

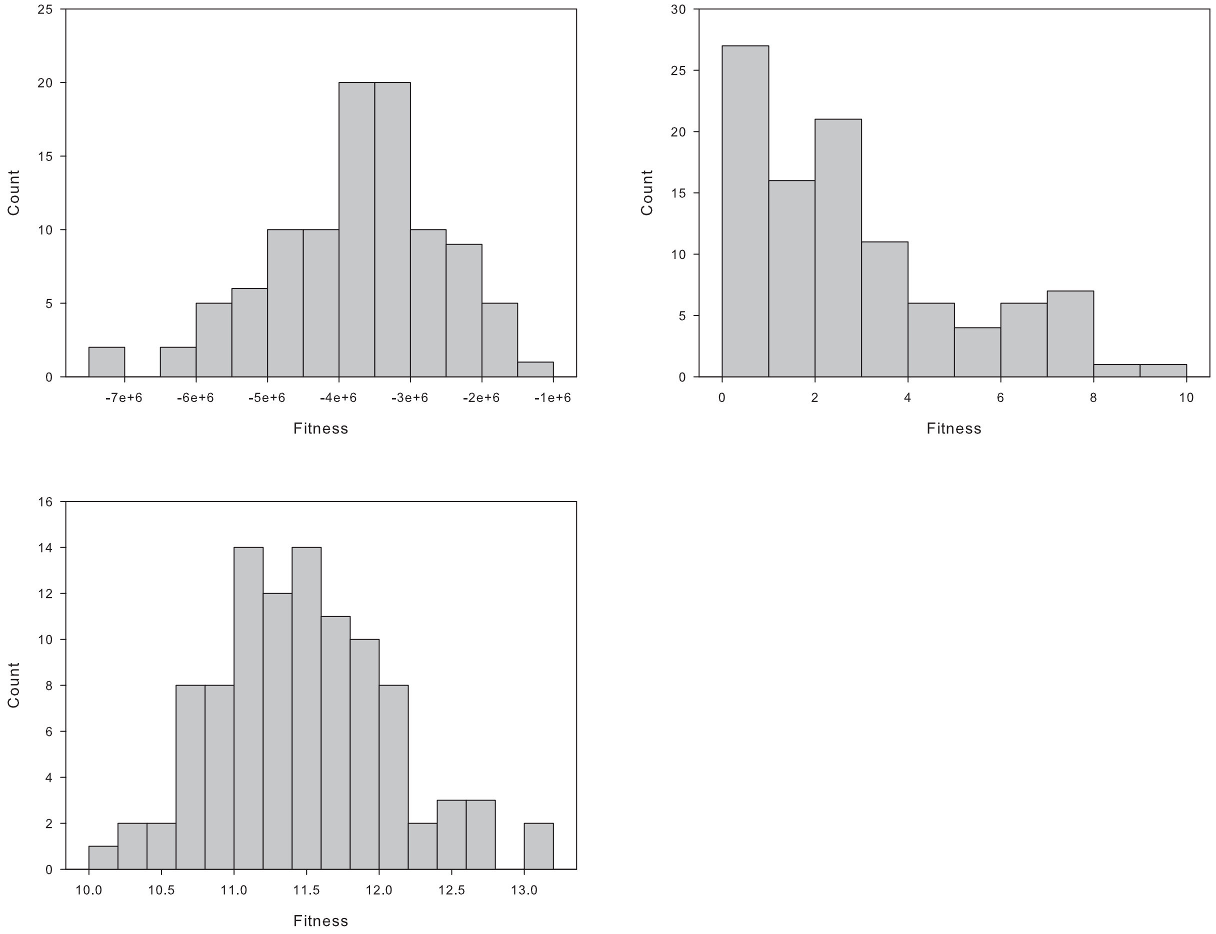


Figure 6.4: Top row shows samples 18 and 19 with sample 20 beneath with Shapiro-Wilk p-values of 0.1136, 1.0273E-06 and 0.6723 respectively

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*

The formula in Equation 6.1 will allow for analysis as to which groups statistically differ from which other groups. The analysis of groups will review 4 samples, with 100 cases per sample and an α value of 0.01. This means than when the mean ranks of groups differs by 47.99, there is a statistically significant difference between these groups.

If normality of data was assumed, numerical summaries would display the mean and standard deviation / standard error. Since normality is not assumed, all numerical summaries will plot the median and the interquartile range. As we are concentrating on finding large differences, any loss of statistical power of data where the data is in fact normally distributed is not important. Also if differences can be obtained under non-parametric tests then standard tests would tend to be more positive.

6.10 Summary

This chapter has outlined the GA parameter settings and the rationale behind their values. This has included the population size, crossover and mutation rates, selection mechanisms and the replacement strategy used for experimentation.

The data set was analysed for normality and it was found that normality could not be assumed for the experiment results. To this end, appropriate non-parametric tests were described for analysing statistical differences between samples, allowing for accurate analysis of differences between crossover approaches. As normality cannot be assumed, it was considered prudent to plot all result output showing the median and interquartile range values.

This chapter has described the experiment method that will be adhered to by the experiments undertaken in this work. All parameter values and statistical tests have been chosen in such a way as to facilitate an unbiased and accurate comparison of the crossover approaches under review.

Section 6.7 recommended that the traditional crossover approaches should first be reviewed for optimising the optimal control problems detailed in Chapter 5. In line with this, the following chapter reviews the abilities of SPC, 2PC and UC for constructing schedules for bio-control application and chemotherapy treatment.

Chapter 7

Evaluation of traditional crossover approaches

In order to determine the performance of the directed crossover approaches described in this work, it is prudent to first examine the performance of existing crossover approaches for deriving schedules for the two problems described in Chapter 5. As described in Section 2.1.6, Single Point Crossover (SPC), two point crossover (2PC) and uniform crossover (UC) are all commonly used GA crossover techniques and each represents a viable approach to test against. Rather than comparing all of the proposed novel techniques against SPC, 2PC and UC, it seems prudent to first review the traditional techniques for the optimal control test problems described in Chapter 5. If one of these techniques is consistently at least as good as the other approaches, it is logical to compare the directed techniques with this best performing traditional crossover approach. This would greatly reduce the required number of statistical tests needed to check for performance difference between the novel approaches and the existing crossover techniques.

With this in mind, the following sections review the abilities of SPC, 2PC and UC for deriving bio-control schedules for mushroom farming. The subsequent section then reviews the performance of these same algorithms for the construction of single drug treatment schedules for chemotherapy.

7.1 Evaluation of traditional crossover techniques for bio-control scheduling

For this evaluation, the bio-control problem is sampled over 5,000 fitness function evaluations (FFEs), at intervals of 100, resulting in 50 data points for review. For each sample point, 100 runs of each crossover approach were recorded for each of the parameter combinations. The best scoring values for each run for each FFE are recorded. This is consistent with the parameters used in Section 6.9. As Section 6.9.1 showed that a normal distribution cannot be assumed in these experiments, the median and interquartile range of these values will be displayed.

Each of the following sections review the effectiveness of SPC, 2PC and UC for bio-control scheduling over a range of penalty values per intervention. As detailed previously, it is important to review the crossover techniques with a range of parameter settings. A complete set of results has therefore been collected and these can be seen in Appendix B. This Appendix contains the complete set of results showing the fitness scores against FFEs associated with these experiments across a range of population sizes, intervention penalties and crossover and mutation rates.

Rather than reviewing each parameter combination in turn, it appears prudent to consider the intermediate parameter settings for each approach, instead of explicitly detailing each possible parameter combination. To this end, each of the following sections of this chapter will review SPC, 2PC and UC for the varying levels of intervention penalty P, as described in Section 5.3.2, with a population of $N = 100$, crossover rate of $p_c = 0.9$ and mutation rates, p_m, of 0, 0.005 and 0.05. One can then compare these against the complete set of results in Appendix B to check for consistency.

Initial analysis for each intervention penalty will review these intermediate parameter settings with a generational replacement strategy. A review of the placement of interventions will follow. As this review is after 5,000 FFEs, this represents the case when each of the crossover approaches have converged. This details the average placement of the interventions by each of the crossover approaches over 100 runs and will provide insight into the dynamics of the problem. This will be followed by analysis of the intermediate parameter settings under a steady state replacement strategy. The replacement mechanism, R, for each experiment will be defined as either generational, $R = g$, or steady state, $R = ss$.

The results for each of the runs could be displayed in a number of ways. Techniques such as Run Length Distributions (RLD) [109] could be used to show the performance of each of the crossover techniques, however, the approach chosen was to use graphs displaying the results for both the fitness scores and intervention usage for the intermediate parameter settings. By plotting the intervention usage and fitness scores versus FFEs over the range of FFE samples, this shows the number of interventions contained in schedules, the associated fitness of the schedules and the fitness trends and convergence characteristics of the approaches.

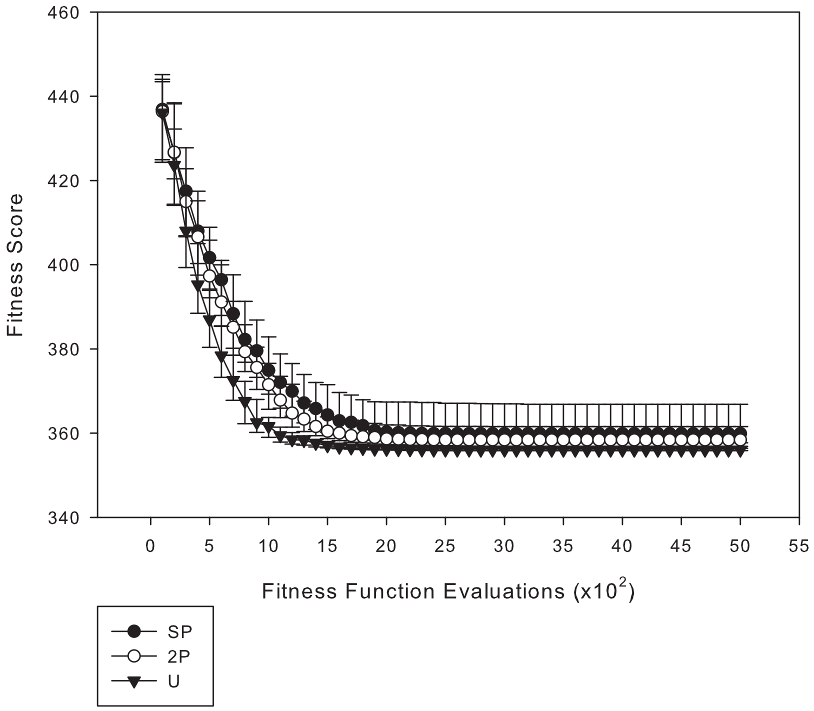
As the task of bio-control optimisation is a minimisation problem, lower scores are fitter than higher valued scores. On the graphs, SP, 2P and U represent SPC, 2PC and UC respectively.

The intervention graphs show the number of interventions associated with the corresponding fitness graph for the same FFE sample point. Intervention usage represents the number of times that the mushroom farmer requires to spray the crops, with an intervention of 0 representing no doses from the farmer and 50 interventions showing a dosage on every day of the schedule.

7.1.1 0 penalty points per intervention

This section reviews SPC, 2PC and UC for bio-control scheduling with a penalty of 0 points per intervention. As described in Section 7.1, this will focus on the intermediate parameter settings. Figures 7.1 to 7.3 show both the fitness scores and associated intervention usage for the intermediate parameter settings under varying levels of mutation, p_m. Figure 7.1 shows the fitness scores and associated intervention usage when p_m is 0, with Figure 7.2 and Figure 7.3 detailing p_m levels of 0.005 and 0.05 respectively.

Figure 7.1: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 0$ and $R = g$



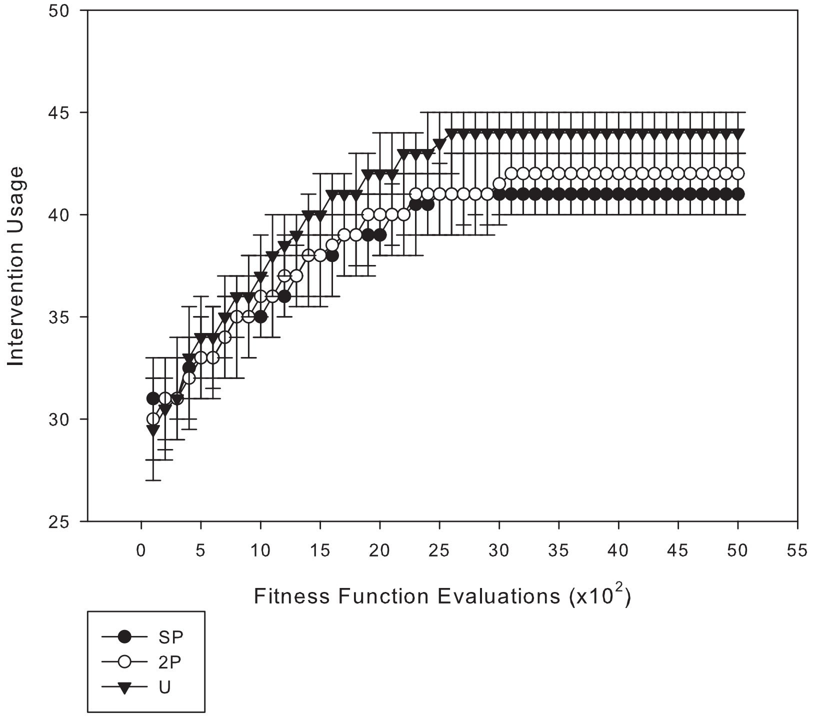
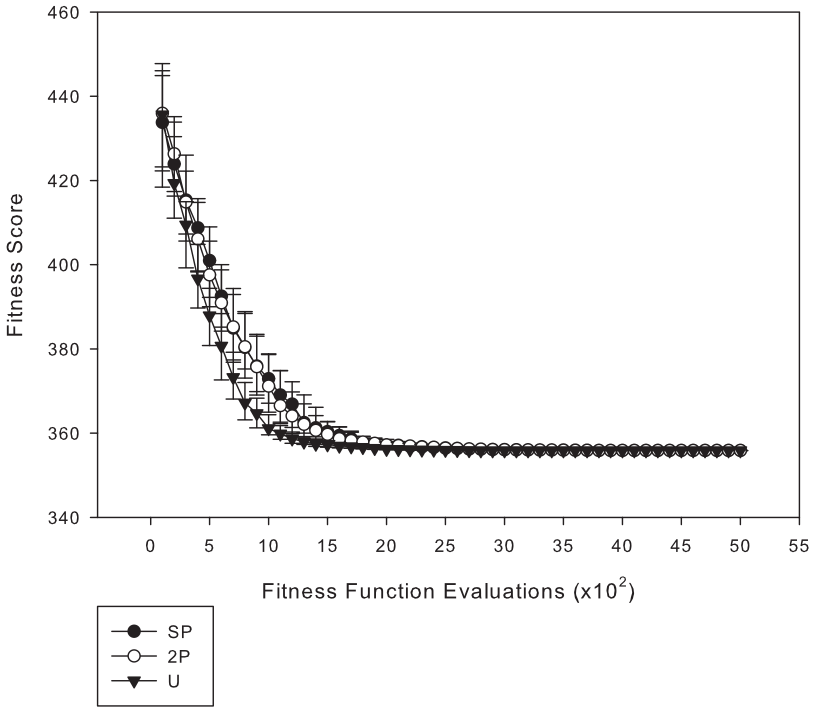


Figure 7.2: Results for $N = 100, p_c = 0.9, p_m = 0.005, P = 0$ and $R = g$



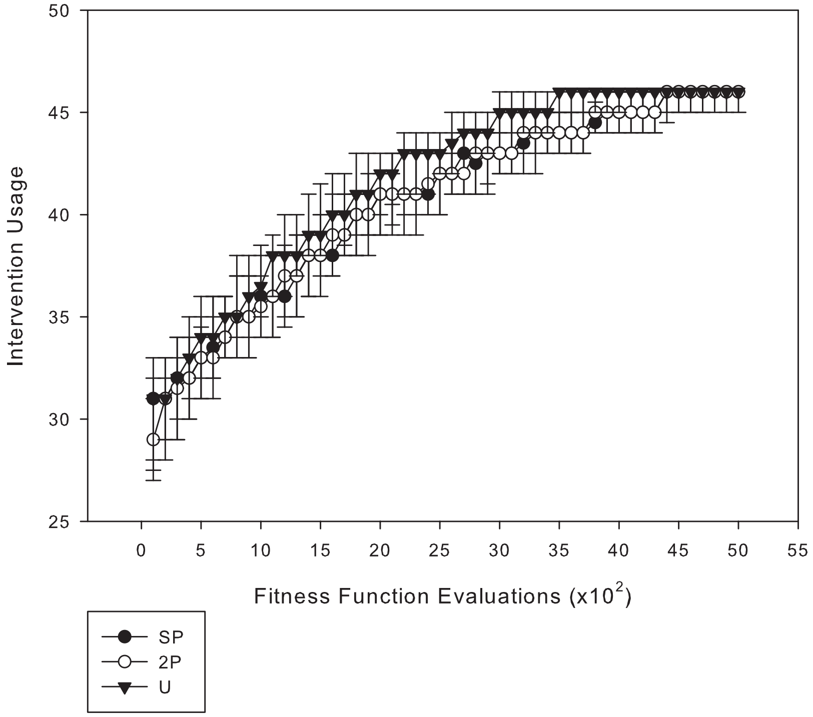
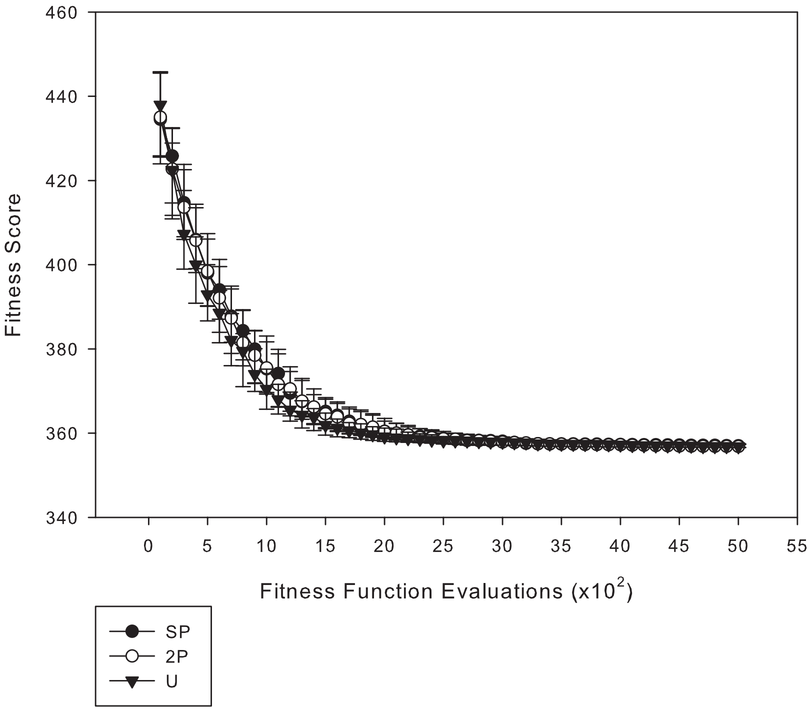
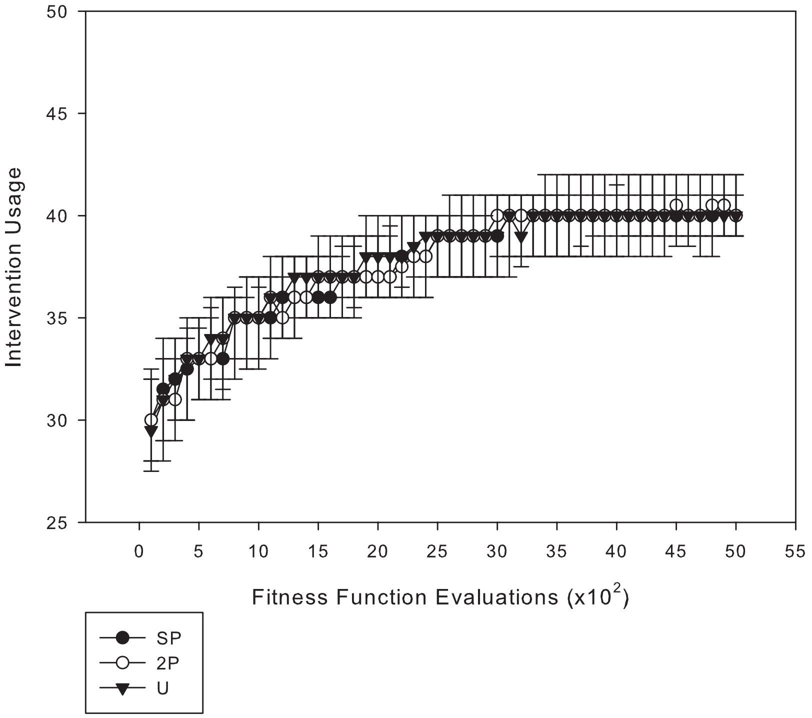


Figure 7.3: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 0$ and $R = g$





Reviewing Figure 7.1, an interesting correlation appears between UC's rapid increase in fitness with the accumulation of interventions. This shows that when there is no penalty, the optimal solution is to use a high number of interventions, a trend which UC is quicker than the other approaches to exploit. Both SPC and 2PC converge using fewer interventions than UC, with 2PC being able to accumulate more interventions than SPC in the latter stages of the experiment.

Figure 7.2 details the fitness scores and intervention usage when p_m is increased to 0.005. Scores of a similar fitness to those found when p_m was 0 are returned, but less FFEs are required to find them when p_m is 0.005. Through reviewing the interventions, UC appears quicker to accumulate interventions and this is reflected in the fitter scores for the first 1,500 FFEs.

Figure 7.3 shows the effects of increasing p_m to 0.05. All of the crossover approaches tend to use a similar number of interventions, and this similarity is displayed in the almost identical fitness scores for this level of mutation.

The high levels of intervention usage displayed by each crossover approach for each p_m level could be attributed to the 0 penalty points associated with each intervention. There is no cost associated with intervening and as the effect of each intervention is to reduce the number of sciarid larvae present, this in turn produces fitter scores. This would account for the high levels of interventions in schedules produced by each of the crossover approaches.

Intervention placement

The previous section outlined the fitness scores and associated intervention usage for the traditional crossover approaches, under a generational replacement strategy for a penalty of 0 points per intervention. In the case of no mutation (Figure 7.1), UC was shown to use more interventions than both SPC and 2PC, with 2PC in turn using more interventions than SPC. The average placements of these interventions at the end of the experiment, after 5,000 FFEs, are shown in Figure 7.4.

UC returned fitter scores after 5,000 FFEs than both of the other approaches and uses more interventions to do so. Figure 7.4 shows that on average, UC focuses more interventions than the other approaches on the first 20 days and also has more interventions on average between days 25 and 39 than the other crossover techniques. As shown previously in Figure 5.2, the first cycle of sciarid production is in the first 20 days, followed by a period of rest, followed by the second cycle of sciarid larvae in days 25 - 40. As UC has on average more applications on these days than the other approaches, this results in fitter scores than the other approaches.

As shown in Figure 7.1, in the absence of mutation, although 2PC is outperformed by UC, it returns fitter scores than SPC. 2PC uses more interventions than SPC and as shown in Figure 7.4, 2PC on average focuses more on the second cycle of larvae growth (days 25-40), than SPC. This more focussed application during the second cycle of larvae growth returns fitter scores for 2PC than the single point alternative.

The average intervention dosage will now be reviewed. This shows the average application doses from the best solutions found over each of the 100 runs. As this is bang bang problem, each intervention is either fully applied or not applied at all. If the average dosage for an intervention is 1, this represents application of the bio-control agent on that day by each of the 100 best solutions. Conversely, if the average dosage is 0, this shows that the intervention is not used by any of the 100 best solutions.

Figure 7.4: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 0$ and $R = g$

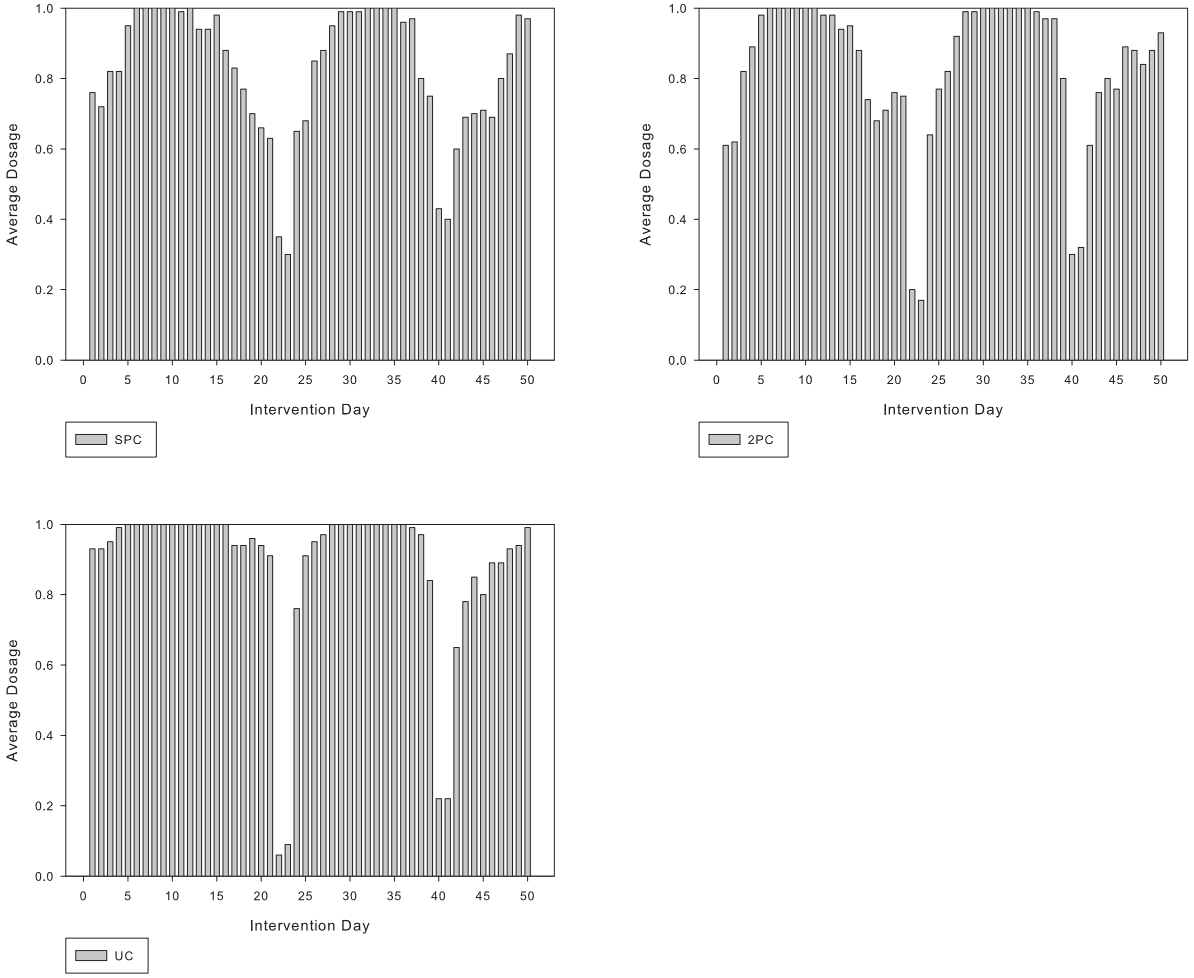
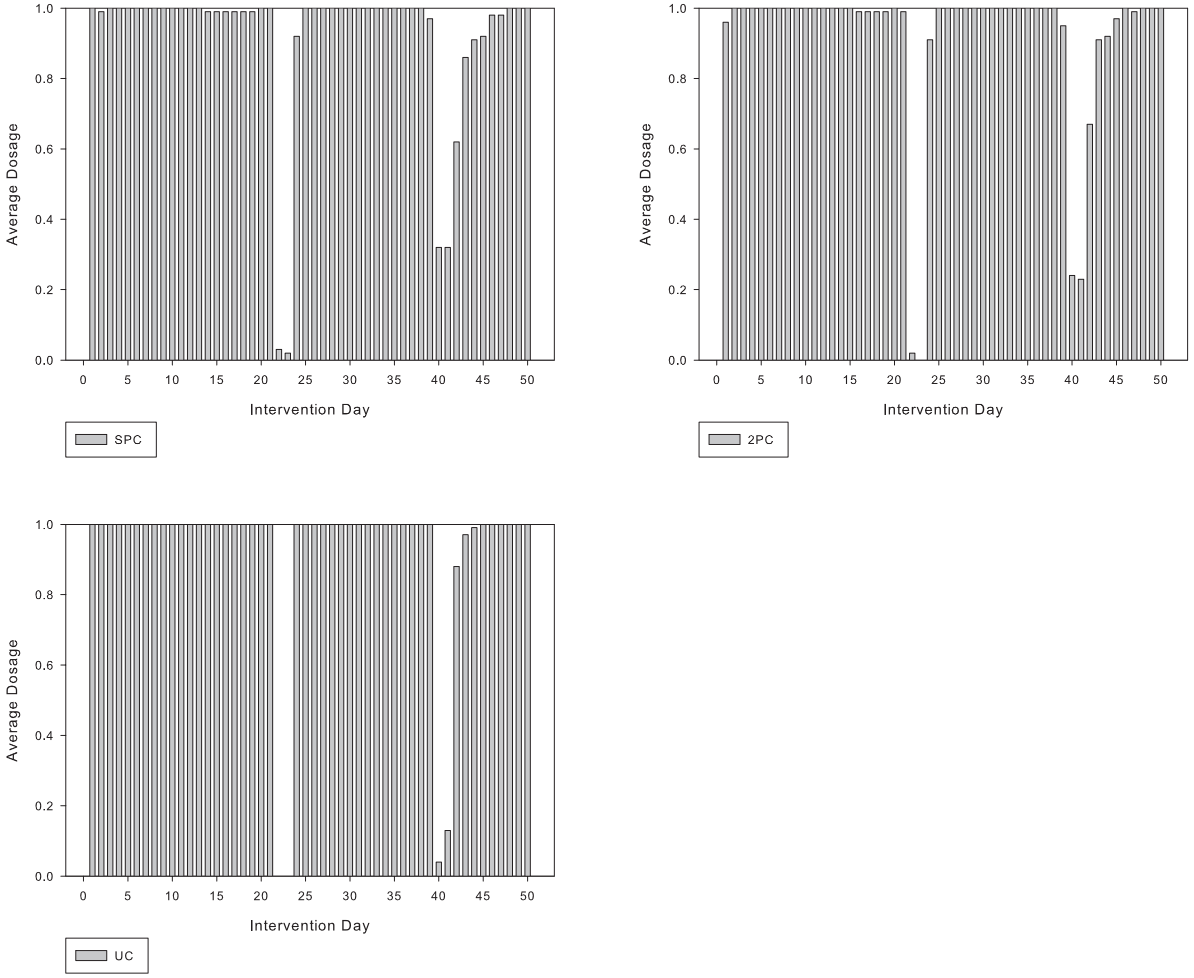


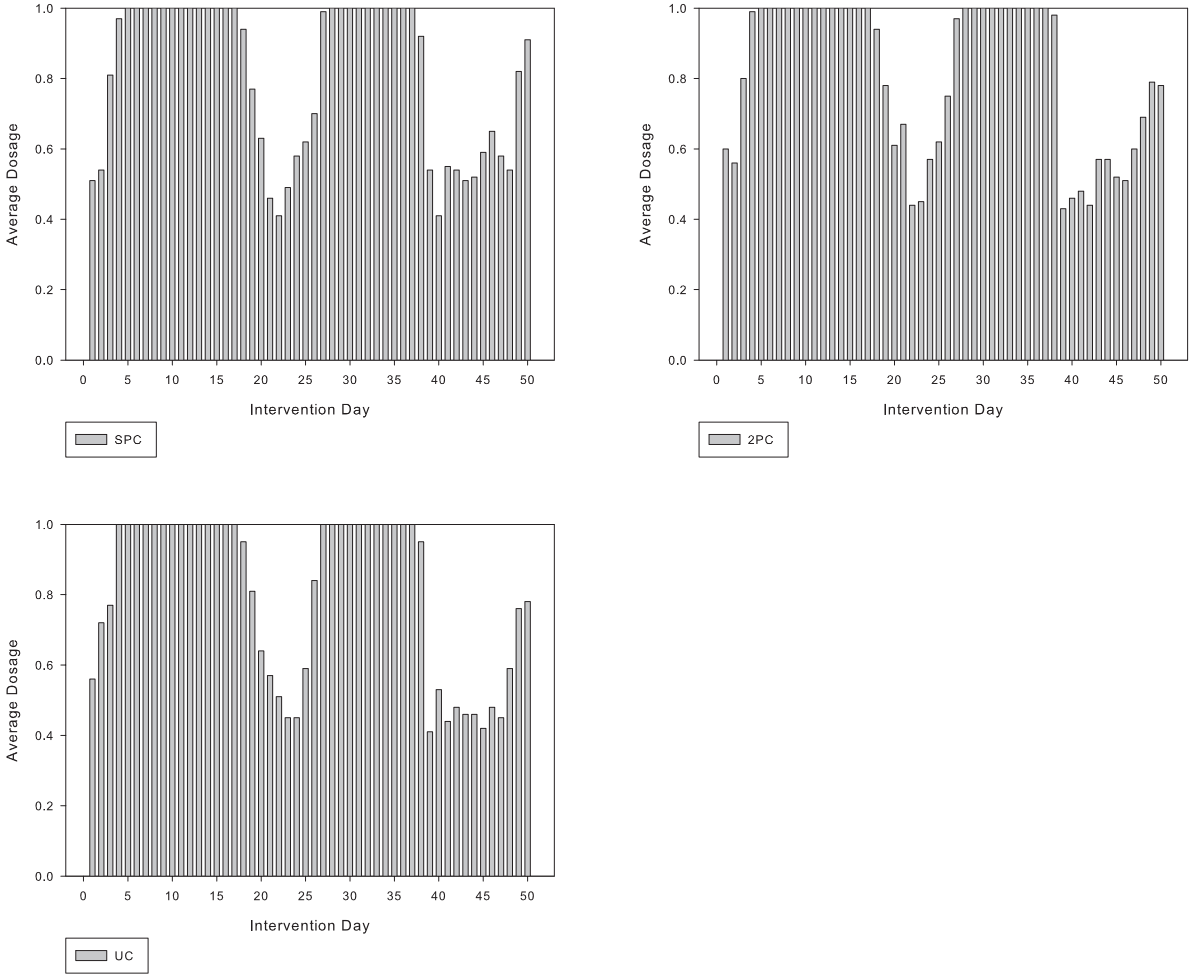
Figure 7.5 shows the average intervention placements for the traditional crossover approaches when p_m is increased to 0.005. As shown in the corresponding fitness results and intervention usage graphs (Figure 7.2), each of the crossover approaches return similar fitness scores for this setting, using a similar number of interventions in schedules. Figure 7.5 shows that each of the crossover approaches do not focus interventions on days 22 - 24 or 40 - 41, but focus largely on the other days. This collective focus in intervention placement explains the similar fitness scores produced for this experiment.

Figure 7.5: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 0$ and $R = g$



The intervention placement when the mutation rate, p_m, is further increased to 0.05 is shown in Figure 7.6. As with a mutation rate of 0.005, each of the crossover approaches return similar fitness scores for this experiment, using the same number of interventions to do so, as shown in Figure 7.3. The intervention placement, as detailed in Figure 7.6, shows that all approaches focus interventions predominantly on days 4 - 18 and 27 - 37. This similar placement of interventions, focussed around the two cycles of larvae development, explains the similar fitness scores returned

Figure 7.6: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 0$ and $R = g$



Steady state replacement

Section 2.1.8 noted that there are two common methods of updating a population, namely generational and steady state replacement. Section 6.8 described how in order to determine the effectiveness of approaches, both of these replacement strategies would need to be considered. As with the generational equivalent described in Section 7.1.1, this experiment details the performance of approaches under steady-state replacement with intermediate parameter settings.

Figures 7.7 to 7.9 show the fitness and intervention usage associated with performing the previous set of tests with a steady state replacement strategy. Figure 7.7 shows that when p_m is 0, each of the crossover approaches are slower to accumulate interventions under steady state replacement when compared to generational replacement. Although each of the crossover approaches utilise marginally fewer interventions than under generational replacement, UC alone returns similar fitness scores under both replacement strategies. Both SPC and 2PC return weaker solutions under steady state replacement than generational, when $p_m = 0$.

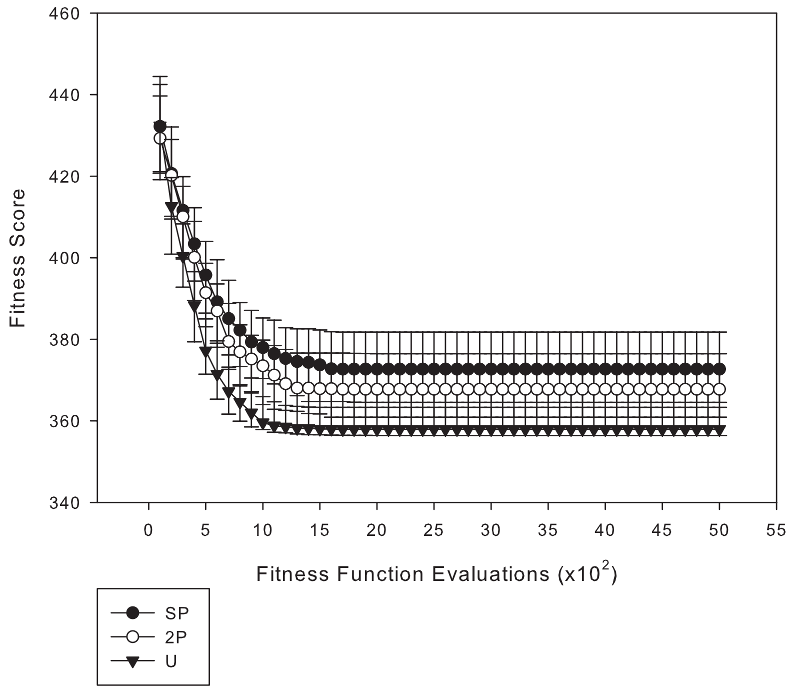
Figure 7.8 shows the fitness score and intervention usage for steady state replacement when $p_m = 0.005$. When compared with Figure 7.2, where the only difference is the replacement strategy, each crossover approach has a similar performance regardless of replacement strategy.

It can be seen that whether under steady state replacement (Figure 7.9) or generational replacement (Figure 7.3), all crossover approaches behave in a similar manner when $p_m = 0.05$.

Through comparing intermediate parameter settings for both steady state and generational replacement for p_m of 0, 0.005 and 0.05, similar behavior is displayed between approaches. When p_m is 0.005 or 0.05, each of the approaches seem unaffected by the choice of replacement strategy.

Under a steady state approach, when $p_m = 0$, both SPC and 2PC converge to less optimal solutions, requiring fewer interventions, than with generational replacement. As no new material is being added into the population through mutation, it would appear that replacing the population as a whole, as with generational replacement provides the variety required by SPC and 2PC, which does not exist when 2 children are introduced to the population each generation. As UC can recombine more genes per crossover process than both SPC and 2PC this would explain its ability to cope with p_m of 0 for steady state or generational replacement in a similar manner.

Figure 7.7: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 0$ and $R = ss$



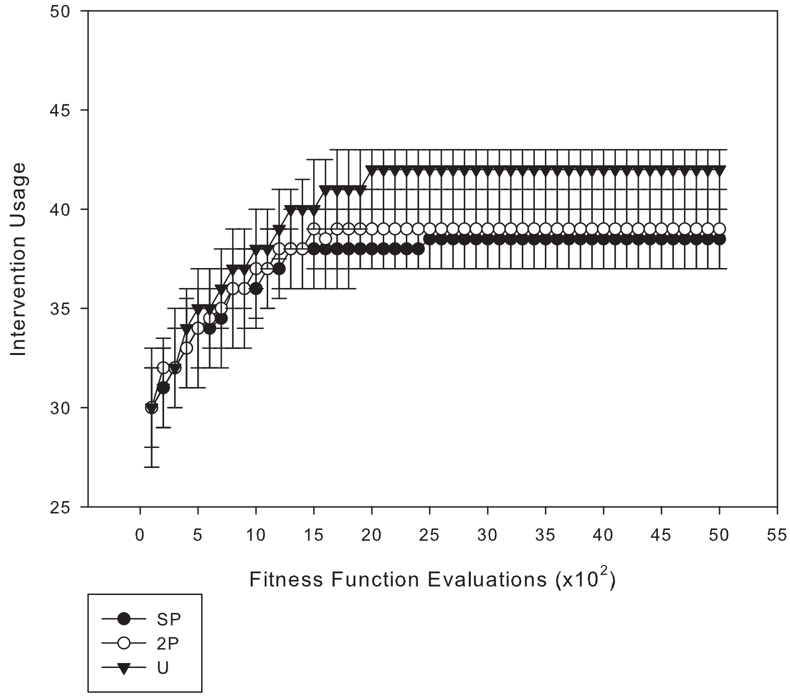
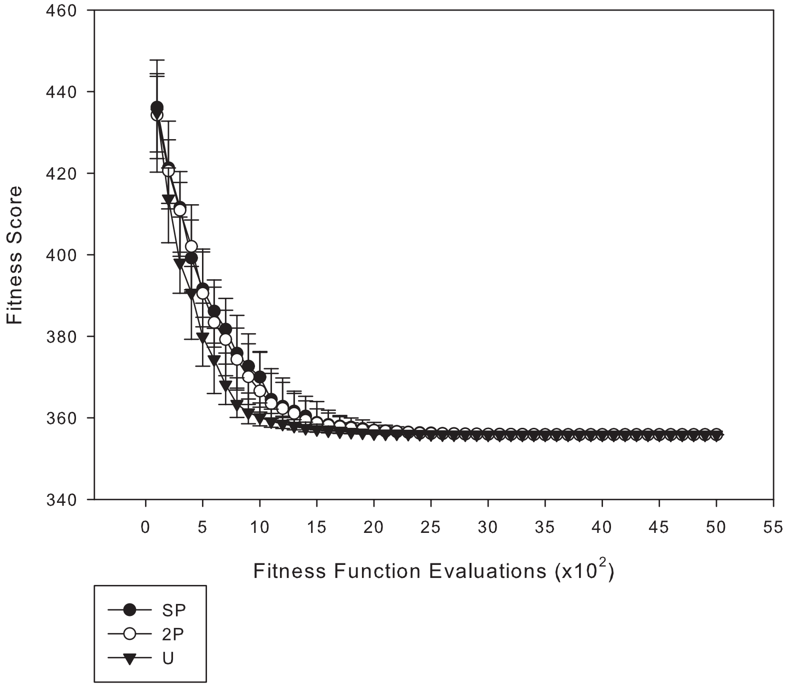


Figure 7.8: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 0$ and $R = ss$



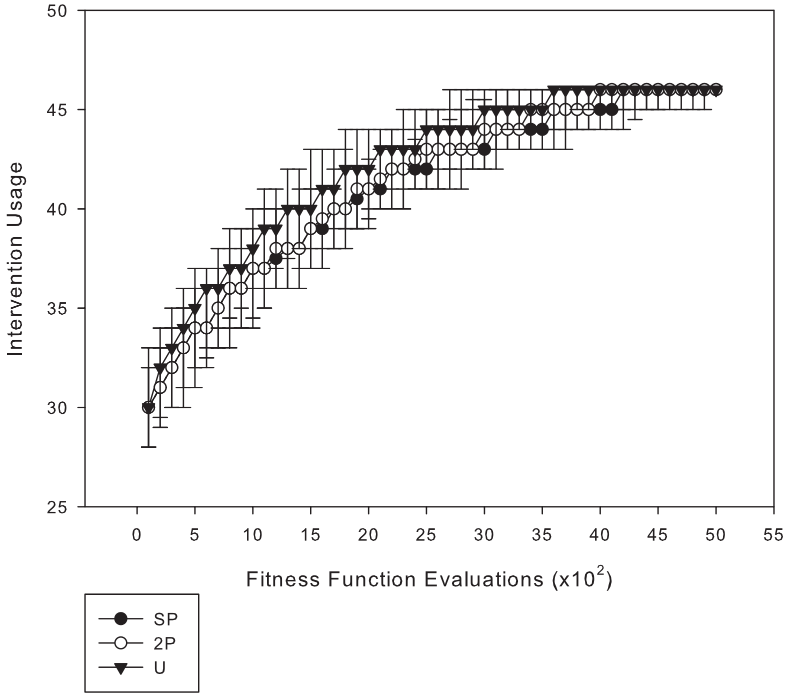
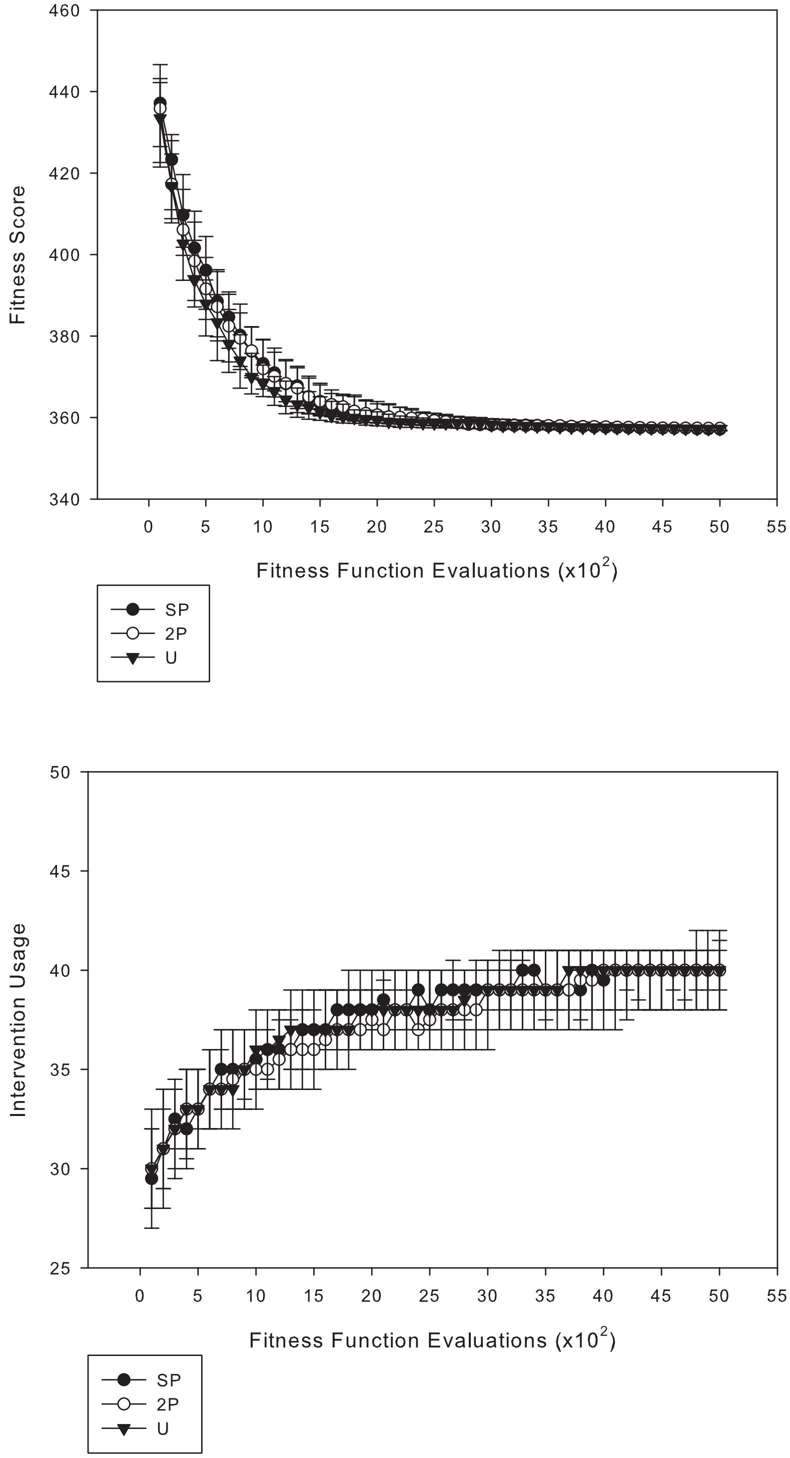


Figure 7.9: Results for $N = 100, p_c = 0.9, p_m = 0.05, P = 0$ and $R = ss$

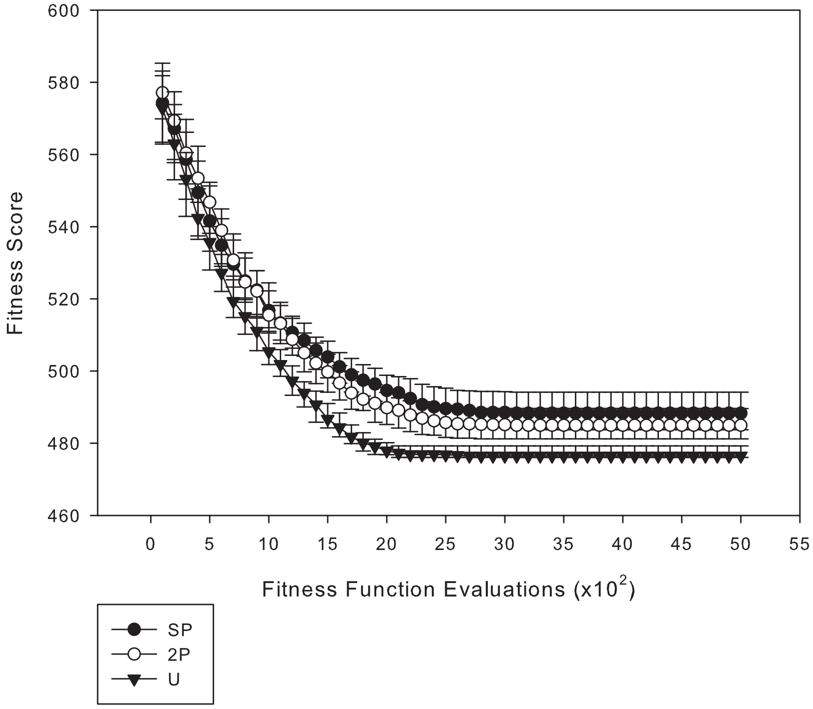


7.1.2 5 penalty points per intervention

As described in Section 5.3.2, intervention penalties are applied for each bio-control application. This section details the effects of increasing the penalty to 5 points per intervention on each of the crossover approaches. As with the experiment described in Section 7.1.1, intermediate parameters of $N = 100$, and $p_c = 0.9$ will be used.

Figure 7.10 shows the fitness scores and associated intervention usage for a mutation rate, p_m, of 0. Figures 7.11 and 7.12 details the fitness and intervention usage for p_m levels of 0.005 and 0.05 respectively.

Figure 7.10: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 5$ and $R = g$



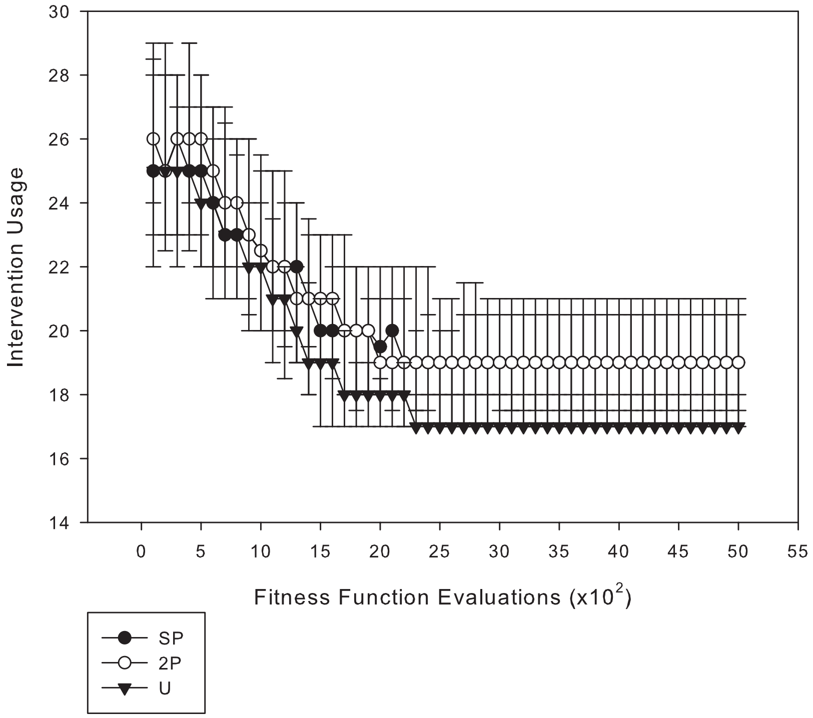


Figure 7.11: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 5$ and $R = g$

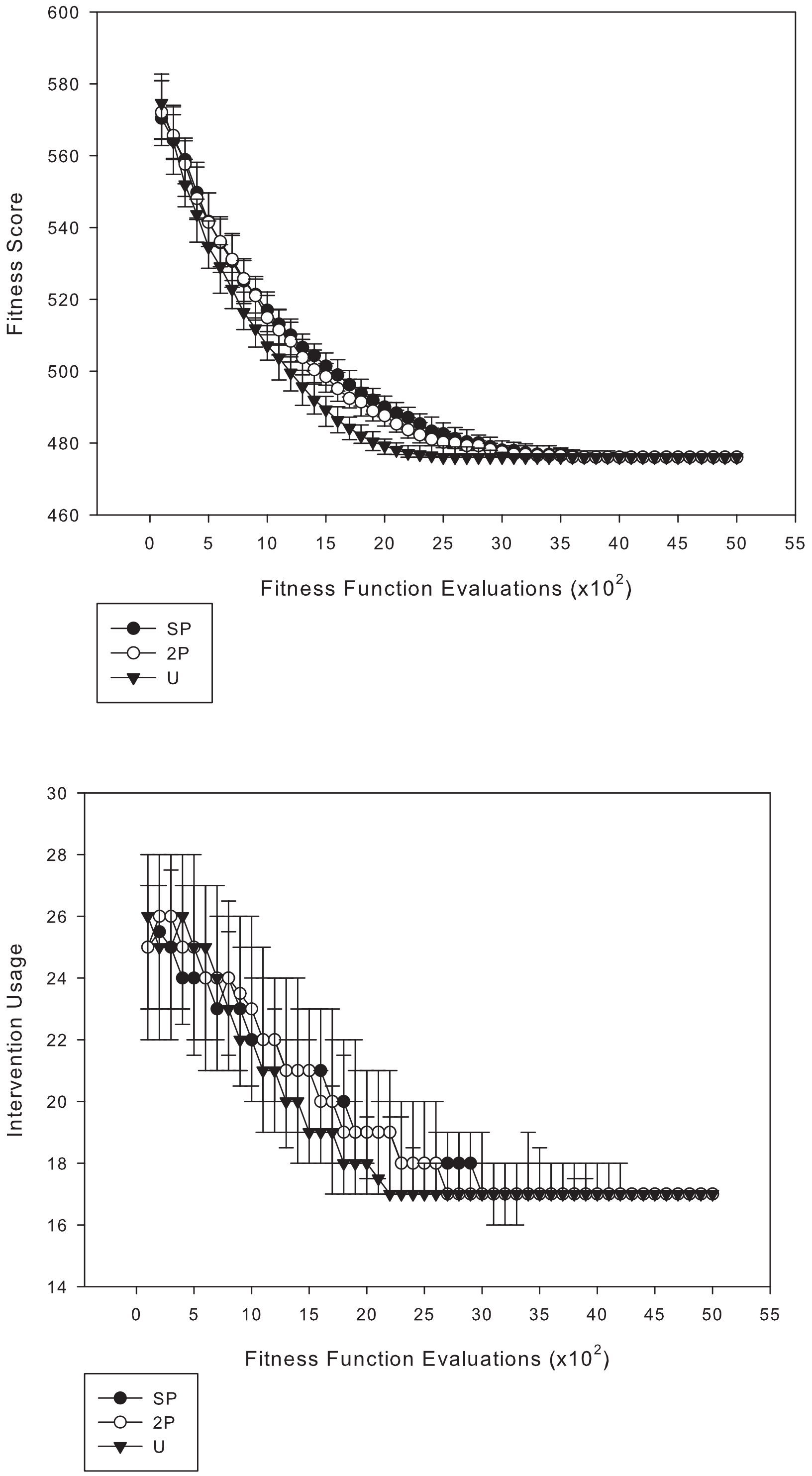


Figure 7.12: Results for $N = 100, p_c = 0.9, p_m = 0.05, P = 5$ and $R = g$

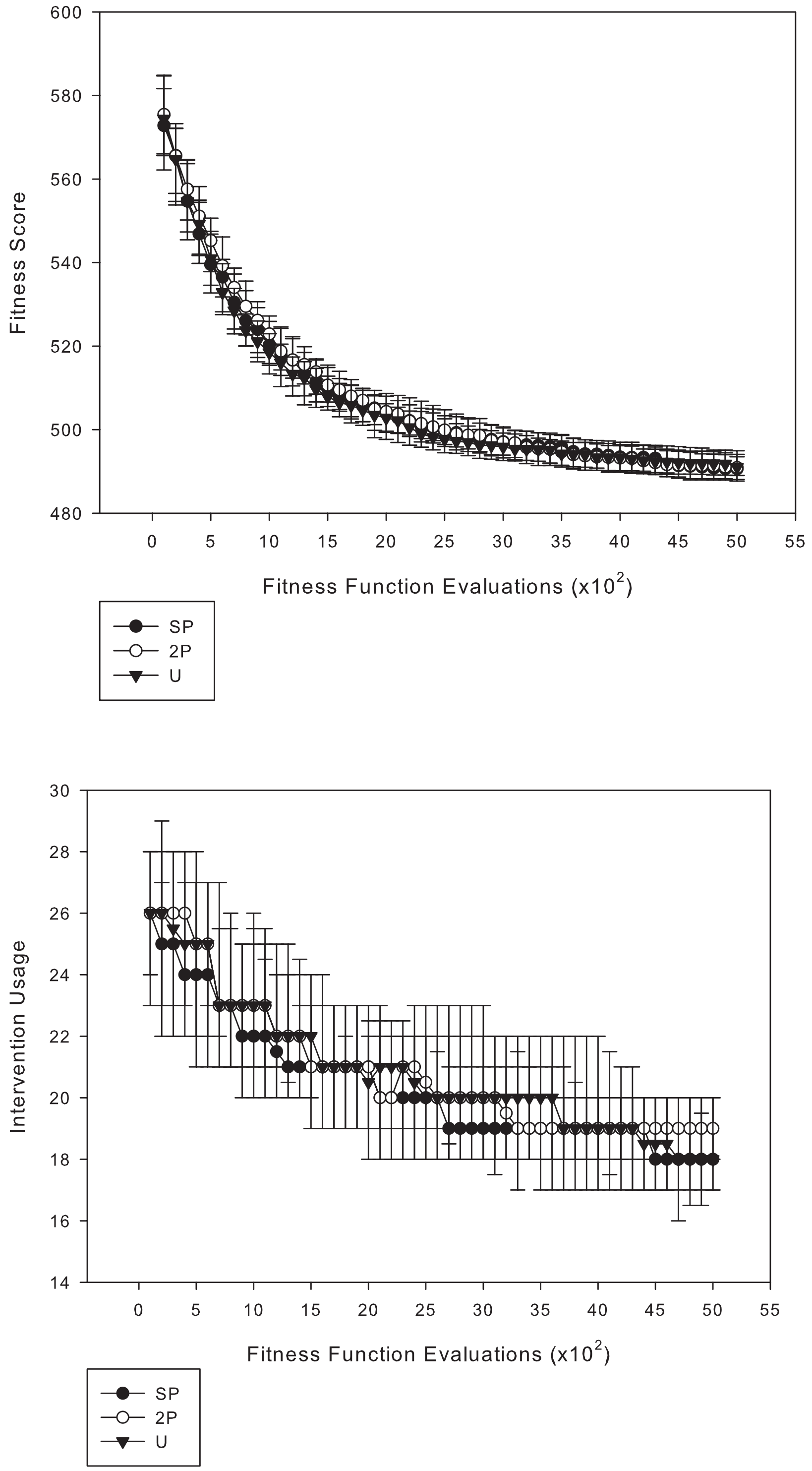


Figure 7.10 shows the fitness scores and intervention usage for the intermediate parameter settings with a penalty of 5 per intervention and p_m of 0. With no intervention penalty, the best solution appears to be to accumulate many interventions in the schedule. With a penalty of 5 per intervention, the best solution when p_m is 0 appears to be to reduce interventions to a schedule containing around 16 - 20 interventions.

UC is quicker to reduce interventions than either SPC or 2PC. After 2,500 FFEs, each of the approaches has settled on a number of interventions,with UC using around 17 interventions and SPC and 2PC requiring about 19. This rapid reduction in intervention levels displayed by UC accounts for the efficiency of its fitness scores for the first 1,500 FFEs. Through using fewer interventions throughout the test, UC settles to solutions with better fitness than either SPC or 2PC.

Figure 7.11 shows the fitness scores and intervention usage when the mutation rate, p_m, is increased to 0.005. As with a p_m of 0, UC is quicker at reducing intervention levels than SPC and 2PC, resulting in fitter scores till around 3,000 FFEs. At this point, all approaches use approximately 17 interventions and return equivalent scores.

Figure 7.12 shows the effect of increasing p_m to 0.05. In this situation, each of the approaches utilise approximately the same numbers of interventions, which accounts for the similar fitness scores displayed.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the traditional crossover approaches, under a generational replacement strategy for a penalty of 5 points per intervention. With no penalty per intervention, the best scoring schedules used a large number of interventions. With a penalty of 5 points per intervention, the crossover approaches use less interventions. Figure 7.10 detailed the fitness and intervention usage in the absence of mutation. Figure 7.13 shows the average placement of these interventions.

When p_m was 0, for a penalty of 5 points per intervention, UC was shown to return fitter scores than both SPC and 2PC, with 2PC in turn, returning fitter scores than SPC. Figure 7.13 shows the average intervention placement for each of the approaches after 5,000 FFEs. This shows that while all approaches focus interventions on the two main cycles of sciarid larvae growth, UC does not apply interventions as regularly at other points. As there is a cost associated with interventions, this accounts for the fitter scores returned by UC for this parameter setting. 2PC was fitter than SPC for this experiment and, as shown in Figure 7.13, although intervention placement is not as focussed as UC, 2PC uses less interventions than SPC to return fitter scores.

Figure 7.13: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 5$ and $R = g$

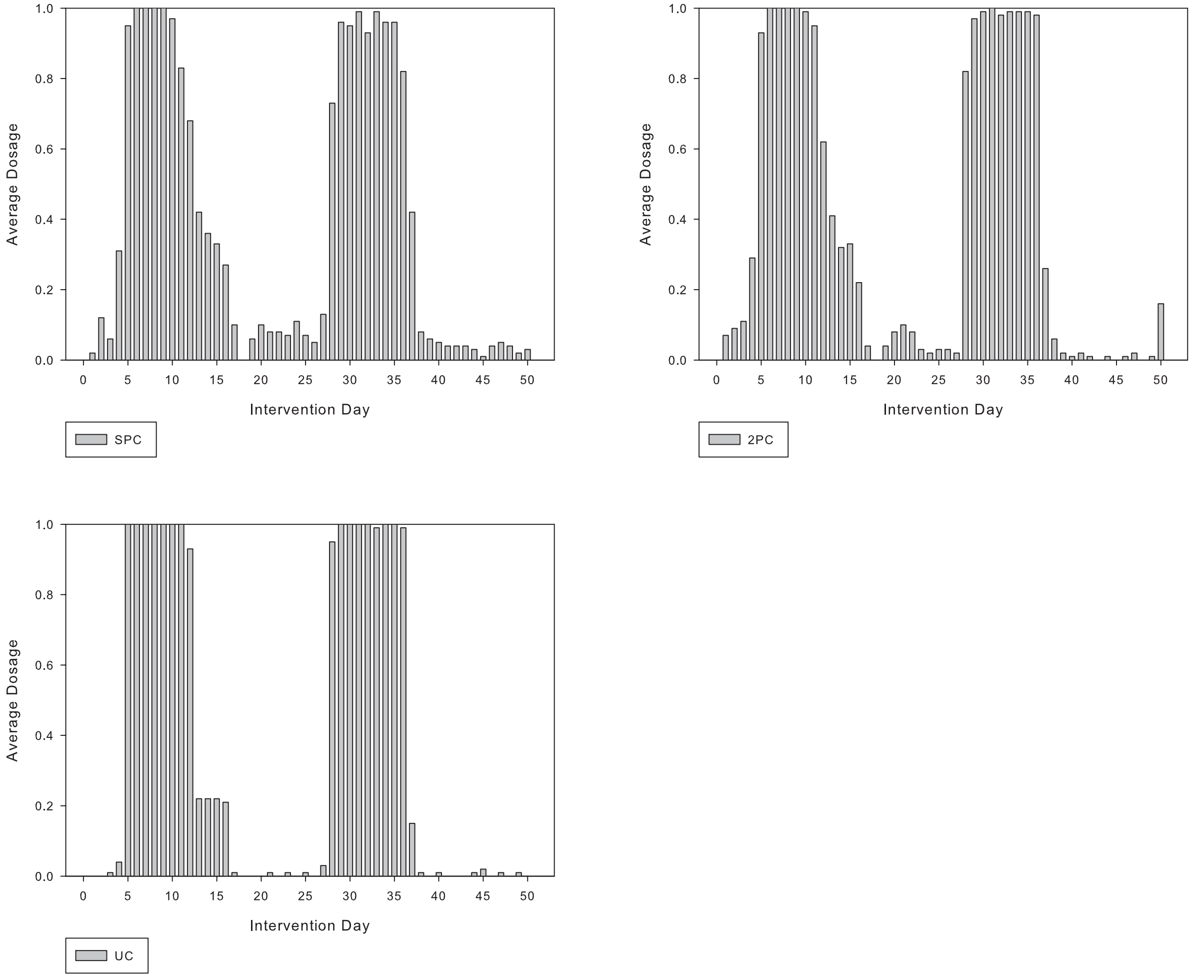


Figure 7.14 shows the average intervention placement when p_m is increased to 0.005. For this parameter setting, each of the crossover approaches returned scores of a similar fitness, using the same number of interventions to do so, as shown in Figure 7.11. As each of the crossover approaches are focussing their interventions in days 4 - 13 and 28 - 36, with limited application on days 13 - 16, this explains the similar fitness scores returned.

Figure 7.14: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 5$ and $R = g$

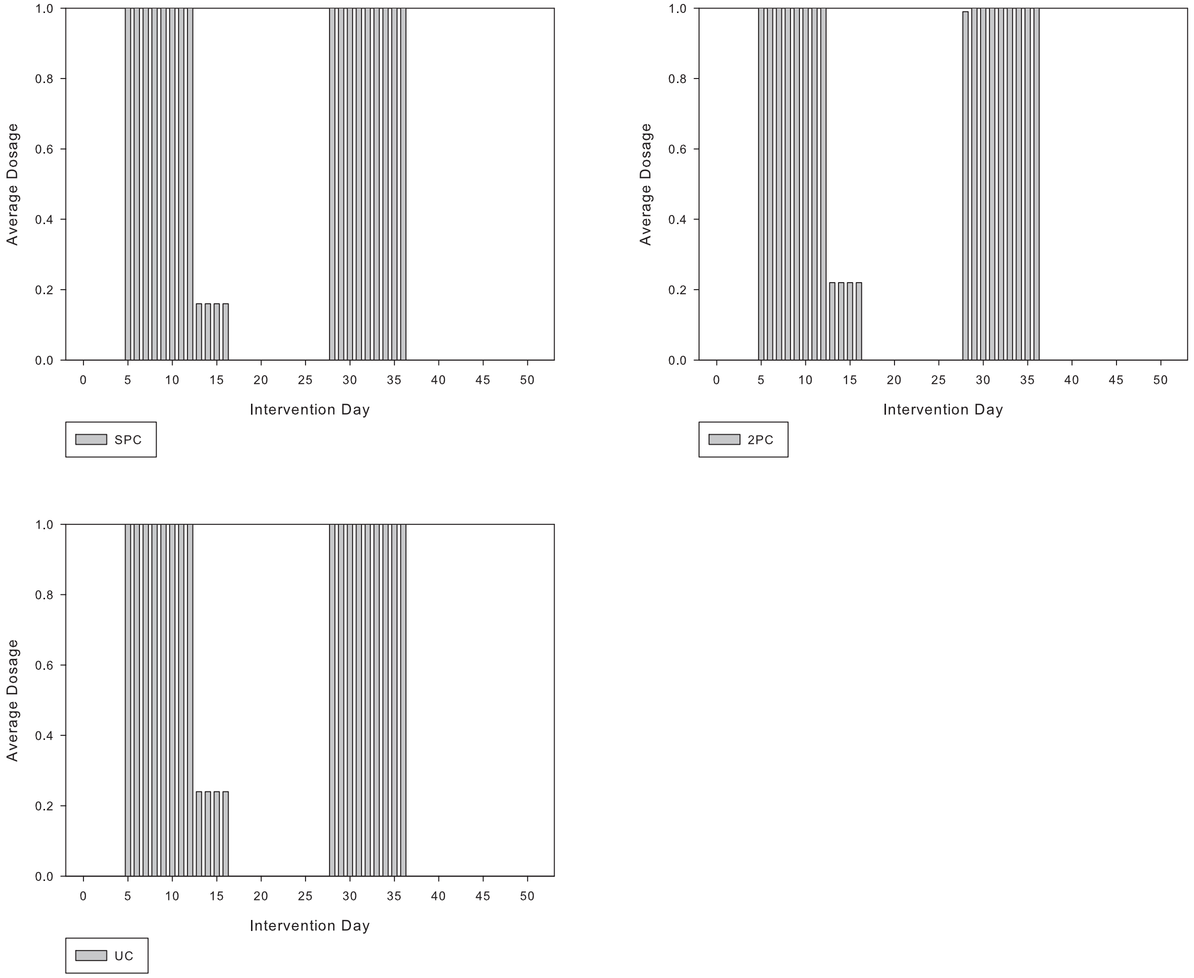
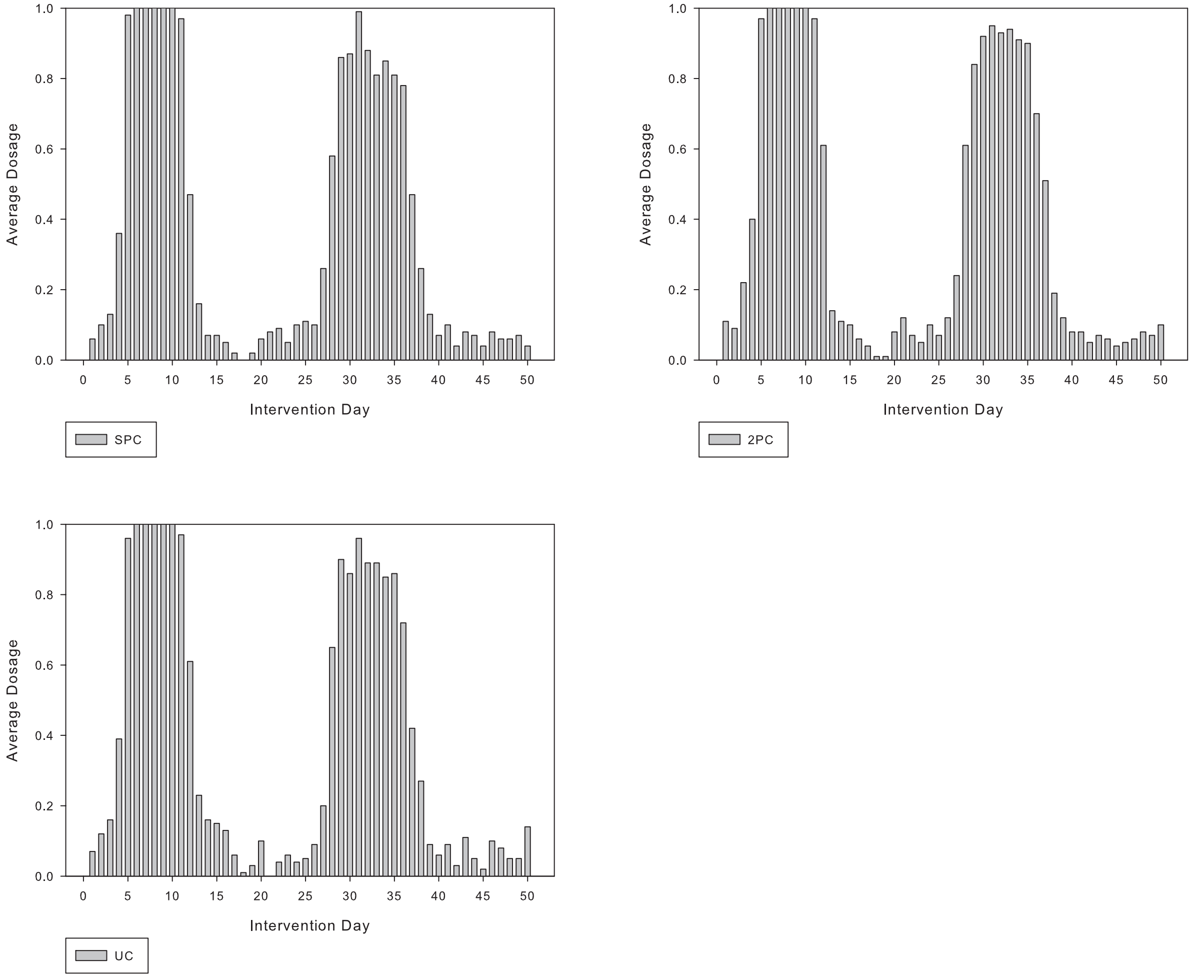


Figure 7.14 shows the average intervention placement when p_m is further increased to 0.05. Each of the crossover approaches use similar numbers of interventions to produce comparable fitness scores, as shown in Figure 7.11. Each of the crossover approaches use similar intervention timings to produce these highly similar fitness scores. As the mutation rate increases from 0.005 to 0.05, more randomness is being added to the population. This randomness is clearly shown through comparing Figures 7.14 and 7.15, in that the placement of interventions is not as focussed for the higher mutation rate as for a p_m of 0.005.

Figure 7.15: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 5$ and $R = g$



Steady state replacement

We now consider the alternative steady state replacement strategy. Figures 7.16 to 7.18 show the fitness scores for SPC, 2PC and UC when $R = ss$ with p_m values of 0, 0.005 and 0.05. There is a penalty of 5 points associated with each intervention.

Figure 7.16 shows that when $p_m = 0$, SPC and 2PC perform worse under steady state than with generational replacement. This was also the case for an intervention penalty of 0. UC is also affected by the change in replacement strategy and returns solutions of less fitness for steady state than for generational replacement. Inspecting the intervention usage, each of the approaches struggle to reduce the level of interventions for steady state replacement compared with generational replacement. This would account for the difference in fitness scores between replacement strategies.

Although each of the approaches are not as efficient under steady state replacement as opposed to generational, UC again returns fitter scores compared with the other approaches for this p_m level.

The next cross comparison required is to check for an increase in the level of mutation. Figure 7.17 details the approaches under steady state replacement when p_m is increased to 0.005. As with the generational case, SPC and 2PC improve dramatically when p_m is increased from 0 to 0.005. There is more overlap between intervention usage for steady state replacement across crossover approaches than was present under a generational replacement strategy. For this reason, the solutions returned are closer together than observed under generational replacement, especially between 1,500 and 2,200 FFEs.

When p_m is further increased to 0.05, as shown in Figure 7.18, each of the approaches utilise approximately the same number of interventions throughout the experiment. This accounts for the very similar scores returned by each of the crossover approaches. These scores are similar to those displayed under generational replacement.

It is interesting to note that when the mutation rate, p_m, is increased from 0.005 to 0.05, regardless of replacement strategy, there is a decline in the fitness scores returned by each of the crossover approaches. This indicates an upper bound for the mutation rate for this problem.

Figure 7.16: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 5$ and $R = ss$

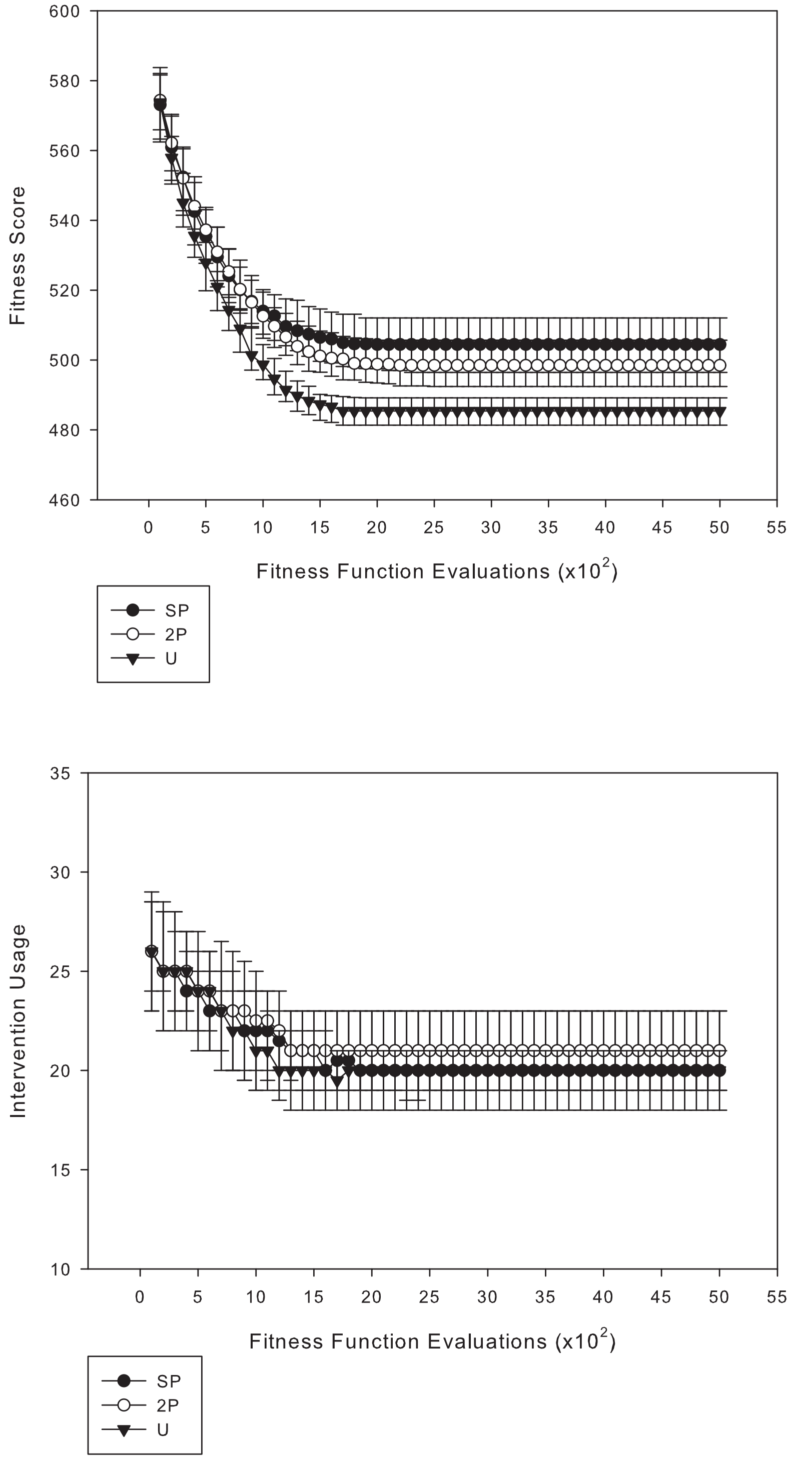
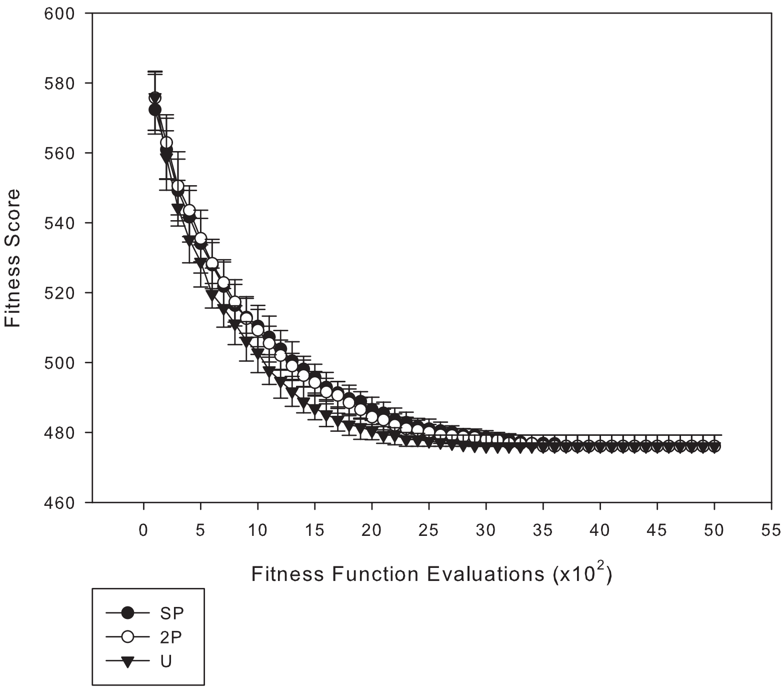


Figure 7.17: Results for $N = 100, p_c = 0.9, p_m = 0.005, P = 5$ and $R = ss$



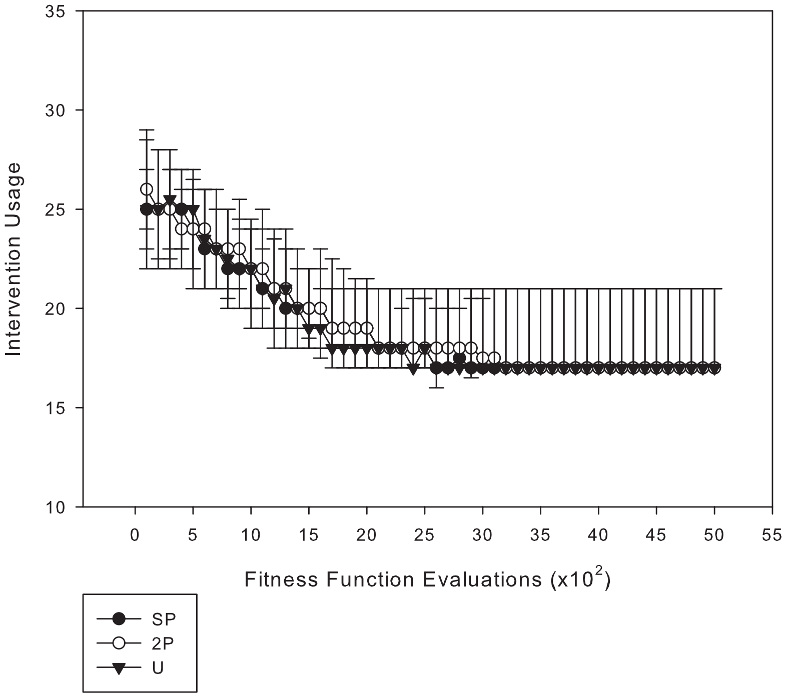
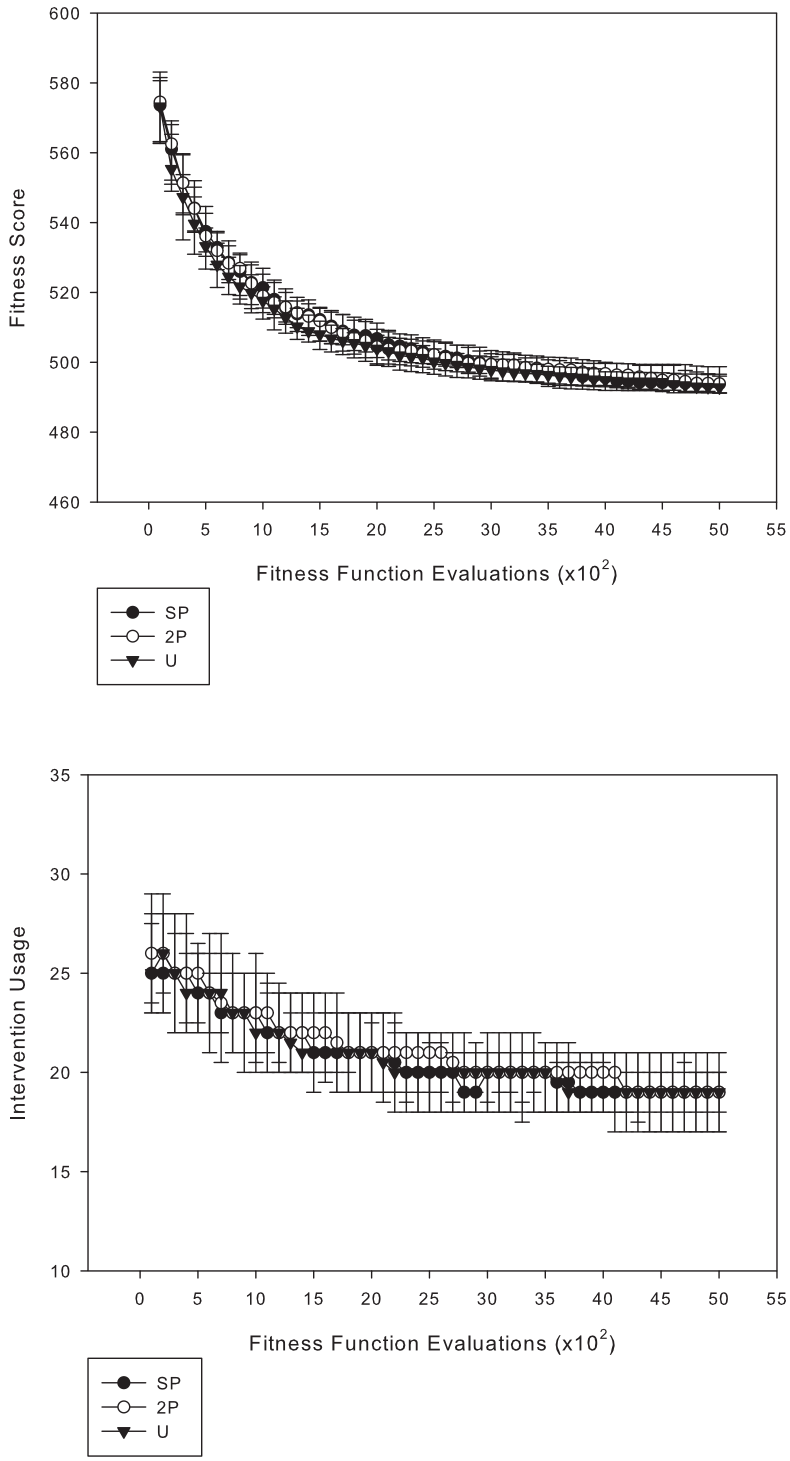


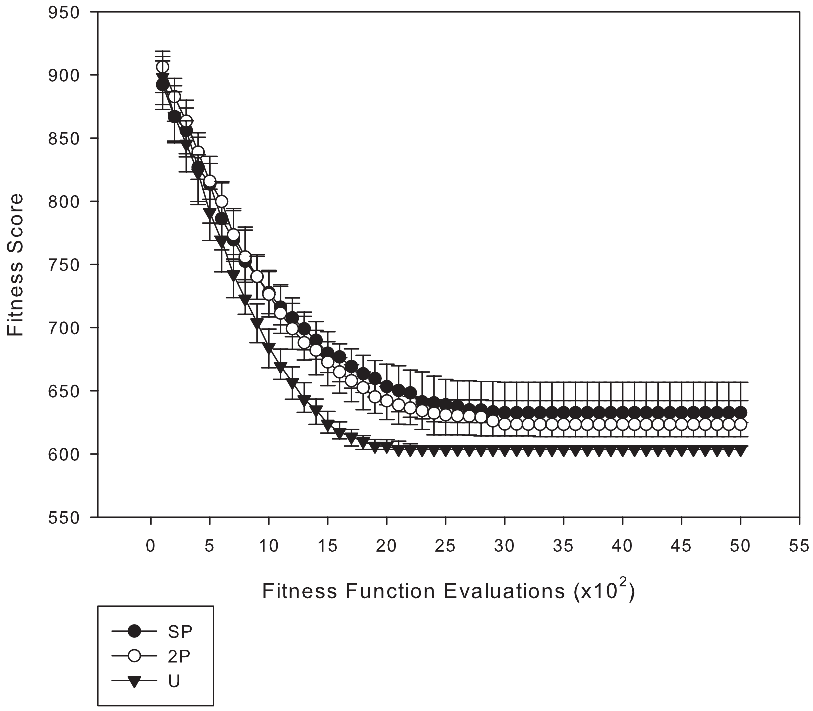
Figure 7.18: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 5$ and $R = ss$



7.1.3 20 penalty points per intervention

The next experiment considers the effect of increasing the number of penalty points per intervention from 5 to 20. Figures 7.19 to 7.21 show the fitness scores and associated intervention usage for a penalty of 20 per intervention and p_m values of 0, 0.005 and 0.05 respectively.

Figure 7.19: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 20$ and $R = g$



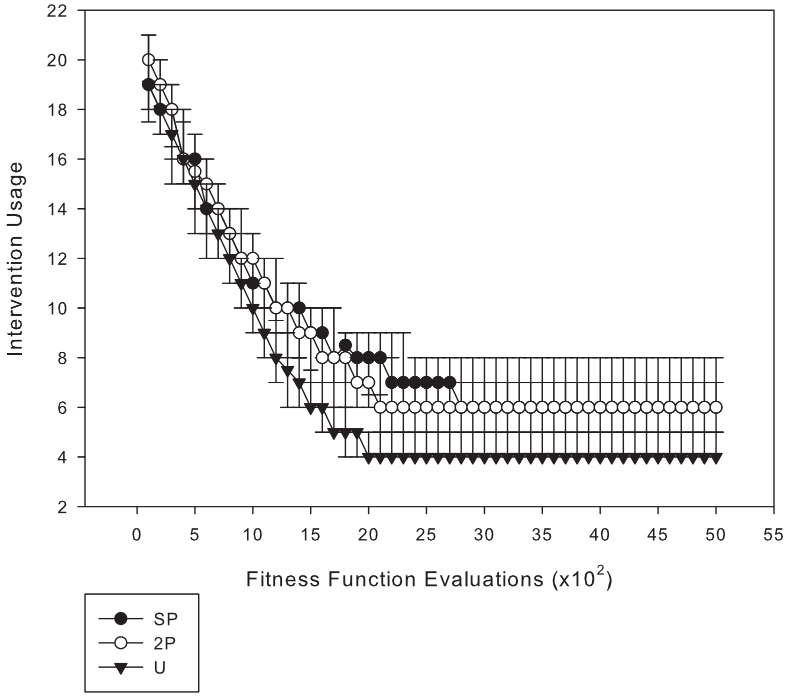
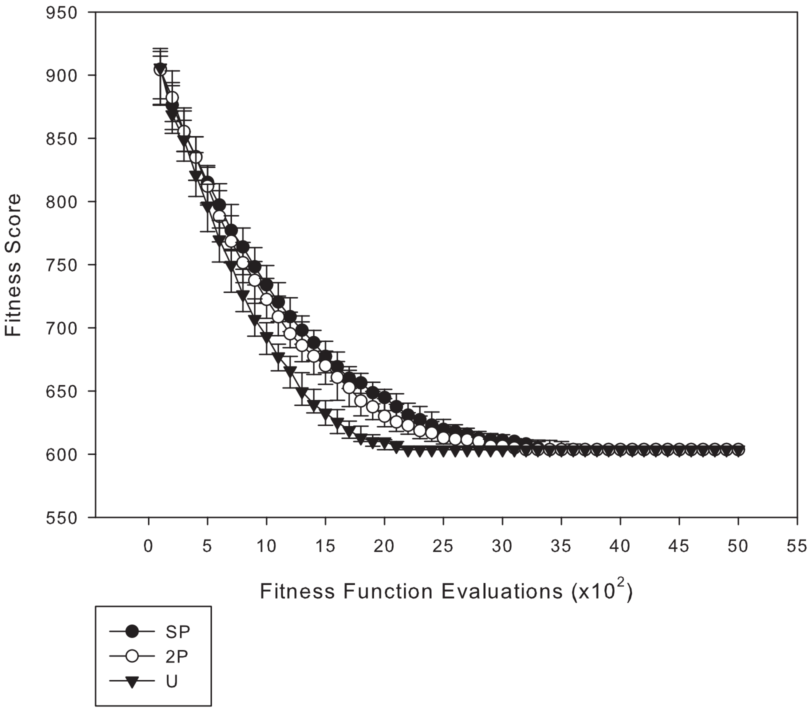


Figure 7.20: Results for $N = 100, p_c = 0.9, p_m = 0.005, P = 20$ and $R = g$



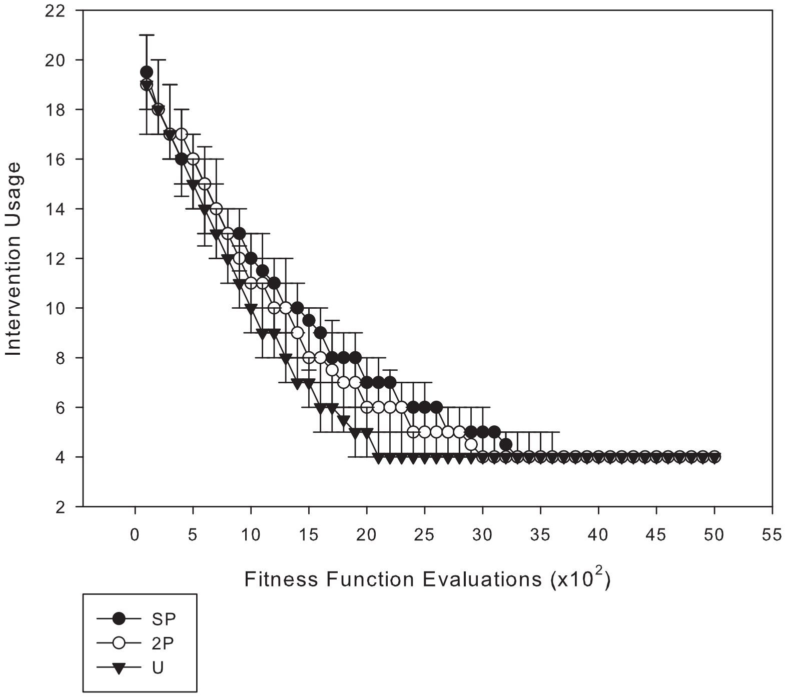


Figure 7.21: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 20$ and $R = g$

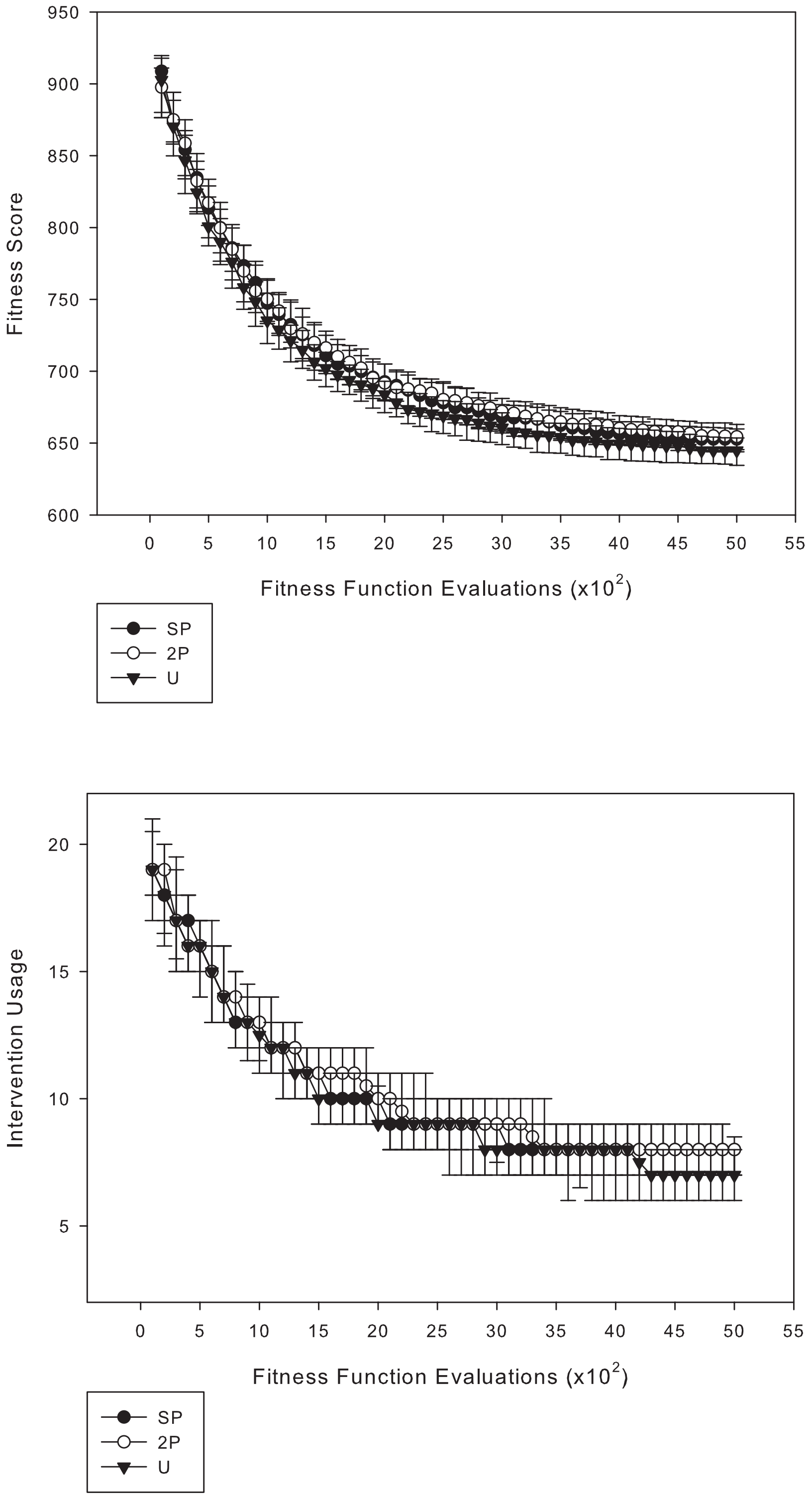


Figure 7.19 shows the fitness scores and intervention usage when p_m is 0 and the penalty per intervention is 20. As with lesser intervention penalty values, UC is quicker at reducing the number of interventions, and returns fitter scores. Both SPC and 2PC reduce interventions over the duration of the experiment, but at a slower pace

than UC. As both SPC and 2PC settle on more interventions than UC this results in convergence to less optimal scores.

Figure 7.20 details the experiment when the probability of mutation, p_m, is increased to 0.005. As with a p_m of 0, UC is quicker to reduce the number of interventions than both SPC and 2PC, resulting in fitter scores for the first 3,000 FFEs. At this point, all crossover approaches tend to the same number of intervention placements and all approaches return similar scores for the rest of the experiment. It would appear that for this level of penalty, placement of 4 interventions during the schedule returns the optimal score.

As p_m is increased to 0.05, as shown in Figure 7.21, each of the approaches use approximately the same number of interventions throughout the experiment. Due to the similarity between the number of interventions in schedules, each of the crossover approaches return similar scores. The scores returned by each of the crossover approaches are less fit with this mutation level compared to those returned when $p_m = 0.005$. As stated in Section 7.1.2, this shows an upper bound for the mutation level in this experiment.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the traditional crossover approaches, under a generational replacement strategy for a penalty of 20 points per intervention. Figure 7.19 detailed the fitness and intervention usage in the absence of mutation. Figure 7.22 shows the average placement of these interventions.

As with the lesser intervention penalties, in the absence of mutation UC returns fitter scores than the other approaches, with 2PC returning fitter scores than SPC. Figure 7.22 shows the placement of interventions for each of the approaches. UC uses fewer interventions than the other approaches, mainly focussing interventions on days 6 - 9. 2PC is not as focussed as UC in the timing of interventions for this experiment and as there is now a cost of 20 points per intervention, this results in less fit scores being returned. SPC focusses applications mainly at the first sciarid larvae cycle, but also shows occasional application to the second cycle. It would appear that the benefit associated with this latter interventions are not enough to outweigh the cost of interventions, hence the lesser fitness scores for this experiment.

Figure 7.22: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 20$ and $R = g$

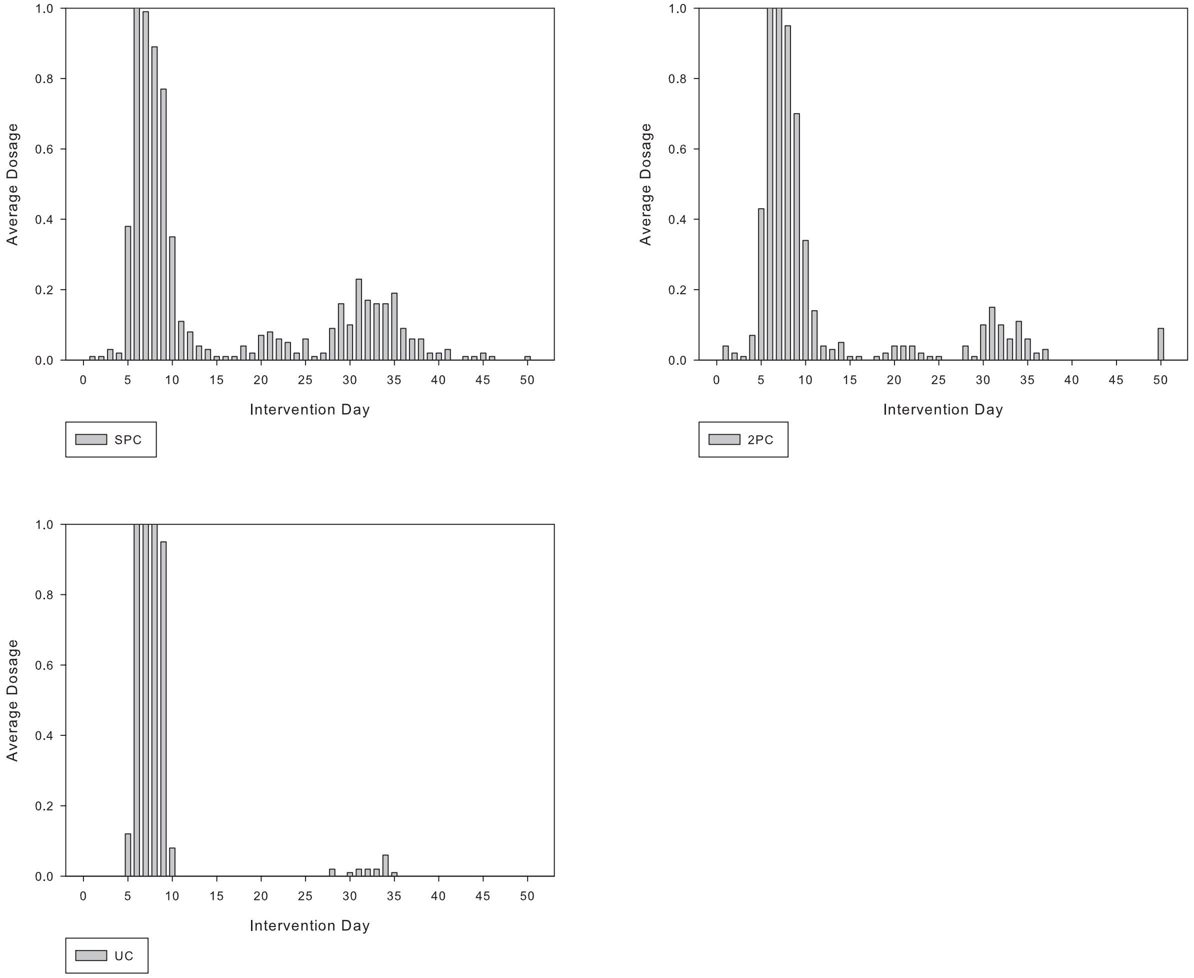


Figure 7.23 shows the intervention placement for the fitness scores and intervention usage shown in Figure 7.20. This shows that for a mutation rate of 0.005, each of the crossover approaches focus on placing interventions on days 6 - 9. This accounts for the similar fitness scores returned by each of the approaches.

Figure 7.23: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 20$ and $R = g$

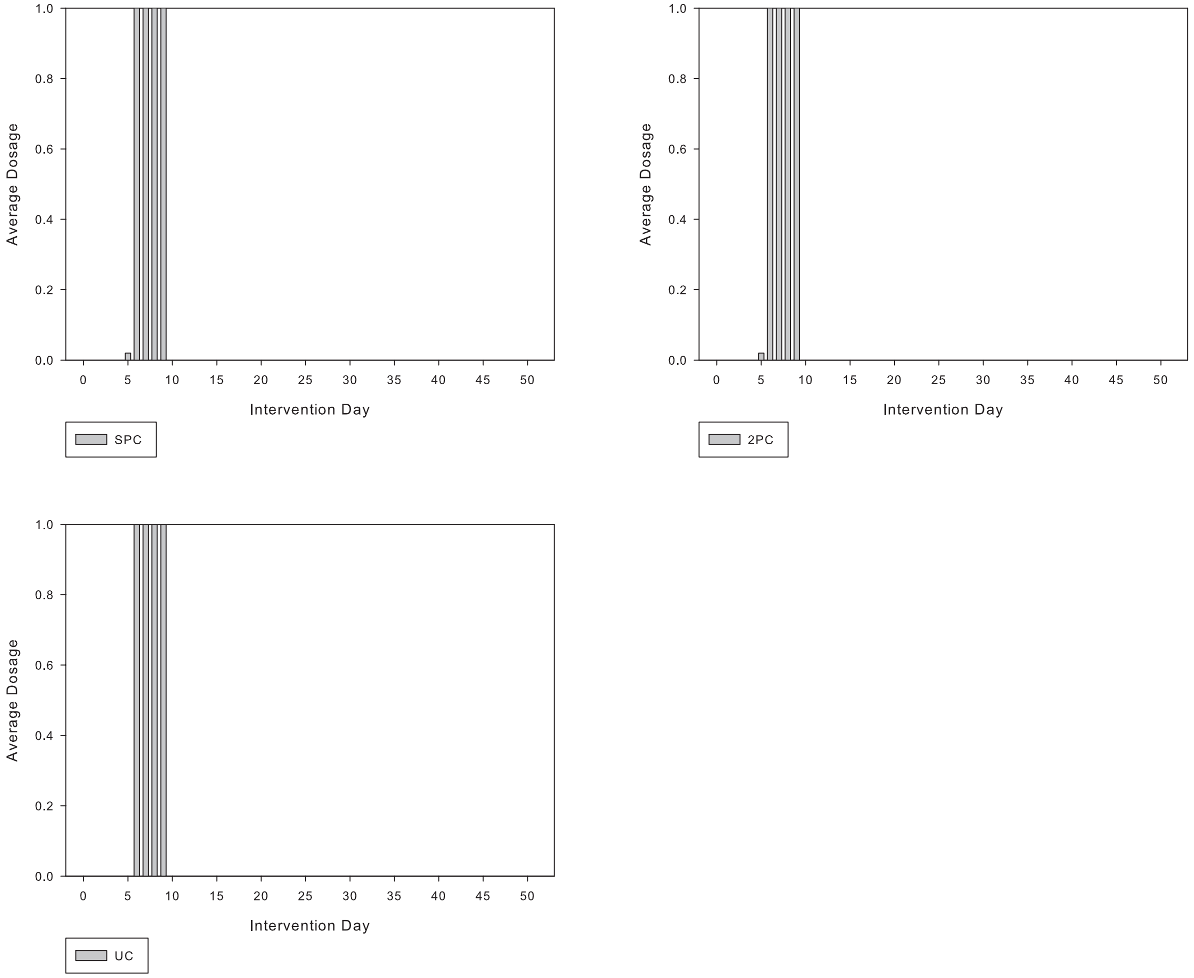
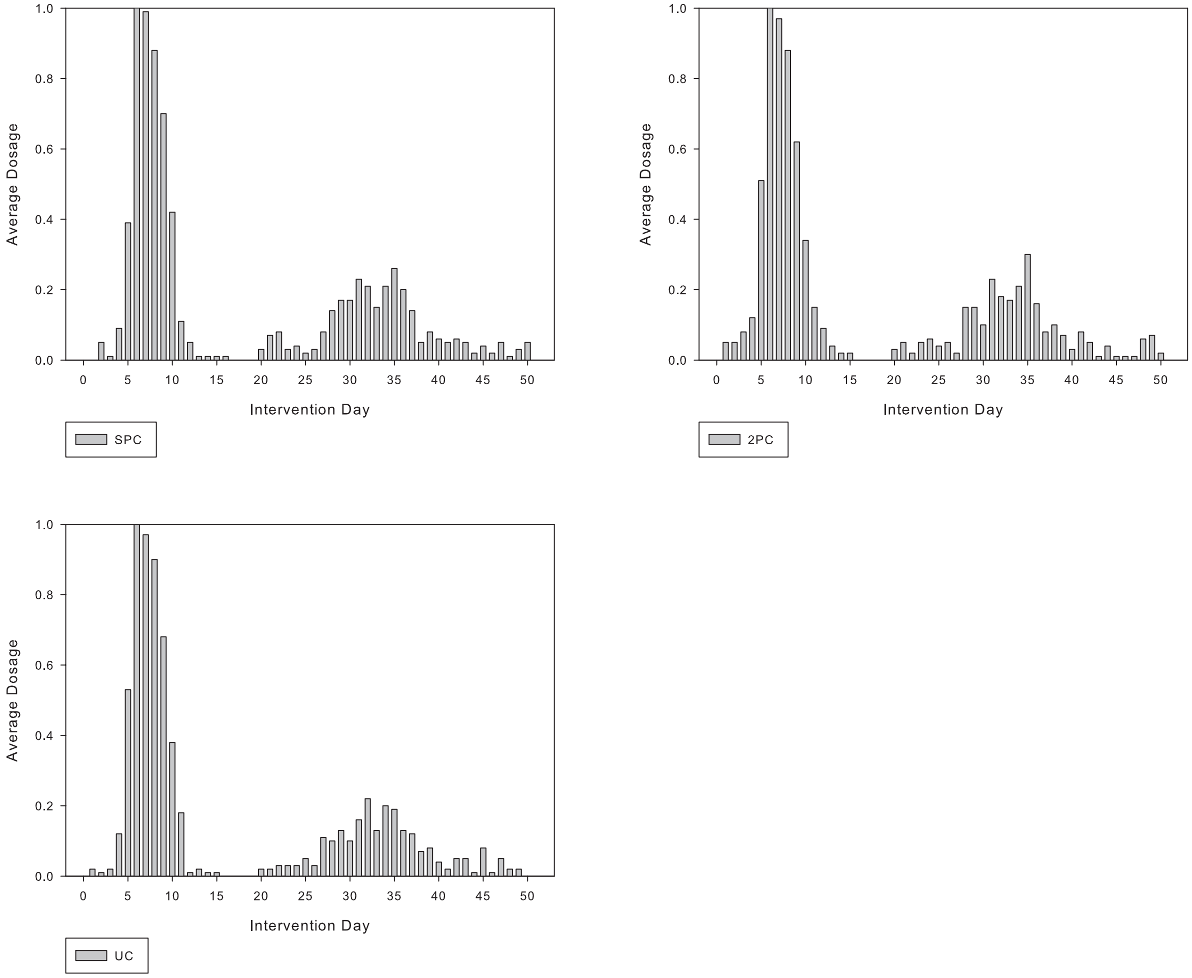


Figure 7.21 showed that each of the crossover approaches use a similar number of interventions to produce fitness scores in the same range. Figure 7.24 shows the placement of these interventions. All approaches focus mainly on the first sciarid larvae cycle, with a limited number of applications across approaches to target the second larvae cycle. This similarity in intervention placement explains the close relation in the fitness scores returned by the crossover approaches. This focus on intervening early in the sciarid larvae cycle, targeting the first larvae cycle, was shown to be the most effective approach in the work of Fenton et al [93]. Related studies by Brownlee et at [110], also found this to be the case. In that work, Brownlee et al concluded that the destruction of the larvae population early in the schedule means few larvae remaining to breed and repopulate. As with the intervention penalty of 5 points per intervention, the higher mutation rate is shown to produce less focussed schedules than when p_m is 0.005, resulting in scores of a lesser fitness.

Figure 7.24: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 20$ and $R = g$



Steady state replacement

We now need to check if the previous observations concerning generational replacement still hold for a steady state replacement strategy. Figures 7.25 to 7.27 show the fitness scores and intervention usage for the same previous intermediate parameter settings and a penalty of 20 per intervention when $R = ss$.

Fitness scores and intervention usage for steady state replacement when $p_m = 0$ are shown in 7.25. In the absence of mutation, as with lower intervention penalties, each of the crossover approaches struggle to find solutions of a similar fitness when $R = ss$ as for the case when $R = g$.

Under generational replacement, when $p_m = 0$, SPC and 2PC settled to schedules requiring placement of 6 interventions, and UC required 4. With steady state replacement and no mutation, SPC settles on 9 interventions, 2PC uses 8 and UC requires 6. This shows that each of the approaches experience difficulties in the production of schedules requiring less interventions under a steady state replacement strategy. One explanation for this is that in the absence of mutation, each of the approaches uses the diversity in the population to help drive the intervention reduction process. As less diversity is introduced with 2 children per generation as opposed to complete replacement of the population, this would account for the difficulties in intervention reduction when $p_m = 0$.

Figure 7.26 shows the fitness scores and intervention usage when p_m is increased to 0.005. As with generational replacement, UC is quicker to reduce the number of interventions in the schedule, resulting in fitter scores for the first 3,000 FFEs. At this point, again in line with generational replacement, each of the approaches produce schedules requiring placement of 4 interventions and return solutions of similar fitness.

The fitness score and intervention usage for the case when p_m is increased to 0.05 is shown in Figure 7.27. In line with experiments using a similar p_m level and a lesser intervention penalty, when the penalty is 20 per intervention, each of the crossover approaches produce similar scores. This is accounted for by the reduction in intervention levels at a similar pace by each of the crossover approaches. At this mutation level, similar scores are produced by each of the crossover approaches, regardless of replacement strategy used.

Figure 7.25: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 20$ and $R = ss$

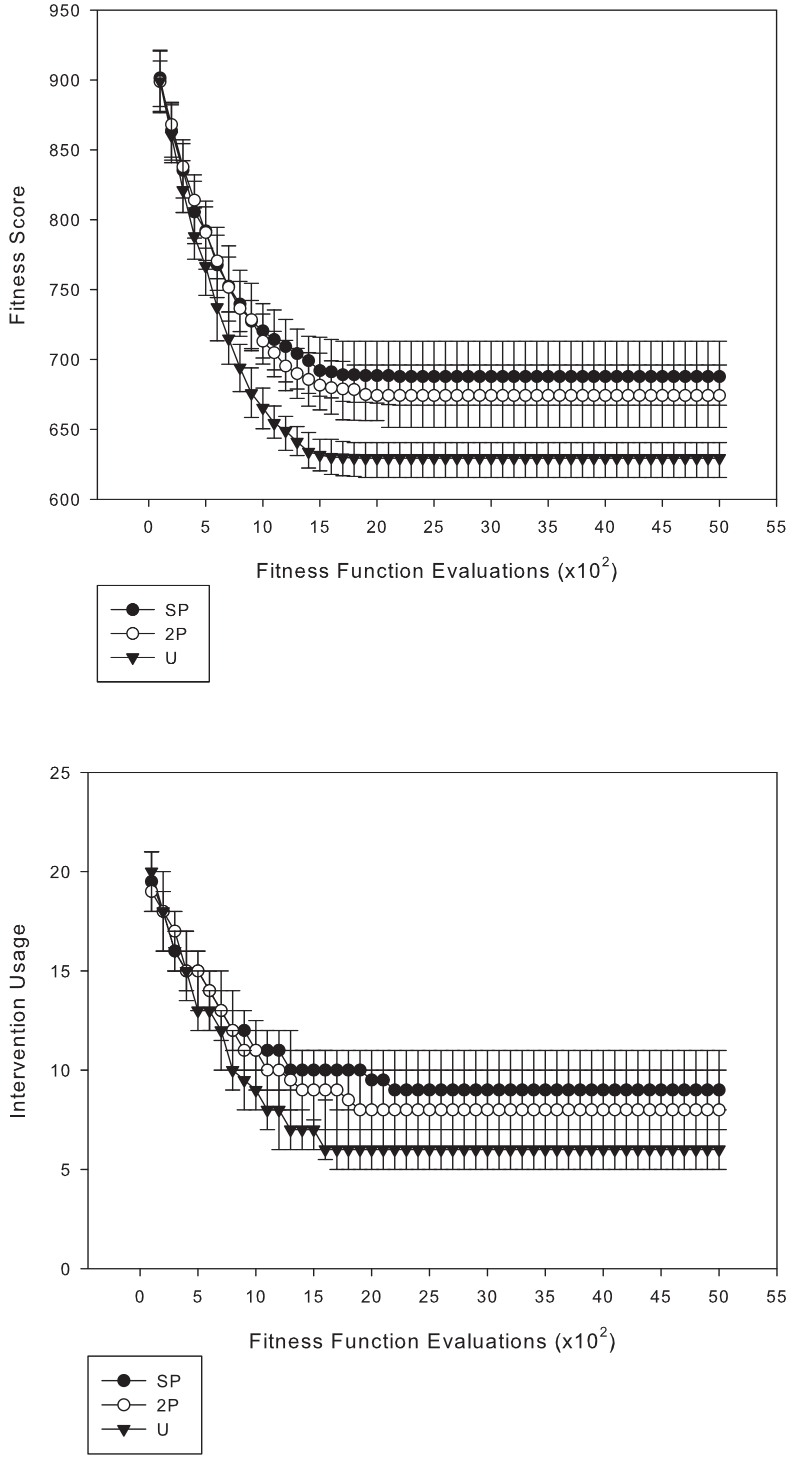
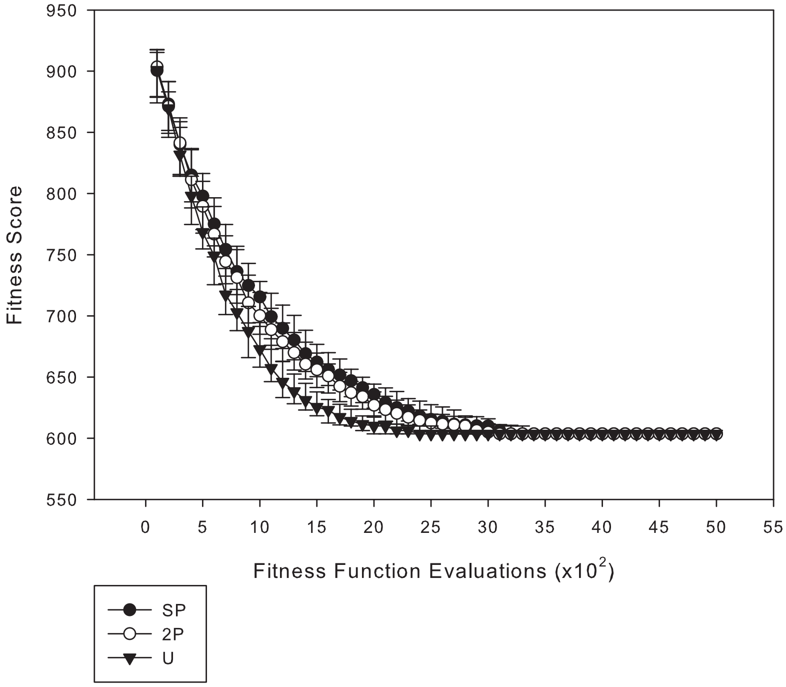


Figure 7.26: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 20$ and $R = ss$



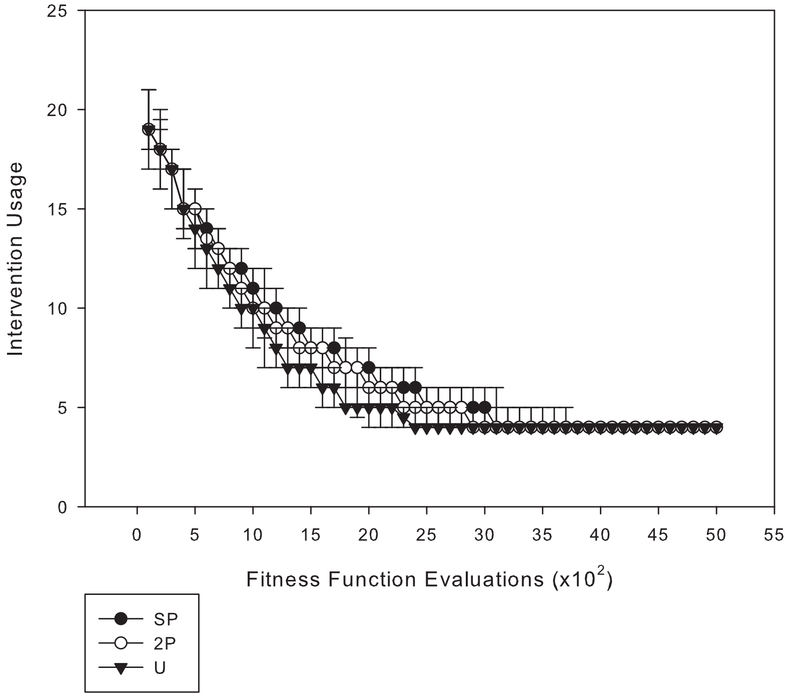
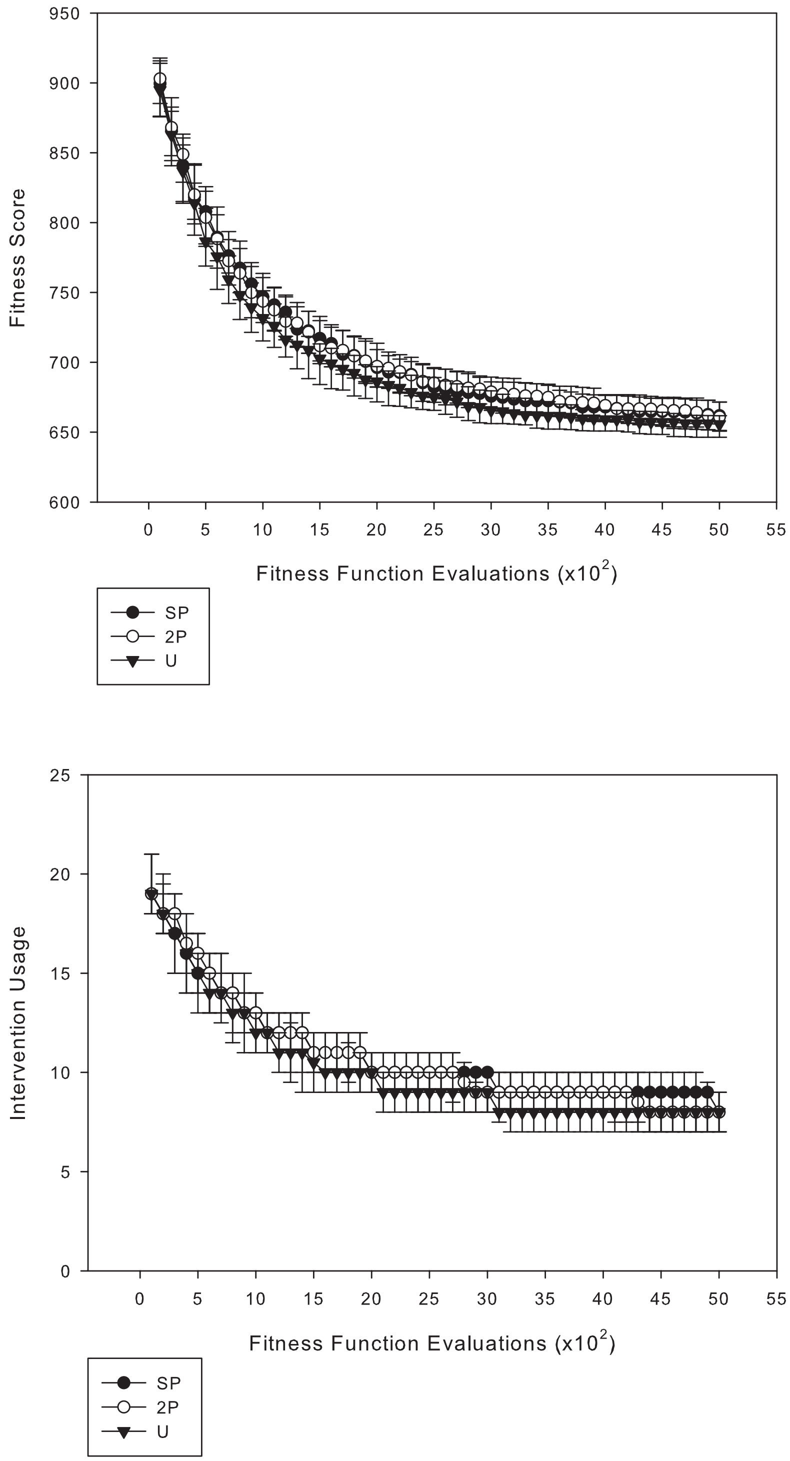


Figure 7.27: Results for $N = 100, p_c = 0.9, p_m = 0.05$, $P = 20$ and $R = ss$

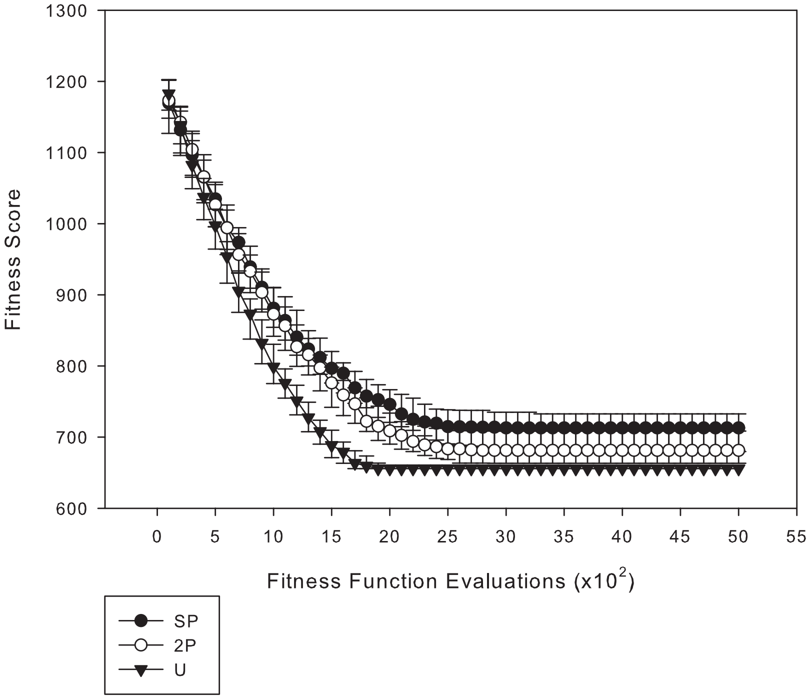


7.1.4 35 penalty points per intervention

With an increasing penalty associated per intervention a varying number of intervention levels have offered fitter scores. For an intervention penalty of 0, each crossover technique settled to approximately 47 interventions. With a penalty of 5 per intervention, approaches tended to around 17 interventions and for a penalty of 20 per intervention, each approach converged to schedules requiring placement of 4 interventions. The next two experiments review the fitness and intervention usage for the crossover approaches when the intervention penalty is increased to 35 and then 50 penalty points per intervention.

Figure 7.28 shows the fitness scores and associated intervention usage when $p_m = 0$. Figures 7.29 and 7.30 detail p_m levels of 0.005 and 0.05 respectively.

Figure 7.28: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 35$ and $R = g$



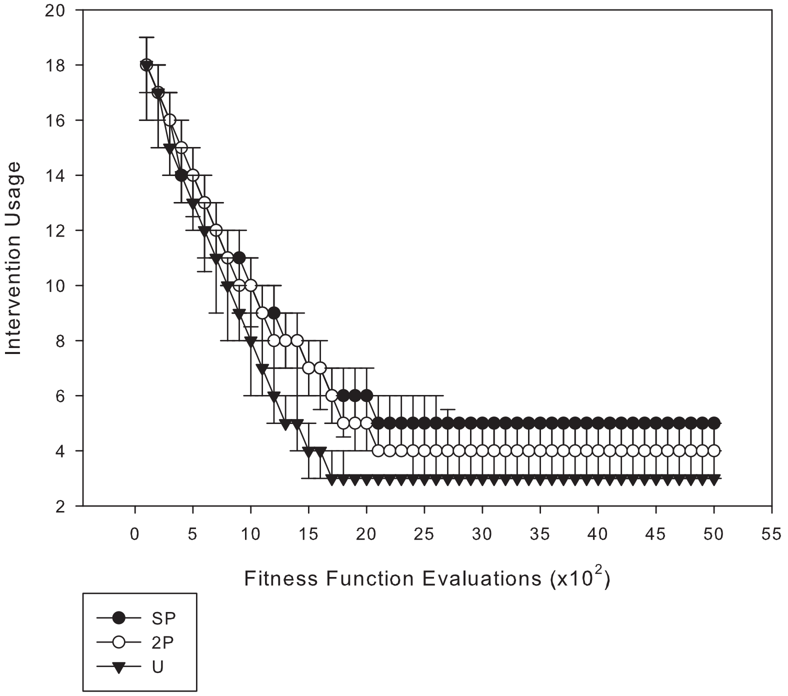
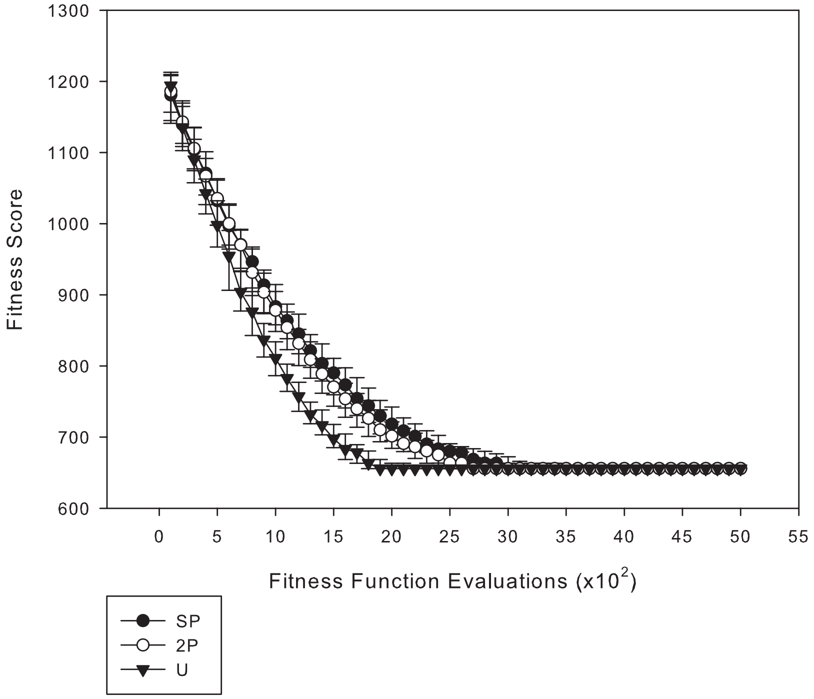


Figure 7.29: Results for $N = 100, p_c = 0.9, p_m = 0.005, P = 35$ and $R = g$



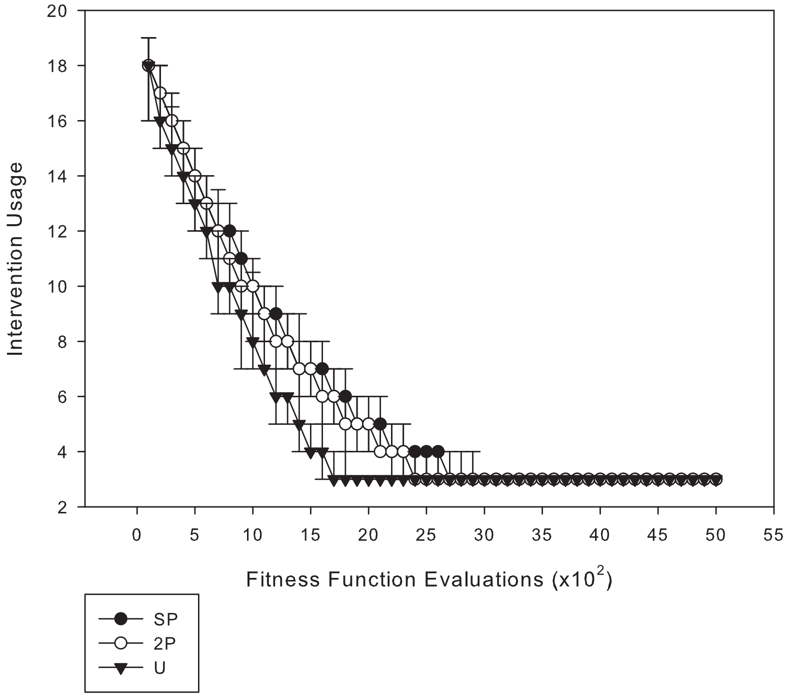
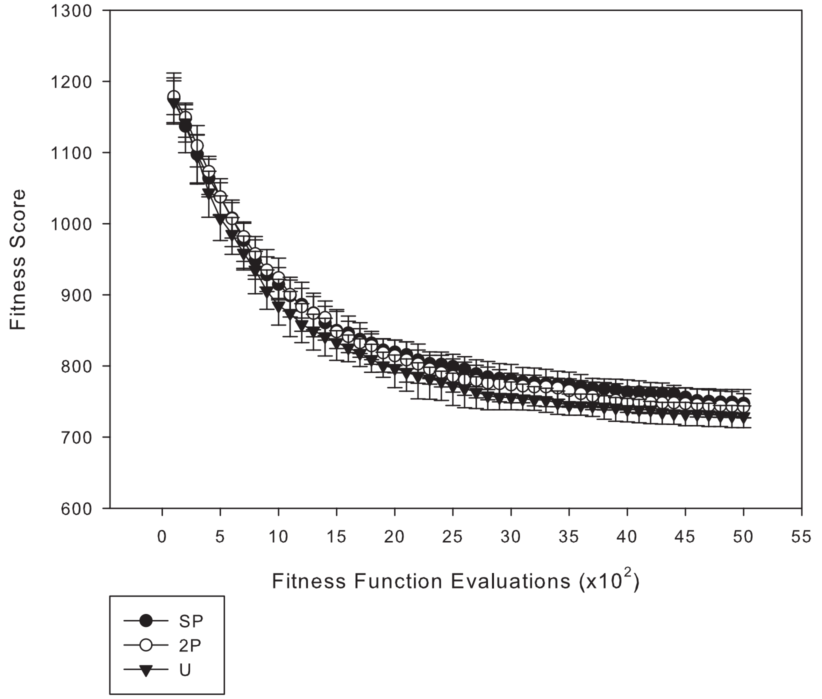


Figure 7.30: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 35$ and $R = g$



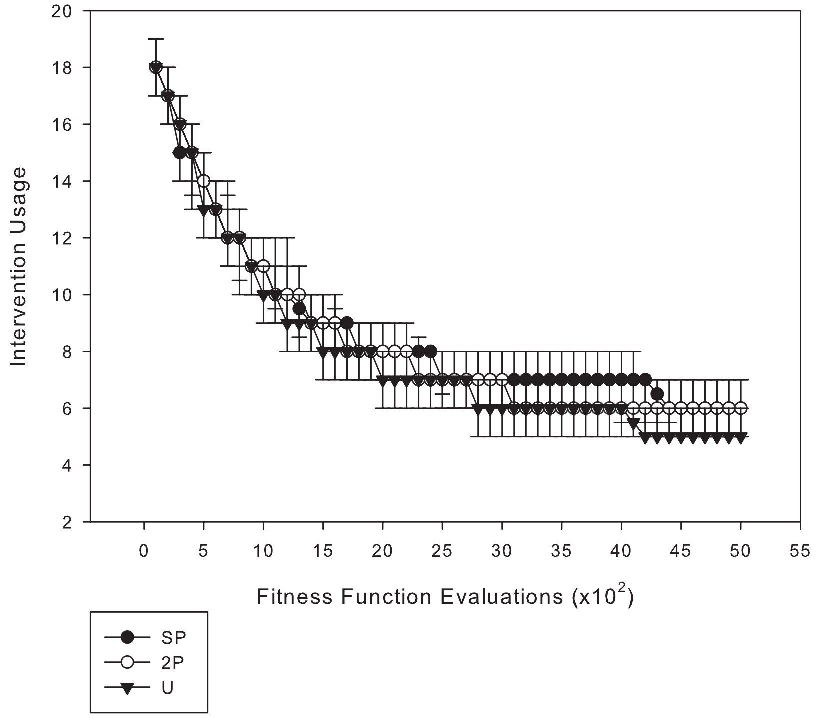


Figure 7.28 shows the intermediate parameter settings with generational replacement and a penalty of 35 points per intervention when $p_m = 0$. In this instance, as with previous penalty levels, UC is quicker to reduce intervention levels than both SPC and 2PC. UC converges to solutions using around 3 interventions as opposed to the 5 interventions used by SPC and 4 by 2PC. This reduced intervention usage returns fitter scores for UC as

opposed to SPC and 2PC.

Figure 7.29 shows the effect of increasing p_m to 0.005. As with a mutation rate of 0, UC converges to schedules using 3 interventions. By around 3,000 FFEs, both SPC and 2PC have also reduced to schedules of 3 interventions. With this similar usage of interventions in schedules, similar fitness scores are returned from each of the crossover approaches from this point.

Figure 7.30 details the effect of increasing the mutation rate, such that $p_m = 0.05$. As with the lesser penalty values, when the mutation rate is at this level, all crossover approaches return solutions with lower fitness scores when compared with a p_m of 0.005. Each of the crossover approaches utilise a similar number of interventions throughout the duration of this experiment and this in turn leads to similar fitness scores being returned by each of the crossover approaches.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the traditional crossover approaches, under a generational replacement strategy for a penalty of 35 points per intervention. Figure 7.28 detailed the fitness and intervention usage in the absence of mutation. Figure 7.31 shows the average placement of these interventions.

As with the lesser intervention penalties, Figure 7.28 shows that in the absence of mutation, UC uses fewer interventions than both SPC and 2PC to produce fitter scores. Also inline with previous experiments for this mutation level, 2PC uses fewer interventions than SPC to produce scores of a higher fitness. Figure 7.31 shows that UC focusses mainly on interventions 6 - 8, and rarely intervenes on any other day. 2PC focuses mainly on these days, but also has interventions on other days on the schedule. SPC, as with 2PC mainly intervenes early in the schedule, but also occasionally intervenes at other points in the schedule. 2PC has more interventions outwith days 6 - 8 than UC, and returns less fit scores. SPC has more interventions outwith days 6 - 8 than 2PC, and is poorer than 2PC. This would imply that the benefit gained by intervening at these points does not outweigh the associated penalties incurred.

Figure 7.31: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 35$ and $R = g$

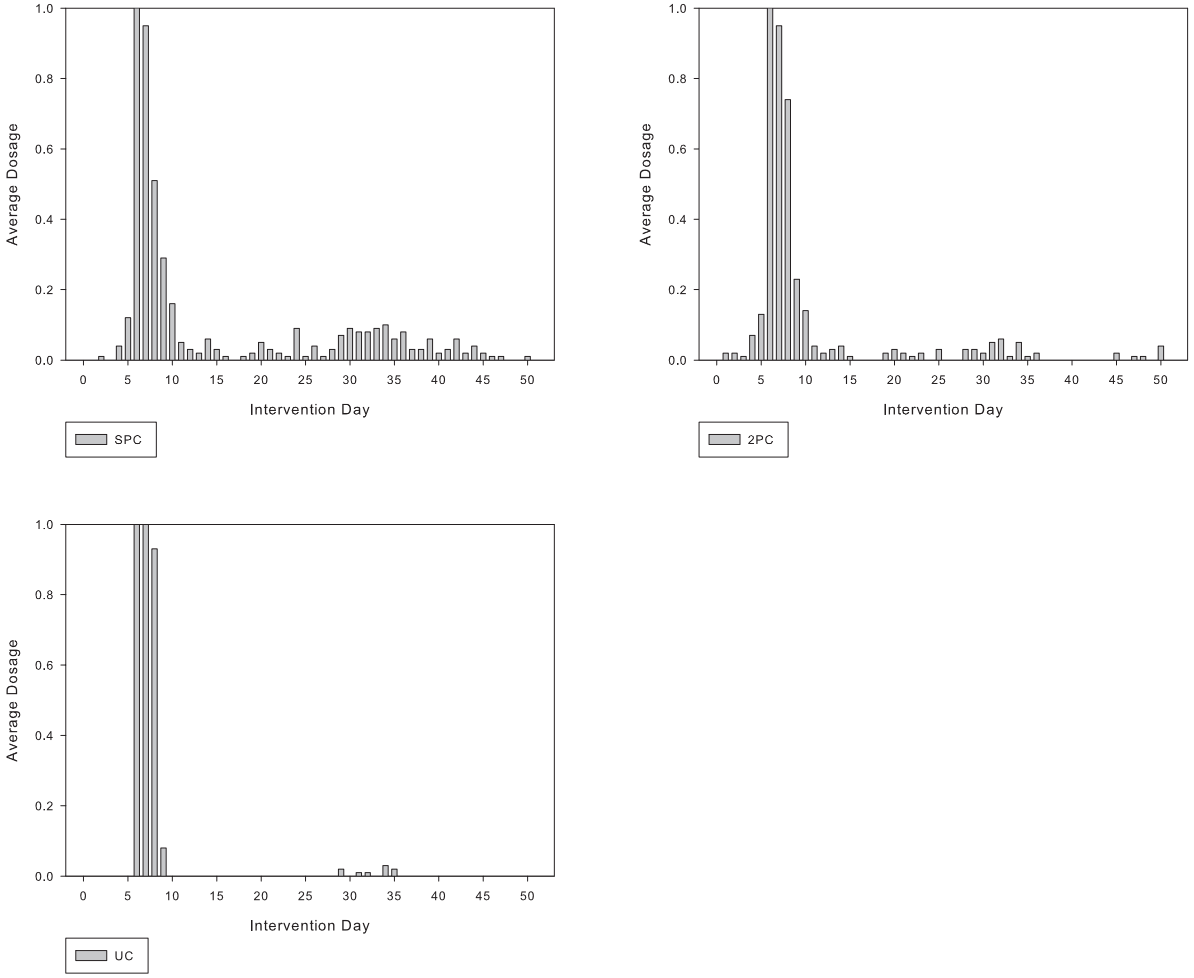


Figure 7.32 shows the intervention placement for the fitness scores and intervention usage detailed in Figure 7.29, when $p_m = 0.005$. Each of the crossover approaches return similar fitness scores for this mutation rate, using the same number of interventions in schedules. Figure 7.32 shows that each of the approaches focus interventions mainly on days 6 - 8, thus explaining the similarity in fitness scores returned.

Figure 7.32: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 35$ and $R = g$

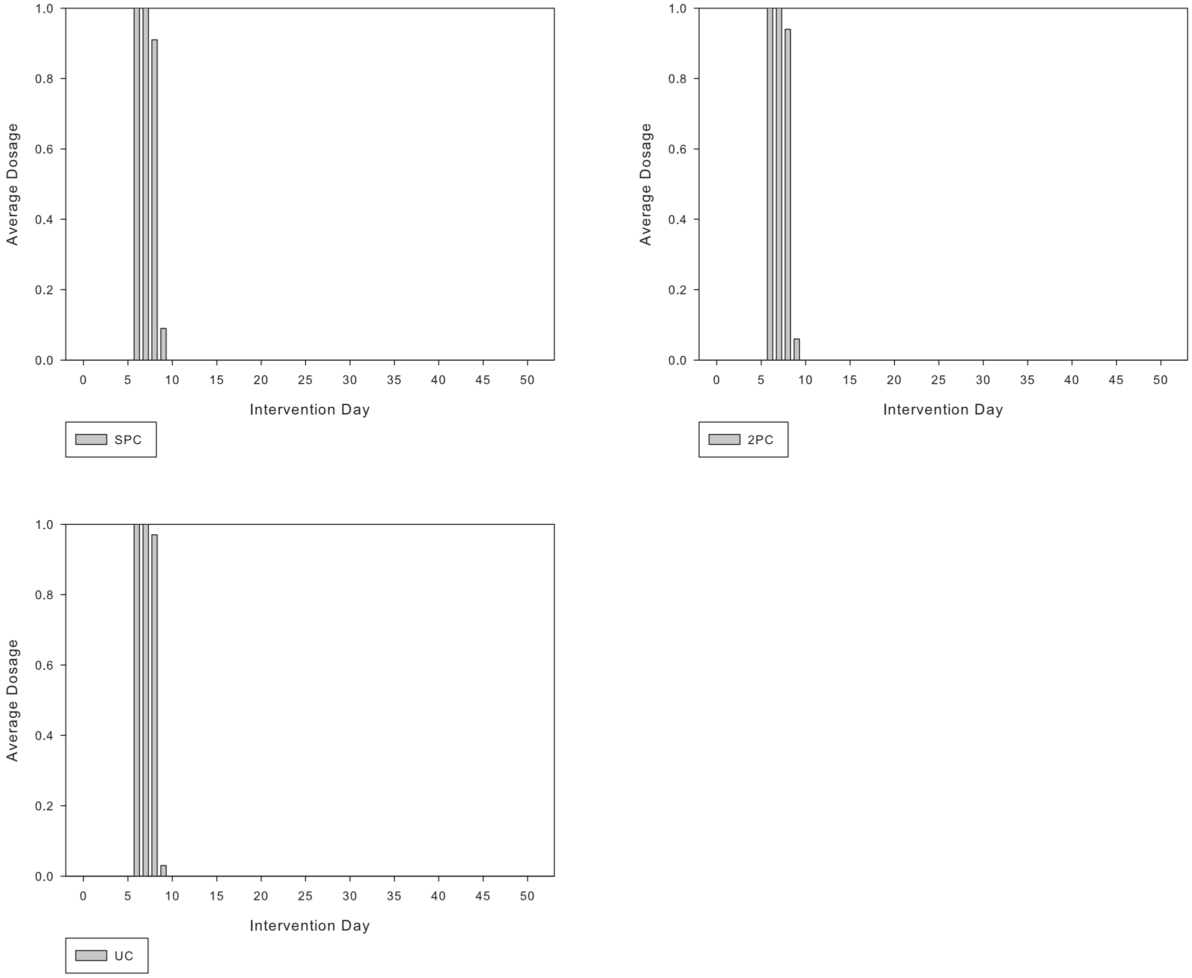
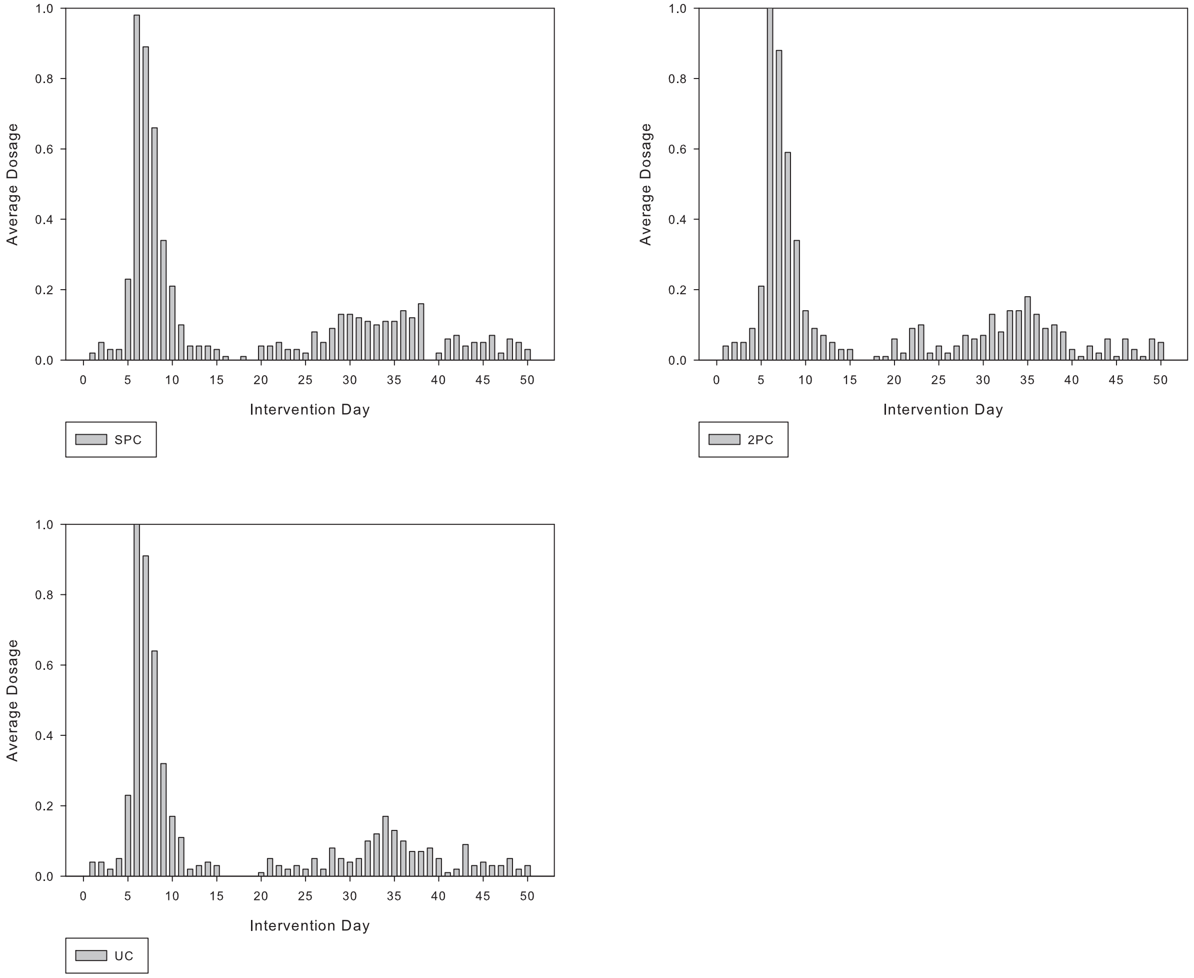


Figure 7.33 shows the intervention placement for the fitness scores and intervention usage detailed in Figure 7.30, when p_m is further increased to 0.05. For this mutation level, less fit scores are returned by each of the crossover approaches compared to a p_m level of 0.005. Figure 7.33 shows that while each of the crossover approaches focus interventions between days 6 and 8 of the schedule, applications on other days of the schedules do appear. This similarity in intervention placement explains the similar fitness scores returned by each of the crossover approaches for this mutation level.

Figure 7.33: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 35$ and $R = g$



Steady state replacement

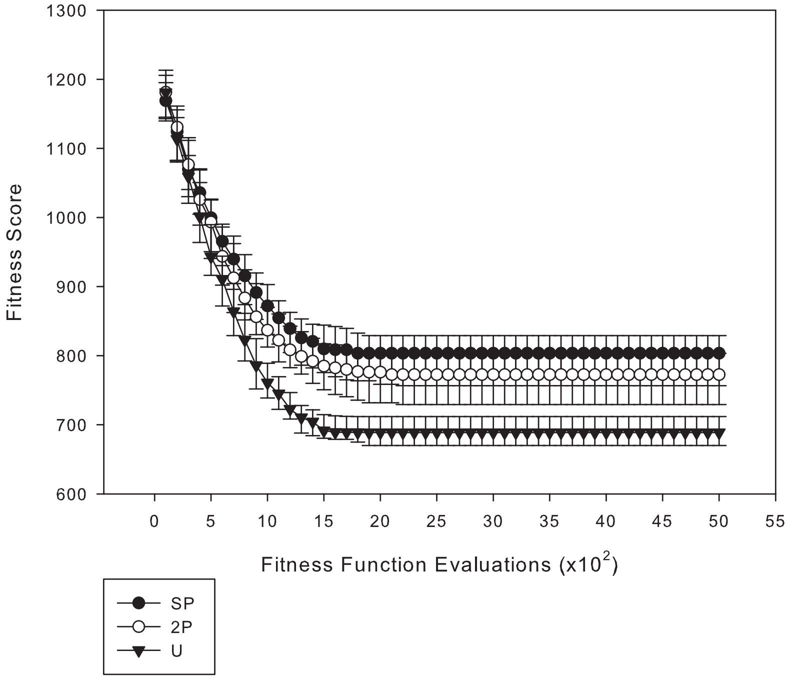
We now repeat the previous evaluation for a steady state replacement strategy. Figures 7.34 to 7.36 shows the fitness scores and intervention usage for each approach with intermediate parameter settings, a penalty of 35 points per intervention and a steady state replacement strategy.

Figure 7.34 shows $p_m = 0$. As with the previous intervention penalties, the steady state scores returned by each of the crossover approaches for this mutation level are less fit than those returned under a generational replacement strategy. This difference in fitness scores between approaches can be explained by the greater number of interventions required by schedules produced under steady state as opposed to generational replacement.

When p_m is increased to 0.005, as shown in Figure 7.35, similar trends hold for both generational and steady state replacement. Regardless of replacement strategy, UC is quicker at reducing interventions, and thus returns better scores for the first 3,000 FFEs. At this point, both SPC and 2PC are consistently producing schedules with a similar number of interventions as UC, and therefore return scores of a similar fitness.

Figure 7.36 shows the effect of further increasing the mutation rate, p_m, to 0.05. As when $p_m = 0.005$, there is little difference between scores returned or interventions used under either a steady state or a generational replacement strategy. As with this mutation level for lesser intervention penalties, each of the approaches create schedules using approximately similar numbers of interventions, resulting in similar fitness scores.

Figure 7.34: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 35$ and $R = ss$



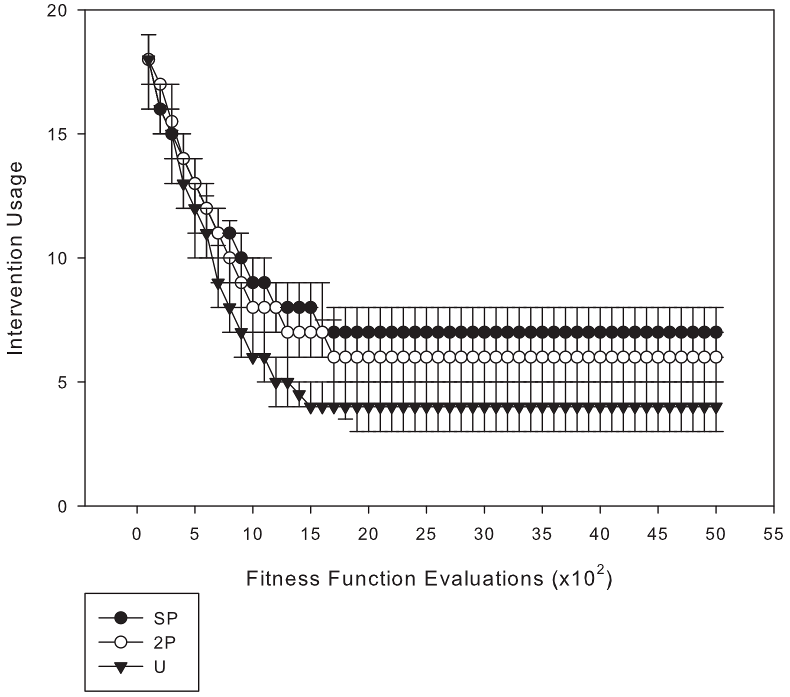


Figure 7.35: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 35$ and $R = ss$

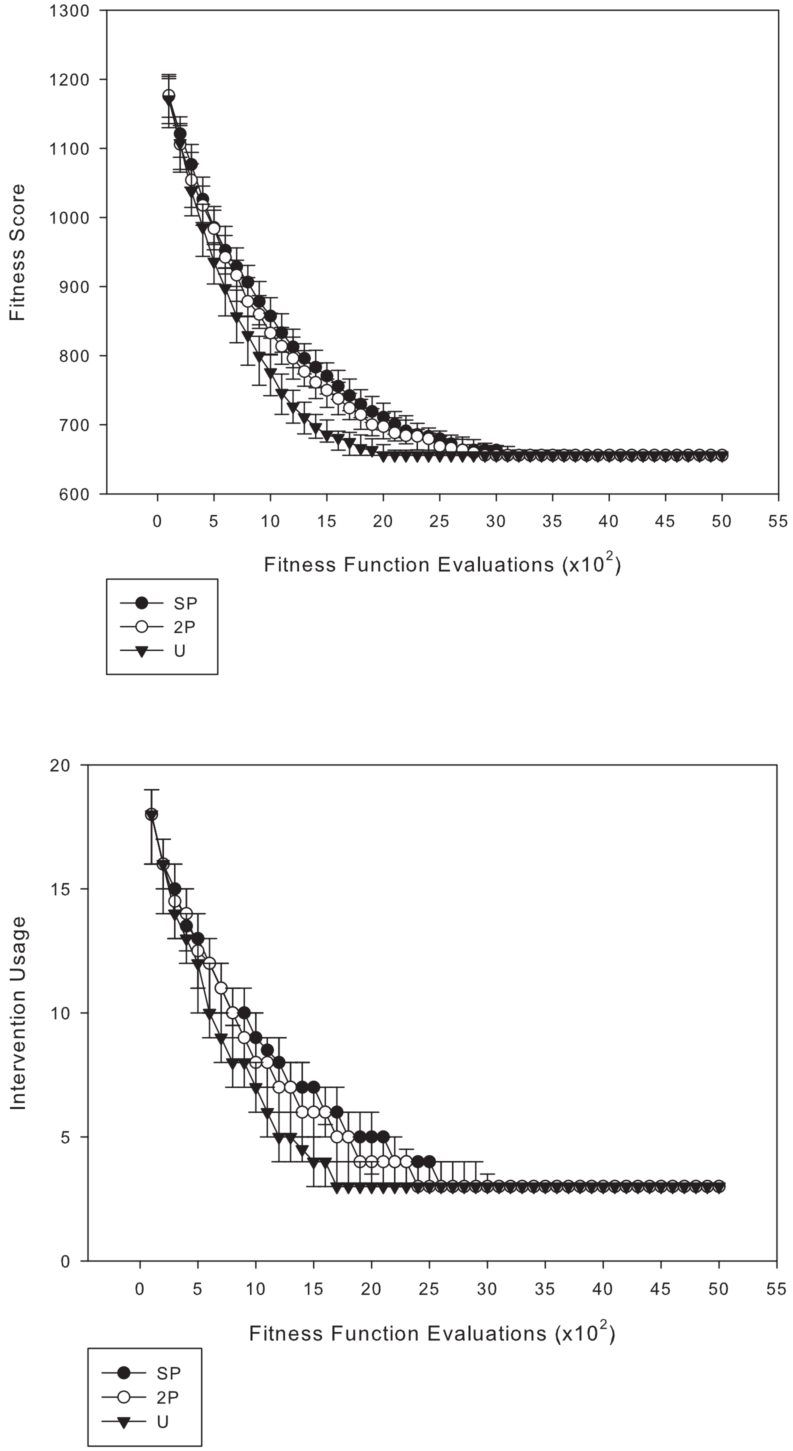
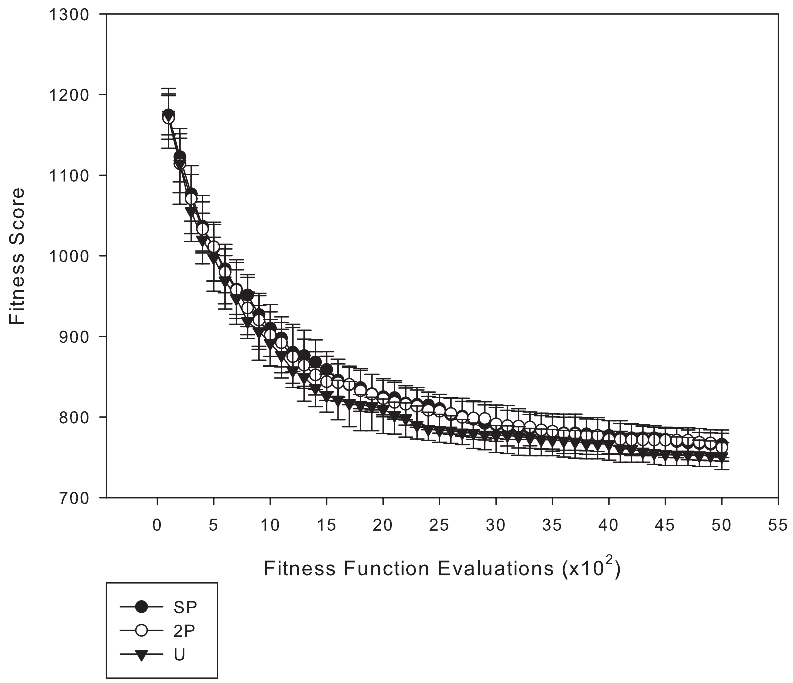
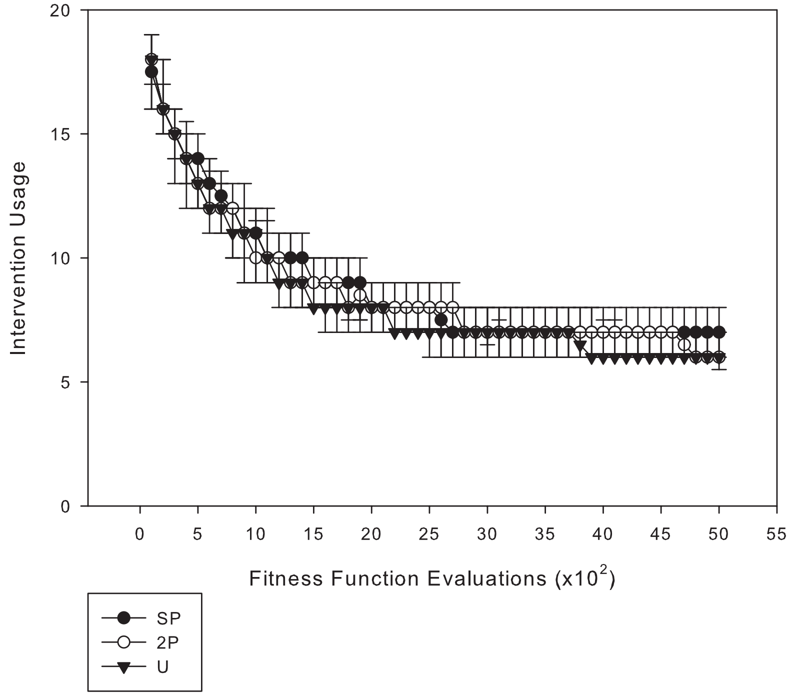


Figure 7.36: Results for $N = 100, p_c = 0.9, p_m = 0.05, P = 35$ and $R = ss$



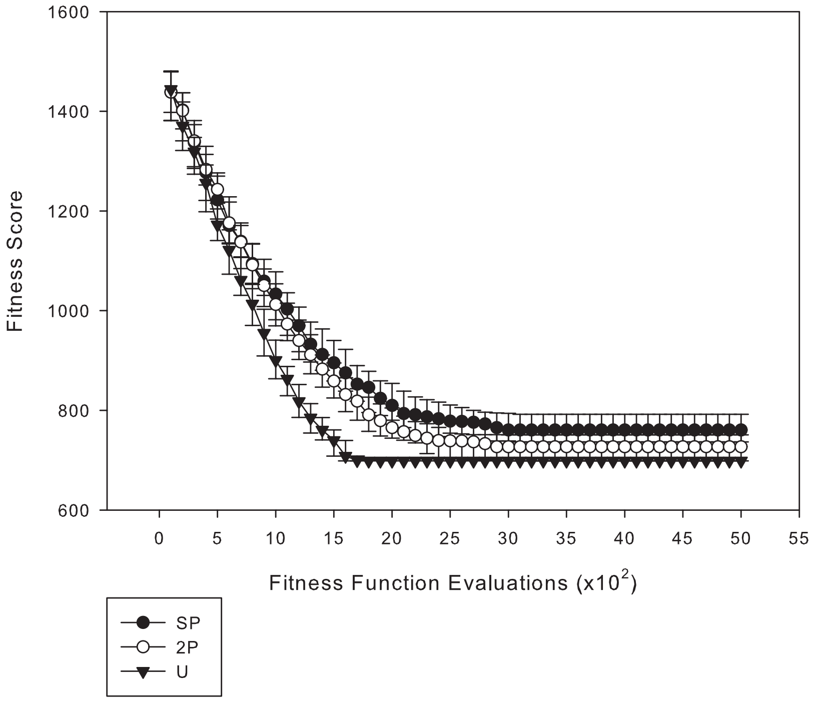


7.1.5 50 penalty points per intervention

The final experiment describes how SPC, 2PC and UC compare when the intervention penalty is further increased to 50 points per intervention. As described in Section 5.3.2, a penalty of 50 points per intervention represents the upper bound for the intervention penalty level.

Figures 7.37 to 7.39 show the fitness scores for each of the crossover approaches for intermediate parameter settings with a penalty of 50 points per intervention.

Figure 7.37: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 50$ and $R = g$



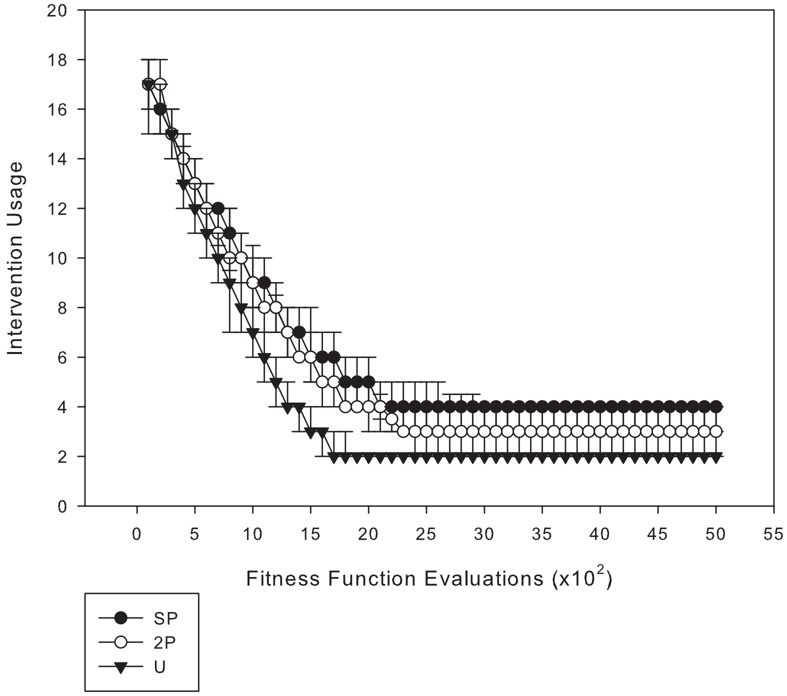
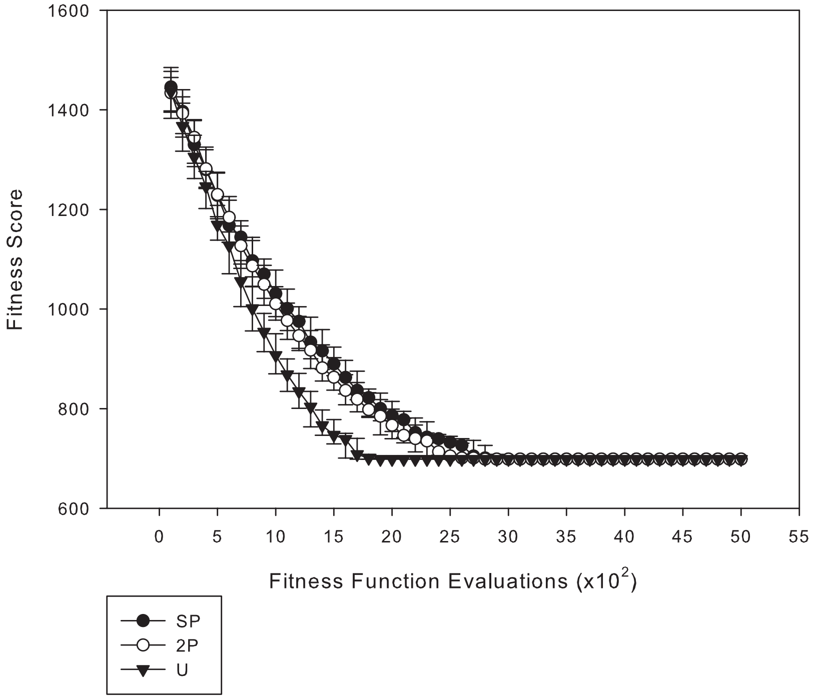


Figure 7.38: Results for $N = 100, p_c = 0.9, p_m = 0.005, P = 50$ and $R = g$



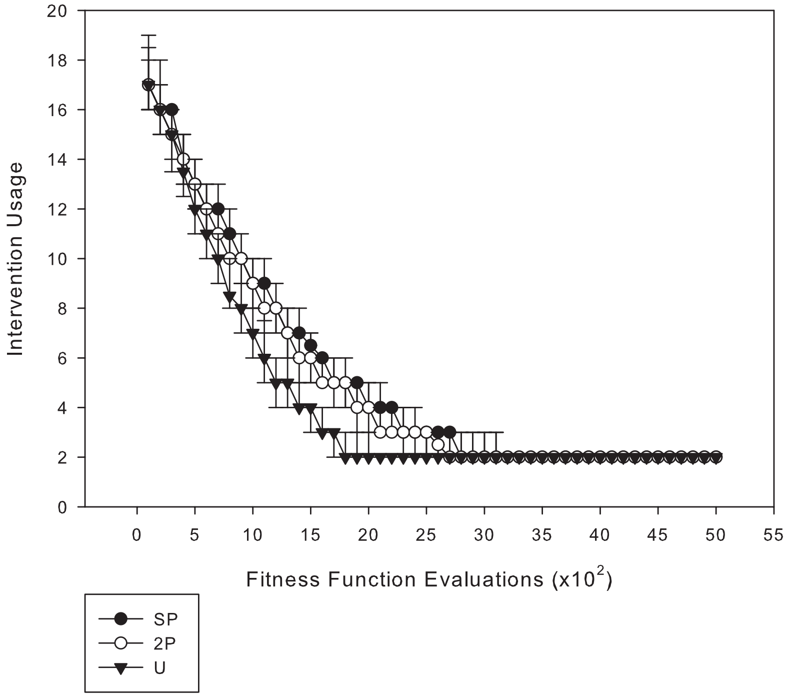
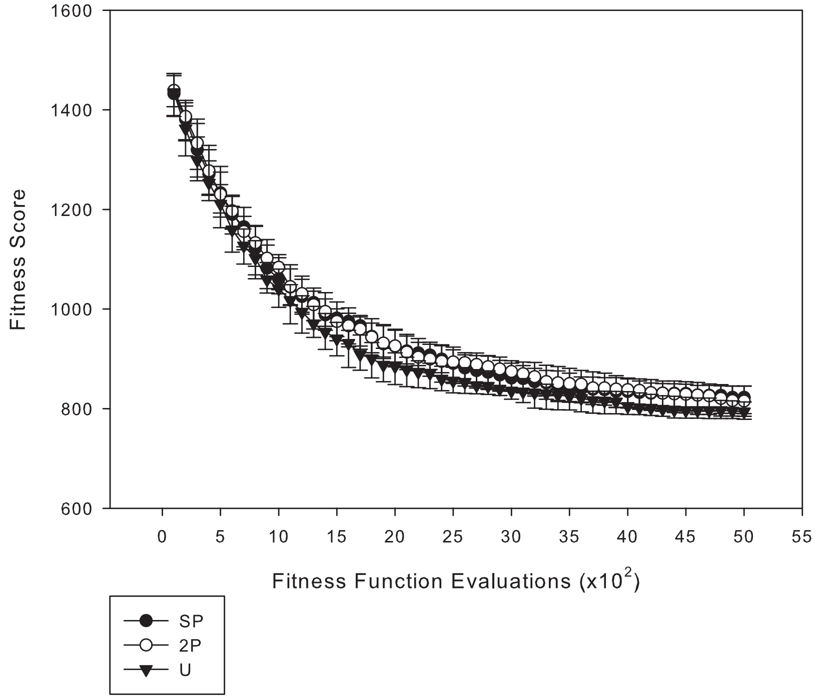


Figure 7.39: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 50$ and $R = g$



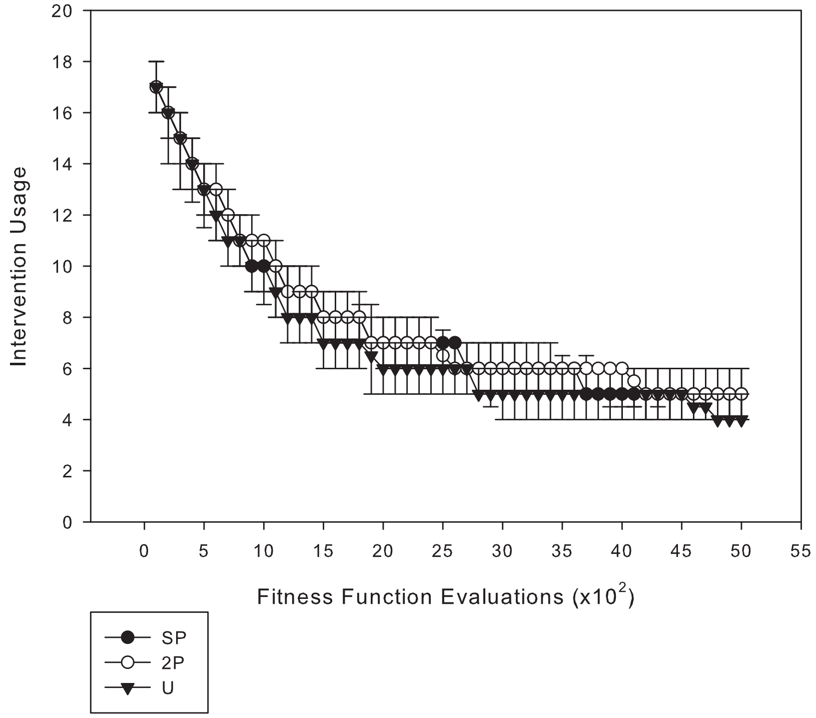


Figure 7.37 shows that when the probability of mutation is 0, as was the case with a smaller intervention penalty, UC is quicker at reducing interventions than the other approaches. UC settles on schedules requiring fewer interventions than those returned by SPC and 2PC. By UC using 2 interventions as opposed to 3 by 2PC or 4 by SPC, fitter scores are returned by UC for this parameter setting.

Figure 7.38 shows the effect of increasing the mutation rate, such that $p_m = 0.005$. After 2,700 FFEs, all approaches use around 2 interventions, resulting in similar fitness scores from this point. As with lesser intervention penalties, UC is quicker at reducing intervention levels and because of this produces fitter scores for the first half of the experiment.

The effect of further increasing the mutation rate to $p_m = 0.05$ is shown in Figure 7.39. The previous experiments have displayed that all approaches struggle to find the optimal scores for this higher mutation level. This remains the case when the intervention penalty is increased to 50. Although all approaches are reducing in the number of interventions, this is not as direct as with the lower mutation rates, resulting in the similar fitness scores returned by each of the crossover approaches.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the traditional crossover approaches, under a generational replacement strategy for a penalty of 50 points per intervention. Figure 7.37 detailed the fitness and intervention usage in the absence of mutation. Figure 7.40 shows the average placement of these interventions.

As with each of the lesser intervention penalties, in the absence of mutation, UC produces fitter scores than both 2PC and SPC, with 2PC scores in turn being fitter than those returned by SPC. Figure 7.40 shows that UC focusses 2 interventions, on days 6 and 7 of the schedule. 2PC and SPC both focus mainly on these points, but both display interventions on other days of the schedule as well, SPC more than 2PC. As there is a high penalty associated with interventing, these extra interventions in the schedules would explain the difference in fitness scores returned by each of the approaches.

Figure 7.40: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 50$ and $R = g$

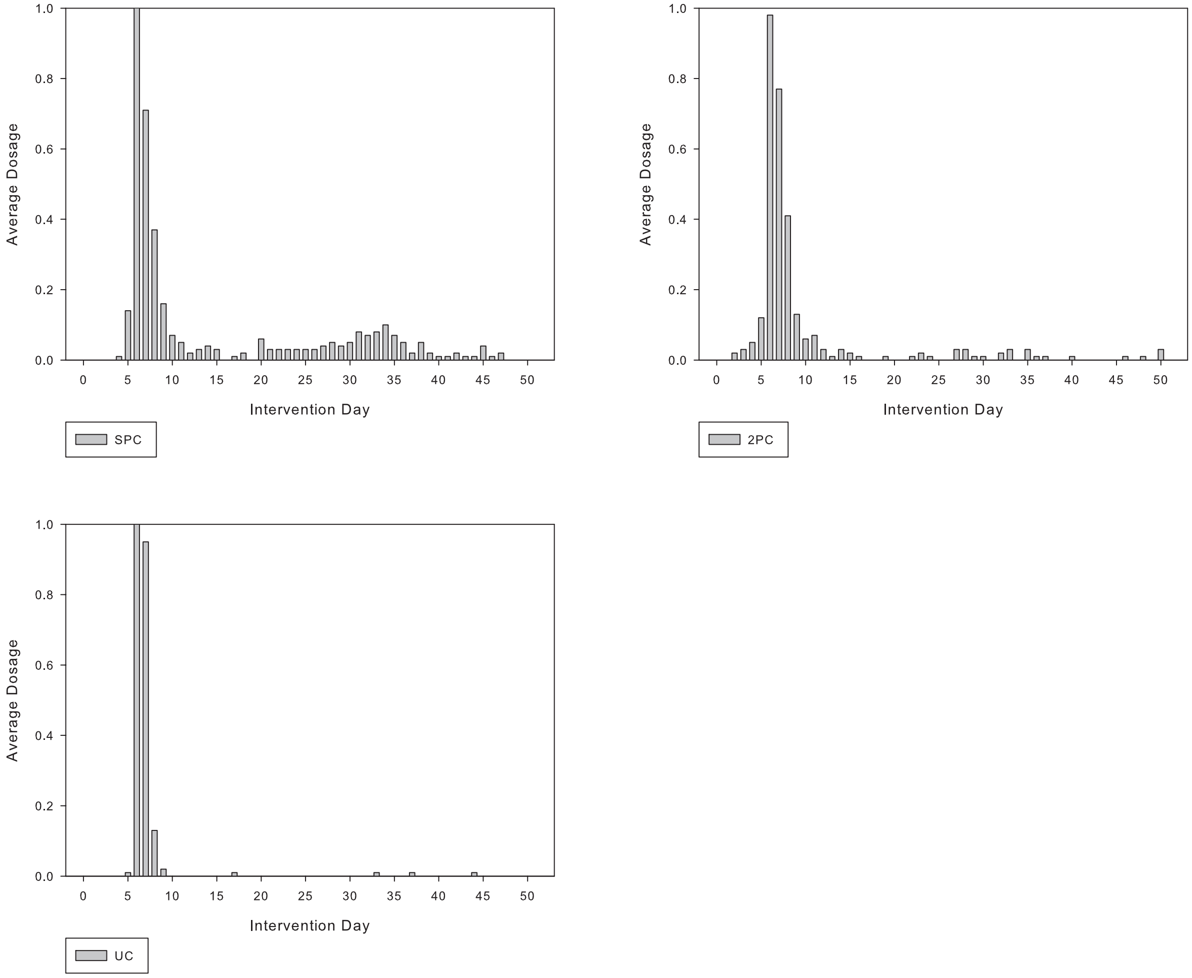
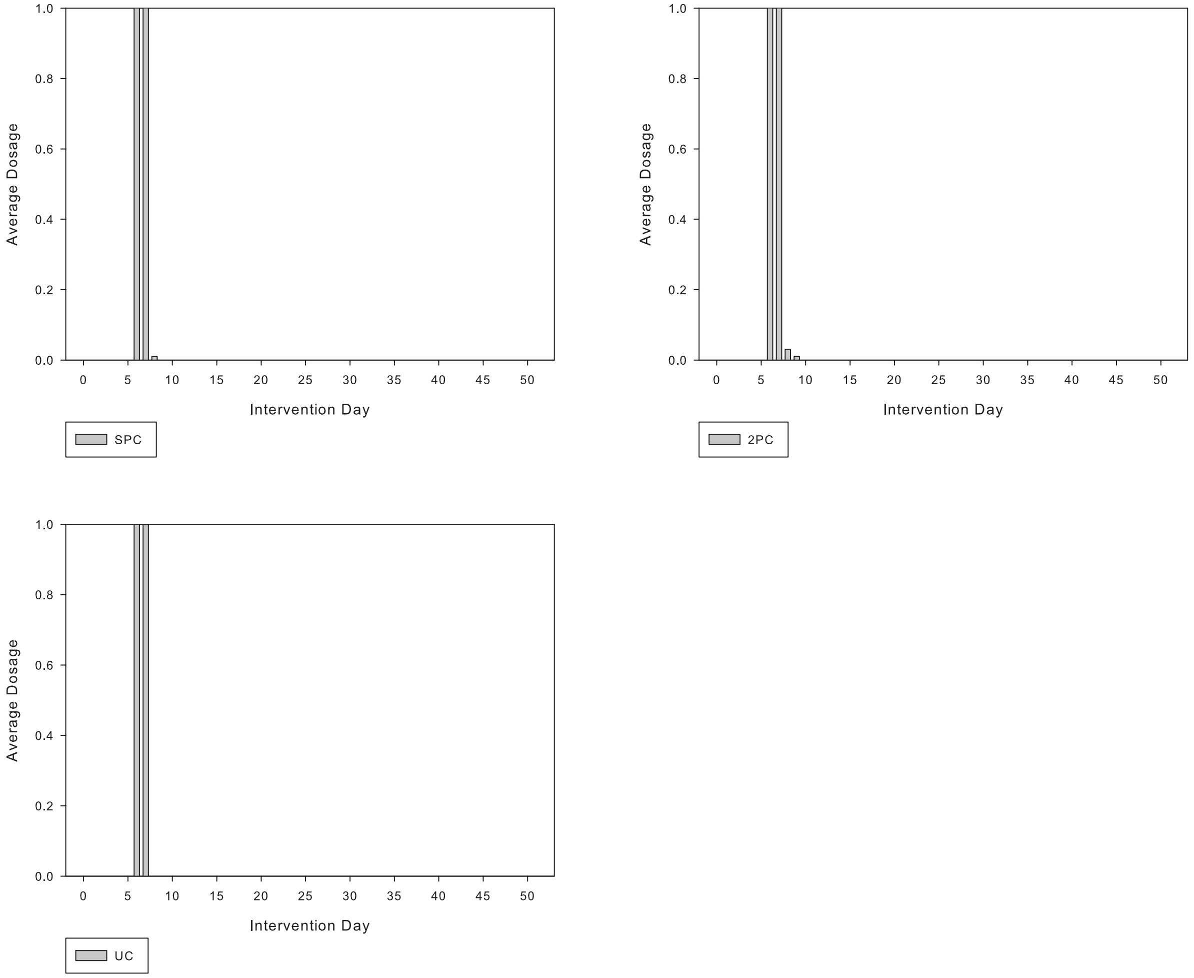


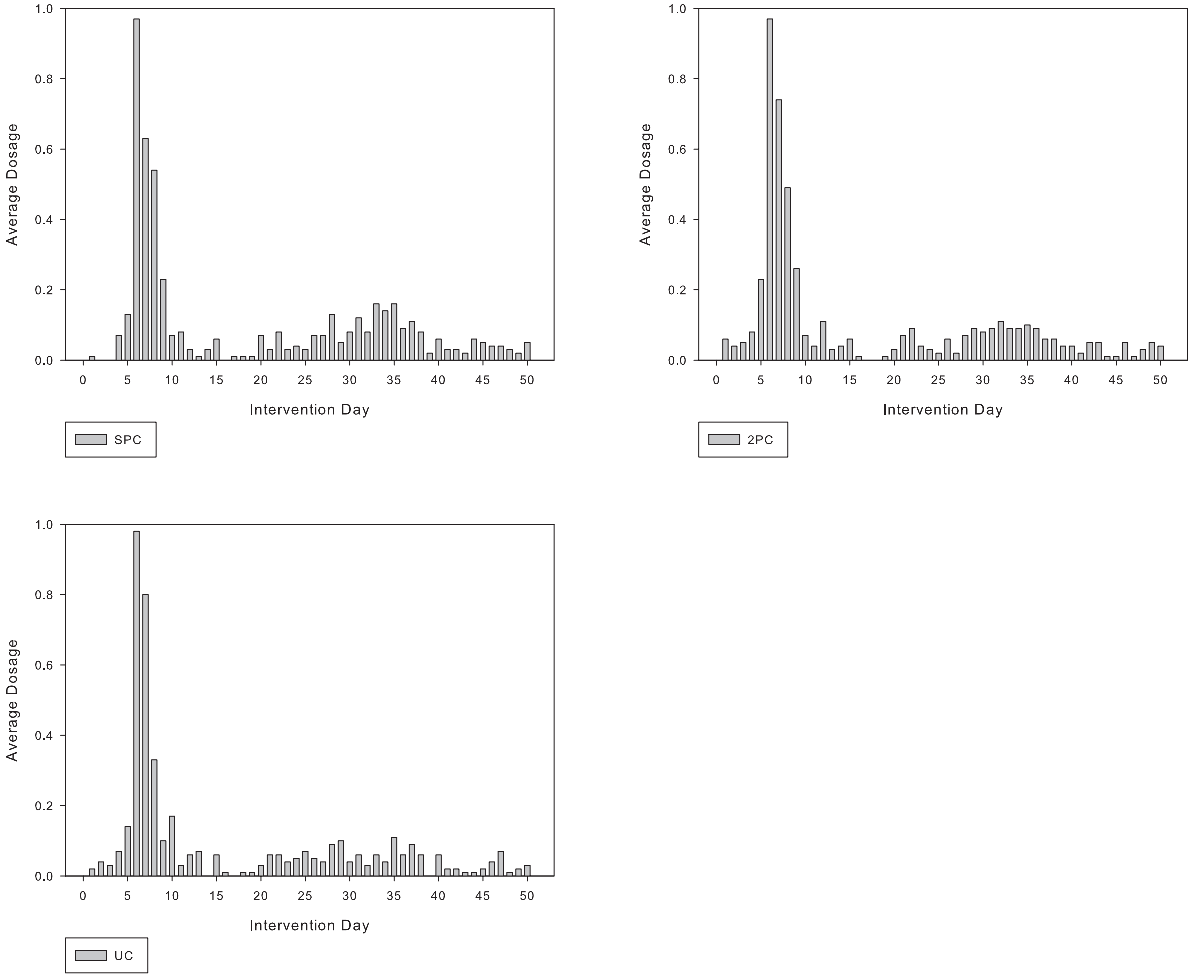
Figure 7.38 detailed the fitness and intervention usage when p_m was increased to 0.005 and Figure 7.41 shows the average placement of these interventions. For this mutation level, each of the approaches focus application on days 6 and 7 of the schedule, and this explains the similarity in fitness scores returned.

Figure 7.41: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 50$ and $R = g$



The effect of further increasing p_m to 0.05 was shown in Figure 7.39. For this mutation level each of the approaches used a similar number of interventions to produce fitness scores in the same range. Figure 7.42 shows the placement of interventions by each of the crossover approaches. Each of the approaches focus application on days 6 and 7 and display similar trends in terms of lesser applications used. This similar placement of interventions by each approach accounts for the fitness scores in the same range returned by each of the crossover approaches.

Figure 7.42: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 50$ and $R = g$



Steady state replacement

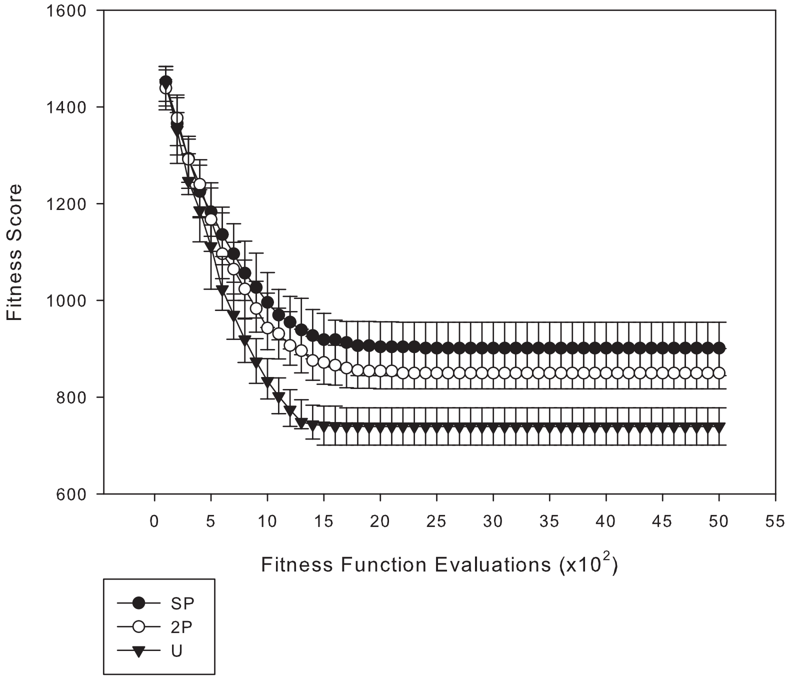
The same experiment will now be reviewed for a steady state replacement strategy. Figures 7.43 to 7.45 show the fitness scores and associated intervention usage for each crossover approach with intermediate parameter settings, a penalty of 50 points per intervention and a steady state replacement strategy.

Figure 7.43 shows that as with lesser intervention penalties, each of the approaches perform worse under steady state replacement as opposed to the generational approach when $p_m = 0$. UC outperforms both SPC and 2PC in that it finds fitter scores in fewer FFEs. UC also converges to solutions of a better fitness than those returned by SPC and 2PC.

The effect of increasing the mutation rate, such that $p_m = 0.005$ is shown in Figure 7.44. As with smaller intervention penalties, the performance of the crossover approaches does not differ between replacement strategies. UC is again quicker at reducing in intervention levels and as a result of this returns fitter scores for the first 2,700 FFEs. At this point, both SPC and 2PC are producing schedules with a similar number of interventions and return solutions of a similar fitness from this point onwards.

Figure 7.45 shows the effect of further increasing p_m from 0.005 to 0.05. As with earlier experiments featuring smaller intervention penalties, each of the crossover approaches perform in a similar way, irrespective of the replacement strategy used. SPC, 2PC and UC produce schedules using a similar number of interventions during the course of this experiment and, as was the case for generational replacement, similar fitness scores are returned by each of the crossover approaches.

Figure 7.43: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 50$ and $R = ss$



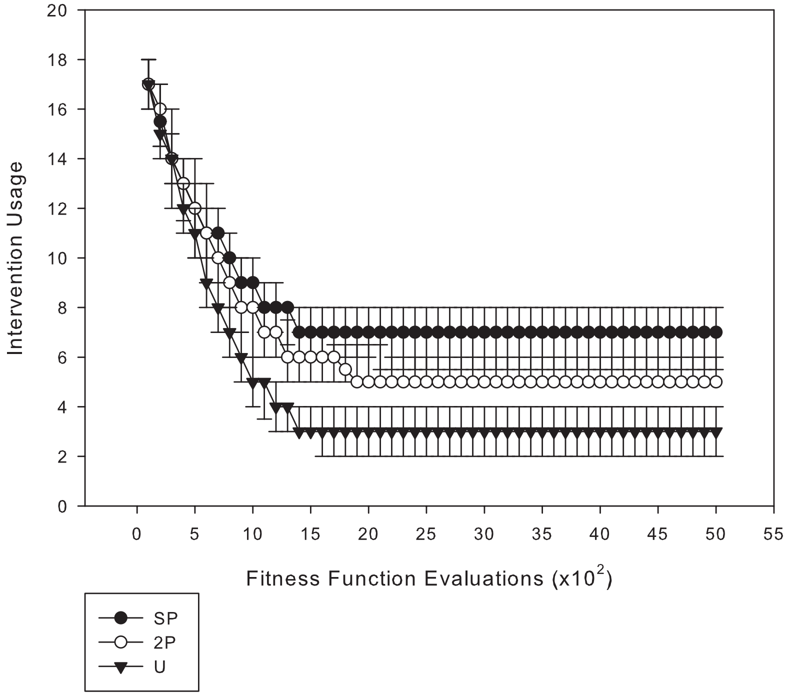


Figure 7.44: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 50$ and $R = ss$

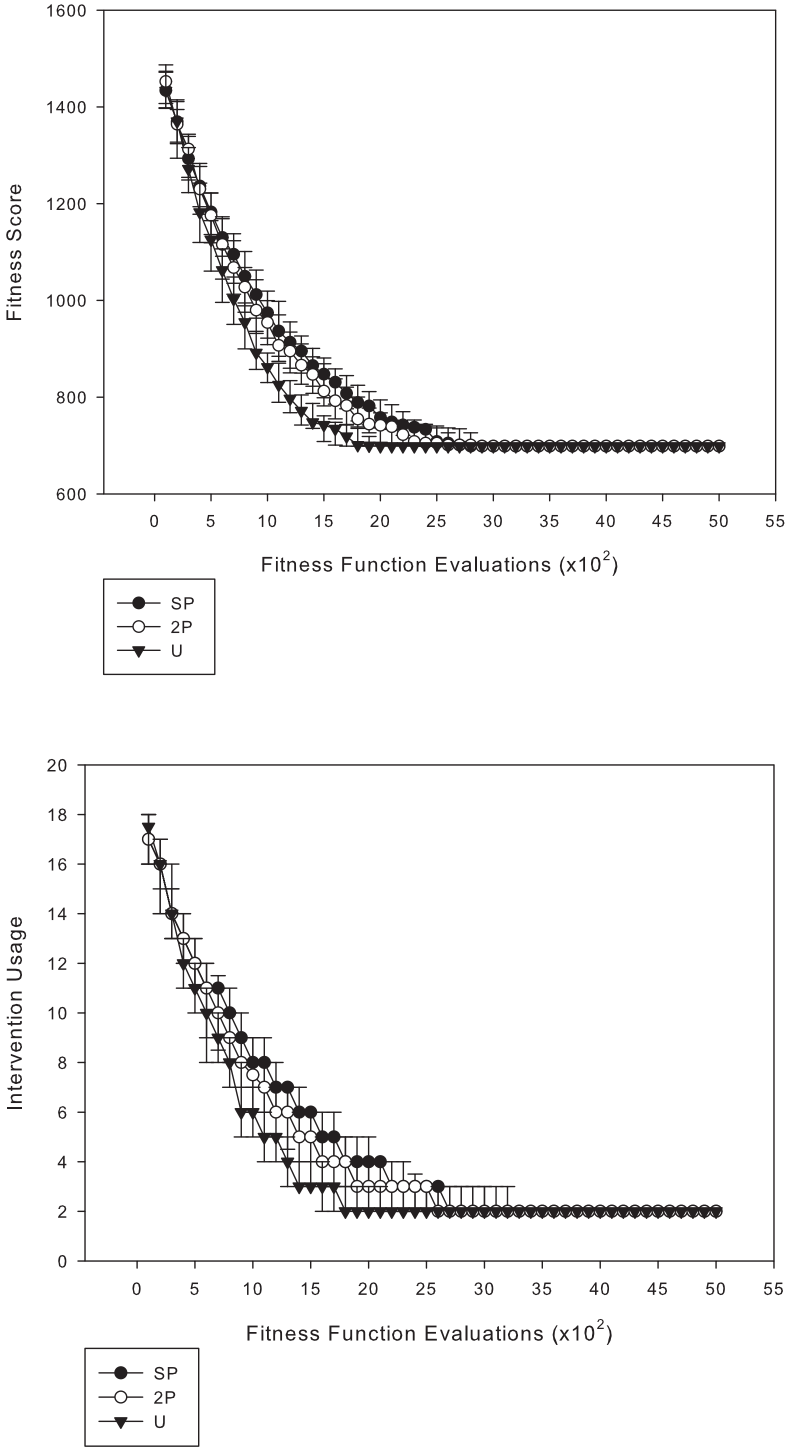
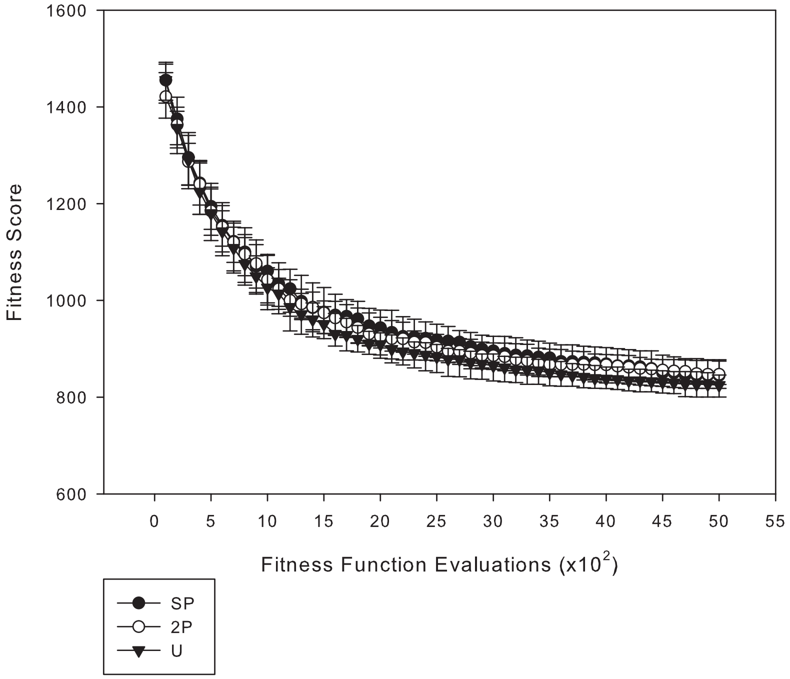
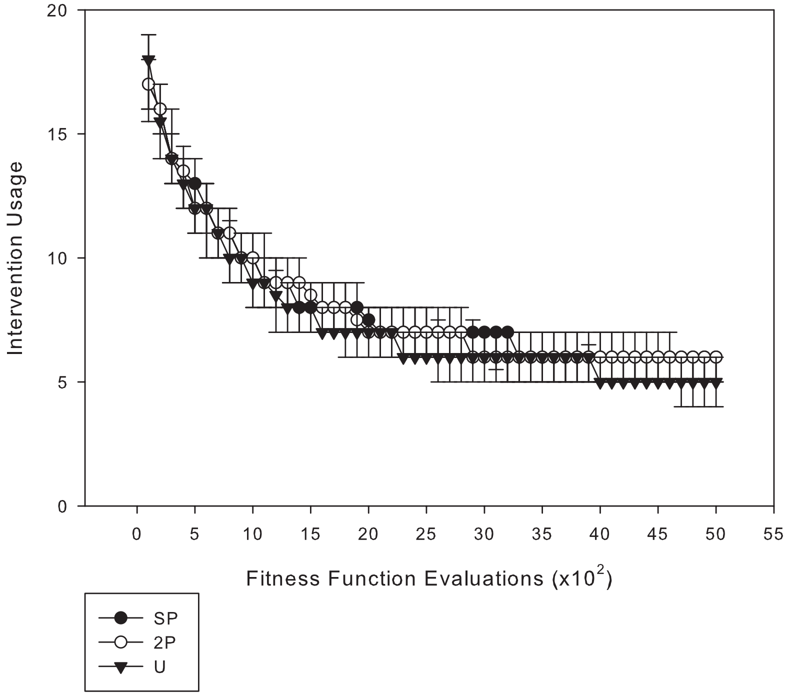


Figure 7.45: Results for $N = 100, p_c = 0.9, p_m = 0.05, P = 50$ and $R = ss$





7.1.6 Summary of bio-control scheduling experiment

This section has reviewed the traditional crossover techniques, SPC, 2PC and UC for deriving optimal control schedules of bio control applications over a range of intervention penalties, mutation rates and replacement strategies. The intervention penalties have ranged from 0, as described in Section 7.1.1 through to a penalty of 50 per intervention, as detailed in Section 7.1.5.

Through reviewing SPC, 2PC and UC over this range of penalties, three trends have appeared, namely:

1. Trend 1: When no mutation is present, UC is better than both SPC and 2PC at deriving fit solutions.
2. Trend 2: With a mutation rate of 0.005, UC is quicker at finding fitter solutions than both SPC and 2PC.
3. Trend 3: When the mutation rate is increased to 0.05, all approaches exhibit similar performance, and each return weaker scores than those found for a p_m level of 0.005.

Each of these trends hold regardless of whether a generational or a steady state replacement strategy is used. It can be further noted that with mutation levels of 0.005 or 0.05, the choice between generational or steady state replacement has little effect on each of the crossover operators. When $p_m = 0$, under a steady state replacement strategy, each of the crossover approaches display weaker performance compared with the generational equivalent. Although each crossover approach is affected by this change in replacement strategy, SPC and 2PC were affected by this change more than UC. Indeed, when $p_m = 0$, SPC and 2PC appear to struggle to converge to good solutions.

An explanation of this difference in fitness scores between replacement strategies when $p_m = 0$ is given in Section 7.1.1. It can be observed that as no new material is being added into the population through mutation, using generational replacement provides more combinations of material into the population than the steady state alternative. As UC can recombine more genes per crossover process than both SPC and 2PC this would explain its ability to be less affected by the choice of replacement strategy when $p_m = 0$.

The placement of interventions during the 50 day schedule has also been reviewed for each of the intervention penalties. This showed that while each of the crossover approaches target both cycles of larvae growth for lesser intervention penalties, as the penalty increases, intervention focuses mainly on the earlier larvae cycle. Section 7.1.3 details that this is in line with previous empirical evaluation of this bio-control problem.

This set of experiments were conducted to find out if one of the traditional crossover approaches was consistently at least as good as the other approaches for the task of optimising bio-control schedules. It was proposed that if one of the traditional crossover approaches was at least as good as the others, it would serve as an appropriate baseline comparison method for the directed intervention techniques. From reviewing SPC, 2PC and UC across varying mutation rates, crossover rates, intervention penalties and replacement strategies, UC is shown to consistently perform at least as effectively as both SPC and 2PC. For this reason, the comparisons of directed intervention crossover techniques for bio-control scheduling as described in Section 8.1 will use UC as a baseline crossover approach for the purposes of comparing performance for the bio-control problem.

7.1.7 Further analysis

The previous sections reviewed SPC, 2PC and UC across a range of intervention penalties using the intermediate parameter settings. From these experiments a number of trends were observed that held for each of the parameter settings, regardless of the penalty value per intervention.

In order to further analyse the abilities of each of the crossover approaches, it appears necessary to review them across a range of parameters outwith the intermediate set discussed previously. As with previous experiments, this review will consider the range of intervention penalty values. Appendix A.1 details the abilities of SPC, 2PC and UC across the range of population sizes, crossover rates, mutation rates and intervention penalties. This describes how in the absence of mutation, UC finds fitter scores than both SPC and 2PC and that when the mutation rate is increased to 0.005, all approaches find comparable solutions, with UC quicker at deriving these. This also describes how each of the crossover approaches return comparable scores when the mutation rate is increased further to 0.05, with these scores being less fit than those found under a mutation rate of 0.005.

Several trends have been demonstrated regardless of the population size or crossover rate for these experiments. For experiments with no mutation, UC appears as the more flexible crossover operator, returning fitter solutions than both SPC and 2PC. When the mutation rate is increased to 0.005, both SPC and 2PC increase in performance relative to UC. Even with this increase in performance, which sees SPC and 2PC converge to similar solutions as UC over time, UC is quicker at finding these fitter solutions. When the mutation rate is further increased to 0.05, it appears that this mutation level is sufficiently high that too much randomness is introduced into the population. This makes it hard for all crossover approaches to find good solutions. For this mutation rate, all crossover approaches appear comparable. Each of these trends are in line with those described previously in Section 7.1.6.

7.2 Evaluation of traditional crossover techniques for chemotherapy treatment scheduling

Each of the traditional crossover approaches will now be reviewed for the task of chemotherapy treatment scheduling. Section 7.1 reviewed SPC, 2PC and UC for the optimisation of bio-control schedules. This section extends on these experiments by reviewing the effectiveness of each of the traditional crossover approaches at deriving chemotherapy treatment schedules. The formulation of this chemotherapy problem was described in Section 5.4.1.

For this evaluation, the chemotherapy treatment scheduling problem is sampled over 40,000 fitness function evaluations (FFEs), at intervals of 1,000, resulting in 40 data points for review. For each sample point, 100 runs of each crossover approach were recorded for each of the parameter combinations. The best scoring values for each run for each FFE are recorded. This is consistent with the parameters used in Section 6.9. As described in Section 6.9.1, as a normal distribution cannot be assumed in these experiments, the median and interquartile range of these values are displayed.

The following sections review the effectiveness of SPC, 2PC and UC for chemotherapy treatment scheduling for both generational and steady state replacement strategies. As detailed previously, it is important to review the crossover techniques with a range of parameter settings. A complete set of results has therefore been collected and these can be seen in Appendix F. This appendix contains the complete set of results showing the fitness scores against FFEs associated with these experiments across a range of population sizes, crossover rates and mutation rates.

As with the bio-control experiments, rather than reviewing each parameter combination in turn, it appears prudent to consider the intermediate parameter settings for each approach, instead of explicitly detailing each possible parameter combination. To this end, each of the following sections will review SPC, 2PC and UC with a population of $N = 100$, crossover rate of $p_c = 0.9$ and mutation rates, p_m, of 0, 0.005 and 0.05. One can then compare these against the complete set of results in Appendix F to check for consistency. Unlike the bio-control scheduling problem, there is no explicit penalty associated with each intervention.

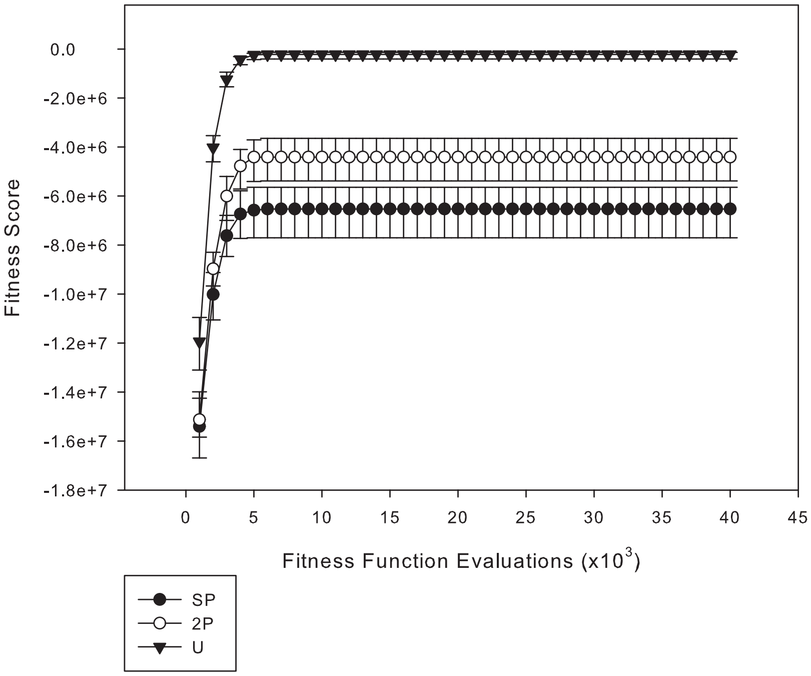
Initial analysis for each intervention penalty will review these intermediate parameter settings with a generational replacement strategy. This will be followed by a corresponding analysis of the intermediate parameter settings under a steady state replacement strategy. As described previously, the replacement mechanism, R, for each experiment will be defined as either generational, $R = g$, or steady state, $R = ss$.

Graphs displaying the results for both the fitness scores and intervention usage for the intermediate parameter settings under both generational and steady state replacement will now be reviewed. Both fitness versus FFEs and intervention usage versus FFEs are plotted over the range of FFE samples. As the task of cancer chemotherapy treatment optimisation is a maximisation problem, higher scores are fitter than lower valued scores.

The intervention graphs show the number of interventions associated with the corresponding fitness graph for the same FFE sample point. Intervention usage represents the number of times that the patient receives chemotherapy drugs, with an intervention of 0 representing no application of chemotherapy drug doses and 100 interventions showing application of the chemotherapy drug on every day of the schedule.

Figures 7.46 to 7.48 show the fitness score and intervention usage for intermediate parameters and a p_m of 0, 0.005 and 0.05 respectively.

Figure 7.46: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$ and $R = g$



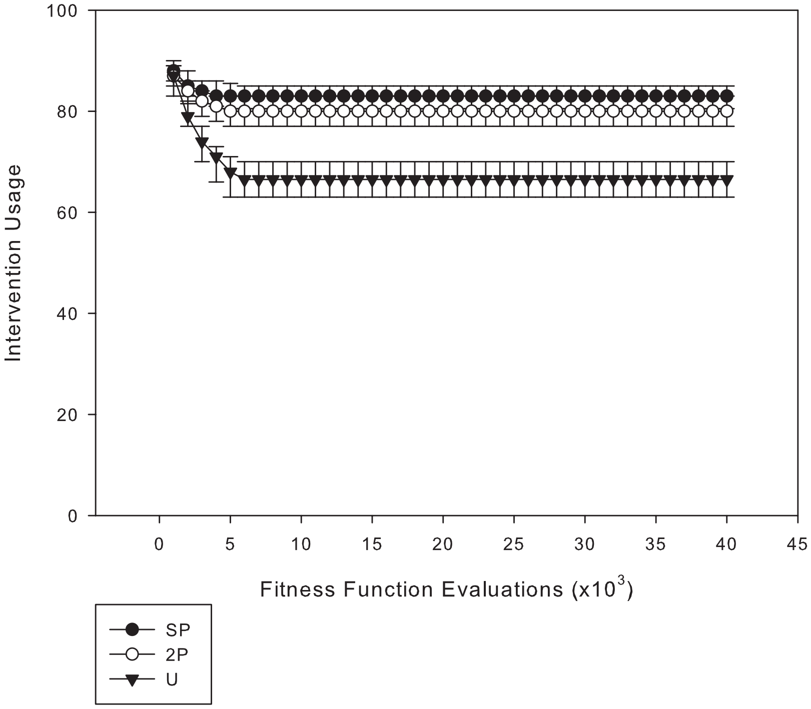
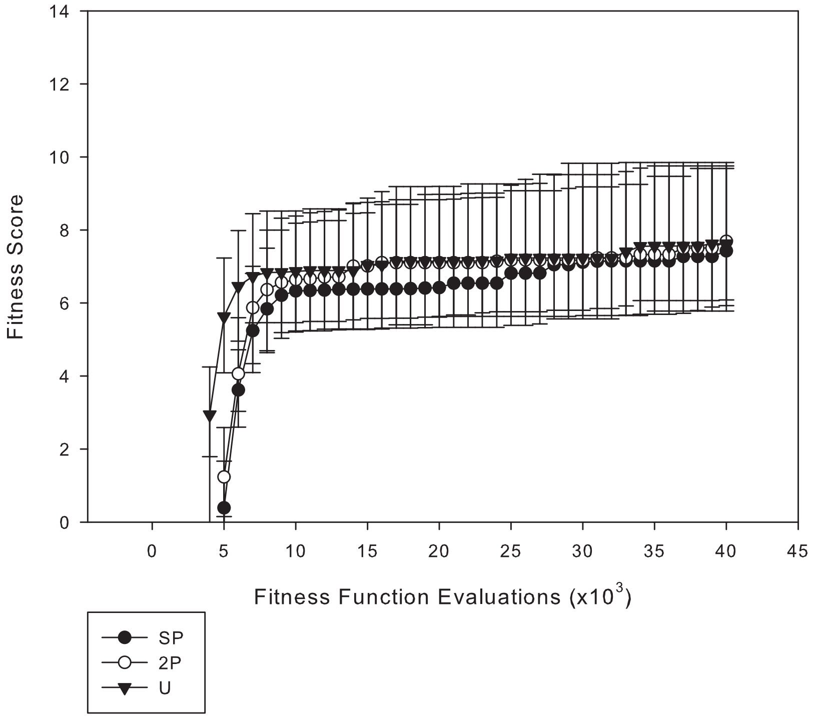


Figure 7.47: Results for $N = 100, p_c = 0.9, p_m = 0.005$ and $R = g$



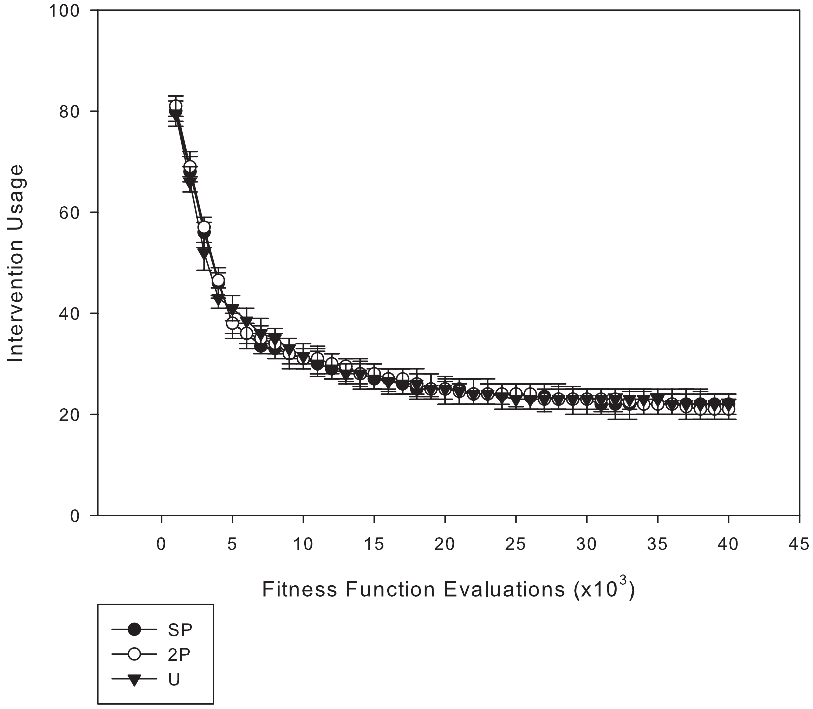
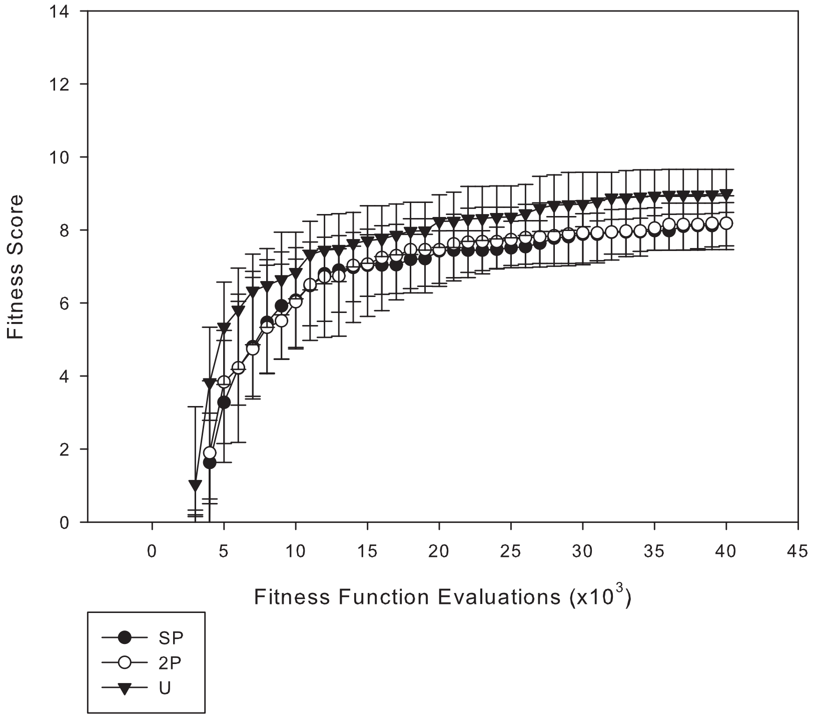
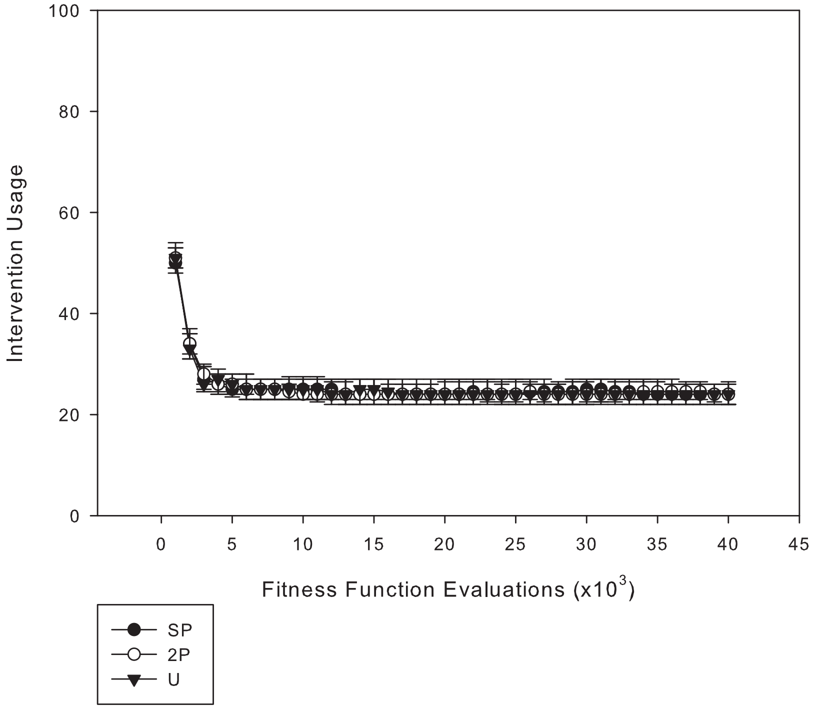


Figure 7.48: Results for $N = 100, p_c = 0.9, p_m = 0.05$ and $R = g$





Reviewing the intervention usage displayed in Figure 7.46, it would appear that UC is finding a different area of solutions when compared with SPC and 2PC when $p_m = 0$. UC is using around 68 interventions, with 2PC using around 80 and SPC requiring approximately 84 interventions. This reduction in intervention usage appears highly beneficial for schedule production, as shown with the associated fitness scores. As both 2PC and UC perform more recombination of genetic material, this would explain why both of these crossover techniques outperform SPC when $p_m = 0$.

Figure 7.47 shows the fitness scores and intervention usage when p_m is increased to 0.005. Unlike intervention usage for p_m of 0, each of the approaches use a similar number of interventions for p_m of 0.005. This may explain the large overlap between fitness scores returned by each of the crossover approaches.

Figure 7.48 details the effect of further increasing p_m to 0.05. As with the intervention usage and fitness scores for a p_m of 0.005, when p_m is 0.05 each approach uses a similar number of interventions, resulting in similar fitness scores across approaches. Unlike the bio-control experiment, this increased mutation level does not represent an upper bound, as the solutions produced when p_m is 0.05 are fitter than those returned with a mutation level of 0.005. This is in accordance with the findings of Petrovski et al [84], as their statistical evaluation of the significant GA factors found a mutation rate of 0.092 to be optimal.

From reviewing the traditional crossover approaches with intermediate parameter settings and a generational replacement strategy, two conclusions can be drawn. Firstly, in the absence of mutation, UC returns scores of considerably higher fitness than those found by both SPC and 2PC, with 2PC in turn producing fitter scores than SPC. The second conclusion drawn from these experiments is that when p_m is 0.005 or 0.05, each of the crossover approaches perform in a similar manner and thus the choice of traditional crossover approach is potentially arbitrary.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the traditional crossover approaches, under a generational replacement strategy for cancer chemotherapy scheduling. The average dosage strength and placement of interventions will now be reviewed. If the average dosage was 15 for an intervention, this would represent a maximum dosage by each of the 100 best solutions recorded at that intervention point. Conversely, if the average dosage is 0, this shows that none of the 100 best solutions apply drugs at that particular intervention.

In the case of no mutation (Figure 7.46), UC was shown to use less interventions than both SPC and 2PC, with 2PC in turn using less interventions than SPC. The average placement of these interventions at the end of the experiment, after 40,000 FFEs, are shown in Figure 7.49.

Figure 7.46 detailed the fitness scores returned by each of the crossover approaches, in the absence of mutation. The average placement and dosage strength detailed in Figure 7.49 show that UC applies less drugs than the other approaches and 2PC slightly less than SPC.

Figure 7.49: Intervention placement for 40,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$ and $R = g$

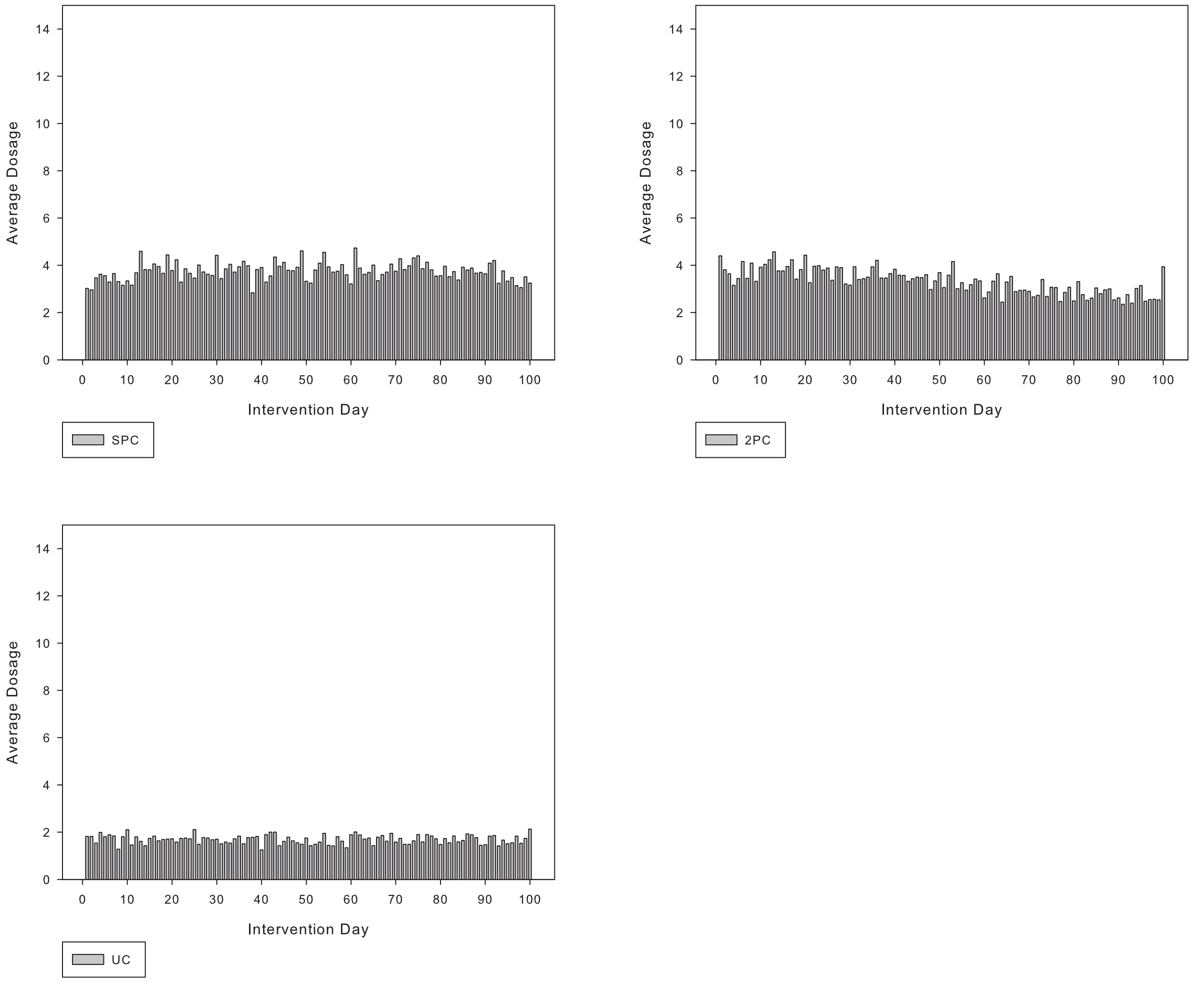


Figure 7.47 showed that each of the crossover approaches returned fitter scores, using fewer interventions, when p_m was increased from 0 to 0.005. Figure 7.49 shows the average placement and dosage strength of these applications. Each of the crossover approaches returned similar fitness scores for this mutation rate, and Figure 7.47 shows that each approach places interventions, in the same dosage range, on common days. This intensive therapy late in the treatment period is in line with the mathematical optimisation work of McCall et al [86].

Figure 7.50: Intervention placement for 40,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$ and $R = g$

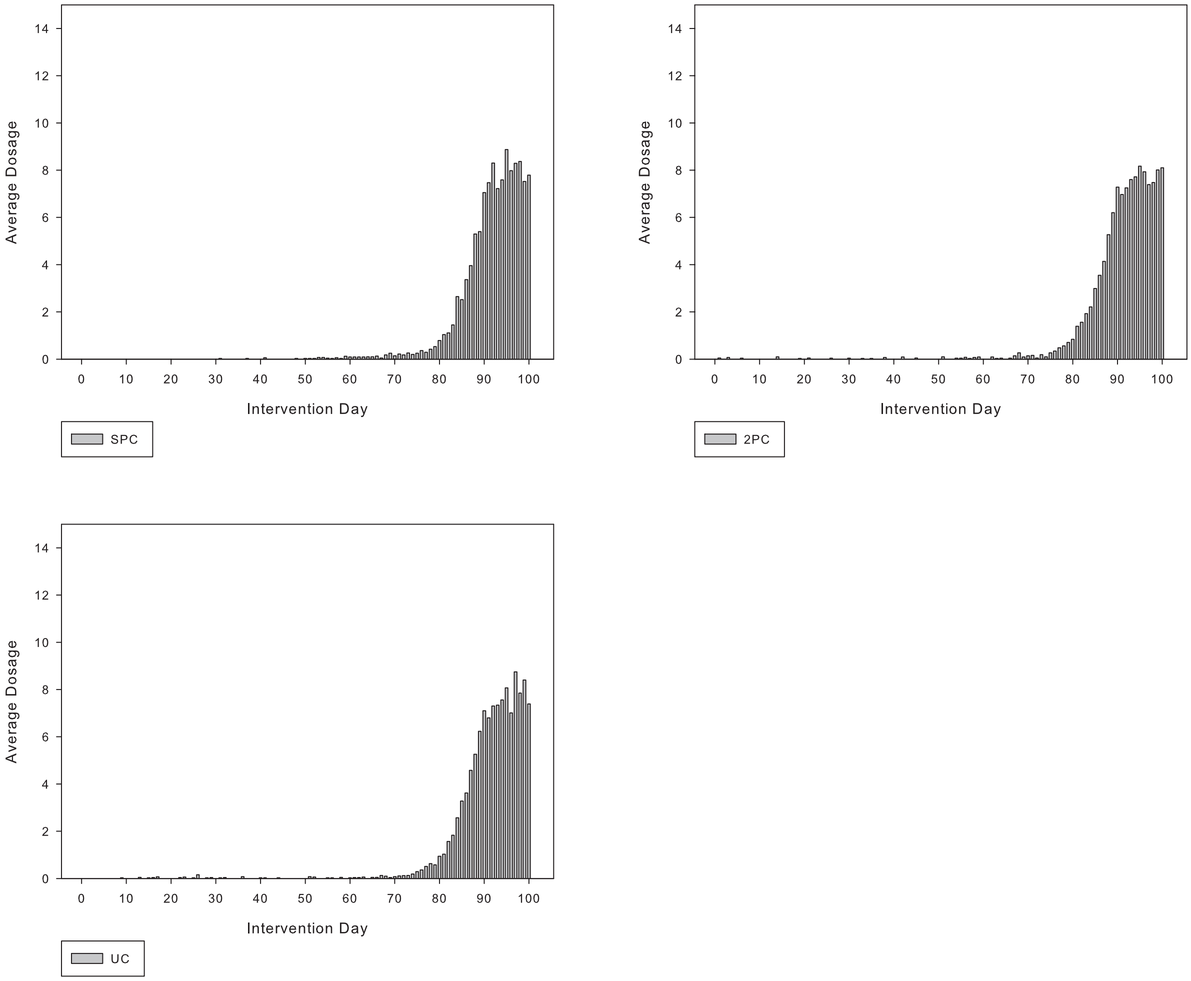
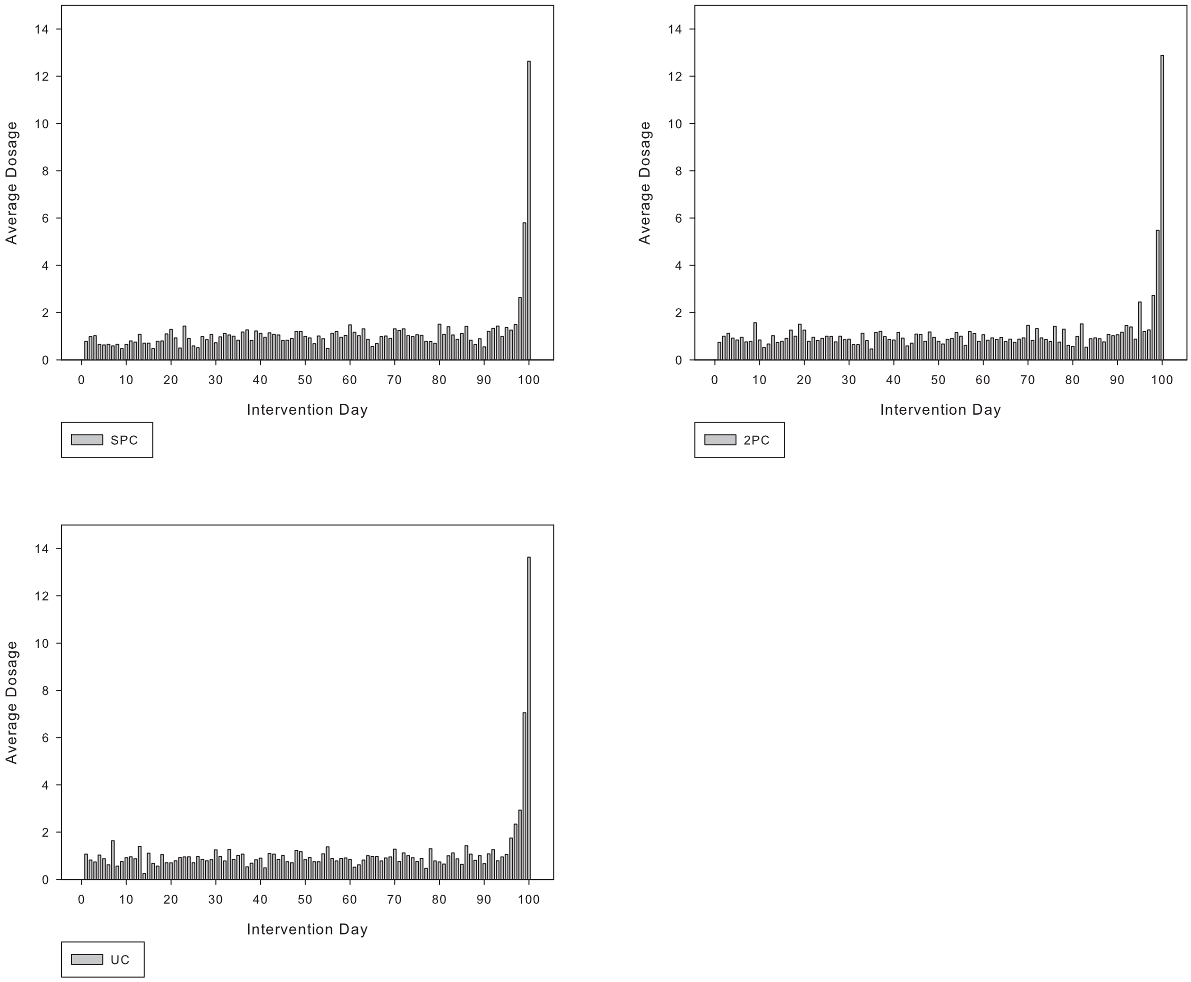


Figure 7.48 detailed the fitness scores returned and interventions used when p_m was increased from 0.005 to 0.05. As detailed previously, the scores returned for this increased mutation rate are fitter than those found when p_m was 0.005, with each of the crossover approaches returning scores of a similar fitness. Figure 7.51 shows that while on average each of the crossover approaches have small doses throughout the schedule, they also have a dose of approximately strength 3 on day 98, an increased dose of strength 5 on day 99 and a final dose on day 100 with strength 13.

Figure 7.51: Intervention placement for 40,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$ and $R = g$

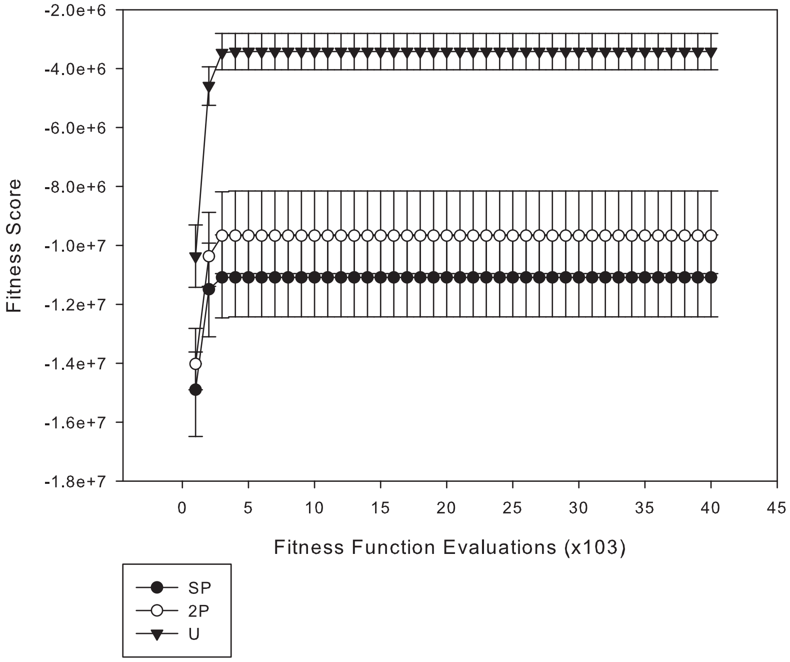


Steady state replacement

Figures 7.52 to 7.54 shows the steady state fitness scores for each approach with intermediate parameter settings. The trends shown with generational replacement are still apparent for steady state replacement. When p_m is 0, as shown in Figure 7.52, UC uses fewer interventions than both SPC and 2PC and returns fitter scores. As with the generational equivalent, 2PC outperforms SPC for this mutation level.

Figures 7.53 and 7.54 show p_m levels of 0.005 and 0.05 respectively. As with generational replacement, for each of these mutation levels each crossover approach uses a similar number of interventions for schedule optimisation. This similarity between intervention usage in schedules would account for the similar fitness scores displayed by each of the crossover approaches. Note that for these mutation levels, a steady state replacement strategy produces scores with fitter scores than with the generational replacement strategy.

Figure 7.52: Results for $N = 100, p_c = 0.9, p_m = 0$ and $R = ss$



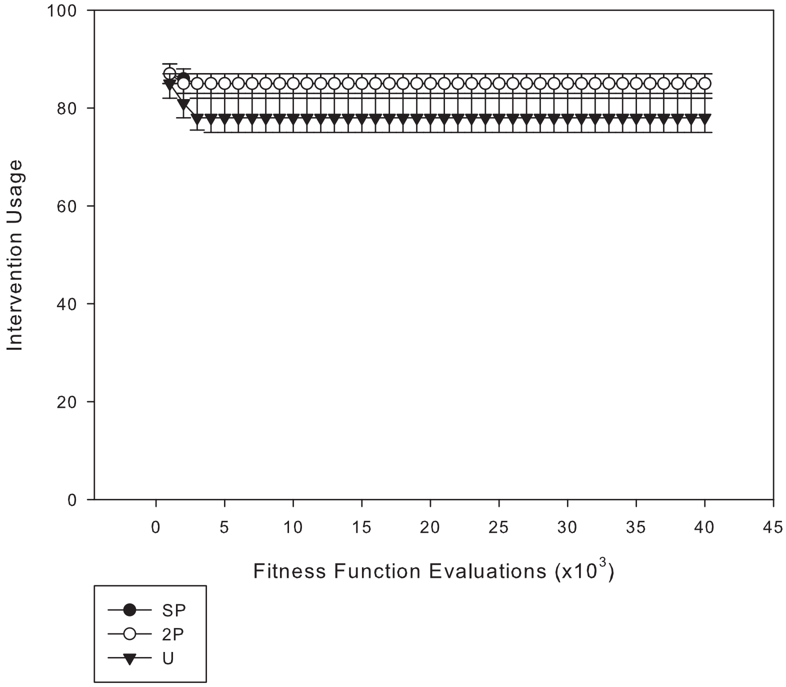
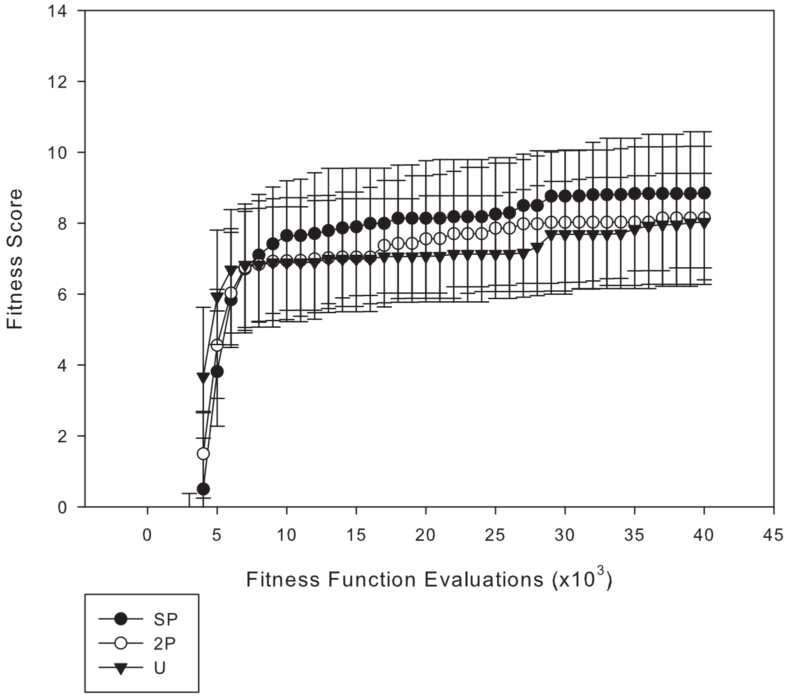


Figure 7.53: Results for $N = 100, p_c = 0.9, p_m = 0.005$ and $R = ss$



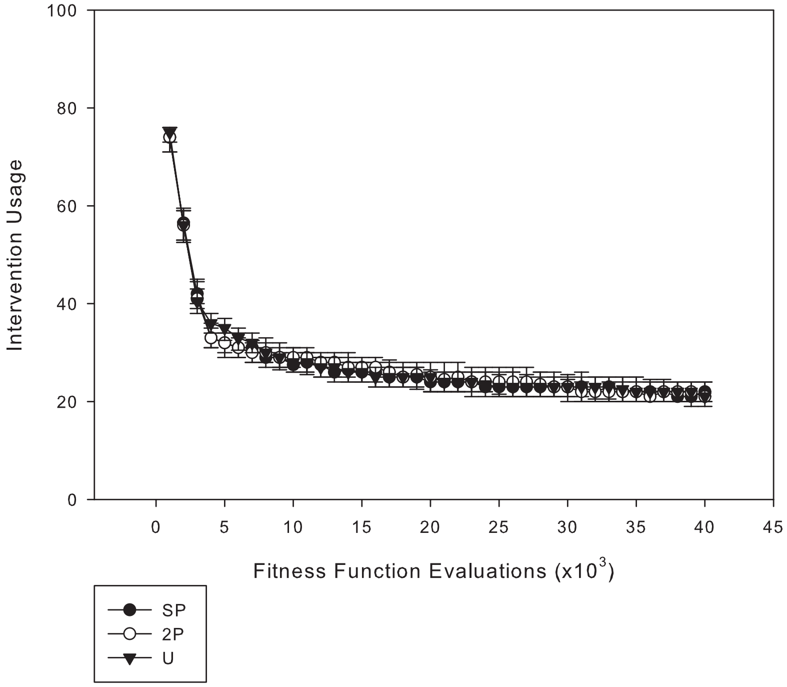
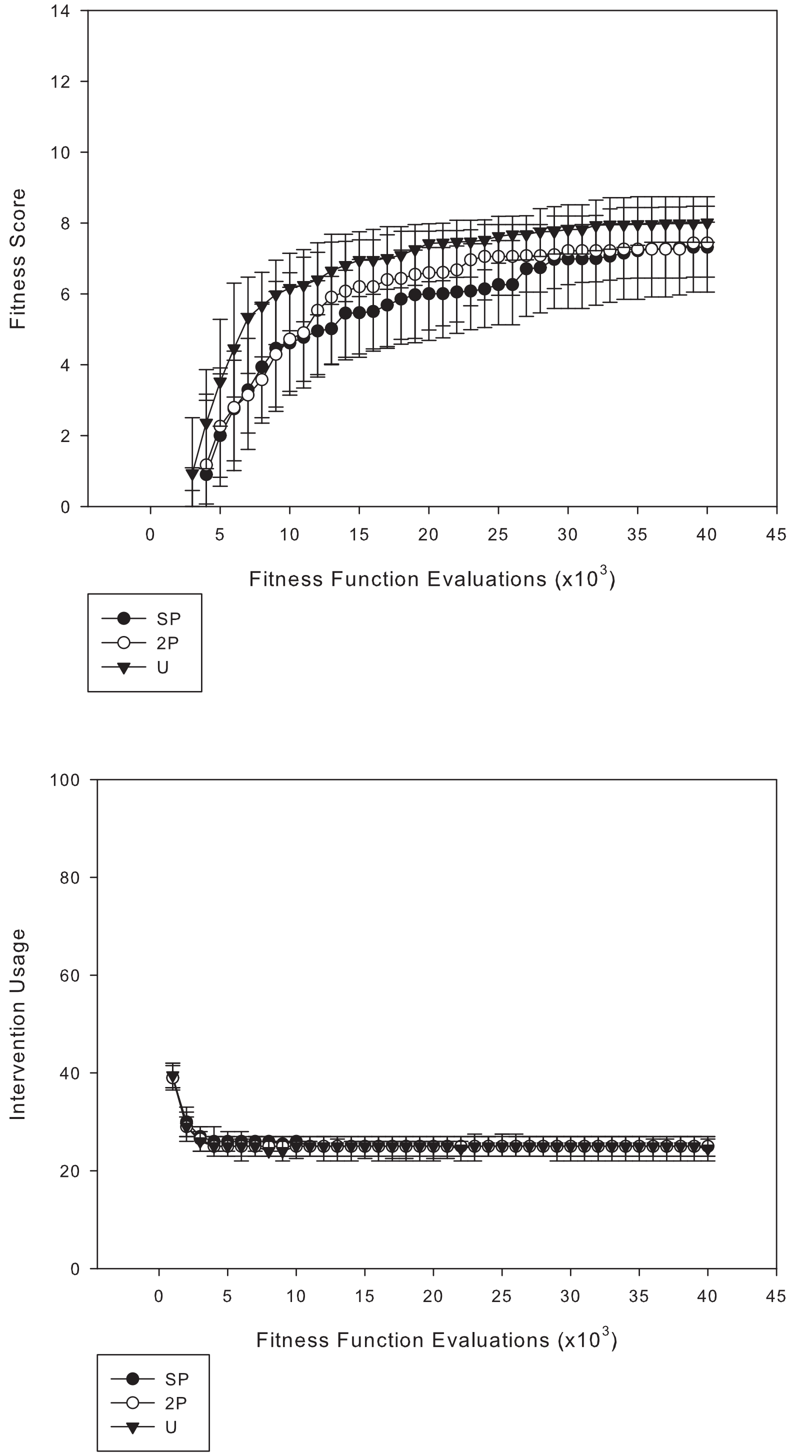


Figure 7.54: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$ and $R = ss$



7.2.1 Summary of experiment for chemotherapy treatment scheduling

This section has reviewed the abilities of SPC, 2PC and UC for deriving schedules for chemotherapy application, over a range of mutation rates and replacement strategies.

Each of the traditional crossover approaches are shown to perform poorly when there is no mutation introduced into the population. When p_m is 0, UC returns scores of a higher fitness than both SPC and 2PC under generational replacement. Under a steady state replacement strategy, UC returns scores of a higher fitness than both SPC and 2PC. The fitness scores returned by each of the crossover approaches are considerably poorer under a steady state replacement strategy. This shows that with a steady state replacement strategy and a p_m level of 0, a lack of genetic variety is being introduced to the population, resulting in scores of lesser fitness than a generational strategy which replaces most of the population in each generation.

Irrespective of replacement strategy, for p_m levels of 0.005 or 0.05, the choice of crossover approach is arbitrary as similar fitness scores are returned by each of the crossover approaches, using a similar number of FFEs to do so.

The intervention placement for each of the crossover approaches was shown to be similar for each of the mutation rates. In the absence of mutation, each approach had small dosages and when p_m was 0.005, each approach applied late in the schedule. For a further increase in p_m to 0.05, each approach had a number of smaller interventions as well as 3 successively larger applications on days 98, 99 and 100.

Over the range of p_m values, UC is always at least as good as the other traditional crossover approaches. In the absence of mutation, regardless of replacement strategy, UC returns scores of a higher fitness than both SPC and 2PC. For these reasons, when evaluating the directed intervention techniques for chemotherapy scheduling in Section 8.2, UC will be used as the baseline crossover operator for comparison.

7.2.2 Further analysis

The previous section reviewed SPC, 2PC and UC using the intermediate parameter settings. In order to further analyse the abilities of each of the crossover approaches, it appears necessary to review them across a range of parameters outwith the intermediate set discussed previously. Appendix A.2 details the abilities of SPC, 2PC and UC across the range of population sizes, crossover rates and mutation rates. This describes how in the absence of mutation, UC finds fitter scores than both SPC and 2PC and that when the mutation rate is increased to 0.005, all approaches find comparable solutions. This also describes how each of the crossover approaches return comparable scores when the mutation rate is increased further to 0.05.

7.3 Summary of the evaluation of traditional crossover approaches

This chapter has reviewed the abilities of SPC, 2PC and UC for deriving schedules of both bio-control treatments and chemotherapy drug treatments. Section 7.1 reviewed these traditional crossover techniques over a range of intervention penalties. Analysis of these experiments found that UC was consistently at least as good as the other crossover approaches when measured over a range of mutation levels. As detailed in Section 7.1.7, this trend also held when varying population sizes, mutation rates and crossover rates were analysed.

Similar analysis was undertaken to compare the abilities of the traditional crossover approaches for chemotherapy scheduling. Section 7.2 reviewed the approaches using the intermediate parameter settings and a range of mutation levels. From this experimentation, UC was shown to be better than both SPC and 2PC at finding fitter solutions in the absence of mutation. Each of the traditional crossover approaches were shown to return similar scores when the mutation rate was increased to 0.005 or 0.05. Further analysis, using a range of population sizes, mutation rates and crossover rates was conducted in Section 7.2.2. As with the intermediate parameter settings, UC was shown to be better at finding fitter scores than the other approaches in the absence of mutation, and with a mutation rate of 0.005 or 0.05, each of the approaches were comparable.

UC has been shown to be consistently at least as good as the other crossover approaches for both problems under review. Due to this, it will therefore be used as the benchmark crossover technique to compare the directed intervention techniques with.

Chapter 8

Evaluation of directed intervention crossover approaches

Chapter 7 reviewed the abilities of SPC, 2PC and UC for both bio-control and chemotherapy schedule optimisation. For both problems, UC was always at least as good as the other approaches and therefore was determined to be the benchmark crossover technique with which to compare the directed intervention crossover approaches.

This chapter compares the directed intervention techniques of CalEB, TInSSel and DUC, described in Chapter 4, against UC for both bio-control and chemotherapy scheduling. One difference to the experiments undertaken in Chapter 7 is that statistical analysis will be carried out for each experiment, as described in Section 6.9. This statistical test is for ascertaining whether there is a statistically significant advantage in using the directed intervention crossover techniques over the standard UC approach. In accordance with the findings discussed in Section 6.9.1, these statistical tests will be non-parametric, as normality cannot be assumed.

The following sections review the abilities of CalEB, TInSSel and DUC, in comparison with UC, for deriving bio-control schedules for mushroom farming. The subsequent section will then review the performance of these same algorithms for the construction of single drug treatment schedules for chemotherapy.

As described in Section 4.1, DUC is a test approach to confirm that a range of offspring intervention levels provides a more efficient search than purely adhering to the number of interventions in the fitter parent. The following experiments should therefore show that both TInSSel and CalEB are more efficient than DUC at deriving schedules of both bio-control and chemotherapy treatments.

8.1 Evaluation of directed intervention crossover techniques for bio-control scheduling

For this evaluation, the same parameter configuration is used as for the traditional crossover experiments detailed in Section 7.1. The bio-control problem is therefore sampled over 5,000 fitness function evaluations (FFEs), at intervals of 100, resulting in 50 data points for review. For each sample point, 100 runs of each crossover approach were recorded for each parameter combination, with the best scoring values for each run for each FFE being recorded.

Each of the following sections review the effectiveness of CalEB, TInSSel and DUC, with comparison to UC, for bio-control scheduling over a range of penalty values per intervention. As with the traditional crossover experiments detailed in Chapter 7, the crossover techniques are tested over a range of parameter settings. A complete set of results across the parameter combinations has therefore been collected and these can be seen in Appendix C. This Appendix contains the complete set of results showing the fitness scores against FFEs associated with these experiments across a range of population sizes, intervention penalties and crossover and mutation rates.

As with the traditional crossover experiments, the intermediate parameter settings for each approach will be reviewed in detail, rather than explicitly detailing each possible parameter combination. The following sections will therefore review CalEB, TInSSel and DUC in comparison to UC, for the varying levels of intervention penalty P, as described in Section 5.3.2, with a population of $N = 100$, crossover rate, $p_c = 0.9$, and mutation rates, p_m, of 0, 0.005 and 0.05. These results can then be compared against the complete set of results in Appendix C to check for consistency.

Inline with the traditional crossover experiments, initial analysis for each intervention point will review the intermediate parameter settings with a generational replacement strategy. The average placement of interventions will then be analysed. Intermediate parameter settings will then be reviewed under a steady state replacement strategy. As with previous experiments, the replacement mechanism, R, for each experiment will be defined as either generational, $R = g$, or steady state, $R = ss$.

The results for both the fitness scores and associated intervention usage for the intermediate parameter settings under both generational and steady state replacement will now be reviewed. As before, graphs showing the fitness scores versus FFEs and intervention usage versus FFEs are plotted over the range of FFE samples. Note that as the bio-control optimisation problem is a minimisation problem, lower scores are fitter than higher valued scores. On the graphs, U, C, T and D represent UC, CalEB, TInSSel and DUC respectively.

8.1.1 0 penalty points per intervention

This section reviews CalEB, TInSSel and DUC, in comparison with UC, for bio-control scheduling with a penalty of 0 points per intervention. As described in Section 8.1, this will focus on the intermediate parameter settings. Figures 8.1 to 8.3 shows the fitness scores and associated intervention usage for p_m levels of 0, 0.005 and 0.05 respectively.

Figure 8.1: Results for $N = 100, p_c = 0.9, p_m = 0, P = 0$ and $R = g$

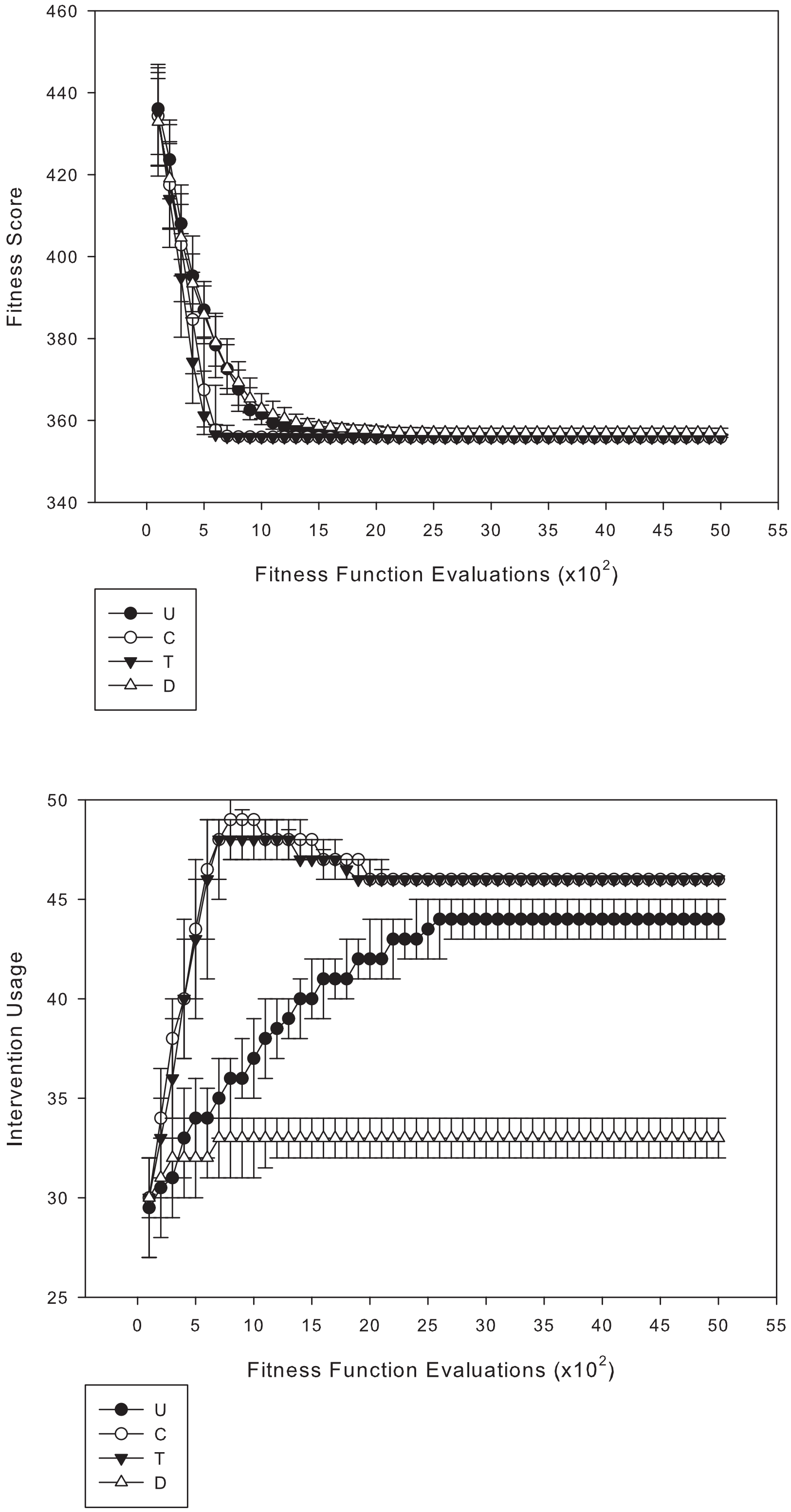


Figure 8.2: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 0$ and $R = g$

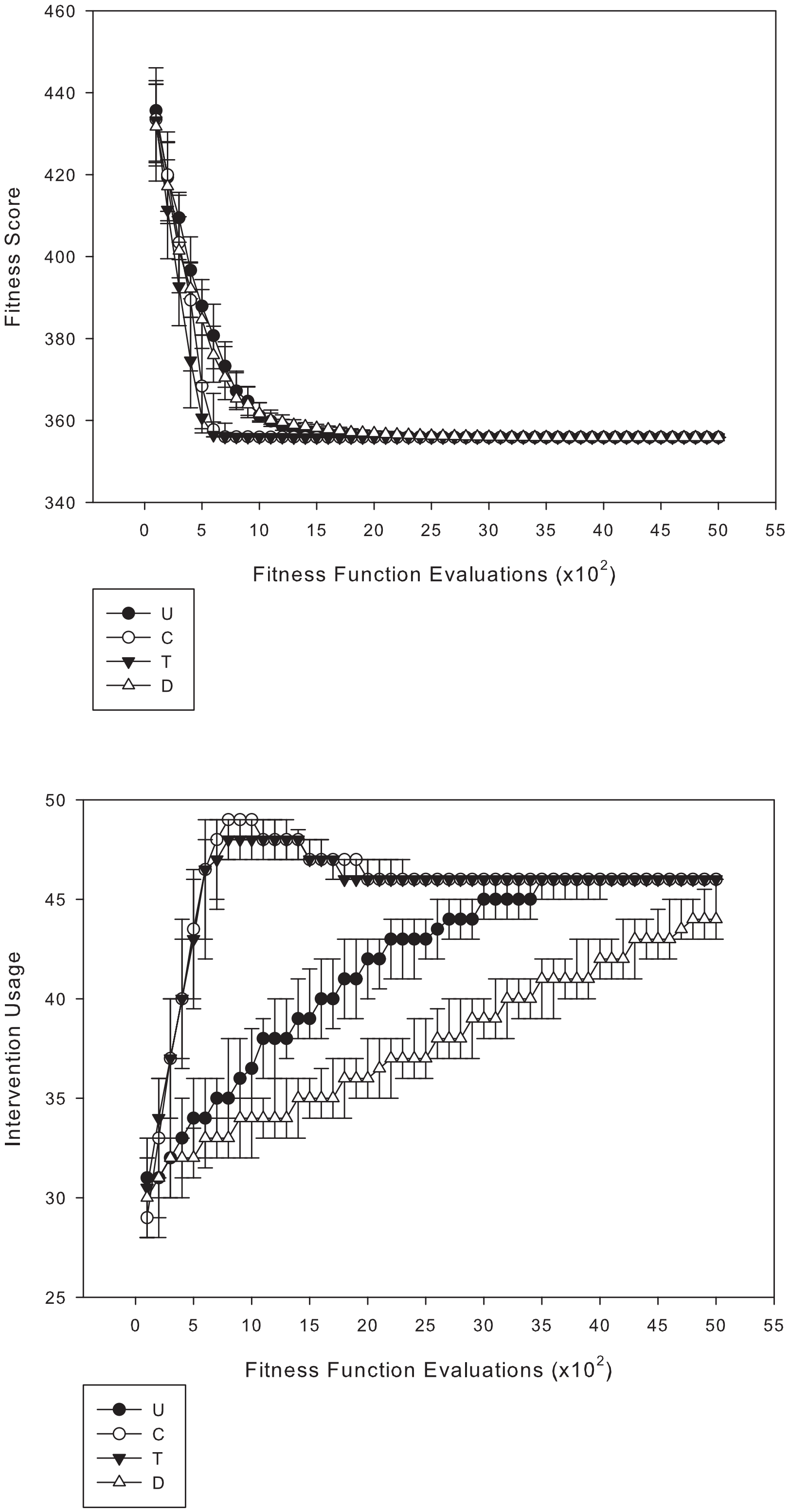
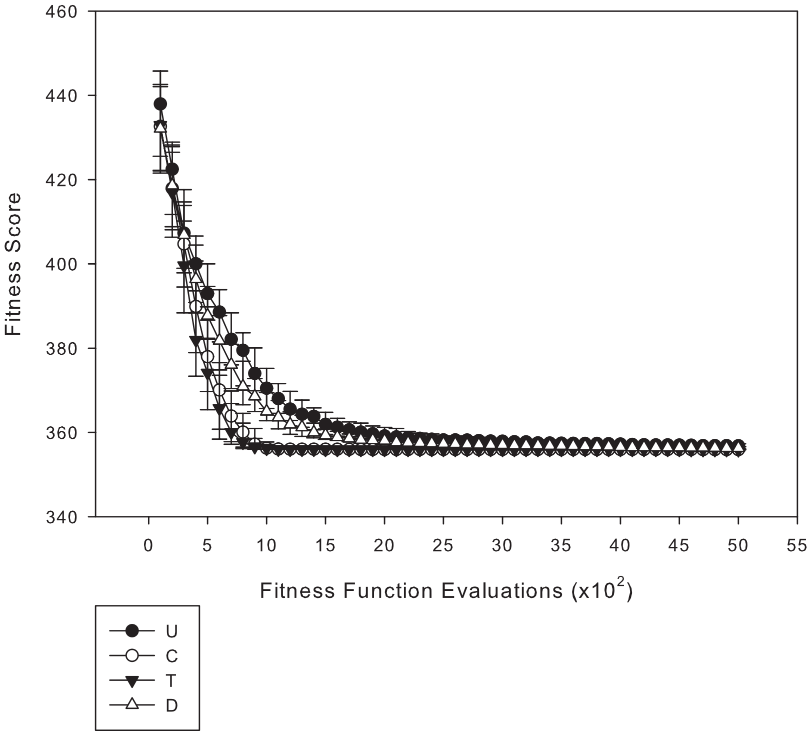
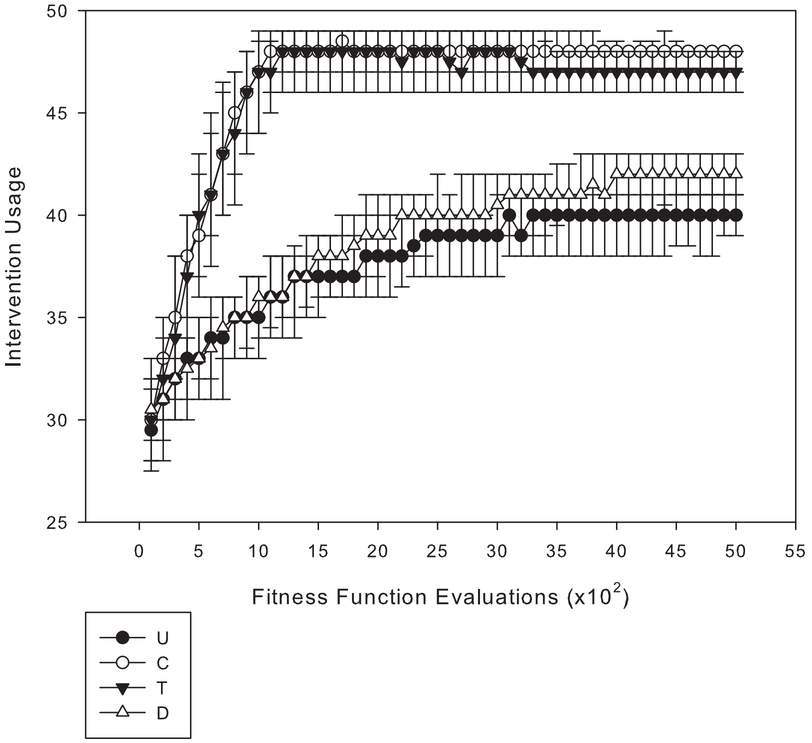


Figure 8.3: Results for $N = 100, p_c = 0.9, p_m = 0.05, P = 0$ and $R = g$





Reviewing Figure 8.1, a correlation appears between CalEB and TInSSel's rapid increase in fitness with the accumulation of interventions. This shows that, in the absence of mutation, while UC steadily gathers interventions, and DUC converges to using around 33 interventions at around 1,000 FFEs, both CalEB and TInSSel rapidly gain interventions, before settling to schedules utilising approximately 45 interventions. This rapid accumulation of interventions match the rapid increase in fitness demonstrated by CalEB and TInSSel.

Figure 8.2 details the fitness scores and intervention usage when p_m is increased to 0.005. All crossover approaches end the optimisation process using approximately the same number of interventions and returning scores in a similar fitness range. As with a p_m level of 0, CalEB and TInSSel rapidly accumulate interventions, whereas UC intervention usage grows steadily, finally settling to the same level as both CalEB and TInSSel. DUC is much slower than the other approaches at increasing intervention usage, but after 5,000 FFEs has increased in intervention usage to similar levels to that of UC, CalEB and TInSSel. As was demonstrated in the experiment with no mutation, a correlation appears between rapid accumulation of interventions for schedules and speedy increase in fitness for CalEB and TInSSel.

Figure 8.3 shows the effects of further increasing p_m to 0.05. For this mutation level two distinct intervention usage patterns appear. CalEB and TInSSel rapidly increase to approximately 48 interventions, whereas both UC and DUC gradually increase the number of interventions to around 39.

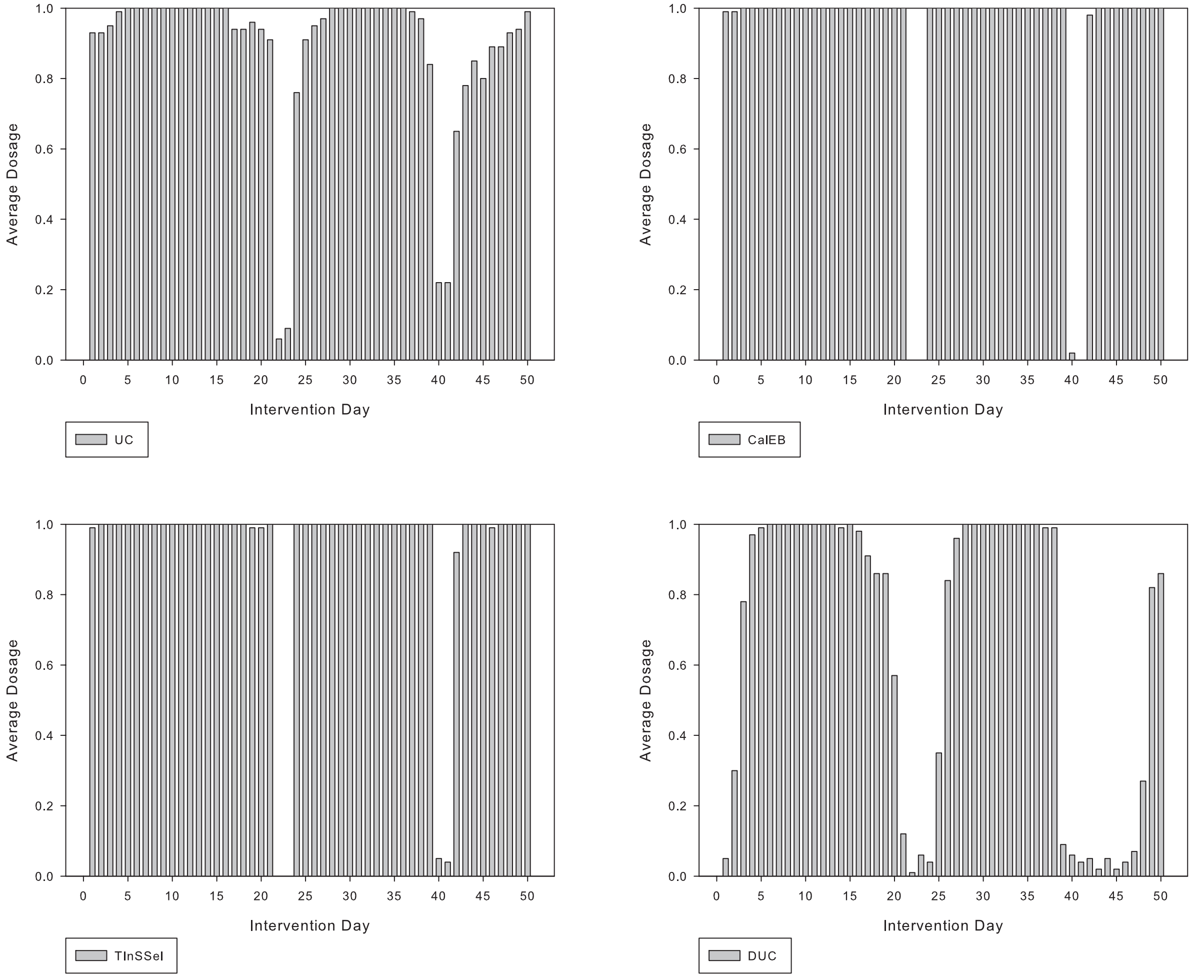
It would appear that the rapid improvement in fitness scores shown by CalEB and TInSSel is directly related to their rapid accumulation of interventions. In this experiment, as each crossover approach settles on higher number of interventions, due to there being no cost associated with intervening, fitness scores in a similar range are produced for all crossover approaches under review.

Intervention placement

Section 8.1.1 outlined the fitness scores and associated intervention usage for the directed intervention crossover approaches, in comparison to UC, under a generational replacement strategy for a penalty of 0 points per intervention.

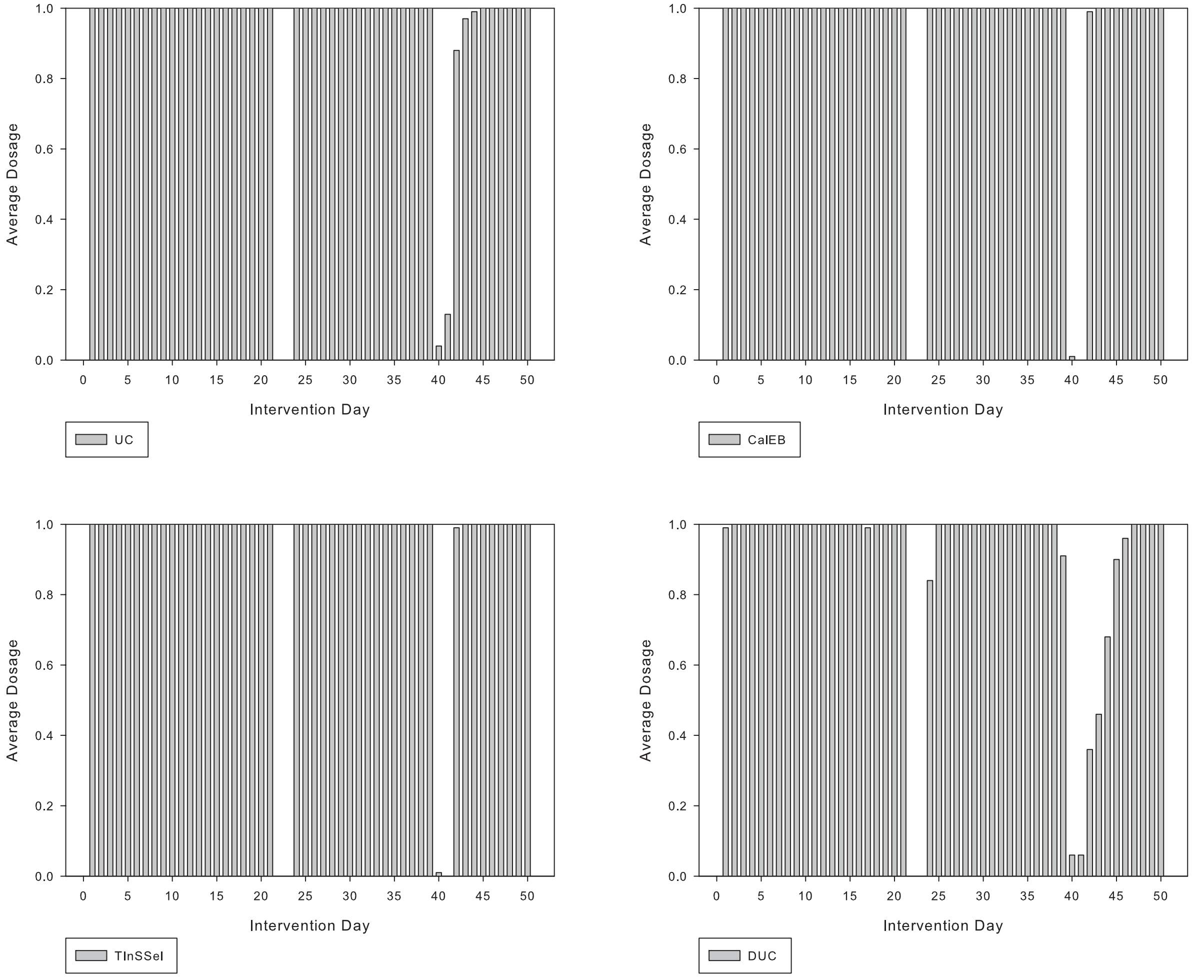
In the case of no mutation, as shown in Figure 8.1, CalEB and TInSSel are shown to use more interventions than both UC and DUC, with UC using more interventions than DUC. From the statistical analysis undertaken in Appendix E.1, after 5,000 FFEs both CalEB and TInSSel were shown to outperform UC and UC, TInSSel and CalEB were shown to outperform DUC. This section reviews the placement of interventions by each of the crossover approaches at 5,000 FFEs. The average placement of these interventions are shown in Figure 8.4. This shows that both CalEB and TInSSel are focusing interventions on days 2 - 21, 24 - 39 and 42 - 50. As UC is not as focussed on each of these days, this accounts for the lesser fitness scores returned by this approach. When UC, CalEB and TInSSel are compared to DUC, DUC is shown to have less focussed intervention placement than the other approaches, resulting in lesser fitness scores.

Figure 8.4: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 0$ and $R = g$



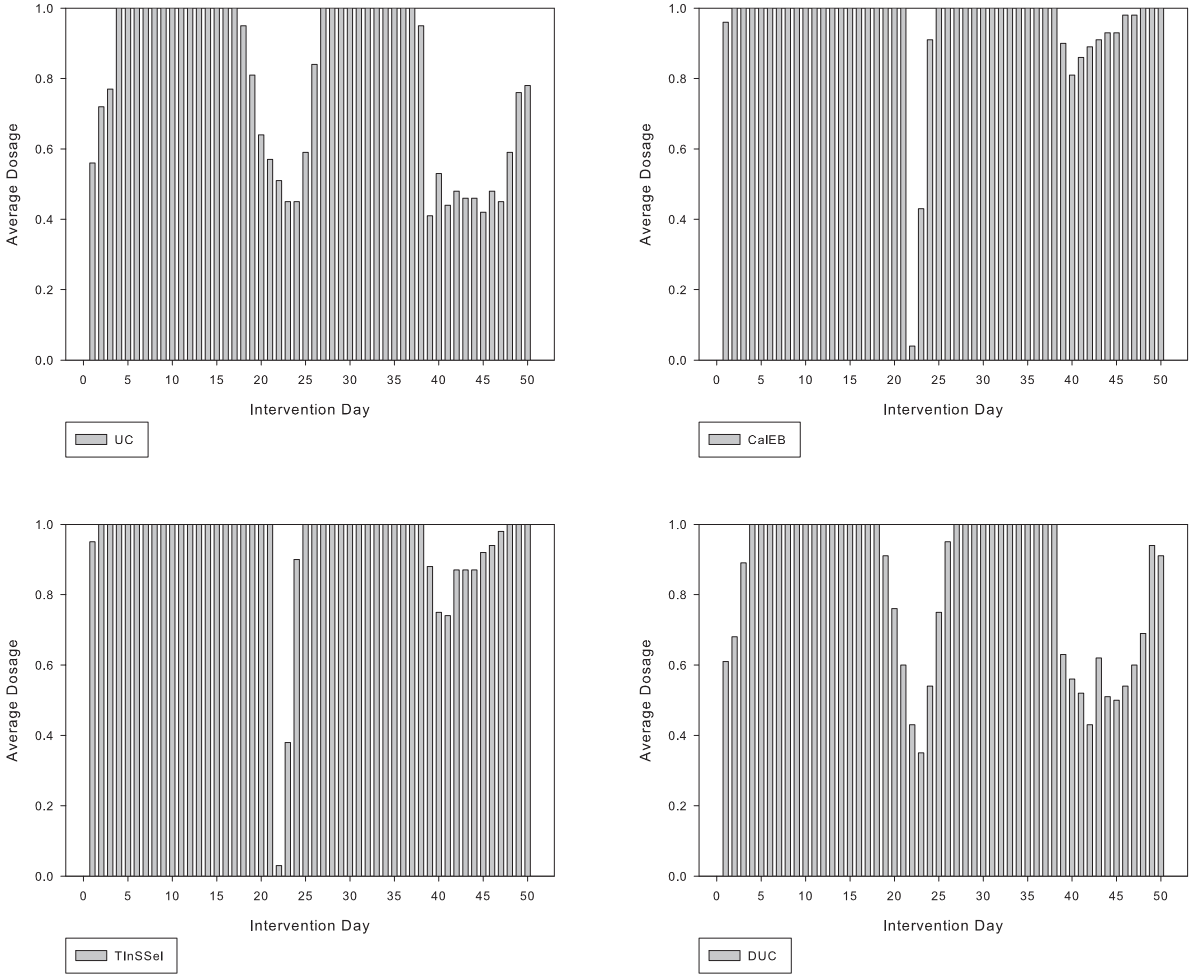
When p_m was increased to 0.005, as shown in Figure 8.2, CalEB, TInSSel and UC are shown to use more interventions than DUC. From the statistical analysis undertaken in Appendix E.2, after 5,000 FFEs, UC, CalEB and TInSSel were shown to outperform DUC. The average placement of interventions are shown in Figure 8.5. This shows that while UC, CalEB and TInSSel have three distinct application areas, DUC targets two of these but is less focussed on the third (days 42 - 50).

Figure 8.5: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 0$ and $R = g$



When p_m was further increased to 0.05, as shown in Figure 8.3, CalEB used the most interventions, followed by TInSSel, then DUC then UC. From the statistical analysis undertaken in Appendix E.3, after 5,000 FFEs, CalEB, TInSSel and DUC were all shown to outperform UC, with both CalEB and TInSSel outperforming DUC. The average placement of interventions for each of the crossover approaches are shown in Figure 8.6. CalEB and TInSSel are shown to regularly intervene on more days than both UC and DUC, with DUC on average, intervening on more days than UC.

Figure 8.6: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 0$ and $R = g$



Steady state replacement

Section 8.1.1 reviewed CalEB, TInSSel and DUC, in comparison with UC, for bio-control scheduling, with an intervention penalty of 0 per intervention and generational replacement. The experiments undertaken in Chapter 7.1, outlined the importance of reviewing the crossover approaches under both generational and steady state replacement strategies. As with the generational equivalent described in Section 8.1.1, this experiment details the performance of approaches under steady state replacement with intermediate parameter settings. Figures 8.7 to 8.9 show the fitness and intervention usage associated with performing the previous set of tests with a steady state replacement strategy.

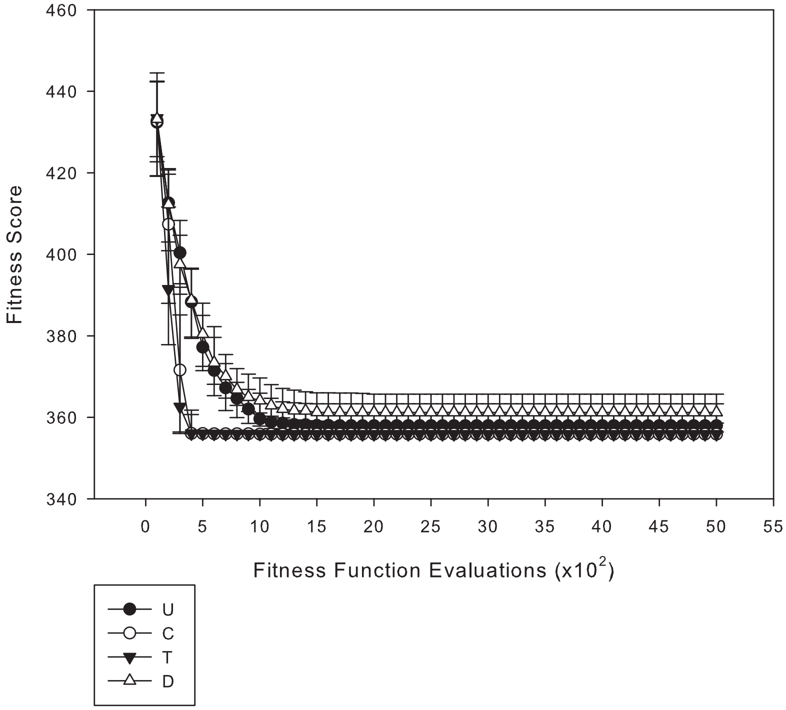
Figure 8.7 shows that similar trends are displayed for steady state replacement as those shown by generational replacement for similar parameter settings when p_m is 0. CalEB and TInSSel are again quicker than UC or DUC at deriving fit solutions. The intervention usage follows a similar pattern to that observed under generational replacement. CalEB and TInSSel rapidly increase the number of interventions in the schedules, with UC increasing intervention levels at a lesser rate. As with generational replacement, DUC quickly converges to around 34 interventions and shows no further increase in intervention levels throughout the rest of the experiment.

Although all the approaches perform in a similar manner with steady state replacement compared to generational for a p_m of 0, there is one exception. When p_m is 0, using a steady state replacement strategy, there is very little new material being added at each generation, and although UC, CalEB and TInSSel are able to deal with this condition, DUC returns scores that are less fit than those observed with the generational equivalent.

Figure 8.8 shows the fitness scores and intervention usage for a steady state replacement strategy when $p_m = 0.005$. As with a p_m of 0, each of the approaches behave in a similar way under steady state replacement as for generational replacement. One exception is that DUC is quicker at accumulating interventions under steady state replacement than under the generational version for these parameter settings. One possible explanation for this is that under a steady state replacement strategy, offspring can be used immediately as part of the breeding pool. This has been shown to make a shift towards the optimal solution in an early part of the optimisation process [45] and could account for the enhanced speed of intervention accumulation displayed by DUC.

It can be seen that whether under steady state replacement (Figure 8.9) or generational replacement (Figure 8.3), all crossover approaches behave in a similar manner when $p_m = 0.05$.

Figure 8.7: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$,$P = 0$ and $R = ss$



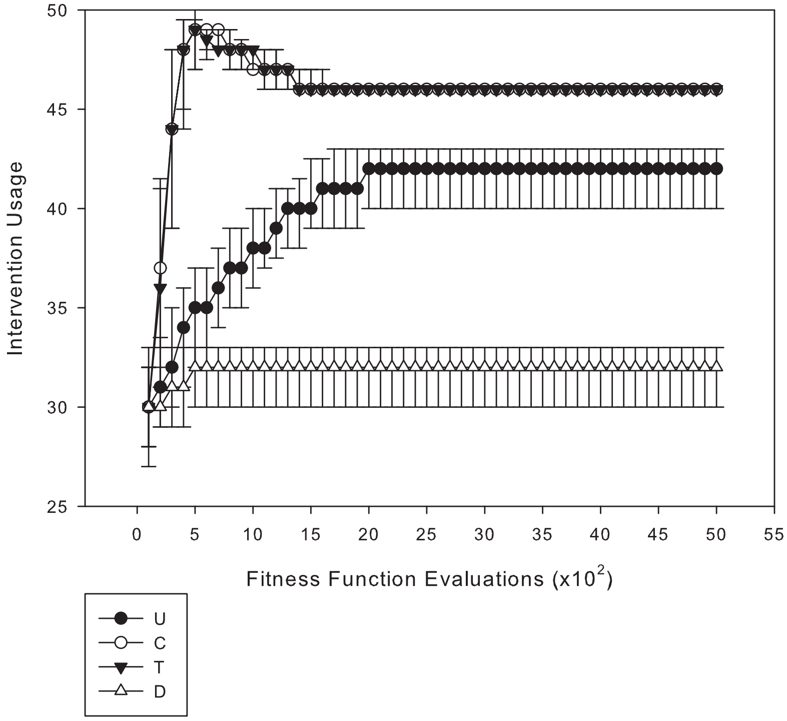


Figure 8.8: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 0$ and $R = ss$

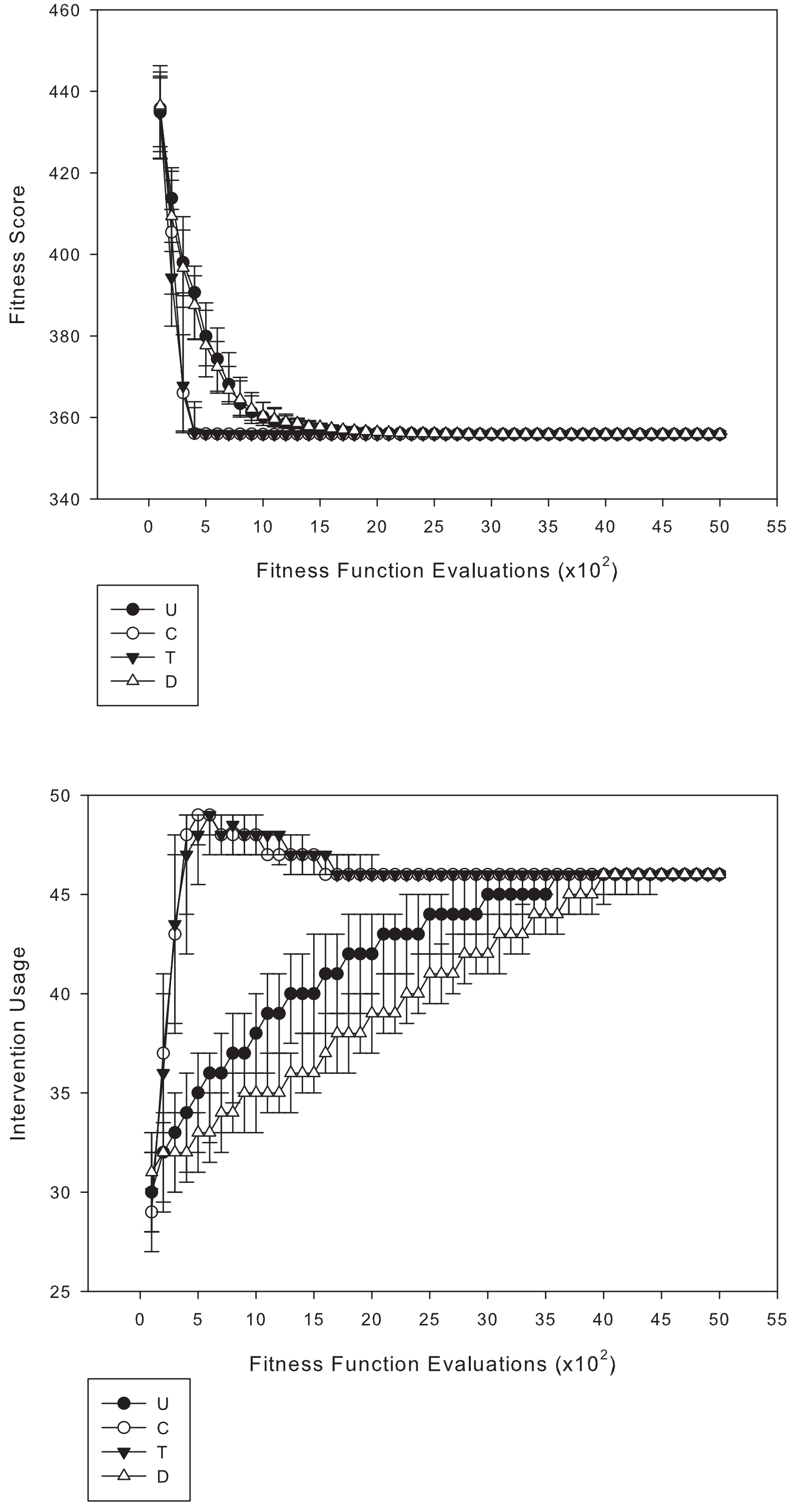
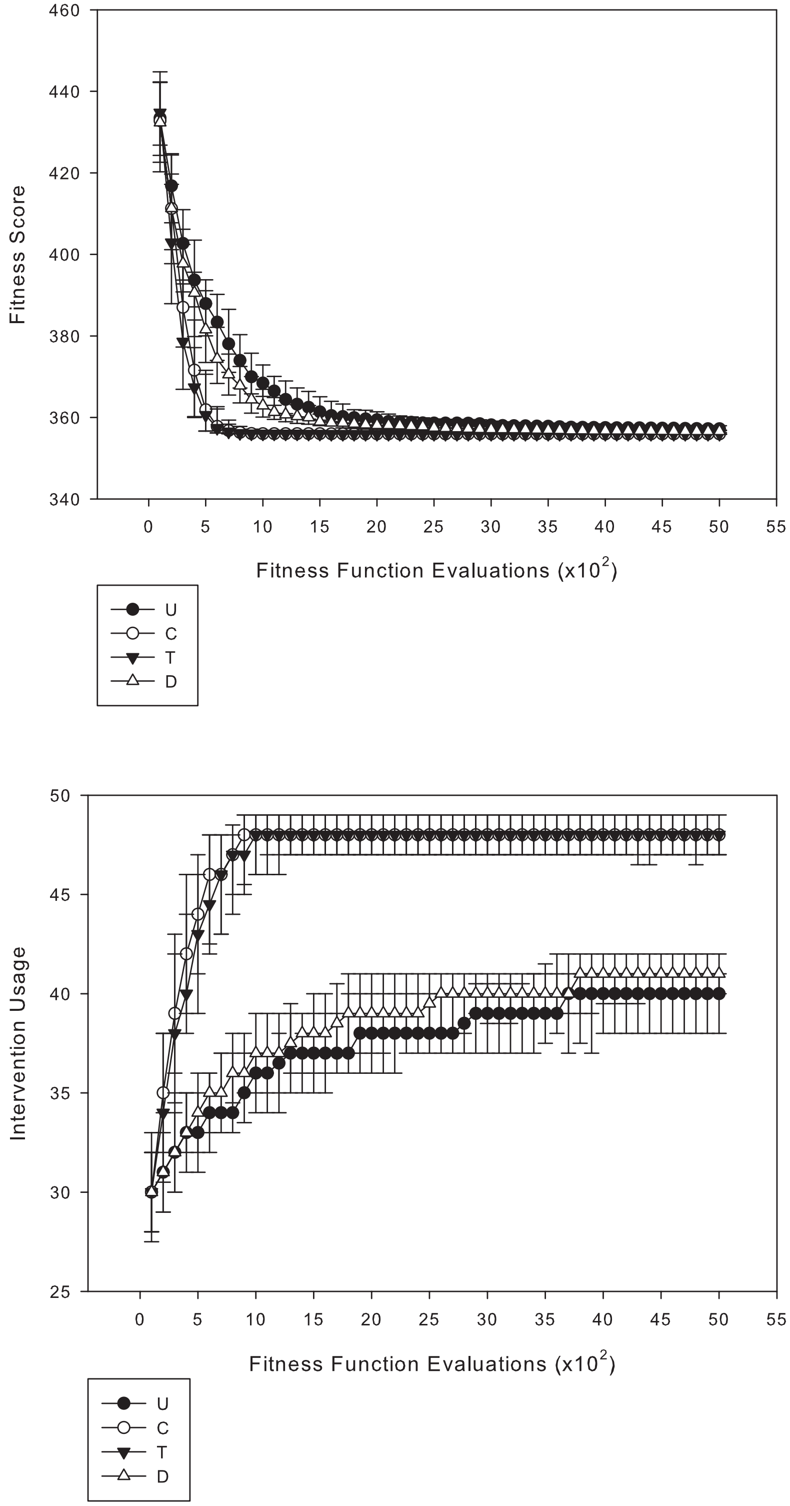


Figure 8.9: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 0$ and $R = ss$

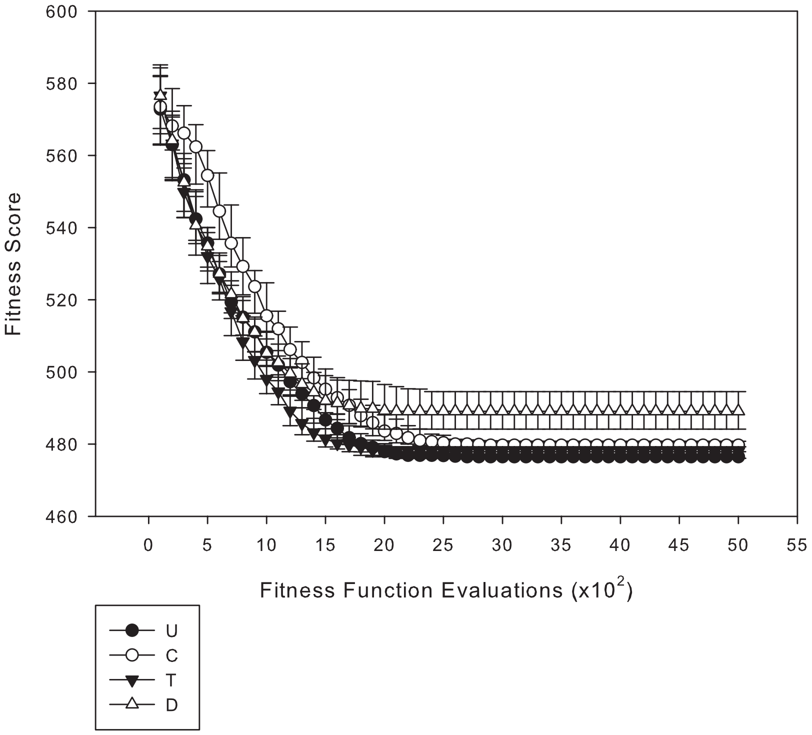


8.1.2 5 penalty points per intervention

The following section investigates how CalEB, TInSSel and DUC compare to UC at bio-control scheduling with a penalty of 5 points per intervention. As with the experiment detailed in Section 8.1.1, intermediate parameters of $N = 100$, and $p_c = 0.9$ will be used.

Figure 8.10 shows the fitness scores and associated intervention usage for a mutation rate p_m, of 0. Figures 8.11 and 8.12 details the fitness and intervention usage for p_m levels of 0.005 and 0.05 respectively. As with the earlier experiments described in Section 7.1, when the intervention penalty is increased from 0 to 5 points per intervention, the strategy changes from that of rapidly accumulating interventions to a more controlled, parsimonious approach of intervention selection.

Figure 8.10: Results for $N = 100, p_c = 0.9, p_m = 0, P = 5$ and $R = g$



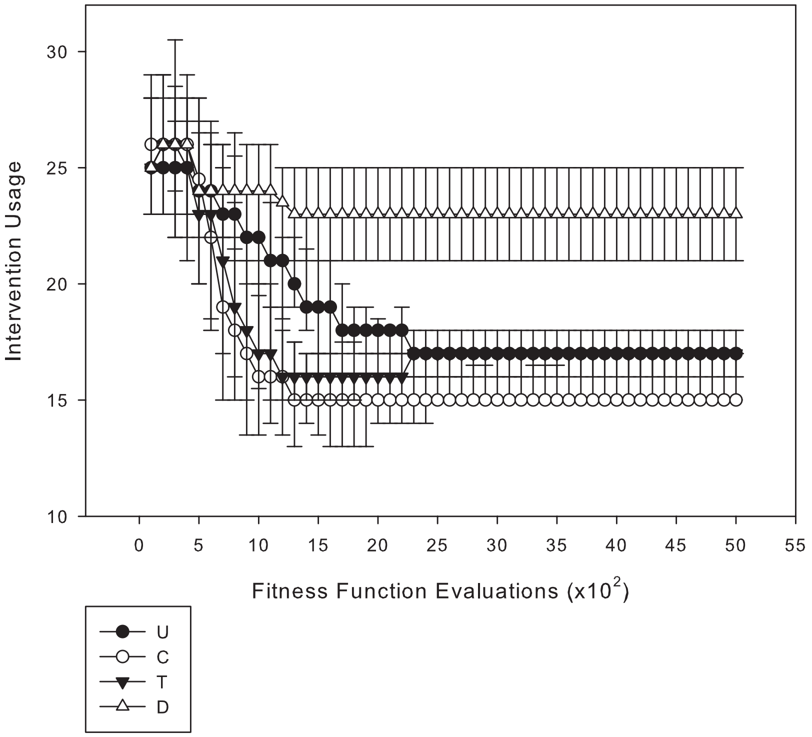
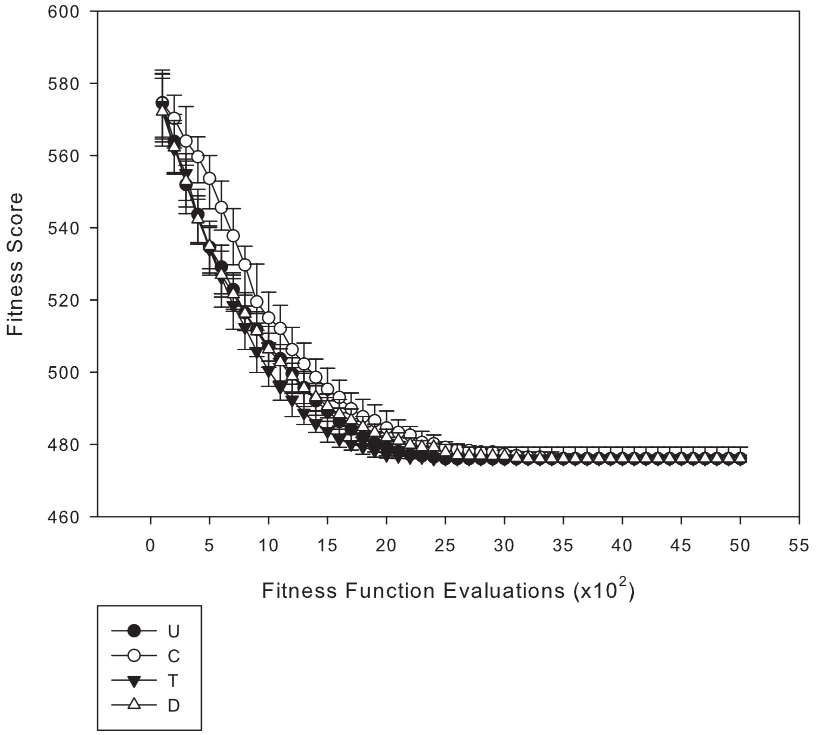


Figure 8.11: Results for $N = 100, p_c = 0.9, p_m = 0.005, P = 5$ and $R = g$



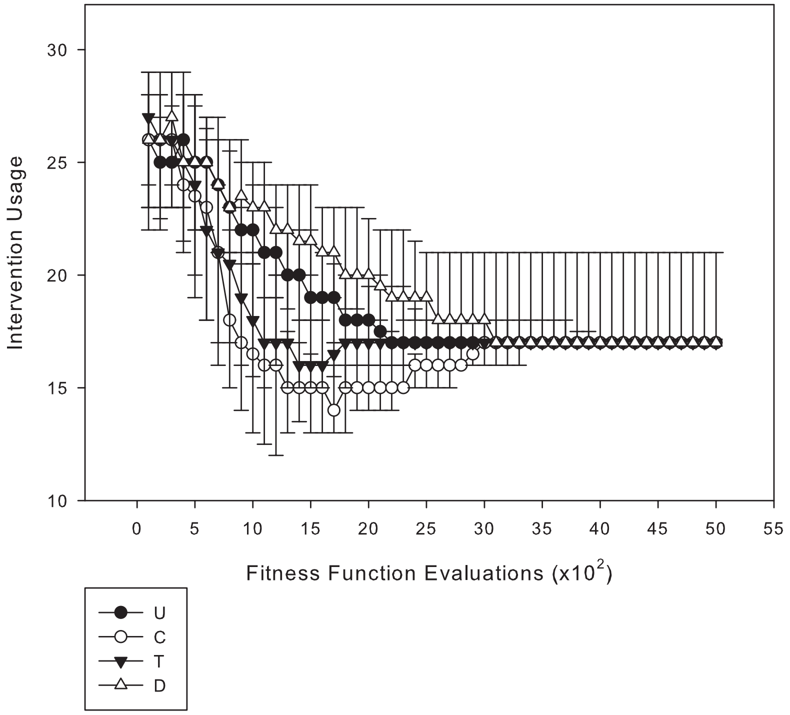


Figure 8.12: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 5$ and $R = g$

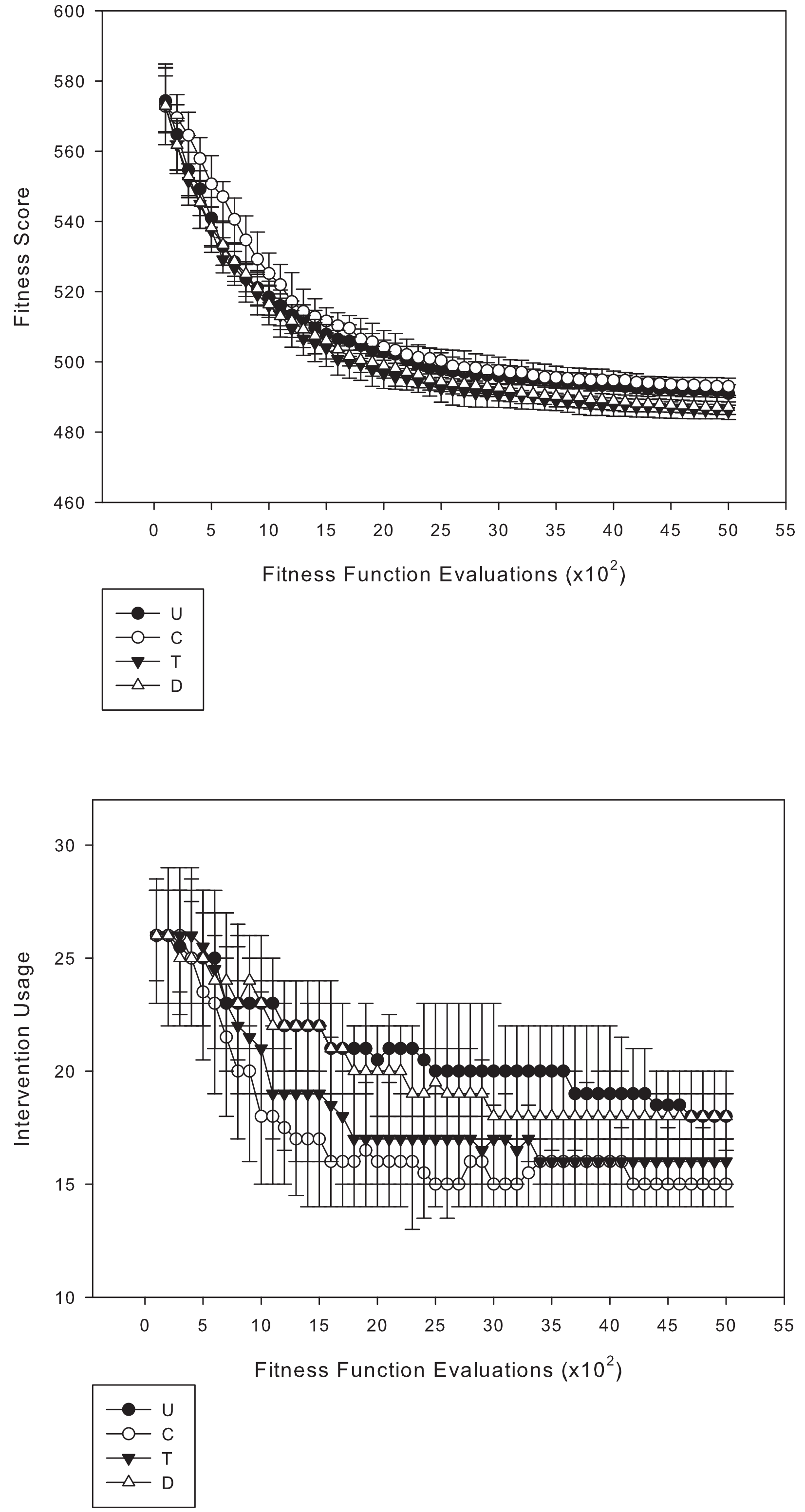


Figure 8.10 shows the fitness scores and intervention usage for the intermediate parameter settings with a penalty of 5 points per intervention and p_m of 0. In the absence of mutation, all approaches reduce in intervention

usage. CalEB and TInSSel rapidly reduce, with CalEB settling on less interventions than TInSSel. UC gradually reduces to a similar level of interventions as TInSSel, while DUC converges to a higher intervention usage than the other approaches. The associated fitness score graph shows that while UC, CalEB and TInSSel reduce to use fewer interventions than DUC, this provides fitter scores. As described in Section 4.2, both CalEB and TInSSel use the same intervention selection calculation. Thus for the first 1,000 FFEs, although both CalEB and TInSSel use the same number of interventions, they return scores of a different fitness. This shows that stochastic selection is more advantageous to a uniform distribution over time for this particular experiment.

Figure 8.11 shows the fitness scores and intervention usage when the mutation rate, p_m, is increased to 0.005. In this instance, DUC is slower at reducing the number of interventions compared to the other approaches. After around 3,000 FFEs however, all approaches have converged to approximately 17 interventions and return similar fitness scores.

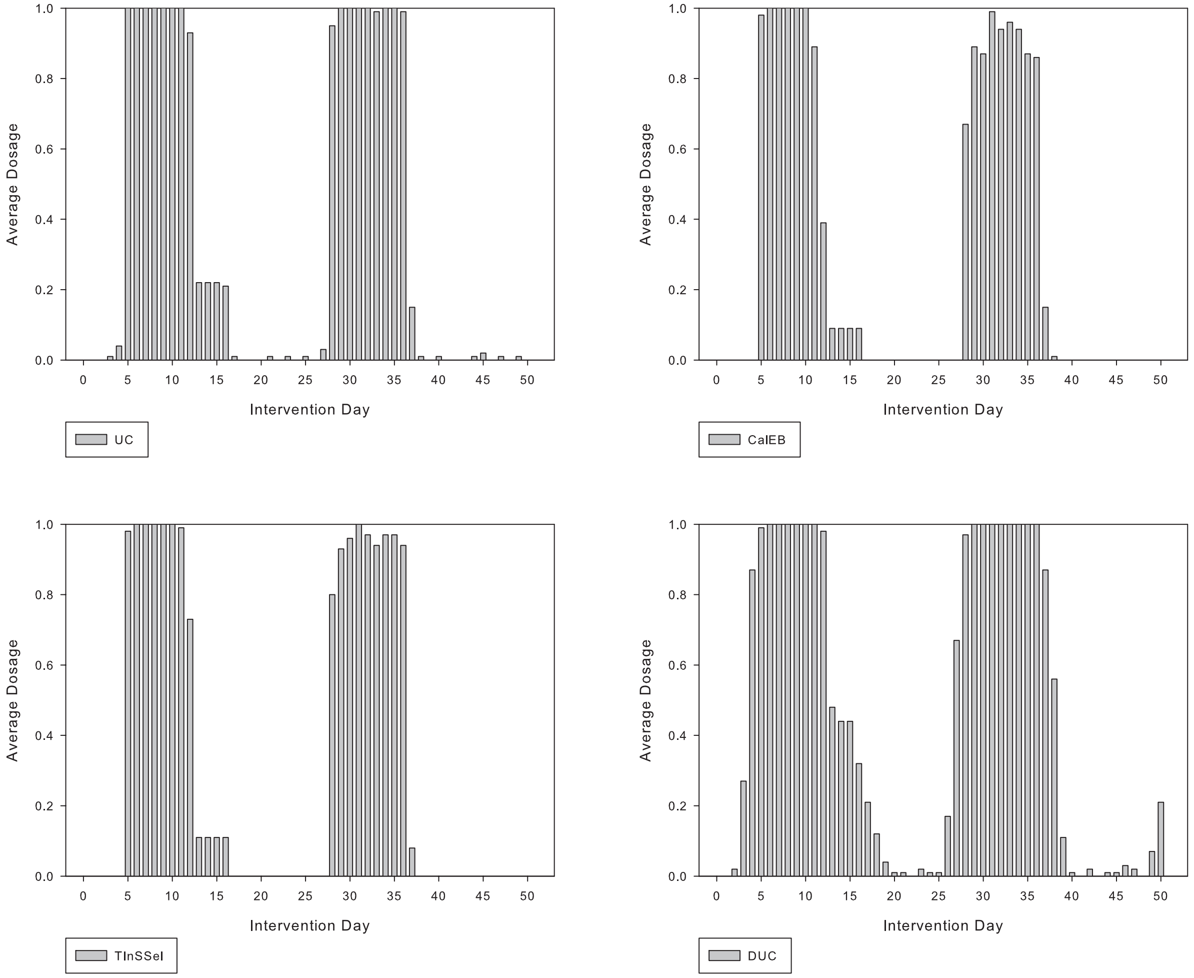
Figure 8.12 shows the effect of increasing p_m to 0.05. In this situation, CalEB and TInSSel reduce to around 15 interventions, whereas both UC and DUC settle on around 17 interventions. Even with this difference in intervention levels, all approaches return similar fitness scores.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the directed intervention crossover approaches, in comparison to UC, under a generational replacement strategy for a penalty of 5 points per intervention.

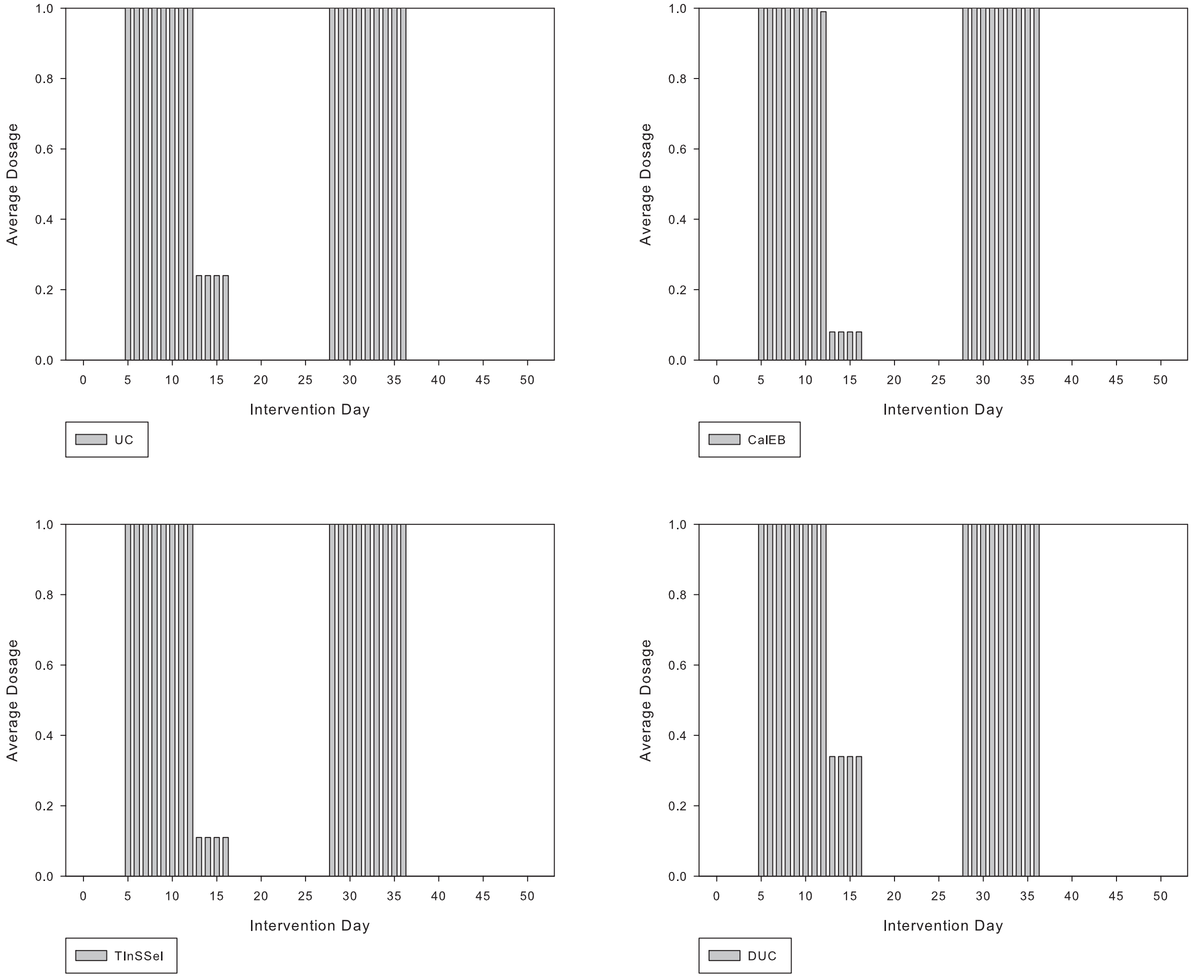
In the case of no mutation, as shown in Figure 8.10, DUC is shown to use the most interventions, with UC and TInSSel using less, and CalEB using the least interventions. From the statistical analysis undertaken in Appendix E.4, after 5,000 FFEs, UC was shown to outperform DUC, TInSSel outperformed CalEB and DUC and CalEB outperformed DUC for this FFE point. This section reviews the placement of interventions by each of the crossover approaches at 5,000 FFEs. The average placement of these interventions are shown in Figure 8.13. This shows that CalEB, TInSSel and UC all use more focussed interventions than DUC, and as there is a cost associated with each intervention, this explains this fitness difference. Both TInSSel and UC are shown to have on average more applications on days 11 and 12 than CalEB.

Figure 8.13: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 5$ and $R = g$



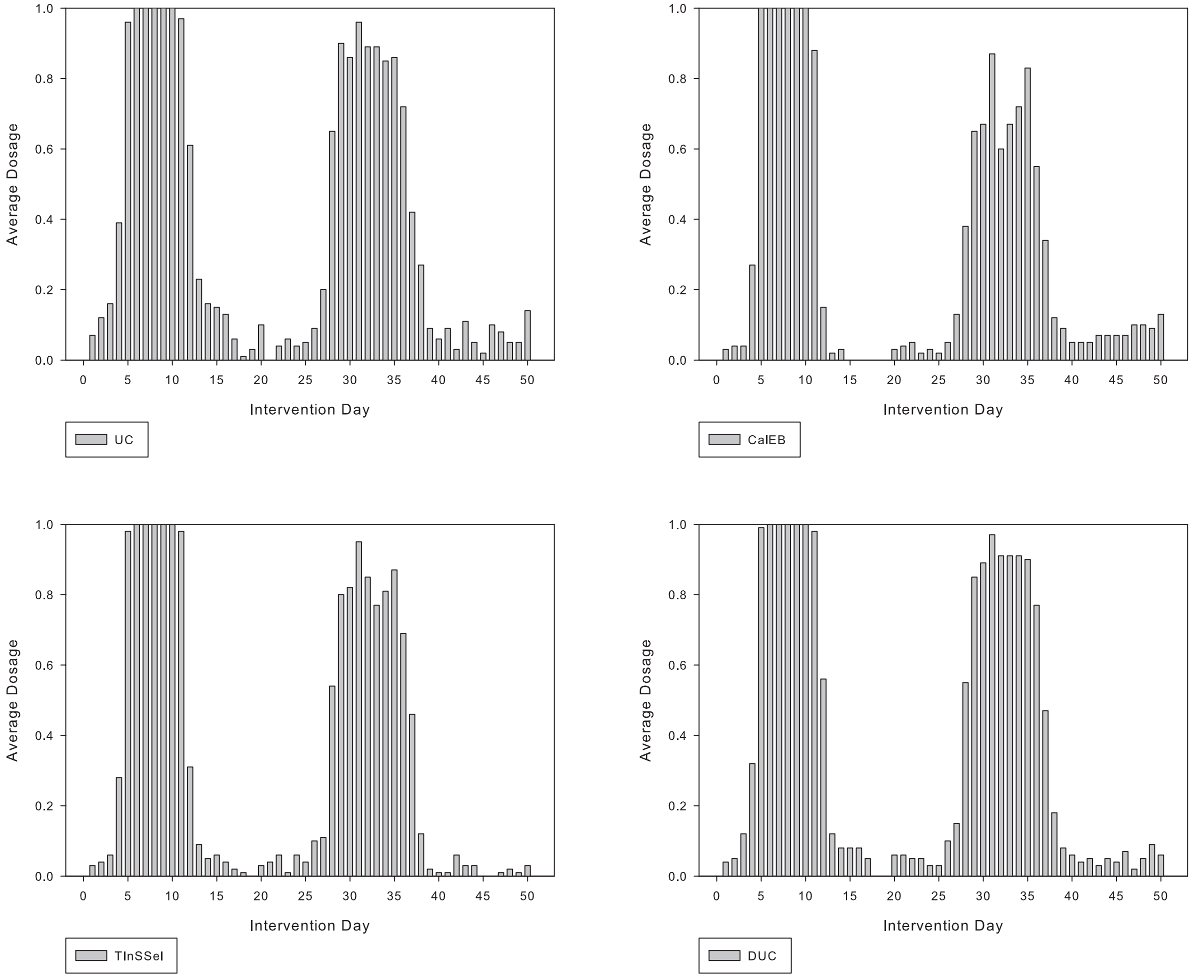
The interventions and fitness scores for a mutation rate, p_m, of 0.005 are shown in Figure 8.11. Each of the approaches were shown to use the same number of interventions, producing similar fitness scores. From the statistical analysis of these experiments, in Appendix E.5, the only statistical difference between approaches for 5,000 FFEs was that CalEB outperformed DUC. From reviewing the average intervention placement for this mutation level, shown in Figure 8.14, each of the approaches use similar intervention placement. The statistical difference between CalEB and DUC can be explained by CalEB using interventions 13 - 16 the least of all the crossover approaches and DUC using these the most.

Figure 8.14: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 5$ and $R = g$



The interventions and fitness scores when p_m is increased to 0.05 are shown in Figure 8.12. DUC and UC use more interventions than the other approaches, with TInSSel using more than CalEB. From the statistical analysis of these experiments, in Appendix E.6, both TInSSel and DUC were shown to outperform both CalEB and UC. From reviewing the average intervention placement for this mutation level, shown in Figure 8.15, CalEB and UC are shown to use interventions on days 40 - 50 more often on average than TInSSel and DUC.

Figure 8.15: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 5$ and $R = g$



Steady state replacement

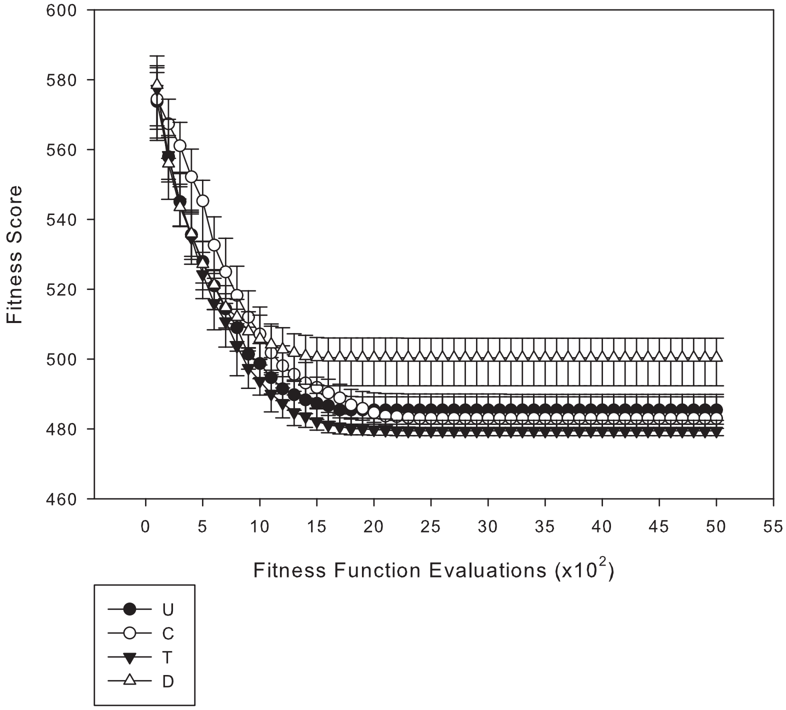
We now consider the alternative steady state replacement strategy. Figures 8.16 to 8.18 show the fitness score and intervention usage when $R = ss$ with p_m values of 0, 0.005 and 0.05. There is a penalty of 5 points associated with each intervention.

Figure 8.16 shows that when $p_m = 0$, DUC struggles to find scores of a comparable fitness for steady state as for generational replacement. This was also the case for 0 penalty points per intervention. Through inspection of intervention usage, DUC converges to schedules using 25 interventions under steady state replacement, while reducing to approximately 23 interventions under generational replacement. In the generational case, both UC and TInSSel settled to a similar number of interventions. This is not the case under steady state replacement, with UC requiring more interventions than TInSSel. Due to this, UC returns scores of a lower fitness under steady state replacement than with the generational equivalent.

Figure 8.17 details the effect of increasing p_m to 0.005. With a steady state replacement strategy, each of the crossover approaches follow the same patterns as demonstrated for a generational replacement strategy. This is in terms of both fitness scores and intervention usage.

When p_m is further increased to 0.05, as shown in Figure 8.18, similar trends are displayed by the crossover approaches, regardless of which replacement strategy is used.

Figure 8.16: Results for $N = 100, p_c = 0.9, p_m = 0$,$P = 5$ and $R = ss$



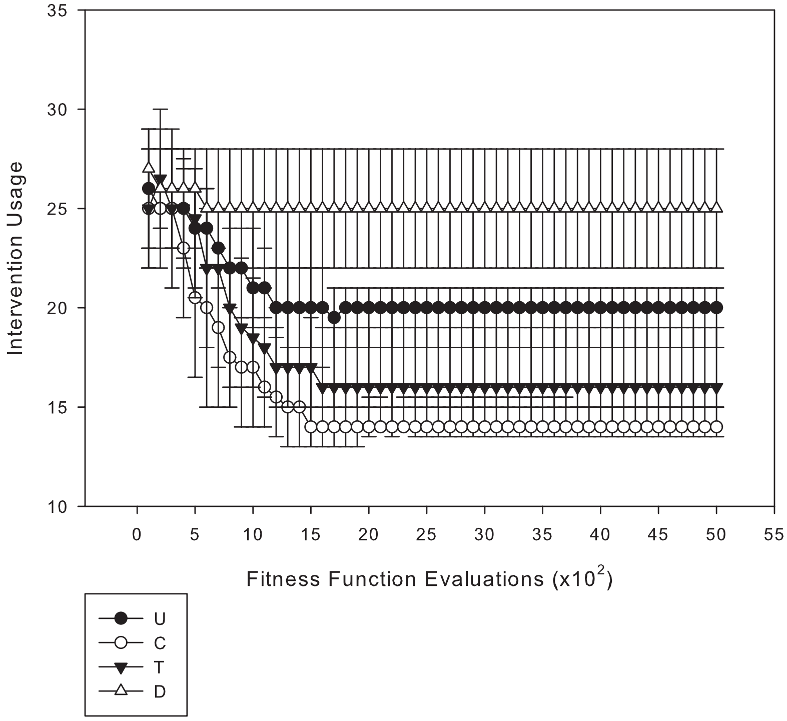
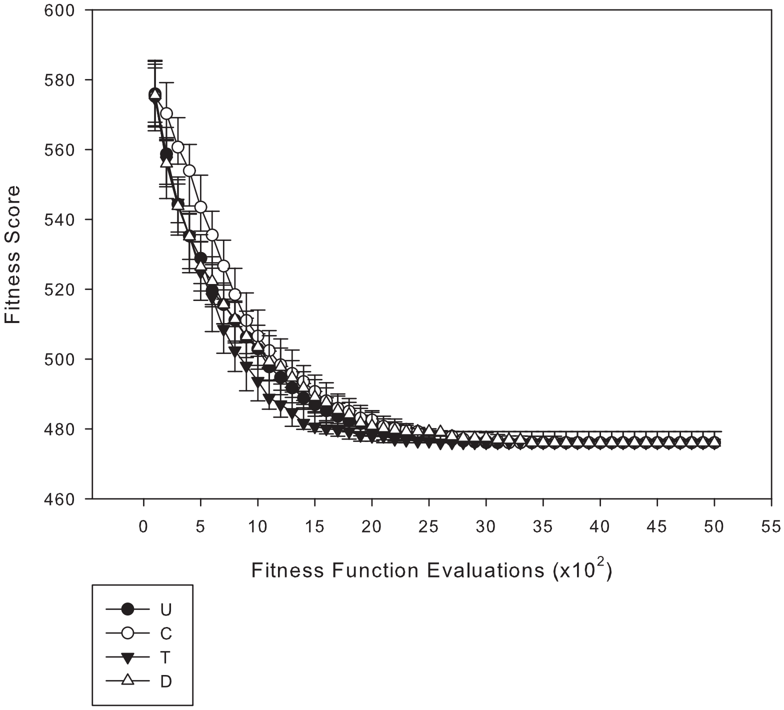


Figure 8.17: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 5$ and $R = ss$



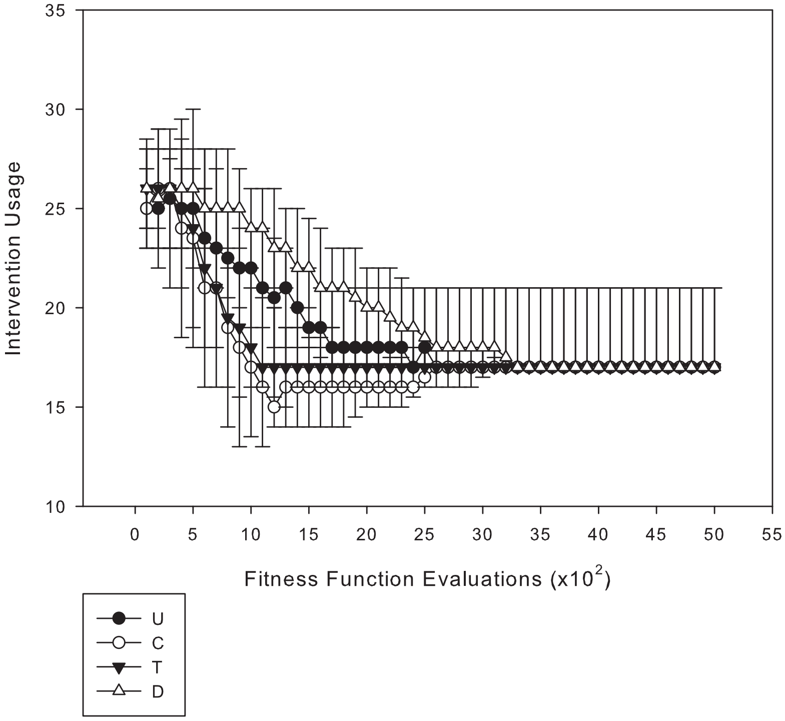
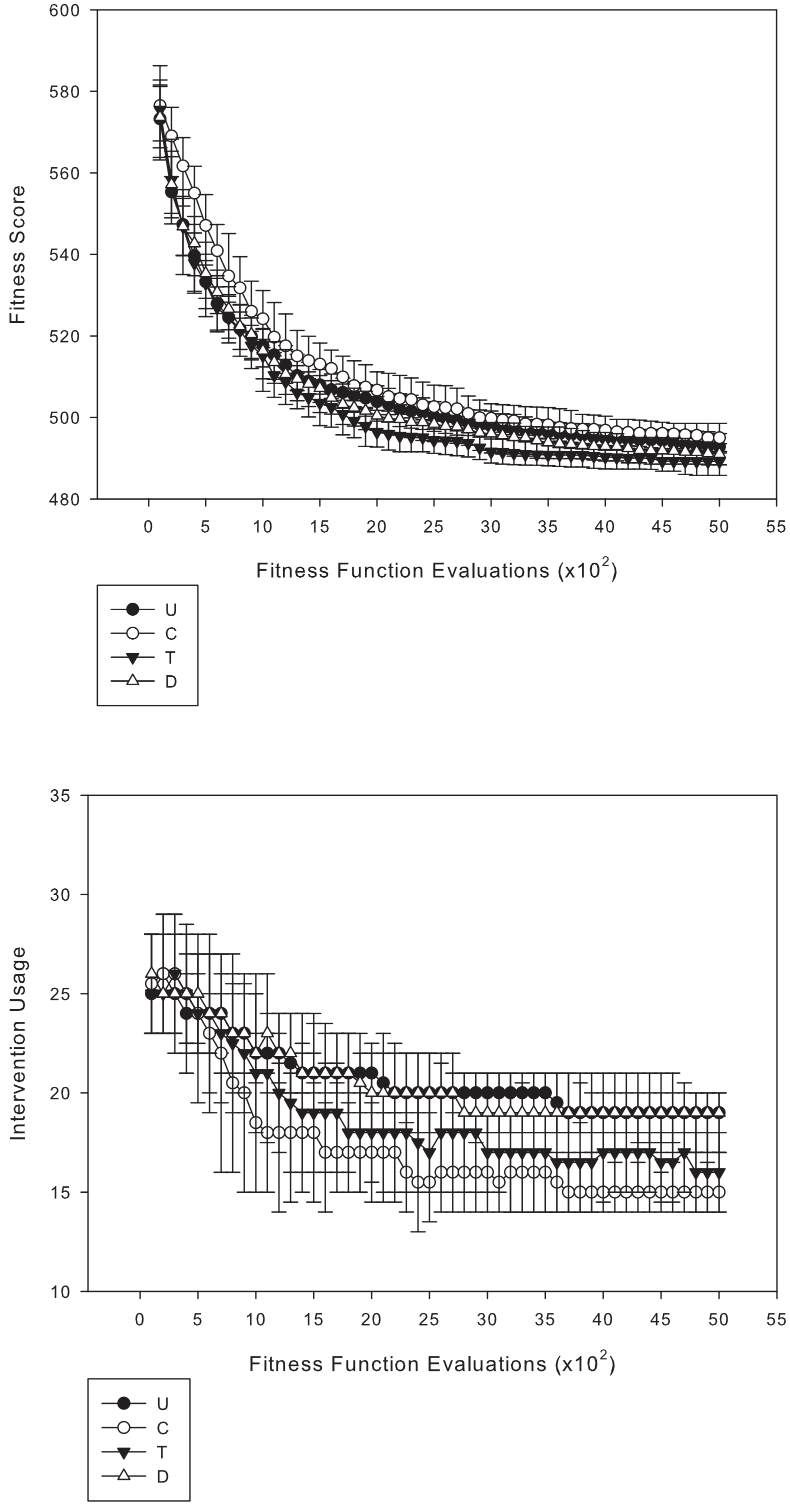


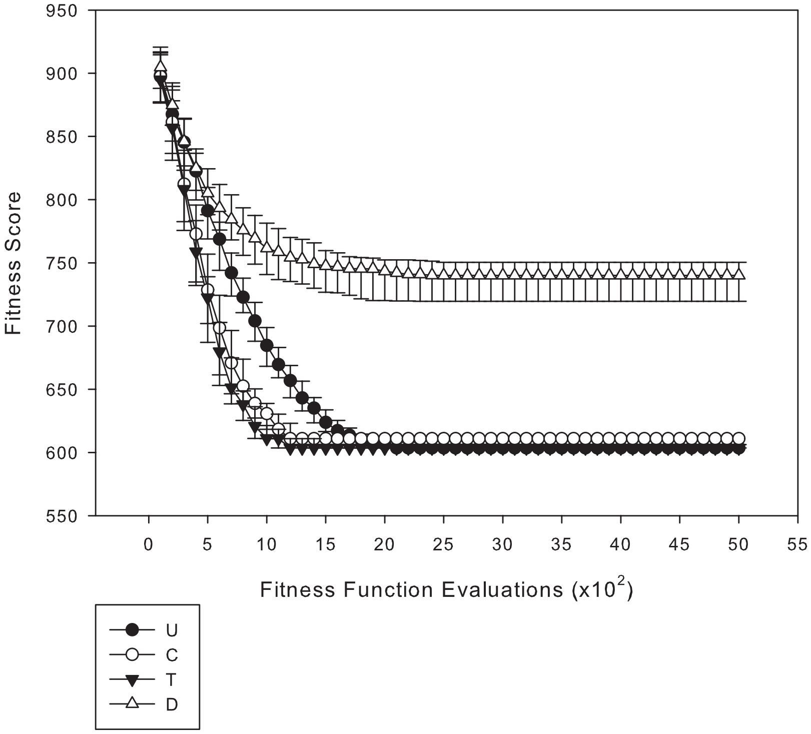
Figure 8.18: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 5$ and $R = ss$



8.1.3 20 penalty points per intervention

The next experiment considers the effect of increasing the number of penalty points per intervention from 5 to 20. Figures 8.19 to 8.21 show the fitness scores and associated intervention usage for a penalty of 20 per intervention and p_m values of 0, 0.005 and 0.05 respectively.

Figure 8.19: Results for $N = 100, p_c = 0.9, p_m = 0$,$P = 20$ and $R = g$



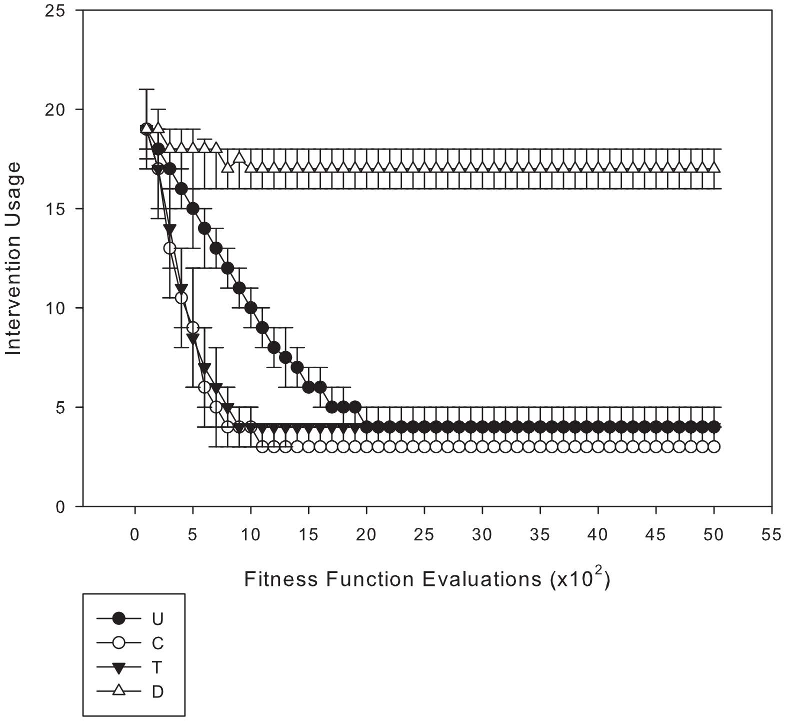
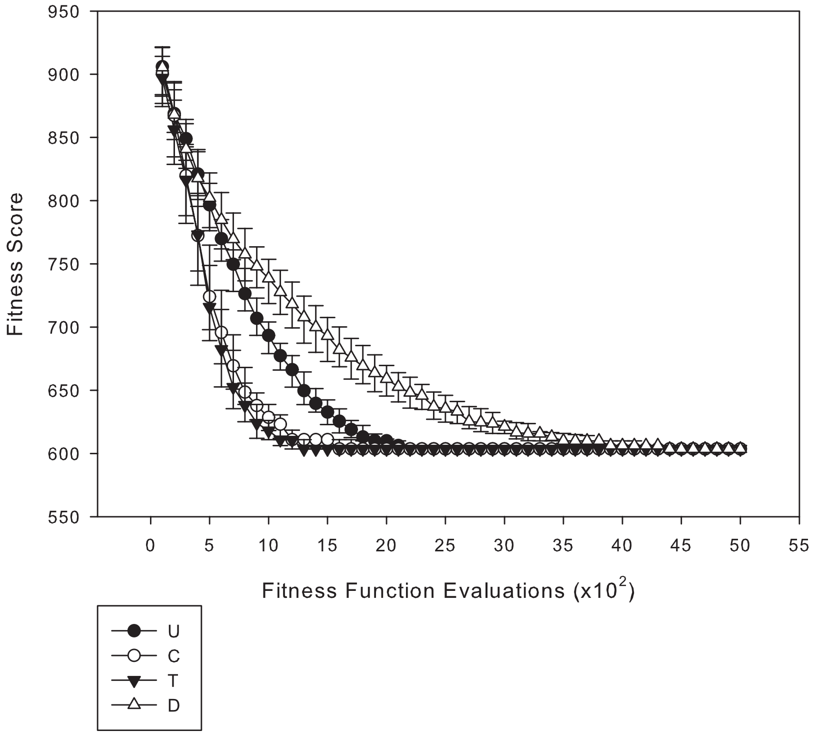


Figure 8.20: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 20$ and $R = g$



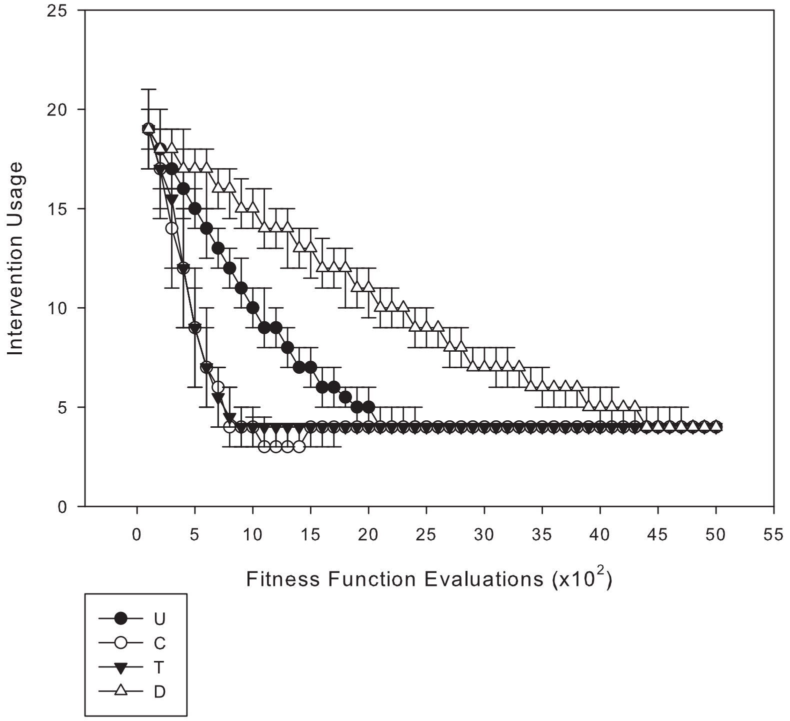
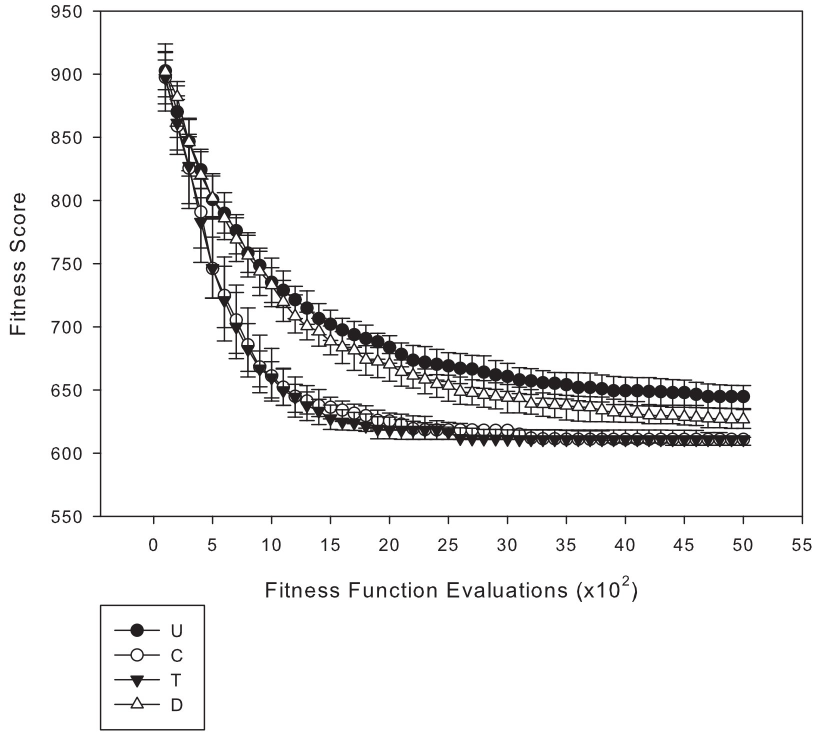
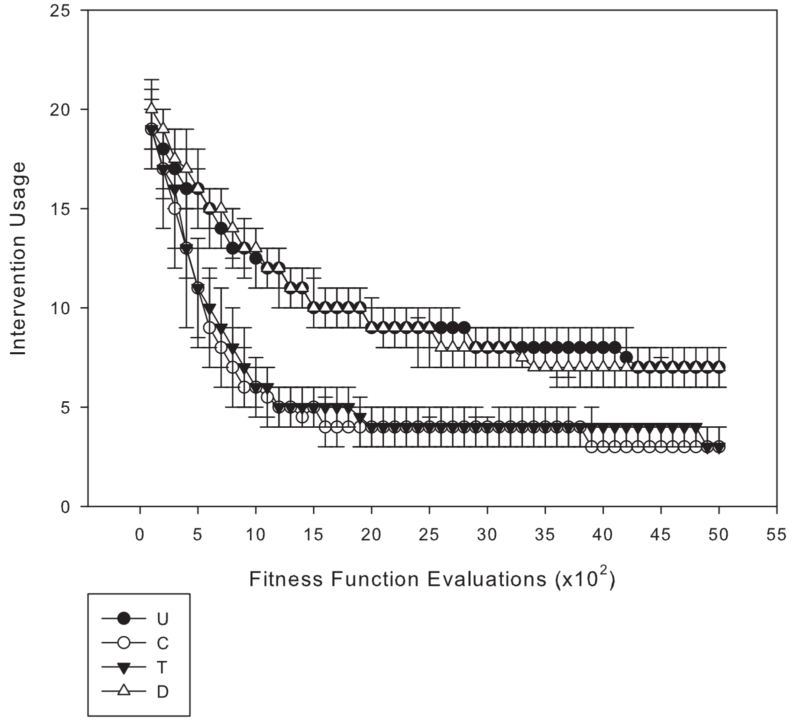


Figure 8.21: Results for $N = 100, p_c = 0.9, p_m = 0.05, P = 20$ and $R = g$





Although there have been correlations between intervention usage and fitness scores for lower intervention penalties, for a penalty of 20 per intervention, the relationship is more apparent.

Figure 8.19 shows that when p_m is 0, DUC settles on approximately 17 interventions, compared to the approximately 5 interventions used by UC, CalEB and TInSSel. A trend is demonstrated in the number of interventions used by the schedule and the associated fitness score. CalEB and TInSSel are quicker to produce schedules requiring a small number of interventions, and this is reflected in the rapid reduction in fitness scores. UC is slower than both CalEB and TInSSel to reduce intervention levels, and this gradual reduction in intervention level are shown in a similar pattern for the fitness score. DUC quickly stagnates to around 17 interventions, and this produces poorer scores for the duration of this experiment.

Figure 8.20 details the experiment when the probability of mutation, $p\ _m$, is increased to 0.005. There is a clear trend between the reduction in intervention levels to approximately 5 and the fitness scores returned from each of the approaches. The order in which the crossover techniques undertake intervention reduction is the same as for a p_m of 0. CalEB and TInSSel quickly reduce intervention levels, UC does so at a more gradual rate and DUC requires approximately double the FFEs to reduce to a similar number of interventions.

When p_m is increased to 0.05, as shown in Figure 8.21, UC and DUC settle to approximately 7 interventions, while CalEB and TInSSel settle on 5. This difference in the number of interventions used is clearly reflected with the different converged fitness scores returned by the approaches.

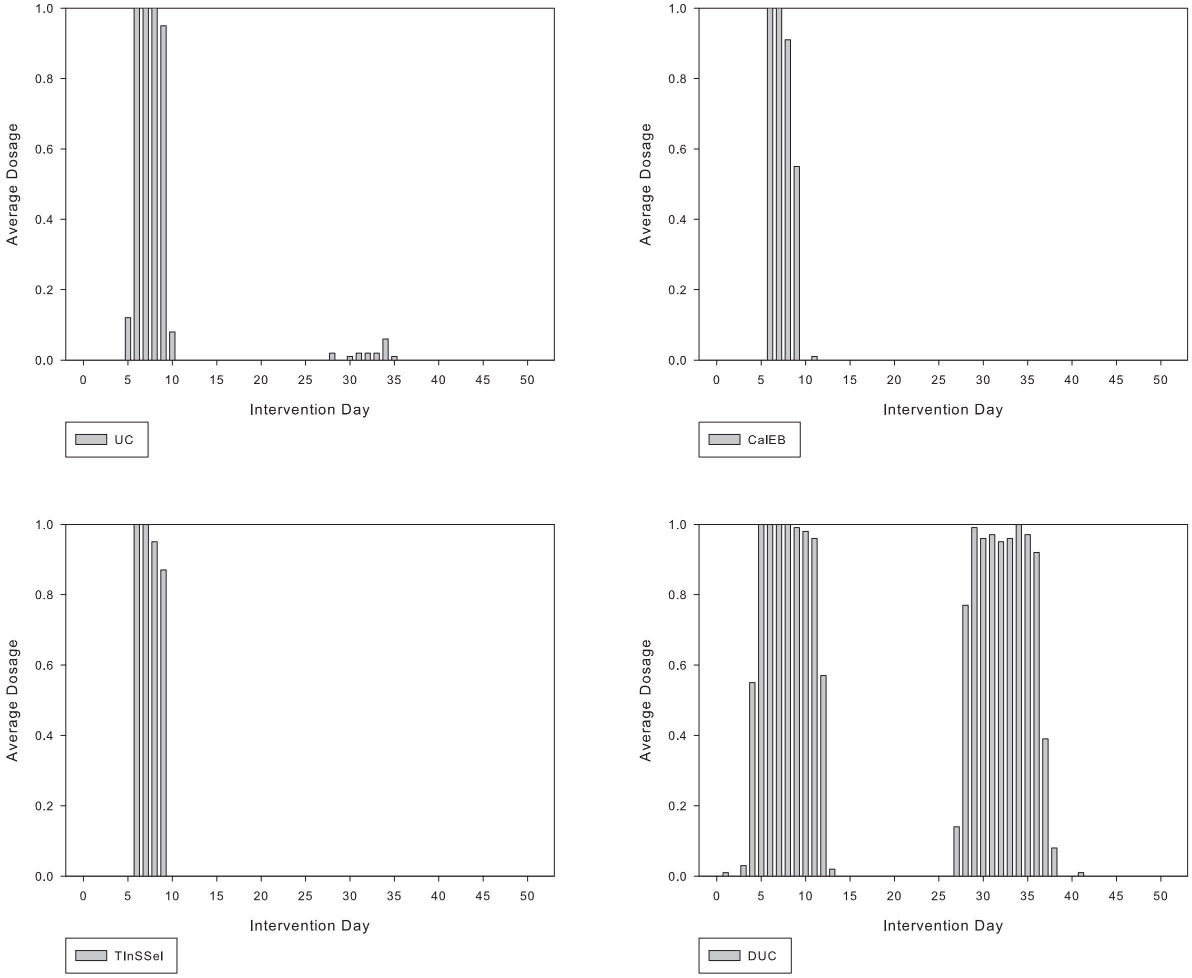
Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the directed intervention crossover approaches, in comparison to UC, under a generational replacement strategy for a penalty of 20 points per intervention.

In the case of no mutation, as shown in Figure 8.19, as with a penalty of 5 points per intervention, DUC is shown to use the most interventions, with UC and TInSSel using less, and CalEB using the least interventions. From the statistical analysis undertaken in Appendix E.7, after 5,000 FFEs, UC, CalEB and TInSSel are all shown to outperform DUC. TInSSel also outperforms CalEB at this mutation rate. This section reviews the placement of interventions by each of the crossover approaches at 5,000 FFEs. The average placement of these interventions are shown in Figure 8.22.

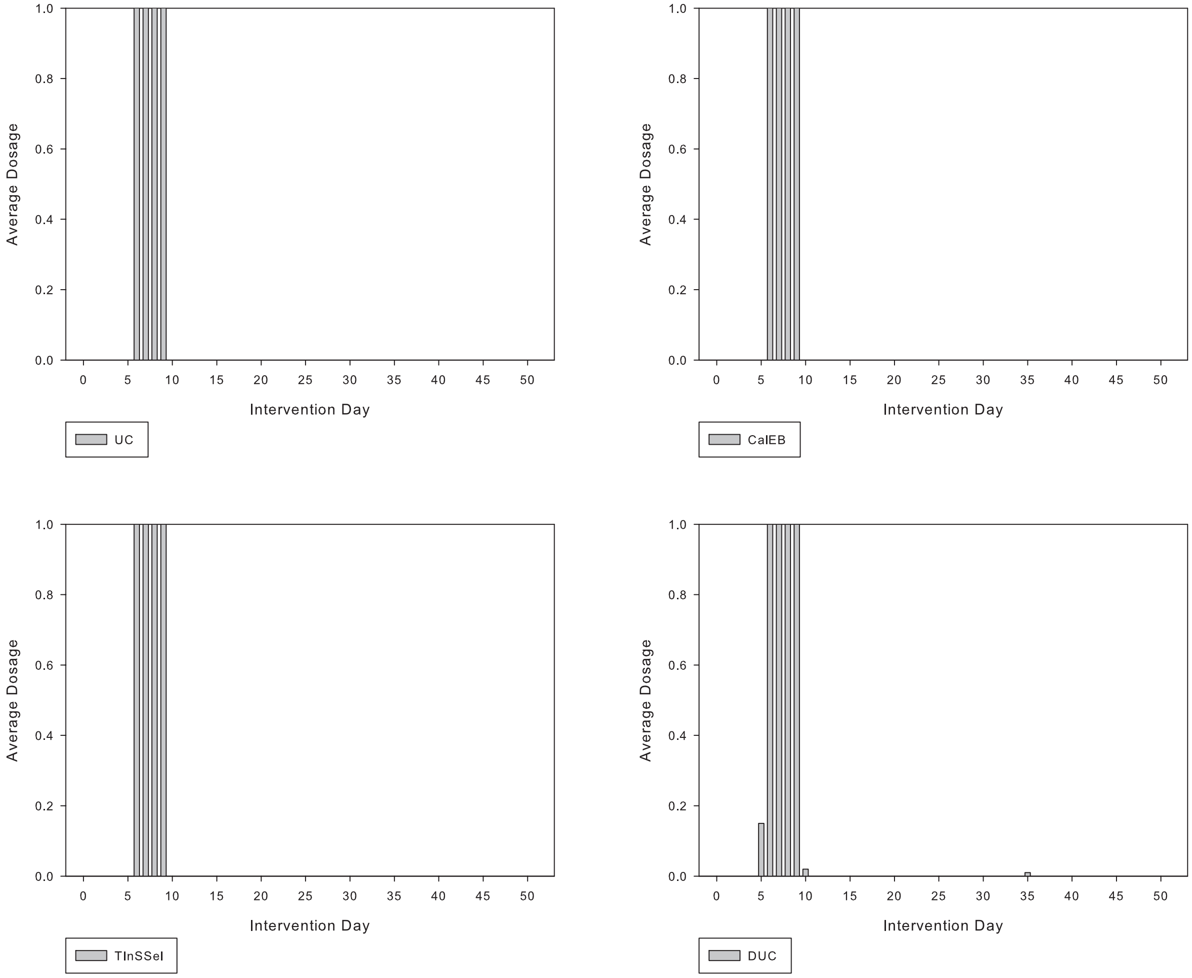
UC, CalEB and TInSSel all focus interventions predominantly on the first sciarid larvae cycle, days 5 - 9. While DUC also places interventions at these points, it also places interventions to target the second sciarid larvae cycle. The benefit achieved through targeting this second cycle appears to be outweighed by the associated intervention penalties. TInSSel outperforms CalEB for this FFE and this appears to be due to TInSSel placing more emphasis on interventions on days 8 and 9 of the schedule.

Figure 8.22: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 20$ and $R = g$



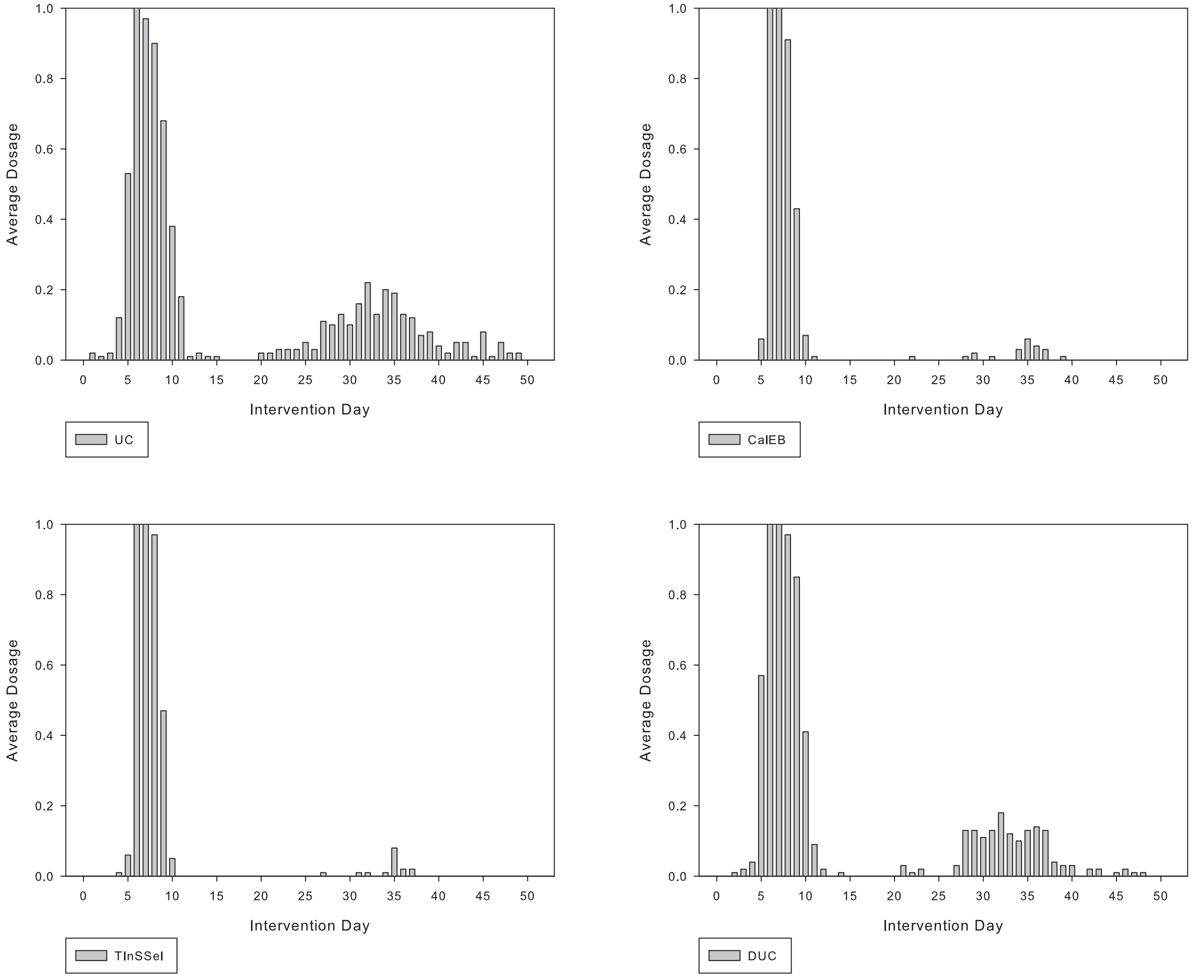
The intervention usage and fitness scores when p_m was increased to 0.005 are shown in Figure 8.20. All approaches use a similar number of interventions at 5,000 FFEs, resulting in no statistical differences between approaches, as described in Appendix E.8. The average placement of these interventions are shown in Figure 8.23. This shows that for this mutation level, each of the crossover approaches focus mainly on intervening on days 6 - 9 of the schedule.

Figure 8.23: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 20$ and $R = g$



The intervention usage and fitness scores when p_m was further increased to 0.05 are shown in Figure 8.21. For this mutation level, UC and DUC use more interventions than CalEB or TInSSel. From the statistical analysis undertaken in Appendix E.9 CalEB, TInSSel and DUC all outperform UC at 5,000 FFEs, with CalEB and TInSSel also outperforming DUC at this point. The average placement of interventions are shown in Figure 8.24. This shows that for this mutation level, while all approaches focus on the first sciarid larvae cycle, both UC and DUC also place more interventions between days 20 and 50 than the other approaches. This explains why CalEB and TInSSel outperform the other approaches. As DUC places less emphasis than UC at the later cycle, this also accounts for DUC outperforming UC at this mutation level.

Figure 8.24: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 20$ and $R = g$



Steady state replacement

We now repeat the previous evaluation for a steady state replacement strategy. Figures 8.25 to 8.27 show the fitness scores and associated intervention usage for a steady state replacement strategy with a penalty of 20 per intervention.

Figure 8.25 shows a steady state replacement strategy and a p_m of 0. At this mutation level, CalEB, TInSSel and DUC behave in a similar manner as they did under a generational replacement strategy. UC converges to around 6 interventions under steady state, as opposed to 4 interventions under generational replacement. This higher number of interventions in schedules results in less optimal scores for UC under steady state replacement.

When p_m is increased to 0.005, as shown in Figure 8.26, UC, CalEB and TInSSel perform in a similar manner as under generational replacement. DUC is quicker at reducing intervention levels under steady state replacement. This results in DUC converging to similar scores as the other crossover approaches after 3,000 FFEs under steady state replacement as opposed to 4,000 FFEs for generational replacement.

Figure 8.27 shows a steady state replacement strategy when p_m is further increased to 0.05. This has a similar intervention usage pattern as for generational replacement, resulting in similar fitness trends.

Figure 8.25: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$,$P = 20$ and $R = ss$

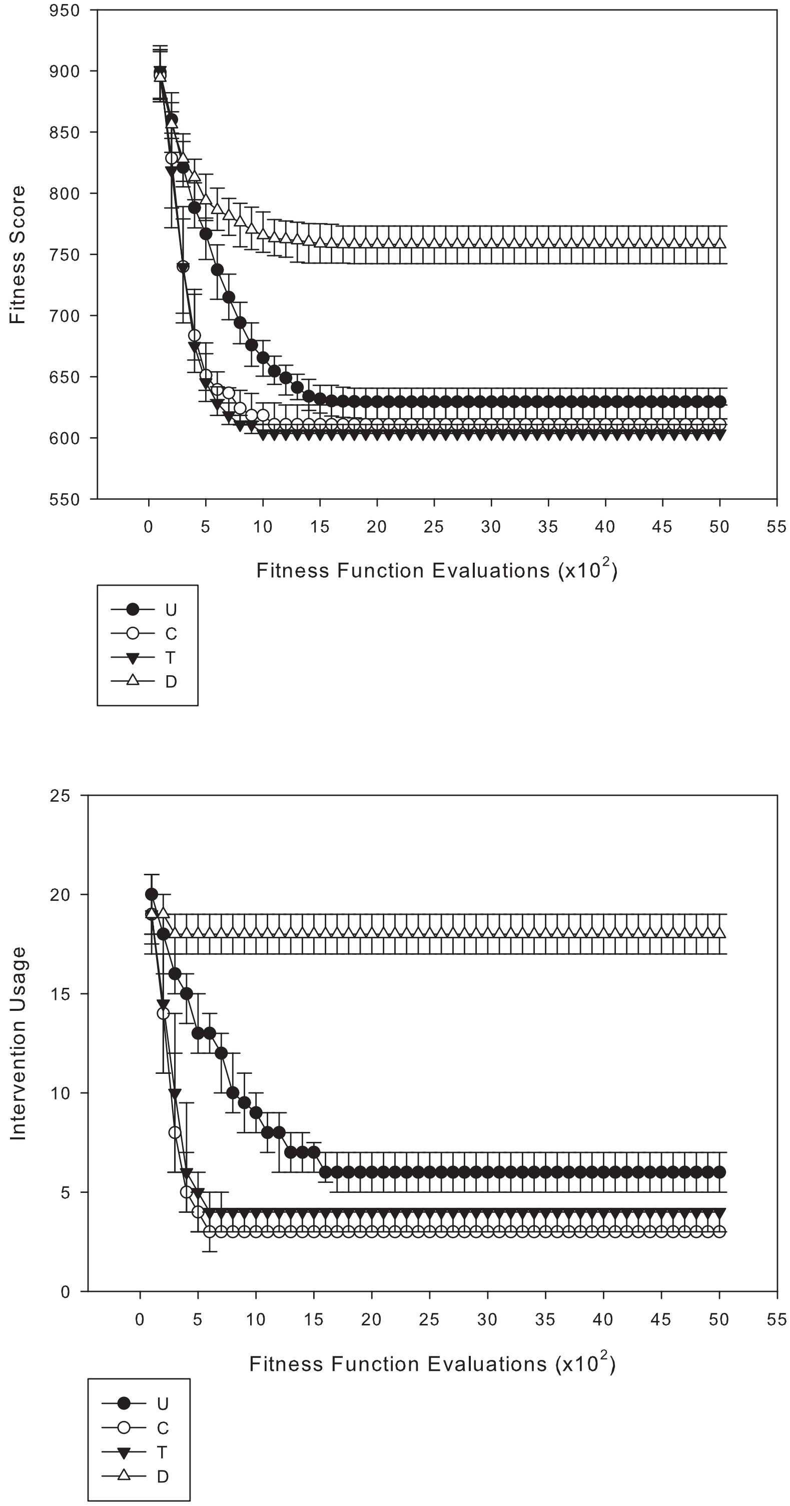
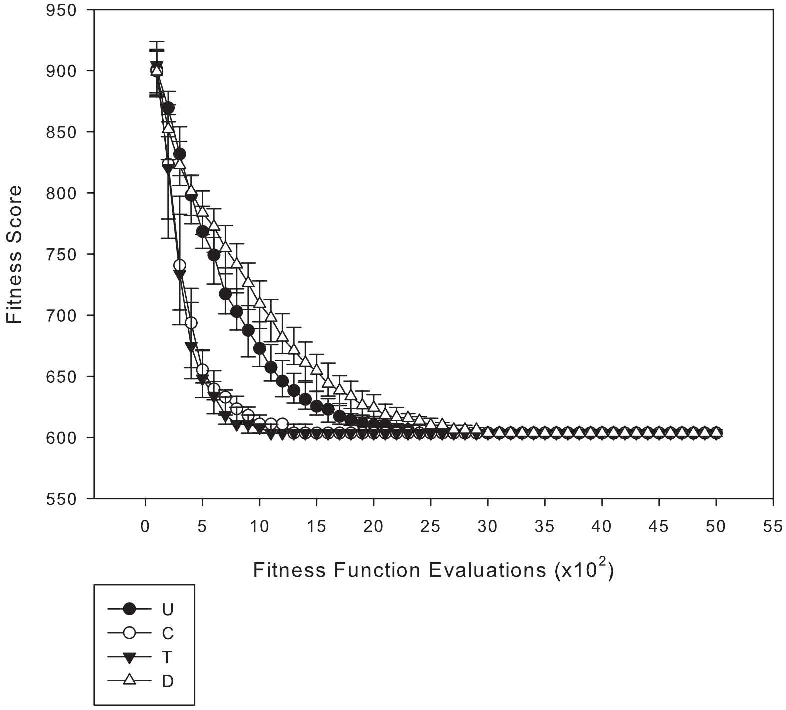


Figure 8.26: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 20$ and $R = ss$



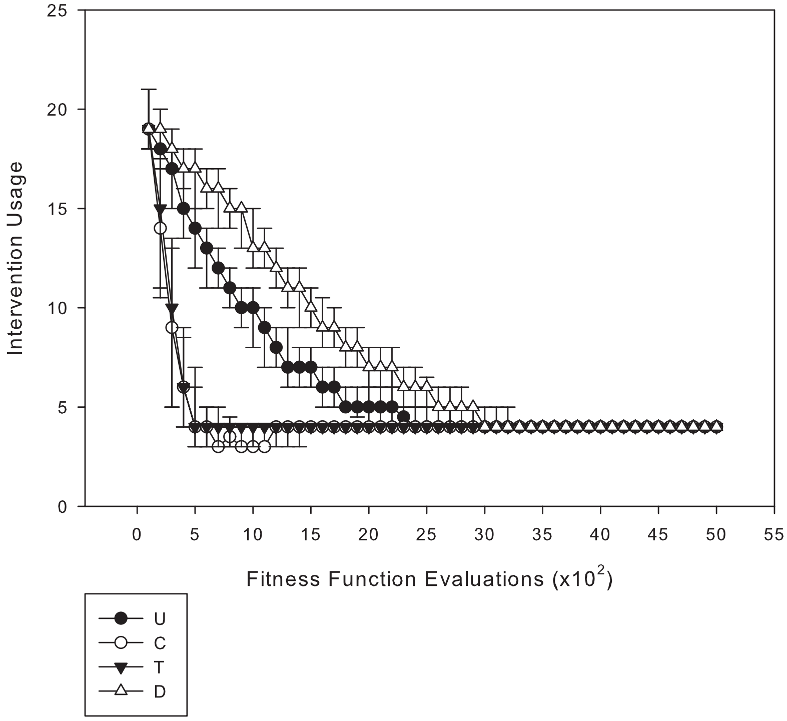
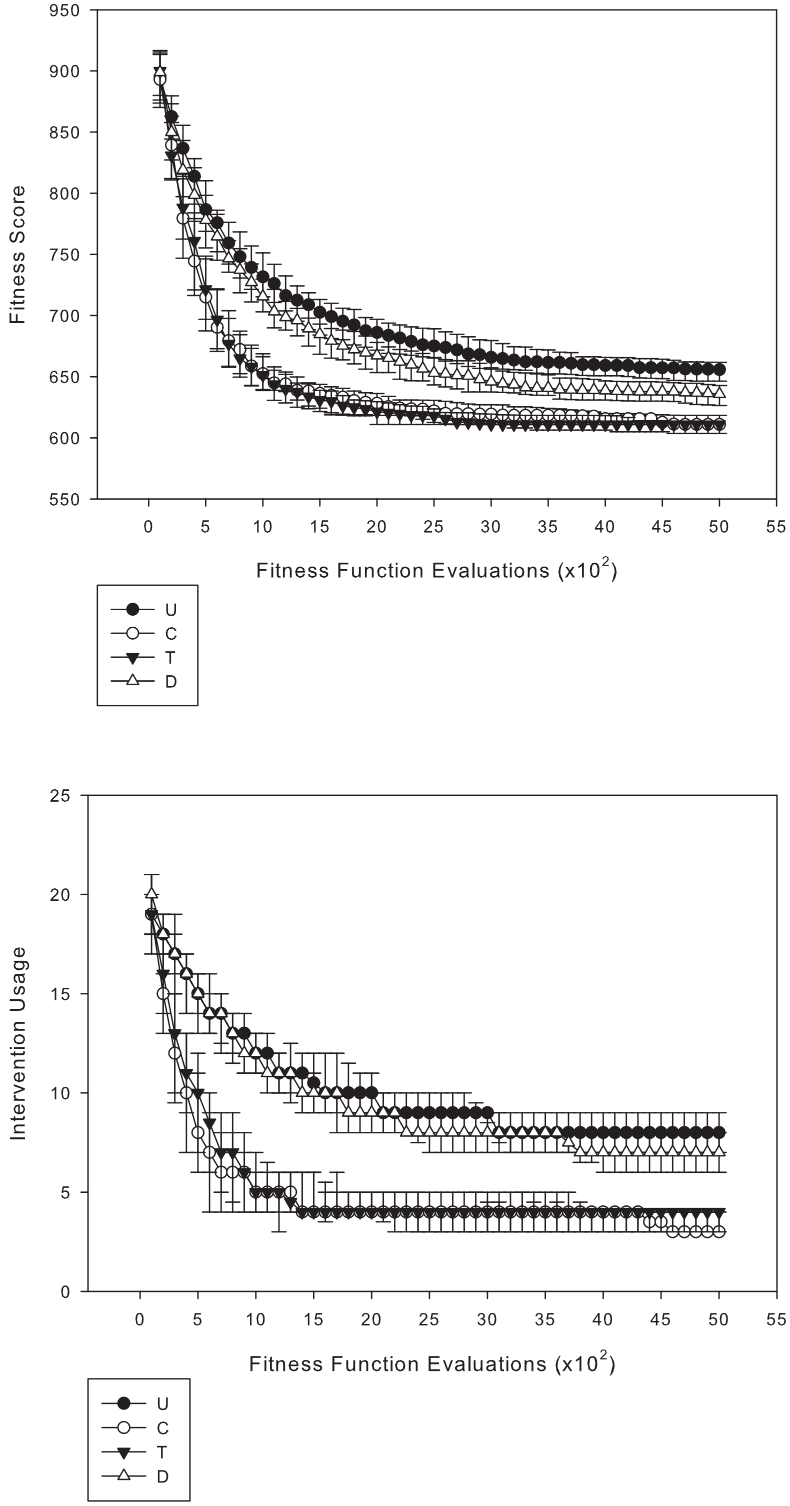


Figure 8.27: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 20$ and $R = ss$



8.1.4 35 penalty points per intervention

The previous sections have detailed the abilities of CalEB, TInSSel, DUC and UC over a range of intervention penalty values. This section reviews the affects of further increasing the penalty per intervention, from 20 to 35 points per intervention.

Figure 8.28 shows the fitness scores and associated intervention usage when $p_m = 0$. Figures 8.29 to 8.30 detail p_m levels of 0.005 and 0.05 respectively.

Figure 8.28: Results for $N = 100, p_c = 0.9, p_m = 0$,$P = 35$ and $R = g$

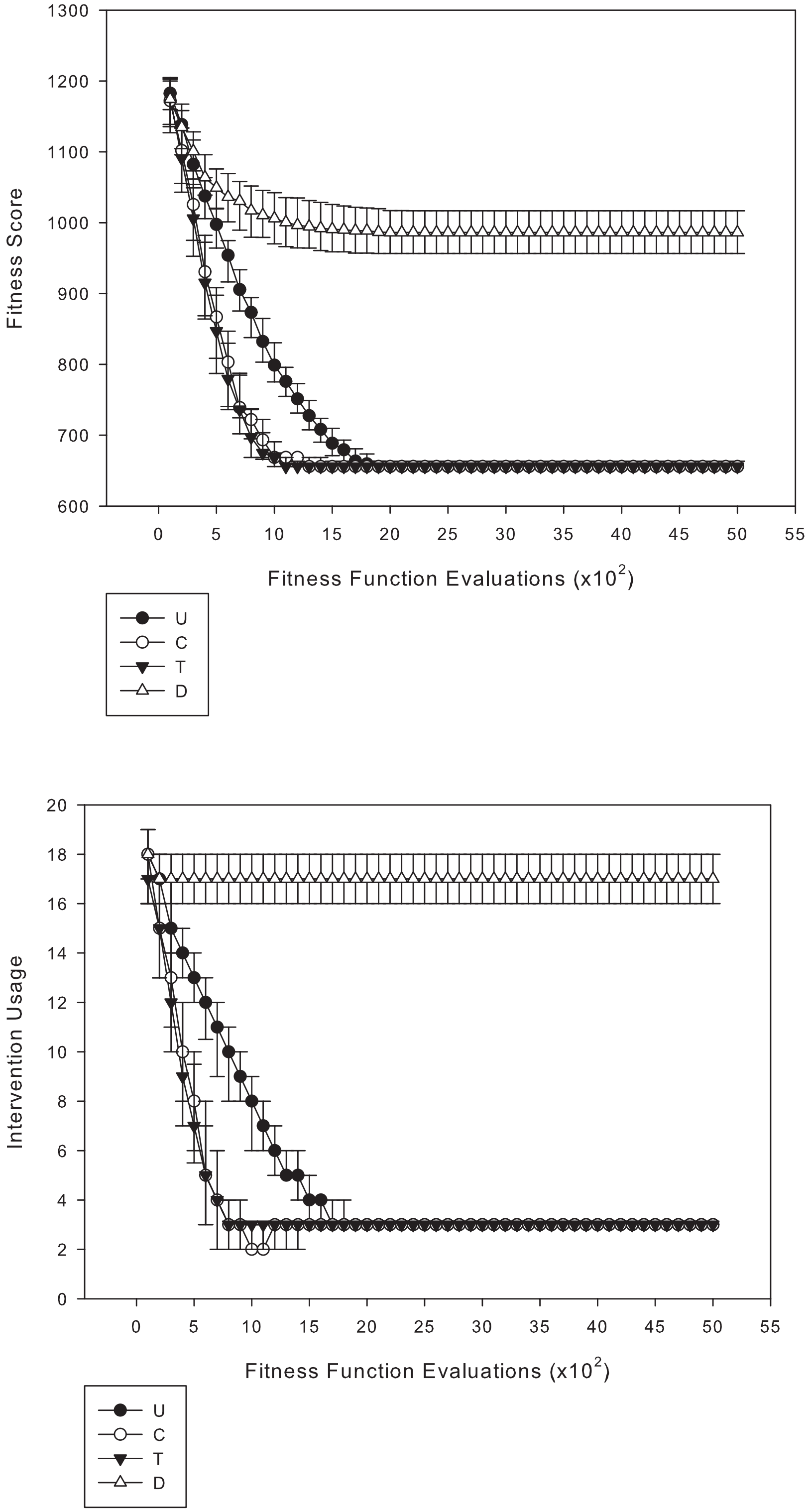
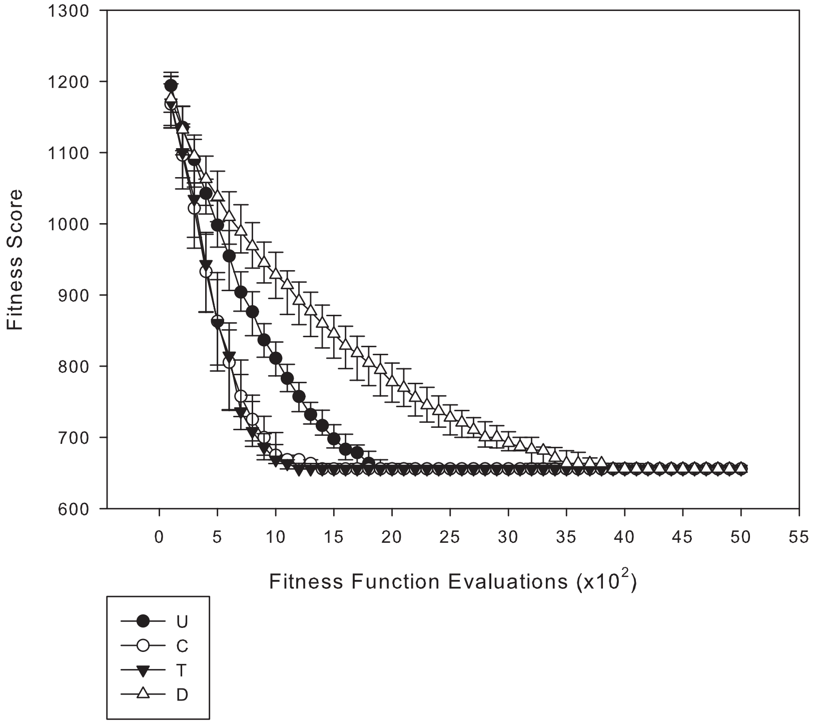


Figure 8.29: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 35$ and $R = g$



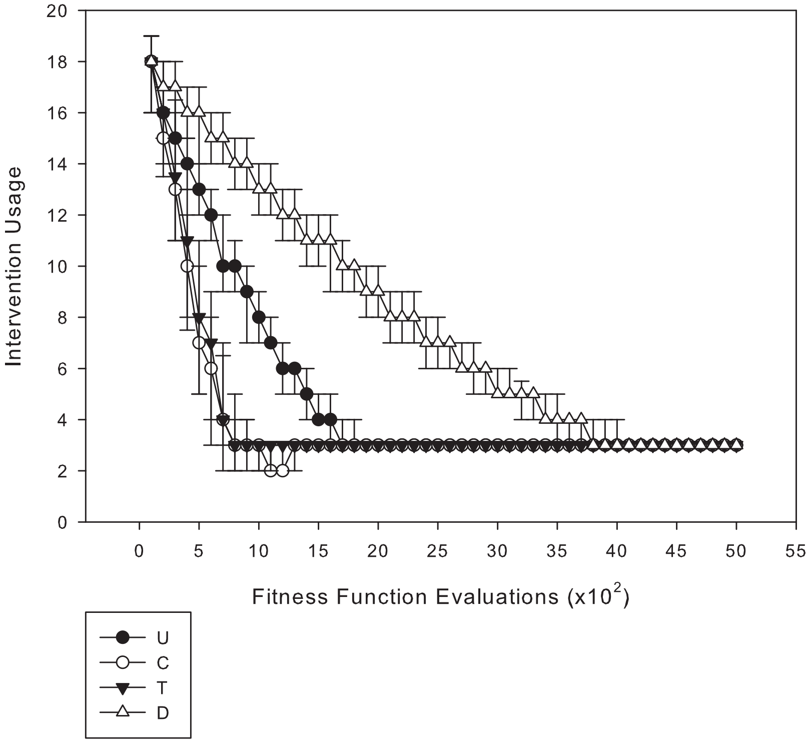
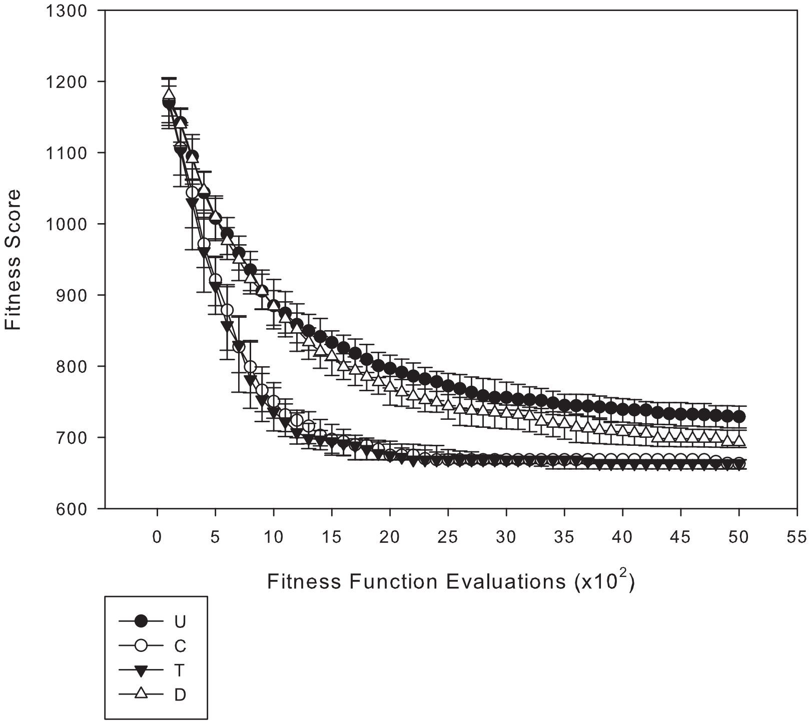
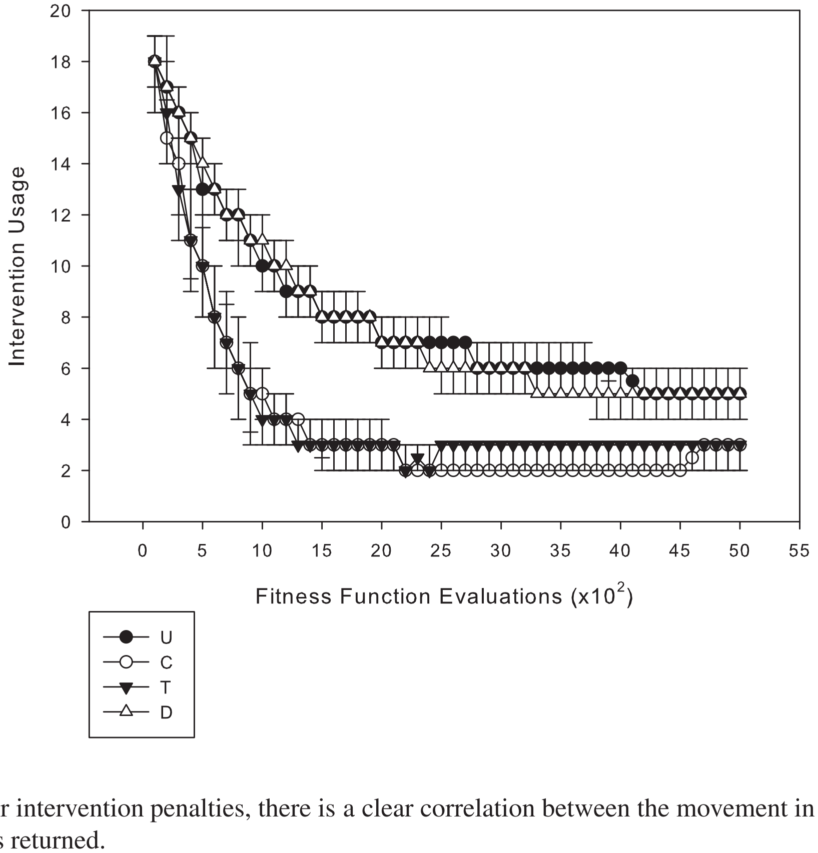


Figure 8.30: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 35$ and $R = g$





As with lower intervention penalties, there is a clear correlation between the movement in intervention usage and fitness scores returned.

Figure 8.28 shows that in the absence of mutation, with a penalty of 35 points per intervention, DUC uses 17 interventions as opposed to the 3 interventions used by UC, CalEB and TInSSel. This correlates with the gap in fitness scores returned by DUC compared to the other approaches for this experiment. There is further correlation between the steady intervention reduction of UC and the gradual progression of fitter scores found. Both CalEB and TInSSel demonstrate a correlation between reduced intervention usage and fitness of solution found, with a rapid decrease in intervention usage for schedules matched by a rapid improvement in fitness scores found.

Figure 8.29 shows the effect of increasing p_m to 0.005. For this mutation level, CalEB and TInSSel are quicker to reduce to 3 interventions than the other approaches. For this reason, they are quicker at deriving highly fit solutions. UC requires another 1,000 FFEs to settle on a similar number of interventions and return solutions of a comparable fitness. DUC requires approximately another 2,000 FFEs than UC to reduce to the same intervention level and fitness scores as the other approaches.

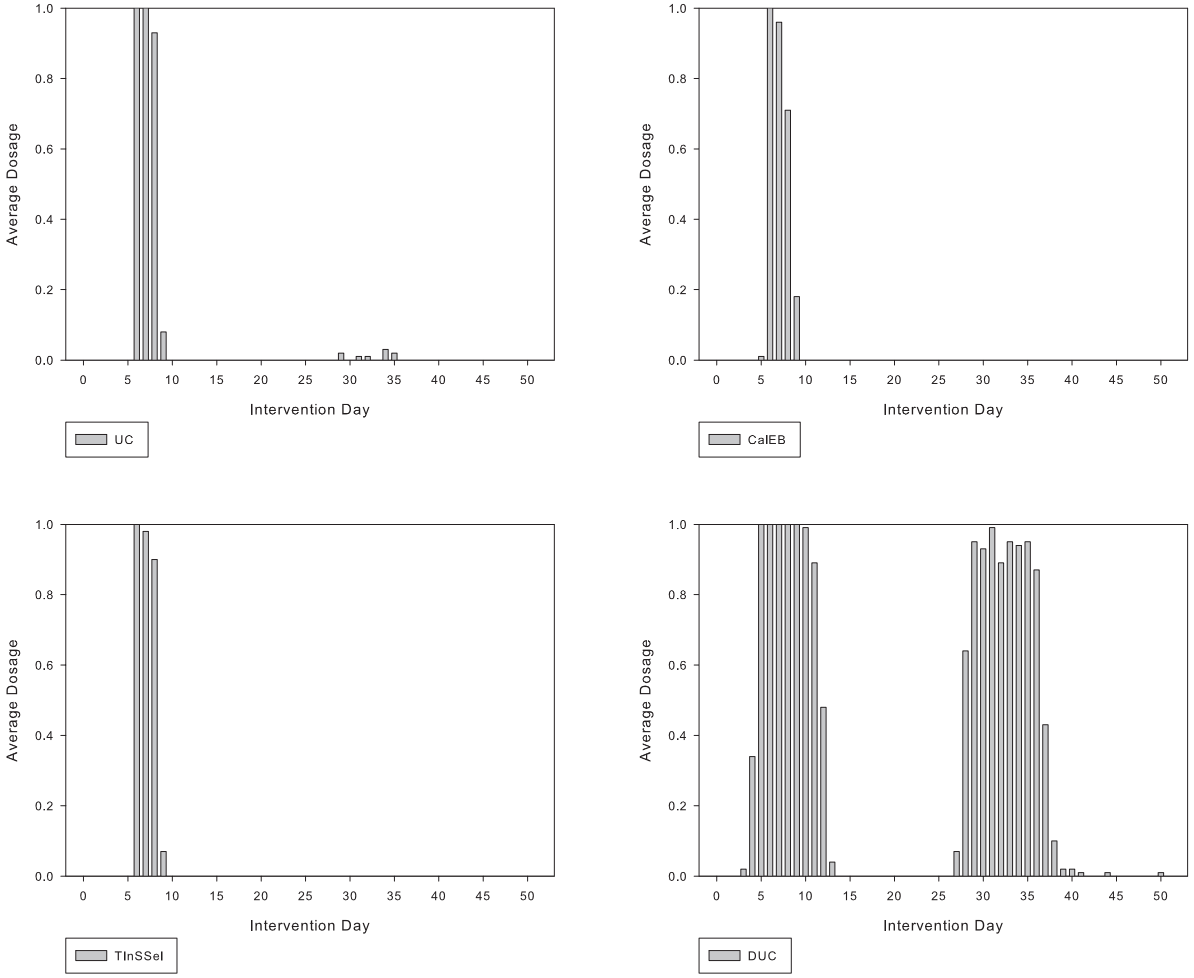
Figure 8.30 shows the effect of further increasing the mutation rate, p_m, to 0.05. For this mutation rate, a correlation appears between the lesser fitness scores returned by UC and DUC compared to CalEB and TInSSel, and the fact that UC and DUC converge to using around 5 interventions, whereas both CalEB and TInSSel settle on 3. It appears that with mutation at such a high level, CalEB and TInSSel can avoid the noise this introduces to reduce to smaller schedules, whereas UC and DUC struggle to reduce the number of interventions in schedules.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the directed intervention crossover approaches, in comparison to UC, under a generational replacement strategy for a penalty of 35 points per intervention.

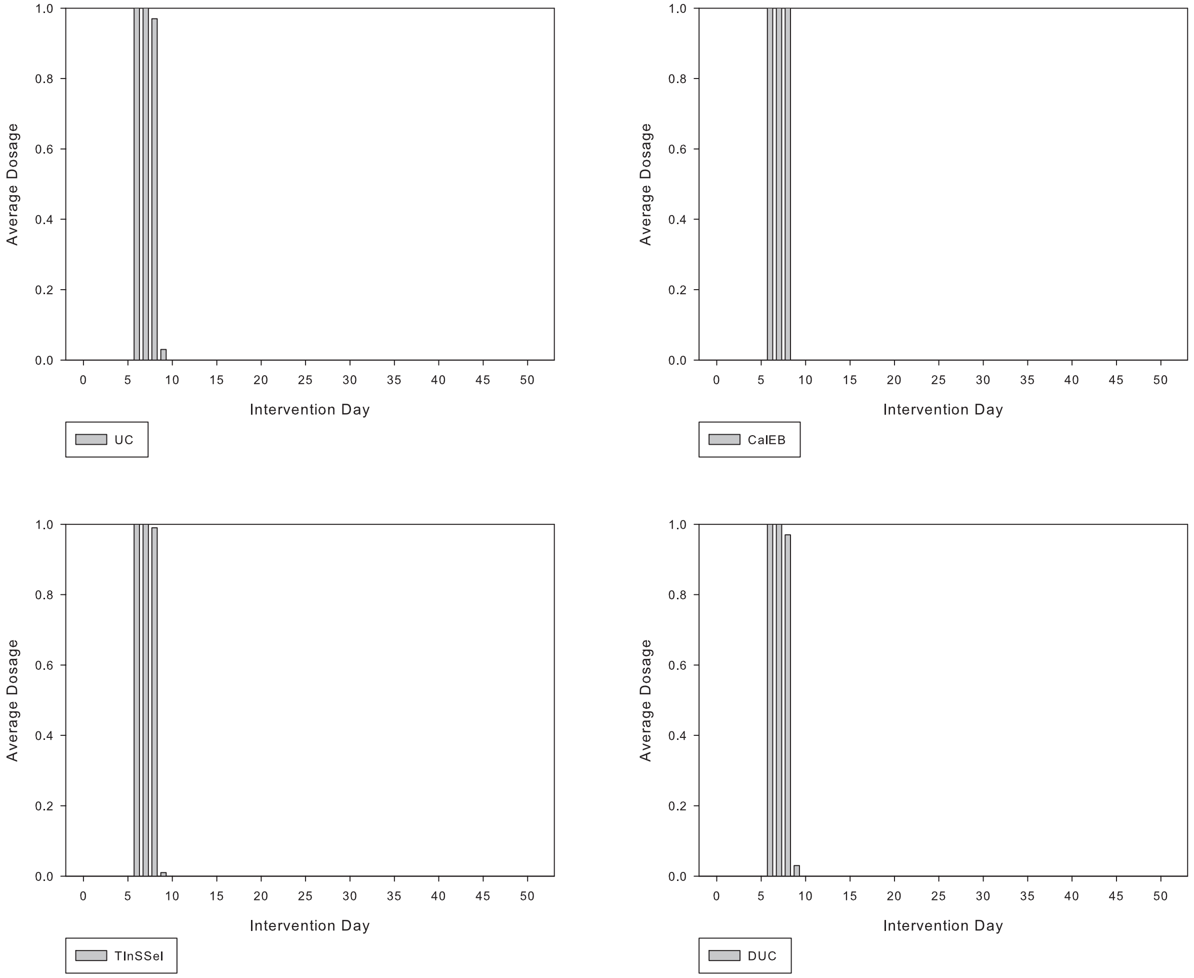
In the case of no mutation, as shown in Figure 8.28, DUC is shown to use more interventions than the other approaches, with CalEB, TInSSel and UC settling to a similar number of interventions. From the statistical analysis undertaken in Appendix E.10, after 5,000 FFEs, UC, CalEB and TInSSel are all shown to outperform DUC for this mutation rate. The average placement of the interventions relating to these findings are shown in Figure 8.31. This shows that UC, CalEB and TInSSel are each focussing interventions on days 6, 7 and 8, DUC focusses on this cycle as well as on the second larvae cycle. The extra interventions used by DUC are accruing extra penalty costs, producing less fit scores than the other approaches which focus solely on the first larvae cycle.

Figure 8.31: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 35$ and $R = g$



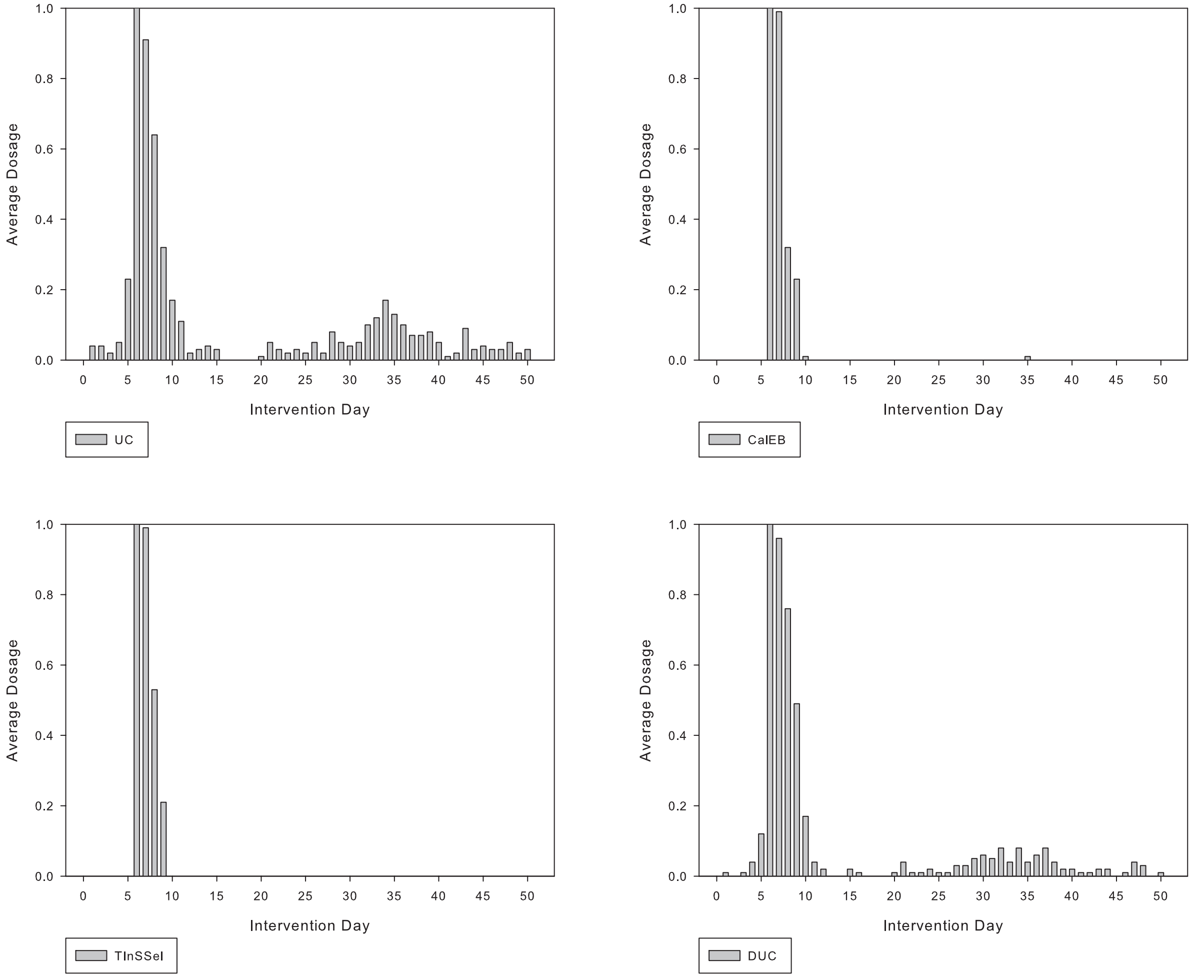
When p_m was increased to 0.005, as shown in Figure 8.29, each of the approaches use the same number of interventions, producing similar fitness scores. The statistical analysis undertaken in Appendix E.11, shows that after 5,000 FFEs, there is no statistical difference between crossover approaches for this mutation rate. The average placement of the interventions relating to these findings are shown in Figure 8.32. This shows that each of the crossover approaches focus on days 6, 7 and 8 of the schedule, focusing on the initial sciarid larvae cycle.

Figure 8.32: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 35$ and $R = g$



When p_m was further increased to 0.05, as shown in Figure 8.30, UC and DUC were shown to use more interventions than TInSSel and CalEB. The statistical analysis undertaken in Appendix E.12, shows that after 5,000 FFEs, each of the directed intervention crossover approaches outperform UC, with CalEB and TInSSel also outperforming DUC for this mutation rate. The average placement of the interventions relating to these findings are shown in Figure 8.33. This shows that while each of the crossover approaches focus on days 6 - 9, UC and DUC both also have interventions in the range 20 - 50. DUC does not use interventions after day 15 of the cycle as often as UC, hence the difference between these approaches.

Figure 8.33: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 35$ and $R = g$



Steady state replacement

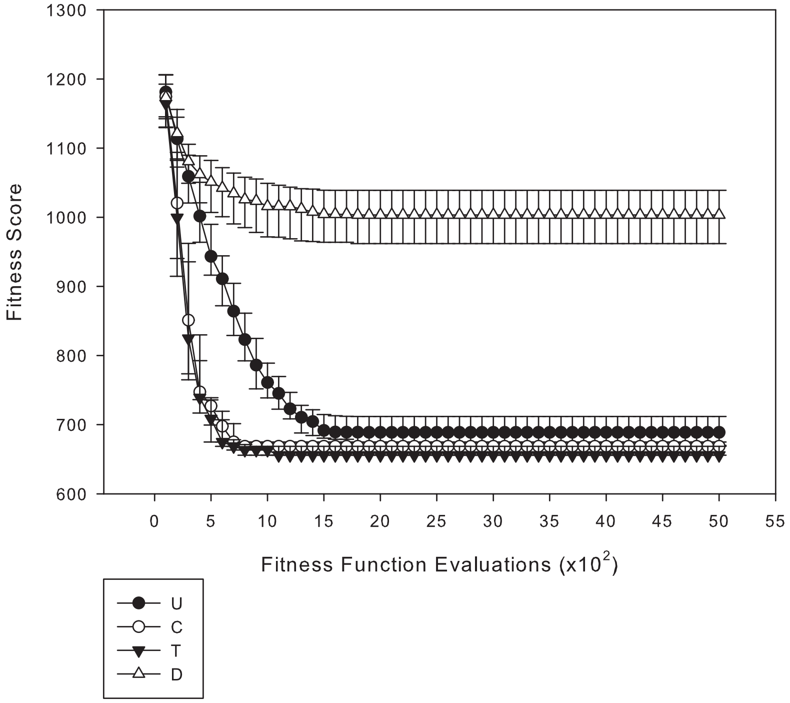
We now repeat the previous evaluation for a steady state replacement strategy. Figures 8.34 to 8.36 shows the fitness scores and associated intervention usage for the intermediate parameter settings, a penalty of 35 points per intervention and a steady state replacement strategy.

Figure 8.34 shows the intervention usage and fitness scores when p_m is 0. As was the case with steady state replacement and a penalty of 20 per intervention, UC settles to schedules with more interventions under steady state replacement than generational replacement when p_m is 0. This results in UC producing less optimal scores under steady state replacement. DUC uses 17 interventions under both replacement strategies and produces similar scores. TInSSel also uses a similar number of interventions under both replacement strategies. CalEB settles on 2 interventions for steady state replacement as opposed to the 3 used in schedules under generational replacement but returns scores of a similar fitness under both replacement approaches.

Figure 8.35 shows the fitness scores and associated intervention usage when p_m is increased to 0.005. UC, CalEB and TInSSel display similar trends under both generational and steady state replacement for these parameter settings. As with a penalty of 20 per intervention, DUC is quicker to reduce intervention levels under steady state replacement as opposed to generational replacement. Under steady state replacement, DUC reduces to 3 interventions after 2,700 FFEs, whereas this takes approximately a 1,000 more FFEs under generational replacement.

Figure 8.36 show the fitness scores and intervention usage when p_m is further increased to 0.05. For this mutation level, similar trends are demonstrated by each of the crossover approaches with steady state replacement as under generational replacement.

Figure 8.34: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$,$P = 35$ and $R = ss$



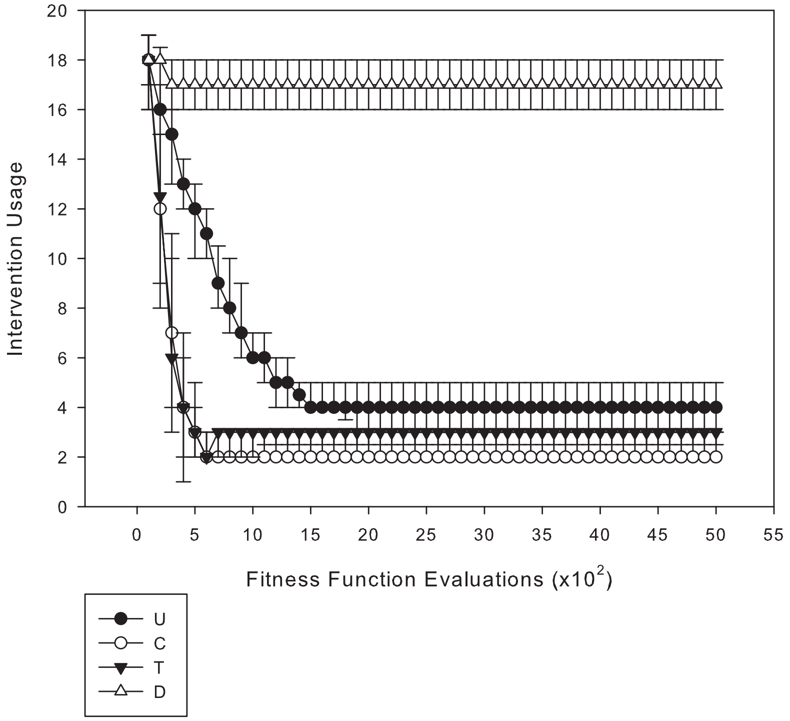


Figure 8.35: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 35$ and $R = ss$

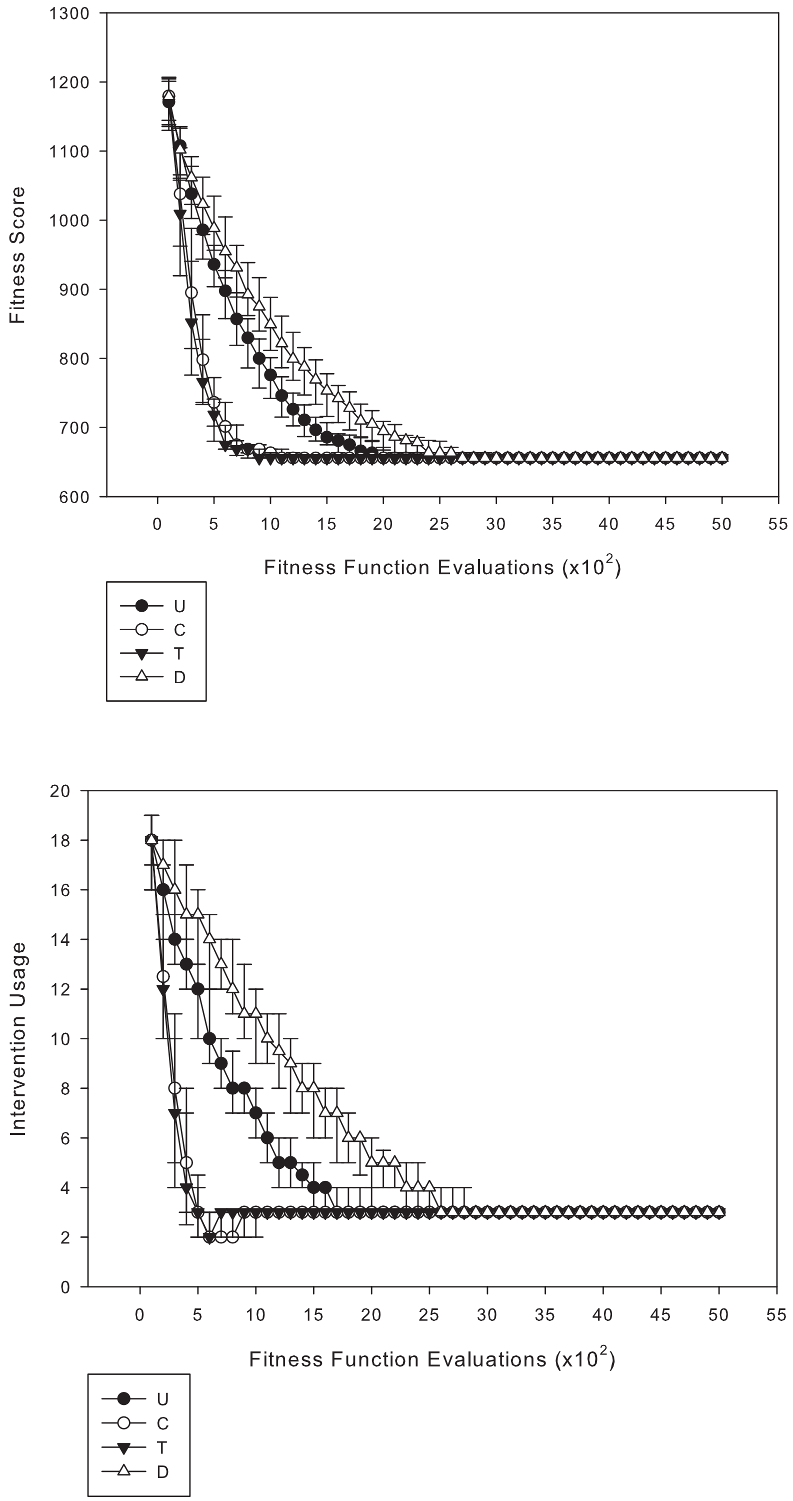
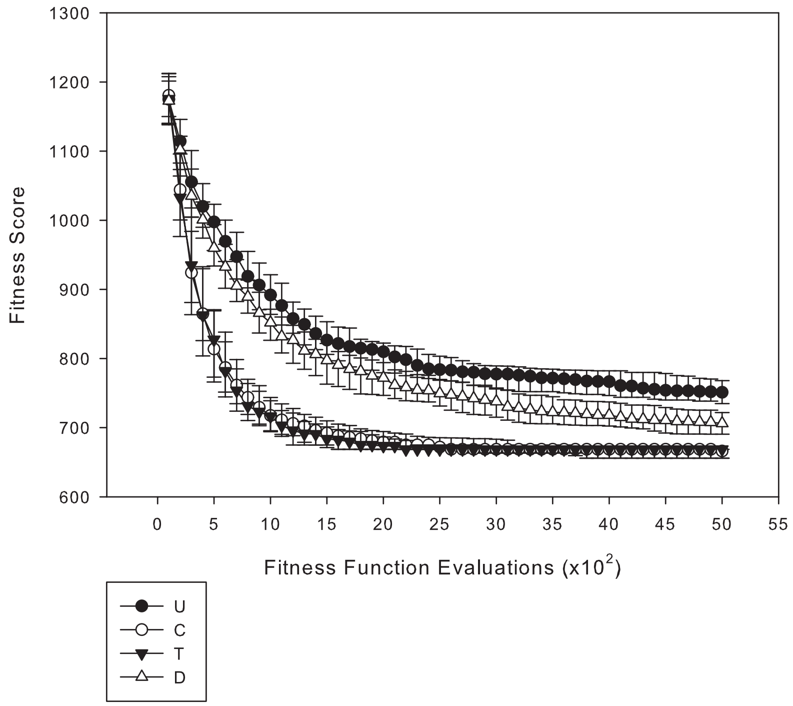
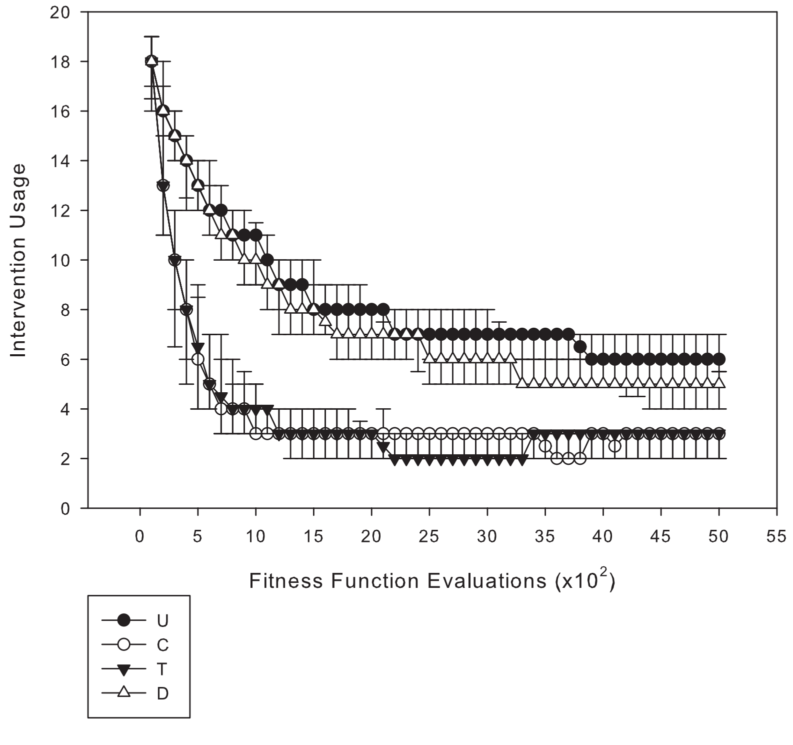


Figure 8.36: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 35$ and $R = ss$

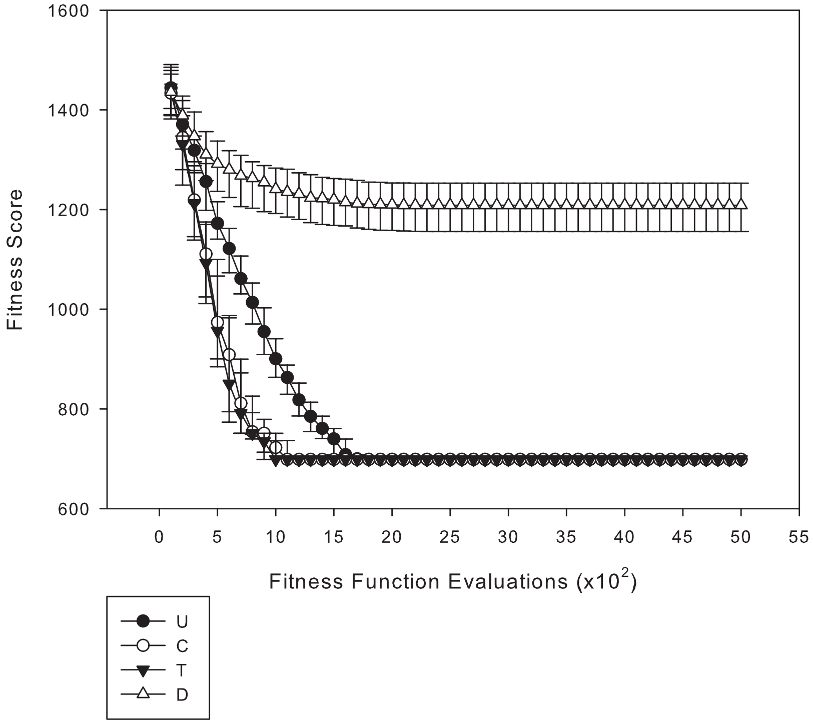




8.1.5 50 penalty points per intervention

The final bio-control experiment describes how CalEB, TInSSel, DUC and UC compare when the intervention penalty is further increased to 50 points per intervention. Figures 8.37 to 8.39 show the fitness scores and associated intervention usage for intermediate parameter settings, a penalty of 50 points per intervention and p_m levels of 0, 0.005 and 0.05 respectively.

Figure 8.37: Results for $N = 100, p_c = 0.9, p_m = 0,P = 50$ and $R = g$



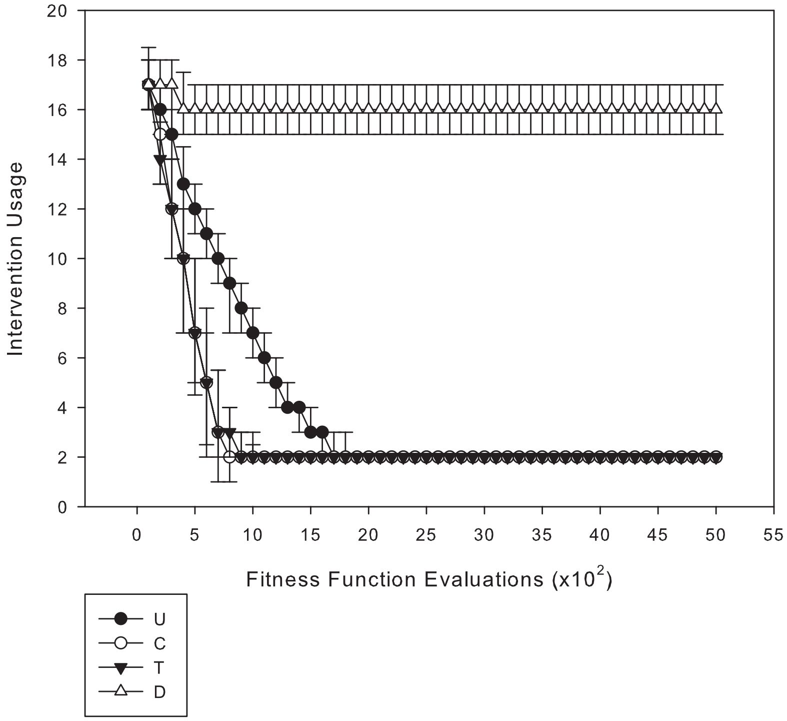
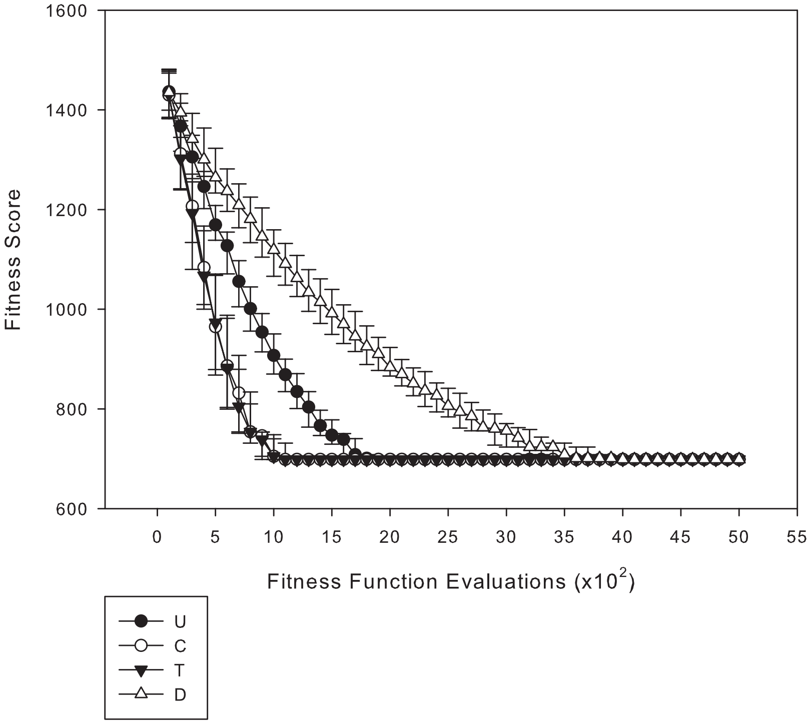


Figure 8.38: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 50$ and $R = g$



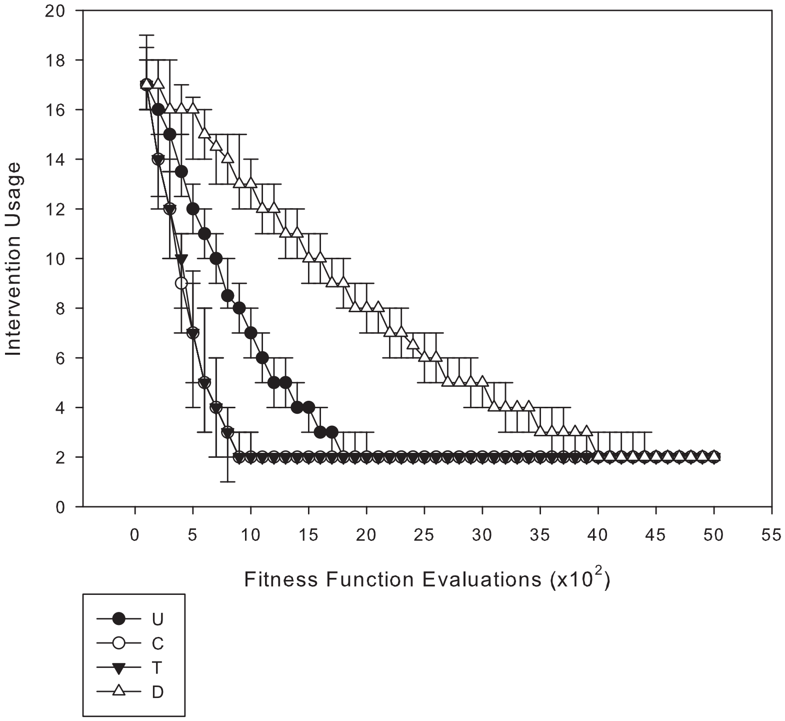
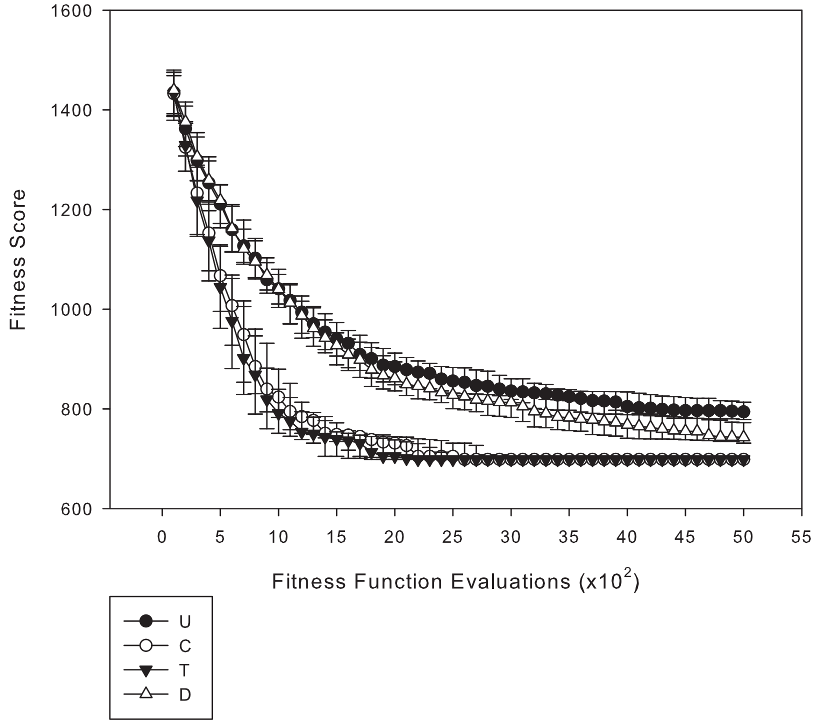


Figure 8.39: Results for $N = 100, p_c = 0.9, p_m = 0.05, P = 50$ and $R = g$



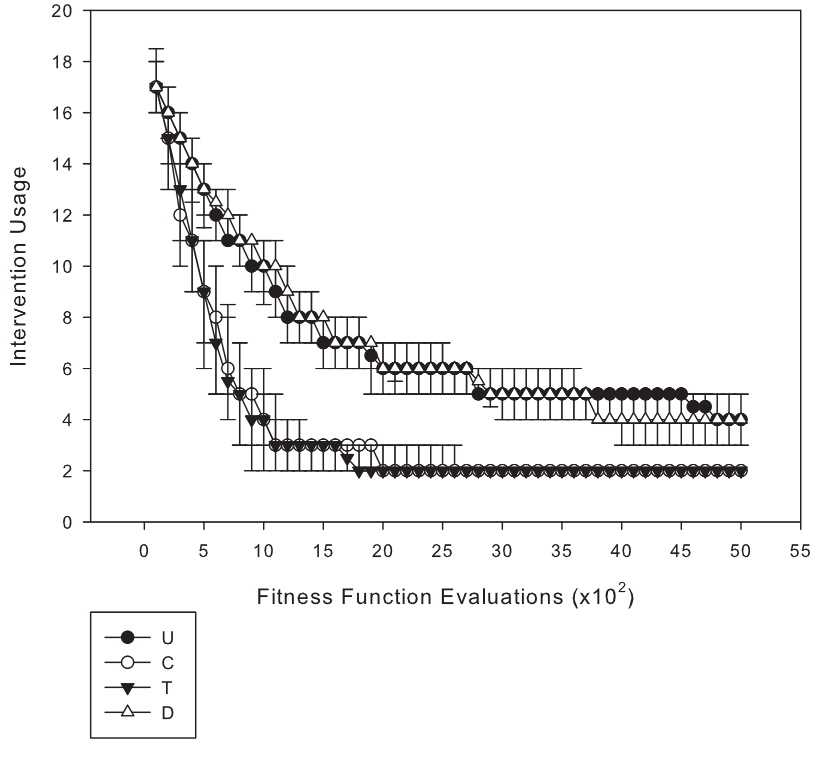


Figure 8.37 shows the fitness scores and associated intervention usage for the intermediate parameter setting in the absence of mutation. As with lower intervention penalties, a direct correlation appears between the number

of interventions used in schedules and the fitness scores returned. CalEB and TInSSel rapidly decrease in intervention usage and this is mirrored by a rapid improvement in fitness scores. UC displays a steady decrease in intervention usage in schedules, which is reflected by the gradual improvement in fitness scores. DUC converges to 16 interventions as opposed to the 2 settled on by the other approaches. This accounts for the difference in fitness scores between DUC and the other approaches for this mutation level.

Figure 8.38 shows the effect of increasing p_m from 0 to 0.005. UC, CalEB and TInSSel reduce in interventions at approximately the same rate as with a p_m of 0. This results in these approaches returning similar fitness scores. DUC requires approximately 4,000 FFEs to reduce the intervention level to the same level as the other approaches, and it is at this point that all four approaches return similar fitness scores.

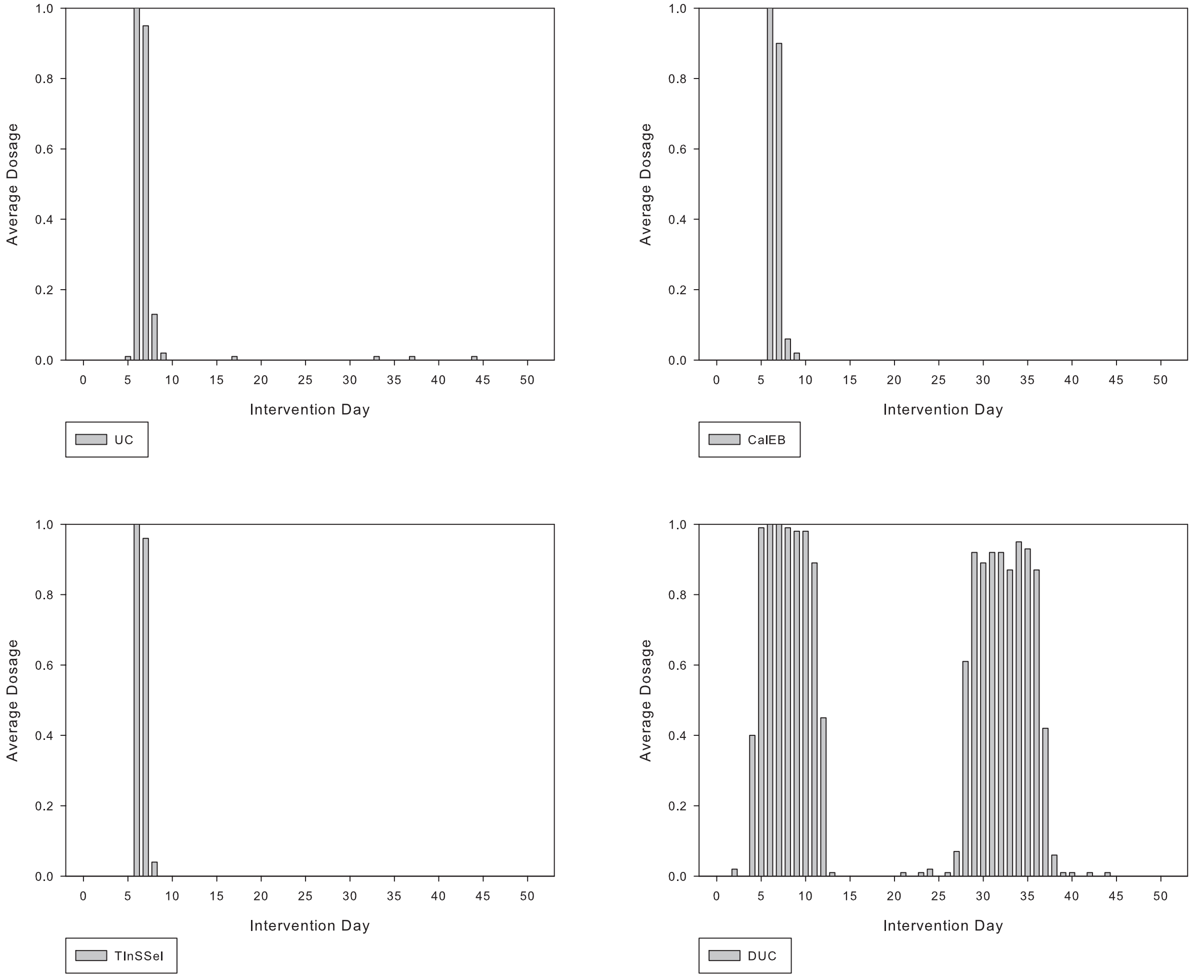
Figure 8.39 shows the fitness scores and intervention usage when p_m is further increased to 0.05. As with lower penalties, UC and DUC produce comparable results and use a similar number of interventions. CalEB and TInSSel use fewer interventions and return fitter scores than both UC and DUC.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the directed intervention crossover approaches, in comparison to UC, under a generational replacement strategy for a penalty of 50 points per intervention.

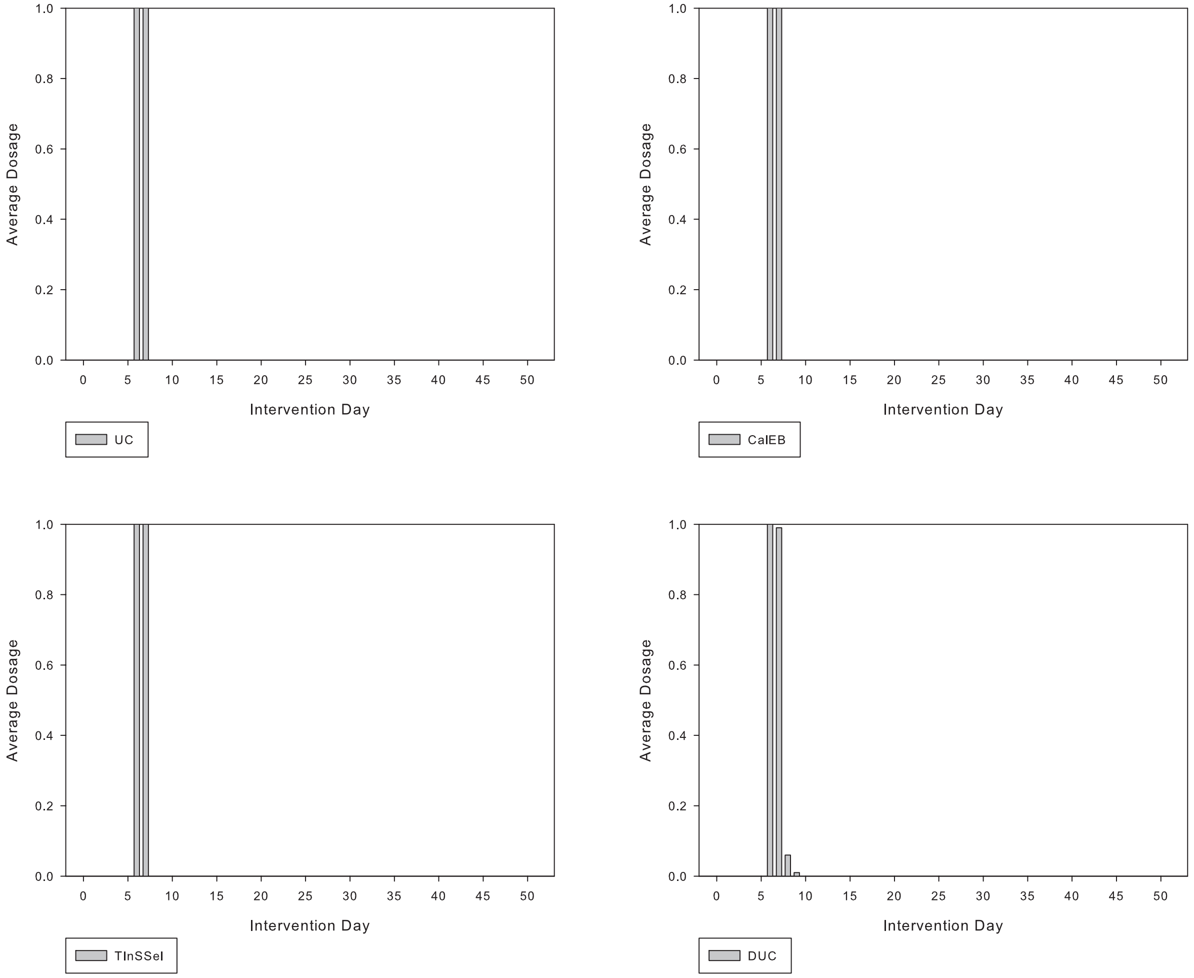
In the case of no mutation, as shown in Figure 8.37, DUC is shown to use more interventions than the other crossover techniques, with UC, CalEB and TInSSel using the same number. From the statistical analysis undertaken in Appendix E.13, after 5,000 FFEs, CalEB, TInSSel and UC all statistically outperform DUC for this mutation level. The average placement of the interventions relating to these findings are shown in Figure 8.40. This shows that while UC, CalEB and TInSSel mainly focus two interventions, at days 6 and 7, DUC uses many more. These extra interventions appear to be costing more in penalty values than they are gaining in sciarid larvae reduction, hence the lesser fitness scores for DUC for this mutation rate.

Figure 8.40: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 50$ and $R = g$



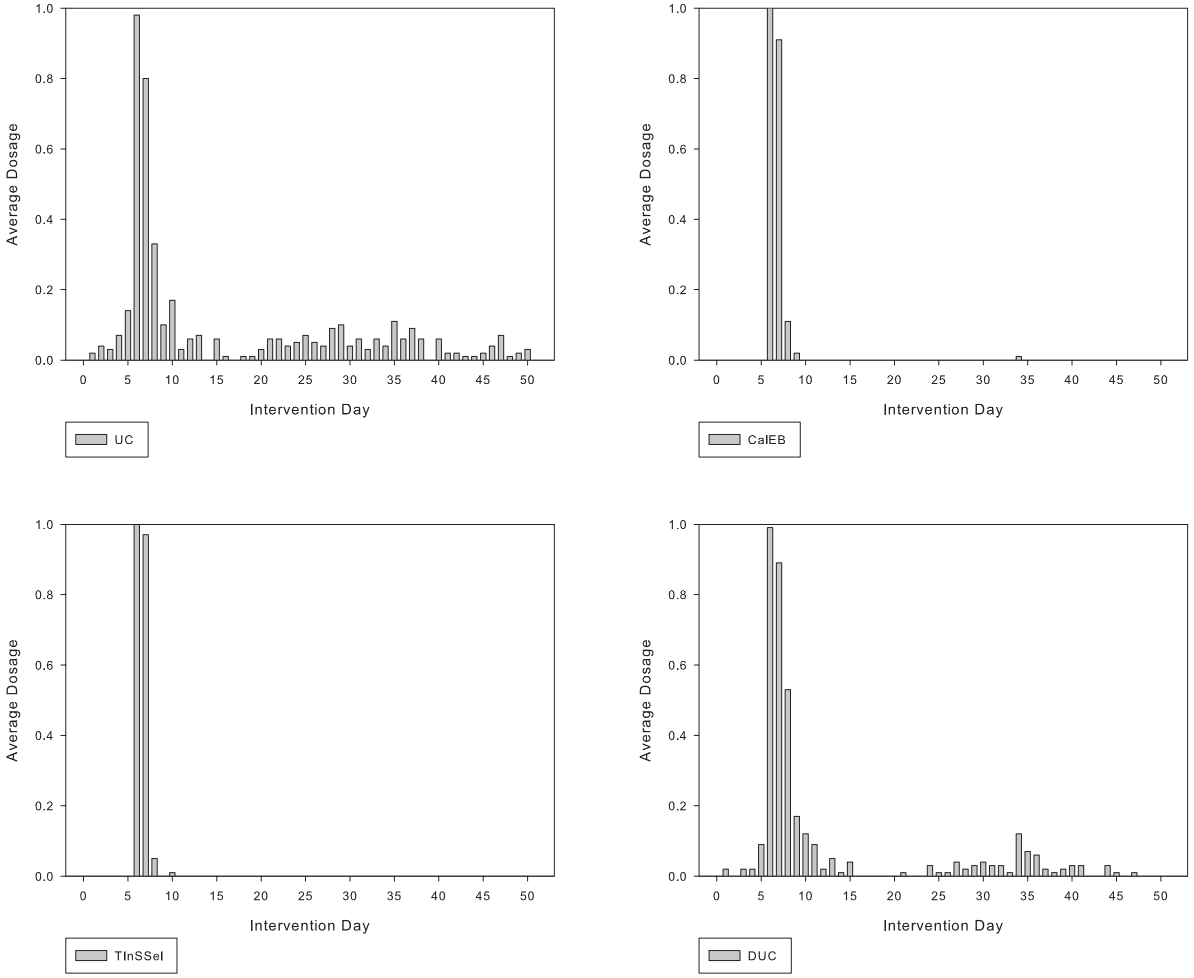
When p_m is increased to 0.005, as shown in Figure 8.38, each of the crossover approaches settle to the same number of interventions and return similar scores after 5,000 FFEs. From the statistical analysis undertaken in Appendix E.14, after 5,000 FFEs, there is no statistical difference between approaches for this mutation level. The average placement of the interventions relating to these findings are shown in Figure 8.41. This shows that each of the approaches focus two interventions, one on day 6 and one on day 7 of the schedule.

Figure 8.41: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 50$ and $R = g$



As p_m is further increased to 0.05, as shown in Figure 8.39, UC and DUC settle to schedules using more interventions than both CalEB and TInSSel. From the statistical analysis undertaken in Appendix E.15, after 5,000 FFEs, each of the directed intervention approaches are shown to statistically outperform UC for this mutation level, with CalEB and TInSSel both also outperforming DUC.The average placement of the interventions relating to these findings are shown in Figure 8.42. This shows that both CalEB and TInSSel mainly focus on interventions on days 6 and 7, with UC and DUC also displaying other interventions throughout the schedule. As DUC has less extra interventions distributed throughout the schedule than UC, this explains why DUC is better than UC for this mutation level.

Figure 8.42: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 50$ and $R = g$



Steady state replacement

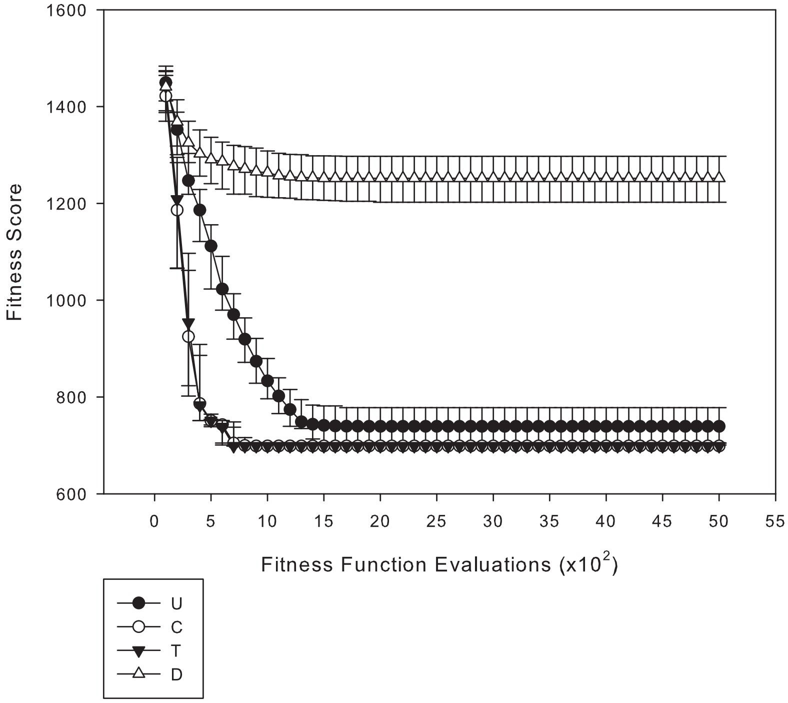
The same experiment will now be reviewed for a steady state replacement strategy. Figures 8.43 to 8.45 shows the fitness scores and associated intervention usage graphs for each approach using intermediate parameter settings, a penalty of 50 points per intervention and a steady state replacement strategy.

Figure 8.43 shows the fitness scores and intervention usage for steady state replacement when p_m is 0. Both CalEB and TInSSel perform in a similar way under steady state as with generational replacement both in terms of intervention usage and fitness scores. DUC settles on approximately 17 interventions for steady state replacement, which is one more intervention than used in schedules under generational replacement. Due to this, there is a slight decline in the fitness scores found by DUC between replacement strategies. UC also uses more interventions under steady state replacement. With a steady state approach, UC settles to 3 interventions, rather than the 2 used under generational replacement. As with DUC, this slight increase in intervention usage results in less fit scores being produced by UC under steady state replacement, when p_m is 0.

Figure 8.44 details the fitness scores and associated intervention usage when p_m is increased to 0.005. As with lesser intervention penalties, DUC is quicker to reduce in interventions under steady state replacement than with generational replacement. UC, CalEB and TInSSel display similar trends for both generational and steady state replacement.

Figure 8.45 shows the fitness scores and intervention usage when p_m is further increased to 0.05. As with the lesser intervention penalties, the behavior of the crossover approaches under steady state replacement is similar to that under the generational equivalent.

Figure 8.43: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$,$P = 50$ and $R = ss$



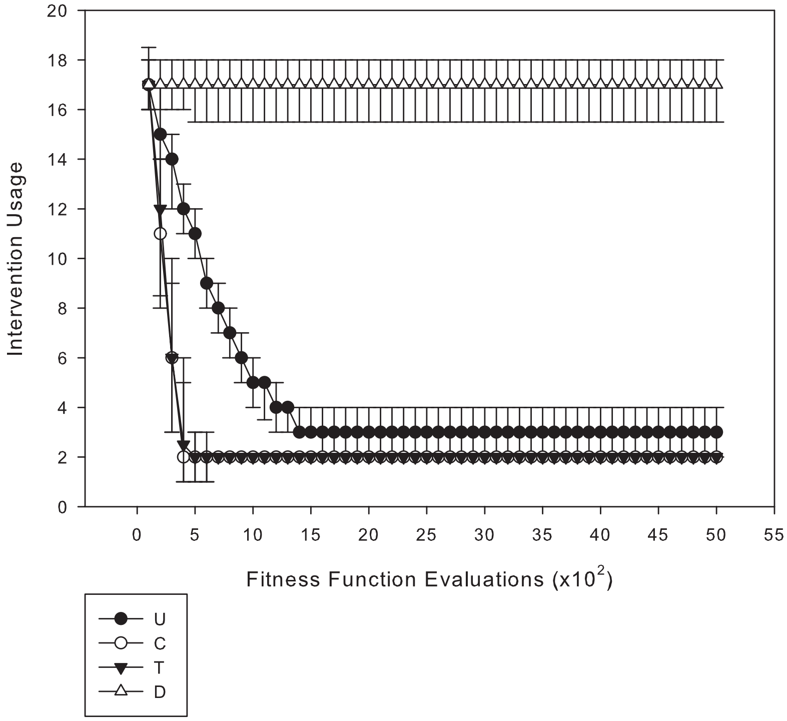
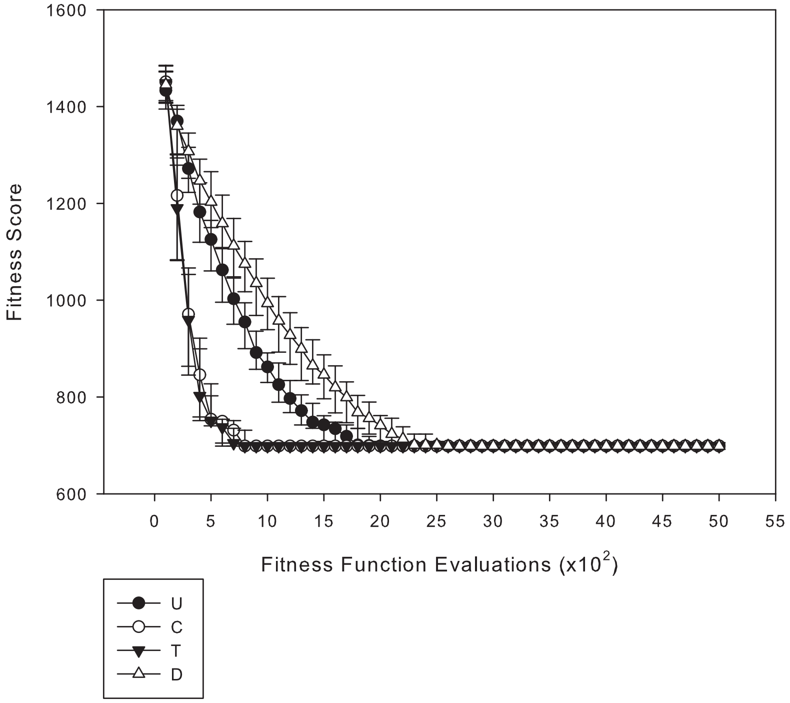


Figure 8.44: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 50$ and $R = ss$



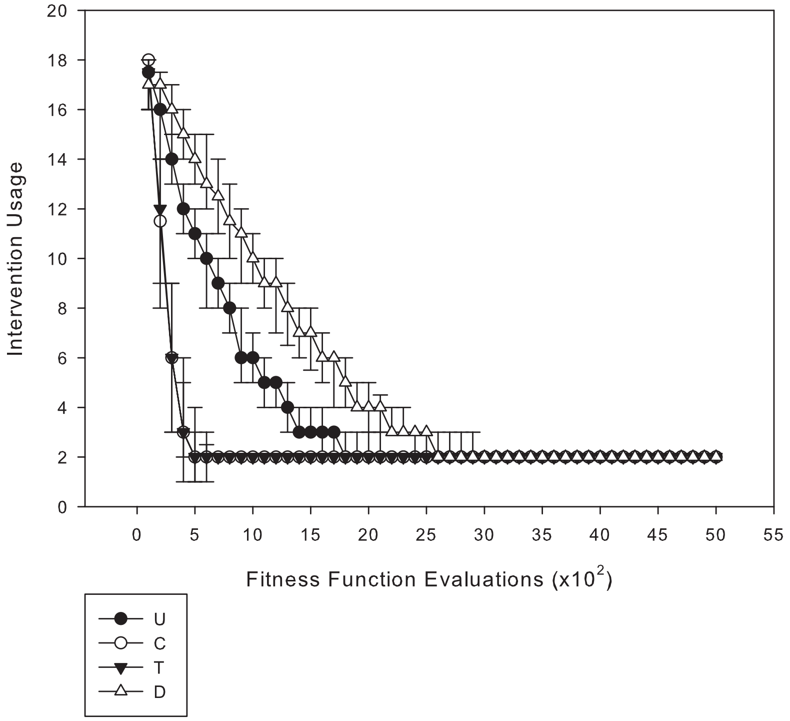
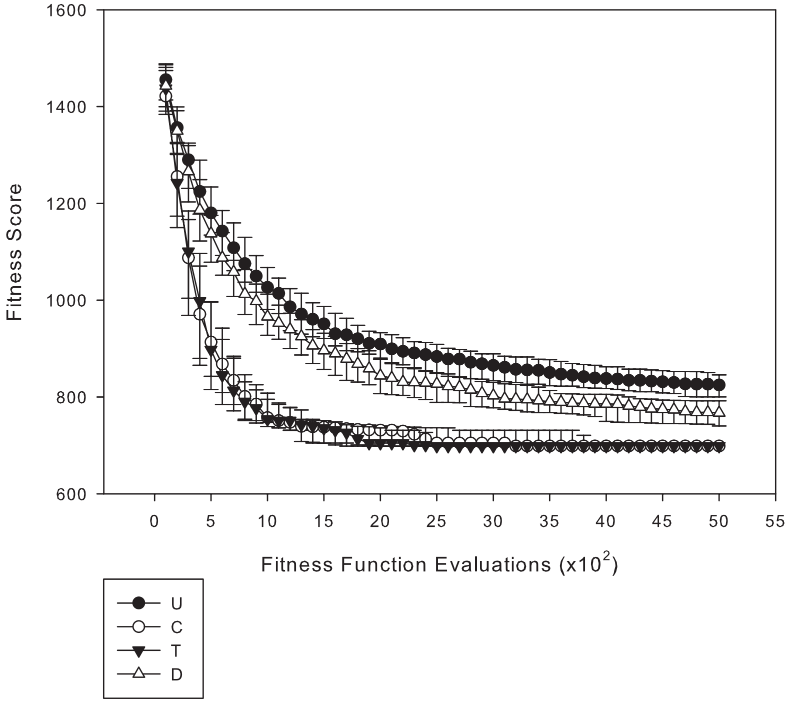
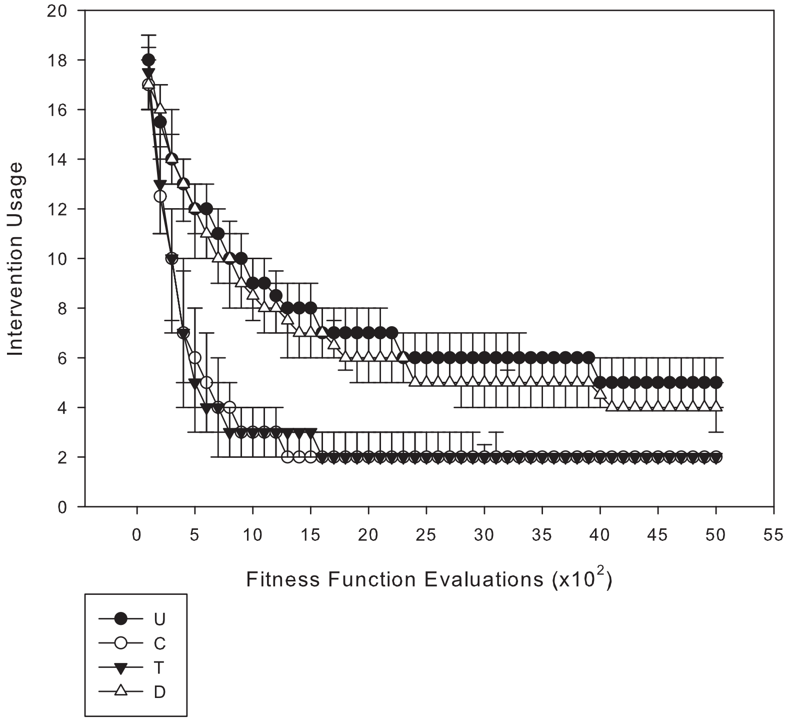


Figure 8.45: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 50$ and $R = ss$





8.1.6 Summary of bio-control scheduling experiment

The previous experiments have reviewed CalEB, TInSSel and DUC, in comparison to UC, for a range of intervention penalties. This section draws together the salient points from each of these experiments, to demonstrate the effectiveness of the directed intervention crossover techniques.

In order to ascertain if there is any statistical significant differences between approaches for this problem, the Kruskal-Wallis test results for the intermediate test cases were calculated. These are shown in Tables E.1 to E.15. This shows the asymptotic significance (AS) value for each FFE for the approaches and the mean rank scores for each of the crossover techniques. The AS value represents the likelihood of a difference existing between samples. If the AS value is less than 0.01 this means that there is a statistically significant difference between at least 2 of the samples at the 99% confidence level. By using Equation 6.1, it is possible to establish which groups are statistically significantly different from which other groups.

As described in Section 6.9.1, if the absolute difference between mean ranks is greater than 47.99, this indicates a statistically significant difference between samples. This value is derived from a comparison of 4 samples, with 100 cases per sample and an α value of 0.01. For clarity, Appendix J shows which crossover approaches are significantly different over the range of FFEs.

Table 8.1 shows an excerpt from one of these tables (Table J.1). This shows the samples with a statistically significant difference for the first 300 FFEs of the experiment with U, C, T and D representing UC, CalEB, TInSSel and DUC respectively. This shows that there is no statistically significant difference between UC and DUC or CalEB and DUC for these FFEs. CalEB outperforms UC after 300 FFEs and TInSSel is the best performing approach for this FFE range, outperforming UC for 200 and 300 FFEs and both CalEB and DUC after 300 FFEs.

Table 8.1: Sample data from Table J.1

FFE(s)	**U/C**	**U/T**	**U/D**	**C/T**	**C/D**	**T/D**
100						
200		T				
300	C	T		T		T

Each of the intervention penalties will be reviewed in turn, with Section 8.1.7 describing the trends demonstrated across the range of penalty values.

Penalty of 0 points per intervention

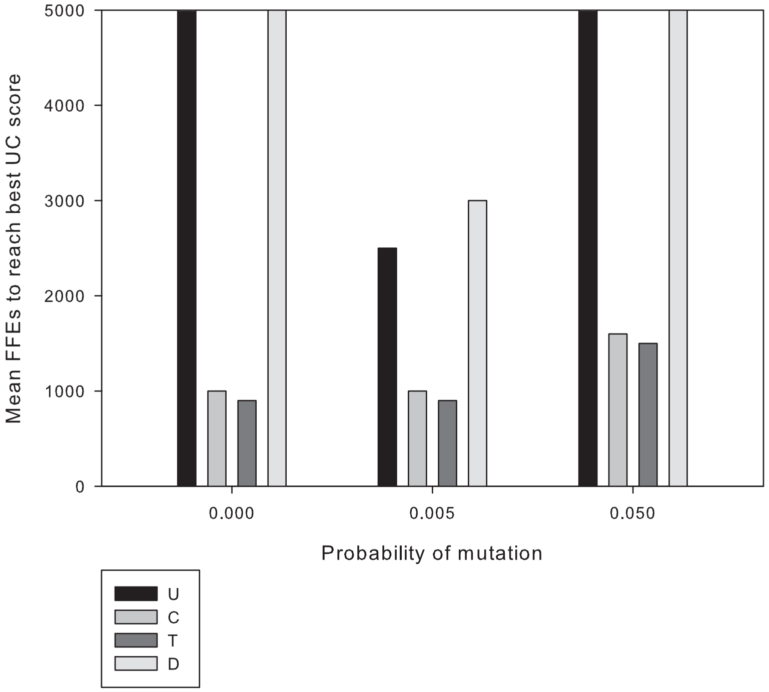


Figure 8.46: FFEs to reach the best score found by UC (355) with a penalty of 0 points per intervention

Section 8.1.1 described the scores found by the directed intervention crossover approaches, in comparison to UC, for a penalty of 0 points per intervention. This showed that both CalEB and TInSSel were faster to increase intervention levels compared with UC and DUC, resulting in fitter scores being returned for these directed intervention approaches.

The statistical analysis detailed in Appendix E.1.1 shows that there is a statistical difference between at least two of the crossover approaches from 300 to 5,000 FFEs for each of the mutation levels. For clarity, the crossover approaches which are statistically different to each other are described in Appendix J.1.1. This shows that for each of the mutation levels, CalEB and TInSSel statistically outperform both UC and DUC for almost all FFEs under observation. DUC is shown to be outperformed by UC for around two thirds of the FFEs when p_m is 0 or 0.005, but is shown to outperform UC from 1,700 to 5,000 FFEs when the mutation rate is increased to 0.05. TInSSel is the only approach that is never outperformed by any other technique, outperforming CalEB for a couple of FFEs early in the experiments.

Over the range of mutation levels, the best score found by UC for a penalty of 0 points per intervention was 355. This was found when the mutation rate was 0.005. Figure 8.46 shows the FFEs required by each of the crossover approaches to find this best score returned by UC, over the range of mutation levels. This shows that in the absence of mutation, both UC and DUC do not find this score, even after 5,000 FFEs. CalEB finds a score of 355 after 1,000 FFEs, with TInSSel requiring only 900 FFEs.

When the mutation rate was increased to 0.005, Figure 8.46 shows that all of the crossover approaches find a score of 355. UC requires 2,500 FFEs, with DUC, CalEB and TInSSel using 3,000, 1,000 and 900 FFEs respectively. When the mutation rate is further increased to 0.05, as with the case of no mutation, both UC and DUC do not find a score of 355, even after 5,000 FFEs. For this higher mutation rate, there is a slight increase in FFEs required by both CalEB and TInSSel, due to the increase in noise introduced via mutation, with CalEB and TInSSel using 1,600 and 1,500 FFEs respectively.

Penalty of 5 points per intervention

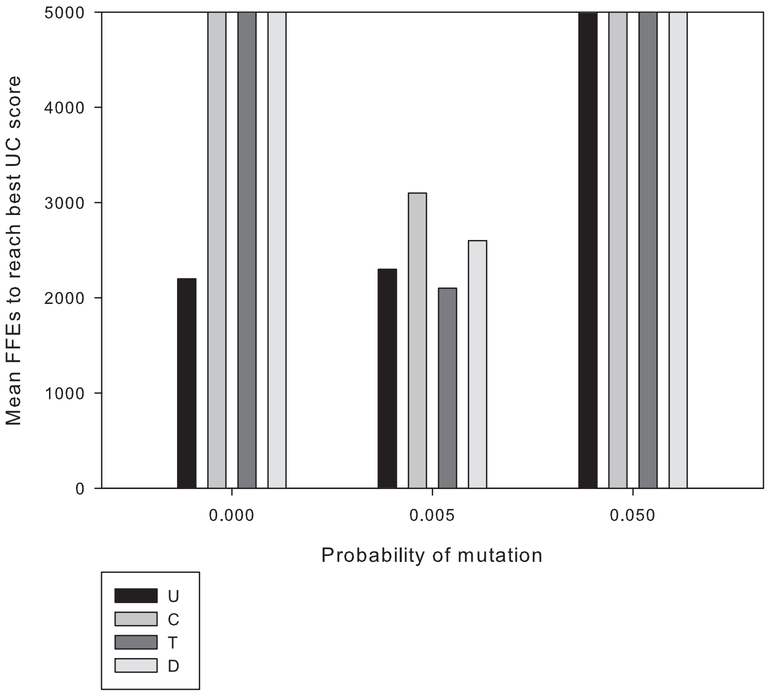


Figure 8.47: FFEs to reach the best score found by UC (476) with a penalty of 5 points per intervention

Section 8.1.2 described the scores found by the directed intervention crossover approaches, in comparison to UC, for a penalty of 5 points per intervention. This showed that while both CalEB and TInSSel were faster to decrease in intervention levels than UC and DUC, TInSSel produced fitter scores than UC and DUC, whereas CalEB did not. It was noted that as CalEB and TInSSel use the same offspring intervention calculation, a stochastic approach for choosing offspring genes was better for this problem than one which picks genes in a uniform distribution over time.

The statistical analysis detailed in Appendix E.1.2 shows that there is a statistical difference between at least two of the crossover approaches from 200 to 5,000 FFEs for each of the mutation levels. As 100 FFEs represents the initial scoring of the randomly created population, one would not expect there to be statistical differences between approaches at this point. The crossover approaches which are statistically different to each other are described in Appendix J.1.2. This shows that unlike the case of 0 penalty points per intervention, CalEB is outperformed by each of the other approaches for at least some FFEs, regardless of the mutation rate. In line with the previous penalty, UC outperforms DUC for a range of FFEs when p_m is 0 or 0.005 but DUC outperforms UC from 1,700 to 5,000 FFEs when the mutation rate is further increased to 0.05. TInSSel is again shown to be the only crossover approach not outperformed by any other technique, regardless of the mutation rate.

Over the range of mutation levels, the best score found by UC for a penalty of 5 points per intervention was 476. This was found when the mutation rate was 0 or 0.005. Figure 8.47 shows the FFEs required by each of the crossover approaches to find this best score returned by UC, over the range of mutation levels. This shows that in the absence of mutation, only UC returns this score, but as stated above, the score found by TInSSel is still close enough to not be statistically different.

When the mutation rate was increased to 0.005, Figure 8.47 shows that all of the crossover approaches find a score of 476. UC requires 2,100 FFEs, with DUC, CalEB and TInSSel using 2,600, 3,100 and 2000 FFEs respectively. When the mutation rate is further increased to 0.05, each of the approaches are shown to be disrupted by the noise introduced via mutation, with none of the crossover approaches finding scores of 476, even after 5,000 FFEs.

Penalty of 20 points per intervention

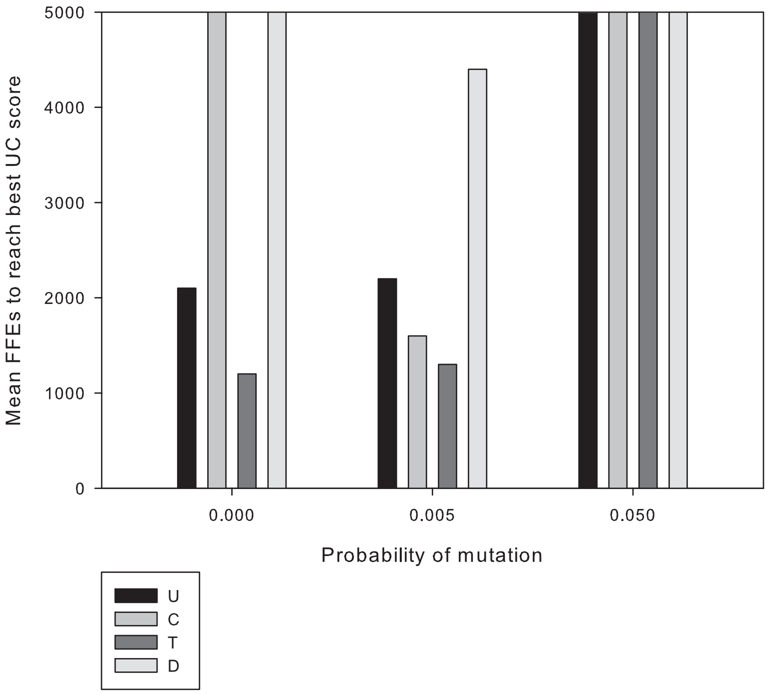


Figure 8.48: FFEs to reach the best score found by UC (603) with a penalty of 20 points per intervention

Section 8.1.3 described the scores found by the directed intervention crossover approaches, in comparison to UC, for a penalty of 20 points per intervention. This showed that both CalEB and TInSSel were faster to decrease in intervention levels than UC and DUC, resulting in fitter scores being returned in fewer FFEs.

The statistical analysis detailed in Appendix E.1.3 shows that there is a statistical difference between at least two of the crossover approaches from 200 to 5,000 FFEs for each of the mutation levels. The crossover approaches which are statistically different to each other are described in Appendix J.1.3. This shows that as with a penalty of 0 points per intervention, both CalEB and TInSSel return significantly better scores than UC and DUC for a range of FFEs, regardless of the mutation rate. In line with the previous experiments, UC is shown to outperform DUC when p_m is 0 or 0.005, but when the mutation rate is further increased to 0.05, DUC outperforms UC for approximately two-thirds of the FFEs. Once again, TInSSel is the only approach to not be outperformed by any other technique, regardless of FFE observed or mutation rate.

Over the range of mutation levels, the best score found by UC for a penalty of 20 points per intervention was 603. This was found when the mutation rate was 0 or 0.005. Figure 8.48 shows the FFEs required by each of the crossover approaches to find this best score returned by UC, over the range of mutation levels. This shows that in the absence of mutation, only UC and TInSSel find this score, with UC doing so after 2,100 FFEs and TInSSel 1,200 FFEs.

When the mutation rate was increased to 0.005, Figure 8.48 shows that all of the crossover approaches find a score of 603. UC requires 2,200 FFEs, with DUC, CalEB and TInSSel using 4,400, 1,600 and 1,300 FFEs respectively. When the mutation rate is further increased to 0.05, each of the approaches are shown to be disrupted by the noise introduced via mutation, with none of the crossover approaches finding scores of 603, even after 5,000 FFEs.

Penalty of 35 points per intervention

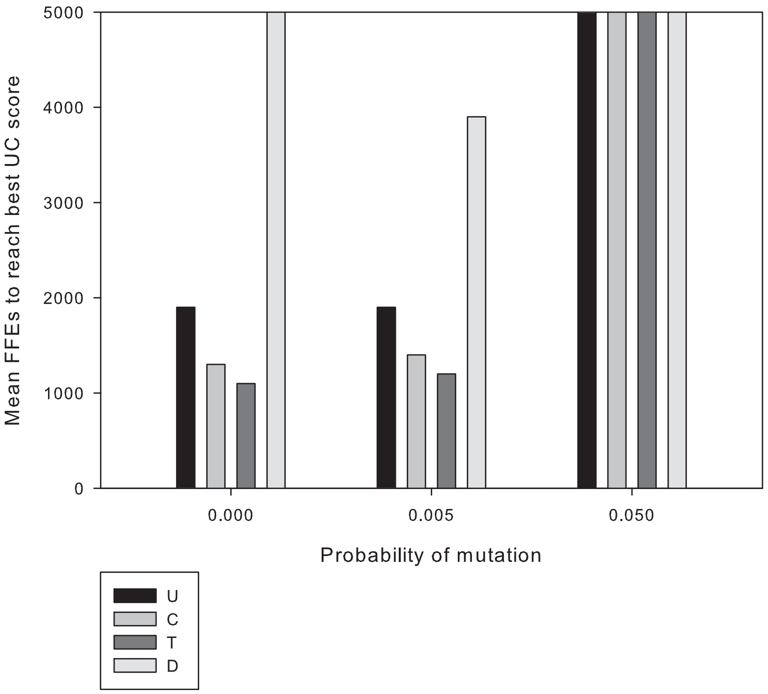


Figure 8.49: FFEs to reach the best score found by UC (655) with a penalty of 35 points per intervention

Section 8.1.4 described the scores found by the directed intervention crossover approaches, in comparison to UC, for a penalty of 35 points per intervention. This showed that as with a penalty of 20 points per intervention, both CalEB and TInSSel were faster to decrease in intervention levels than UC and DUC, resulting in fitter scores being returned in fewer FFEs.

The statistical analysis detailed in Appendix E.1.4 shows that there is a statistical difference between at least two of the crossover approaches from 200 to 5,000 FFEs for each of the mutation levels. The crossover approaches which are statistically different to each other are described in Appendix J.1.4. This shows that as with a penalty of 20 points per intervention, both CalEB and TInSSel return significantly better scores than UC and DUC for a range of FFEs, regardless of the mutation rate. As demonstrated with lower intervention penalties, UC is shown to outperform DUC when p_m is 0 or 0.005, but when the mutation rate is further increased to 0.05, DUC outperforms UC for a range of FFEs. As with all previous intervention penalties, TInSSel is the only approach to not be outperformed by any other technique, however for this intervention penalty, there is no statistical difference between CalEB or TInSSel for any FFE.

Over the range of mutation levels, the best score found by UC for a penalty of 35 points per intervention was 655. This was found when the mutation rate was 0 or 0.005. Figure 8.49 shows the FFEs required by each of the crossover approaches to find this best score returned by UC, over the range of mutation levels. This shows that in the absence of mutation, DUC is the only approach not to find a score of this fitness, with UC, CalEB and TInSSel finding such scores using 1,900, 1,300 and 1,100 FFEs respectively.

When the mutation rate was increased to 0.005, Figure 8.49 shows that all of the crossover approaches find a score of 655. UC requires 1,900 FFEs, with DUC, CalEB and TInSSel using 3,900, 1,400 and 1,200 FFEs respectively. When the mutation rate is further increased to 0.05, none of the crossover approaches find scores of 655, even after 5,000 FFEs.

Penalty of 50 points per intervention

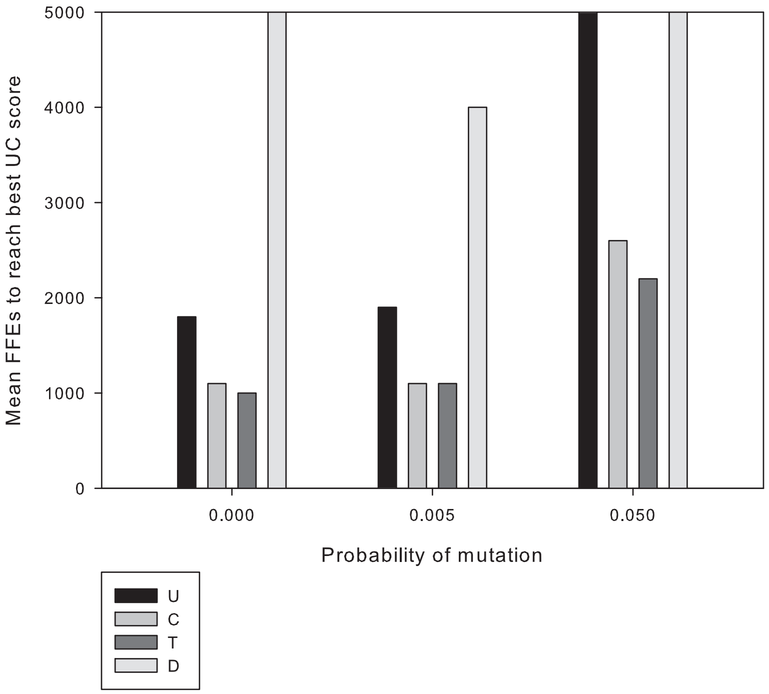


Figure 8.50: FFEs to reach the best score found by UC (698) with a penalty of 50 points per intervention

Section 8.1.5 described the scores found by the directed intervention crossover approaches, in comparison to UC, for a penalty of 50 points per intervention. This showed that as with a penalties of 20 and 35 points per intervention, both CalEB and TInSSel were faster to decrease in intervention levels than UC and DUC, resulting in fitter scores being returned in fewer FFEs.

The statistical analysis detailed in Appendix E.1.5 shows that there is a statistical difference between at least two of the crossover approaches from 200 to 5,000 FFEs for each of the mutation levels. The crossover approaches which are statistically different to each other are described in Appendix J.1.5. This shows that as with a penalty of 20 or 35 points per intervention, CalEB and TInSSel return significantly better scores than UC or DUC for a range of FFEs. As with the trend observed for lesser intervention penalties, while UC outperforms DUC for mutation rates of 0 or 0.005, when p_m is further increased to 0.05, DUC is shown to return scores significantly better to those found by UC. As with a penalty of 35 points per intervention, there is no statistical difference between CalEB or TInSSel regardless of FFE observed or mutation rate.

Over the range of mutation levels, the best score found by UC for a penalty of 50 points per intervention was 698. This was found when the mutation rate was 0 or 0.005. Figure 8.50 shows the FFEs required by each of the crossover approaches to find this best score returned by UC, over the range of mutation levels. This shows that in the absence of mutation, DUC is the only approach not to find a score of this fitness, with UC, CalEB and TInSSel finding such scores using 1,800, 1,100 and 1,000 FFEs respectively.

When the mutation rate was increased to 0.005, Figure 8.50 shows that all of the crossover approaches find a score of 698. UC requires 1,900 FFEs, with DUC, CalEB and TInSSel using 4,000, 1,100 and 1,100 FFEs respectively. When the mutation rate is further increased to 0.05, only CalEB and TInSSel find scores of 698, with CalEB doing so after 2,600 FFEs and TInSSel 2,200 FFEs.

8.1.7 Summary

The previous sections have reviewed the directed intervention crossover approaches, in comparison to UC, over a range of intervention penalties. This has outlined the statistical differences between approaches and the number of FFEs required to find the best score found by UC.

Over the range of parameter settings and intervention penalties, TInSSel is the only crossover approach that is not statistically outperformed by any other technique. CalEB is the next most successful technique and is only outperformed by UC or DUC when there is a penalty of 5 points per intervention. As stated previously, as CalEB and TInSSel use the same calculation for offspring intervention levels, for this low penalty, picking genes in a uniform distribution is not as effective as picking offspring genes in a stochastic way.

DUC is outperformed by all 3 of the other approaches when p_m is 0 or 0.005. When p_m is increased to 0.05, DUC finds solutions of a lesser fitness than both CalEB and TInSSel, but tends to solutions significantly better than those returned by UC. This shows that for a high mutation level, i.e. when there is a lot of noise, even a simple offspring intervention target such as that used by DUC is more effective than UC.

From inspecting the crossover approaches over a range of intervention penalties and parameter values, certain deductions can be made:

- As TInSSel is always at least as good as the other approaches, it would appear to be the crossover technique of choice for the bio-control scheduling problem, regardless of intervention penalty and parameter values.

- Both CalEB and TInSSel use the same intervention sizing technique for children and only differ in the intervention selection process. This means that for this problem, as CalEB never outperforms TInSSel, but is outperformed by TInSSel at times, enforcing a uniform distribution for selection of genetic material is less efficient than a purely stochastic selection. This is most apparent with a penalty of 5 per intervention.

- DUC uses the same intervention selection mechanism as TInSSel, but has a simpler method for calculating the number of interventions to use in offspring. For every intervention penalty and parameter setting, DUC never outperforms TInSSel, but is often statistically outperformed by it. As mentioned in Section 8, both TInSSel and CalEB were expected to outperform DUC for this problem. This has been shown to be the case, demonstrating the benefit of using a window of selection around the fitter parent, as opposed to strictly using the number of interventions in the fitter parent as the target for this problem.

- When p_m is 0.05, a lot of genetic material is being added to the system, and thus there is a lot of noise present. In this situation, CalEB, TInSSel and DUC tend to outperform UC. This shows that even a simplistic target for the number of interventions, such as that used by DUC, is more efficient in the presence of large quantities of noise, than having no intervention target to aim for.

Both CalEB and TInSSel are robust to the replacement strategy used, returning similar scores under both generational and steady state replacement. UC tends to produce poorer scores under steady state replacement, when p_m is 0. As steady state replacement introduces new population members in smaller numbers than the complete population replacement used by the generational approach, in the absence of mutation, it would appear that there is not enough diversity being introduced for UC to exploit. This would explain the poorer scores produced by UC for this setting. When p_m is 0.005, UC, TInSSel and CalEB are generally unaffected by the replacement strategy used. DUC is quicker to reduce the number of interventions under steady state replacement as opposed to generational for this mutation level. As stated in Section 8.1.1, this could be attributed to the child solutions being immediately available for selection after insertion to the population earlier in the process with steady state replacement than with generational replacement.

As described above, TInSSel is the most efficient crossover approach under review for this problem. When compared to UC, TInSSel is shown to require far fewer FFEs to find similar scores, and TInSSel also statistically outperforms UC for many FFEs across the penalty range.

Figure 8.51 shows the percentage of improvement, in terms of both fewer FFEs required and statistically better scores found, through using TInSSel instead of UC. This shows the results for a mutation rate of 0.005, which was where each of the crossover approaches on average performed best. This shows that for a penalty of 0 points per intervention, TInSSel requires 64% fewer FFEs than UC to find the same score and that for 86% of the FFEs under review, TInSSel returns significantly better scores.

When the penalty was increased to 5 penalty points per intervention, the increase in performance demonstrated by TInSSel over UC is less than for a penalty of 0 points per intervention, but is still sizeable. For this penalty value, there was an 4.7% reduction in FFEs required for TInSSel to find the same score as UC and TInSSel statistically outperformed UC for 24% of all FFEs.

A further increase in penalty value, from 5 to 20, demonstrates a larger performance gain in using TInSSel when compared to UC. For this setting TInSSel finds the best score found by UC using 40% less FFEs. TInSSel also statistically outperforms UC for 40% of the FFEs under review.

For a penalty of 35 points per intervention, TInSSel requires 36% fewer FFEs to find the best score returned by UC. For this penalty setting the scores returned by TInSSel are statistically better than those found by UC for 34% of the FFEs. A further increase in penalty value, to 50 points per intervention, shows that TInSSel requires 42% less FFEs to discover the final value returned by UC. As with a penalty of 35 points per intervention, TInSSel finds statistically better scores for 34% of all FFEs.

The average gain in performance over all penalty values for a mutation rate of 0.005 is shown in Figure 8.51. This shows that over all 5 penalty levels, TInSSel requires 37% fewer FFEs to find the best score returned by UC. On average, TInSSel finds statistically better scores for 43% of all FFEs under review.

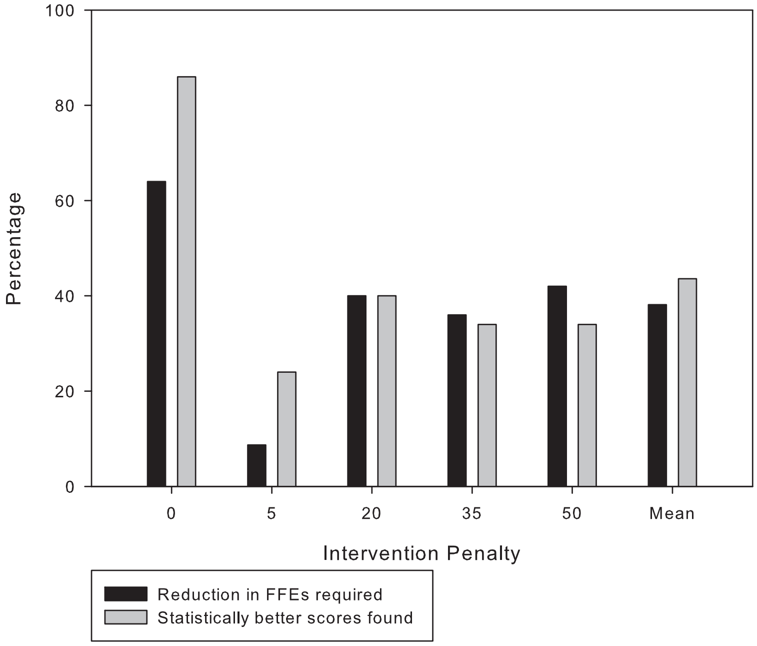


Figure 8.51: Performance improvement between UC and TInSSel when $p_m = 0.005$

8.1.8 Further analysis

The previous sections reviewed CalEB, TInSSel and DUC in comparison to UC across a range of intervention penalties using the intermediate parameter settings.

In order to further analyse the abilities of each of the crossover approaches, it is necessary to review these techniques across a range of parameter settings, outwith the intermediate set used in the previous sections.

Appendix C details the performance of CalEB, TInSSel, DUC and UC across a range of population sizes, crossover rates, mutation rates and intervention penalties. A review of the crossover approaches over this range of parameters are detailed in Appendix A.3. In this analysis over the range of parameter settings, each of the trends outlined in Section 8.1.6 are shown to hold across the range of parameter settings.

8.2 Evaluation of directed intervention crossover techniques for chemotherapy scheduling

The directed intervention crossover approaches, CalEB, TInSSel and DUC will now be reviewed for the task of chemotherapy treatment scheduling. As with the bio-control experiments detailed in Section 8.1, these techniques will be compared with UC.

The methodology used in this section is the same as that used for analysis of the traditional crossover approaches for chemotherapy scheduling in Section 7.2. This leads to the result being sampled over 40,000 FFE, at intervals of 1,000 FFEs, resulting in 40 data points for review. For each sample point, 100 runs of each crossover approach were recorded for each of the parameter combinations. The best scoring values for each run for each FFE are recorded.

As with the traditional crossover experiments detailed in Chapter 7, the crossover techniques are tested over a range of parameter settings. A complete set of results across the parameter combinations has therefore been collected and these can be seen in Appendix G. This Appendix contains the complete set of results showing the fitness scores against FFEs associated with these experiments across a range of population sizes, crossover rates and mutation rates.

As with previous experiments, the intermediate parameter settings for each approach will be reviewed in detail, rather than explicitly detailing each possible parameter combination. The following analysis will therefore review CalEB, TInSSel and DUC in comparison to UC, with a population of $N = 100$, crossover rate of $p_c = 0.9$ and mutation rates, p_m, of 0, 0.005 and 0.05. Inline with the previous crossover experiments, initial analysis for each intervention point will review the intermediate parameter settings with a generational replacement strategy. This will be followed by a statistical evaluation of these results, to ascertain if there is any statistically significant differences between crossover approaches. The average placement and dosage levels will be analysed. Intermediate parameter settings will then be reviewed under a steady state replacement strategy.

As stated previously, as the cancer chemotherapy treatment optimisation problem is a maximisation problem, higher scores are fitter than lower valued scores.

Figures 8.52 to 8.54 show the fitness score and intervention usage for the intermediate parameter settings and a p_m of 0, 0.005 and 0.05 respectively.

Figure 8.52: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$ and $R = g$

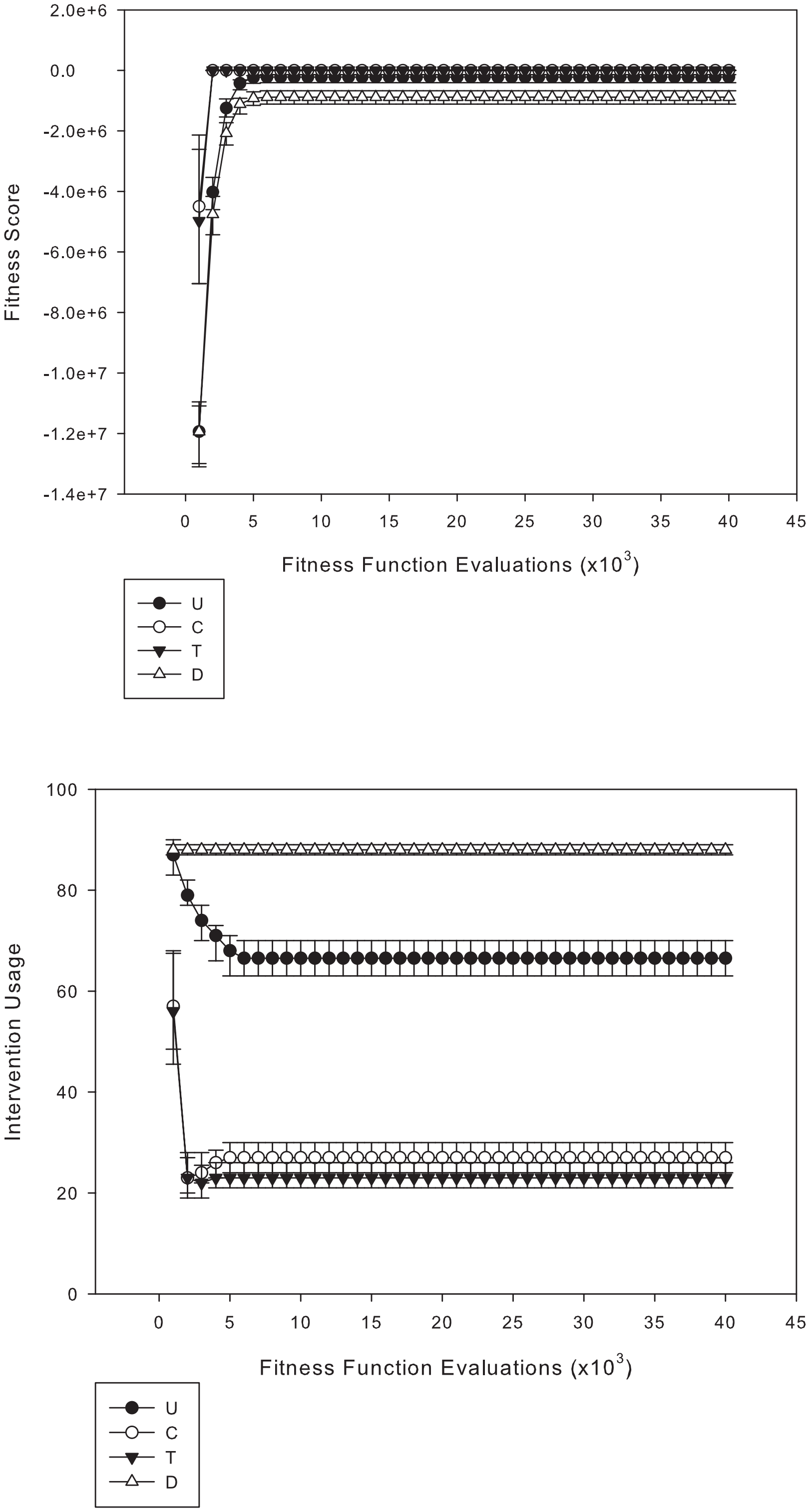
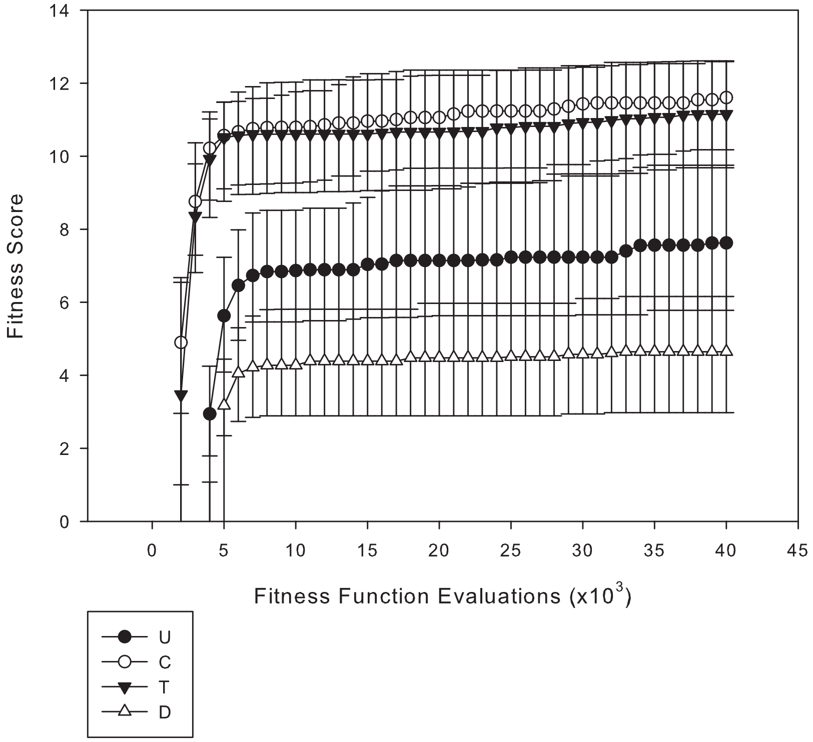


Figure 8.53: Results for $N = 100, p_c = 0.9, p_m = 0.005$ and $R = g$



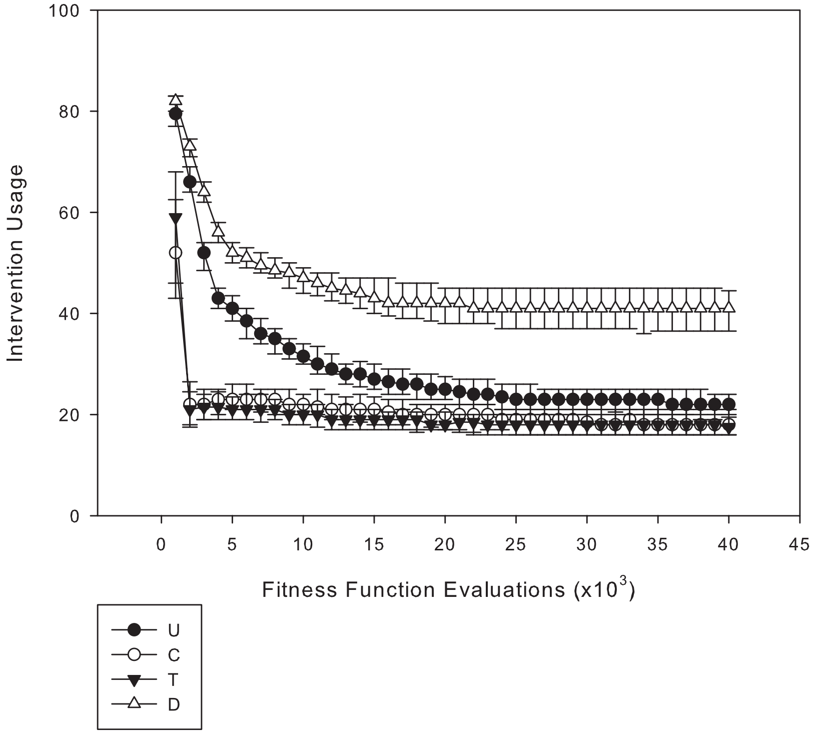


Figure 8.54: Results for $N = 100, p_c = 0.9, p_m = 0.05$ and $R = g$

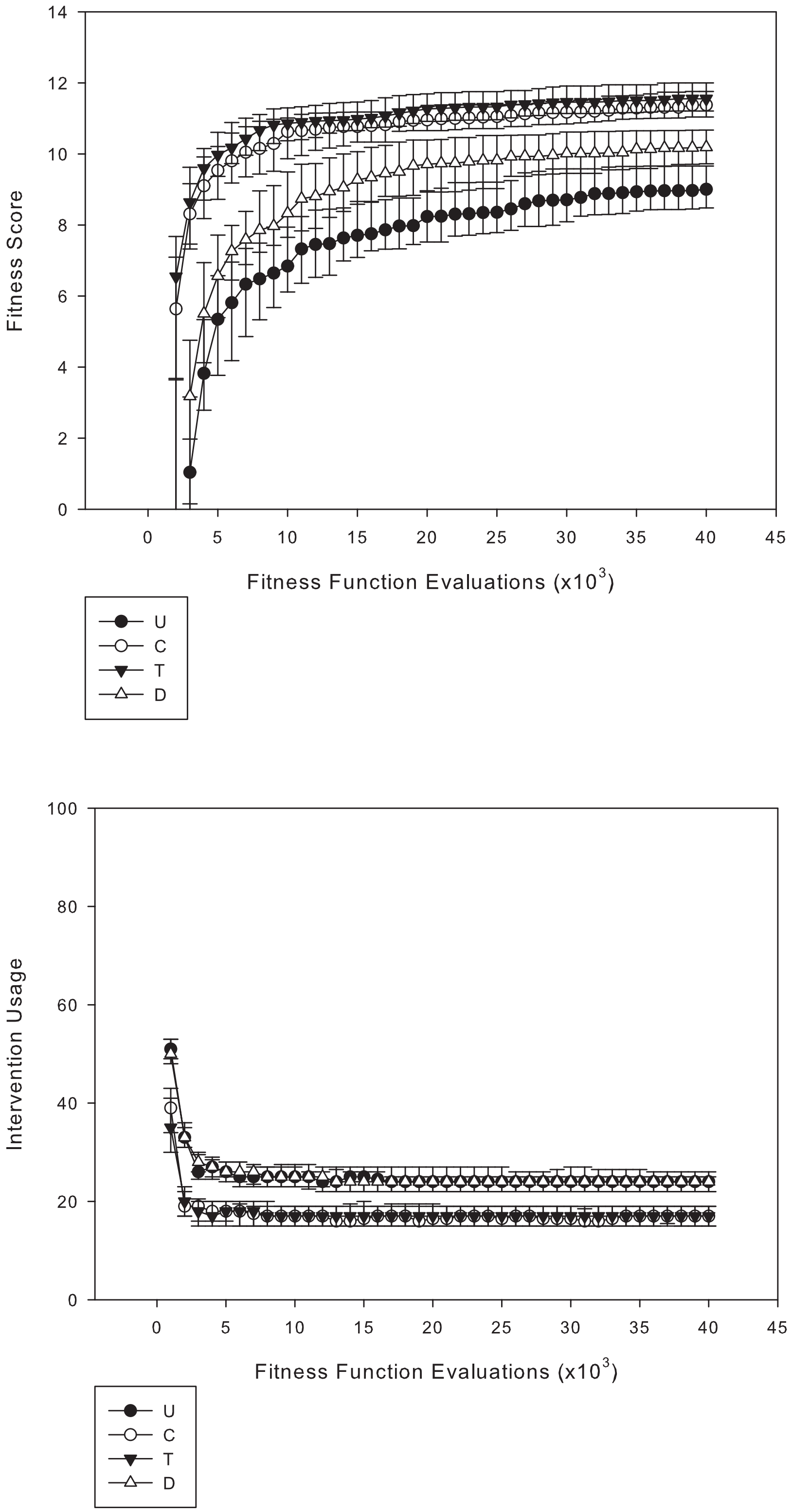


Figure 8.52 shows the performance of each crossover approach for treatment scheduling in the absence of mutation. When p_m is 0, UC and DUC produce schedules using considerably different numbers of interventions

compared to those returned by CalEB and TInSSel. UC and DUC utilise more interventions than both CalEB and TInSSel but this results in lower scores than those produced under the more parsimonious intervention allocation of CalEB and TInSSel.

Figure 8.53 shows the intervention usage and fitness scores when p_m is increased to 0.005. As with a p_m of 0, CalEB and TInSSel produce schedules using approximately 20 interventions. UC uses a similar number of interventions to CalEB and TInSSel and this produces fitness scores closer to those produced by CalEB and TInSSel. DUC settles on approximately 45 interventions and this results in poorer scoring schedules than those produced by CalEB, TInSSel and UC.

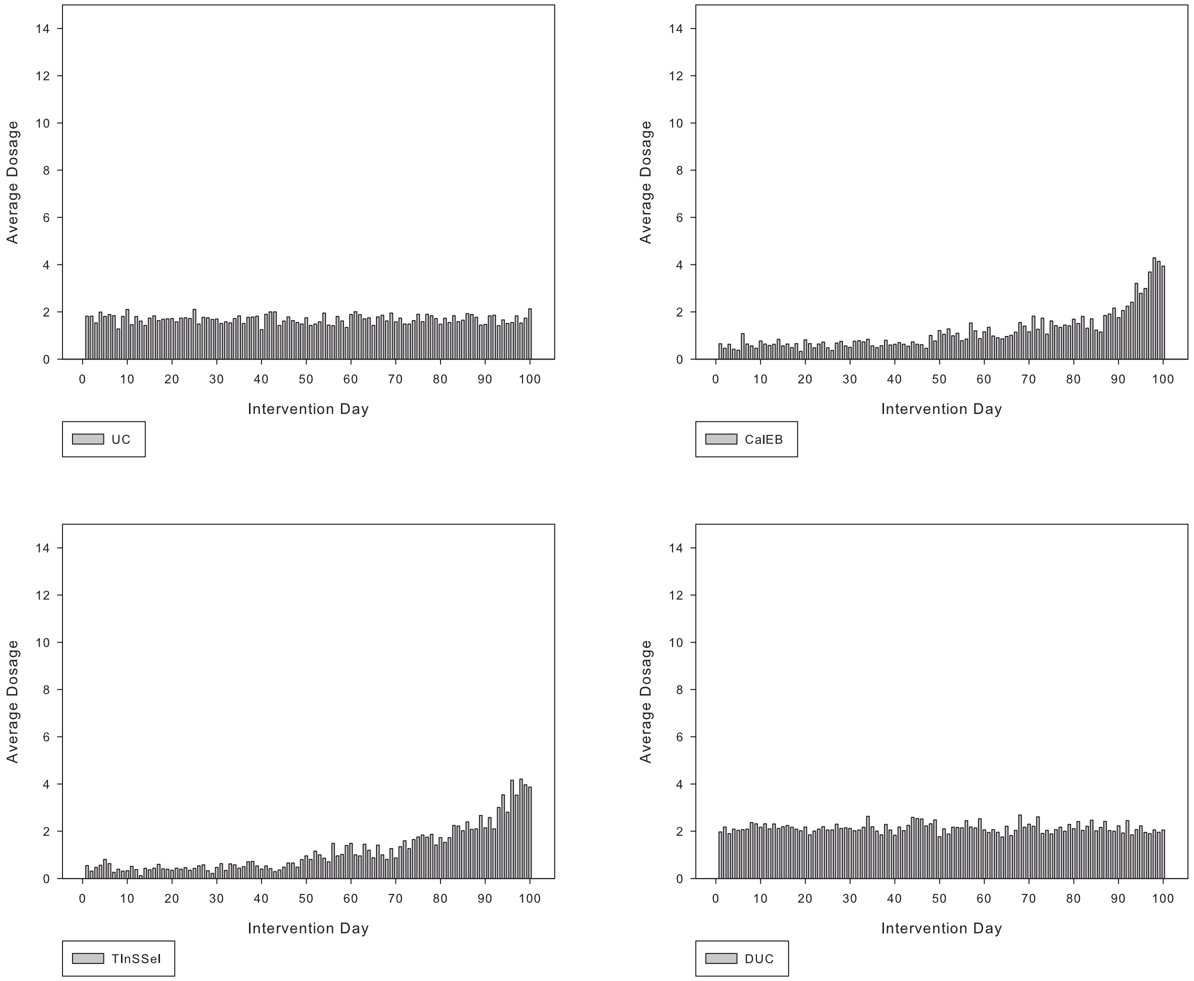
Figure 8.54 details the effect of further increasing p_m to 0.05. As with p_m levels of 0 and 0.005, CalEB and TInSSel produce schedules using approximately 20 interventions. Both DUC and UC return schedules using approximately 26 interventions. CalEB and TInSSel return fitness scores of a similar value, with DUC producing scores that are fitter than those returned by UC.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the directed intervention crossover approaches, in comparison to UC, under a generational replacement strategy for cancer chemotherapy treatment scheduling.

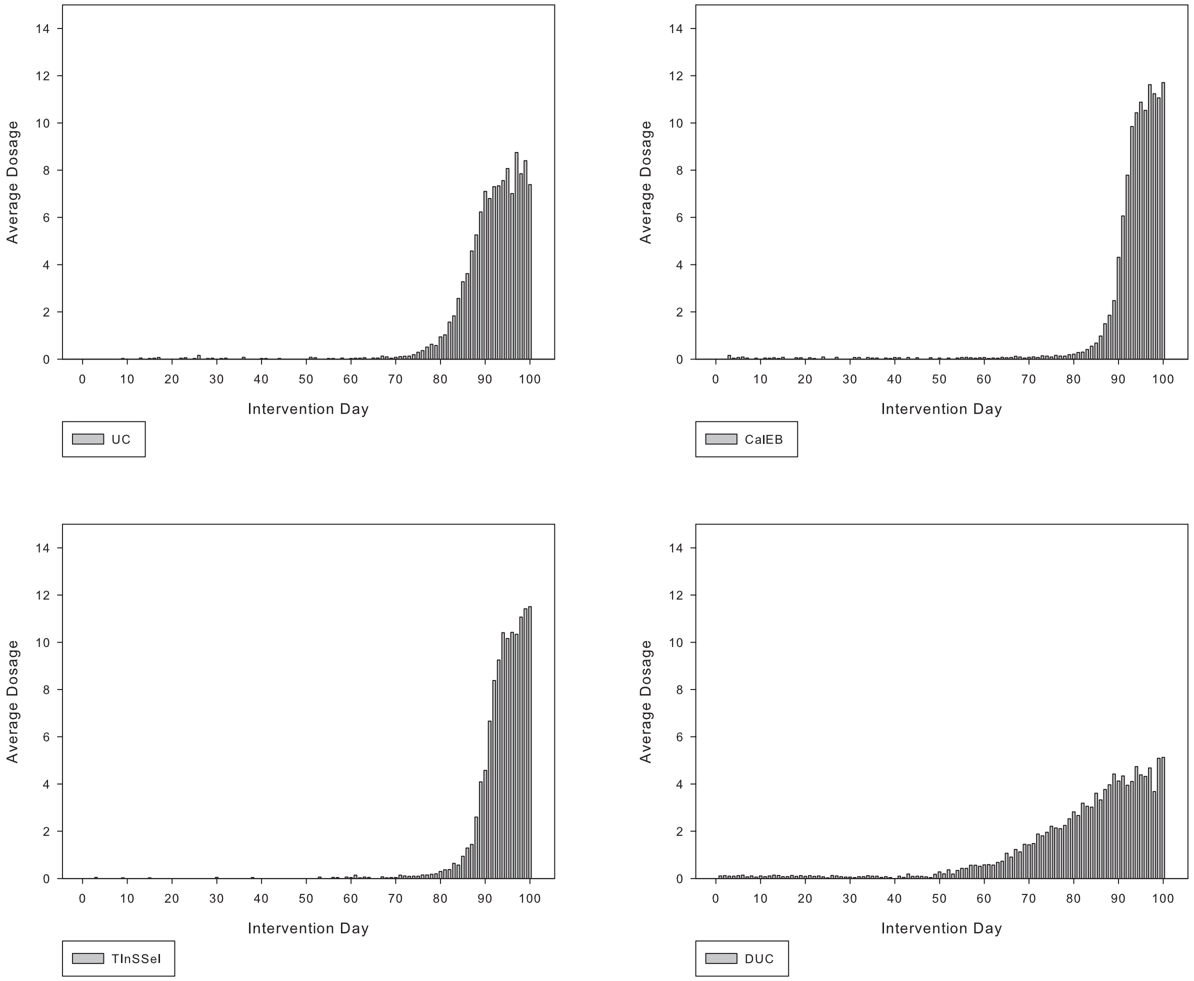
In the absence of mutation, as shown in Figure 8.52, DUC uses the most interventions, then UC, CalEB and finally TInSSel after 40,000 FFEs. From the statistical analysis undertaken in Appendix I.1, after 40,000 FFEs, CalEB, TInSSel and UC all outperform DUC, with CalEB and TInSSel also outperforming UC for this mutation level. The average placement of the interventions relating to these findings are shown in Figure 8.55. This shows that DUC, on average, uses higher dosage strengths than each of the other approaches. Both CalEB and TInSSel also focus more drug dosage on the later interventions than the other approaches.

Figure 8.55: Intervention placement for 40,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$ and $R = g$



When p_m is increased to 0.005, as shown in Figure 8.53, DUC uses the most interventions, followed by UC, then CalEB and TInSSel. From the statistical analysis undertaken in Appendix I.2, after 40,000 FFEs, CalEB, TInSSel and UC outperform DUC, and CalEB and TInSSel also outperform UC. The average placement of the interventions relating to these findings are shown in Figure 8.56. This shows that DUC has the least focussed interventions at the end of the schedule, resulting in weaker fitness scores. Both TInSSel and CalEB use larger dosages than UC, resulting in higher fitness scores for this mutation rate.

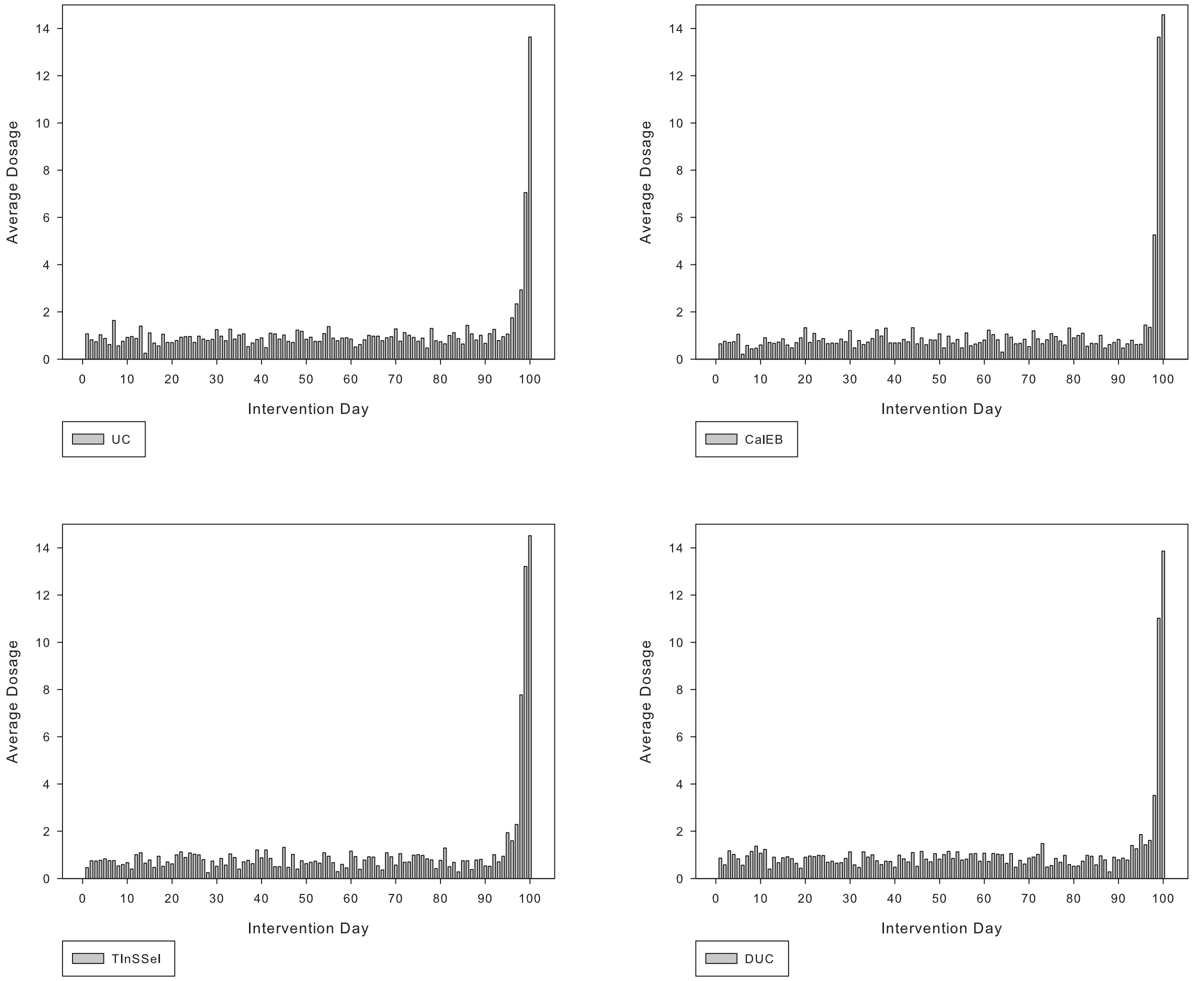
Figure 8.56: Intervention placement for 40,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$ and $R = g$



When p_m is further increased to 0.05, as shown in Figure 8.54, DUC and UC use the same number of interventions, which are more than those used by both CalEB and TInSSel. From the statistical analysis undertaken in Appendix I.3, after 40,000 FFEs, CalEB, TInSSel and DUC outperform UC, with CalEB and TInSSel both also outperforming DUC. The average placement of the interventions relating to these findings are shown in Figure 8.57. CalEB, TInSSel and DUC each have larger doses over the last 3 interventions than UC. DUC applies smaller dosages over the last 3 interventions than CalEB and TInSSel, resulting in poorer fitness scores for this mutation rate.

This shows that UC has the least focussed interventions at the end of the schedule, resulting in poorer fitness scores. Both TInSSel and CalEB use larger dosages than DUC, resulting in better fitness scores for this mutation rate.

Figure 8.57: Intervention placement for 40,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$ and $R = g$



Steady state replacement

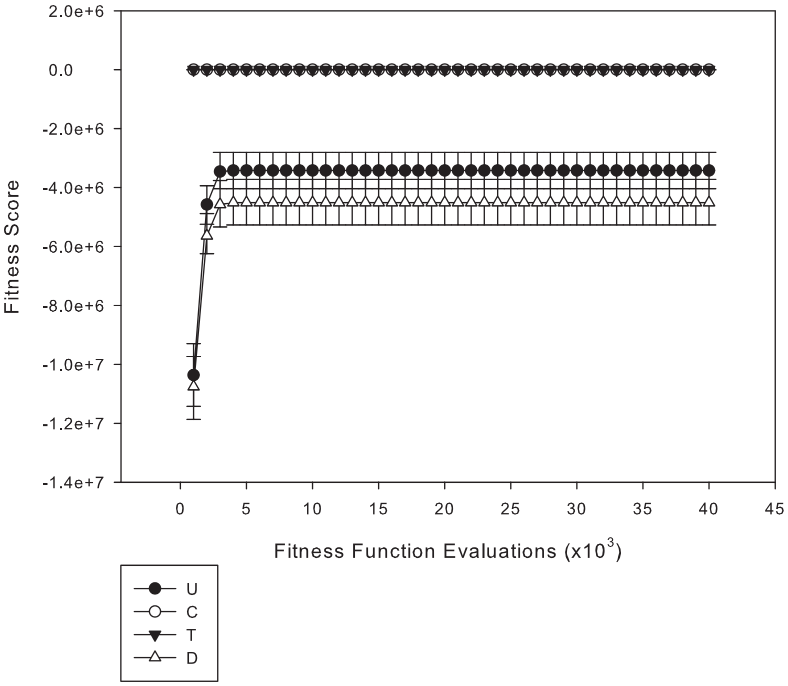
The same experiment will now be reviewed for a steady state replacement strategy. Figures 8.58 to 8.60 show the fitness scores and associated intervention usage for intermediate parameter settings and a steady state replacement strategy.

Figure 8.58 details the approaches in the absence of mutation. This shows that when p_m is 0, both CalEB and TInSSel utilise a similar number of interventions under steady state replacement as for a generational replacement strategy. This results in similar fitness scores being returned by these crossover approaches. DUC uses a similar number of interventions under steady state replacement as for generational replacement, whereas UC produces schedules requiring approximately 10 more interventions than schedules produced under a generational replacement strategy. Both UC and DUC produce scores with a weaker fitness under steady state replacement as opposed to the generational equivalent when p_m is 0.

Figure 8.59 shows the fitness and intervention graphs for the crossover approaches under steady state replacement when p_m is increased to 0.005. As with the generational equivalent, CalEB and TInSSel still produce schedules with approximately 20 interventions, with UC using slightly more. DUC again uses more interventions than UC, CalEB and TInSSel. The fitness scores are similar to those produced under generational replacement, with CalEB and TInSSel again outperforming UC, which in turn outperforms DUC.

Figure 8.60 details the fitness scores and intervention usage when p_m is further increased to 0.05. The intervention usage is similar to that used under generational replacement, and this results in similar fitness scores returned by each of the approaches under a steady state replacement strategy.

Figure 8.58: Results for $N = 100, p_c = 0.9, p_m = 0$ and $R = ss$



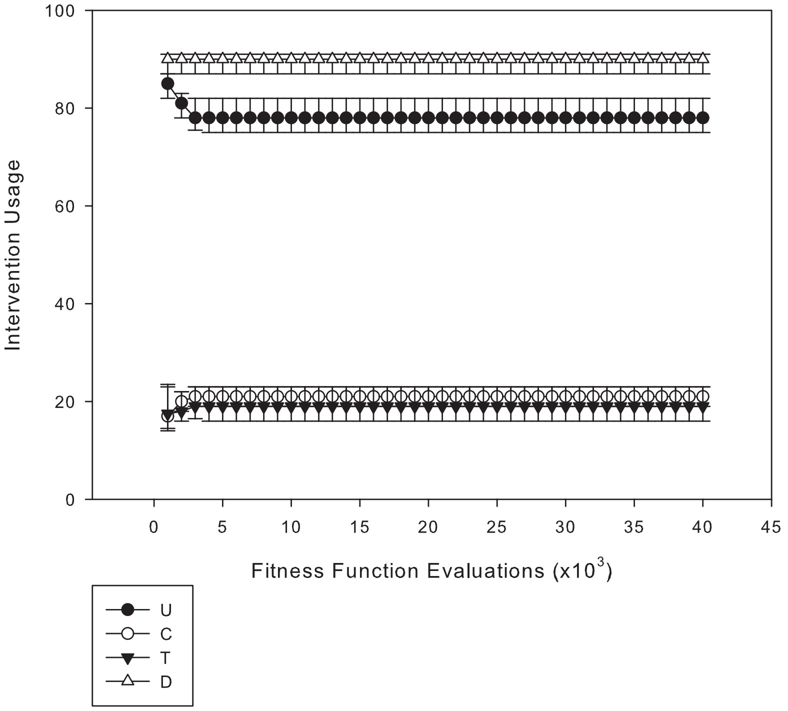
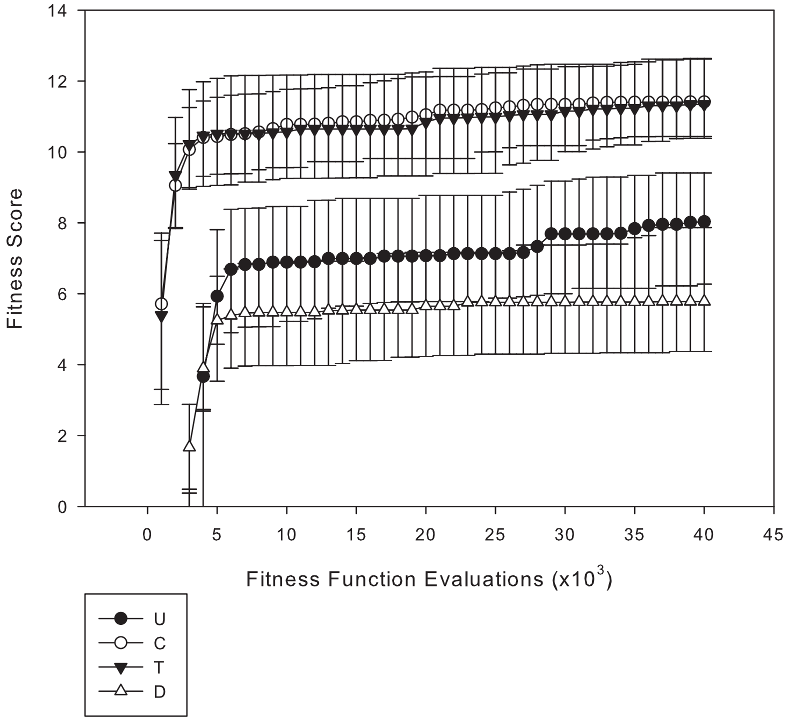


Figure 8.59: Results for $N = 100, p_c = 0.9, p_m = 0.005$ and $R = ss$



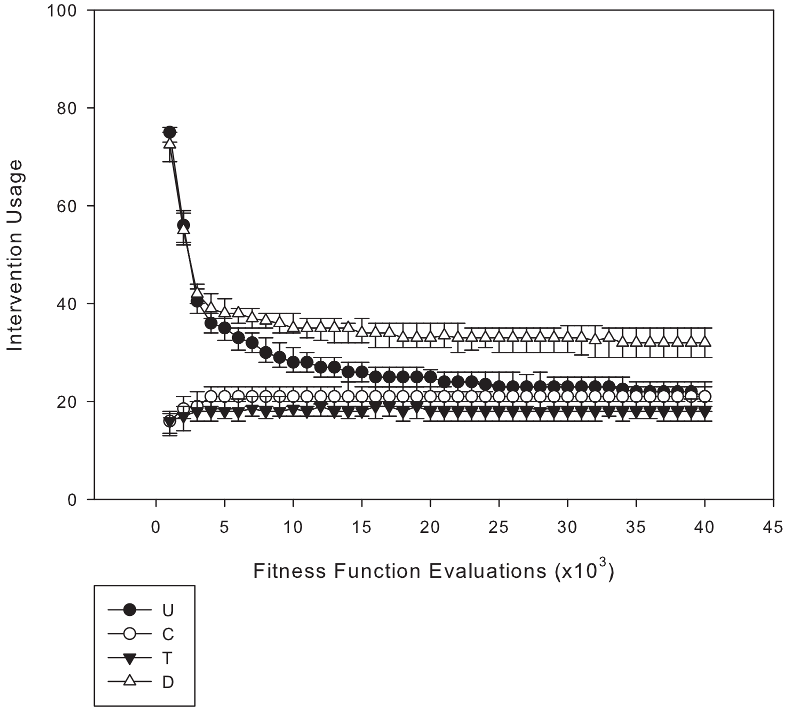
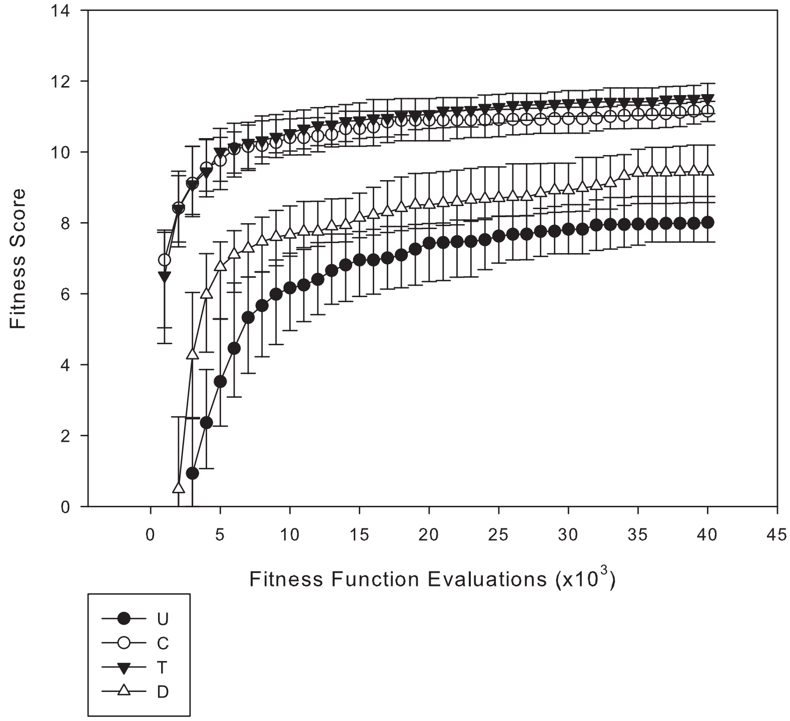
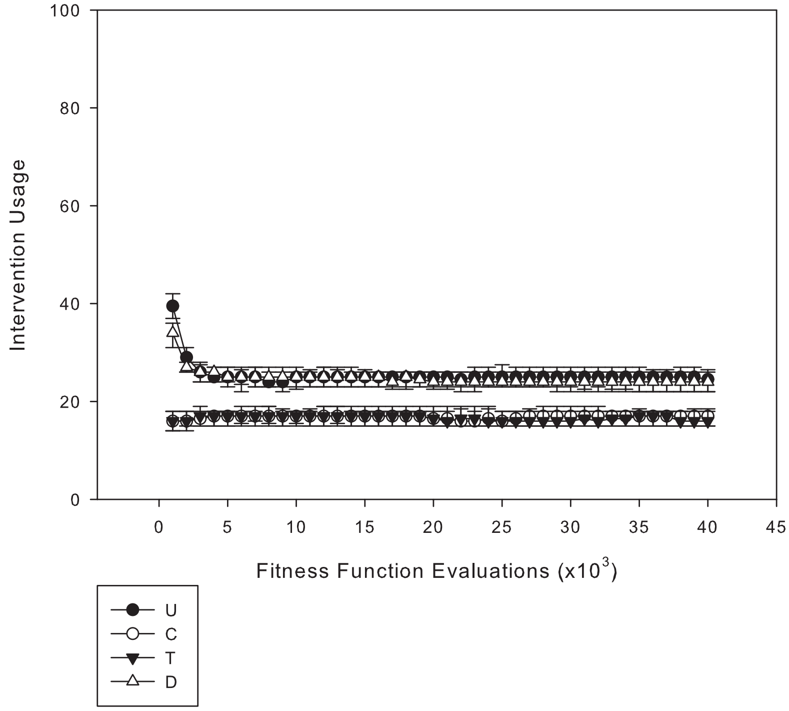


Figure 8.60: Results for $N = 100, p_c = 0.9, p_m = 0.05$ and $R = ss$





8.2.1 Summary of experiment for chemotherapy treatment scheduling

The previous experiment reviewed CalEB, TInSSel and DUC, in comparison to UC, for the task of cancer chemotherapy scheduling. This section draws together the salient points from these experiments, to demonstrate the effectiveness of the directed intervention crossover techniques.

In order to ascertain if there is any statistical significant differences between approaches for this problem, the Kruskal-Wallis test results for the intermediate test cases were calculated. These are shown in Tables I.1 to I.3. As with the bio-control experiments discussed previously, for clarity, Appendix J.2 shows which crossover approaches are significantly different over the range of FFEs. As detailed in Section 6.9, the chemotherapy experiments are evaluated over 40,000 FFEs, in observable increments of 1,000 FFEs. This means that for each p_m level, there are 40 FFE points under observation for each approach, ranging from 1,000 FFEs to 40,000 FFEs.

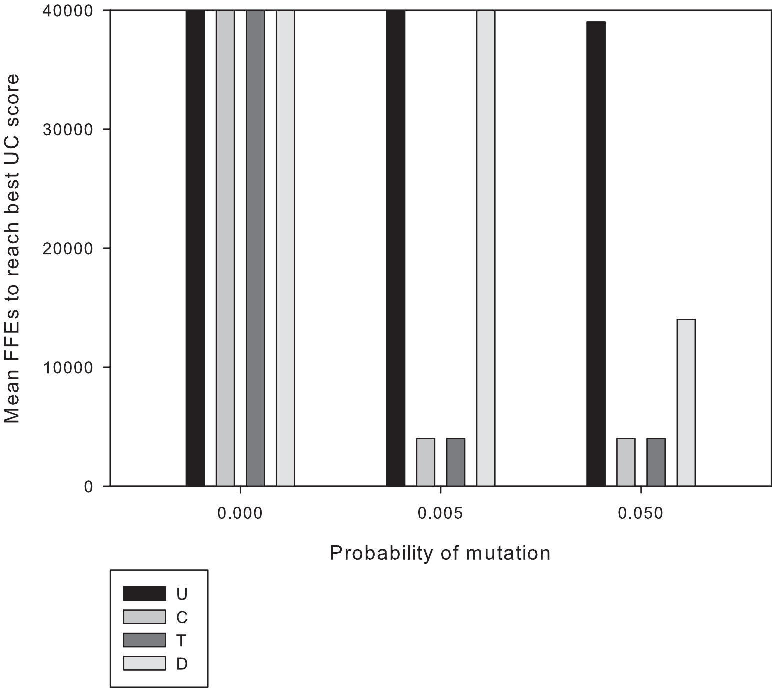


Figure 8.61: FFEs to reach the best score found by UC (9)

This section has reviewed the directed intervention crossover approaches against UC, for the production of cancer chemotherapy treatment schedules. This analysis has been undertaken over a range of mutation rates and replacement strategies.

The statistical analysis detailed in Appendix I.1 shows that there is a statistical difference between at least two of the crossover approaches from 1,000 to 40,000 FFEs for each of the mutation levels. For clarity, the crossover approaches which are statistically different to each other are described in Appendix J.2. This shows that for each of the mutation levels, CalEB and TInSSel statistically outperform both UC and DUC for each FFE point under observation. DUC is shown to be outperformed by UC for 3,000 to 40,000 FFEs for mutation rates of 0 and 0.005, but this is reversed for a mutation rate of 0.05, with DUC outperforming UC in this FFE range.

Section 8.1.6 found that for bio-control scheduling, with large values of p_m, DUC returned fitter scores than those found by UC. This was also the case for chemotherapy treatment optimisation with a large mutation value. As with the bio-control problem, this shows that in the presence of large quantities of new genetic material being introduced through mutation, having a target number of interventions, even a simple target like DUC, is more efficient than UC for this problem since UC finds it hard to get rid of interventions.

As stated previously, DUC uses the same intervention selection mechanism as TInSSel, but has a different method for calculating the number of interventions to be present in offspring. For every p_m setting, regardless of replacement strategy, DUC never outperforms TInSSel, but is often statistically outperformed by it. As stated in

Section 8, DUC was expected to be poorer than both CalEB and TInSSel for this problem, due to its simplistic intervention target calculation. This has been shown to be the case, demonstrating the benefit of using a window of selection around the fitter parent, as opposed to strictly using the number of interventions in the fitter parent as the target for this chemotherapy scheduling problem.

As CalEB and TInSSel are always shown to return fitter scores than the other crossover approaches, they would appear to be the crossover techniques of choice for the chemotherapy scheduling problem, regardless of $p\ _m$ setting or replacement strategy used.

The previous analysis has shown that both CalEB and TInSSel are faster to discover fitter intervention levels, resulting in better scores being returned by these approaches when compared with UC and DUC.

Over the range of mutation levels, the best score found by UC for the cancer chemotherapy scheduling problem was 9. This was found when the mutation rate was 0.05. Figure 8.61 shows the FFEs required by each of the crossover approaches to find this best score returned by UC, over the range of mutation levels. This shows that in the absence of mutation, none of the crossover approaches return this score.

When the mutation rate was increased to 0.005, Figure 8.61 shows that both CalEB and TInSSel find this score, after 4,000 FFEs. Even after 40,000 FFEs, neither DUC or UC find a score of this fitness. For a further mutation increase to 0.05, each of the crossover approaches find a score of at least 9. UC requires 39,000 FFEs to find this value and DUC uses 14,000. As with a mutation rate of 0.005, both CalEB and TInSSel still require only 4,000 FFEs to find the same solution.

From inspecting the crossover approaches over a range of intervention penalties and parameter values, certain deductions can be made:

- Both CalEB and TInSSel outperform the other approaches, and are therefore the crossover technique of choice for the chemotherapy scheduling problem.

- DUC uses the same intervention selection mechanism as TInSSel, but has a simpler method for calculating the number of interventions to be present in offspring. For every intervention penalty and parameter setting, DUC never outperforms TInSSel, but is always statistically outperformed by it. As previously mentioned in Section 8, both TInSSel and CalEB were expected to outperform DUC for this problem. This has been shown to be the case, demonstrating the benefit of using a window of selection around the fitter parent, as opposed to strictly using the number of interventions in the fitter parent as the target for this problem.

- When p_m is 0.05, a lot of genetic material is being added to the system, and thus there is a lot of noise present. It is only for this mutation level that DUC or UC find fit solutions. As CalEB, TInSSel and DUC outperform UC for this mutation rate, this again shows the benefit of using a target for the interventions for offspring. Even a simple directed approach like DUC is more beneficial for this problem than no target.

As described above, CalEB and TInSSel are the most efficient crossover approach under review for this problem, displaying identical performance over the range of mutation rates. When compared to UC, CalEB and TInSSel are shown to require fewer FFEs to find scores which statistically outperforms UC for all FFEs.

Figure 8.62 shows the percentage of improvement, in terms of both fewer FFEs required and statistically better scores found, through using TInSSel instead of UC. This shows the results for a mutation rate of 0.05, which was where each of the crossover approaches on average performed best. This shows that TInSSel requires 89% fewer FFEs than UC to find the same score and that for all of the FFEs under review, TInSSel returns significantly better scores.

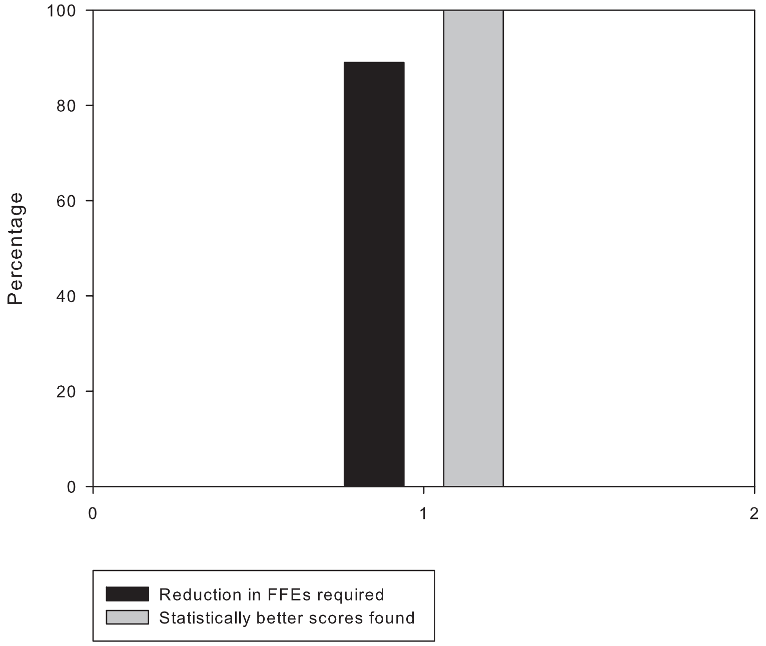


Figure 8.62: Performance improvement between UC and TInSSel when $p_m = 0.05$

8.2.2 Further analysis

The previous section reviewed CalEB, TInSSel and DUC in comparison to UC for chemotherapy treatment scheduling, using the intermediate parameter settings.

In order to further analyse the abilities of each of the crossover approaches, it was necessary to review them across a range of parameter settings, outwith the intermediate set used in the previous sections.

Appendix G details the performance of CalEB, TInSSel, DUC and UC across the range of population sizes, crossover rates, and mutation rates. A review of the crossover approaches over this range of parameters are detailed in Appendix A.4. This review describes how the trends demonstrated in Section 8.2.1 hold with changing mutation rates, crossover rates and population sizes.

8.3 Summary of the evaluation of directed intervention crossover approaches

This chapter has reviewed the abilities of CalEB, TInSSel and DUC, in comparison with UC, for deriving schedules of both bio-control treatments and chemotherapy drug treatments. Section 8.1 reviewed these techniques for bio-control scheduling over a range of intervention penalties. Analysis of these experiments found that TInSSel was statistically better than all of the other crossover approaches, regardless of the penalty per intervention, both in terms of number of FFEs required and fitness score of solutions found. On average, TInSSel requires 37% fewer FFEs to find the best score returned by UC.

Similar analysis was undertaken to compare the abilities of CalEB, TInSSel and DUC, in comparison with UC, for chemotherapy scheduling. This found that both CalEB and TInSSel statistically outperform both UC and DUC for every FFE point. Both CalEB and TInSSel demonstrated a 89% reduction in the number of fitness functions evaluations required to find the best score returned by UC.

From the experiments described in Sections 8.1 and 8.2, certain trends have appeared which hold for both bio-control scheduling and cancer chemotherapy scheduling. As described in Section 8, DUC is a test crossover approach which was not expected to perform as well as the other directed intervention crossover approaches. DUC has a simplistic offspring target intervention calculation which simply uses the number of interventions in the fitter parent as the target for offspring intervention levels. From both of the test problems described previously, DUC is shown to perform poorer than CalEB or TInSSel. This highlights that a window of offspring intervention levels is critical to the successful search of directed intervention crossover approaches.

DUC is outperformed by UC for mutation levels of 0 or 0.005. It is interesting to note that for both problems, when the mutation rate is further increased to 0.05, DUC is a more effective crossover approach than UC. This shows that when large quantities of genetic noise are being added to the population through a high mutation rate, having a target number of interventions for offspring, even a simple target like DUC, is more efficient than UC which has no offspring intervention target.

Although both CalEB and TInSSel perform in a similar manner for cancer chemotherapy scheduling, for bio-control scheduling TInSSel is shown to outperform CalEB for many of the FFEs under review. This shows that from the problems under review, there is never utility in enforcing a uniform distribution of gene selection as stochastic selection of genetic material is always at least as good as a uniformly distributed approach.

From reviewing the bio-control problem, as fitness scores are generally poorer when p_m is increased from 0.005 to 0.05, this shows a limit for mutation. This is not the case however for cancer chemotherapy scheduling, as the best results are found under a mutation rate, p_m, of 0.05.

The abilities of directed intervention crossover has been demonstrated. They have been shown to be effective on both problems with an explicit penalty value per intervention, that of bio-control scheduling, and on chemotherapy scheduling, whereby the intervention penalty is less pronounced. This has shown that stochastic selection and a window of target intervention levels for offspring can dramatically improve the GA crossover process for problems of this type. The next chapter introduces another novel crossover approach, Fitness Directed Crossover (FDC). This further extends the windowed approach of TInSSel, while adhering to the stochastic selection of genetic material.

Chapter 9

Fitness Directed Crossover

Chapter 8 reviewed the directed intervention crossover techniques of CalEB, TInSSel and DUC compared with UC for both bio-control and cancer chemotherapy scheduling. This has shown the benefits, in terms of both fitness scores and the reduction of FFEs required, of using directed intervention crossover techniques compared with UC, for the problems under investigation.

From the analysis undertaken in Chapter 8, TInSSel was shown to be the best performing crossover approach for the problems under investigation. This uses stochastic selection of genetic material once the target number of interventions has been calculated. Although this target calculation uses the difference in intervention levels between parents, it does not take into account the relative difference in fitness between parents selected for crossover.

This chapter introduces a further extension to the directed intervention crossover techniques termed Fitness Directed Crossover (FDC). This new approach reviews the benefit of incorporating the relative fitness difference of parents into the intervention target calculations. Section 9.1 will describe the FDC technique. Section 9.2 compares FDC with UC, CalEB and TInSSel for bio-control scheduling. This is followed by Section 9.3 which compares FDC, UC, CalEB and TInSSel for the scheduling of cancer chemotherapy treatments.

9.1 Fitness Directed Crossover

Both the TInSSel and CalEB techniques use the difference in the number of interventions between parents to scale the window of possible intervention sizes around the fitter parent. These approaches use the fitness of the parents purely to decide which parent's intervention level should be used for I_F. The previous studies raise the question of whether additional information is contained in the fitness of parents, especially on their comparative relative fitness levels when compared with the rest of the population that can be used to further guide the search process? If so, there may be a better way of determining I_T which would utilise both the difference in intervention levels between parents as well as the relative difference in fitness scores? The Fitness Directed Crossover technique is an approach designed to utilise these features and determine if such benefits can be gained.

FDC uses both the difference in the number of interventions between parents as well as the relative fitness difference to calculate the size of potential offspring. The FDC approach places an emphasis on selecting offspring 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. In essence, this technique looks for information on which direction potentially offers the highest utility and moves in this direction based on potential return, where this return is scaled based on the relative fitness difference between parents.

The concept of extending in a direction to attempt higher gain is not new in GAs. Work by Deb et al has created the self-adaptive simulated binary crossover operator [111; 112]. This technique reviews if the created child solution is better than the participating parent solutions, if so, the child is extended further in the hope of creating an even better solution. Simulated binary crossover uses a distribution index parameter n_c. Large values of n_c produce offspring close to the parents and conversely, for small values of n_c, solutions away from the parent are likely to be created. Unlike the directed intervention work, this extension is not in terms of intervention usage, but of binary representation of real parameter values, however the underlying principle of extending in promising directions is similar.

As with the other crossover approaches discussed earlier, the FDC 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 used 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 9.1, where F is the score being normalised.

$$F_{norm} = \frac{F - F_{min}}{F_{max} - F_{min}} \tag{9.1}$$

The number of interventions used by the fitter of the two parents is recorded as I_F and the target 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 then calculated as shown in Equation 9.2.

$$I_T = I_F + (2T - 1)(I_1 - I_2)(F_1 - F_2) \tag{9.2}$$

As with the other directed intervention techniques, 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}. As with the previous directed intervention crossover approaches, I_{max} is limited to the size of the set of interventions present in both parents.

The FDC calculation for 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.

To demonstrate the operation of FDC, consider a GA with a population of 10 chromosomes as shown below:

0	0	0	0	0	0	0	1	1	0
0	0	1	1	1	0	0	1	0	0
0	0	1	1	1	0	0	1	1	0
0	0	1	1	1	0	0	1	1	1
1	1	1	1	1	0	0	1	0	1
1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1

In this example, a minimisation problem is assumed, thus the closer a fitness score is to zero, the better. If the 10 chromosomes described above had fitness scores of 10, 25, 30, 45, 59, 69, 72, 88, 91 and 101 respectively, and if chromosomes 2 and 6 were randomly selected for breeding (with fitness scores of 25 and 69 respectively), FDC would proceed in the following manner:

- The normalised fitness score would be calculated for each parent based on equation 9.1. The minimum score in the population is 10 and the maximum score is 101. Thus Parent 1 would have a normalised score of 0.165, calculated as $\frac{25-10}{101-10}$. Parent 2 would have a normalised score of 0.648 calculated as $\frac{69-10}{101-10}$.
- Equation 9.2 is then used to calculate the number of interventions for offspring I_T. The offspring intervention size is then calculated as shown in equation 9.3, where $T = 0$ as it is a minimisation problem, $I_F = 4$, $I_1 = 4, I_2 = 9$ and $F_1 = 0.165, F_2 = 0.648$.
- In the above equation, $I_T = 1.585$. As all I_T values are rounded down, this results in $I_T = 1$. The required number of interventions I_T are then selected in the same manner as for the TInSSel or DUC approaches.

$$I_T = 4 + (-1)(4 - 9)(0.165 - 0.648) \tag{9.3}$$

Example	T	I_1	I_2	F_1	F_2	I_F	F_F	I_T
1	1	6	4	0.9	0.4	6	0.9	7
2	0	6	4	0.9	0.4	4	0.4	3
3	1	6	4	0.4	0.9	4	0.9	3
4	0	6	4	0.4	0.9	6	0.4	7
5	1	4	6	0.9	0.4	4	0.9	3
6	0	4	6	0.9	0.4	6	0.4	7
7	1	4	6	0.4	0.9	6	0.9	7
8	0	4	6	0.4	0.9	4	0.4	3

Table 9.1: Example of FDC operation

9.1.1 FDC examples

This section further demonstrates the operation of the FDC approach. Table 9.1 shows the FDC operation for both minimisation and maximisation problems, with the "Example" column allowing identification of each row. For clarity, an extra parameter F_F is introduced which shows the fitter normalised fitness score for each example.

- Example 1 shows that for a maximisation problem ($T = 1$), as F_1 is fitter, $I_F = 6$. With the FDC calculation, the intervention size is further increased to 7.
- Example 2 shows that for a minimisation problem, as the fitter parent contains less interventions than the lesser fit parent, I_T is decreased to 3.
- Example 3 shows a maximisation problem where $I_F = 4$. As fewer interventions produce a fitter score, FDC uses this information to further decrease I_T to 3.
- Example 4 shows that for a minimisation problem, if more interventions appears fitter, I_T will contain more interventions than I_F.
- Example 5 shows that for a maximisation problem, if less interventions appears to offer more utility, I_T will be even less than I_F.
- Example 6 shows that for a minimisation problem, if larger schedules offer more utility, I_T will be increased beyond I_F.
- Example 7 shows that for a maximisation problem, if the larger schedule is fitter, I_T will be extended beyond this. Finally, Example 8 shows that for a minimisation problem, if a smaller schedule is fitter, I_T will contain less interventions than I_F.

This section has introduced the concept of fitness directed crossover. The following sections will compare FDC to TInSSel, CalEB and UC at the task of deriving bio-control schedules for mushroom farming. The subsequent section then reviews the performance of these same algorithms for the construction of cancer chemotherapy treatment schedules.

9.2 Evaluation of FDC for bio-control scheduling

For this evaluation, the same parameter configuration is used as for the traditional crossover experiments detailed in Chapter 7 and the directed intervention experiments described in Chapter 8. The bio-control problem is therefore sampled over 5,000 fitness function evaluations (FFEs), at intervals of 100, resulting in 50 data points for review. For each sample point, 100 runs of each crossover approach were recorded for each parameter combination, with the best scoring values for each run for each FFE being recorded.

Each of the following sections review the effectiveness of FDC, in comparison to CalEB, TInSSel and UC, for the task of bio-control scheduling. This is undertaken over a range of penalty values per intervention. As with the previous experiments, the crossover techniques are tested over a range of parameter settings. A complete set of results across the parameter combinations has also been collected and these can be seen in Appendix D.

This Appendix contains the complete set of results showing the fitness scores against FFEs associated with these experiments across a range of population sizes, intervention penalties and crossover and mutation rates.

As with the previous experiments, the intermediate parameter settings for each approach will be reviewed in detail, rather than explicitly detailing each possible parameter combination. The following sections will therefore review FDC, compared to CalEB, TInSSel and UC, for the varying levels of intervention penalty P, as described in Section 5.3.2, with a population of $N = 100$, crossover rate of $p_c = 0.9$ and mutation rates, p_m, of 0, 0.005 and 0.05.

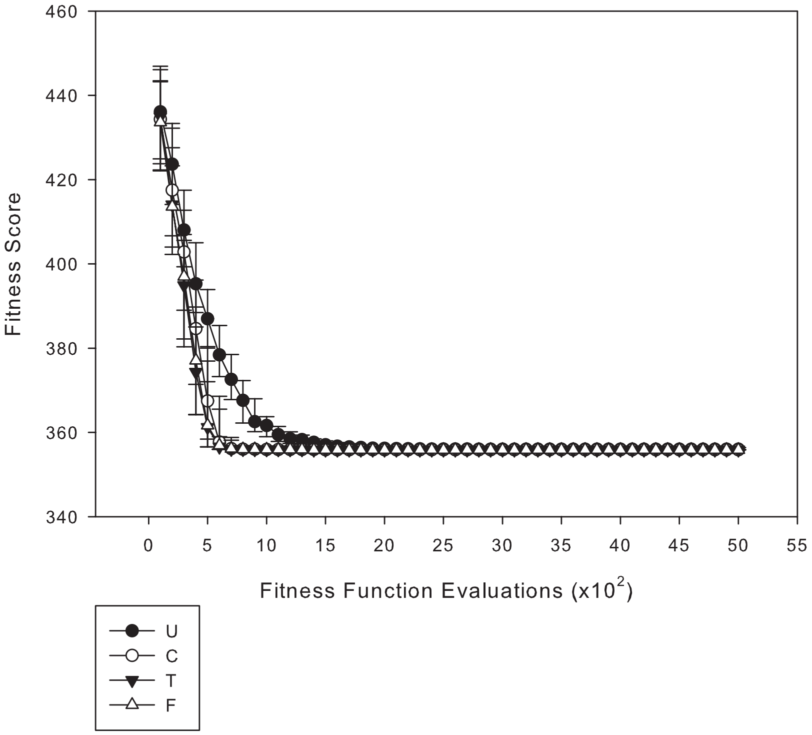
As with the directed intervention experiments in Chapter 8 the intermediate parameter settings will use a generational replacement strategy and will be followed by analysis of the average placement of interventions. Intermediate parameter settings will then be reviewed under a steady state replacement strategy. As with previous experiments, the replacement mechanism, R, for each experiment will be defined as either generational, $R = g$, or steady state, $R = ss$.

Graphs displaying the results for both the fitness scores and associated intervention usage for the intermediate parameter settings under both generational and steady state replacement will now be reviewed. Both fitness scores versus FFEs and intervention usage versus FFEs are plotted over the range of FFE samples. As stated previously, as the bio-control optimisation problem is a minimisation problem, lower scores are fitter than higher valued scores. On the graphs, U, C, T and F represent UC, CalEB, TInSSel and FDC respectively.

9.2.1 0 penalty points per intervention

This section reviews FDC, in comparison to CalEB, TInSSel and UC, for bio-control scheduling with a penalty of 0 points per intervention. As described in Section 9.2, this will focus on the intermediate parameter settings. Figures 9.1 to 9.3 show the fitness scores and associated intervention usage for p_m levels of 0, 0.005 and 0.05 respectively.

Figure 9.1: Results for $N = 100, p_c = 0.9, p_m = 0$,$P = 0$ and $R = g$



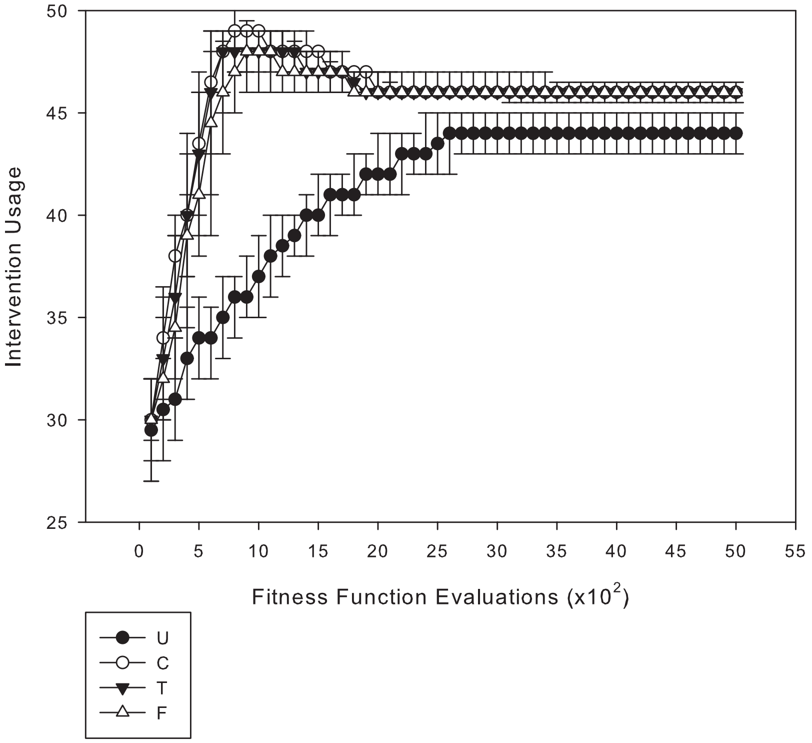


Figure 9.2: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 0$ and $R = g$

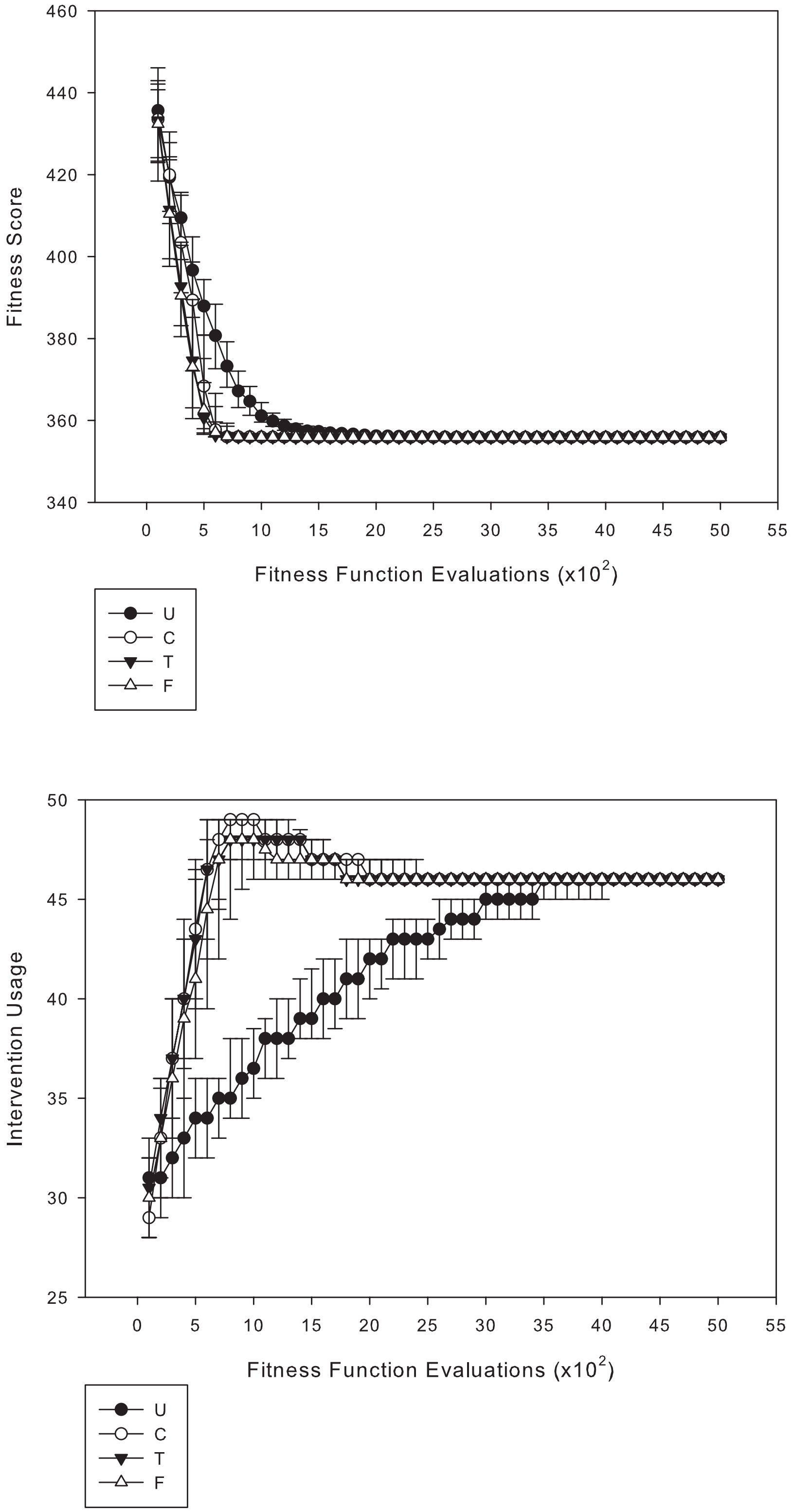
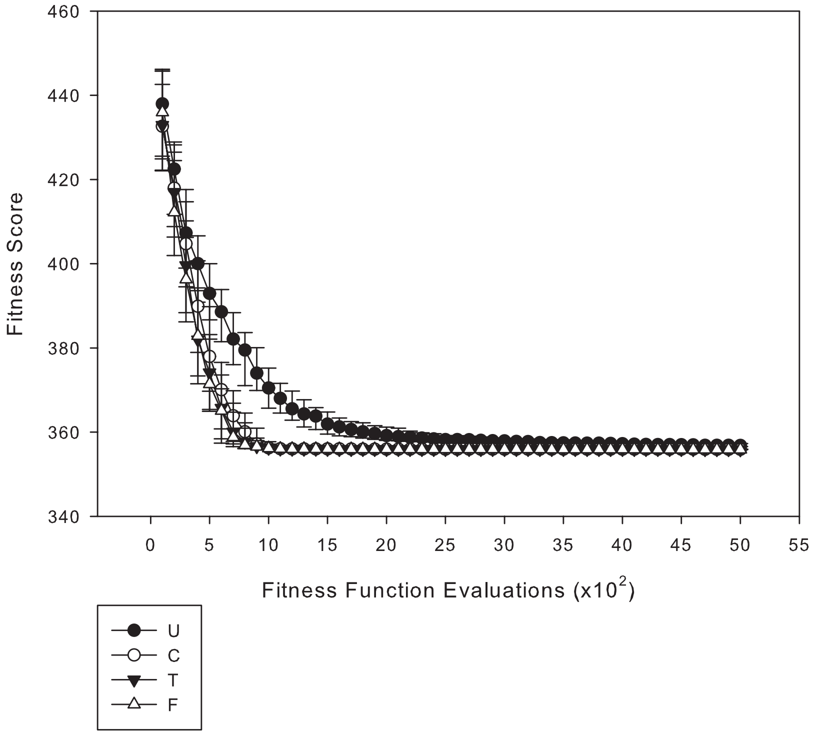
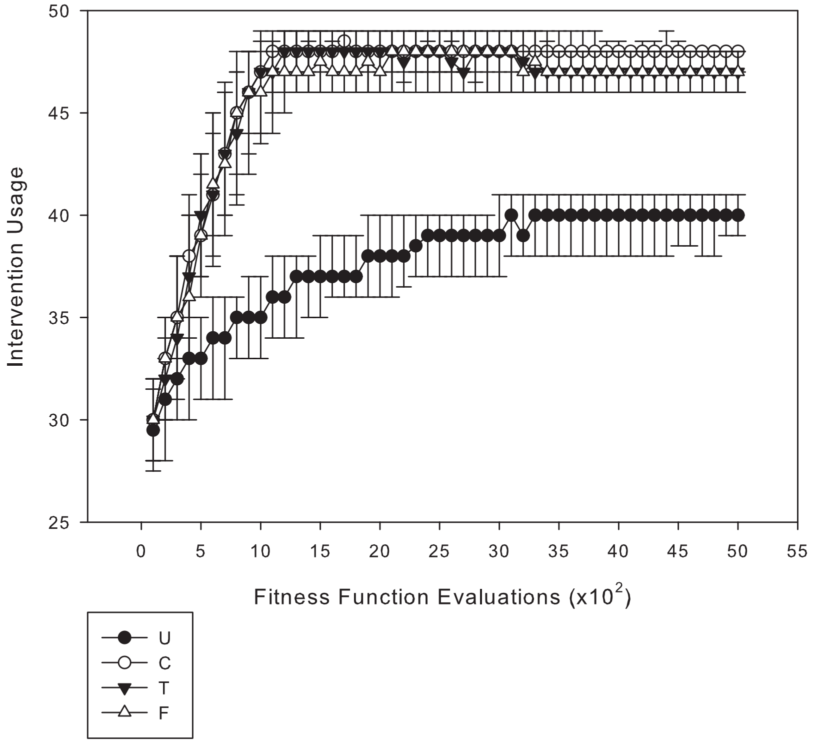


Figure 9.3: Results for $N = 100, p_c = 0.9, p_m = 0.05, P = 0$ and $R = g$





Reviewing Figure 9.1 shows that for this parameter setting, FDC follows a similar trend to both CalEB and TInSSel. This is in terms of both the accumulation of interventions into schedules and fitness scores returned.

Each of the directed intervention techniques settle to a similar number of interventions after approximately 1,700 FFEs.

Figure 9.2 details the fitness scores and intervention usage when p_m is increased to 0.005. As was the case when p_m was 0, each of the directed intervention techniques follow similar trends in terms of fitness scores and intervention usage. Each of the directed approaches also settle to a similar number of interventions.

Figure 9.3 shows the effect of increasing p_m to 0.05. As with the lower mutation values, each of the directed techniques follow similar trends in terms of both fitness scores returned and intervention usage.

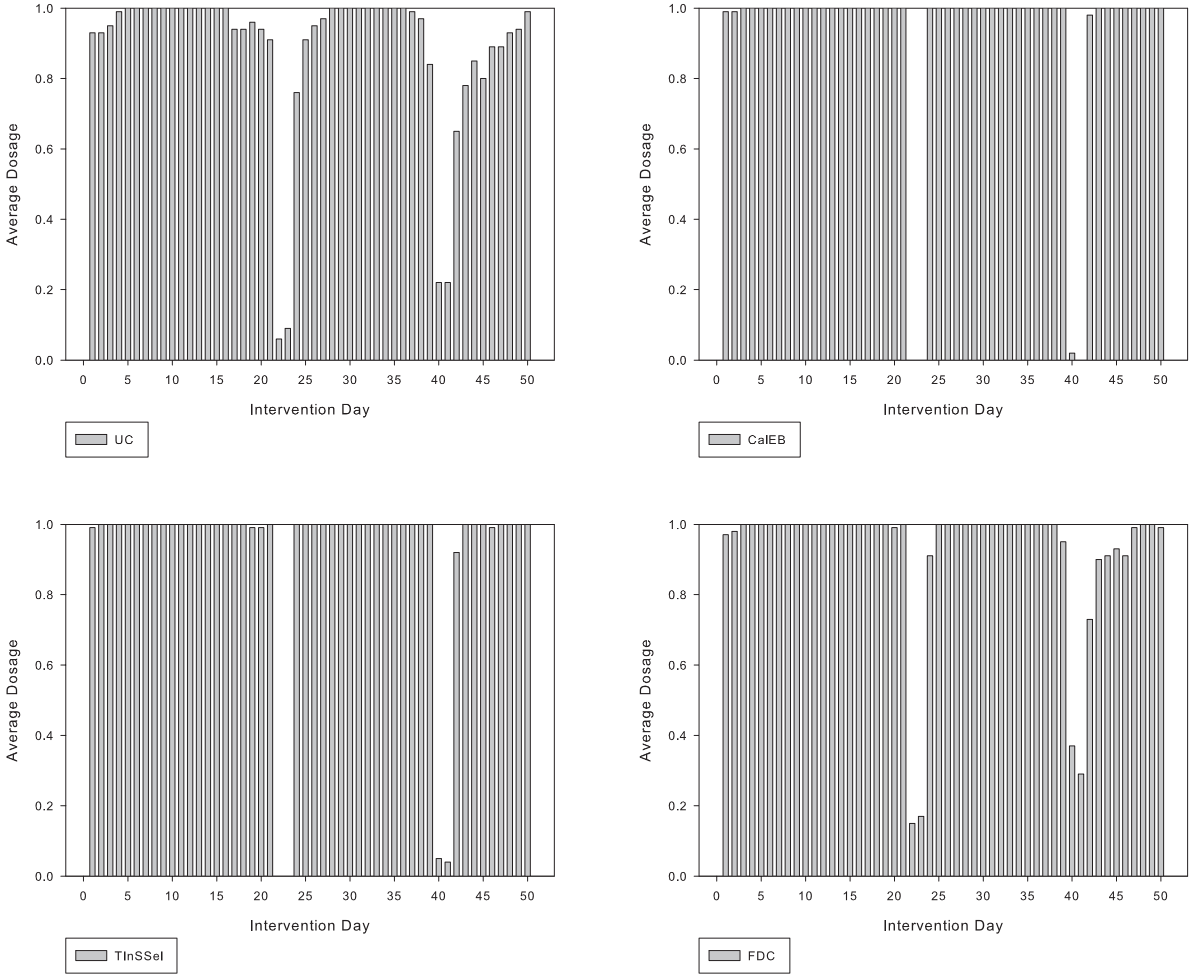
From these experiments, each of the directed intervention approaches are shown to follow similar trends, regardless of p_m, when the penalty is 0 points per intervention. Although a penalty of 0 points per intervention is unrealistic, it provides a test scenario where each approach should rapidly accumulate interventions, as there is no cost associated per intervention. Each of the crossover approaches rapidly increase the number of interventions in schedules, with the directed intervention techniques quicker to adapt to this trend.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the FDC crossover approach, in comparison to CalEB, TInSSel and UC, under a generational replacement strategy for a penalty of 0 points per intervention.

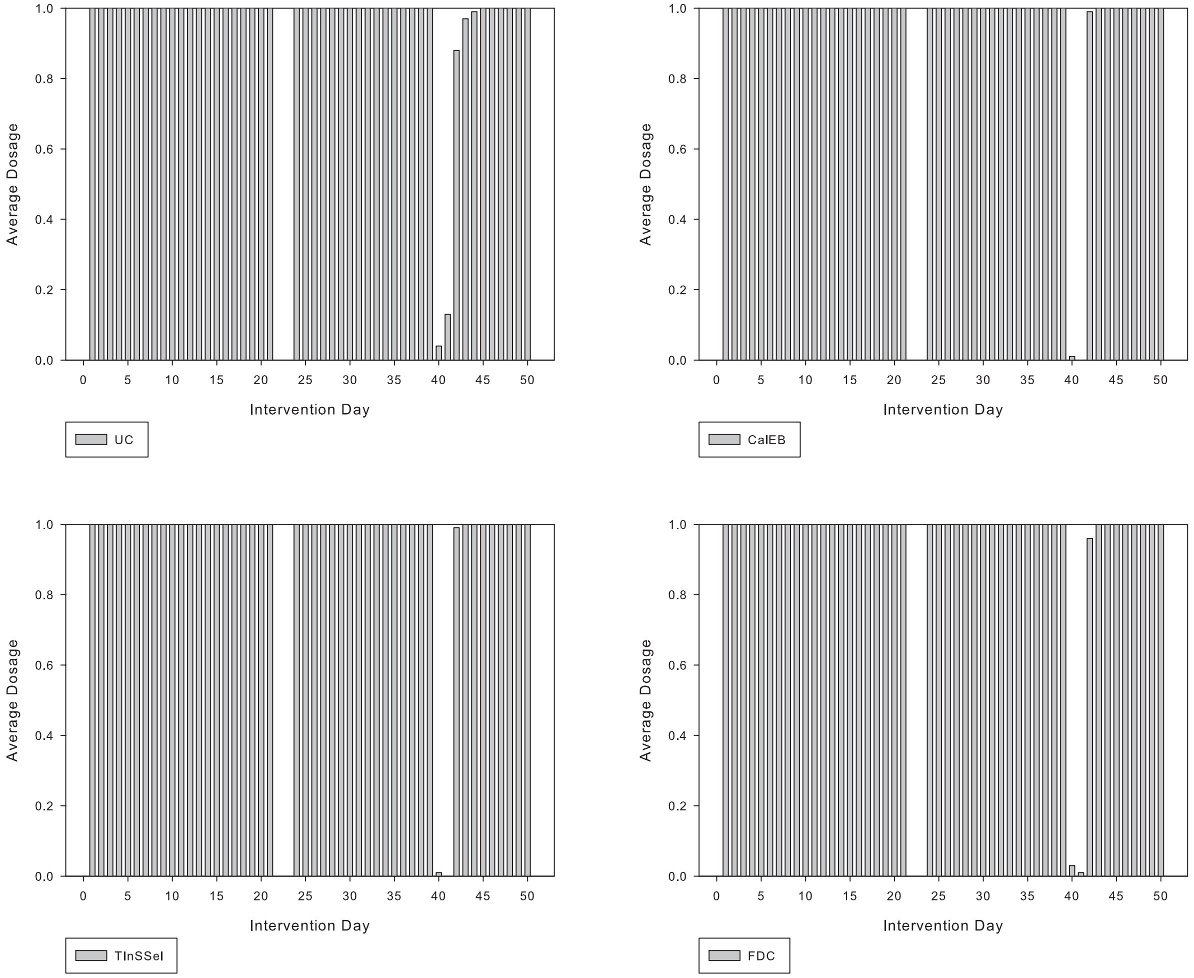
In the case of no mutation, as shown in Figure 9.1, FDC, CalEB and TInSSel are shown to use more interventions than UC. From the statistical analysis undertaken in Appendix E.16, after 5,000 FFEs, FDC is shown to outperform UC but is also outperformed by both TInSSel and CalEB. This section reviews the placement of interventions by each of the crossover approaches at 5,000 FFEs. The average placement of these interventions, in the case of no mutation, are shown in Figure 9.4. From this FDC can be seen to focus interventions on days 1 - 21 and 24 - 39 more often than UC, but less than both CalEB and TInSSel. This would explain the fitness difference between these approaches for this parameter setting.

Figure 9.4: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 0$ and $R = g$



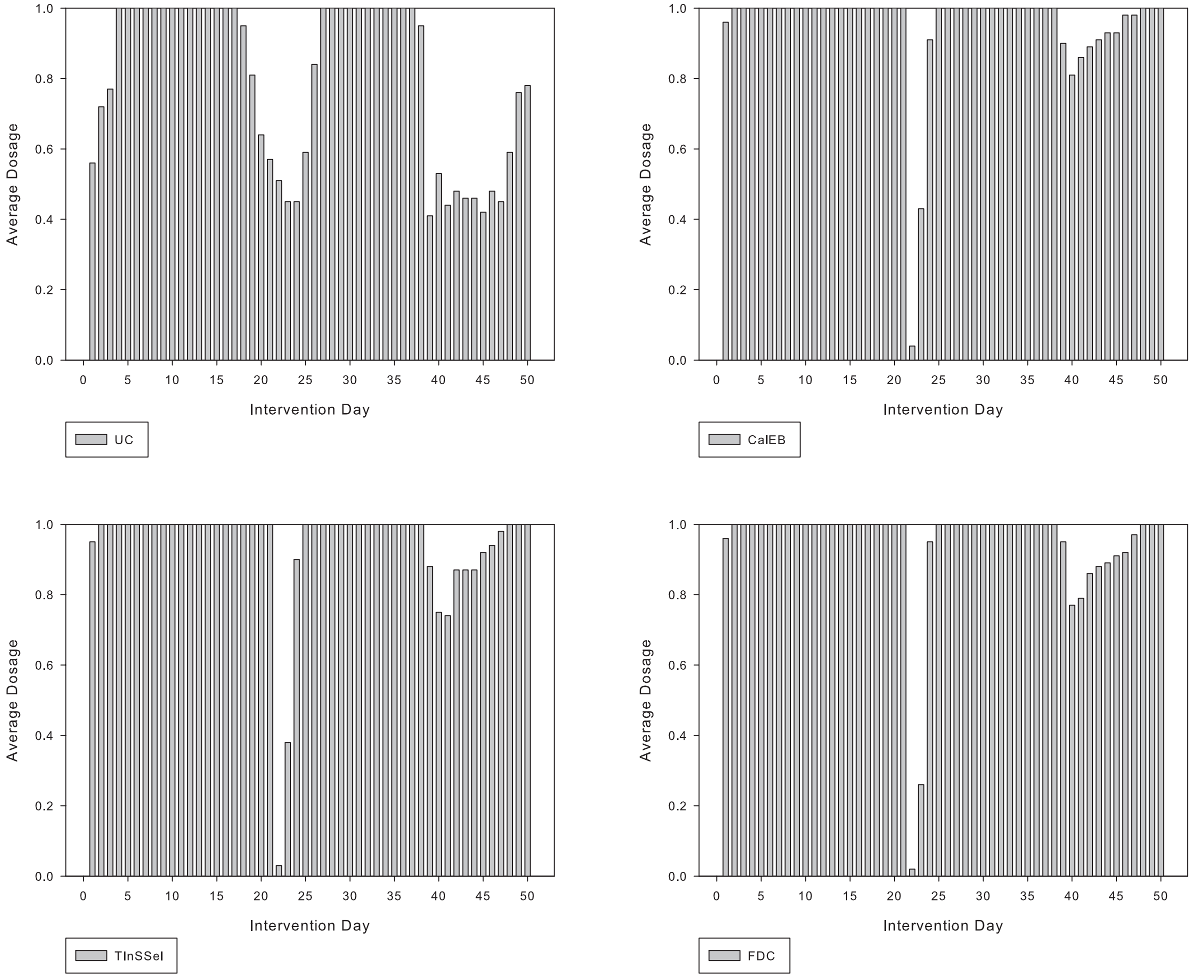
When the mutation rate, p_m, is increased to 0.005, as shown in Figure 9.2, each of the crossover approaches use a similar number of interventions. From the statistical analysis undertaken in Appendix E.17, after 5,000 FFEs, while there is no significant difference between FDC and CalEB or TInSSel, FDC is shown to outperform UC. The average placement of interventions by these approaches is shown in Figure 9.5. This shows that each of the directed approaches target days 40 and 41 less often than UC, while placing more emphasis on days 42 and 43.

Figure 9.5: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 0$ and $R = g$



When p_m is further increased to 0.05, as shown in Figure 9.3, CalEB uses the most interventions, followed by FDC and TInSSel, with UC using the least interventions. From the statistical analysis undertaken in Appendix E.18, after 5,000 FFEs, as with a mutation rate of 0.005, there is no statistical difference between the directed approaches, and FDC again outperforms UC. The average placement of interventions is shown in Figure 9.6. This shows that while the directed approaches follow a similar placement of interventions, UC does not target as many days as these approaches on average.

Figure 9.6: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 0$ and $R = g$

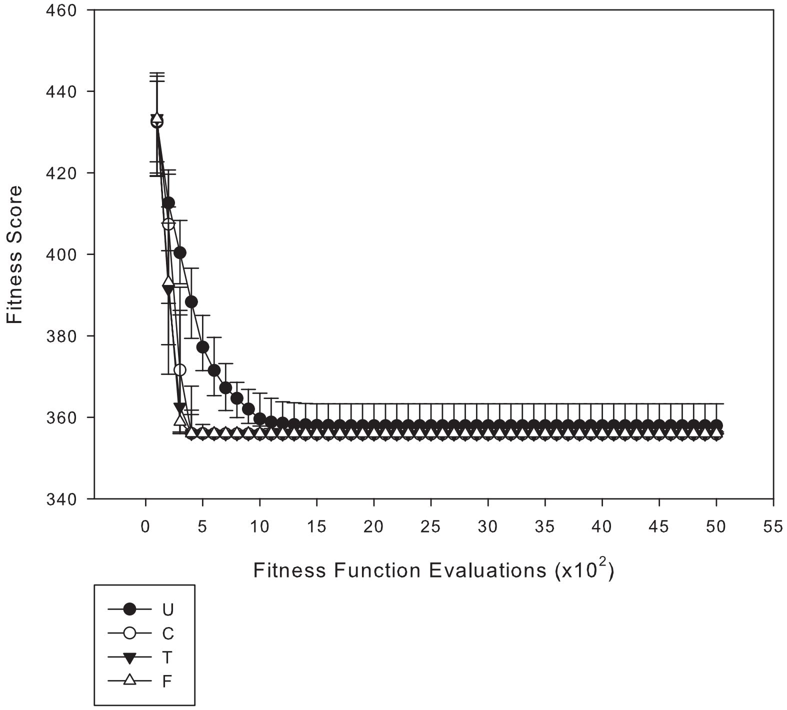


Steady state replacement

Section 9.2.1 reviewed FDC, compared to CalEB, TInSSel and UC for bio-control scheduling, with an intervention penalty of 0 points per intervention and generational replacement. This experiment details the performance of approaches under a steady state replacement strategy with intermediate parameter settings. Figures 9.7 to 9.9 show the fitness and intervention usage associated with these experiments for p_m levels of 0, 0.005 and 0.05 respectively.

Figure 9.7 shows that similar trends are shown by FDC in terms of both fitness scores and intervention usage regardless of replacement strategy when p_m is 0. These trends are similar to those of TInSSel. This pattern of FDC performing in a similar manner to TInSSel, both in terms of fitness scores and intervention usage is shown to hold when p_m is increased to 0.005 (Figure 9.8) and further increased to 0.05 (Figure 9.9). As with TInSSel, FDC is largely unaffected by the choice of replacement strategy applied for these experiment settings.

Figure 9.7: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$,$P = 0$ and $R = ss$



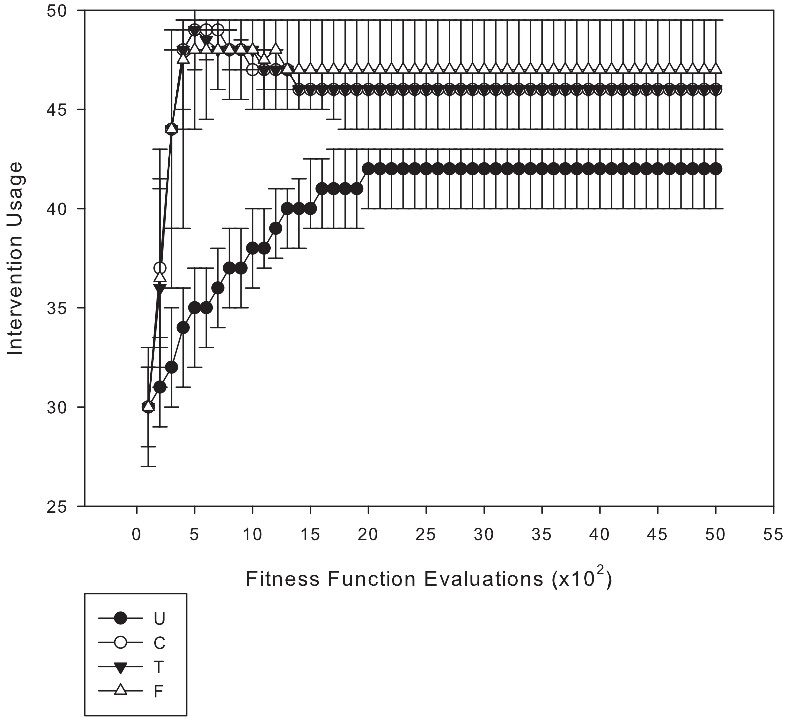
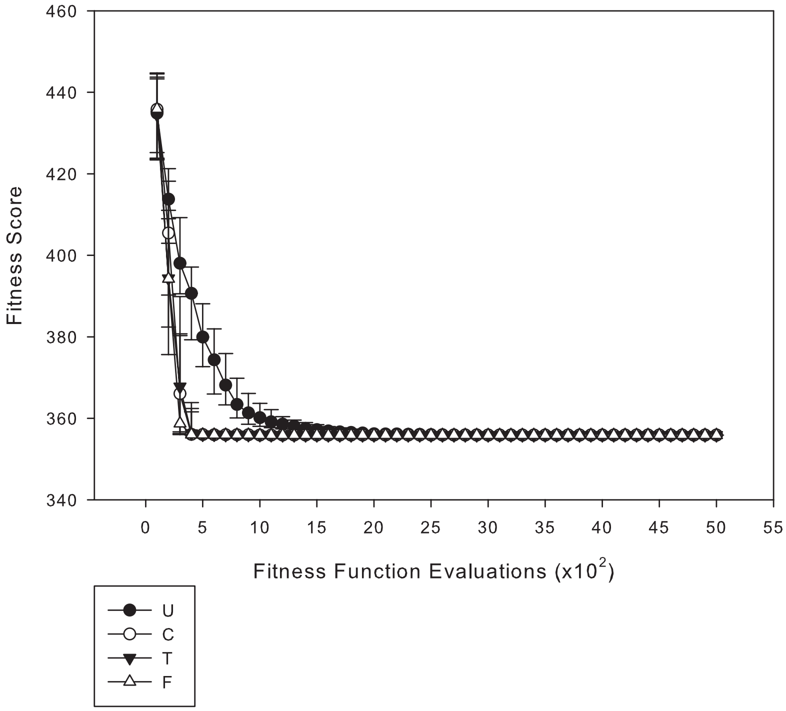


Figure 9.8: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 0$ and $R = ss$



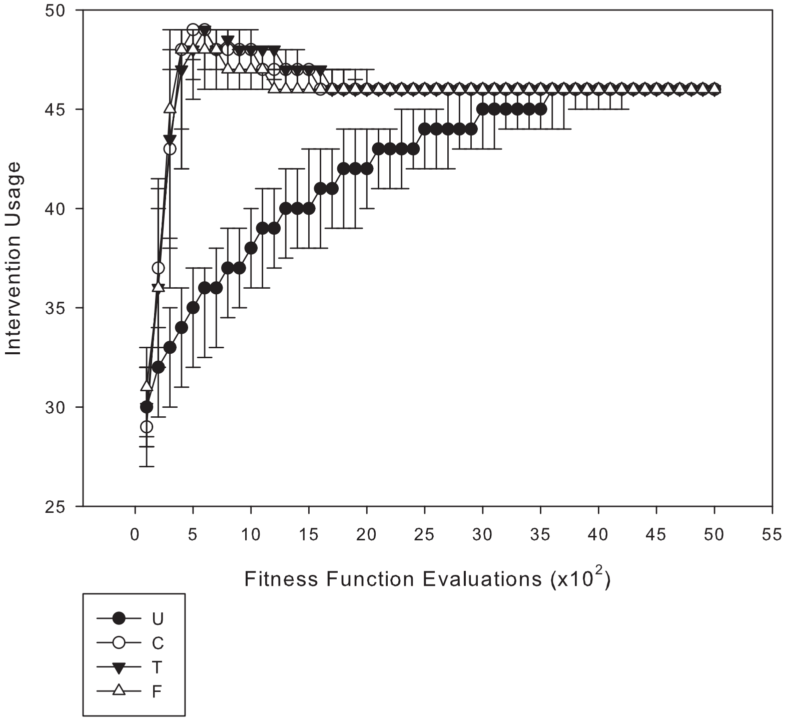
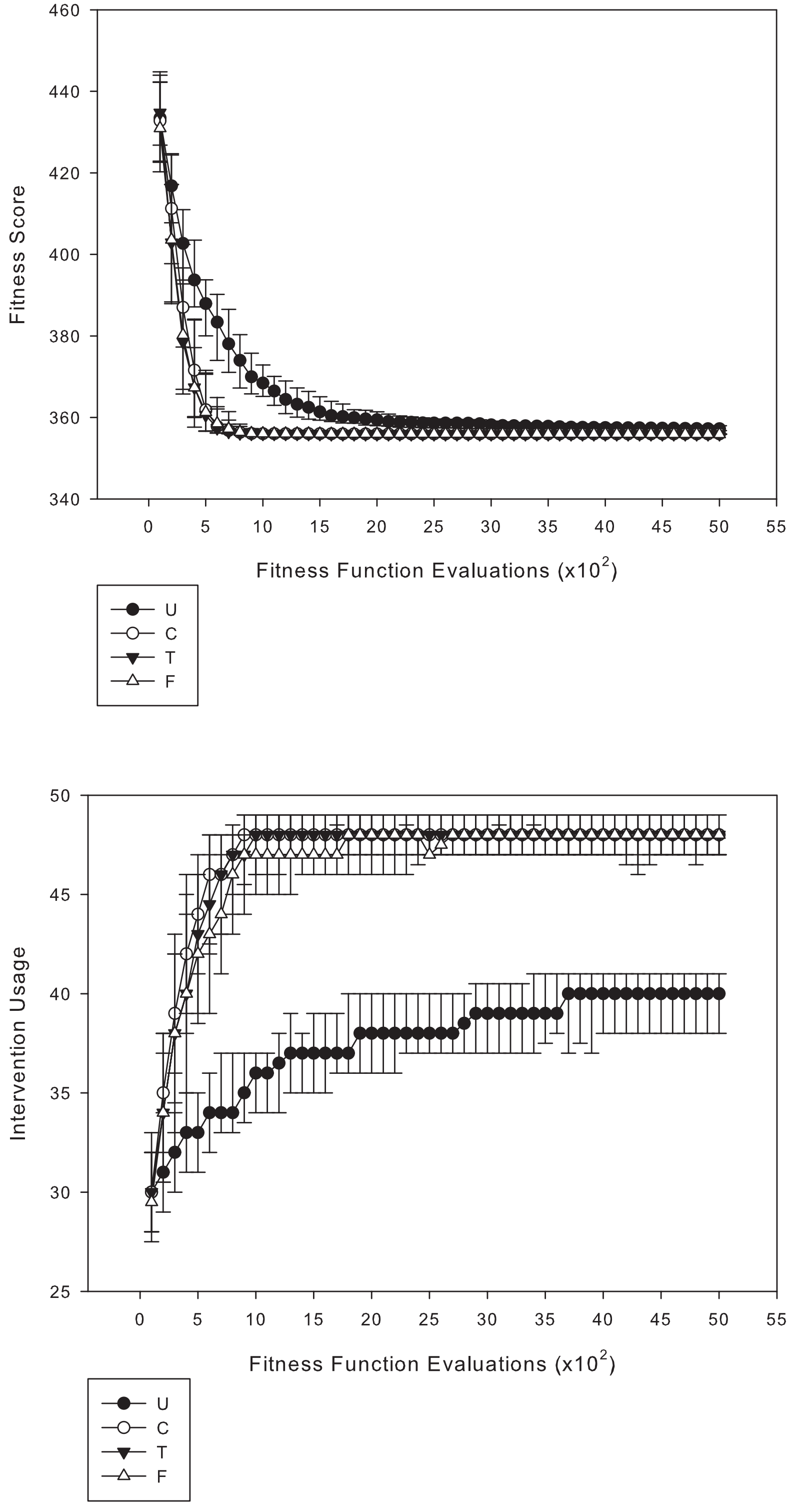


Figure 9.9: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 0$ and $R = ss$

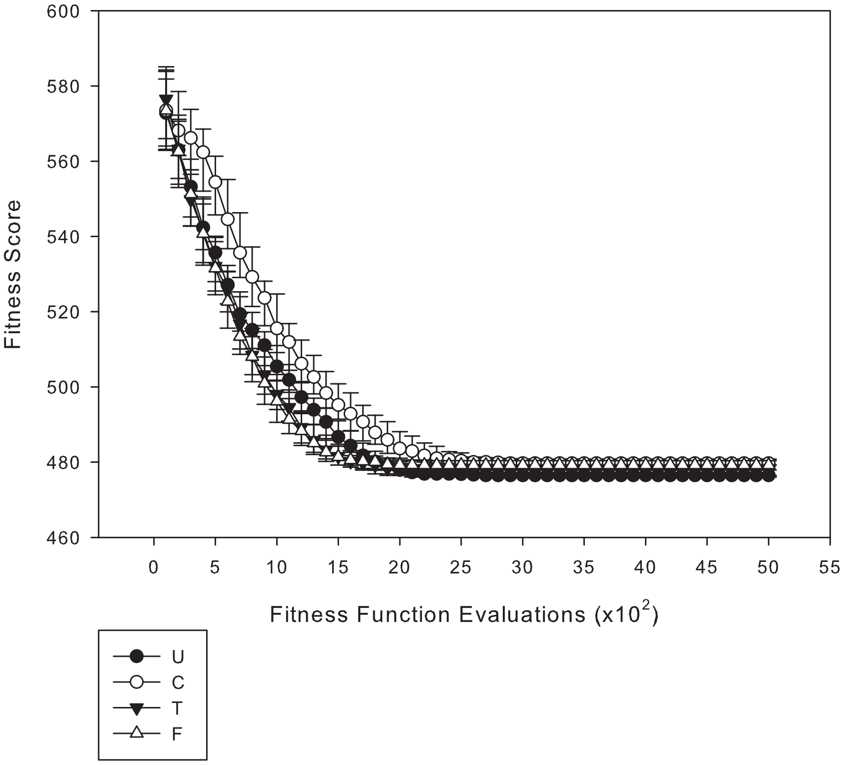


9.2.2 5 penalty points per intervention

This section describes how FDC compares to CalEB, TInSSel and UC at bio-control scheduling with a penalty of 5 points per intervention. As with the experiment detailed in Section 9.2.1, intermediate parameters of $N = 100$, and $p_c = 0.9$ will be used.

Figure 9.10 shows the fitness scores and associated intervention usage for a mutation rate, p_m, of 0. Figures 9.11 and 9.12 detail the fitness and intervention usage for approaches with p_m levels of 0.005 and 0.05 respectively.

Figure 9.10: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$,$P = 5$ and $R = g$



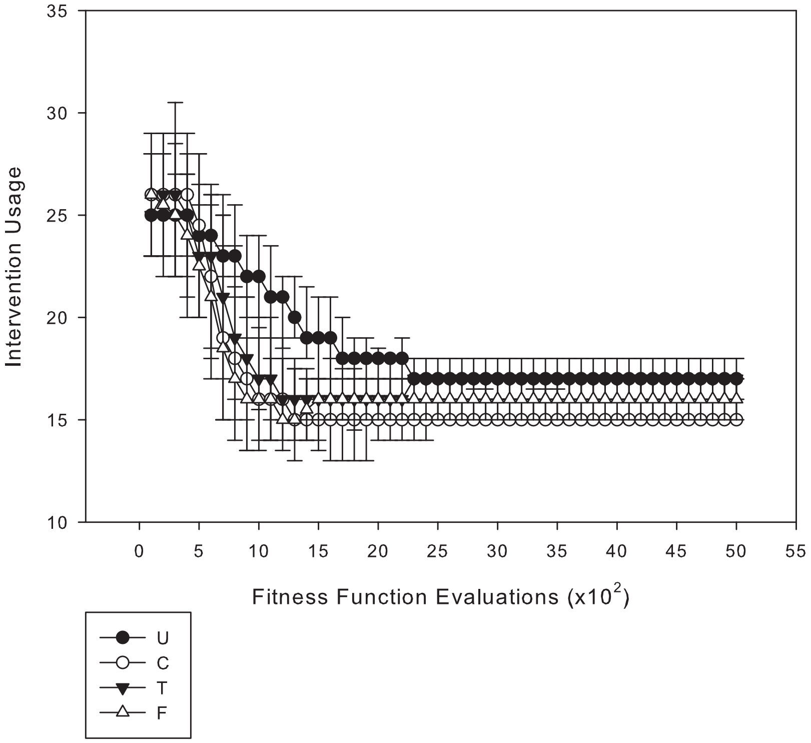
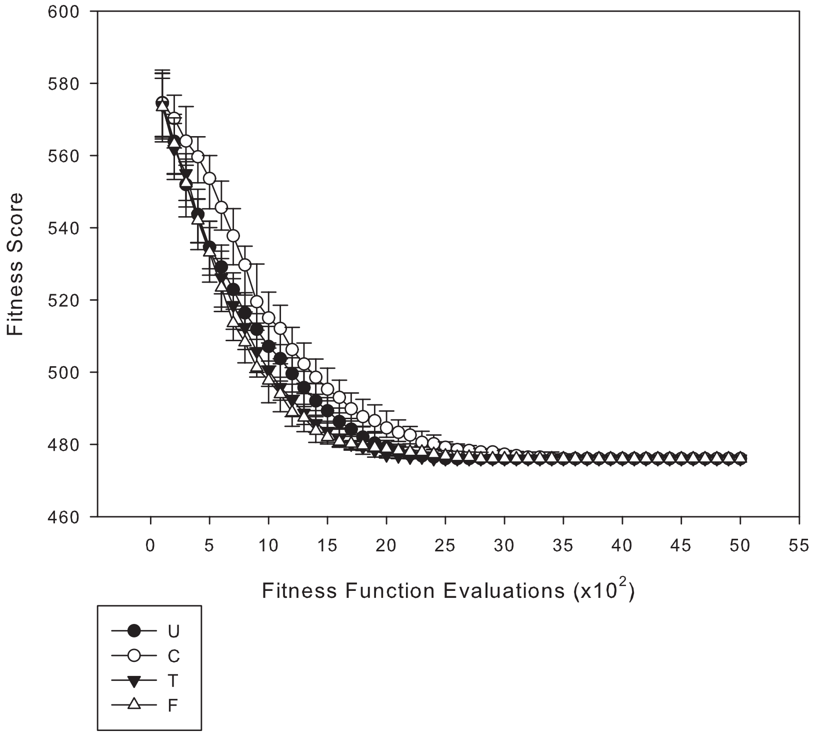


Figure 9.11: Results for $N = 100, p_c = 0.9, p_m = 0.005, P = 5$ and $R = g$



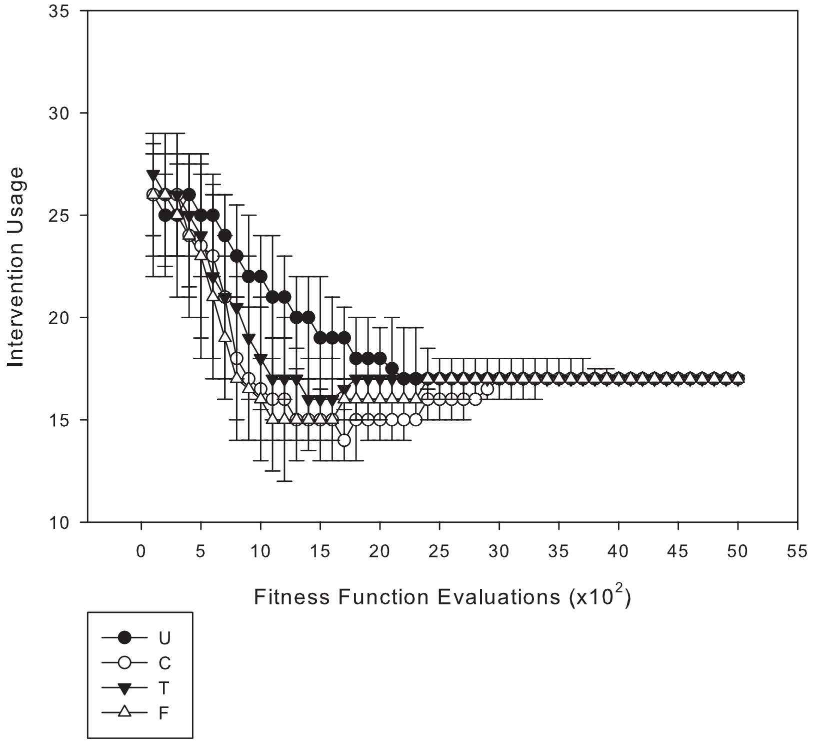
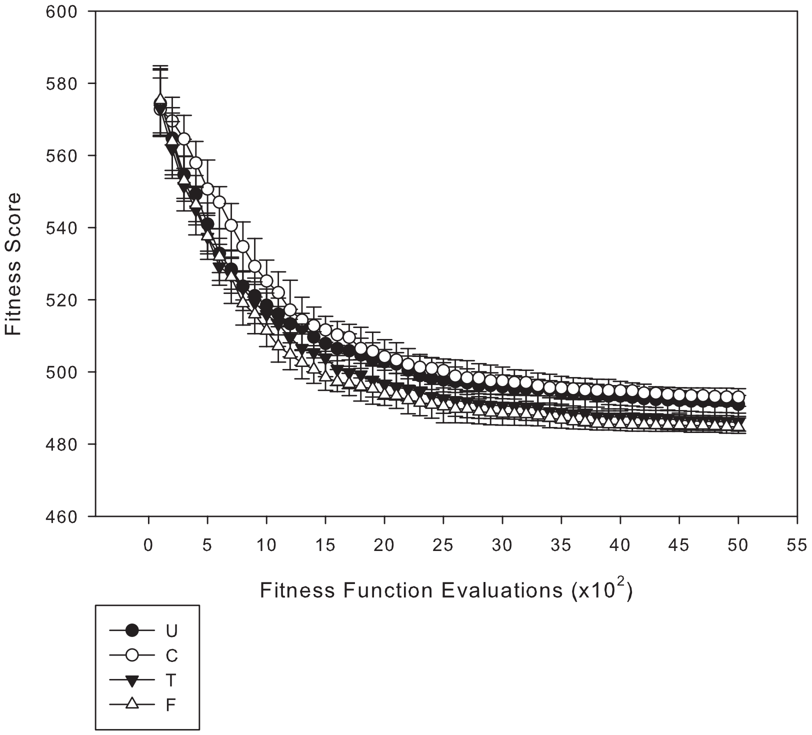


Figure 9.12: Results for $N = 100, p_c = 0.9, p_m = 0.05, P = 5$ and $R = g$



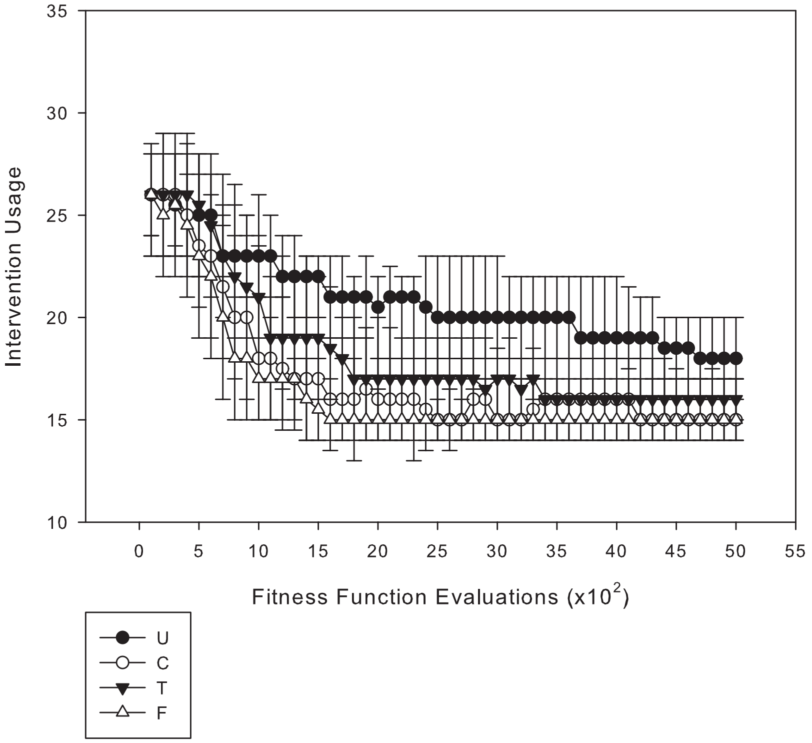


Figure 9.10 shows the fitness scores and intervention usage for the intermediate parameter settings with a penalty of 5 points per intervention and a p_m of 0. This shows that as with CalEB and TInSSel, FDC rapidly

reduces intervention levels. The fitness scores returned by FDC follow a similar trend to TInSSel, with the exception that FDC appears to settle on scores of slightly less fitness than those found by the other approaches when there is no mutation.

Figure 9.11 shows the fitness scores and intervention usage when the mutation rate, p_m, is increased to 0.005. For this mutation level, FDC is quicker to reduce intervention levels than TInSSel. The fitness of returned solutions are very similar between FDC and TInSSel for this mutation level.

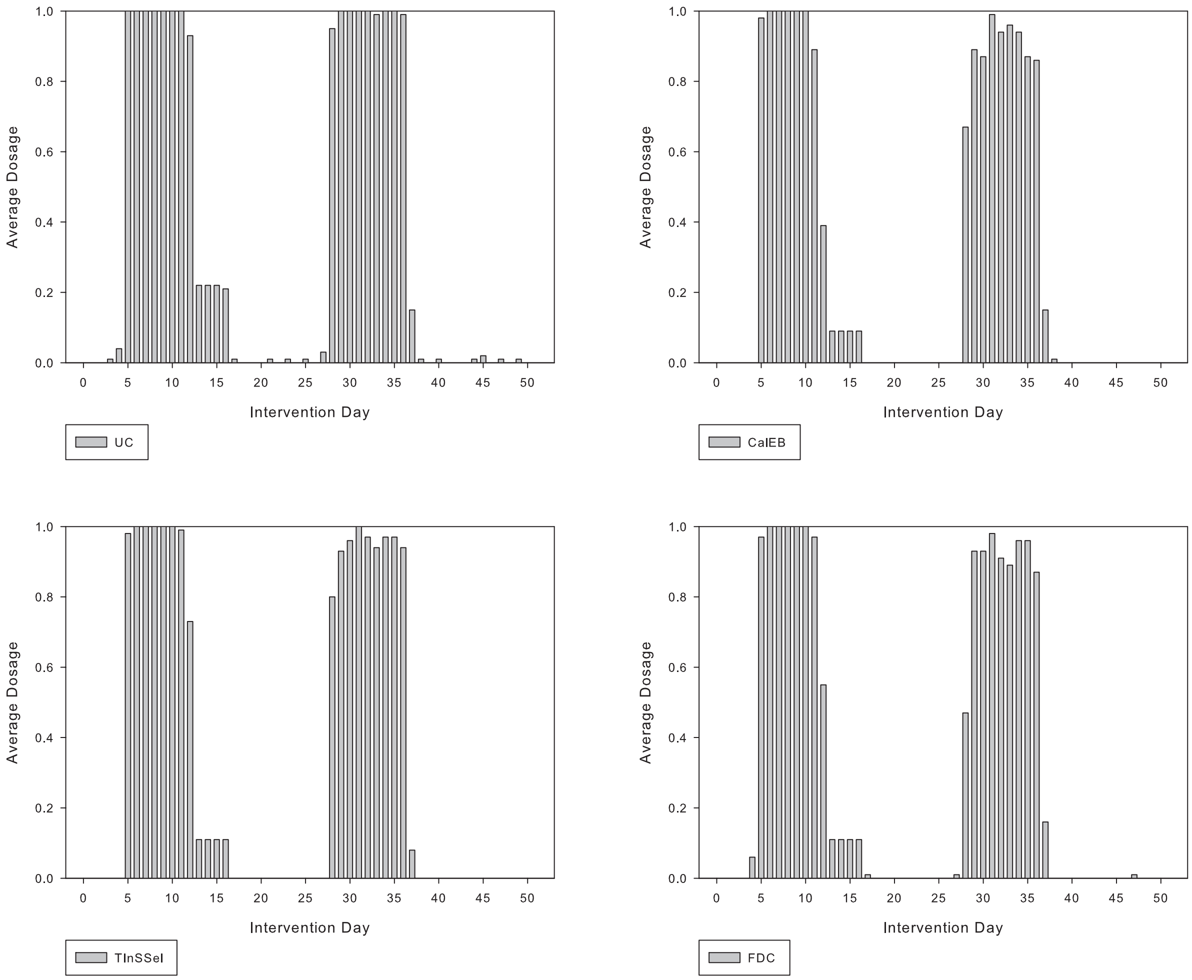
The effect of further increasing p_m to 0.05 is shown in Figure 9.12. FDC is quicker than the other approaches at reducing intervention levels for this mutation level. FDC also appears to return fitter scores than each of the other approaches for this level of mutation.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the FDC crossover approaches, in comparison to CalEB, TInSSel and UC, under a generational replacement strategy for a penalty of 5 points per intervention.

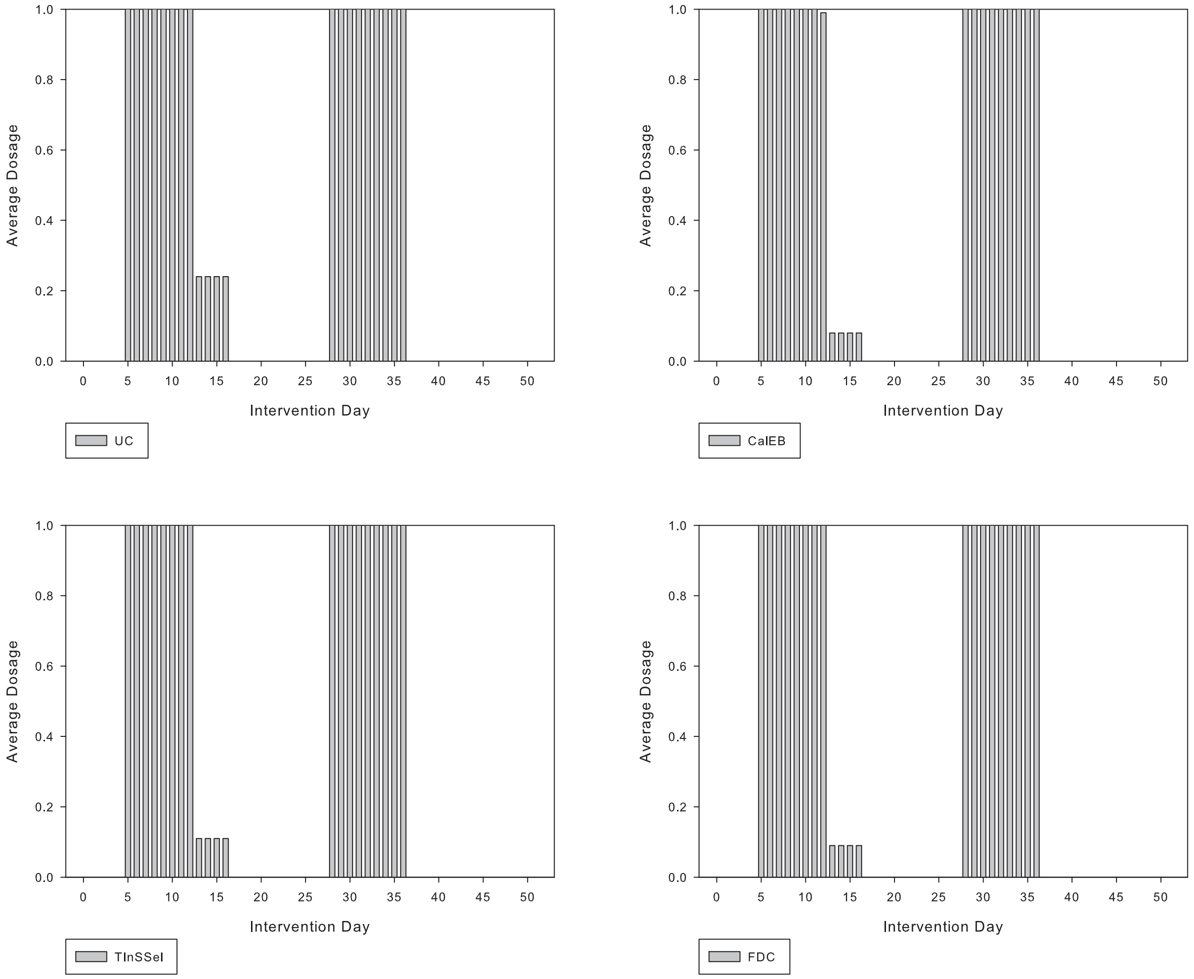
In the case of no mutation, as shown in Figure 9.10, UC and TInSSel are shown to use the most interventions, followed by FDC and then CalEB. From the statistical analysis undertaken in Appendix E.19, after 5,000 FFEs, FDC is shown to be outperformed by both UC and TInSSel. Through reviewing the average placement of these interventions, shown in Figure 9.13, FDC is shown to target days 5 - 12 and 28 - 36 less than both UC and TInSSel.

Figure 9.13: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 5$ and $R = g$



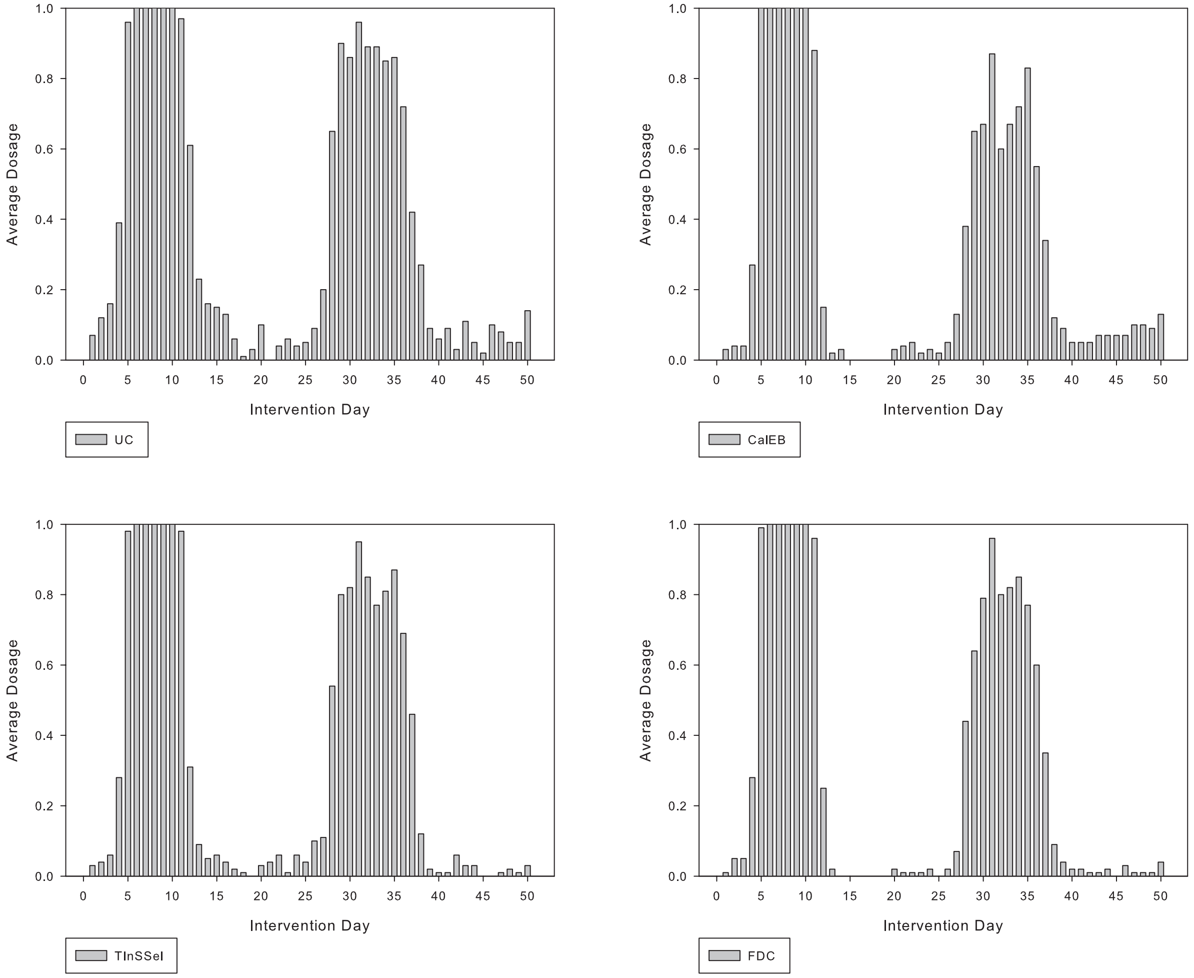
When the mutation rate is increased to 0.005, as shown in Figure 9.11, each of the crossover approaches are shown to converge to solutions using a similar number of interventions. From the statistical analysis undertaken in Appendix E.20, after 5,000 FFEs, there is no statistical difference between FDC and any of the other crossover approaches. Through reviewing the average placement of these interventions, shown in Figure 9.14, each of the approaches are shown to focus interventions on similar days, hence the similarity in fitness scores returned.

Figure 9.14: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 5$ and $R = g$



When the mutation rate is further increased to 0.05, as shown in Figure 9.12, UC uses more interventions than each of the directed approaches. From the statistical analysis undertaken in Appendix E.21, after 5,000 FFEs, FDC is shown to outperform both UC and CalEB. The average placement of these interventions are shown in Figure 9.15. This shows that FDC focuses interventions mainly on days 4 - 12 and 28 - 38 and has less interventions on other days than both UC and CalEB.

Figure 9.15: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 5$ and $R = g$



Steady state replacement

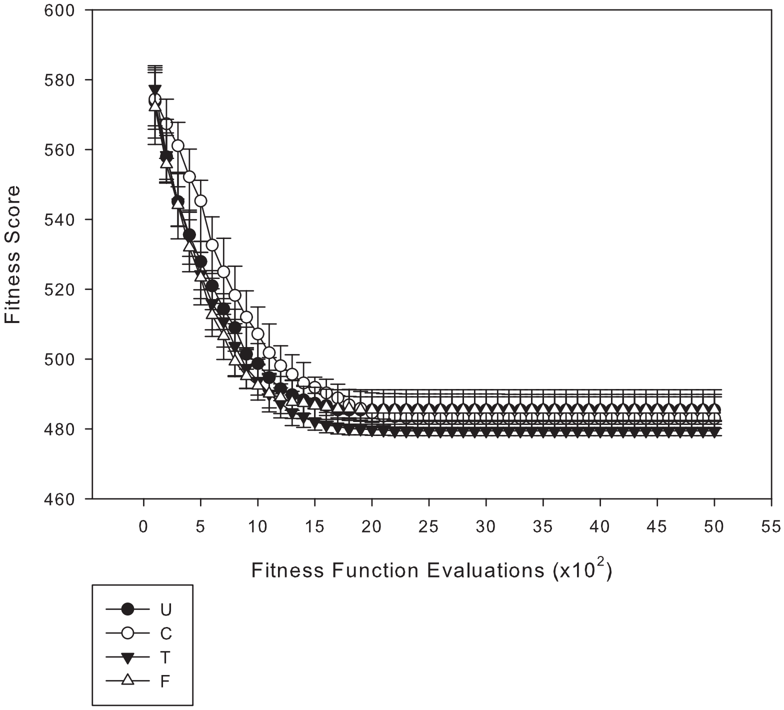
We now consider the steady state replacement strategy. Figures 9.16 to 9.18 show the fitness scores and intervention usage when $R = ss$ with p_m values of 0, 0.005 and 0.05 respectively.

Figure 9.16 shows that when p_m is 0 the FDC approach uses the same number of interventions as CalEB, which is less than those used by TInSSel. As with the generational equivalent, FDC does not find scores of the same fitness as those returned by TInSSel for this p_m level.

Figure 9.18 details the effect of increasing p_m to 0.005. Although TInSSel, CalEB and FDC all reduce intervention levels, FDC does so in a more direct way. FDC is quicker than both TInSSel and CalEB to reduce in intervention levels, and also reduces further in intervention levels than both TInSSel and CalEB. After 1,000 FFEs, FDC has reduced to approximately 14 interventions and then increases in intervention usage to approximately 17 interventions. This is the same number of interventions that UC, TInSSel and CalEB settle to for this problem.

When p_m is further increased to 0.05, as shown in Figure 9.18, similar trends are displayed by FDC in terms of both intervention usage and fitness scores regardless of the choice of replacement strategy.

Figure 9.16: Results for $N = 100, p_c = 0.9, p_m = 0$,$P = 5$ and $R = ss$



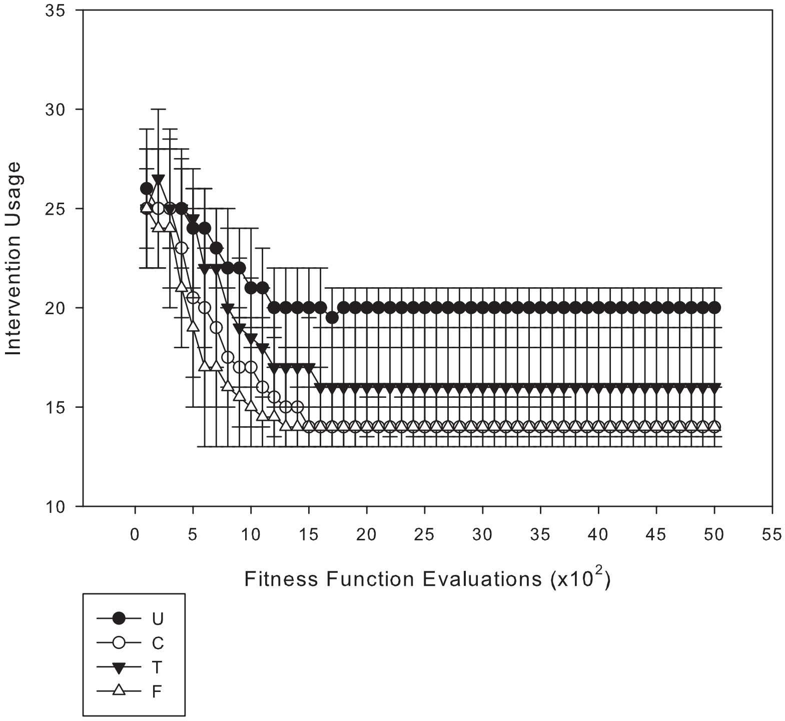


Figure 9.17: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 5$ and $R = ss$

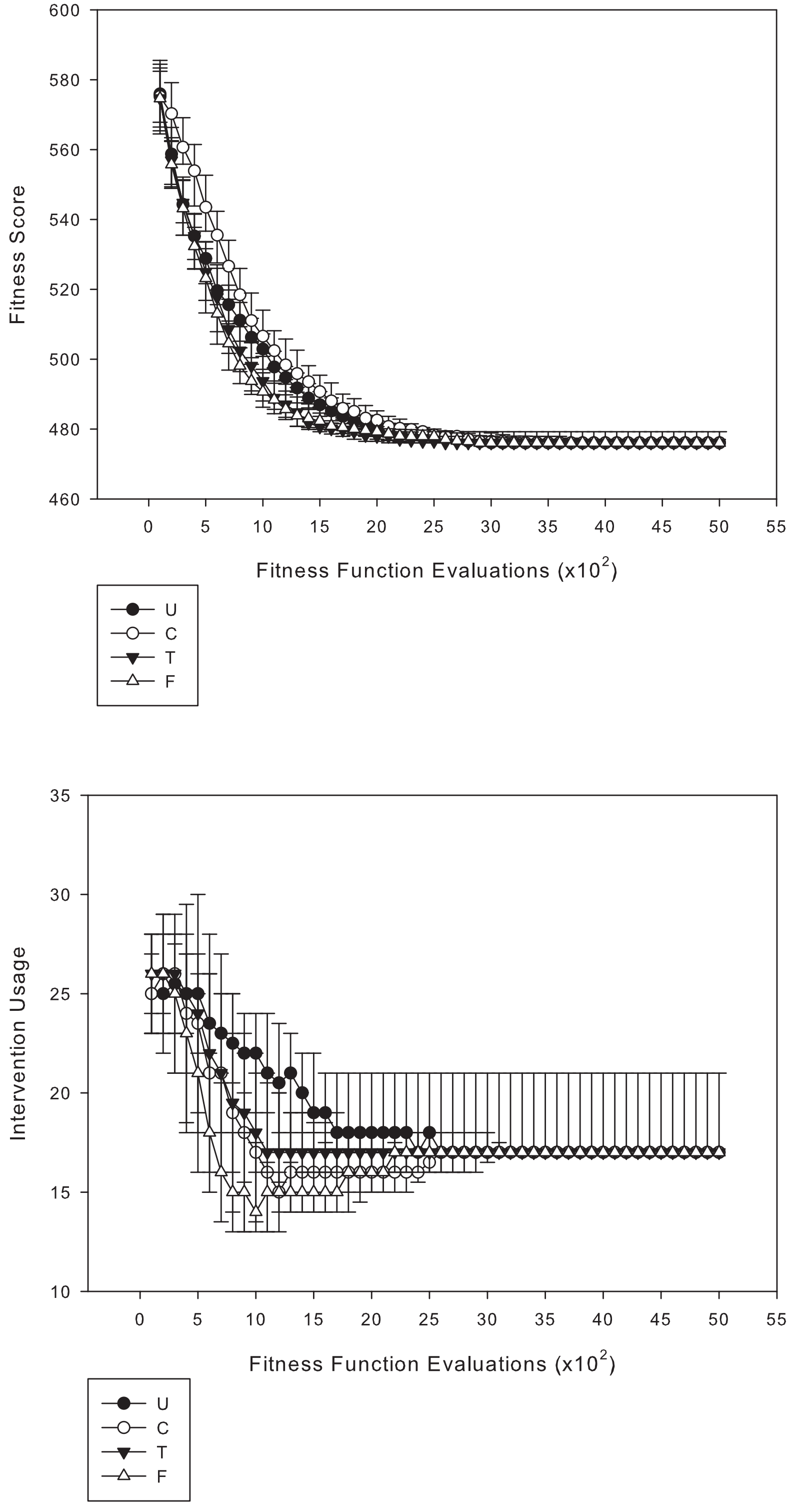
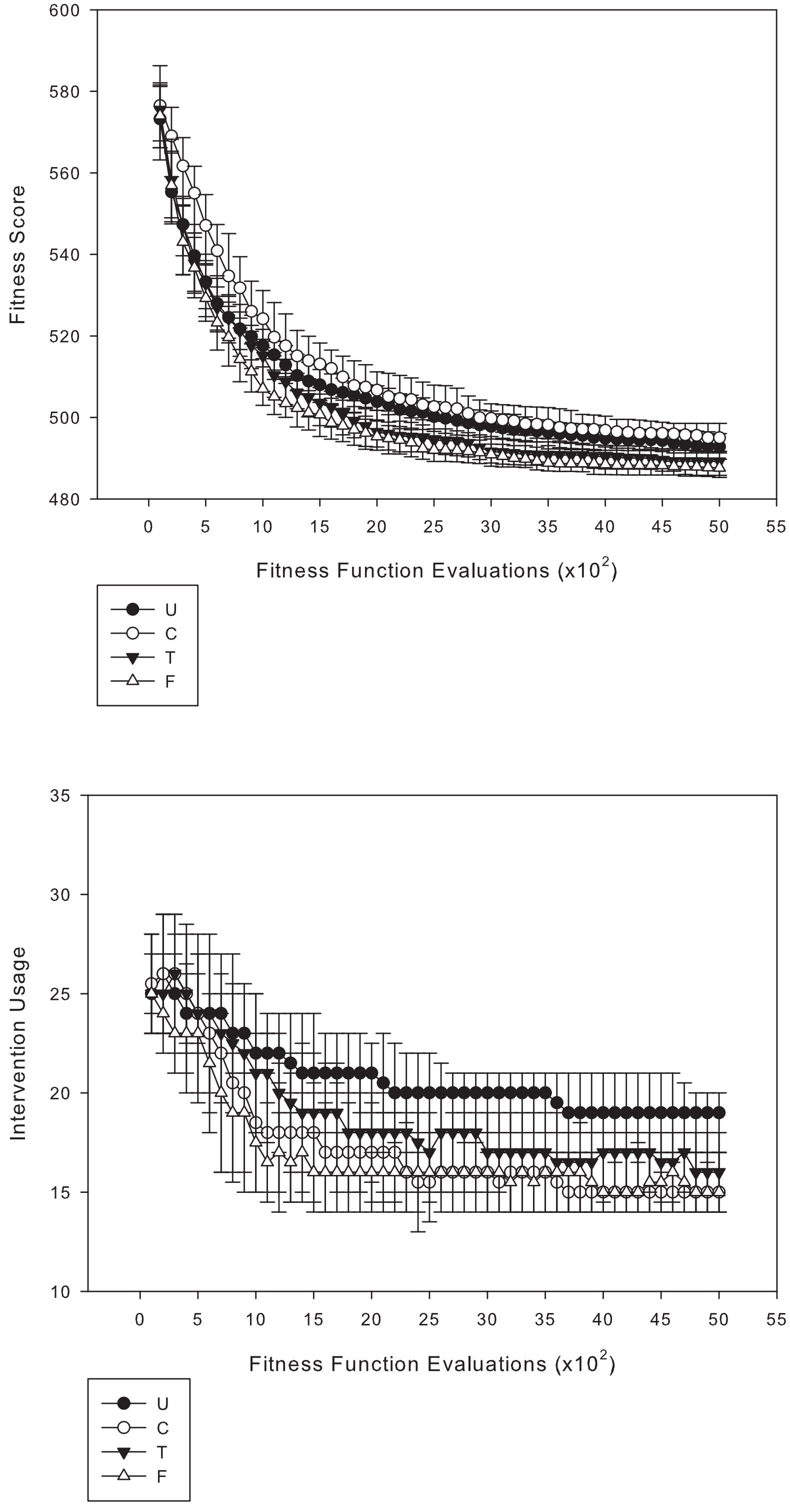


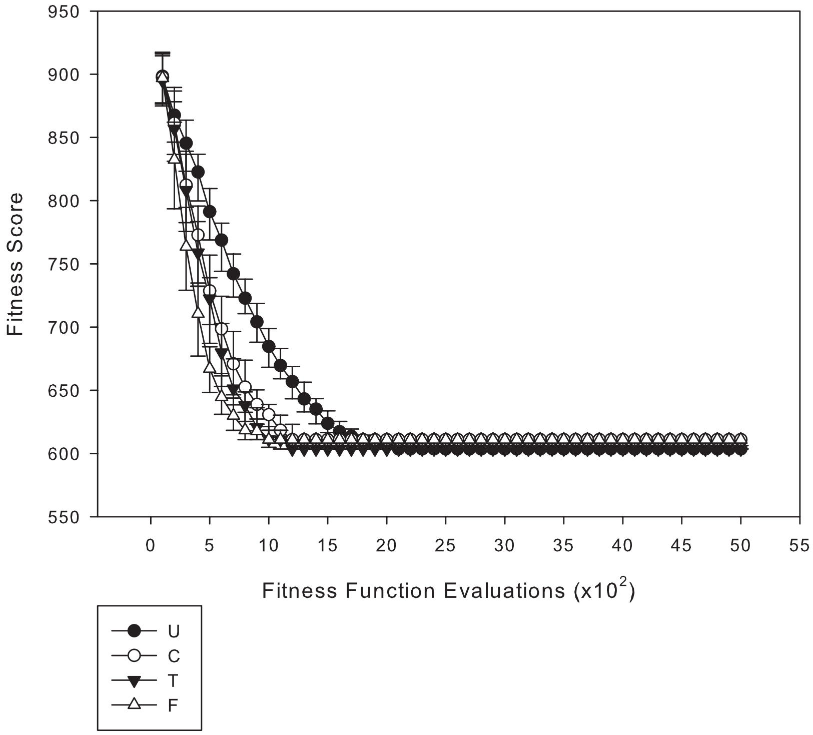
Figure 9.18: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 5$ and $R = ss$



9.2.3 20 penalty points per intervention

The next experiment considers the effect of increasing the number of penalty points per intervention from 5 to 20. Figures 9.19 to 9.21 show the fitness scores and associated intervention usage for a penalty of 20 points per intervention and p_m values of 0, 0.005 and 0.05 respectively.

Figure 9.19: Results for $N = 100, p_c = 0.9, p_m = 0, P = 20$ and $R = g$



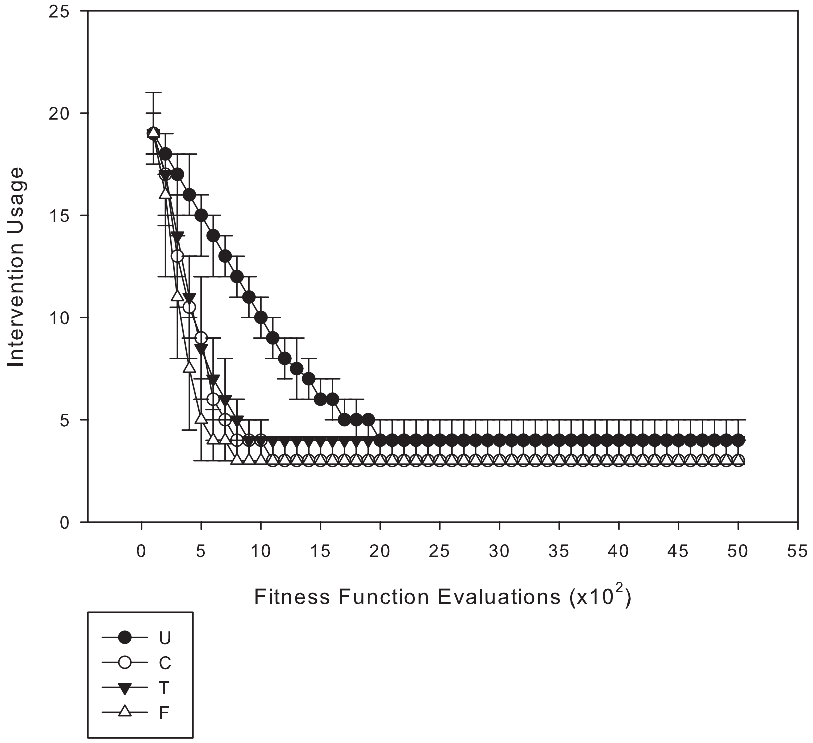
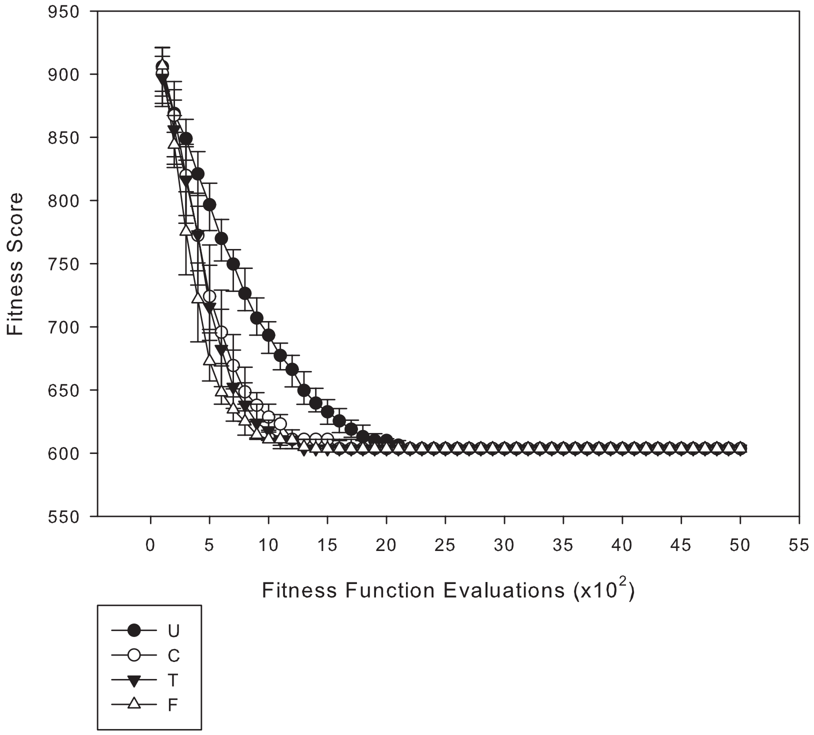


Figure 9.20: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 20$ and $R = g$



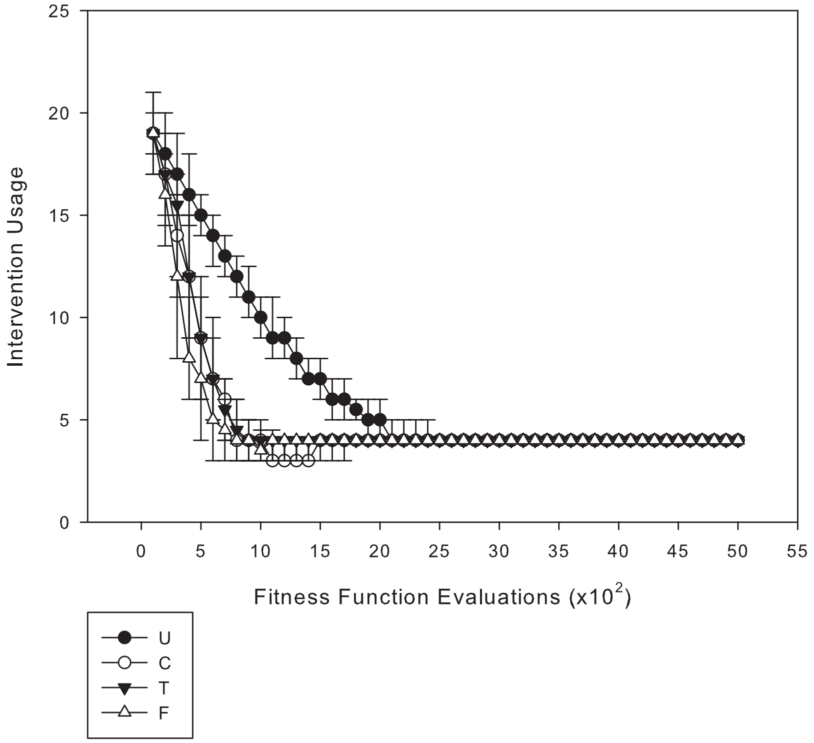
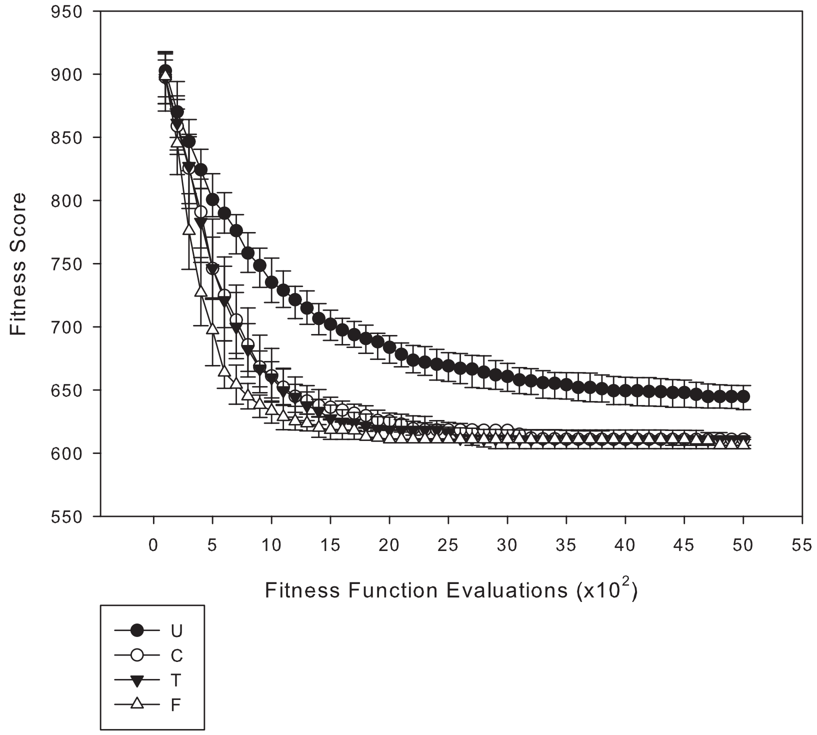


Figure 9.21: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 20$ and $R = g$



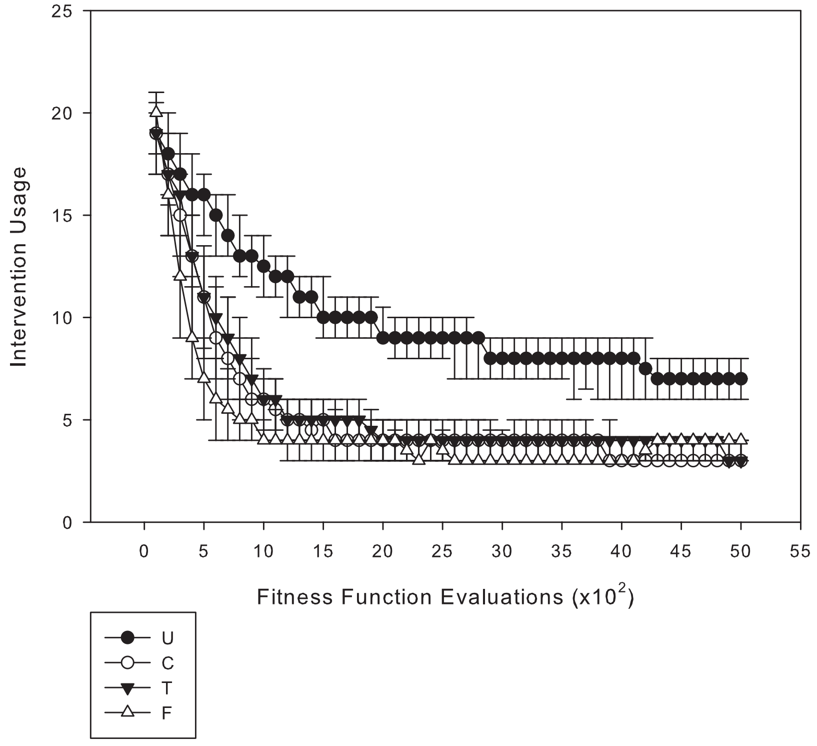


Figure 9.19 shows that when p_m is 0, FDC is quicker than CalEB and TInSSel at reducing intervention levels. Although this rapid reduction in intervention levels returns fitter scores for approximately the first 1,000 FFEs,

from this point FDC returns less fit scores than the other approaches.

Figure 9.20 details the experiment when the probability of mutation, p_m is increased to 0.005. As with the case when p_m was 0, FDC is quicker to reduce intervention levels than the other approaches. This accounts for the fitter scores returned by FDC over the first 1,000 FFEs.

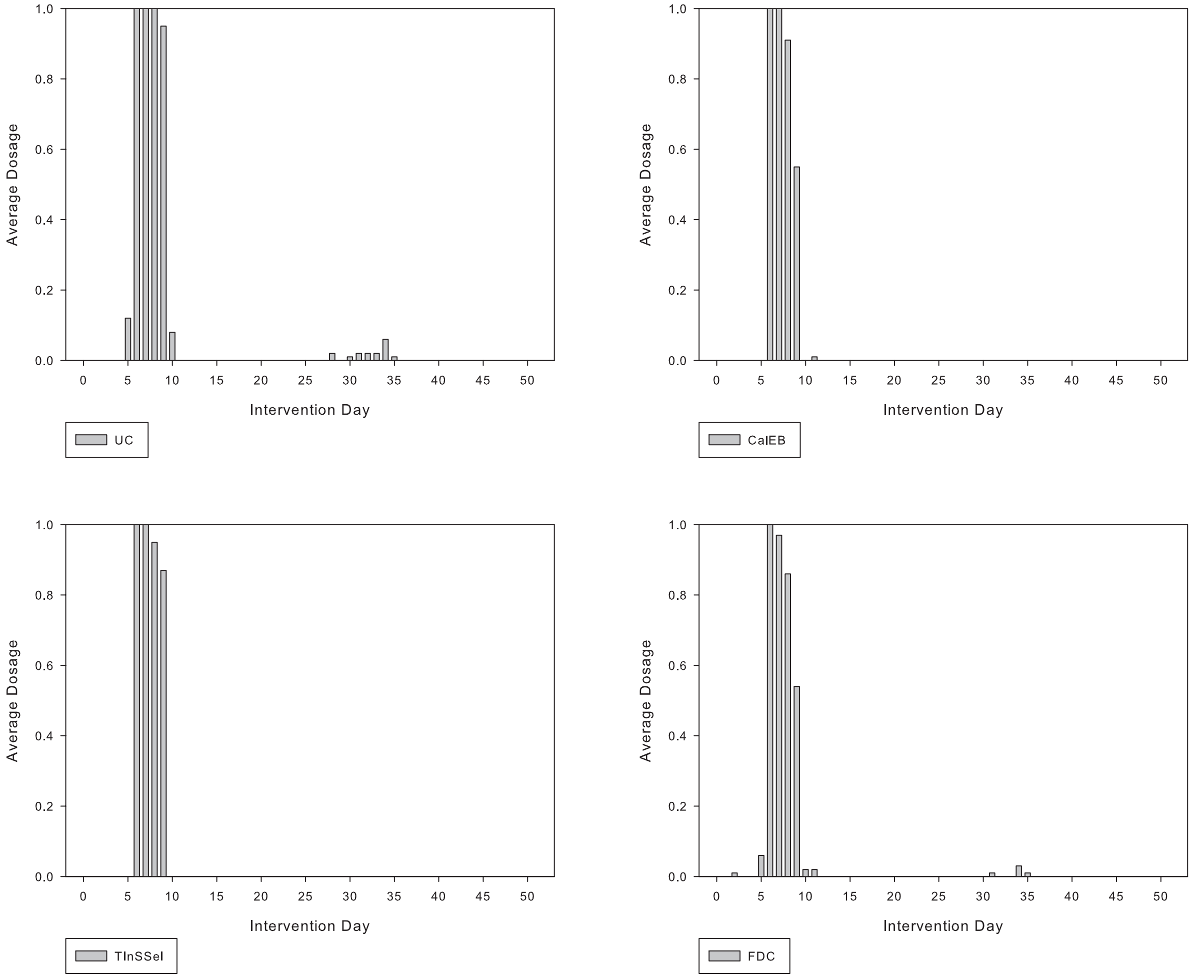
When p_m is increased further to 0.05, as shown in Figure 9.21, FDC is again quicker at reducing intervention levels. This rapid reduction in intervention levels correlates to FDC producing fitter scores than the other approaches, most evidently in the first 1,500 FFEs.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the FDC crossover approaches, in comparison to CalEB, TInSSel and UC, under a generational replacement strategy for a penalty of 20 points per intervention.

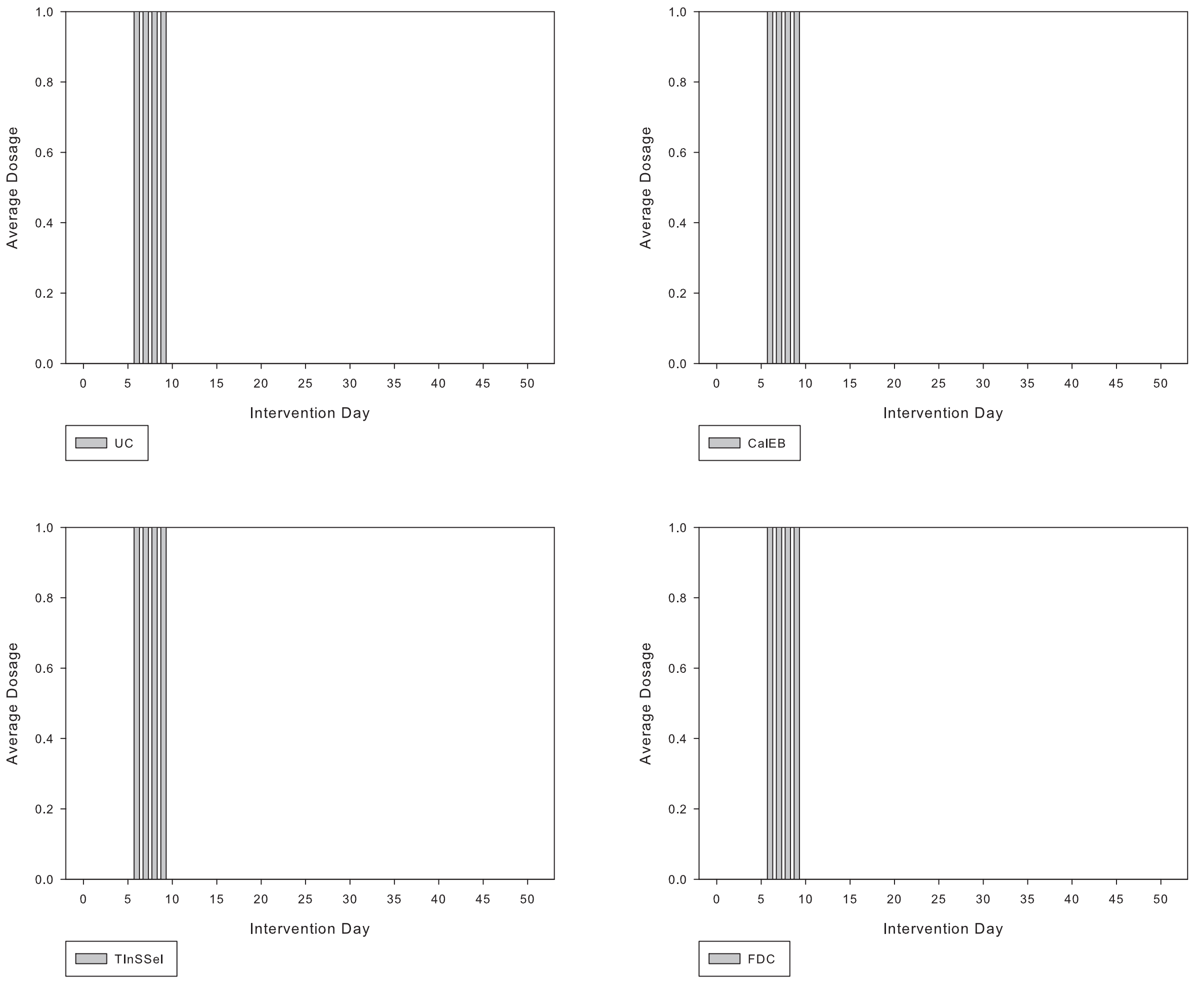
In the case of no mutation, as shown in Figure 9.19, TInSSel and UC are shown to use more interventions than both CalEB and FDC. From the statistical analysis undertaken in Appendix E.22, after 5,000 FFEs, FDC is shown to be outperformed by both TInSSel and UC. The average placement of interventions are shown in Figure 9.22. This shows that FDC does not target days 5 - 9 as strongly as both UC and TInSSel.

Figure 9.22: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 20$ and $R = g$



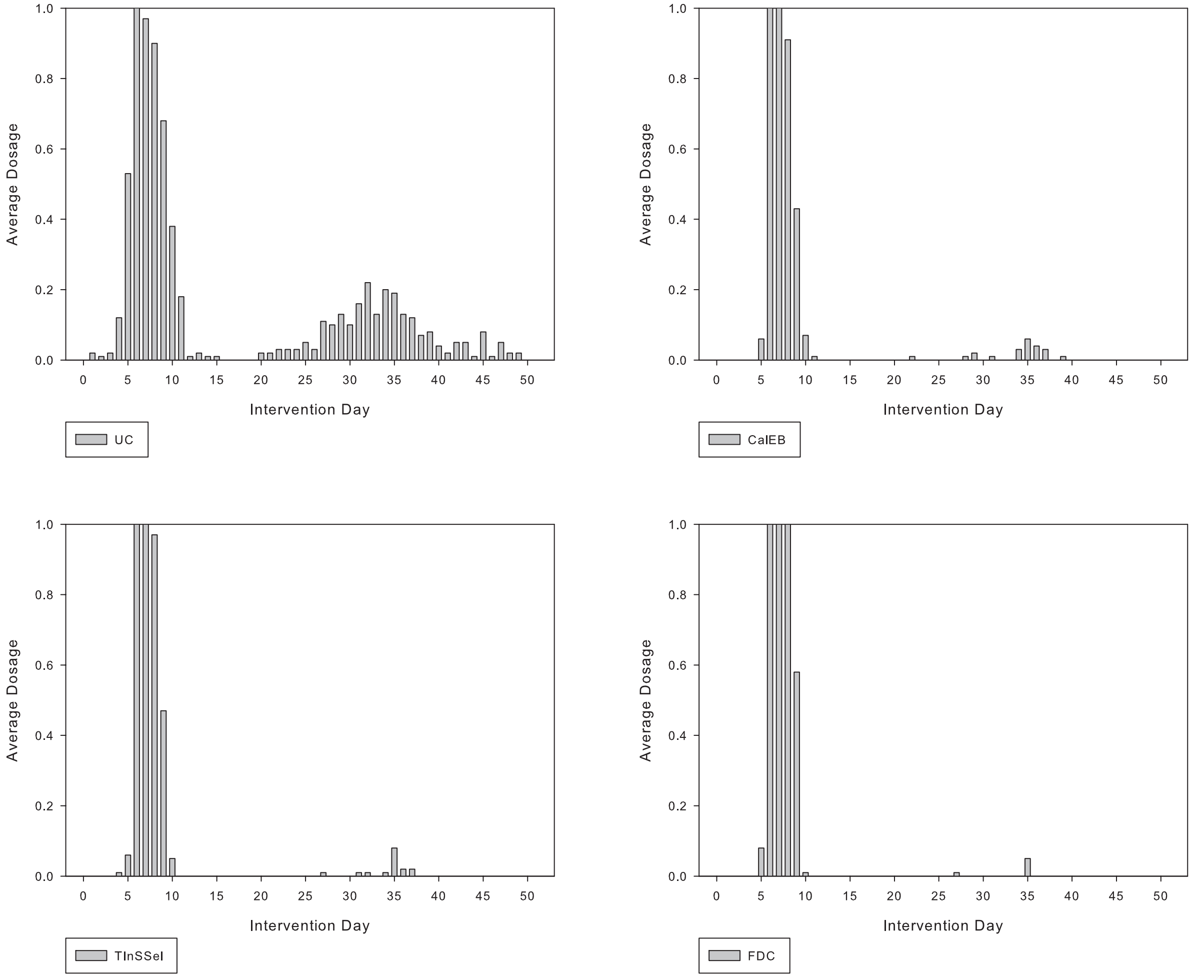
When the mutation rate is increased to 0.005, as shown in Figure 9.20, each of the crossover approaches are shown to use a similar number of interventions. From the statistical analysis undertaken in Appendix E.23, after 5,000 FFEs, there was no statistical difference between crossover approaches. The average placement of interventions are shown in Figure 9.23 and this shows that each of the approaches target days 6 - 9, hence the similar fitness scores returned.

Figure 9.23: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 20$ and $R = g$



When the mutation rate is further increased to 0.05, as shown in Figure 9.21, UC is shown to use the most interventions, followed by FDC and then TInSSel and CalEB. From the statistical analysis undertaken in Appendix E.24, after 5,000 FFEs, FDC is shown to outperform both UC and CalEB. The average placement of interventions are shown in Figure 9.24. This shows that FDC targets days 6 - 9 more strongly than both CalEB and UC and also has less interventions on other days of the schedule.

Figure 9.24: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 20$ and $R = g$



Steady state replacement

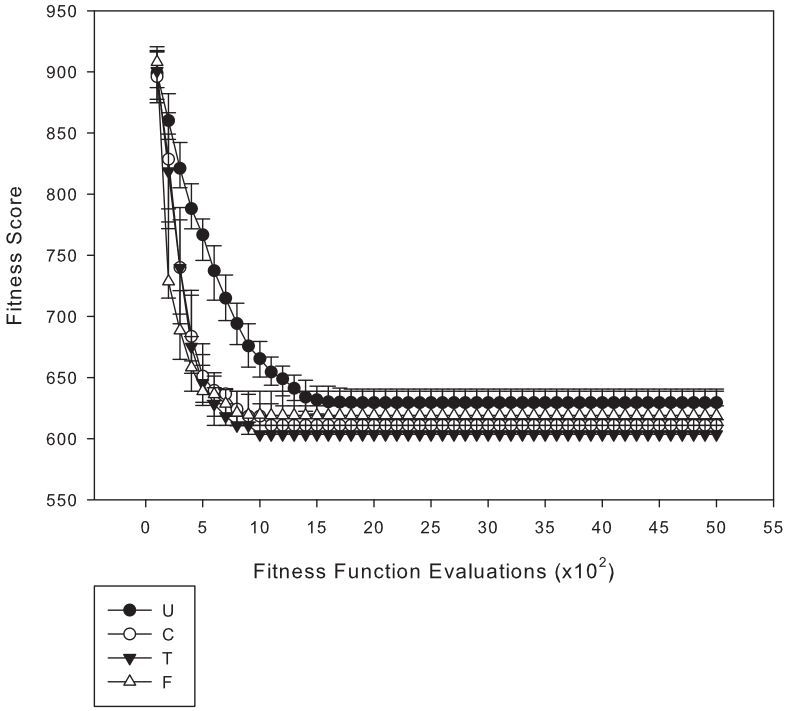
We now repeat the previous evaluation for a steady state replacement strategy. Figures 9.25 to 9.27 show the fitness scores and associated intervention usage for a steady state replacement strategy with a penalty of 20 points per intervention.

For a p_m of 0, as shown in Figure 9.25, FDC uses a similar number of interventions as CalEB. This was also the case for a generational replacement strategy. The fitness scores returned by FDC are less fit than those found under the generational equivalent for this mutation level.

Figure 9.26 shows the fitness scores and intervention usage when p_m is increased to 0.005. For this mutation level FDC displays trends, in terms of both fitness and intervention usage, very similar to those shown with a generational replacement strategy.

When p_m is further increased to 0.05, as shown in Figure 9.27, FDC again displays similar fitness and intervention patterns as demonstrated under a generational replacement strategy for this mutation level.

Figure 9.25: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$,$P = 20$ and $R = ss$



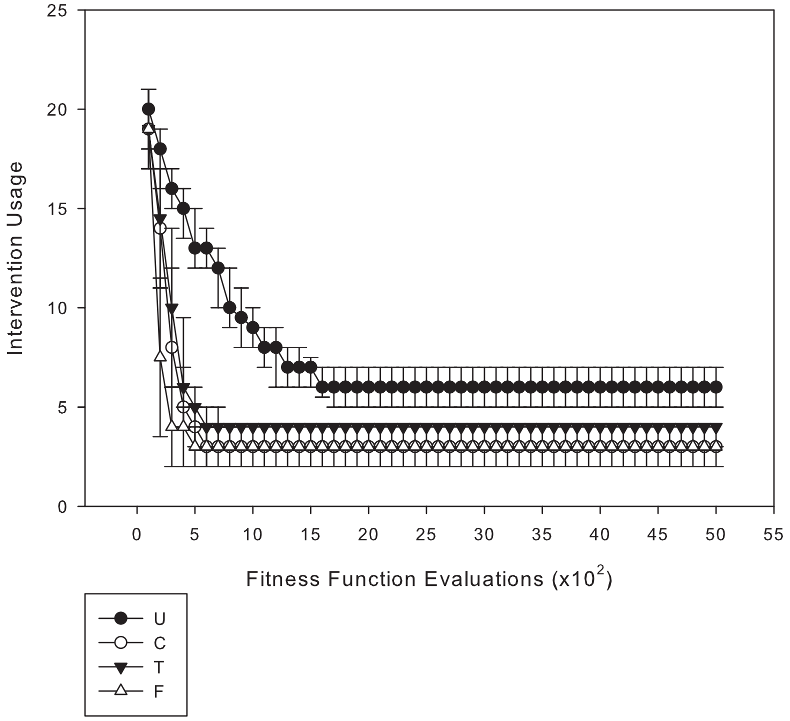


Figure 9.26: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 20$ and $R = ss$

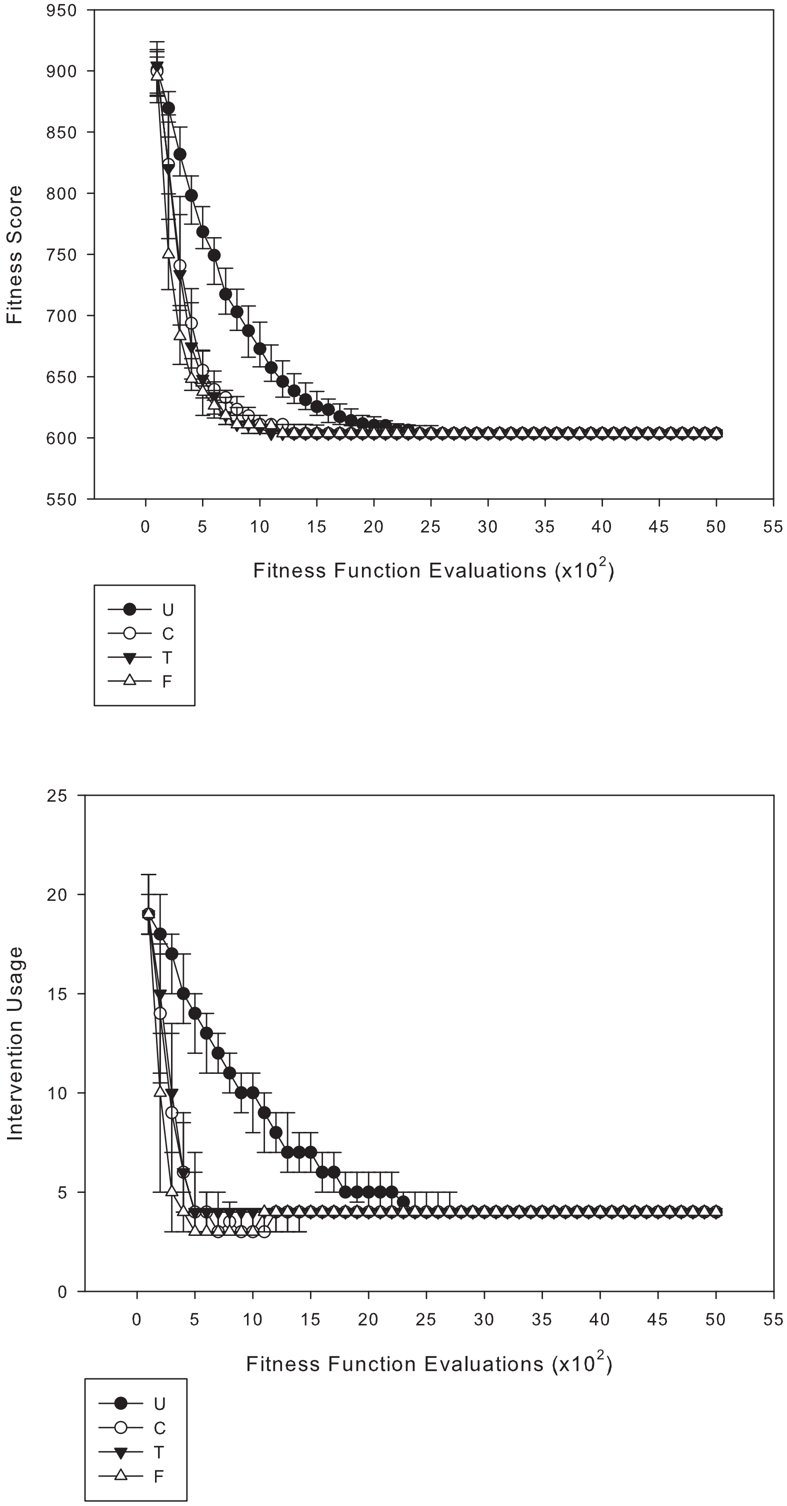
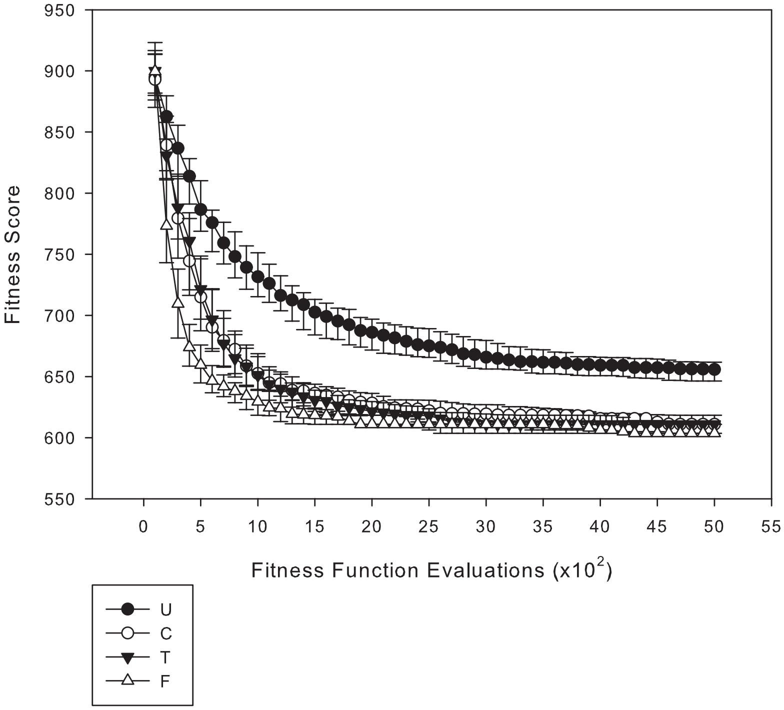
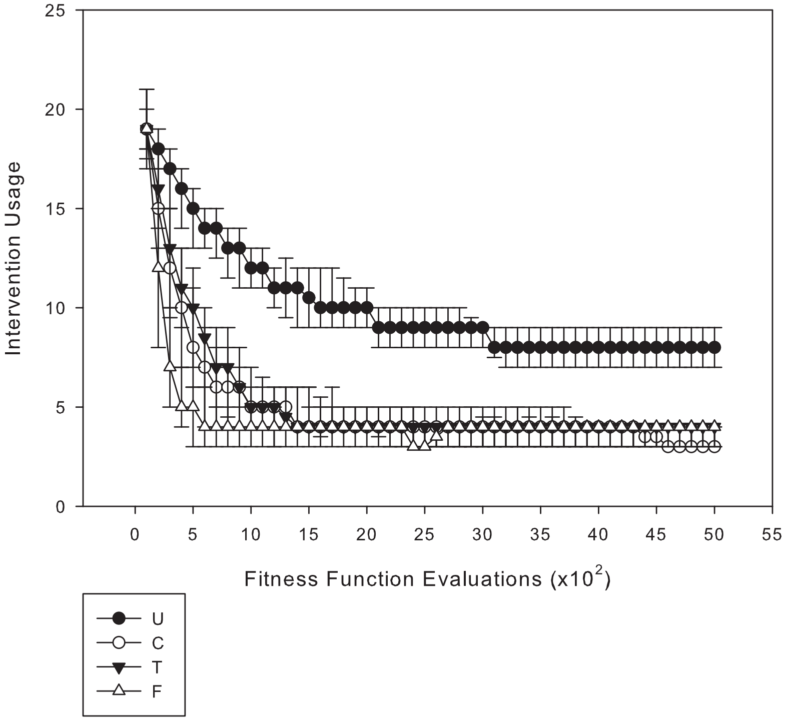


Figure 9.27: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 20$ and $R = ss$





9.2.4 35 penalty points per intervention

The previous sections have detailed the abilities of FDC, in comparison with CalEB, TInSSel and UC, over a range of intervention penalty values. This section reviews the effect on FDC of further increasing the intervention penalty, from 20 to 35 penalty points per intervention.

Figures 9.28 to 9.30 show the fitness scores and associated intervention usage for p_m levels of 0, 0.005 and 0.05 respectively.

Figure 9.28: Results for $N = 100, p_c = 0.9, p_m = 0$,$P = 35$ and $R = g$

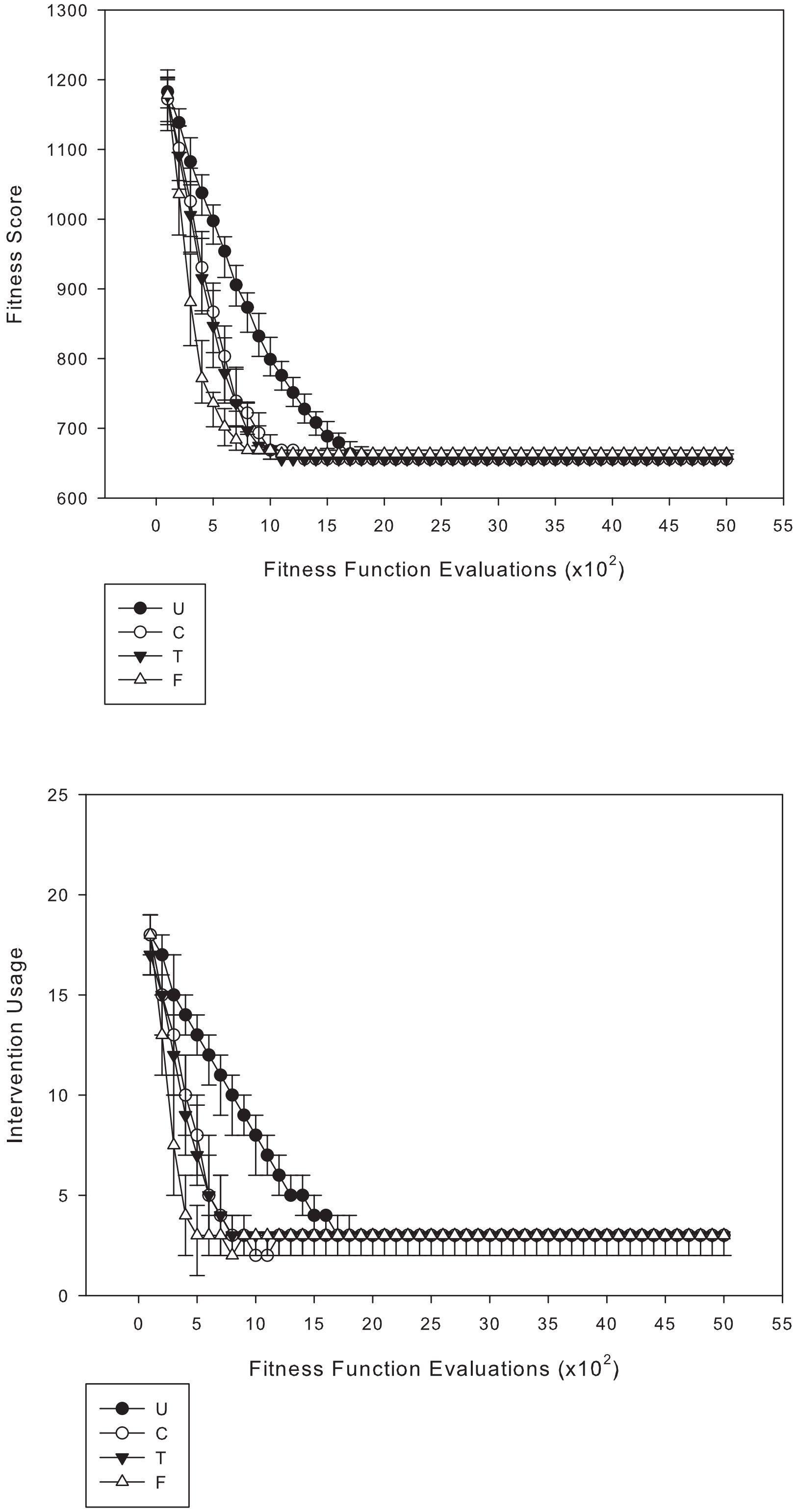
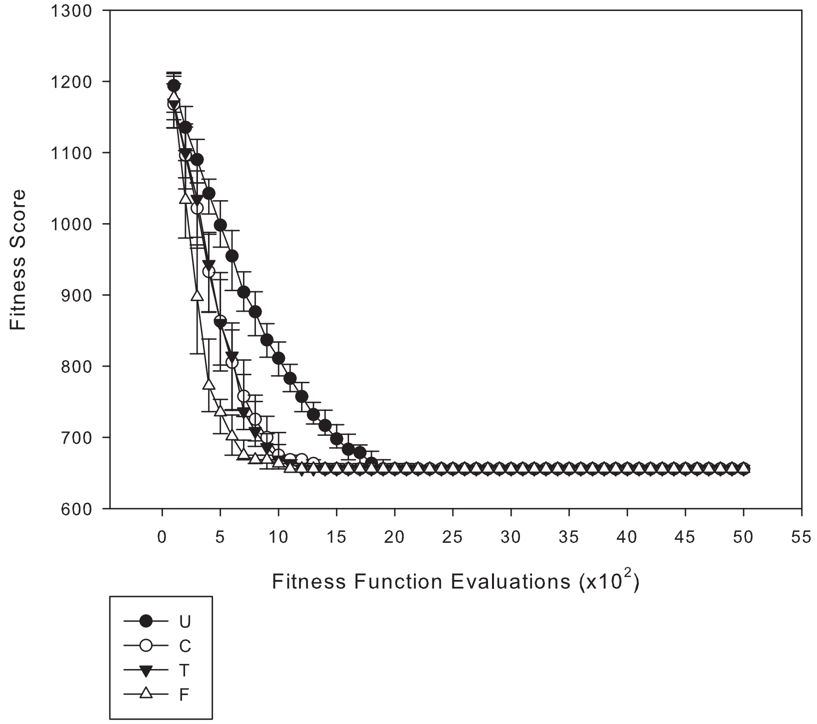


Figure 9.29: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 35$ and $R = g$



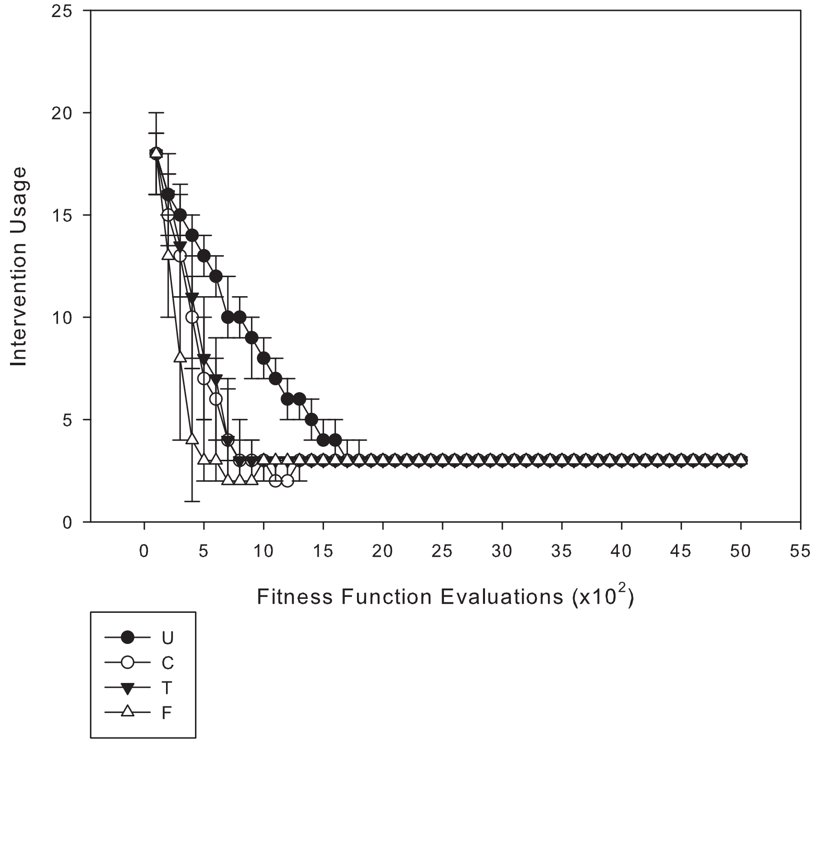
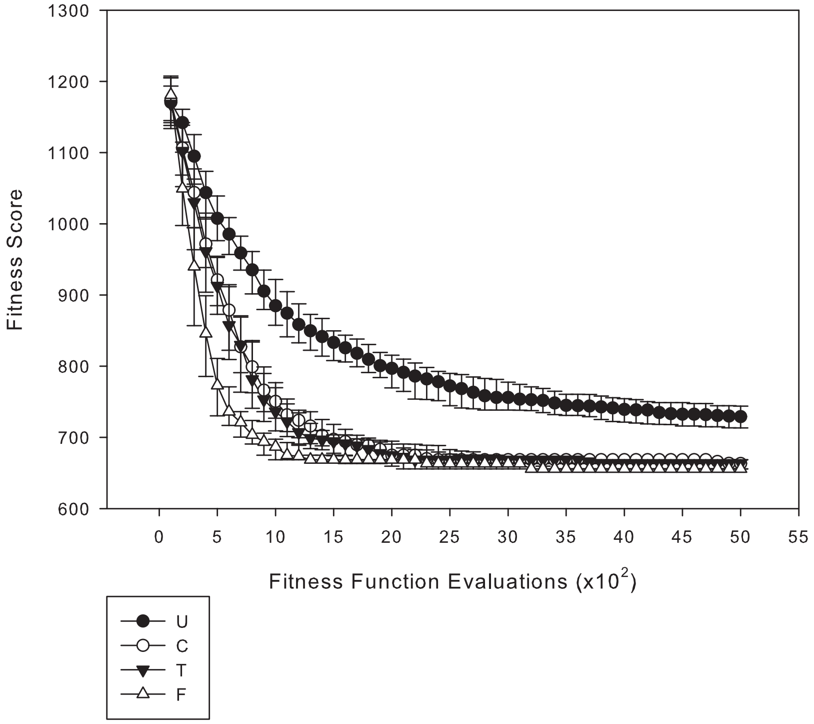


Figure 9.30: Results for $N = 100, p_c = 0.9, p_m = 0.05, P = 35$ and $R = g$



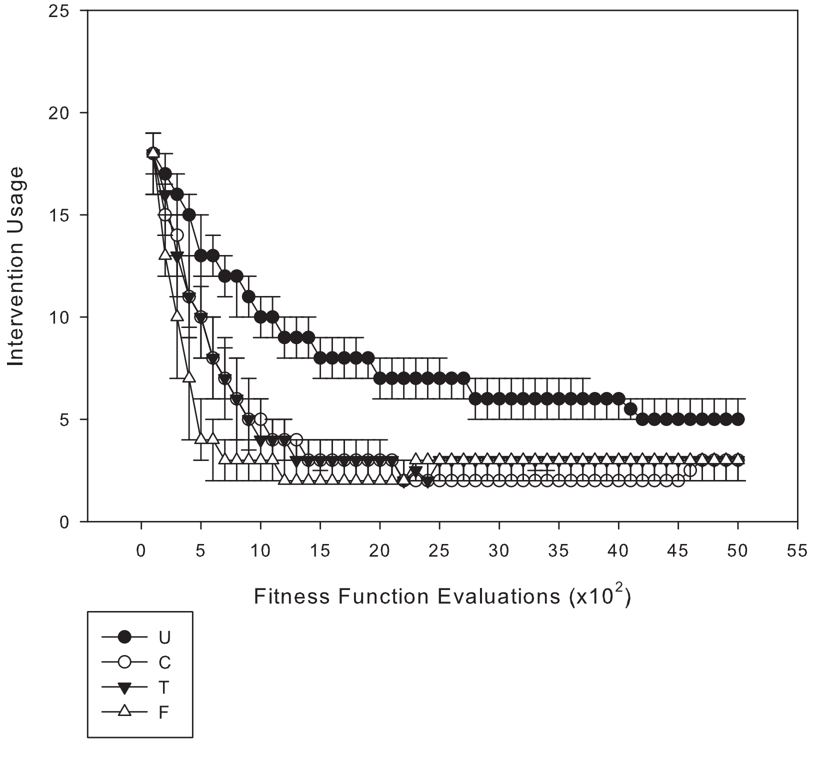


Figure 9.28 shows that in the absence of mutation, FDC is quicker than the other approaches at reducing intervention levels. Due to this, FDC is also quicker at producing fit scores, but, as with lesser intervention

penalties, FDC appears to return less fit scores than the other approaches later in the run.

Figure 9.29 shows the effect of increasing p_m to 0.005. For this mutation level, FDC is again quicker at reducing intervention levels than the other approaches and returns fitter scores for approximately the first 1,000 FFEs. Unlike the case with no mutation, FDC appears to produce scores of a similar fitness for the rest of the experiment.

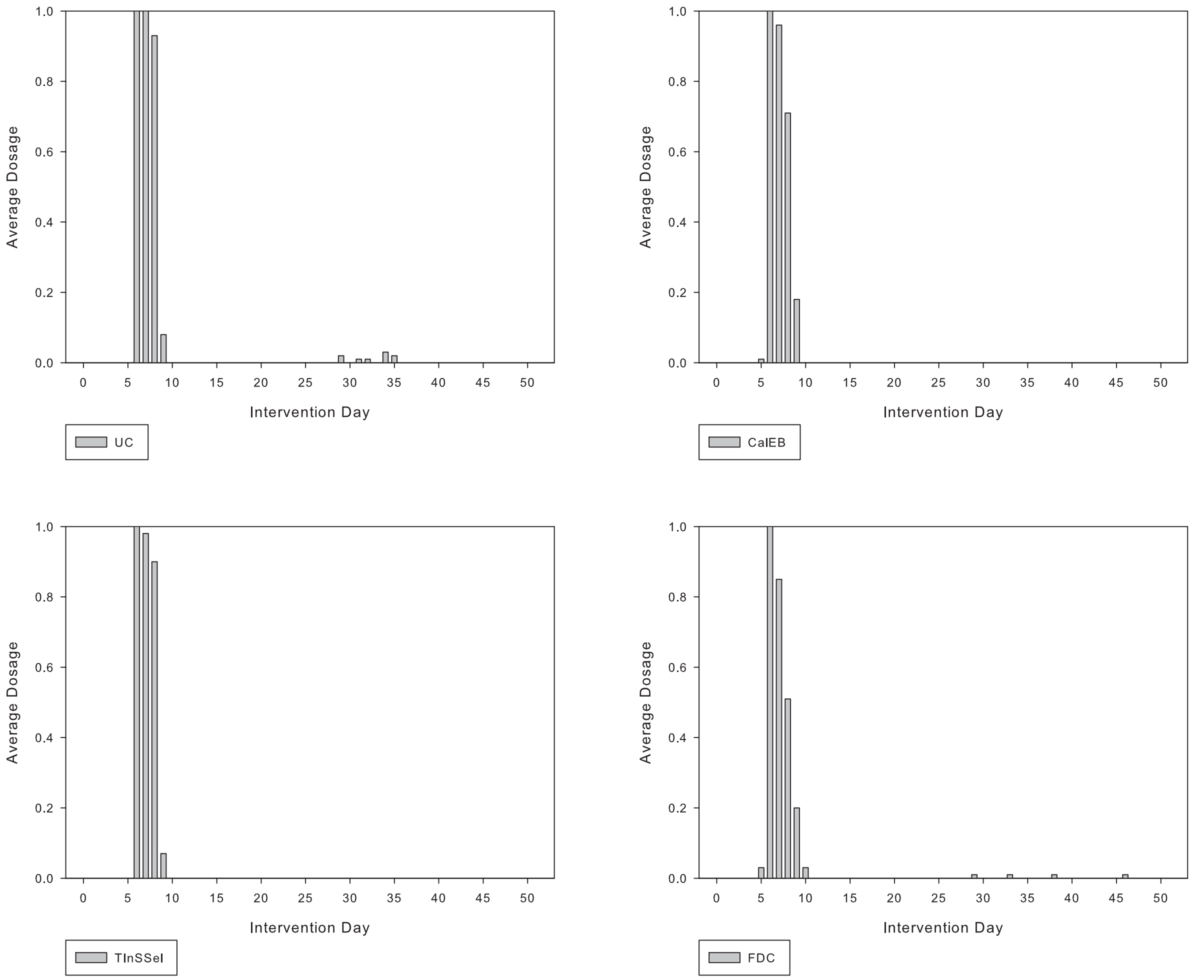
The effect of further increasing p_m to 0.05 is shown in Figure 9.30. For this mutation level, as with the lesser mutation rates, FDC is quicker at reducing intervention levels than the other approaches. FDC settles on approximately the same number of interventions as TInSSel and returns scores at least as good as the other approaches for this mutation level.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the FDC crossover approaches, in comparison to CalEB, TInSSel and UC, under a generational replacement strategy for a penalty of 35 points per intervention.

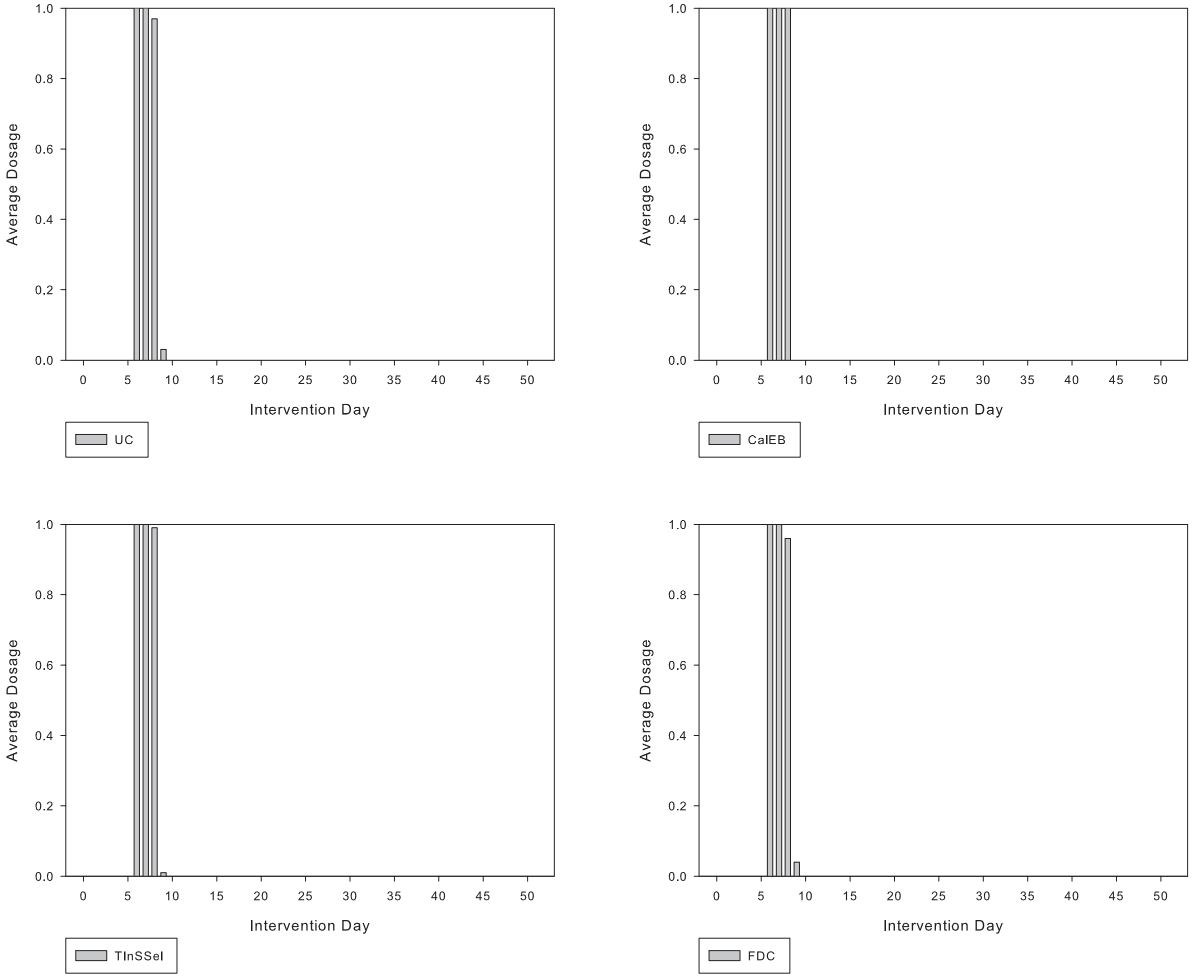
In the case of no mutation, as shown in Figure 9.28, each of the crossover approaches are shown to converge to a similar number of interventions. From the statistical analysis undertaken in Appendix E.25, after 5,000 FFEs, FDC is shown to be outperformed by TInSSel, CalEB and UC. The average placement of interventions are shown in Figure 9.31. This shows that FDC does not target days 6 - 8 as often as the other crossover approaches.

Figure 9.31: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 35$ and $R = g$



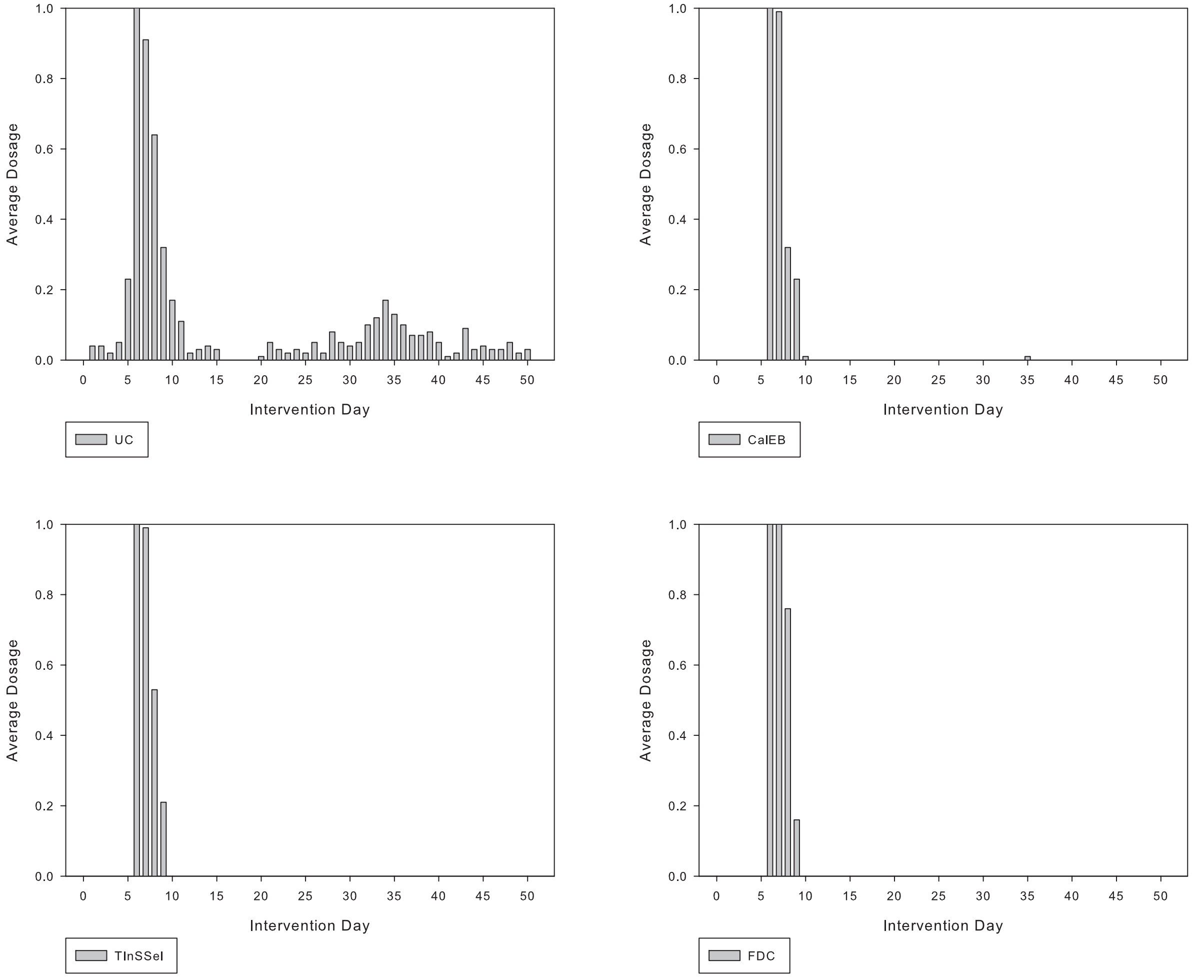
When the mutation rate is increased to 0.005, as shown in Figure 9.29, each of the crossover approaches are shown to converge to a similar number of interventions. From the statistical analysis undertaken in Appendix E.26, after 5,000 FFEs, no statistical difference was shown between any of the crossover approaches. The average placement of interventions are shown in Figure 9.32. This shows that each of the crossover approaches target the same interventions, days 6 - 8.

Figure 9.32: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 35$ and $R = g$



When the mutation rate is further increased to 0.05, as shown in Figure 9.30, UC is shown to use more interventions than each of the directed crossover approaches. From the statistical analysis undertaken in Appendix E.27, after 5,000 FFEs, FDC was shown to statistically outperform UC, CalEB and TInSSel. The average placement of interventions are shown in Figure 9.33. This shows that UC is less focussed on days 6 and 7 than FDC, and also places interventions on other days of the schedule where FDC does not. Both CalEB and TInSSel target days 6 and 7 in a similar way to FDC but place less emphasis on interventions on day 8.

Figure 9.33: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 35$ and $R = g$



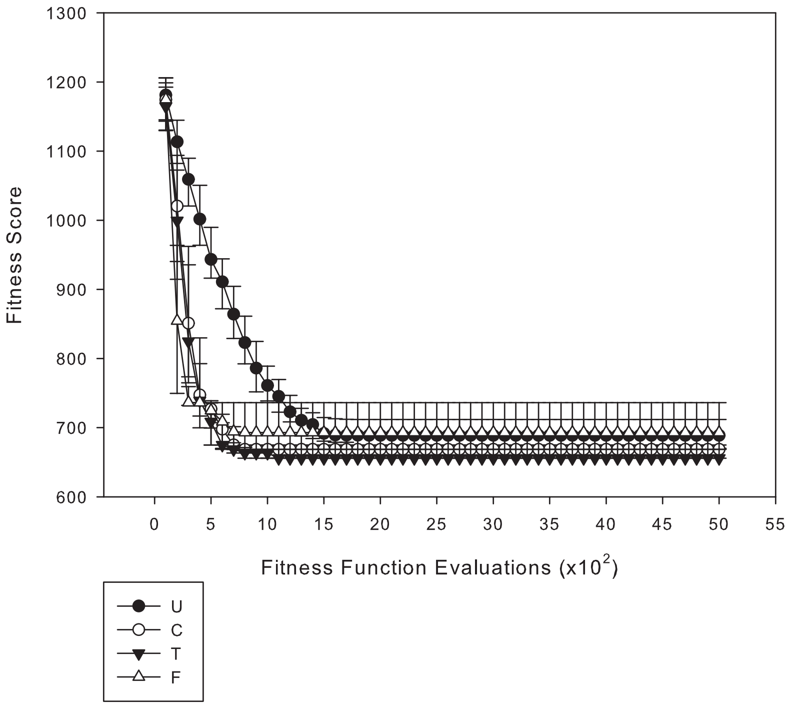
Steady state replacement

The previous experiment is now repeated for a steady state replacement strategy. Figures 9.34 to 9.36 show the fitness scores and intervention usage for intermediate parameter settings, a penalty of 35 points per intervention and a steady state replacement strategy.

Figure 9.34 shows that for steady state replacement, with no mutation, FDC settles to 2 interventions as does CalEB. In line with the steady state experiment with 20 penalty points per intervention, for a penalty of 35 points per intervention FDC produces slightly poorer scores than with the generational equivalent.

For a steady state replacement strategy and mutation levels of 0.005 (Figure 9.35) and 0.05 (Figure 9.36), FDC displays similar trends to those demonstrated with generational replacement, both in terms of fitness scores returned and intervention usage. This was also the case for a penalty of 20 points per intervention.

Figure 9.34: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$,$P = 35$ and $R = ss$



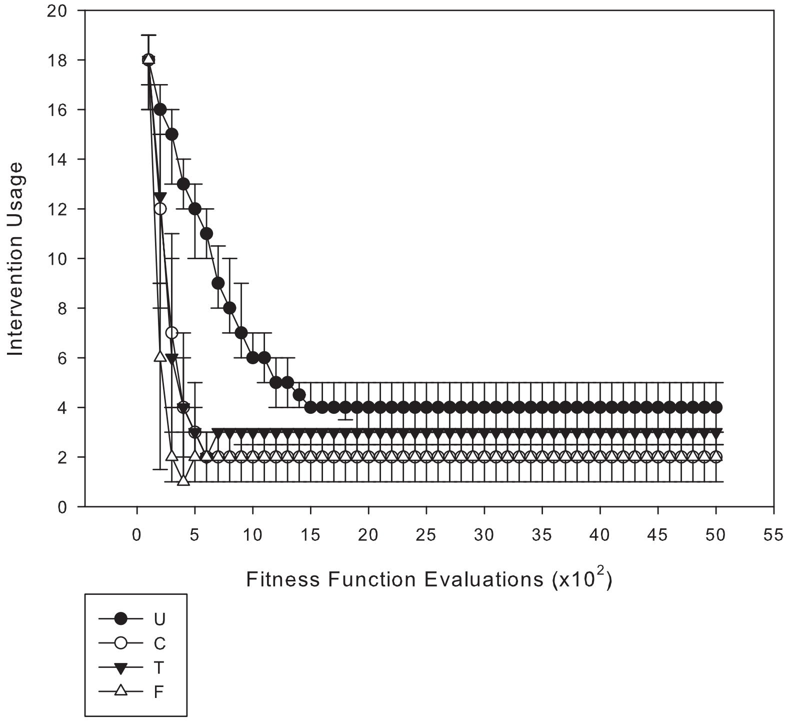


Figure 9.35: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 35$ and $R = ss$

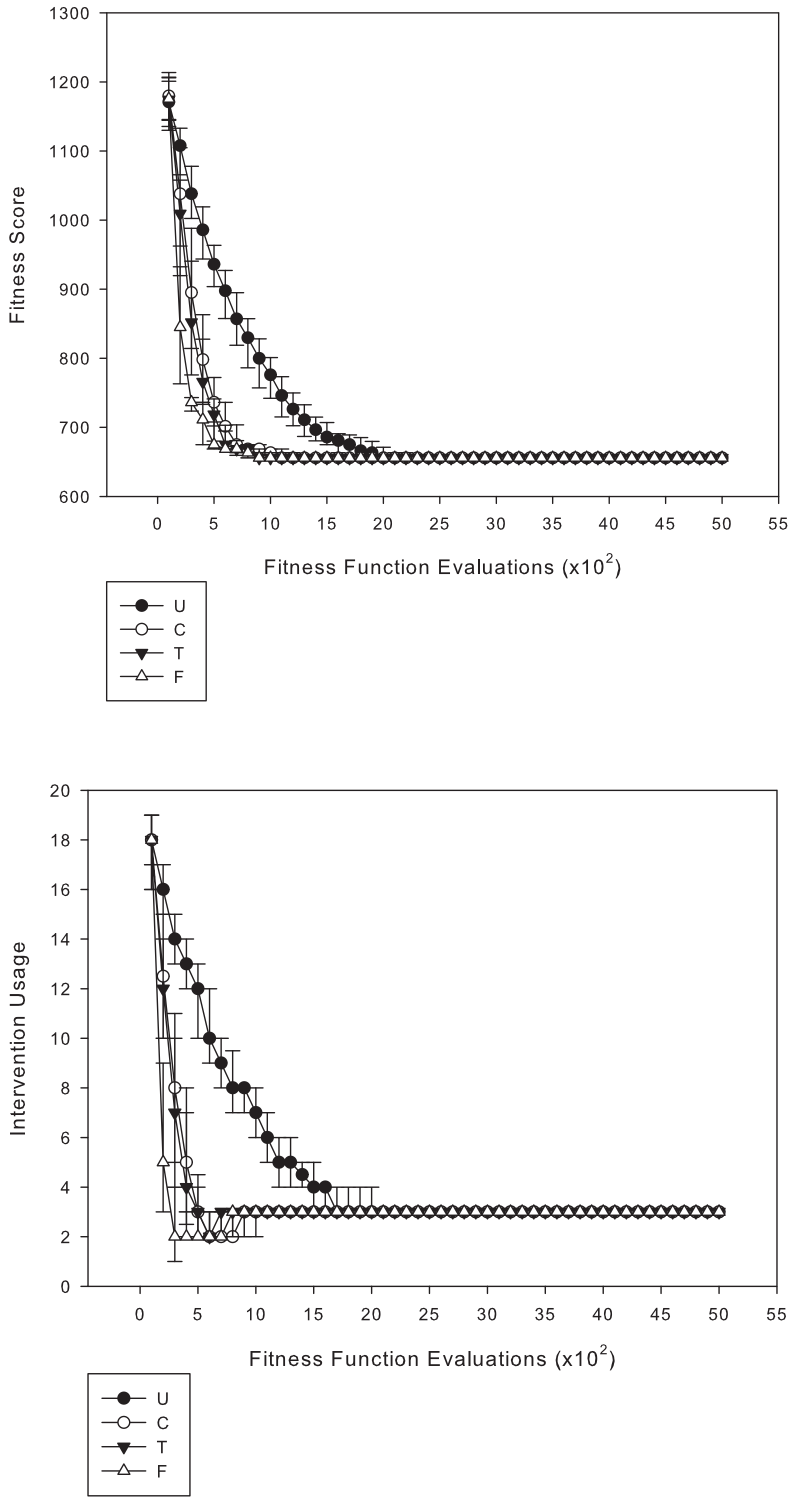
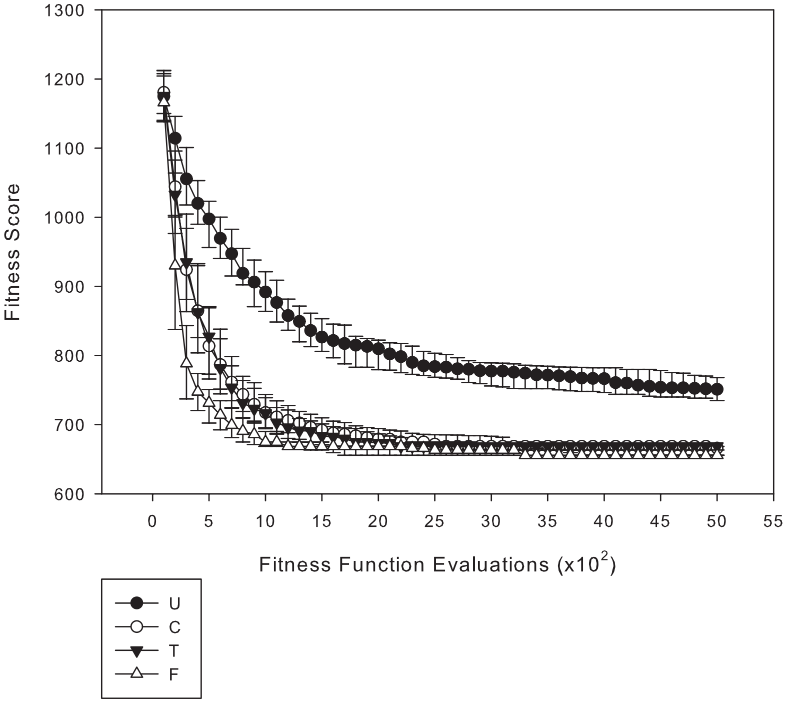
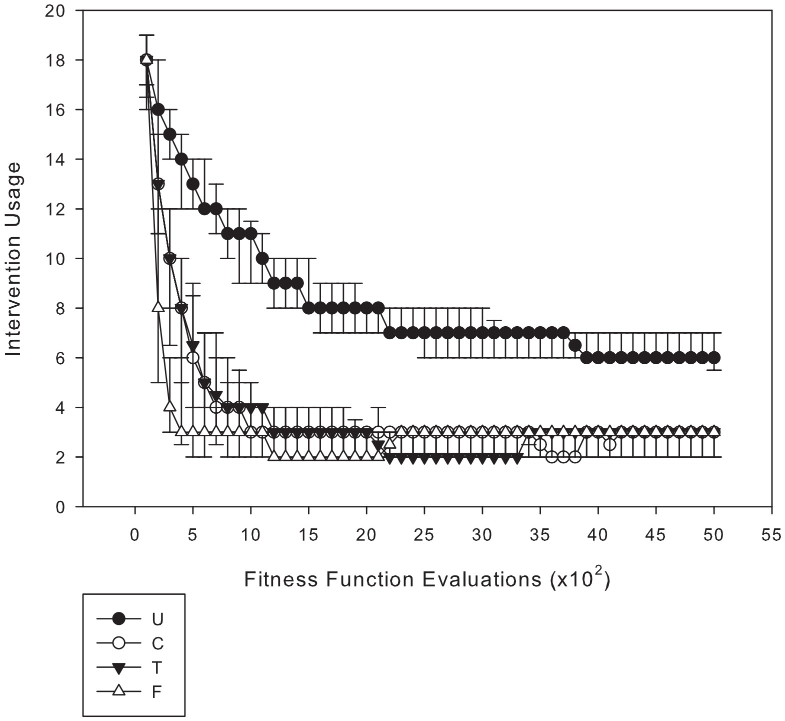


Figure 9.36: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 35$ and $R = ss$

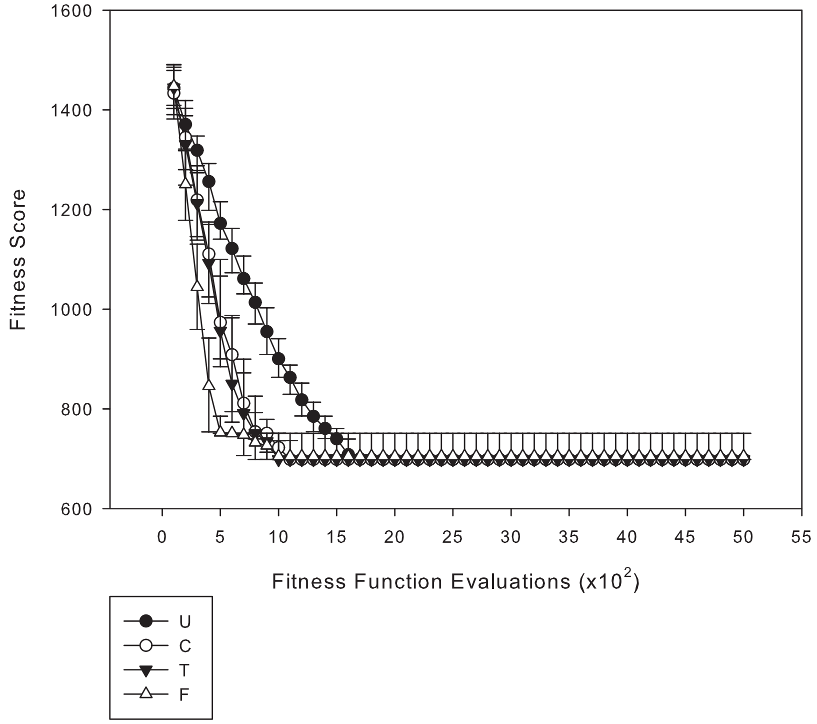




9.2.5 50 penalty points per intervention

The final bio-control experiment describes how FDC compares to CalEB, TInSSel and UC with a penalty of 50 points per intervention. Figures 9.37 to 9.39 show the fitness scores and associated intervention usage for intermediate parameter settings, a penalty of 50 points per intervention and p_m levels of 0, 0.005 and 0.05 respectively.

Figure 9.37: Results for $N = 100, p_c = 0.9, p_m = 0,P = 50$ and $R = g$



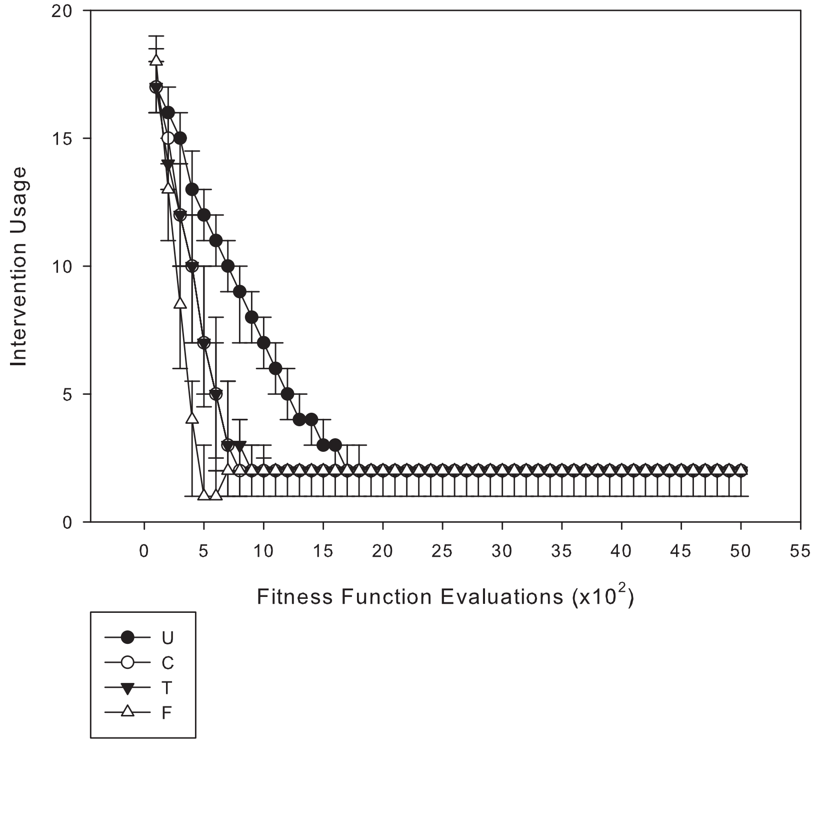


Figure 9.38: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 50$ and $R = g$

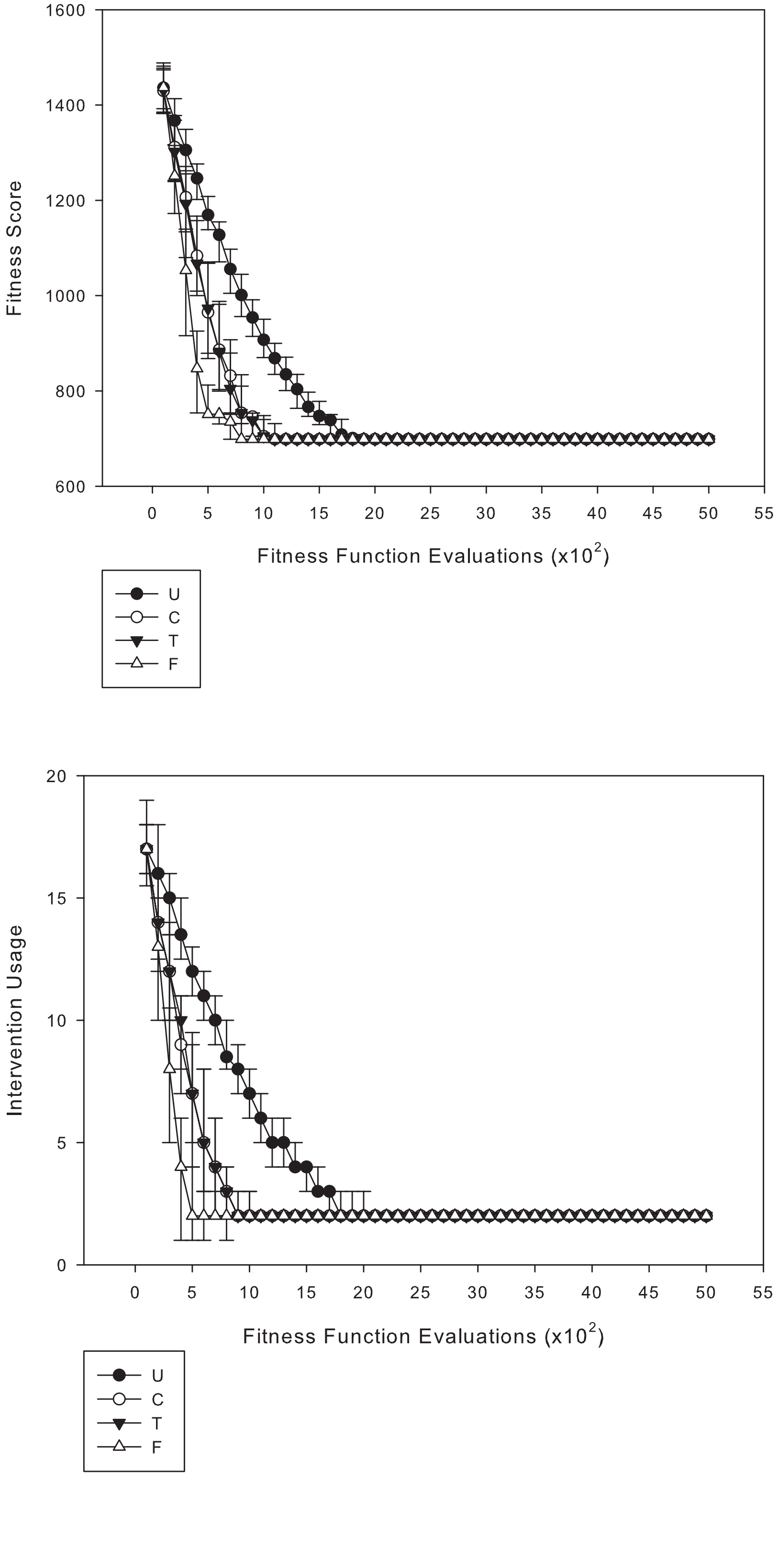


Figure 9.39: Results for $N = 100, p_c = 0.9, p_m = 0.05, P = 50$ and $R = g$

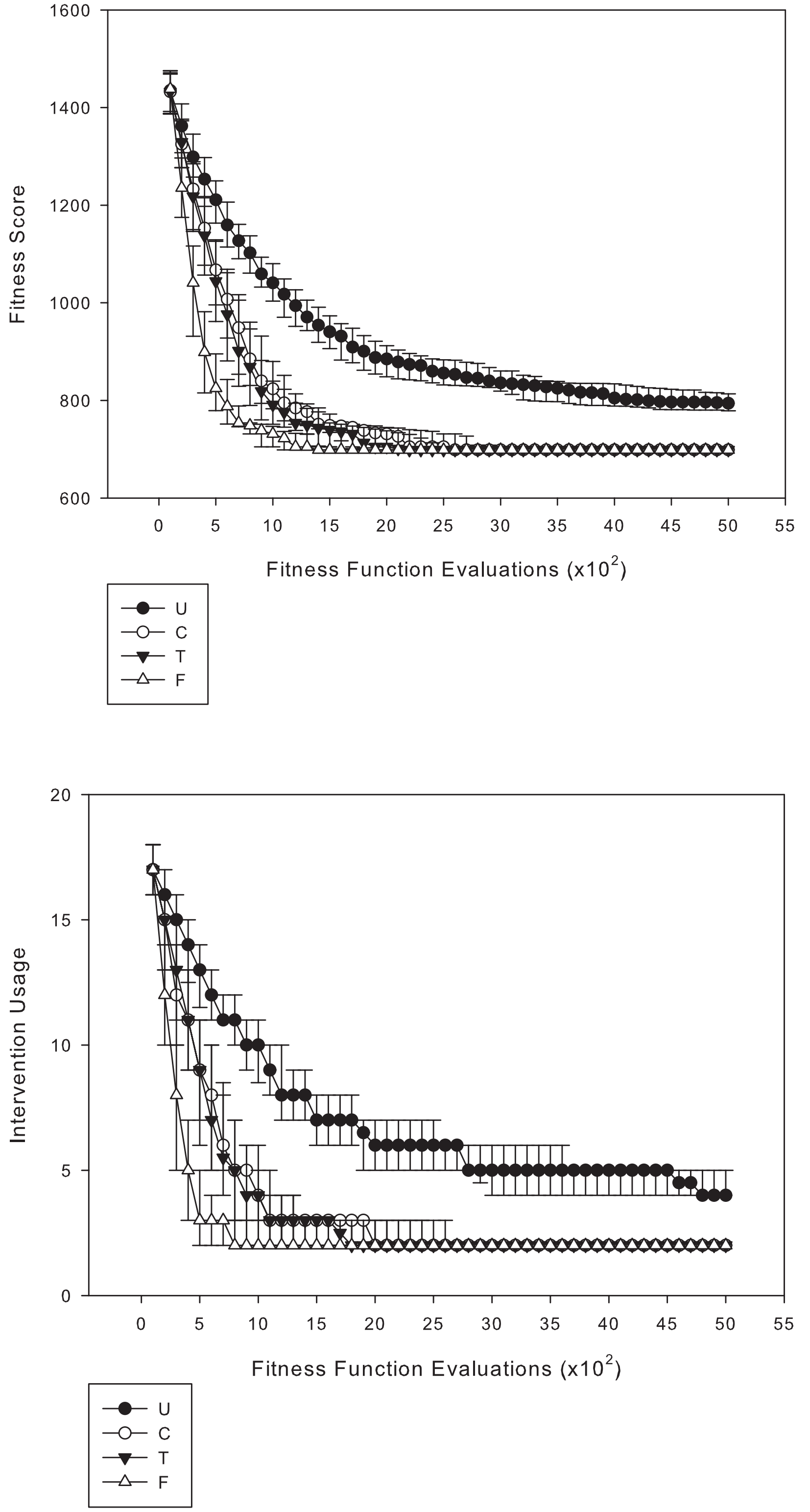


Figure 9.37 shows the approaches when p_m is set to 0. For this mutation level, as with lesser intervention penalties, FDC is once again quicker to return fit scores than the other approaches. Also in accordance with the

lesser intervention penalties, FDC appears to prematurely converge and settle to less fit scores than those returned by the other techniques.

The effect of increasing the p_m value to 0.005 is shown in Figure 9.38. FDC is again quicker than the other approaches at returning fit solutions, and unlike the case with no mutation, FDC settles to scores of a similar fitness than those found by the other techniques.

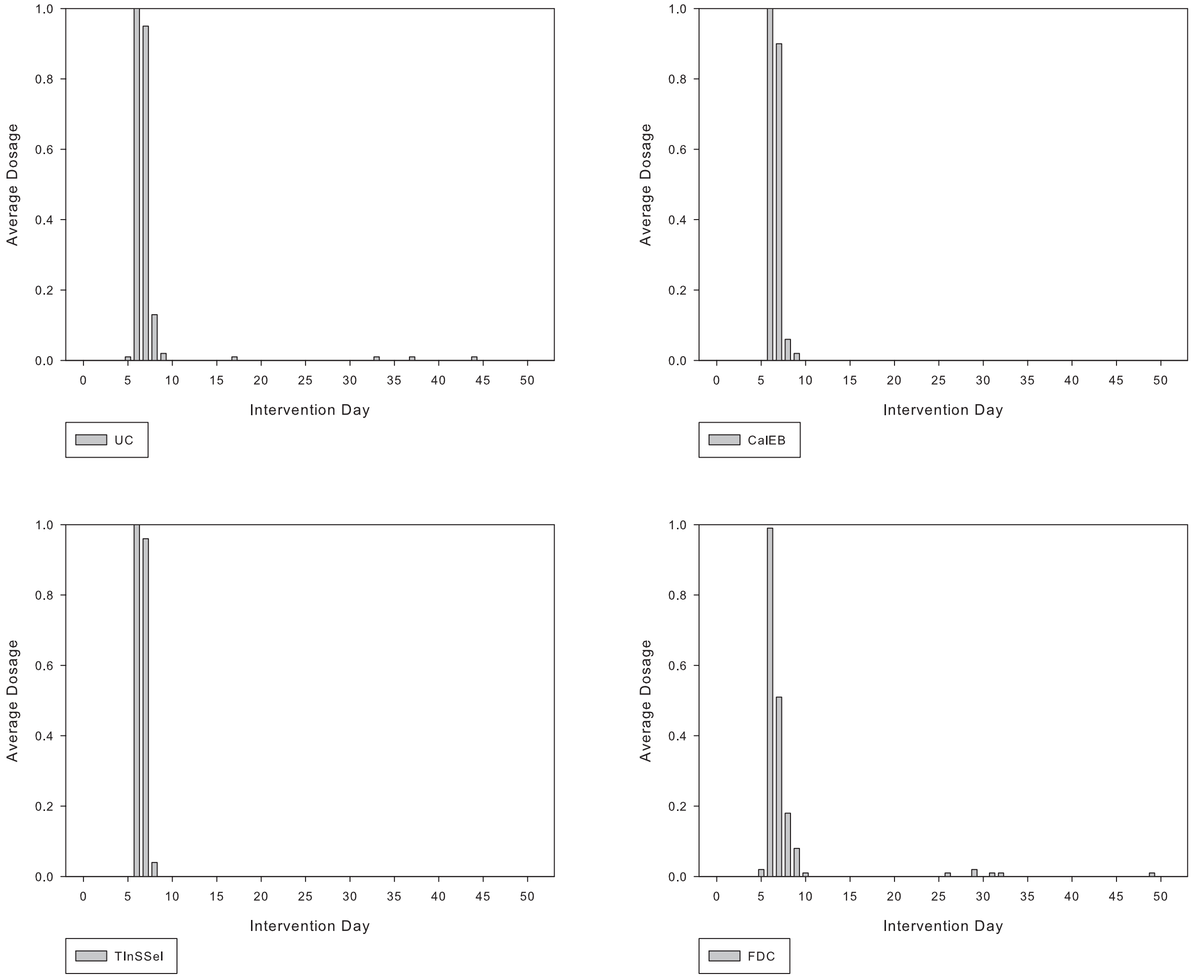
Figure 9.39 shows the effect of further increasing p_m from 0.005 to 0.05. For this increased mutation level, FDC returns fitter scores than the other directed approaches until approximately 2,500 FFEs, and returns fitter scores than UC for the duration of the experiment.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the FDC crossover approaches, in comparison to CalEB, TInSSel and UC, under a generational replacement strategy for a penalty of 50 points per intervention.

In the case of no mutation, as shown in Figure 9.37, each of the crossover approaches are shown to converge to a similar number of interventions. From the statistical analysis undertaken in Appendix E.28, after 5,000 FFEs, FDC is shown to be outperformed by TInSSel, CalEB and UC. The average placement of interventions are shown in Figure 9.40. This shows that FDC does not target day 7 as often as the other crossover approaches.

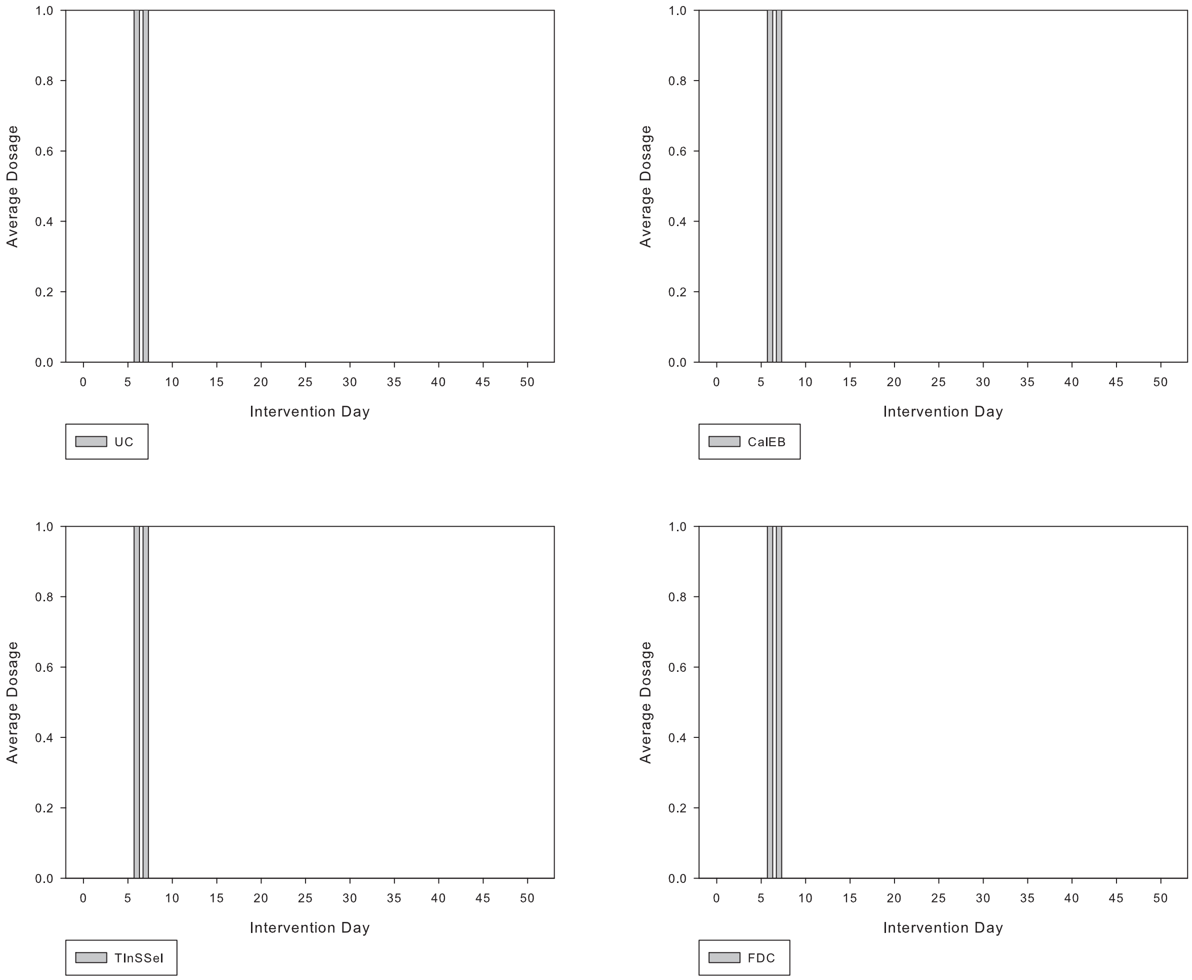
Figure 9.40: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$, $P = 50$ and $R = g$



When the mutation rate, p_m, is increased to 0.005, as shown in Figure 9.38, each of the crossover approaches are shown to converge to a similar number of interventions. From the statistical analysis undertaken in Appendix

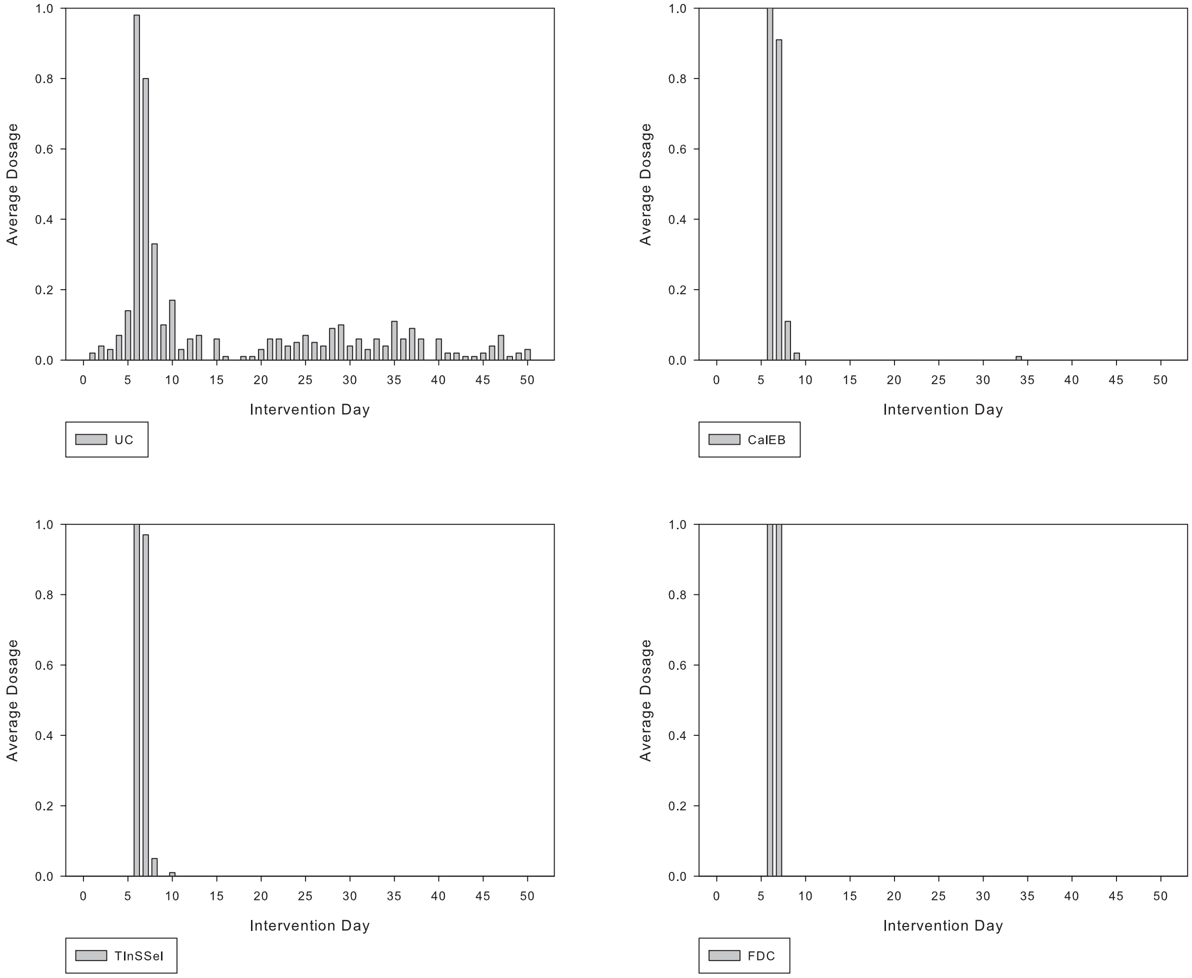
E.29, after 5,000 FFEs, there is shown to be no statistical difference between approaches. The average placement of interventions are shown in Figure 9.41. This shows that each of the crossover approaches target interventions exclusively on days 6 and 7 of the schedule.

Figure 9.41: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$, $P = 50$ and $R = g$



When the mutation rate is further increased to 0.05, as shown in Figure 9.39, UC is shown to use more interventions than the directed intervention approaches. From the statistical analysis undertaken in Appendix E.30, after 5,000 FFEs, there is shown to be no statistical difference between FDC, CalEB and TInSSel, with FDC shown to outperform UC. The average placement of interventions are shown in Figure 9.42. This shows that UC is less focussed on days 6 and 7 than FDC and that UC also targets many other days of the schedule, whereas FDC focusses solely on days 6 and 7.

Figure 9.42: Intervention placement for 5,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$, $P = 50$ and $R = g$



Steady state replacement

The same experiment will now be reviewed for a steady state replacement strategy. Figures 9.43 to 9.45 show the fitness scores and associated intervention usage graphs for each approach using the intermediate parameter settings, a penalty of 50 points per intervention and a steady state replacement strategy.

Figure 9.43 shows that in the absence of mutation, FDC settles on fewer interventions than under generational replacement. This leads to less fit scores being returned by FDC for steady state replacement for this mutation level. This was also the case for intervention penalties of 20 and 35 points per intervention.

When the mutation rate is increased to 0.005, as shown in Figure 9.44, FDC behaves in a similar way under steady state as for generational replacement. This is both in terms of fitness scores and intervention usage.

A further mutation increase to 0.05 is shown in Figure 9.45. As with a p_m of 0.005, FDC behaves in a similar way, in terms of both fitness scores and intervention usage, regardless of the replacement strategy used.

Figure 9.43: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$,$P = 50$ and $R = ss$

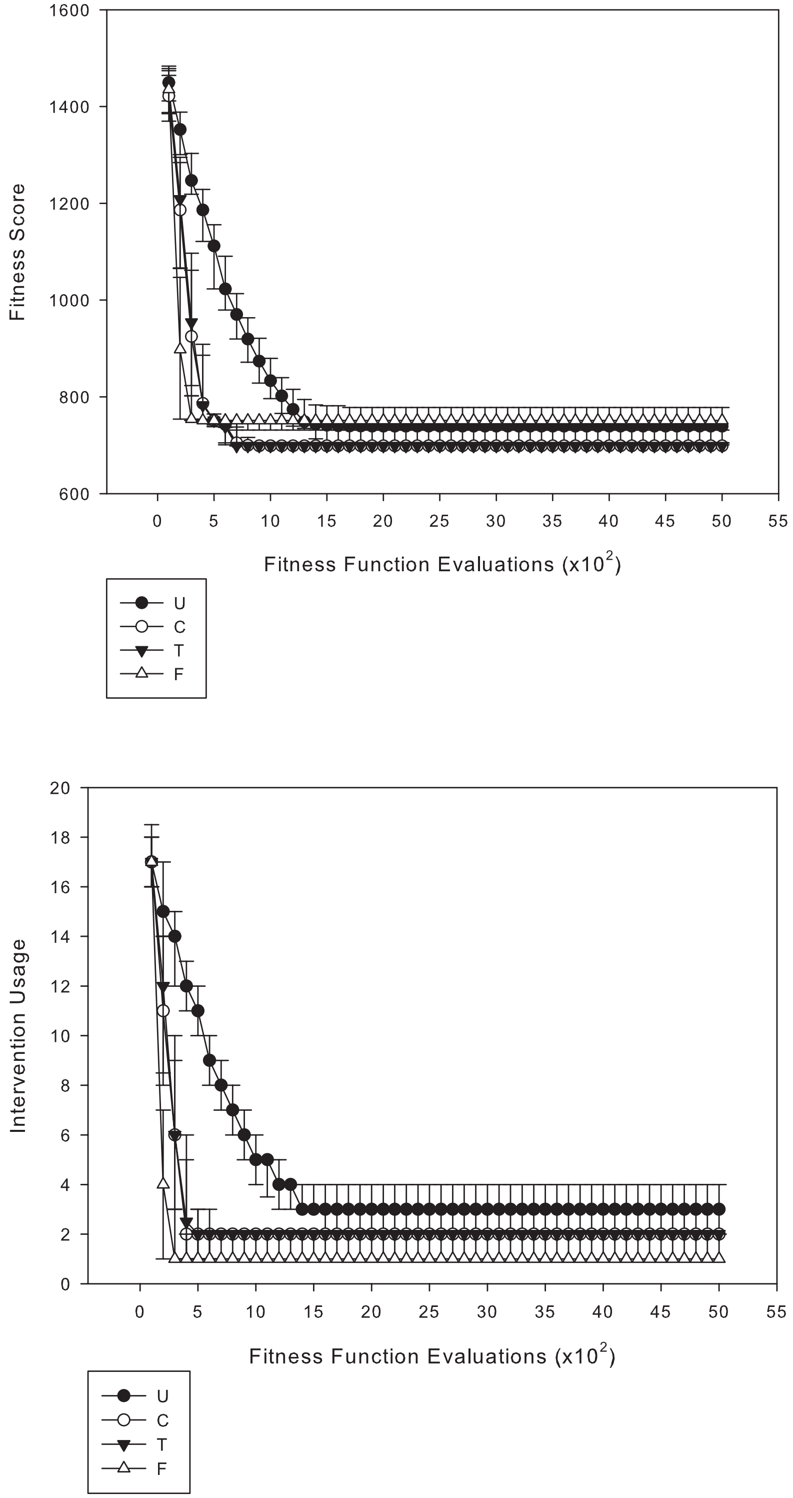
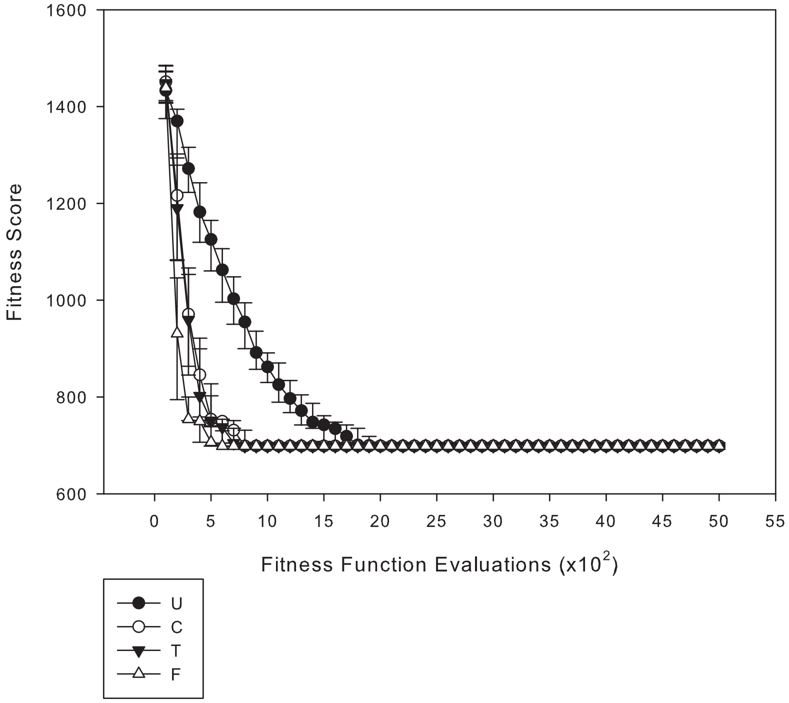


Figure 9.44: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$,$P = 50$ and $R = ss$



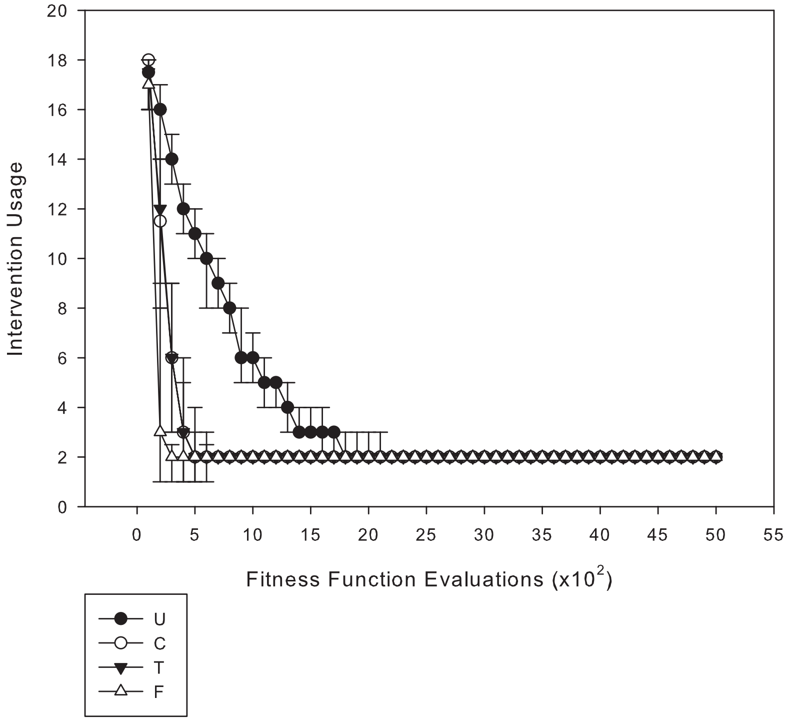
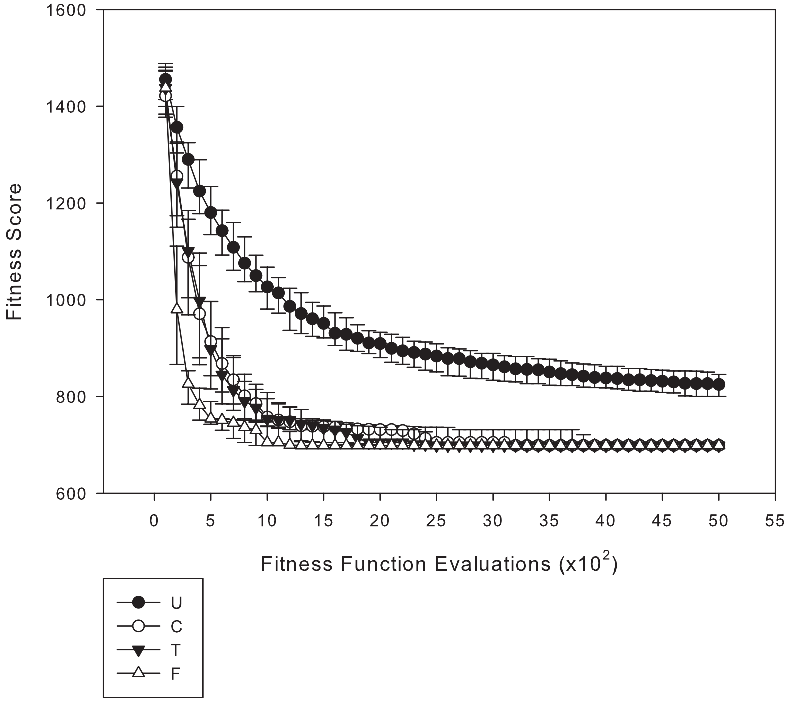
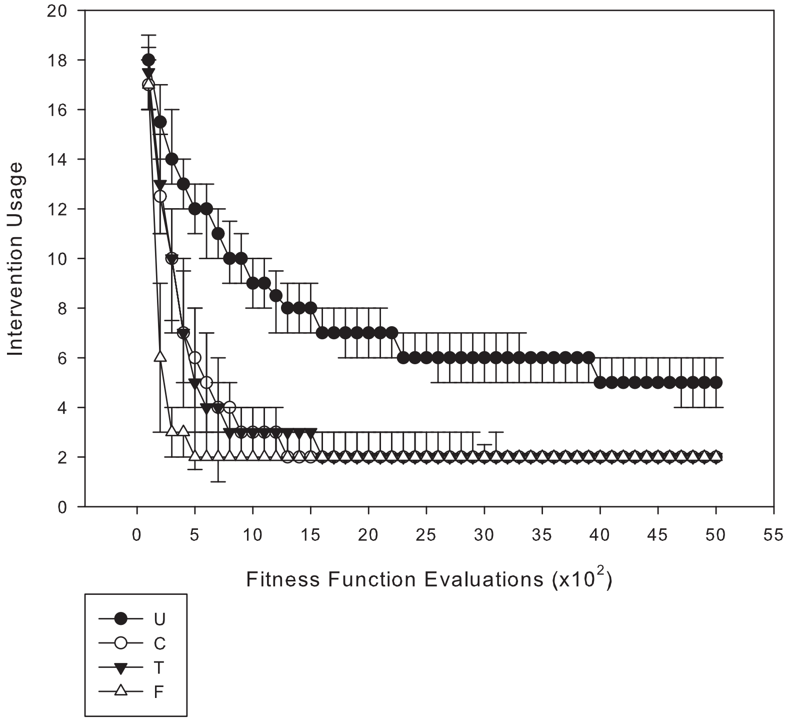


Figure 9.45: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.05$,$P = 50$ and $R = ss$





9.2.6 Summary of bio-control scheduling experiment

The previous experiments have reviewed FDC in comparison to UC, CalEB and TInSSel for a range of intervention penalties. This section draws together the salient points from each of these experiments.

In order to ascertain if there is any statistical significant differences between approaches for this problem, the Kruskal-Wallis test results for the intermediate test cases were calculated and are shown in Tables E.16 to E.30. For clarity, Appendix J.3 shows which crossover approaches are significantly different over the range of FFEs.

Each of the intervention penalties will be reviewed in turn, with Section 9.2.7 describing the trends demonstrated across the range of penalty values.

Penalty of 0 points per intervention

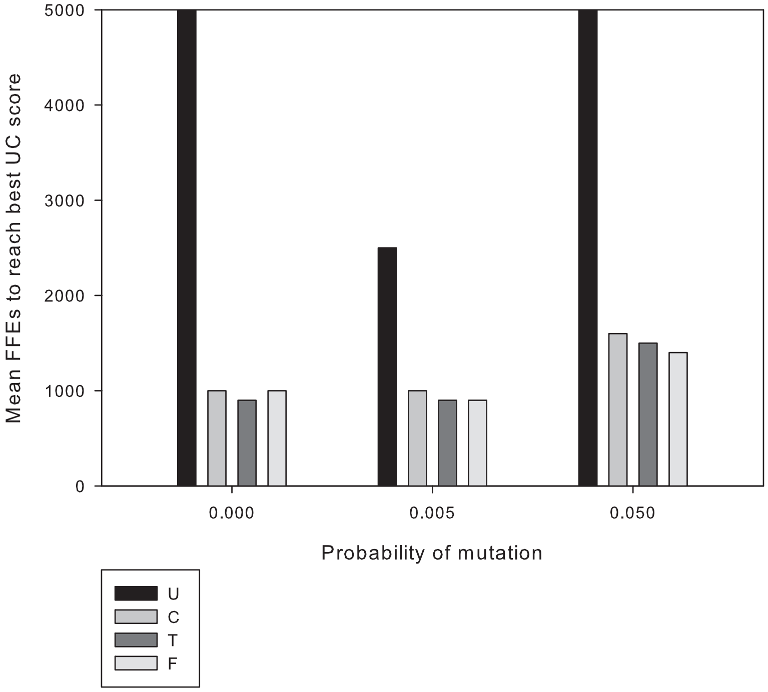


Figure 9.46: FFEs to reach the best score found by UC (355) with a penalty of 0 points per intervention

Section 9.2.1 described the scores found by FDC, in comparison to UC, CalEB and TInSSel for a penalty of 0 points per intervention. This showed that FDC, as with CalEB and TInSSel, is faster to increase intervention levels than UC, resulting in fitter scores being returned by these directed intervention approaches.

The statistical analysis detailed in Appendix E.2.1 shows that there is a statistical difference between at least two of the crossover approaches from 200 to 5,000 FFEs for each of the mutation levels. For clarity, the crossover approaches which are statistically different to each other are described in Appendix J.3.1.

This shows that regardless of mutation level, FDC outperforms UC for almost all FFEs under review. FDC does appear to prematurely converge in the absence of mutation though as FDC outperforms CalEB at 200 and 400 FFEs but is statistically outperformed by CalEB from 1,800 FFEs onwards. Although FDC is comparable to TInSSel for the first 1,500 FFEs, due to the premature convergence of FDC, TInSSel outperforms FDC for the remaining FFEs, 1,600 to 5,000.

When the mutation rate is increased to 0.005, FDC outperforms CalEB from 300 to 500 FFEs and is only outperformed by TInSSel at 2,400 FFEs. A further increase in mutation rate to 0.05 shows that while TInSSel and FDC return similar scores, FDC outperforms CalEB from 300 to 500 FFEs.

As described in Section 8.1.6, over the range of mutation levels, the best score found by UC for a penalty of 0 points per intervention was 355. This was found when the mutation rate was 0.005. Figure 9.46 shows the FFEs required by each of the crossover approaches to find this best score returned by UC, over the range of mutation levels. This shows that FDC uses the same number of FFEs for CalEB in the absence of mutation, which is 100 more FFEs than TInSSel. When the mutation rate is increased to 0.005, FDC uses the same number of FFEs as TInSSel, and for a further mutation increase to 0.05, FDC uses 100 less FFEs than TInSSel.

Penalty of 5 points per intervention

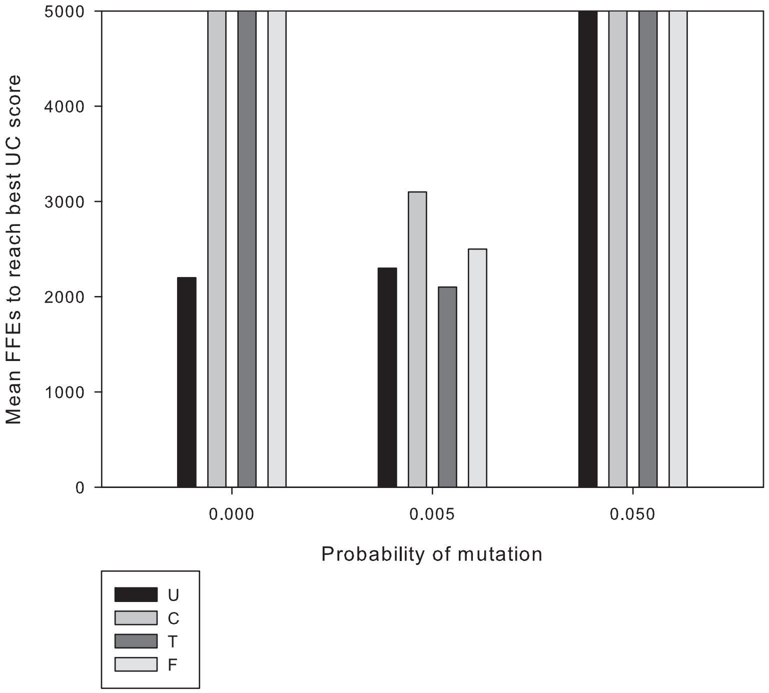


Figure 9.47: FFEs to reach the best score found by UC (476) with a penalty of 5 points per intervention

Section 9.2.2 described the scores found by FDC, in comparison to UC, CalEB and TInSSel for a penalty of 5 points per intervention. This showed that FDC, as with CalEB and TInSSel, were faster to reduce in intervention levels than UC, resulting in fitter scores being returned by these directed intervention approaches.

The statistical analysis detailed in Appendix E.2.2 shows that there is a statistical difference between at least two of the crossover approaches from 200 to 5,000 FFEs for each of the mutation levels. The crossover approaches which are statistically different to each other are described in Appendix J.3.2.

FDC was shown to prematurely converge in the absence of mutation for a penalty of 0 points per intervention. This is also the case with an intervention penalty of 5 points per intervention. Although initially better than UC, from 700 to 1,700 FFEs, from 2,100 FFEs onward UC outperforms FDC. TInSSel and FDC are similar for the first 2,000 FFEs but from this point onwards, TInSSel statistically outperforms FDC. When the mutation rate is increased to 0.005, FDC outperforms UC from 600 to 1,800 FFEs, and is only outperformed by TInSSel from 2,200 to 2,500 FFEs. A further mutation increase to 0.05 shows a benefit in using FDC. FDC outperforms UC for most of the FFEs and also outperforms TInSSel from 1,000 to 2,400 FFEs. CalEB was previously shown to be poor for this experiment and due to this FDC outperforms CalEB for at least half of all FFEs regardless of mutation rate

As described in Section 8.1.6, over the range of mutation levels, the best score found by UC for a penalty of 5 points per intervention was 476. This was found when the mutation rate was 0.005. Figure 9.47 shows the FFEs required by each of the crossover approaches to find this best score returned by UC, over the range of mutation levels. This shows that FDC, as with the other directed approaches, do not find this result in the absence of mutation. When p_m is increased to 0.005, FDC requires slightly more FFEs than both UC and TInSSel to find this value, but requires fewer than CalEB. None of the approaches find a score of 476 when the mutation rate is increased to 0.05.

Penalty of 20 points per intervention

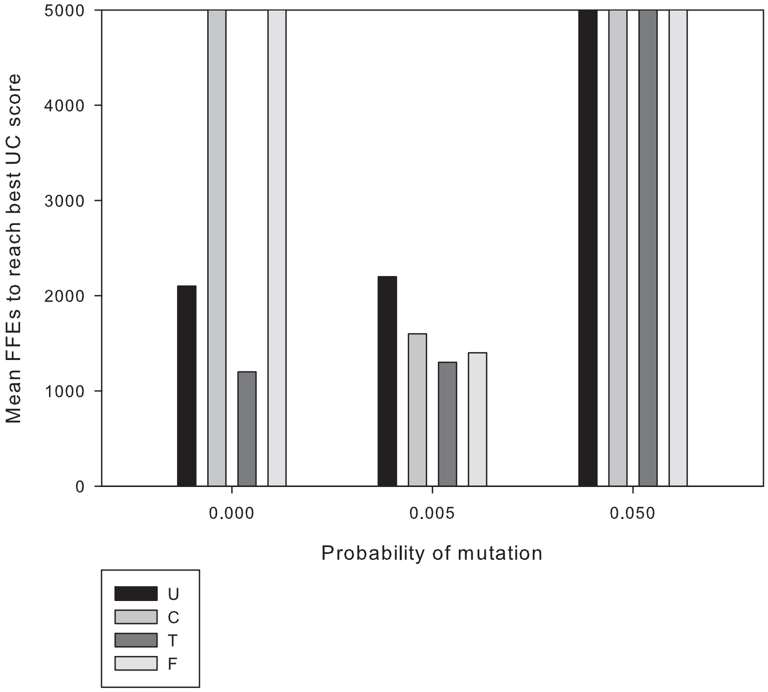


Figure 9.48: FFEs to reach the best score found by UC (603) with a penalty of 20 points per intervention

Section 9.2.3 described the scores found by FDC, in comparison to UC, CalEB and TInSSel for a penalty of 20 points per intervention. This showed that FDC, as with CalEB and TInSSel, is faster to reduce in intervention levels than UC, resulting in fitter scores being returned by these directed intervention approaches.

The statistical analysis detailed in Appendix E.2.3 shows that there is a statistical difference between at least two of the crossover approaches from 200 to 5,000 FFEs for each of the mutation levels. The crossover approaches which are statistically different to each other are described in Appendix J.3.3.

FDC was shown to prematurely converge in the absence of mutation for previous intervention penalties. This is also the case with an intervention penalty of 20 points per intervention. Although FDC outperforms UC from 700 to 1,700 FFEs, it is subsequently outperformed from 2,100 to 5,000 FFEs. TInSSel also outperforms FDC for the latter half of the FFEs in the absence of mutation. FDC outperforms CalEB for the first quarter of FFEs.

When the mutation rate is increased to 0.005, the ability of FDC improves. For this mutation level FDC outperforms each of the other crossover approaches for at least some of the initial FFEs. FDC outperforms UC from 200 to 2,400 FFEs, CalEB from 200 to 1,400 FFEs and TInSSel from 300 to 900 FFEs. A further increase in mutation rate sees FDC outperform all of the other crossover approaches for almost every FFE point.

As described in Section 8.1.6, over the range of mutation levels, the best score found by UC for a penalty of 20 points per intervention was 603. This was found when the mutation rate was 0.005. Figure 9.48 shows the FFEs required by each of the crossover approaches to find this best score returned by UC, over the range of mutation levels. This shows that in the absence of mutation, FDC does not find this target. When the mutation rate is increased to 0.005, FDC takes 100 more FFEs to reach the target than TInSSel. No crossover approach finds a score of 603 when the mutation rate is further increased to 0.05.

Penalty of 35 points per intervention

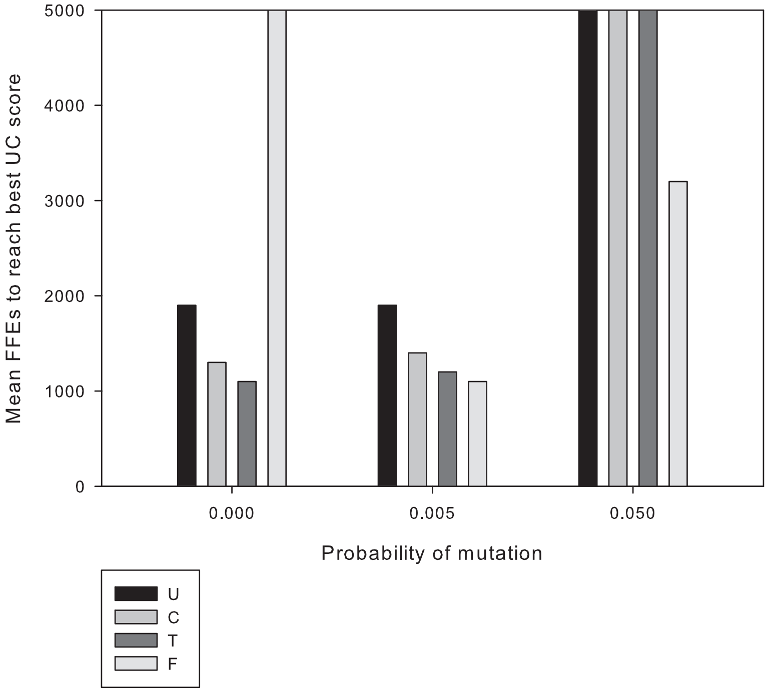


Figure 9.49: FFEs to reach the best score found by UC (655) with a penalty of 35 points per intervention

Section 9.2.4 described the scores found by FDC, in comparison to UC, CalEB and TInSSel for a penalty of 35 points per intervention. This showed that FDC was faster than the other directed intervention approaches to reduce the number of interventions in schedules resulting in fitter scores being returned by FDC for the initial FFEs.

The statistical analysis detailed in Appendix E.2.4 shows that there is a statistical difference between at least two of the crossover approaches from 200 to 5,000 FFEs for mutation levels of 0 and 0.05. For a mutation rate of 0.005 there exists a difference between approaches from 200 to 2,200 FFEs.

As with previous experiments, in the absence of mutation FDC is shown to prematurely converge. FDC outperforms the other approaches for the first fifth of the FFEs but is later outperformed by each of the other crossover approaches.

When the mutation rate is increased to 0.005, the ability of FDC improves. For this mutation level FDC outperforms each of the other crossover approaches for at least some of the initial FFEs. FDC outperforms UC, CalEB and TInSSel from 200 to 2,000, 1,300 and 1,000 FFEs respectively. A further increase in mutation rate sees FDC outperform both UC and CalEB for almost every FFE and TInSSel for three quarters of the FFE range.

As described in Section 8.1.6, over the range of mutation levels, the best score found by UC for a penalty of 35 points per intervention was 655. This was found when the mutation rate was 0.005. Figure 9.49 shows the FFEs required by each of the crossover approaches to find this best score returned by UC, over the range of mutation levels. This shows that in the absence of mutation, FDC does not find this target. When the mutation rate is increased to 0.005, FDC is faster than both TInSSel and CalEB to find this value. For a further increase in the mutation rate, FDC is the only approach to discover a score of at least 655.

Penalty of 50 points per intervention

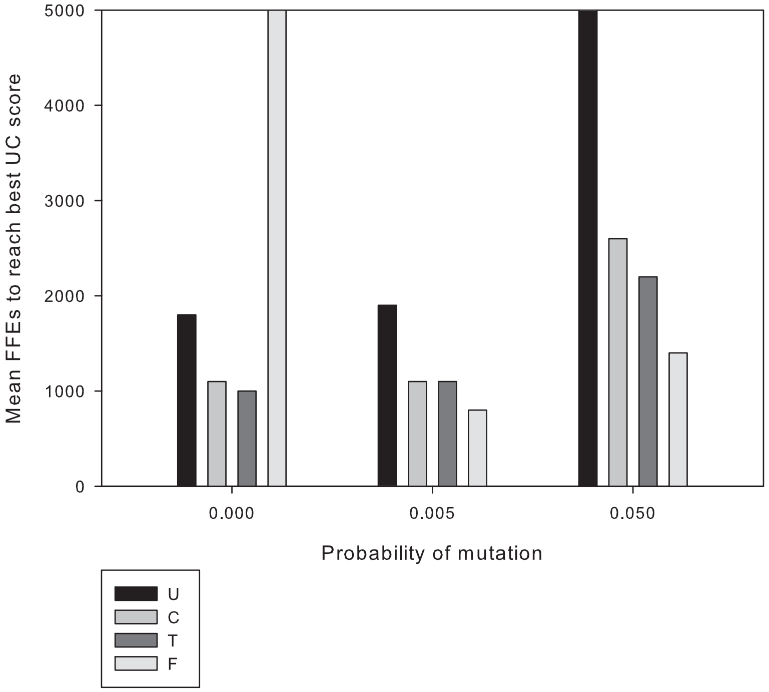


Figure 9.50: FFEs to reach the best score found by UC (698) with a penalty of 50 points per intervention

Section 9.2.5 described the scores found by FDC, in comparison to UC, CalEB and TInSSel for a penalty of 50 points per intervention. This showed that as with a penalty of 35 points per intervention, FDC was faster than the other directed intervention approaches to reduce the number of interventions in schedules resulting in fitter scores being returned by FDC for the initial FFEs.

The statistical analysis detailed in Appendix E.2.5 shows that there is a statistical difference between at least two of the crossover approaches from 200 to 5,000 FFEs for mutation levels of 0 and 0.05. For a mutation rate of 0.005 there is no statistical difference between approaches from 2,700 to 3,200 FFEs.

As with previous experiments, in the absence of mutation FDC is shown to prematurely converge. FDC again outperforms the other approaches for the initial FFEs but is later outperformed by each of the other crossover approaches.

When the mutation rate is increased to 0.005, the ability of FDC improves. For this mutation level FDC outperforms each of the other crossover approaches for at least some of the initial FFEs. FDC outperforms UC, CalEB and TInSSel from 200 to 2,000, 1,000 and 1,000 FFEs respectively. A further increase in mutation rate sees FDC outperform both UC for almost every FFE point and outperform CalEB and TInSSel for approximately half of the FFE range.

As described in Section 8.1.6, over the range of mutation levels, the best score found by UC for a penalty of 50 points per intervention was 698. This was found when the mutation rate was 0.005. Figure 9.50 shows the FFEs required by each of the crossover approaches to find this best score returned by UC, over the range of mutation levels. This shows that in the absence of mutation, FDC does not find this target. When the mutation rate is increased to 0.005, FDC is faster than both TInSSel and CalEB to find this value. For a further increase in the mutation rate, FDC is again faster than the other approaches to find a score of 698.

9.2.7 Summary

This section has reviewed FDC, in comparison to CalEB, TInSSel and UC for deriving optimal control schedules of bio-control applications. These experiments have been conducted over a range of intervention penalties, mutation rates and replacement strategies. The intervention penalties have ranged from 0, as described in Section 9.2.1, through to a penalty of 50 points per intervention, as detailed in Section 9.2.5.

When there is no mutation, FDC is shown to converge to less fit scores than those found by the other approaches. As detailed in Section 2.1.7, mutation diversifies the search directions and avoids convergence to a local optima [41]. GAs typically use mutation, thus FDC not performing in the absence of mutation is of no major detriment to the technique.

For a mutation level of 0.005, FDC is shown to outperform the other approaches for the initial FFEs, when the intervention penalty is increased beyond 5 points per intervention. This can be ascribed to FDC using the intervention penalty to drive the intervention sizing process. If there is not a strong intervention penalty, FDC is unable to determine the direction to move intervention levels, thus returning poorer scores. When there is a penalty per intervention of 20, 35 or 50, this gives FDC a clearer direction to move in, thus explaining the rapid production of good scores for these penalty levels.

For a mutation rate of 0.05, FDC is clearly the best performing crossover operator. Regardless of intervention penalty, for this mutation level FDC is quicker to identify promising intervention levels and thus returns fitter scores than the other approaches for a range of FFEs. Note that the only difference between TInSSel and FDC is the calculation of the target number of interventions. This points to using the relative differences in parent numbers as a more robust approach to directing offspring in the presence of large quantities of noise, as introduced via mutation.

When FDC is shown to outperform the other approaches, this is generally for FFEs in the early stages of the experiment. This implies that FDC is better at identifying trends in terms of intervention usage and moves in these directions in a more prompt way than the other directed intervention techniques.

FDC is shown to be poorer under a steady state replacement strategy as opposed to a generational approach for certain intervention penalties. As FDC is better with a generational approach, which replaces a large proportion of the population per generation, this points to FDC performing better with large amounts of genetic variation in the population. This would explain why FDC is less efficient under generational replacement with a p_m of 0, as opposed to mutation levels of 0.005 and 0.05.

Figure 9.51 shows the percentage of improvement, in terms of both FFEs required and statistically better scores found, through using FDC instead of UC. This shows the results for a mutation rate of 0.005, which was where UC on average performed best. This shows that for a penalty of 0 points per intervention, FDC requires 64% fewer FFEs than UC to find the same score and that for 84% of the FFEs under review, FDC returns significantly better scores.

When the penalty was increased to 5 penalty points per intervention, FDC is shown to be poorer than UC. As stated previously, FDC uses the intervention penalties to drive the intervention level selection. Therefore with a low penalty there is a low drive. Although FDC is not as quick at finding the target score as UC, this difference in performance is only 8%. For this penalty value, FDC statistically outperforms UC for 24% of the FFE range.

A further increase in penalty value, from 5 to 20, provides enough direction for FDC to accurately calculate offspring intervention levels. For this setting, FDC finds the best score found by UC using 36% less FFEs. FDC also statistically outperforms UC for 44% of the FFEs under review.

For a penalty of 35 points per intervention, FDC requires 42% fewer FFEs to find the best score returned by UC. For this penalty setting the scores returned by FDC are statistically better than those found by UC for 36% of the FFEs. A further increase in penalty value, to 50 points per intervention, shows that FDC requires 57% less FFEs to discover the final value returned by UC. As with a penalty of 35 points per intervention, FDC finds statistically better scores for 36% of all FFEs.

The average gain in performance over all penalty values for a mutation rate of 0.005 is shown in Figure 9.51. This shows that over all 5 penalty levels, FDC requires 38.2% fewer FFEs to find the best score returned by UC. On average, FDC finds statistically better scores for 44.8% of all FFEs under review. These mean values are slightly better than the values returned by TInSSel in comparison to UC. This shows that even with the poor score shown for a penalty of 5 points per intervention, on average FDC is as effective for bio-control scheduling as TInSSel.

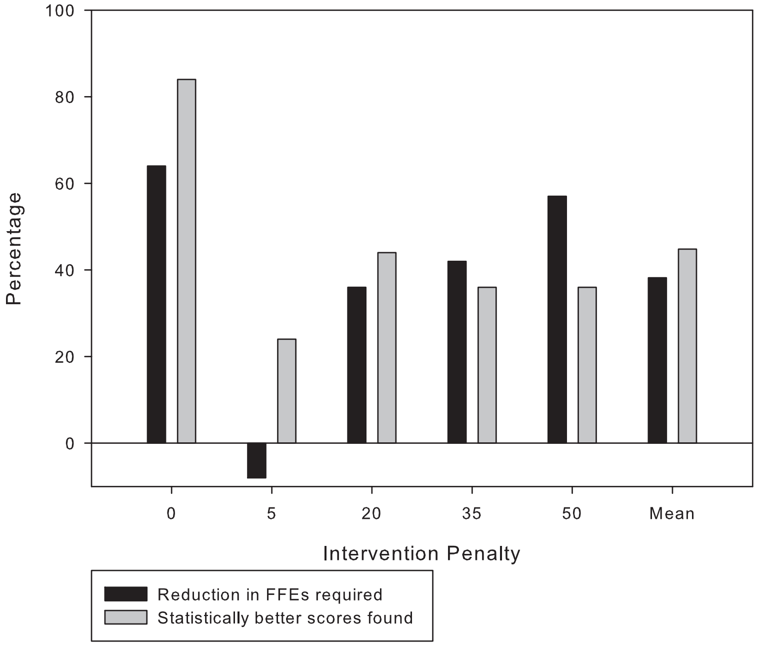


Figure 9.51: Performance improvement between UC and FDC when $p_m = 0.005$

9.2.8 Further analysis

The previous sections reviewed FDC in comparison to CalEB, TInSSel and UC for the task of bio-control scheduling. This analysis was conducted over a range of intervention penalties using intermediate parameter settings.

In order to further analyse the abilities of FDC, compared to the other crossover approaches, it is necessary to review them across a range of parameter settings, outwith the intermediate set used in the previous sections.

Appendix D details the abilities of FDC, CalEB, TInSSel and UC across a range of population sizes, crossover rates, mutation rates and intervention penalties. A review of the crossover approaches over the range of parameters are detailed in Appendix A.5. In that review, FDC was shown to follow similar trends to TInSSel for most experiments, with the exception of experiments with small population sizes or low mutation rates. These findings are in line with the points described in Section 9.2.6, as FDC appears to work best in populations with greater genetic diversity, hence the problems with small population sizes and no mutation.

9.3 Evaluation of FDC for chemotherapy treatment scheduling

FDC will now be compared to CalEB, TInSSel and UC for the task of chemotherapy treatment scheduling. The methodology used is the same as for previous chemotherapy treatment scheduling experiments. The problem is again sampled over 40,000 FFEs, at intervals of 1,000 FFEs, resulting in 40 data points for review. For each sample point, 100 runs of each crossover approach were recorded for each of the parameter combinations. The best scoring values for each run for each FFE were recorded.

A complete set of results across the parameter combinations has therefore been collected and these can be seen in Appendix H. This Appendix contains the complete set of results showing the fitness scores against FFEs associated with these experiments across a range of population sizes and crossover and mutation rates.

As with the previous experiments, the intermediate parameter settings for each approach will be reviewed in detail, rather than explicitly detailing each possible parameter combination. As with Section 9.2, the following sections will review FDC, compared to CalEB, TInSSel and UC, with a population of $N = 100$, crossover rate of $p_c = 0.9$ and mutation rates, p_m of 0, 0.005 and 0.05.

As with the bio-control experiments, described in Section 9.2 the intermediate parameter settings will use a generational replacement strategy. The placement of interventions will then be analysed. Intermediate parameter settings will then be reviewed under a steady state replacement strategy. As with previous experiments, the replacement mechanism, R, for each experiment will be defined as either generational, $R = g$, or steady state, $R = ss$.

Graphs displaying the results for both the fitness scores and associated intervention usage for the intermediate parameter settings under both generational and steady state replacement will now be reviewed. Both fitness scores versus FFEs and intervention usage versus FFEs are plotted over the range of FFE samples. As stated previously, as the cancer chemotherapy treatment optimisation problem is a maximisation problem, higher scores are fitter than lower valued scores.

Figures 9.52 to 9.54 show the fitness score and associated intervention usage for the intermediate parameter settings and a p_m of 0, 0.005 and 0.05 respectively.

Figure 9.52: Results for $N = 100$, $p_c = 0.9$, $p_m = 0$ and $R = g$

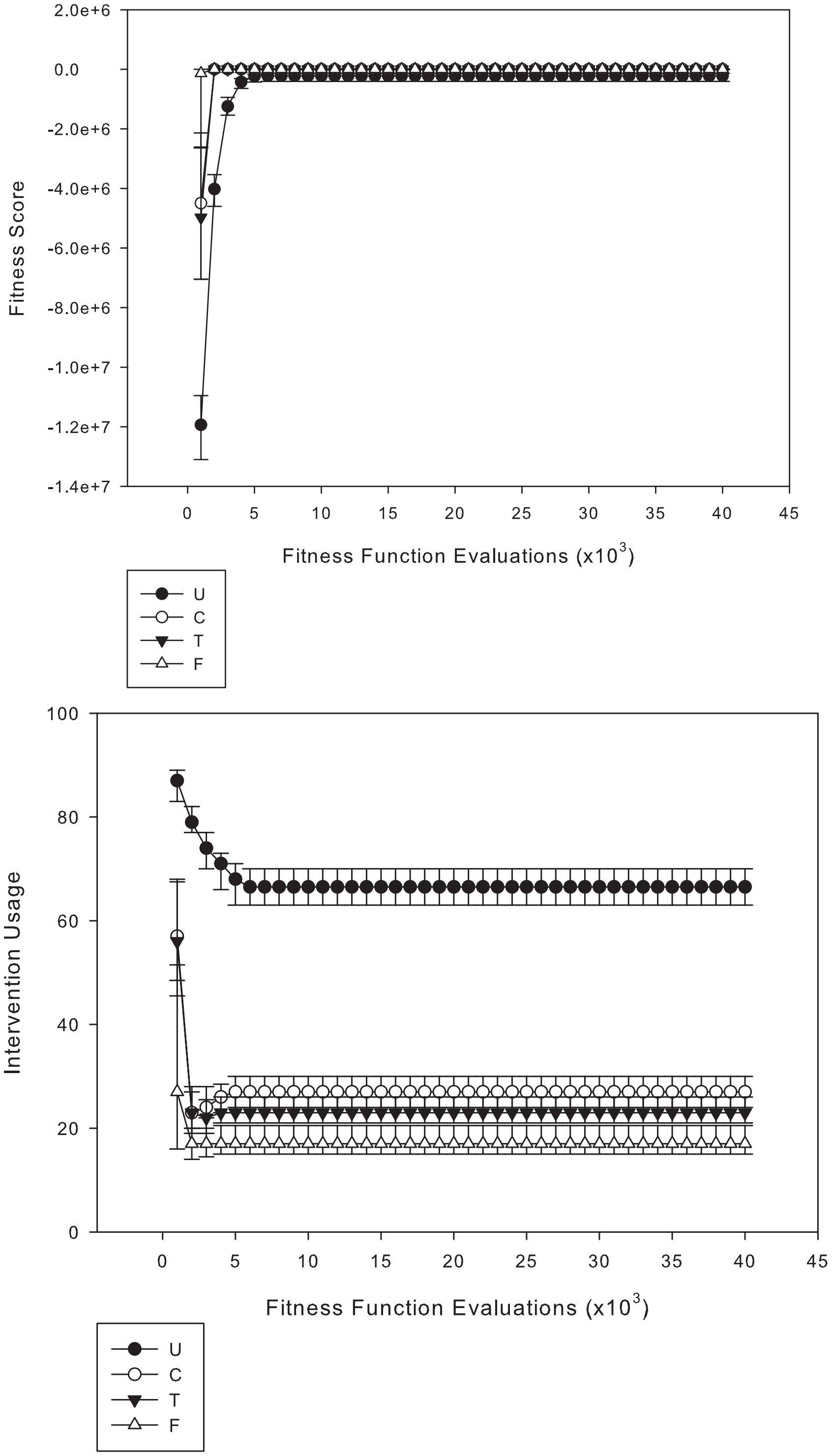


Figure 9.53: Results for $N = 100, p_c = 0.9, p_m = 0.005$ and $R = g$

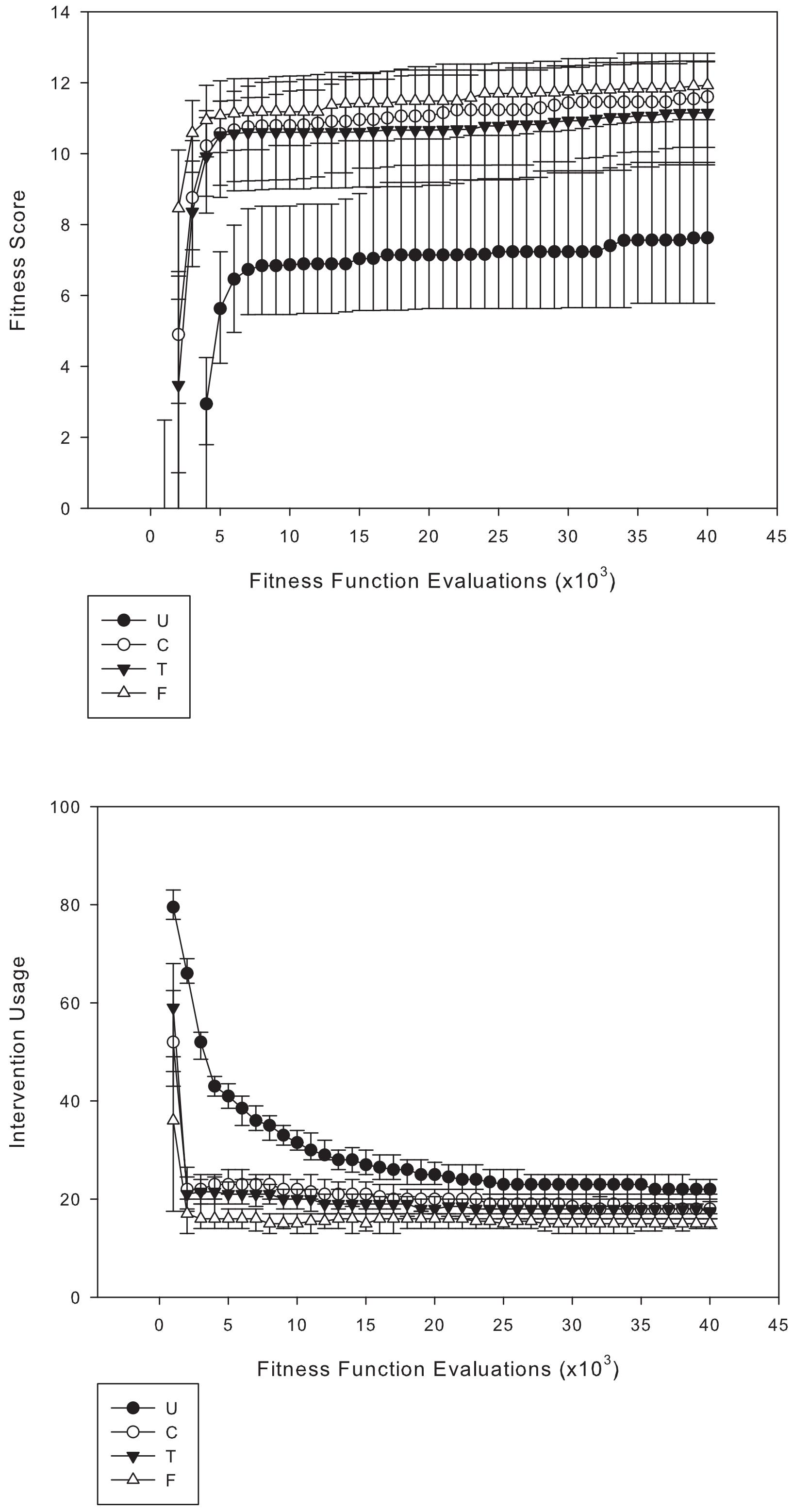


Figure 9.54: Results for $N = 100, p_c = 0.9, p_m = 0.05$ and $R = g$

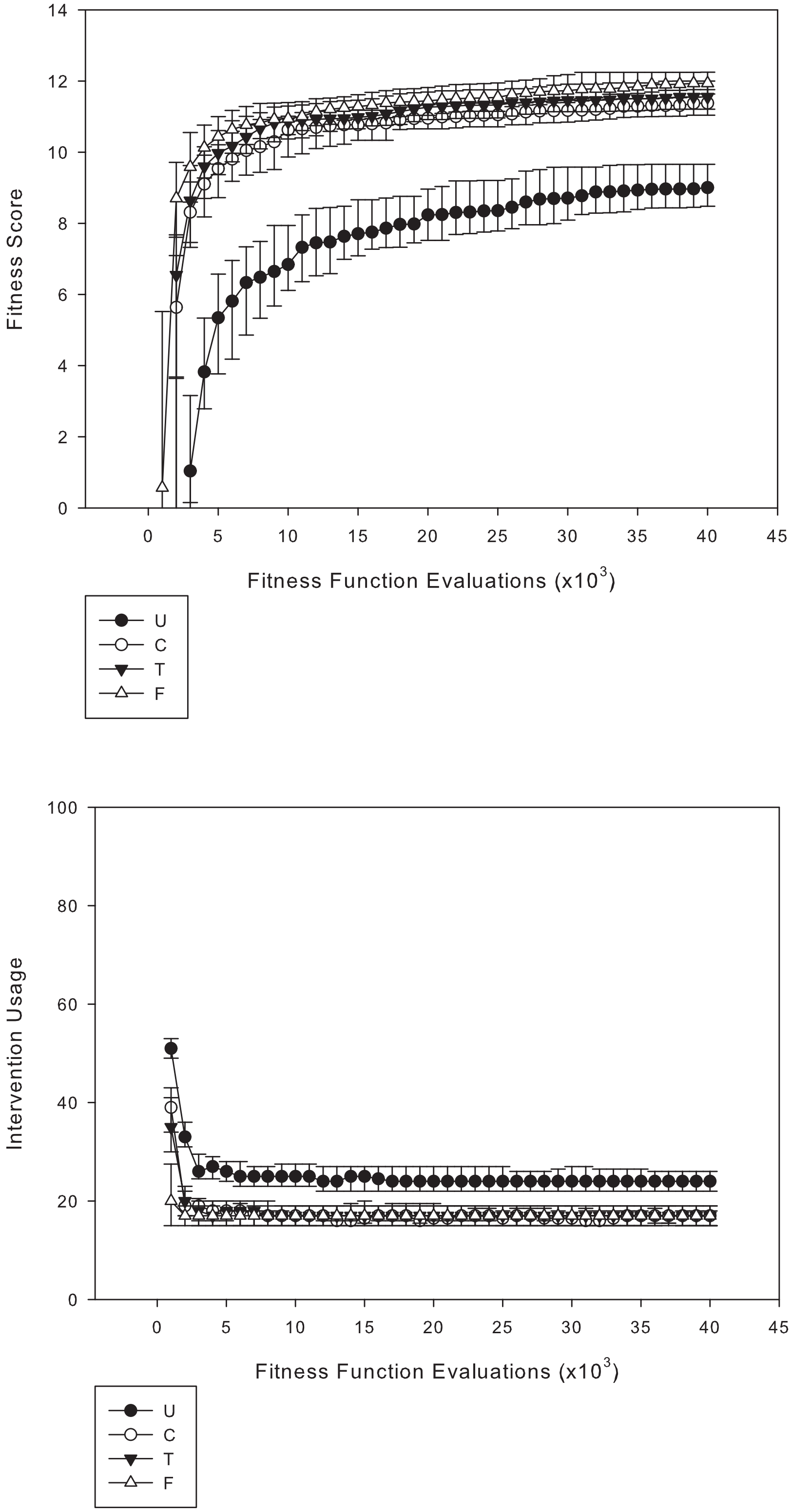


Figure 9.52 shows that in the absence of mutation, FDC uses fewer intervention than each of the other approaches. This returns fitness scores in a similar range as those produced by TInSSel and CalEB.

Figure 9.53 shows the fitness scores and intervention usage when p_m is increased to 0.005. For this mutation level, as with the case of no mutation, FDC uses fewer interventions than each of the other crossover approaches. This results in FDC, as with CalEB and TInSSel, outperforming UC for each FFE point.

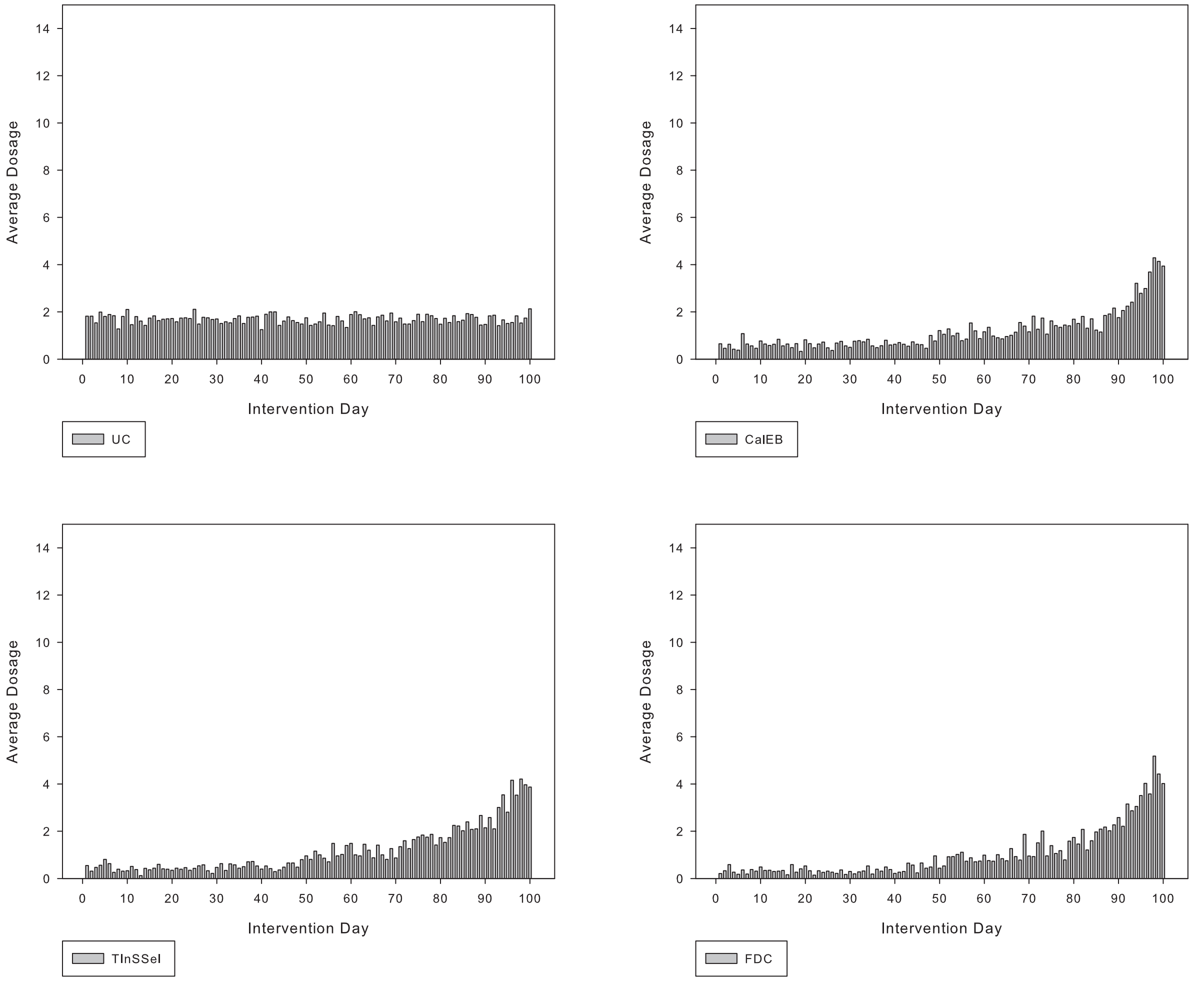
Figure 9.54 details the effect of further increasing p_m to 0.05. For this increased mutation level, FDC uses approximately the same number of interventions as TInSSel and CalEB, and returns fitter scores than each of the other approaches.

Intervention placement

The previous section has outlined the fitness scores and associated intervention usage for the FDC crossover approaches, in comparison to CalEB, TInSSel and UC, under a generational replacement strategy for cancer chemotherapy treatment scheduling.

In the case of no mutation, as shown in Figure 9.52, UC is shown to use the most interventions, followed by CalEB, then TInSSel and finally FDC. From the statistical analysis undertaken in Appendix I.4, after 40,000 FFEs, FDC is shown to outperform UC. The average placement of interventions and dosage strengths are shown in Figure 9.55. This shows that FDC, as well as CalEB and TInSSel, place larger doses at the end of the intervention schedule, whereas UC does not.

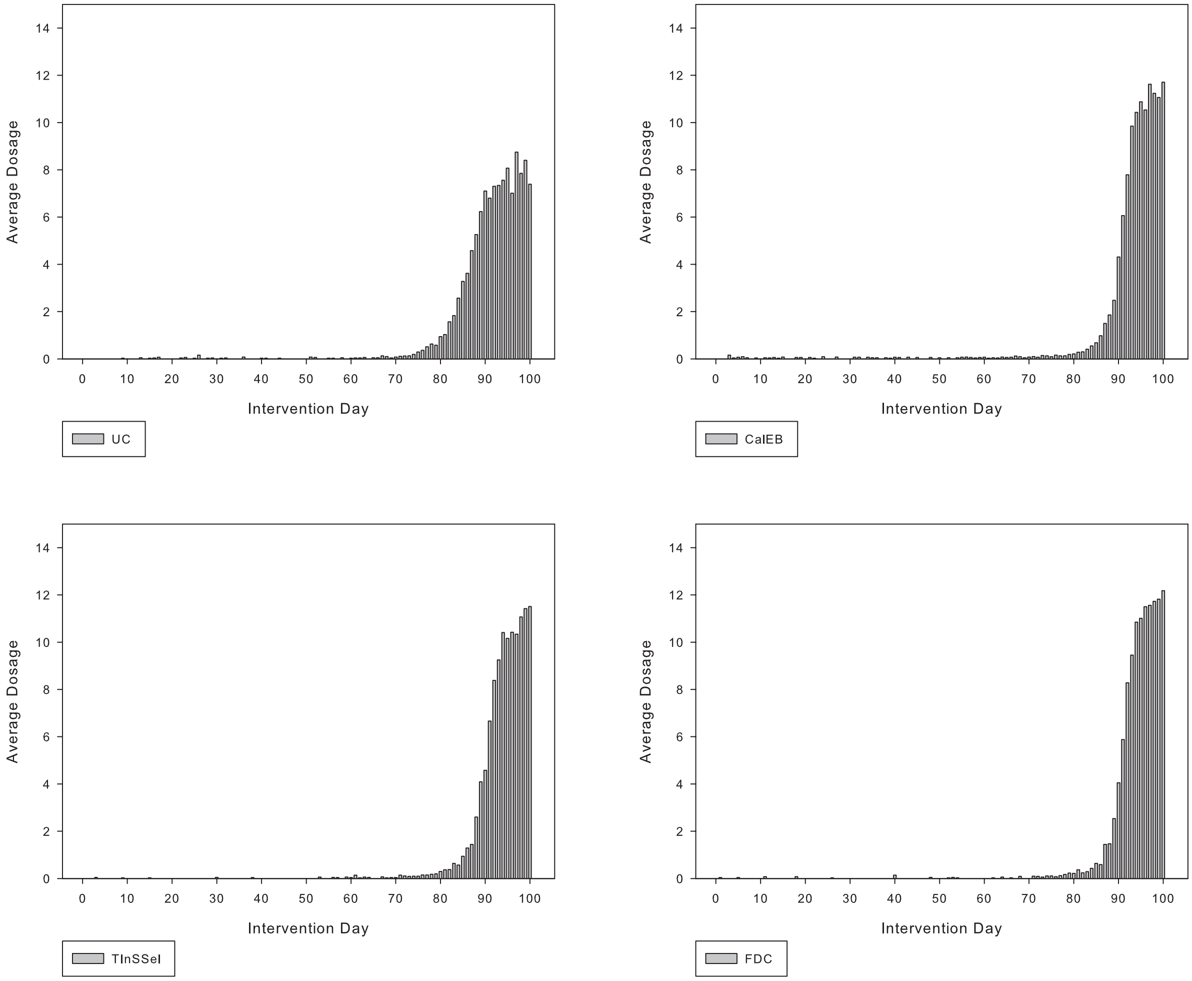
Figure 9.55: Intervention placement for 40,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0$ and $R = g$



When the mutation rate is increased to 0.005, as shown in Figure 9.53, UC uses the most interventions. CalEB and TInSSel use the same number of interventions and FDC uses less. From the statistical analysis undertaken in

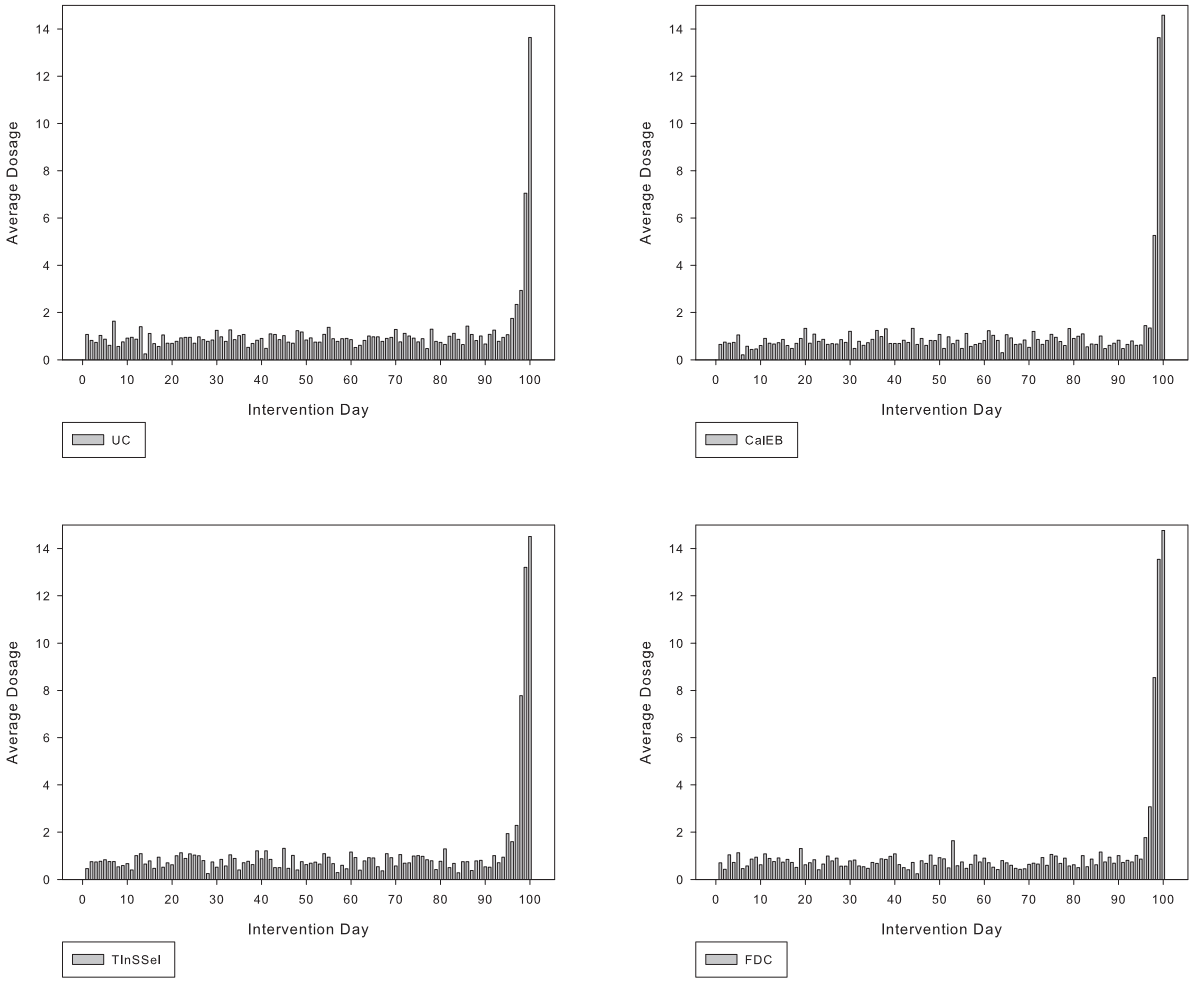
Appendix I.5, after 40,000 FFEs, FDC is shown to outperform UC. The average placement of interventions and dosage strengths are shown in Figure 9.56. This shows that while all approaches target interventions late in the treatment schedule, UC does not apply as large a dosage as the other approaches.

Figure 9.56: Intervention placement for 40,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.005$ and $R = g$



When the mutation rate is further increased to 0.05, as shown in Figure 9.54, as with previous mutation levels, UC uses the most interventions. CalEB, TInSSel and FDC converge to a similar number of interventions. From the statistical analysis undertaken in Appendix I.6, after 40,000 FFEs, FDC is shown to outperform UC, CalEB and TInSSel. The average placement of interventions and dosage strengths are shown in Figure 9.57. This shows that while all approaches target interventions late in the treatment schedule, FDC uses larger dosages on days 97 - 100 than each of the other crossover approaches.

Figure 9.57: Intervention placement for 40,000 FFEs, $N = 100$, $p_c = 0.9$, $p_m = 0.05$ and $R = g$



Steady state replacement

The same experiment will now be reviewed for a steady state replacement strategy. Figures 9.58 to 9.60 show the fitness scores and associated intervention usage for intermediate parameter settings and a steady state replacement strategy.

Figure 9.58 shows that, in the absence of mutation, as with generational replacement FDC, CalEB and TInSSel return similar fitness scores. Each of these crossover approaches are quicker to reduce intervention levels for a steady state approach, resulting in the use of fewer FFEs to find positive scores. It can also be noted that FDC uses fewer interventions under steady state replacement as opposed to the generational equivalent.

Figure 9.59 details the fitness scores and intervention usage when p_m is increased to 0.005. For this mutation level, as with the case of no mutation, FDC, CalEB and TInSSel are quicker at reducing in intervention levels under a steady state approach. For most FFEs, FDC returns similar scores under both a generational and a steady state replacement strategy.

Figure 9.60 shows that for a mutation rate, p_m, of 0.05, similar trends are displayed as for the lower mutation rates. FDC, CalEB and TInSSel are quicker to reduce intervention levels under a steady state replacement strategy, as opposed to the generational equivalent. Similar scores are returned by FDC for most FFEs, regardless of the replacement strategy used.

Figure 9.58: Results for $N = 100, p_c = 0.9, p_m = 0$ and $R = ss$

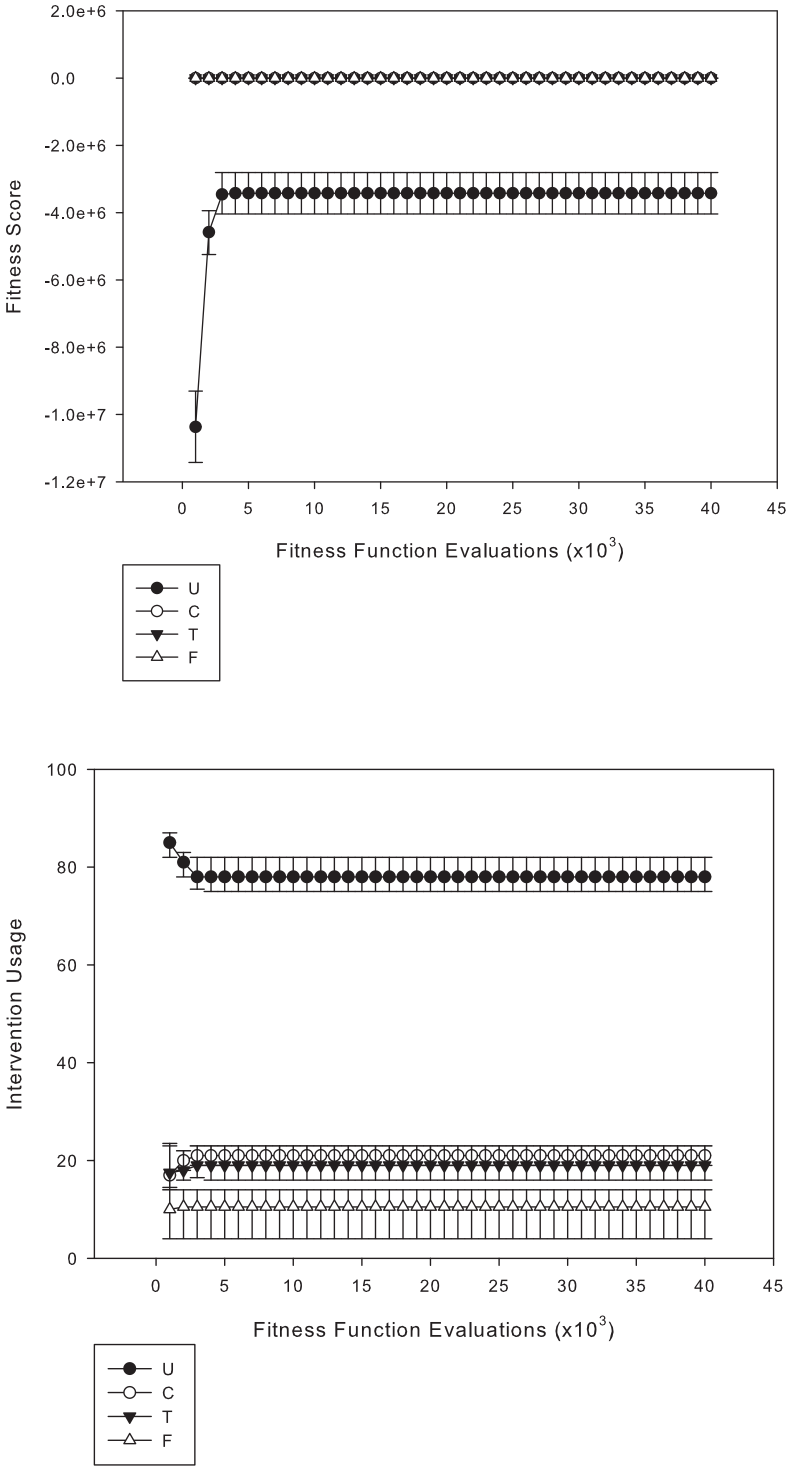
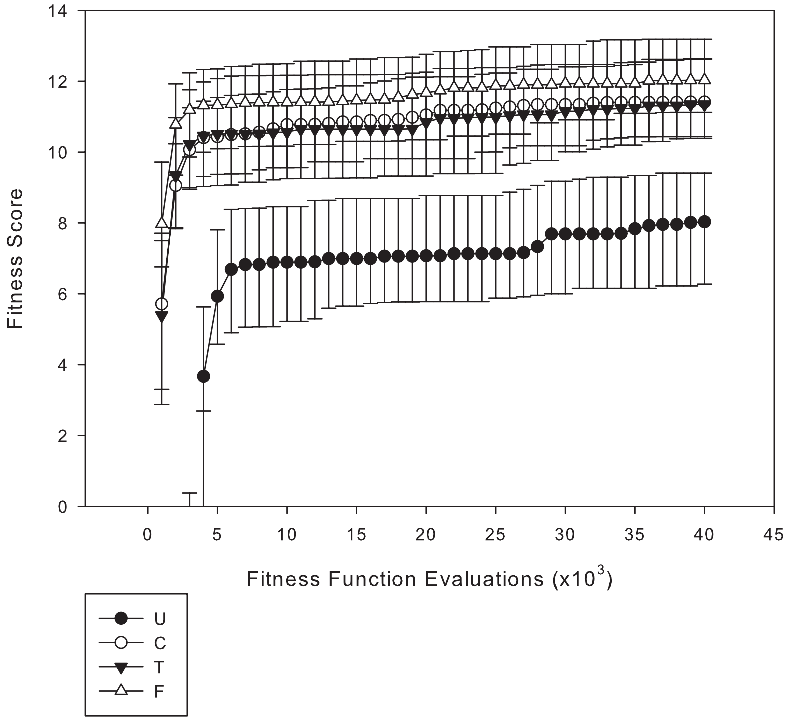


Figure 9.59: Results for $N = 100$, $p_c = 0.9$, $p_m = 0.005$ and $R = ss$



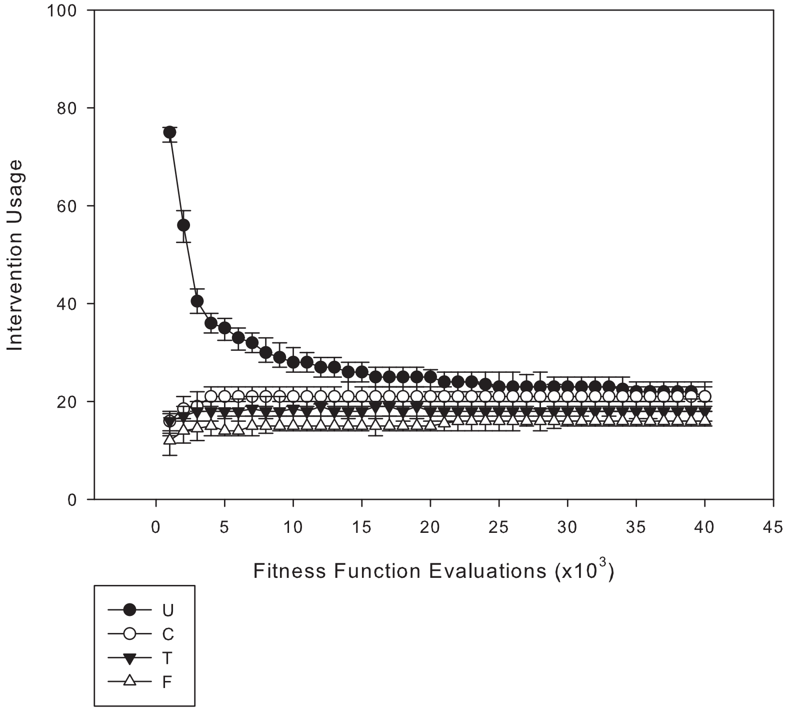
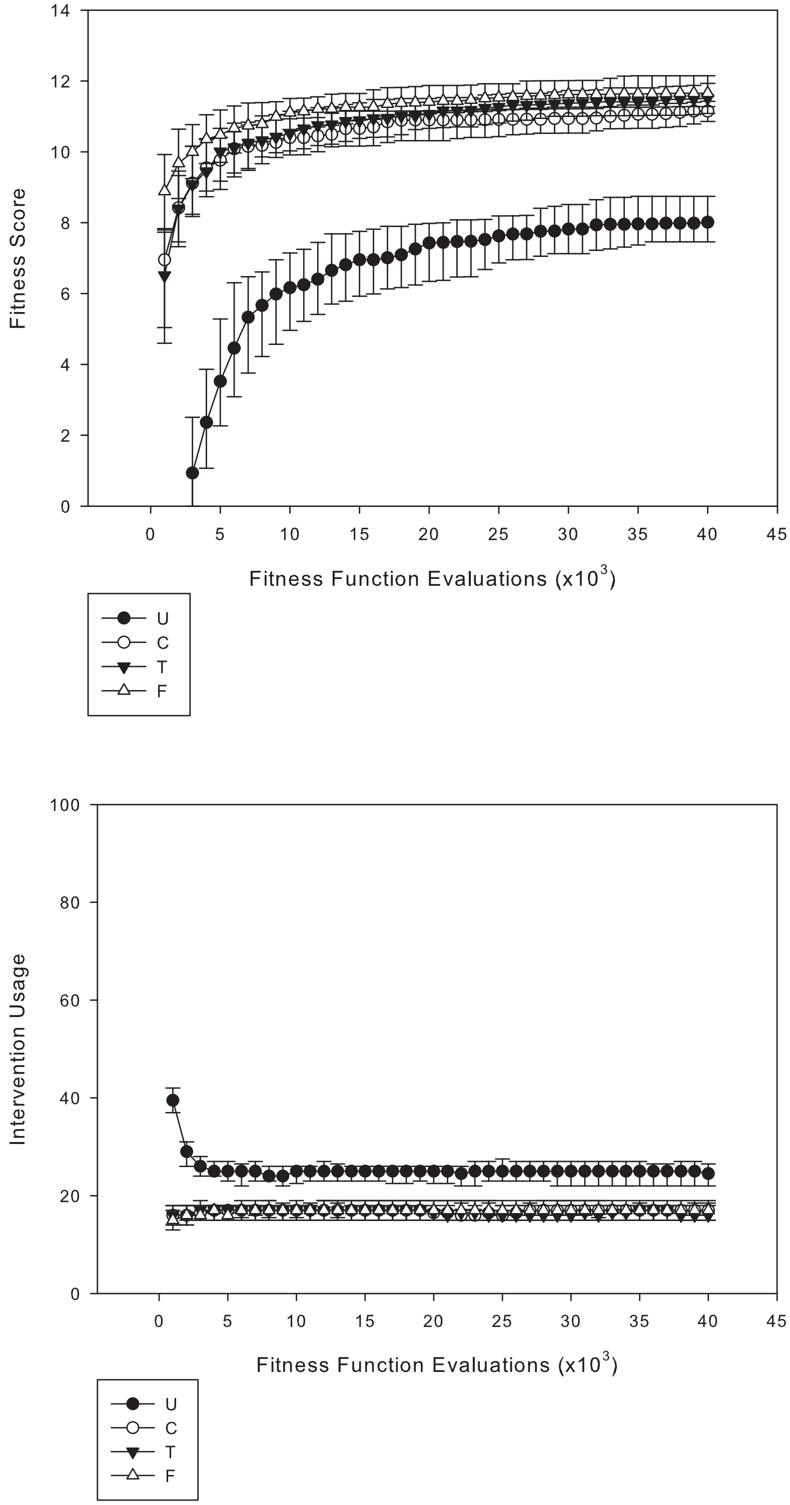


Figure 9.60: Results for $N = 100, p_c = 0.9, p_m = 0.05$ and $R = ss$



9.3.1 Summary of chemotherapy treatment scheduling experiment

The previous experiment reviewed FDC in comparison to CalEB, TInSSel and UC, for the task of cancer chemotherapy scheduling. This section draws together the salient points from this experimentation.

In order to ascertain if there is any statistical significant differences between approaches for this problem, the Kruskal-Wallis test results for the intermediate test cases were calculated. These are shown in Tables I.4 to I.6. As with the bio-control experiments discussed previously, for clarity, Appendix J.4 shows which crossover approaches are significantly different over the range of FFEs. As detailed in Section 6.9, the chemotherapy experiments are evaluated over 40,000 FFEs, in observable increments of 1,000 FFEs. This means that for each p_m level, there are 40 FFE points under observation for each approach, ranging from 1,000 FFEs to 40,000 FFEs.

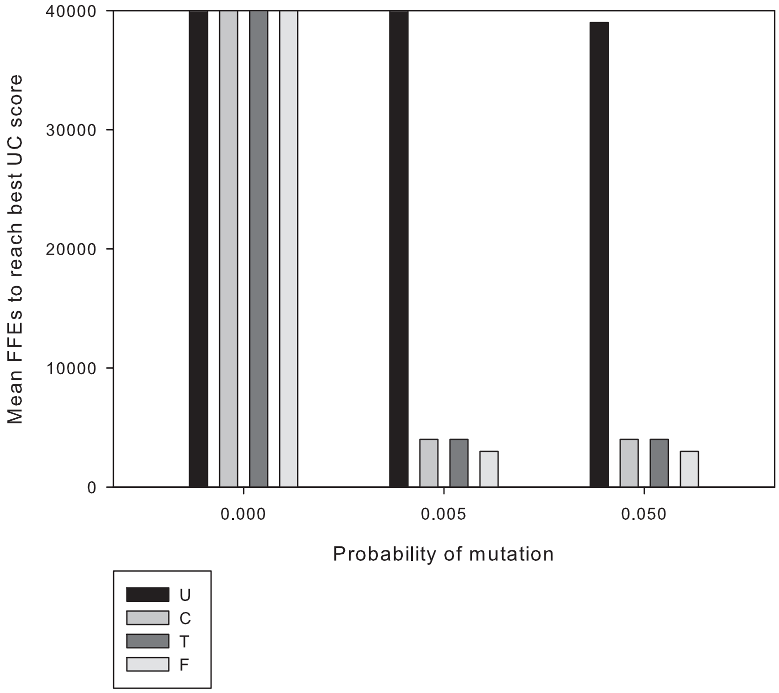


Figure 9.61: FFEs to reach the best score found by UC (9)

This section has reviewed the directed intervention crossover approaches against UC, for the production of cancer chemotherapy treatment schedules. This analysis has been undertaken over a range of mutation rates and replacement strategies.

The statistical analysis detailed in Appendix I.2 shows that there is a statistical difference between at least two of the crossover approaches from 1,000 to 40,000 FFEs for each of the mutation levels. For clarity, the crossover approaches which are statistically different to each other are described in Appendix J.4. This shows that for each of the mutation levels, FDC outperforms UC for each FFE. In the absence of mutation, FDC outperforms both CalEB and TInSSel for 1,000 FFEs. As the mutation rate is increased, so is the number of FFEs with which FDC outperforms both CalEB and TInSSel. For a mutation rate of 0.005, FDC outperforms CalEB and TInSSel from 1,000 to 4,000 and 3,000 FFEs respectively. For a further mutation increase to 0.05, FDC outperforms CalEB for every FFE and TInSSel for a fifth of all FFEs.

As FDC is always at least as good as CalEB and TInSSel for this problem, it would appear to be the crossover techniques of choice for the chemotherapy scheduling problem, regardless of p_m setting.

Over the range of mutation levels, the best score found by UC for the cancer chemotherapy scheduling problem was 9. This was found when the mutation rate was 0.05. Figure 9.61 shows the FFEs required by each of the crossover approaches to find this best score returned by UC, over the range of mutation levels. This shows that in the absence of mutation, none of the crossover approaches return this score.

When the mutation rate was increased to 0.005, Figure 9.61 shows that FDC is quicker than both CalEB and

TInSSel to reach a score of 9, requiring 3,000 FFEs, with CalEB and TInSSel both using 4,000 FFEs. For a further mutation increase to 0.05, each of the crossover approaches find a score of at least 9, the FFEs required by FDC, CalEB and TInSSel are the same as for a mutation rate of 0.005.

From inspecting the crossover approaches over a range of intervention penalties and parameter values, certain deductions can be made:

- FDC is always at least as good as the other directed intervention approaches for this problem and is therefore the crossover approach of choice for this problem.
- As this problem has more genetic variety than the bio-control scheduling problem, this shows that FDC is better for problems with more variety in the population
- FDC works for problems where the penalty per intervention is not as pronounced as the penalty in the bio-control problem.

Figure 9.62 shows the percentage of improvement, in terms of both fewer FFEs required and statistically better scores found, through using FDC instead of UC. This shows the results for a mutation rate of 0.05, which was where each of the crossover approaches on average performed best. This shows that FDC requires 92% fewer FFEs than UC to find the same score and that for 100% of the FFEs under review, FDC returns significantly better scores. As TInSSel and CalEB used 89% fewer FFEs than UC, this shows a further 3% improvement in FFEs required by using FDC. When FDC is compared to TInSSel, FDC is shown to statistically outperform TInSSel for a quarter of the FFE range and outperform CalEB for 100% of FFEs.

The previous experiments in Section 8.2.1 concluded that CalEB or TInSSel are the best performing crossover operators for this problem. This section has shown that FDC often statistically outperforms both of these techniques for chemotherapy scheduling, regardless of the mutation rate chosen. It has also shown that FDC requires fewer FFEs than either UC, CalEB or TInSSel to find these scores. Due to this, FDC is demonstrated to be a more efficient crossover approach for this problem.

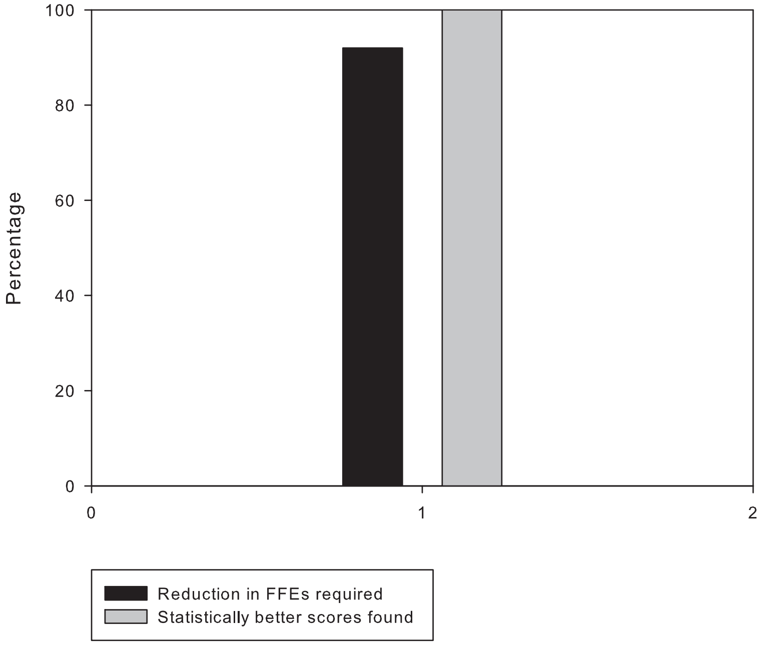


Figure 9.62: Performance improvement between UC and FDC when $p_m = 0.05$

9.3.2 Further analysis

The previous experiment reviewed FDC in comparison to CalEB, TInSSel and UC for the task of cancer chemotherapy treatment scheduling. This analysis was conducted over a range of intervention penalties using the intermediate parameter settings.

In order to further analyse the abilities of FDC, compared to the other crossover approaches, it is necessary to review them across a range of parameter settings, outwith the intermediate set used in the previous sections.

Appendix H details the abilities of CalEB, TInSSel, FDC and UC across a range of population sizes, crossover rates and mutation rates. A review of the crossover approaches over this range of parameters are detailed in Appendix A.6.

In that review, although FDC was shown to follow similar trends to TInSSel for most experiments, FDC often found fit scores at earlier stages of the experiment. As with the findings detailed in Section 9.3, FDC is often at least as good as TInSSel, and this is shown to hold regardless of population size, mutation rate or crossover rate.

9.4 Summary of the evaluation of Fitness Directed Crossover

This chapter has introduced the FDC approach as an extension to the directed intervention crossover approaches detailed earlier. The performance of FDC has been reviewed in comparison to CalEB, TInSSel and UC, for both bio-control scheduling (Section 9.2) and cancer chemotherapy treatment scheduling (Section 9.3).

FDC was shown to initially outperform the other crossover techniques in the absence of mutation for bio-control scheduling, but converged to less fit solutions than other approaches. As the mutation rate was increased, so did the performance of FDC for this problem, with FDC outperforming all of the other approaches for intervention penalties of 20, 35 and 50, when p_m was 0.005. When the mutation was further increased to 0.05, FDC was shown to outperform the other techniques, across the range of intervention penalty values. Further analysis found that FDC was at least as good as the other approaches when the mutation rate was greater than 0 and the population was greater than 50. For this problem, FDC required 38.2% fewer FFEs on average to find the best score returned by UC and outperformed UC for 44.8% of FFEs. This mean performance is a slight increase on TInSSel.

For the cancer chemotherapy scheduling problem, FDC is never outperformed by any of the other techniques and is shown to have a statistical advantage over the other approaches, regardless of mutation levels. In the further analysis of FDC for this problem, FDC was shown to perform at least as well as TInSSel, regardless of the population size or mutation rate. For this problem, FDC was shown to find the best score returned by UC in 92% fewer FFEs and returns significantly better scores than those found by UC for every FFE point.

This raises the question as to why FDC is capable of outperforming the other approaches, when p_m is 0 and $N = 50$ for the cancer chemotherapy problem but is poorer than the other approaches for equivalent parameters for bio-control scheduling? The most logical reason behind this trend would appear to be that of genetic variation in the population. FDC is shown to improve as the mutation rate and population size increases for bio-control scheduling, and in that problem, each population member is represented by a 50 bit schedule. For the cancer chemotherapy problem, each chromosome is represented by a 100 integers, where each integer has a value in the range of 0 - 15. The representation space for these two problems are therefore very different. A randomly initialised population of 50 for the chemotherapy problem could be expected to contain more variety, due to the number of genes / allowable range per gene. As FDC is shown to perform better with more variability in the population, this would explain the ability of FDC to cope with small populations and low mutation rates as long as the population has enough variety.

The ability of FDC to cope well with a large degree of genetic variation is a very useful trait. By adhering to a target number of interventions FDC can avoid much of the genetic noise introduced via large mutation rates and as many real world problems contain a lot of genetic noise, this could provide an area where the FDC approach could prove extremely useful. Further analysis would assist in further exploring the potential of FDC and indeed the other directed intervention crossover approaches. This as well as other recommendations for the future of this work are contained in the next chapter.

Chapter 10

Conclusions

This work has introduced the concept of directed intervention crossover techniques for application to optimal control problems. The previous chapters have described the motivation for these techniques, their methodology and their effectiveness compared to traditional crossover techniques for deriving schedules for both bio-control application and cancer chemotherapy treatments. This chapter discusses the findings relating to this work, showing how the hypothesis described in Chapter 1 has been proven, and outlines possible areas for further investigation.

Over the course of this work, a number of observations have been made. The core of these relate to directed intervention crossover techniques while other observations are more general in nature. Section 10.1 discusses these observations and is then followed by Section 10.2, which details the contributions to research of this work, as well as reflection on its key strengths and limitations. This chapter is concluded with Section 10.3, which details areas for further investigation relating to directed intervention crossover techniques.

10.1 Summary of findings

As stated above, both findings relating specifically to directed intervention crossover and other, more general findings, have been made. It appears prudent to consider these findings separately, therefore Section 10.1.1 details the general findings from this work and Section 10.1.2 describes those findings relating specifically to the directed intervention crossover techniques.

10.1.1 Summary of general findings

This work has reviewed both traditional crossover approaches and directed intervention crossover approaches for creating schedules for bio-control application and cancer chemotherapy treatments. Through this analysis, findings have been made, which, although not directly relating to the directed intervention techniques described in this work, are informative for problems of this type.

Chapter 7 detailed analysis of SPC, 2PC and UC for both bio-control scheduling and cancer chemotherapy treatment scheduling. This experimentation across a range of parameter values offers insight into the comparative abilities of SPC, 2PC and UC for these problems. This has shown that, regardless of mutation rates, crossover rates, population sizes or penalty values, UC is empirically demonstrated to be more robust and effective for these optimal control problems than either SPC or 2PC.

Chapters 7, 8 and 9 detail experimentation on both the bio-control scheduling problem and the cancer chemotherapy treatment problem. This work has produced a wide range of data pertaining to the fitness of solutions and the number and placement of interventions required to produce these scores. These findings should inform further research into these problems.

10.1.2 Summary of directed intervention crossover findings

Chapter 3 described the issues relating to the application of GAs to optimal control problems. One of the main drawbacks was that they were slower to converge to good solutions than other heuristic techniques.

The hypothesis put forward in Chapter 1 was that GAs can be improved for intervention based optimal control problems through using the number of interventions present in parent schedules to direct the offspring to promising areas of the search space. In order to prove this position, the novel crossover approaches described in this work were developed and tested.

Through comparison with traditional GA crossover approaches for optimisation of bio-control and cancer chemotherapy schedules, the validity of the novel crossover techniques have been demonstrated.

The experiments detailed in Chapters 8 and 9 demonstrate the benefits of using directed intervention techniques. These benefits are both in terms of better solution fitness and fewer fitness function evaluations required to find good solutions.

For the bio-control scheduling problem, TInSSel was shown to require, on average, 37% fewer FFEs than UC to find the best score returned by UC. FDC improves on this further, requiring on average, 38.2% fewer FFEs than UC to find the best score found by UC. Both TInSSel and FDC are shown to outperform UC, on average, for 44% of all FFEs.

When applied to the task of cancer chemotherapy scheduling, the gains in using TInSSel or FDC are even more apparent. TInSSel requires 89% fewer FFEs than UC to return the best score found by UC. FDC is even better, requiring 92% fewer FFEs than UC to return the best score found by UC. Both TInSSel and FDC are shown to return statistically better scores than UC for 100% of the FFEs under review.

From this work, various finding have been made with regard to the directed intervention crossover processes. Through using DUC as a test approach, this has shown that directed intervention crossover approaches are always better with a window of interventions for offspring as opposed to strictly adhering to the intervention levels of the fitter parent. Although this was the expected outcome, by creating DUC as a test method, this hypothesis has been proven. This has shown that without a window of interventions, the directed intervention process stagnates and poor scores are returned. It can be noted however, that in the presence of large quantities of noise introduced by mutation, even the simple intervention calculation of DUC is more effective than UC for each of the problems.

Initially CalEB was created to ascertain if there was any benefit to selecting genetic material in a uniform distribution over time. Through experimentation over a range of problems and parameter settings, CalEB was never once better than TInSSel, but was often outperformed by it. As both CalEB and TInSSel use the same intervention sizing calculation, this shows that for selection of genetic material, a stochastic approach to selection is more efficient than one which is uniformly distributed over time.

FDC was developed to utilise these findings, both in terms of the importance of a window for offspring intervention levels and for stochastic selection of genetic material. This has shown that the FDC approach is driven by a penalty and that when the penalty is weak, FDC is not the most effective approach. However, as shown for bio-control scheduling with larger intervention penalties and also for the chemotherapy scheduling problem, for certain problems, FDC can discover and exploit intervention trends in a much faster way than the other approaches. In the presence of large quantities of mutation, FDC is always shown to be the best performing technique. This means that for problems with large quantities of noise, FDC is the crossover approach of choice.

This work has shown that by using a target for the intervention levels for offspring, many benefits can be achieved. This can better direct offspring to promising areas of the search space, thus requiring fewer FFEs. Directing offspring intervention levels can also direct search into areas whereby the traditional approaches such as UC are not exploiting. This is most clearly shown with both TInSSel and FDC returning statistically better scores than UC for 100% of the FFE range.

The use of directed intervention approaches has also improved the robustness of approaches to mutation rate. When the mutation rate was increased to 0.05, the directed intervention approaches were less affected by this parameter change than UC. This can be explained by UC only being able to shed intervention via crossover and mutation. With all the extra interventions being added by a high mutation rate, UC takes longer to reduce intervention levels. The directed intervention approaches do not have this problem as they work to a target level of intervention. This means that regardless of the number of interventions being added via mutation, they can still produce intervention schedules with a few or a lot of interventions, depending on the pattern which offers most utility.

One of the main difficulties in applying any EA to real-world applications is the large number of fitness function evaluations typically required [113]. The reduced evaluations required by the directed intervention techniques described in this work would allow for application of GAs to computationally expensive problems. As shown in the experiments detailed in Chapters 8 and 9, significant reductions in the number of FFEs required can be achieved through using directed intervention techniques. This reduction in fitness function evaluations required would become more significant as the computational time of each fitness function increased.

The directed intervention techniques have been shown as effective at identifying fitness trends, such as rapidly increasing or decreasing intervention levels to return fitter scores. This may prove beneficial for their application to problems whose landscape changes over time and thus require prompt analysis of trends. This does raise an implication for the directed intervention techniques, that of misleading or deceptive fitness landscapes. Section 10.3 describes this as well as other areas which form the suggested future work for directed intervention crossover approaches.

The contributions to research of the current work into analysis of directed intervention crossover techniques will now be reviewed.

10.2 Contributions of research

This research has tested the hypothesis, described in Chapter 1, that GAs can be improved for intervention based optimal control problems through using the number of interventions present in parent schedules to direct the offspring to promising areas of the search space.

As described in Section 10.1.2, this hypothesis has been confirmed and the validity of the novel crossover approaches demonstrated. This work has therefore provided insight into the effects of directing offspring through using the intervention information contained in parent solutions. Other contributions to research include:

- An introduction to the directed intervention crossover approaches, allowing future researchers to apply these techniques to a wider range of problems.
- As the directed intervention techniques have been shown to significantly reduce the number of FFEs required for GAs to find fit solutions to optimal control problems, this allows for GAs to be applied to computationally intensive problems, which may previously have been infeasible.
- This work has provided empirical information about the abilities of SPC, 2PC and UC for optimal control problems, over a range of mutation rates, crossover rates and population sizes.

10.2.1 Key strengths

As stated above, this research has contributed to the field in a number of ways. The following are some other key strengths of this work:

- In order to process the volume of experiments described in this work in a feasible time period, multiple computers were required. For this reason, the Condor tool [114] was deployed and this provided for distribution of experiments over approximately 75 of the department workstations. This allowed for the range of experimentation described in this work in a reasonable amount of time. Now that this setup is in place, further researchers in the department are able to harness this environment to run computationally intensive experiments in considerably less time.
- This work has used both bio-control and cancer chemotherapy treatment scheduling test problems for analysis. The novel research conducted in this work has provided insight into these problems as well as independently confirming the findings of previous researchers as described in Sections 7.1.3 and 7.2.
- This work provides a statistically robust review of the abilities of both traditional GA crossover approaches and directed intervention crossover approaches for bio-control and cancer chemotherapy scheduling.
- This research has produced 9 conference publications. Work on turning the findings discussed herewith into a journal publication is already underway.
- This work has both further developed existing collaborative involvement and opened new areas for collaboration with a range of researchers in the field.

10.2.2 Limitations

Due to the empirical nature of this work to review the contributions of directed intervention crossover techniques, a range of experiments were required. The experimental method, as described in Chapter 6, outlined the values chosen for this experimentation. This detailed the rationale behind each of the experiment parameters which include population size, crossover rate and mutation rate. Although the values for these parameters were chosen to facilitate a fair comparison of the directed crossover techniques with existing crossover approaches, these are only a small subset of the infinite number of experiments that could be undertaken.

It was in order to facilitate a fair comparison of crossover approaches that two separate optimal control test problems were chosen for review. Although other problems could also provide valuable insight into the directed intervention crossover approaches, due to the time required in setting up, testing and reviewing experiments, further test problems were not viable.

Although there exist many other potential optimal control test problems and parameter settings for experimentation, as this work has provided key insights into the directed intervention crossover techniques, it would appear that correct experimental choices have been made. Further analysis using more test problems and parameter combinations could provide even more information about the operation and usefulness of the directed intervention crossover techniques. These as well as other potential areas for future work are now described.

10.3 Future work

This work has introduced the concept of directed intervention crossover techniques to enhance GA performance for optimal control problems. These techniques have been demonstrated as effective on the two test problems used in this work, compared to conventional SPC, 2PC and UC crossover approaches.

The main objective of this work was to determine if GAs could be made more efficient for deriving schedules of interventions for optimal control problems by reviewing the number of interventions in parents to direct offspring intervention levels. As stated in Section 10.1, this objective has been achieved for both bio-control and chemotherapy scheduling through the use of directed intervention crossover approaches.

Due to the novel nature of the directed intervention crossover approaches, a range of directions for further work exist. For clarity these areas for further work are grouped into two sections, one reviewing further analysis requiring no change to the current directed intervention approaches and the other involving change to the novel crossover techniques. Sections 10.3.1 and 10.3.2 detail each of these areas respectively.

10.3.1 Future work requiring no change to the directed intervention techniques

This work has introduced directed intervention crossover approaches as a novel way to enhance the performance of GAs for application to optimal control problems. Further use of these techniques can be achieved that do not require any modification to the approaches described in this work.

Application to other domains

In this work the novel approaches have been applied to the optimisation of intervention schedules. This task required the techniques to select from a range of potential interventions in order to optimise fitness criterion.

Although this work has focussed on selecting from a range of interventions, this could be abstracted to other application areas requiring selection of optimal members from a set of options. One such application area where the novel techniques described in this work could be applied is that of feature selection.

Work by Jourdan et al into discovering genetic features and environmental factors that are involved in multifactorial diseases used GAs to assist in feature selection [115]. This used a two stage process. The first stage involved selection of significant features from a very large dataset using the GA. The next step used a k-means clustering algorithm to pick out individuals according to the features identified by the GA. In that work, Jourdan et al found this approach to be robust and able to identify interesting associations, which were later validated by biologists.

Many of the datasets used for feature selection involve tens or hundreds of thousands of variables [116]. Although there are a large number of variables, for some problems, typically fewer than 5% of features are significant [117]. This therefore appears conducive to the directed intervention approaches. The directed intervention techniques have been empirically shown as beneficial in terms of both number of FFEs required and solutions produced for selecting from a large range of potentials, and thus could aid GA performance for the task of feature selection.

Comparison with other search heuristics

As stated in Section 3.2, GAs have proven slower at deriving good solutions to optimal control problems than other techniques, including evolution strategies, evolutionary programming, simulated annealing and particle swarm optimisation. As demonstrated in the previous chapters, the performance of GAs for deriving intervention schedules for optimal control problems has been significantly enhanced through the use of directed intervention crossover techniques. Future work could therefore include an empirical analysis of the directed intervention techniques compared to these other search heuristics. This would show how the increase in solution fitness and reduction in evaluations required to find good solutions gained through GA directed intervention techniques compares with these other search heuristics.

As described in Section 3.2, EDAs use a probabilistic sampling of the population rather than mutation or crossover and use the information gained from this sampling to produce the next set of population members. When viewed in an abstract way, the directed intervention techniques also perform a probabilistic sampling, not on large quantities of the population as per EDAs, but on the two parents selected for crossover. This relationship between EDAs and the directed intervention techniques could be further analysed to determine further similarities and differences. This could offer more understanding of the theoretical underpinnings of both of these techniques.

Multi-drug chemotherapy scheduling

The cancer chemotherapy model used in this work only reviews schedules for application of one drug over the treatment time period. As detailed in Section 5.4, GAs have previously been successfully used for multi-drug scheduling.

It would seem prudent for further analysis to consider the abilities of the directed intervention techniques at multi-drug scheduling, as this may provide key insights into their strengths and weaknesses for a more complex problem representation. As mentioned in Section 10.1.2, this would provide a more expensive fitness function, in terms of computational time per FFE, with which to apply the directed intervention techniques.

Extension of directed intervention crossover to multi-objective problems

As detailed in Section 5.4, previous work by Petrovski and McCall have reviewed representing the chemotherapy scheduling problem in a multi-objective (MO) manner [100]. As many real world problems are multi-objective in nature, it would be informative to evaluate the directed intervention techniques on problems of this type. Future work could therefore consider how best to adapt the directed intervention crossover techniques for a MO environment. These MO directed intervention crossover techniques could then be applied to the multi-objective cancer chemotherapy model. The bio-control problem described in Section 5.1 could also be extended to a MO representation. For example, a two objective extension is to minimise the number of Sciarid larvae present, and to minimise the number of nematodes or applications required.

Application of the directed intervention crossover approaches to multi-objective problems would require no changes to the crossover approaches, however the GA method used in this work would have to be modified. Common multi-objective GA approaches include Non-dominated Sorting Genetic Algorithm (NSGA2) [118] and Multi-Objective Genetic Algorithm (MOGA) [119]. Both of these approaches return a set of trade-off solutions across the multiple objectives, termed the Pareto set. This set of solutions is then evaluated by the decision maker. This would determine if the benefits demonstrated by the directed intervention techniques apply to a broader range of approaches.

Dynamic and deceptive problems

Although GAs are commonly applied to static fitness functions, or non-static problems represented in a static way, dynamic fitness landscapes are receiving increased attention [120]. In dynamic problems the fitness landscape changes over time. Solutions of such problems therefore seem to require an approach which is quicker to react to the changing landscape. As demonstrated in this work, the directed intervention techniques are quicker to identify and subsequently exploit fitness trends. Dynamic problems would therefore appear as a logical application area for techniques described in this work.

As outlined in Section 10.1, one of the key characteristics of directed intervention techniques is their effectiveness at identifying fitness trends, such as rapidly increasing or decreasing intervention levels to return fitter scores. This raises the question as to how the directed techniques would perform on problems which have a function which leads to a local optimum instead of the global optimum? Test functions of this type, termed deceptive problems, have been studied widely for GAs [121].

To ascertain the abilities of the directed intervention approach on such problems would require creation of appropriate optimal control deceptive fitness functions. This would provide useful information relating to the robustness of the directed intervention techniques to problems of this type.

Parameter settings

This work has evaluated the crossover approaches across a range of parameter settings, described in detail in Chapter 6. The range of parameter settings could be further extended to provide more detailed analysis of the novel crossover techniques.

As described in Section 6.6, binary tournament selection was used for the experiments undertaken in this work. This approach has a low selection pressure and offered no advantage to any particular crossover approach under review and thus allowed for a fair and unbiased analysis of each of the crossover approaches. Further analysis could try a range of selection operators, in order to ascertain the effect of selection type and pressure on the directed intervention crossover approaches. By increasing the tournament size, the number of competing individuals for selection, the selection pressure of tournament selection is increased. This provides for increased breeding rights to fitter solutions. Further analysis could review the effect of the tournament size or could use one of the range of other selection operators available. This would show how the convergence properties of the directed intervention approaches are linked with the choice of selection method and associated selection pressure applied.

The crossover rates used in this work are in the range 0.8, 0.9 and 1.0 as described in Section 6.4 and the mutation rates used, as described in Section 6.5, were 0, 0.005 and 0.05. This allowed for experimentation across a range of parameter settings. Further work into the abilities of the directed crossover approaches could consider reviewing other mutation and crossover rates. One such approach would be to use a low crossover rate, in the range 0.2, with a high mutation rate, in the range of 0.5. This would allow inspection of how the directed techniques progress when presented with a plethora of new material for schedules to incorporate.

As described in Section 5.3.2, a range of intervention penalty values have been experimented with for the bio-control scheduling problems. These values were 0, 5, 20, 35 and 50 points per intervention and were constant throughout the experiment. One area for further review would be to modify the intervention penalty value as the experiment is running. Starting with a low penalty would encourage a broad search of the solution space, and as the penalty increased throughout the experiment, optimal placement of interventions would become more important, thus inspecting a smaller area of the search space. This approach would provide interesting insights into the directed crossover approaches and also to the underlying bio-control model.

Framework inclusion

A framework has been proposed by Vieira and Fonseca for a conceptual model of optimisation problems [122; 123]. This aims to separate the problem specific and solver specific aspects of optimisation, to facilitate a fairer assessment and comparison of different optimisation methods [122]. Although this framework is still in development, Fonseca has proposed including the directed intervention techniques detailed in this work into this framework. This would allow for analysis of the novel techniques by a range of GA practitioners, on a wide range of problems. This could provide further insight and understanding of the directed intervention crossover techniques introduced in this work.

10.3.2 Future work requiring change to the directed intervention techniques

Section 10.3.1 details potential future work associated with the directed intervention crossover approaches that do not require any modification to the techniques described in this work. This section details other interesting areas for future work which require changes to the underlying directed intervention crossover approaches.

Retaining memory of interventions

The directed intervention techniques use the parent solutions to calculate the number of interventions to be present in offspring. Each calculation for offspring intervention levels is based on the number of interventions contained in parents and a stochastic element. Over the entire GA run, information relating to many parents and their associated intervention levels and fitness are processed. One area for further analysis could be to build a memory of this parent information over the course of the GA run and use this to assist in optimally sizing and placing offspring interventions. This would use information gathered from many parents, being refined over the duration of the GA run, and could potentially better direct offspring than the current approach which simply uses the information contained in the parents selected for breeding.

Changing intervention application calculation

As described previously, the directed intervention crossover approaches select material in a similar manner to UC. When an integer representation of genes is being used, as in the chemotherapy scheduling problem, this means that if an intervention is selected for offspring which is present in both parents, one of the parent intervention values will be copied to the offspring. Consider an instance where an intervention was chosen for addition to offspring and this was present in both parents with a dosage strength of 5 for the first parent and a dosage strength of 9 for the other. The current UC style of selection gives an offspring intervention a 50% chance of being strength 5 and a 50% chance of being strength 9. A further extension to this approach could be to review the utility of performing some form of arithmetic crossover between the dosage strengths, such as the mean value, which would create an intervention with a dosage strength of 7 from the previous example. Another approach could be to scale towards the value of the fitter parents dosage while also considering the other dosage level. This approach would allow for information relating to dosage strength to be used from both parents, as opposed to purely selecting the dosage strength from one parent and may allow for a more extensive search of the solution space.

Changing parent numbers

GAs commonly use two parents to produce children, as this is the most prevalent approach displayed in nature. Multi-parent GAs have been reviewed [124] and have been found to improve performance, with the largest performance increase being when the number of parents were changed from 2 to 3. This offered more utility than any successive increase in parent numbers.

Further work could review the effects of using more than two parents to produce offspring. This would require modification of the intervention sizing calculation, described in Section 4.1, but could allow information from a range of parents to provide a more detailed calculation of the number of interventions required to optimally direct offspring.

Increasing disruption with homogeneity

De Jong stated the usefulness in an adaptive crossover operator which would increase in disruptive potential as homogeneity of solutions increased [40]. As the population converges, many copies of similar solutions are present in the population. This may limit the usefulness of the directed intervention crossover approaches, as they use the difference in parents to calculate the offspring.

An obvious way to adapt directed intervention crossover approaches would be to increase the intervention window size as the population converges. This adapted approach could therefore be evaluated to determine if fitter solutions were found quicker than with standard directed intervention crossover approaches. Such improvement would have to be balanced against the additional cost of measuring population convergence.

References

[1] J. H. Holland, *Adaption in Natural and Artificial Systems*. University of Michigan Press, 1975.

[2] C.-F. Huang and L. M. Rocha, “Tracking extrema in dynamic environments using a coevolutionary agent-based model of genotype edition,” in *GECCO ’05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, (New York, NY, USA), pp. 545–552, ACM, 2005.

[3] Z. Michalewicz, J. Krawczyk, M. Kazemi, and C. Janikow, “Genetic algorithms and optimal control problems,” in *Proceedings of the 29th IEEE Conference on Decision and Control, 1990*, vol. 3, (Honolulu, HI, USA), pp. 1664–1666, 1990.

[4] C. Hervás-Martínez and D. Ortiz-Boyer, “Analyzing the statistical features of CIXL2 crossover offspring,” *Soft Comput.*, vol. 9, no. 4, pp. 270–279, 2005.

[5] M. Villasana and G. Ochoa, “Heuristic design of cancer chemotherapies,” in *IEEE Transactions on Evolutionary Computation*, vol. 8, (New York, NY, USA), pp. 513–521, IEEE Press, 2004.

[6] 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*, (Heidelberg, Germany), pp. 633–641, Springer, 2004.

[7] 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*, (New York, NY, USA), pp. 413–418, ACM Press, 2006.

[8] D. Vrajitoru, “Large population or many generations for genetic algorithms? Implications in information retrieval,” in *In F. Crestani & G. Pasi (Eds.), Soft computing in information retrieval. Techniques and applications*, pp. 199–222, Physica-Verlag, 2000.

[9] K. A. De Jong and W. M. Spears, “Learning Concept Classification Rules using Genetic Algorithms,” in *Proceedings of the Twelfth International Conference on Artificial Intelligence IJCAI-91*, vol. 2, pp. 651–656, Morgan Kaufmann, 1991.

[10] D. Garrett, J. Vannucci, R. Silva, D. Dasgupta, and J. Simien, “Genetic algorithms for the sailor assignment problem,” in *GECCO ’05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, (New York, NY, USA), pp. 1921–1928, ACM, 2005.

[11] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[13] Y.-J. Kim and J. Ghaboussi, “A New Genetic Algorithm Based Control Method Using State Space Reconstruction,” in *Proceedings of the Second World Conference on Struc. Control*, (Kyoto, Japan), pp. 2007–2014, 1998.

[14] F. G. Lobo, *The parameter-less genetic algorithm: Rational and automated parameter selection for simplified genetic algorithm operation*. PhD thesis, Universidade Nova de Lisboa, Lisbon, Portugal, 2000.

[15] J. He, X. Hu, and H. Lue, “Nonstandard optimal control by utilizing genetic algorithms,” *World Journal of Modelling and Simulation*, vol. 1, no. 2, pp. 75–80, 2005.

[16] D. Whitley, K. Mathias, and P. Fitzhorn, “Delta coding: An iterative search strategy for genetic algorithms,,” in *Proceedings of the Fourth International Conference on Genetic Algorithms* (R. Belew and L. Booker, eds.), (San Mateo, CA), pp. 77–84, Morgan Kaufman, 1991.

[17] C. L. Ramsey and J. J. Grefenstette, “Case-based initialization of genetic algorithms,” in *Proc. of the Fifth Int. Conf. on Genetic Algorithms* (S. Forrest, ed.), (San Mateo, CA), pp. 84–91, Morgan Kaufmann, 1993.

[18] Z. Michalewicz and C. Z. Janikow, “GENOCOP: a genetic algorithm for numerical optimization problems with linear constraints,” *Commun. ACM*, p. 175, 2007.

[19] H. Aytug and G. J. Koehler, “New stopping criterion for genetic algorithms,” *European Journal of Operational Research*, vol. 126, pp. 662–674(13), 1 November 2000.

[20] P. C. Pendharkar and G. J. Koehler, “A general steady state distribution based stopping criteria for finite length genetic algorithms,” *European Journal of Operational Research*, vol. 176, no. 3, pp. 1436–1451, 2007.

[21] T. Back, D. B. Fogel, and Z. Michalewicz, eds., *Handbook of Evolutionary Computation*. Bristol, UK, UK: IOP Publishing Ltd., 1997.

[22] P. J. B. Hancock, “An empirical comparison of selection methods in evolutionary algorithms,” in *Evolutionary Computing, AISB Workshop*, pp. 80–94, 1994.

[23] T. Bäck, “Evolutionary algorithms,” *SIGBIO Newsl.*, vol. 12, no. 2, pp. 26–31, 1992.

[24] J. E. Baker, “Adaptive selection methods for genetic algorithms,” in *Proceedings of the 1st International Conference on Genetic Algorithms*, (Mahwah, NJ, USA), pp. 101–111, Lawrence Erlbaum Associates, Inc., 1985.

[25] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Wiley, 2001.

[26] M. P. Fernando G. Lobo, David E. Goldberg, “Time complexity of genetic algorithms on exponentially scaled problems,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)* (D. Whitley, D. Goldberg, E. Cantú-Paz, L. Spector, I. Parmee, and H.-G. Beyer, eds.), (Las Vegas, Nevada, USA), pp. 151–158, Morgan Kaufmann, 2000.

[27] H. Mühlenbein, “Parallel genetic algorithms, population genetics and combinatorial optimization,” in *Proceedings of the third international conference on Genetic algorithms*, (San Francisco, CA, USA), pp. 416–421, Morgan Kaufmann Publishers Inc., 1989.

[28] T. Blickle and L. Thiele, “A comparison of selection schemes used in genetic algorithms,” Tech. Rep. 11, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Gloriastrasse 35, 8092 Zurich, Switzerland, 1995.

[29] T. Blickle and L. Thiele, “A mathematical analysis of tournament selection,” in *Proceedings of the Sixth International Conference on Genetic Algorithms* (L. Eshelman, ed.), (San Francisco, CA), pp. 9–16, Morgan Kaufmann, 1995.

[30] H. Mühlenbein and D. Schlierkamp-Voosen, “Predictive models for the breeder genetic algorithm, I.: continuous parameter optimization,” *Evol. Comput.*, vol. 1, no. 1, pp. 25–49, 1993.

[31] W. M. Spears, “Crossover or mutation?,” in *Foundations of Genetic Algorithms 2* (L. D. Whitley, ed.), pp. 221–237, San Mateo, CA: Morgan Kaufmann, 1993.

[32] K. Deb and R. B. Agrawal, “Simulated binary crossover for continuous search space,” *Complex Systems*, vol. 9, pp. 115–148, 1995.

[33] W. M. Spears and V. Anand, “A study of crossover operators in genetic programming,” in *Proceedings of the Sixth International Symposium on Methodologies for Intelligent Systems ISMIS 91* (Z. W. Ras and M. Zemankova, eds.), pp. 409–418, Springer-Verlag, 1991.

[34] S. Baluja and R. Caruana, "Removing the genetics from the standard genetic algorithm," in *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 38–46, Morgan Kaufmann Publishers, 1995.

[35] D. H. Ackley, *A connectionist machine for genetic hillclimbing*. Norwell, MA, USA: Kluwer Academic Publishers, 1987.

[36] G. Sywerda, "Uniform crossover in genetic algorithms," in *Proceedings of the third international conference on Genetic algorithms*, (San Francisco, CA, USA), pp. 2–9, Morgan Kaufmann Publishers Inc., 1989.

[37] K. Sastry, D. Goldberg, and G. Kendall, "Genetic algorithms," in *Search Methodologies Introductory Tutorials in Optimization and Decision Support Techniques* (E. K. Burke and G. Kendall, eds.), ch. 4, pp. 97–125, Springer US, 2005.

[38] W. M. Spears and K. A. De Jong, "On the virtues of parameterized uniform crossover," in *Proceedings of the Fourth International Conference on Genetic Algorithms* (R. Belew and L. Booker, eds.), (San Mateo, CA), pp. 230–236, Morgan Kaufman, 1991.

[39] 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*, pp. 55–62, IEEE press, 2007.

[40] K. A. De Jong and W. M. Spears, "An analysis of the interacting roles of population size and crossover in genetic algorithms," in *PPSN I: Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, (London, UK), pp. 38–47, Springer-Verlag, 1991.

[41] T.-P. Hong, H.-S. Wang, and W.-C. Chen, "Simultaneously applying multiple mutation operators in genetic algorithms," *Journal of Heuristics*, vol. 6, no. 4, pp. 439–455, 2000.

[42] K. A. De Jong, *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, University of Michigan, Ann Arbor, MI, USA, 1975.

[43] K. A. De Jong and J. Sarma, "Generation gaps revisited," in *FOGA* (L. D. Whitley, ed.), pp. 19–28, Morgan Kaufmann, 1992.

[44] D. Whitley, "The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best," in *Proceedings of the Third International Conference on Genetic Algorithms* (J. D. Schaffer, ed.), (San Mateo, CA), pp. 116–121, Morgan Kaufman, 1989.

[45] F. Vavak and T. C. Fogarty, "Comparison of steady state and generational genetic algorithms for use in nonstationary environments," in *International Conference on Evolutionary Computation*, pp. 192–195, 1996.

[46] M. Mitchell, S. Forrest, and J. H. Holland, "The royal road for genetic algorithms: Fitness landscapes and GA performance," in *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, 1991* (F. J. Varela and P. Bourgine, eds.), (Paris), pp. 245–254, A Bradford book, The MIT Press, 1992.

[47] M. D. Vose, "Generalizing the notion of schema in genetic algorithms," *Artif. Intell.*, vol. 50, no. 3, pp. 385–396, 1991.

[48] D. E. Goldberg and K. Sastry, "A practical schema theorem for genetic algorithm design and tuning," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)* (L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, eds.), (San Francisco, California, USA), pp. 328–335, Morgan Kaufmann, 2001.

[49] A. Petrovski, *An Application of Genetic Algorithms to Chemotherapy Treatment*. PhD thesis, Robert Gordon University, Aberdeen, Scotland, 1998.

[50] Y. Gao, "Population size and sampling complexity in genetic algorithms," in *Proceedings of the Bird of a Feather Workshops(GECCO2003)—Learning, Adaptation, and Approximation in Evolutionary Computation*, pp. 178–181, 2003.

[51] C. R. Reeves, "Using genetic algorithms with small populations," in *Proceedings of the 5th International Conference on Genetic Algorithms*, (San Francisco, CA, USA), pp. 92–99, Morgan Kaufmann Publishers Inc., 1993.

[52] F. G. Lobo and C. F. Lima, "A review of adaptive population sizing schemes in genetic algorithms," in *GECCO '05: Proceedings of the 2005 workshops on Genetic and evolutionary computation*, (New York, NY, USA), pp. 228–234, ACM, 2005.

[53] J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Trans. Syst. Man Cybern.*, vol. 16, no. 1, pp. 122–128, 1986.

[54] S. Meyer-Nieberg and H.-G. Beyer, "Self-adaptation in evolutionary algorithms," in *Parameter Setting in Evolutionary Algorithms* (F. G. Lobo, C. F. Lima, and Z. Michalewicz, eds.), Berlin: Springer, 2007.

[55] R. Hinterding, Z. Michalewicz, and T. C. Peachey, "Self–adaptive genetic algorithm for numeric functions," in *Parallel Problem Solving from Nature – PPSN IV* (H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, eds.), (Berlin), pp. 420–429, Springer, 1996.

[56] G. R. Harik and F. G. Lobo, "A parameter-less genetic algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference* (W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, eds.), vol. 1, (Orlando, Florida, USA), pp. 258–265, Morgan Kaufmann, 1999.

[57] E. Smorodkina and D. Tauritz, "Greedy population sizing for evolutionary algorithms," in *Proceedings of the IEEE Congress on Evolutionary Computation 2007*, pp. 2181–2187, Intitute of Electrical and Electronics Engineers, Inc., 2007.

[58] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.

[59] M. Athans and P. Falb, *Optimal Control: An Introduction to the Theory and Its Applications*. McGraw-Hill, 1966.

[60] R. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 1957.

[61] A. Petrovski, J. McCall, and E. Forrest, "An application of genetic algorithms to optimization of cancer chemotherapy," *International Journal of Mathematics in Education, Science and Technology*, vol. 29, no. 3, pp. 377–388, 1998.

[62] G. K. M. Pedersen, *Towards Automatic Controller Design using Multi-Objective Evolutionary Algorithms*. PhD thesis, Dept. of Control Engineering, Aalborg University, Aalborg, Denmark, 2005.

[63] Z. Michalewicz, C. Janikow, and J. Krawczyk, "A modified genetic algorithm for optimal control problems," *Computers Math. Applic.*, vol. 23, pp. 83–94, 1992.

[64] J. Bobbin and X. Yao, "Solving optimal control problems with a cost on changing control by evolutionary algorithms," in *Proc. of the 1997 IEEE Int'l Conf. on Evolutionary Computation (ICEC'97)*, (New York, NY, USA), pp. 331–336, IEEE Press, 1997.

[65] J. McCall and A. Petrovski, "Searching for optimal strategies in cancer chemotherapy using genetic algorithms," in *Seventh IMA Conference on Mathematics in Medicine and Biology*, (Oxford, England), Oxford University Press, 1996.

[66] 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:*, (Amsterdam, Netherlands), pp. 65–70, IOS Press, 1999.

[67] J. McCall and A. Petrovski, "OWCH- a decision support system for designing novel cancer chemotherapies," in *Proceedings of the First International Symposium on Soft Computing in Biomedicine*, pp. 504–510, ICSC Academic Press, 1999.

[68] A. Petrovski, A. Wilson, and J. McCall, "Statistical identification and optimisation of significant GA factors," in *Proceedings of 5th Joint Conference on Information Sciences (JCIS'2000)*, vol. 1, pp. 1027–1030, 2000.

[69] K. Boonlong, N. Chaiyaratana, and S. Kuntanapreeda, "Using a co-operative co-evolutionary genetic algorithm to solve optimal control problems in a hysteresis system," in *CEC '02: Proceedings of the Evolutionary Computation on 2002. CEC '02*, (Washington, DC, USA), pp. 1504–1509, IEEE Computer Society, 2002.

[70] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature*, (London, UK), pp. 249–257, Springer-Verlag, 1994.

[71] Y. Liang, K.-S. Leung, and T. Mok, "A novel evolutionary drug scheduling model in cancer chemotherapy," in *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, pp. 237–245, 2006.

[72] T. Back, G. Rudolph, and H.-P. Schwefel, "Evolutionary programming and evolution strategies: Similarities and differences," in *In Proceedings of the Second Annual Conference on Evolutionary Programming*, pp. 11–22, 1993.

[73] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution*. New York, USA: John Wiley, 1966.

[74] S. Smith, "An evolutionary program for a class of continuous optimal control problems," in *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, (New York, NY, USA), pp. 418–422, IEEE Press, 1995.

[75] S. Smith and R. J. Stonier, "Applying evolution program techniques to constrained continuous optimal control problems," in *International Conference on Evolutionary Computation*, (New York, NY, USA), pp. 285–290, IEEE Press, 1996.

[76] M. Schoenauer and Z. Michalewicz, "Evolutionary computation," *Control and Cybernetics*, vol. 26, no. 3, pp. 307–338, 1997.

[77] I. Rechenberg, *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution (PhD thesis). Reprinted by Fromman-Holzboog (1973)*. PhD thesis, Technical University of Berlin, Berlin, Germany, 1971.

[78] H.-P. Schwefel, *Numerical Optimization of Computer Models*. New York, NY, USA: John Wiley & Sons, Inc., 1981.

[79] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science, Number 4598, 13 May 1983*, vol. 220, 4598, pp. 671–680, 1983.

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

[81] Z. Agur, R. Hassin, and S. Levy, "Optimizing Chemotherapy Scheduling Using Local Search Heuristics," *Operations Research*, vol. 54, no. 5, pp. 829–846, 2006.

[82] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE Int. Conf. on Neural Networks*, pp. 1942–1948, IEEE Press, 1995.

[83] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions I. Binary parameters," in *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, (London, UK), pp. 178–187, Springer-Verlag, 1996.

[84] A. Petrovski, A. Brownlee, and J. McCall, "Statistical optimisation and tuning of GA factors," in *The 2005 IEEE Congress on Evolutionary Computation*, pp. 758–764, 2005.

[85] Y. Crispin, "An evolutionary approach to nonlinear discrete-time optimal control with terminal constraints," in *Informatics in control, automation and robotics I* (J. Braz, H. Araújo, A. Vieira, and B. Encarnação, eds.), pp. 89–97, Springer Netherlands, 2006.

[86] J. McCall, A. Petrovski, and S. Shakya, "Evolutionary algorithms for cancer chemotherapy optimization," in *Computational Intelligence in Bioinformatics* (G. B. Fogel, D. W. Corne, and Y. Pan, eds.), ch. 12, pp. 265–296, Wiley IEEE Press, 2008.

[87] A. Fenton, R. Norman, J. Fairbairn, and P. Hudson, “Modelling the efficacy of entomopathogenic nematodes in the regulation of invertebrate pests in glasshouse crops,” *Journal of Applied Ecology*, vol. 37, pp. 309–320(0), April 2000.

[88] R.-U. Ehlers., “Current and future use of nematodes in biocontrol: Practice and commercial aspects with regard to regulatory policy issues,” *Biocontrol Science and Technology*, vol. 6, pp. 303–316(14), 1 September 1996.

[89] I. Glazer, “Survival mechanisms of entomopathogenic nematodes,” *Biocontrol Science and Technology*, vol. 6, pp. 373–378(6), 1 September 1996.

[90] R.-U. Ehlers, “Mass production of entomopathogenic nematodes for plant protection,” *Applied Microbiology and Biotechnology*, vol. 56, pp. 623–633(5), September 2001.

[91] D. H. Gouge and N. G. M. Hague, “The susceptibility of different species of sciarid flies to entomopathogenic nematodes,” *Journal of Helminthology*, vol. 69, pp. 313–318(4), 1995.

[92] R.-U. Ehlers and H. M. T.Hokkanen, “Insect biocontrol with non-endemic entomopathogenic nematodes (steinernema and heterorhabditis spp.): Conclusions and recommendations of a combined OECD and cost workshop on scientific and regulatory policy issues,” *Biocontrol Science and Technology*, vol. 6, pp. 295–302(8), 1 September 1996.

[93] A. Fenton, R. L. Gwynn, A. Gupta, R. Norman, J. P. Fairbairn, and P. J. Hudson, “Optimal application strategies for entomopathogenic nematodes: integrating theoretical and empirical approaches,” *Journal of Applied Ecology*, vol. 39, no. 3, pp. 481–492, 2002.

[94] Y. Wu, J. McCall, P. Godley, A. Brownlee, and D. Cairns, “Bio-control in mushroom farming using a markov network EDA,” in *IEEE CEC 2008: Proceedings of the IEEE Congress On Evolutionary Computation*, (New York, NY, USA), IEEE press, 2008.

[95] K. Deb, “An efficient constraint handling method for genetic algorithms,” in *Computer Methods in Applied Mechanics and Engineering*, pp. 311–338, 2000.

[96] World Health Organization, “Fact sheet no. 297,” July 2007.

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

[98] A. S. Matveev and A. V. Savkin, “Optimal control applied to drug administration in cancer chemotherapy: the case of several toxicity constraints,” in *Proceedings of the 39th IEEE Congress on Decision and Control*, vol. 5, pp. 4851–4856, Institute of Electrical and Electronics Engineers, Inc., 2000.

[99] 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*, pp. 117–122, 1999.

[100] A. Petrovski and J. McCall, “Multi-objective optimisation of cancer chemotherapy using evolutionary algorithms,” in *EMO* (E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello, and D. Corne, eds.), vol. 1993 of *Lecture Notes in Computer Science*, pp. 531–545, Springer, 2001.

[101] H. J. C. Barbosa and A. M. e Sá, “On adaptive operator probabilities in real coded genetic algorithms,” in *SCCC ’00: Proceedings of The In Workshop on Advances and Trends in Artificial Intelligence for Problem Solving*, 2000.

[102] I. S. Lowndes, T. Fogarty, and Z. Y. Yang, “The application of genetic algorithms to optimise the performance of a mine ventilation network: the influence of coding method and population size,” *Soft Comput.*, vol. 9, no. 7, pp. 493–506, 2005.

[103] D. E. Goldberg, “Sizing populations for serial and parallel genetic algorithms,” in *Proceedings of the third international conference on Genetic algorithms*, (San Francisco, CA, USA), pp. 70–79, Morgan Kaufmann Publishers Inc., 1989.

[104] J. Alander, “On optimal population size of genetic algorithms,” in *Proceedings of Computer Systems and Software Engineering*, (New York, NY, USA), pp. 65–70, IEEE press, 1992.

[105] I. Inza, M. Merino, P. Larranaga, J. Quiroga, B. Sierra, and M. Girala, “Feature subset selection by genetic algorithms and estimation of distribution algorithms - a case study in the survival of cirrhotic patients treated with TIPS,” *Artificial Intelligence in Medicine*, vol. 23, no. 2, pp. 187–205, 2001.

[106] R. Cavill, S. L. Smith, and A. M. Tyrrell, “Variable length genetic algorithms with multiple chromosomes on a variant of the onemax problem,” in *GECCO ’06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, (New York, NY, USA), pp. 1405–1406, ACM Press, 2006.

[107] A. Sokolov and D. Whitley, “Unbiased tournament selection,” in *GECCO ’05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, (New York, NY, USA), pp. 1131–1138, ACM, 2005.

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

[109] H. H. Hoos and T. Stutzle, “Towards a characterisation of the behaviour of stochastic local search algorithms for sat,” *Artificial Intelligence*, vol. 112, no. 1-2, pp. 213 – 232, 1999.

[110] A. E. Brownlee, Y. Wu, J. A. McCall, P. M. Godley, D. E. Cairns, and J. Cowie, “Optimisation and fitness modelling of bio-control in mushroom farming using a markov network EDA,” in *GECCO ’08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, (New York, NY, USA), pp. 465–466, ACM, 2008.

[111] K. Deb, K. Sindhya, and T. Okabe, “Self-adaptive simulated binary crossover for real-parameter optimization,” in *GECCO ’07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, (New York, NY, USA), pp. 1187–1194, ACM, 2007.

[112] K. Deb and H.-G. Beyer, “Self-adaptive genetic algorithms with simulated binary crossover,” *Evolutionary Computation*, vol. 9, no. 2, pp. 197–221, 2001.

[113] Y. Jin, “A comprehensive survey of fitness approximation in evolutionary computation,” *Soft Comput.*, vol. 9, no. 1, pp. 3–12, 2005.

[114] The Condor Project, “Condor project homepage.” http://www.cs.wisc.edu/condor/overview/, March 2007.

[115] L. Jourdan, C. Dhaenens, and E. ghazali Talbi, “A genetic algorithm for feature selection in data-mining for genetics,” in *MIC2001- 4th Metaheuristics International Conference*, pp. 29–34, 2001.

[116] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[117] L. Jourdan, C. Dhaenens, and E. ghazali Talbi, “Discovery of genetic and environmental interactions in disease data using evolutionary computation,” in *Evolutionary Computation in Bioinformatics* (G. B. Fogel and D. W. Corne, eds.), ch. 14, pp. 297–316, Morgan Kaufmann, 2003.

[118] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: Nsga-ii,” in *PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, (London, UK), pp. 849–858, Springer-Verlag, 2000.

[119] C. M. Fonseca and P. J. Fleming, “Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization,” in *Proceedings of the 5th International Conference on Genetic Algorithms*, (San Francisco, CA, USA), pp. 416–423, Morgan Kaufmann Publishers Inc., 1993.

[120] R. K. Ursem, “Multinational GAs: Multimodal optimization techniques in dynamic environments,” in *In Proceedings of the Second Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, 2000.

[121] D. Dasgupta, “Handling deceptive problems using a different genetic search,” in *Proceedings of the First IEEE Conference on Evolutionary Computing vol.1*, pp. 27–29, 1994.

[122] C. C. Vieira and C. M. Fonseca, “A unified model of optimisation problems,” in *GECCO ’07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, (New York, NY, USA), pp. 1537–1537, ACM, 2007.

[123] C. C. Vieira and C. M. Fonseca, “A conceptual model of optimisation problems,” in *IEEE WSSEC 2008: Workshop and Summer School on Evolutionary Computing 2008*, IEEE, 2008.

[124] A. E. Eiben, P.-E. Raué, and Z. Ruttkay, “Genetic algorithms with multi-parent recombination,” in *PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature*, (London, UK), pp. 78–87, Springer-Verlag, 1994.

Appendix A

Further analysis of crossover approaches over a range of parameter settings

Chapters 7, 8 and 9 give analysis of the results of crossover approach experiments using only intermediate parameter settings. The complete set of experiments, across the range of parameter settings are recorded in Appendixes B, C, D, F, G and H.

It is prudent to identify whether the trends described for crossover approaches with the intermediate parameter settings are displayed across the entire range of parameter values. This would ensure one, or a collection of crossover approaches, which are identified as better than others, are better throughout the range of experiments and not simply on the intermediate parameter settings.

The rest of this section is organised as follows. Section A.1 reviews the traditional crossover approaches of SPC, 2PC and UC for bio-control scheduling. Section A.2 reviews the same crossover approaches for the task of cancer chemotherapy scheduling. As described in Chapter 7, as UC is shown to consistently perform at least as well as the other approaches, it is therefore used as the benchmark for further experiments. Section A.3 reviews UC, CalEB, TInSSel and DUC for bio-control scheduling, and Section A.4 reviews the crossover approaches for cancer chemotherapy scheduling. Sections A.5 and A.6 review UC, CalEB, TInSSel and FDC for the task of bio-control and cancer chemotherapy scheduling respectively.

The range of parameter settings are probability of crossover, p_c, of 0.8, 0.9 and 1.0, probability of mutation, p_m, of 0, 0.005 and 0.05, and population size, N, of 50, 100 and 150. The bio-control experiments are also reviewed over a range of intervention penalty values, P, of 0, 5, 20, 35 and 50. Each of the results discussed in this section are for a generational replacement strategy, $R = g$.

As the bio-control scheduling problem is a minimisation problem, lower scores are fitter. The cancer chemotherapy scheduling problem is a maximisation problem, with higher scores reflecting higher fitness.

A.1 Further analysis of SPC, 2PC and UC for bio-control scheduling

This section reviews the traditional crossover approaches for the task of bio-control scheduling over the range of parameter settings. Appendix B contains graphs detailing the results of these experiments across the range of penalties per intervention.

The results for experiments over this range of parameter settings for a penalty of 0 points per intervention are shown in Figures B.1 to B.5. For a population size of 50, when there is no mutation, UC returns fitter scores than both SPC and 2PC. UC is more flexible than both SPC and 2PC, in that it can produce more combinations of parent genes. Even with a small population with no new material introduced through mutation, UC can exploit the available genetic material to perform a more thorough search than both SPC and 2PC. As 2PC performs more gene exchange than SPC it should return better scores than SPC for small populations. As shown in populations of 50 in Figures B.1 to B.5, this is the case.

Another result of having a population size of 50 is that regardless of the crossover rate, when the mutation rate is 0.005, UC appears to derive fit solutions quicker than both SPC and 2PC. When the mutation rate is increased to 0.05, this represents a large increase in genetic material introduced through mutation. All crossover approaches under review perform in a similar way, returning scores which are less fit that those found when p_m is 0.005.

When population size is increased to 100 or 150, the patterns described for the smaller population of 50 are still displayed. With increased population size there is more variation in the population for both SPC and 2PC to exploit. Due to this, both SPC and 2PC return fitter scores in the absence of mutation when the population is 100, compared to when the population size was 50. Since the larger population size increases the material in the population, each of the approaches take longer to find good solutions. If the p_m level of 0.005 is compared between populations of 50 and 100, similar fitness trends are displayed, but it takes longer for these highly fit solutions to be found with a larger population.

This experiment has shown that probability of crossover, p_c, has little effect on each of the crossover approaches, regardless of p_m, population size or penalty level.

Figures B.6 - B.10, B.11 - B.15, B.16 - B.20 and B.21 - B.25 shows the results for each of the crossover approaches over the range of parameter settings for intervention penalties of 5, 20, 35 and 50 respectively. When the intervention penalty is increased to any of these points per intervention, the patterns described for intervention penalties of 0 are still shown.

A.2 Further analysis of SPC, 2PC and UC for cancer chemotherapy scheduling

This section reviews the traditional crossover approaches for the task of cancer chemotherapy scheduling over the range of parameter settings. The relevant graphs are contained in Appendix F.

The results for a population of 50 are shown in Figures F.1 to F.3 and these detail each of the crossover approaches with p_c of 0.8, 0.9 and 1.0 respectively. For each of these experiments, UC outperforms both SPC and 2PC when p_m is 0. This is inline with the findings discussed in Section 7.2.1. It should be noted that in the absence of mutation, although the scores returned by UC are of a much higher fitness than those returned by SPC or 2PC, these scores are still poor.

When p_m is increased to 0.005, regardless of the p_c level, each of the approaches return scores of a similar fitness. This is also seen when p_m is further increased to 0.05, but note that this increased mutation rate produces better scores for each of the crossover approaches compared with a p_m of 0.005.

The effects of increasing the population size from 50 to 100 are shown in Figures F.4 to F.6. The trends displayed with the population of 50 hold. UC outperforms both SPC and 2PC when p_m is 0, with each of the approaches returning fitter scores with this increased population setting. Each of the crossover approaches appear similar for p_m levels of 0.005 and 0.05 for most FFE levels. Once again, fitter scores are returned by each of the crossover approaches when p_m is 0.05 compared to a mutation rate of 0.005. These trends hold regardless of p_c being 0.8, 0.9 or 1.0.

The effect of further increasing the population size to 150 is shown in Figures F.7 to F.9. As with populations of 50 and 100, when p_m is 0, UC outperforms both SPC and 2PC. The increased population size produces fitter scores by each of the crossover approaches in the absence of mutation. When p_m is increased to 0.005 or 0.05, approaches return similar scores over the range of FFEs, however, UC appears to be outperforming both of the other approaches for the first 6,000 FFEs when p_m is 0.005.

As with the bio-control experiments described in Appendix A.1 it is interesting to note that p_c has little effect on each of the crossover approaches for this experimentation, regardless of mutation probability or population size.

A.3 Further analysis of UC, CalEB TInSSel and DUC for bio-control scheduling

This section reviews UC, CalEB, TInSSel and DUC for the task of bio-control scheduling over the range of parameter settings. Appendix C contains graphs detailing the results of these experiments across the range of penalties per intervention.

Fitness graphs for comparison of UC, CalEB, TInSSel and DUC for the bio-control scheduling problem with 0 penalty per intervention are shown in Figures C.1 to C.5. When the population size is 50, as shown in Figures C.1 and C.2, when p_m is 0, regardless of p_c level, CalEB and TInSSel are quicker to derive fit solutions. UC does find solutions with similar fitness scores, but requires more FFEs to do so. DUC converges to solutions of a lesser fitness than those found by the other approaches. When p_m is increased to 0.005 or 0.05, CalEB and TInSSel find similar solutions to UC and DUC, but require fewer FFEs to do so. For a mutation probability of 0.05, DUC is quicker to find good solutions than UC and each of the crossover approaches are slower to find comparable solutions to those found when $p_m = 0.005$.

When the population size is increased to 100, as shown in Figures C.2 to C.3, similar trends are demonstrated as for a population of 50. CalEB and TInSSel are consistently faster at deriving solutions of at least the same fitness as the other approaches. When p_m is 0.05, DUC is faster than UC at converging to good solutions.

If the population size is further increased to 150, the trend demonstrated for populations of 50 and 100 still hold. Figures C.4 and C.5 show the fitness scores for this population size. Although all of the crossover approaches converge to similar solutions when p_m is 0.005 or 0.05, there is a marked difference in the number of FFEs required for convergence. CalEB and TInSSel are faster than UC or DUC at discovering the solutions to which each of the approaches later converge.

A penalty of 5 penalty points per intervention is now considered. The fitness scores for these tests are shown in Figures C.6 to C.10.

For a population of 50, and p_m of 0, regardless of p_c, DUC converges to less fit solutions than CalEB, TInSSel and UC. CalEB appears slower than UC, TInSSel and DUC when the population is 50, regardless of p_m, although it does converge to solutions in the same fitness range after approximately 1000 FFEs. This is most clearly displayed when p_m is 0.05.

When the population size is increased to 100, as shown in Figures C.7 and C.8, regardless of p_c, CalEB appears slower to find good solutions than the other approaches. As with a population of 50, DUC struggles when p_m is 0 and converges to less optimal solutions than those returned by UC, CalEB and TInSSel. When $p_m = 0.005$, TInSSel is faster to converge to good solutions. When p_m is 0.05, all approaches return scores in a similar fitness range, especially after 1000 FFEs. Note that all of the crossover approaches are slower to converge to solutions of a similar fitness score when the population is 100 as opposed to a population of 50.

Figures C.9 and C.10 show the fitness scores for approaches when the population size is increased to 150. As with populations of 50 and 100, DUC again converges to sub-optimal solutions when p_m is 0, although the scores are fitter than with lesser population sizes. When p_m is increased to 0.005 or 0.05, all crossover approaches eventually produce similar solutions. CalEB however, is slower to converge to good solutions than the other approaches. With the increased population size, each of the approaches take longer to find solutions of a similar fitness to those found with a smaller population.

An intervention penalty of 20 penalty points per intervention is now considered. Figures C.11 to C.15 show the fitness scores associated with these experiments.

When the population is 50, in the absence of mutation, DUC converges to less optimal solutions than UC, CalEB or TInSSel. Both CalEB and TInSSel are quicker to reduce to good solutions than UC, with CalEB and TInSSel converging to good solutions by 600 FFEs, while UC takes around 1000 FFEs to find solutions in a similar fitness range.

When p_m is increased to 0.005, regardless of p_c, the ordering of approaches for convergence to fit solutions is consistent with that already seen. CalEB and TInSSel are quicker to reduce in fitness scores, requiring around 700 FFEs, with UC requiring 1200 FFEs to derive solutions of a similar fitness. DUC finds solutions of a similar fitness as the other approaches but requires around 2000 FFEs to achieve this.

When p_m is further increased to 0.05, a distinct separation occurs between the scores returned by UC and DUC compared to the scores produced by CalEB and TInSSel. For p_c of 0.8 or 0.9, even after 5000 FFEs, there is a difference in fitness of solutions produced by these groups. When p_c is 1.0, UC still converges to sub-optimal solutions, whereas after 5000 FFEs, DUC, CalEB and TInSSel are returning fitter scores.

Figures C.12 and C.13 show the fitness scores when the population is increased to 100. When p_m is 0, as with populations of 50, DUC converges to less fit solutions than UC, CalEB and TInSSel. When p_m is increased to 0.005, CalEB and TInSSel quickly converge to good solutions after approximately 1000 FFEs. UC is slightly slower than both CalEB and TInSSel, finding solutions of a similar fitness, but requiring around 1700 FFEs to do so. DUC does find similar solutions as the other approaches but is much slower to do so, requiring between 2000 FFEs to 3000 FFEs to achieve this.

When p_m is increased to 0.05, CalEB and TInSSel converge to better scores than both UC and DUC. The converged scores for DUC lie between the worse scores of UC and the better scores of both CalEB and TInSSel.

Figures C.14 and C.15 show the fitness scores when the population is increased to 150. Similar trends are displayed as those shown for populations of 50 and 100. When p_m is 0, CalEB and TInSSel rapidly find fit solutions, requiring about 1500 FFEs to do so. UC finds scores of a similar fitness as CalEB and TInSSel, but requires around 2300 FFEs to do so. DUC converges to solutions of less fitness than all of the other approaches for this level of p_m.

When p_m is increased to 0.005, all approaches converge to similar solutions. CalEB and TInSSel find these scores after approximately 1500 FFEs, UC requires around 2500 FFEs and DUC is only starting to converge to solutions of similar fitness at around 5000 FFEs. Each of the approaches take longer to find solutions of a similar fitness for the increased population size.

When p_m is further increased to 0.05, as with a population of 100, CalEB and TInSSel converge to fitter solutions than both UC and DUC, with DUC again producing solutions with a fitness in between that found by UC and those returned by CalEB and TInSSel.

Several trends have been identified in this experiment. In the absence of mutation, CalEB and TInSSel are quicker to converge to fitter solutions than both UC and DUC. UC returns less fit scores than CalEB and TInSSel, with DUC returning scores of less fitness than all other approaches. When p_m is increased to 0.005, CalEB and TInSSel are quicker to find fit solutions. When p_m is further increased to 0.05, CalEB and TInSSel are again quicker to converge to fit solutions. For this mutation level, DUC returns solutions of a lesser fitness than both CalEB and TInSSel, but of a higher fitness than those returned by UC. Each of these trends hold with an increase in the population. One result of increasing the population size is the ability of DUC to return fitter scores in the absence of mutation, due to the increased genetic variation in the population. Another result of increasing the population size is that with each population increase, the time taken to find solutions of a similar fitness in terms of FFEs, by each of the crossover approaches, increases.

Each of the trends identified for an intervention penalty of 20 penalty points per intervention hold as the intervention penalty is increased to 35 or 50. Figures C.16 to C.20 and Figures C.21 to C.25 show the fitness scores for these experiments under varying levels of population sizes, p_c and p_m with a penalty of 35 penalty points per intervention and 50 penalty points per intervention respectively.

A.4 Further analysis of UC, CalEB TInSSel and DUC for cancer chemotherapy scheduling

This section reviews UC, CalEB, TInSSel and DUC for cancer chemotherapy scheduling over the range of parameter settings. Appendix G contains graphs detailing the results of these experiments across the range of parameter settings.

Graphs detailing the fitness results for each crossover approach for the chemotherapy scheduling problem are shown in Figures G.1 to G.9. When the population size is 50, as shown in Figures G.1 and G.3, when p_m is 0, CalEB and TInSSel return scores of a higher fitness than both UC and DUC. For this mutation level, UC returns less fit scores in comparison to CalEB and TInSSel, with DUC producing scores of a lesser fitness than UC.

When p_m is increased to 0.005, for this population size, CalEB and TInSSel are quicker to find fitter scores than both UC and DUC over the first 2000 FFEs. By reviewing the zoomed graphs of this setting, CalEB and TInSSel always produce higher scores than UC or DUC. As was the case for a p_m level of 0, UC returns scores that are less fit than those produced by CalEB and TInSSel, with DUC producing scores of a lesser fitness than UC.

When p_m is further increased to 0.05, as with a p_m of 0.005, CalEB and TInSSel are quicker than UC and DUC at producing high fitness scores over the initial 2000 FFEs. Unlike the previous mutation levels, DUC outperforms UC in terms of fitness scores. Each of the approaches return scores of a higher fitness with this increased mutation level, when compared to lesser p_m levels.

The fitness graphs when the population is increased to 100 are shown in Figures G.4 to G.6. The trends identified for smaller population settings hold for this increased population size. Similarly for a population of 150, as shown in Figures G.7 to G.9.

As with the experiments reviewing the traditional crossover approaches for cancer chemotherapy scheduling as described in Appendix A.2 note that p_c has little effect on each of the crossover approaches for this experimentation, regardless of mutation probability or population size.

A.5 Further analysis of UC, CalEB TInSSel and FDC for bio-control scheduling

This section reviews UC, CalEB, TInSSel and FDC for the task of bio-control scheduling over the range of parameter settings. As TInSSel has consistently produced at least as fit scores as UC and CalEB, as described in Section A.3, this section will focus on evaluation of FDC with respect to TInSSel. Appendix D contains graphs detailing the results of these experiments across the range of penalties per intervention.

The results for experiments over this range of parameter settings for a penalty of 0 points per intervention are shown in Figures D.1 to D.5. For each of the mutation levels, FDC follows similar trends to TInSSel for each of the population sizes / crossover levels.

When a penalty of 5 points per intervention is introduced, the dynamics of the problem are changed. Figures D.6 to D.10 show graphs representing the results of these runs. In these experiments, FDC is seen to return scores in similar fitness to TInSSel, requiring fewer FFEs to do so. One exception is that for a population size of 50, in the absence of mutation, FDC returns scores with lesser fitness compared to those found by TInSSel or CalEB.

The results for increasing the penalty per intervention further to 20 points per intervention are shown in Figures D.11 to D.15. As was the case for a penalty of 5 points per intervention, in the absence of mutation, for a small population size, FDC returns less fit scores than TInSSel. Apart from this exception, for all other parameter settings, FDC returns similar fitness scores to TInSSel, requiring fewer FFEs to do so. Both of these trends are demonstrated for further penalty increases, as shown by an intervention penalty of 35 penalty points per intervention (Figures D.16 to D.20) or 50 penalty points per intervention (Figures D.21 to D.25).

A.6 Further analysis of UC, CalEB TInSSel and FDC for cancer chemotherapy scheduling

This section reviews UC, CalEB, TInSSel and FDC for the task of cancer chemotherapy scheduling over the range of parameter settings. Appendix H contains graphs detailing the results of these experiments. As with the bio-control experiments described in Section A.5, FDC will be compared with TInSSel as this has been shown to always be at least as good as the other crossover approaches.

Figures H.1 to H.3 show the crossover rates and mutation rate combinations for a population size of 50. In the absence of mutation, FDC requires fewer FFEs to find similar fitness scores to those found by TInSSel. When the mutation rate is increased to 0.005, FDC is again quicker at finding fit scores than TInSSel, with the fitness of scores found being higher than those found by TInSSel. With a further mutation rate increase to 0.05, FDC is quicker at finding fit scores than TInSSel, with both of the approaches returning scores of a similar fitness. As with all of the other crossover approaches, FDC produces fitter scores with the increased mutation rate.

The effects of increasing the population size to 100 are shown in Figures H.4 to H.6. Regardless of the mutation rate, FDC requires fewer FFEs to produce scores of a similar fitness to those returned by TInSSel. This is also the case when the population is further increased to 150, as shown in Figures H.6 to H.9.

A.7 Summary of further analysis of crossover approaches over the range of parameter settings

The previous sections have reviewed the crossover approaches over a range of parameter settings for both bio-control and cancer chemotherapy scheduling.

Various trends have been identified across approaches. Regardless of the optimisation problem or crossover approach used, p_c has little effect on the results produced. This has previously been identified for the cancer chemotherapy problem by Petrovski et al [84].

For the bio-control experiments, a mutation rate of 0.005 was found to be more optimal than a higher setting of 0.05. This was not the case with the chemotherapy scheduling problem, with fitter results being returned by the larger mutation rate. This result is in line with the findings of Petrovski et al [84], who found a mutation rate of 0.092 as optimal for the cancer chemotherapy scheduling problem.

The range of population sizes chosen has proved sufficient for the bio-control problem. This is demonstrated by crossover approaches finding similar scores with larger population sizes, but requiring more FFEs to do so compared to a smaller population setting.

SPC, 2PC, UC and DUC all produce better results with a p_m of 0 with a larger population. This is logical in that larger populations allow more variety in the genetic representations for the crossover approaches to exploit.

Through reviewing DUC, it is interesting to note that although poor when p_m is 0 or 0.005, it is better than UC when p_m is 0.05. This leads to the conclusion that a target intervention, even a basic approach as used by DUC, is more beneficial than no target when large quantities of material are being added through mutation.

Appendix B

Bio-control scheduling graphs for SPC, 2PC and UC

B.0.1 0 penalty points per intervention

Figure B.1: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

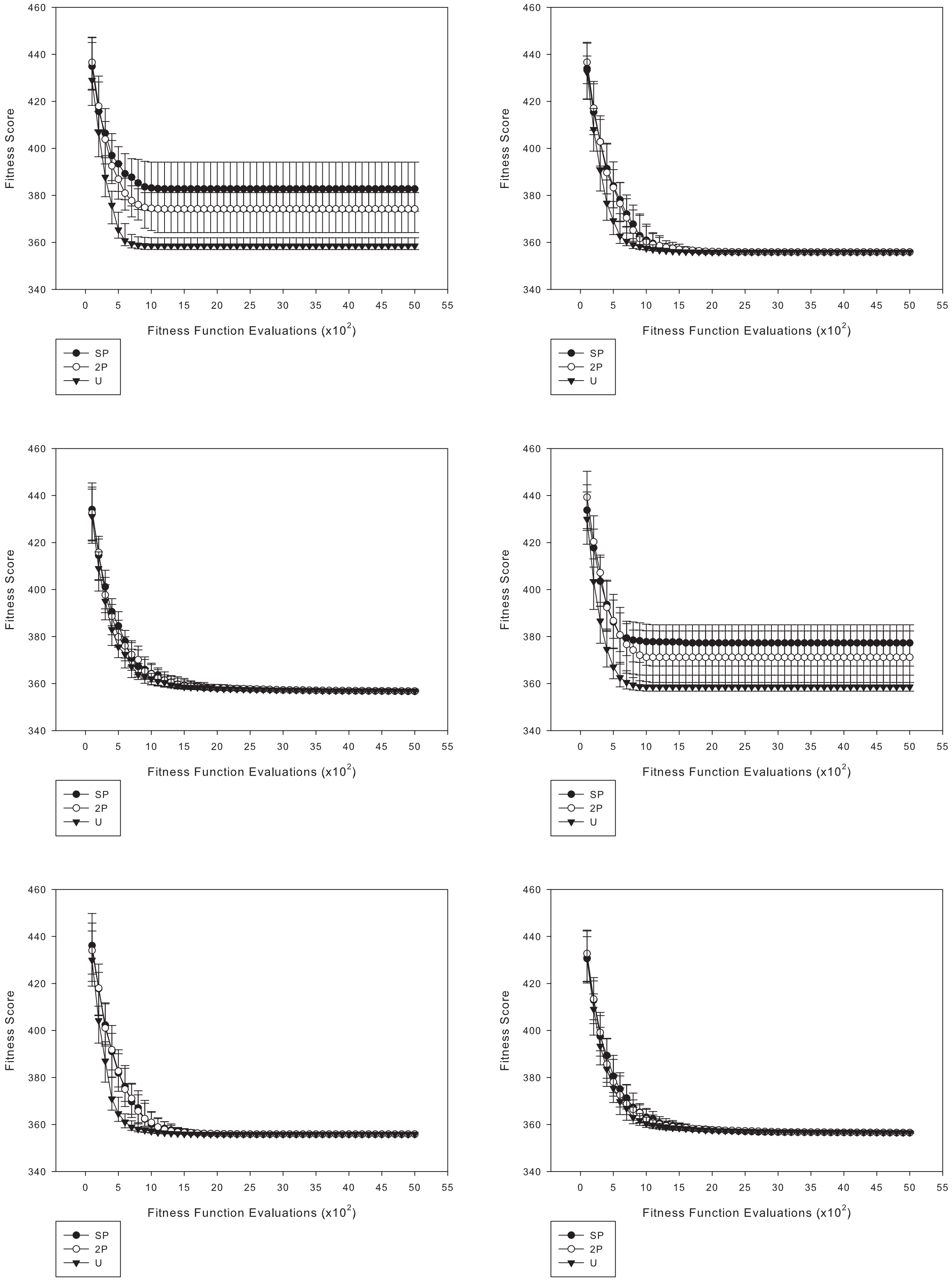


Figure B.2: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

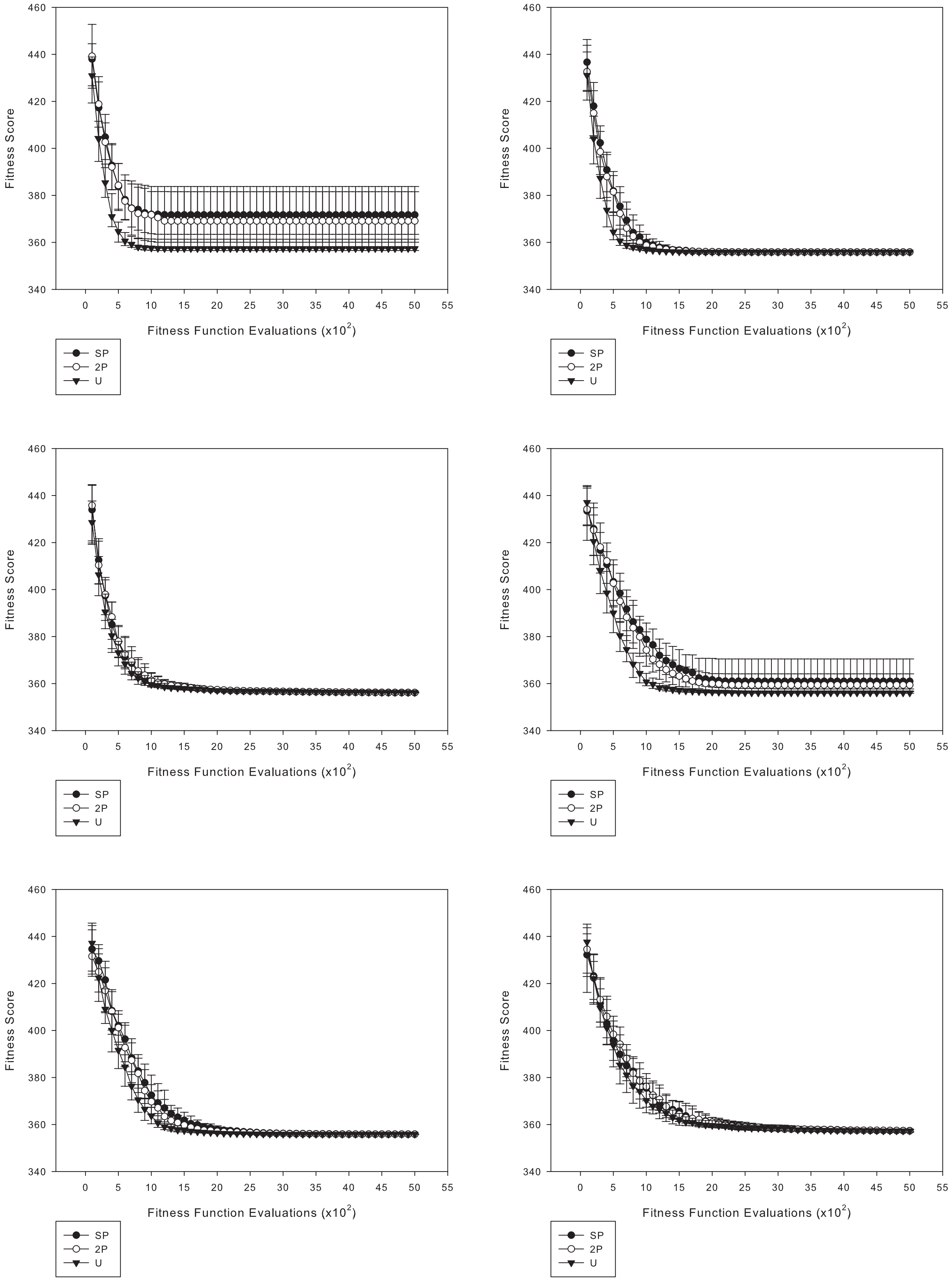


Figure B.3: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

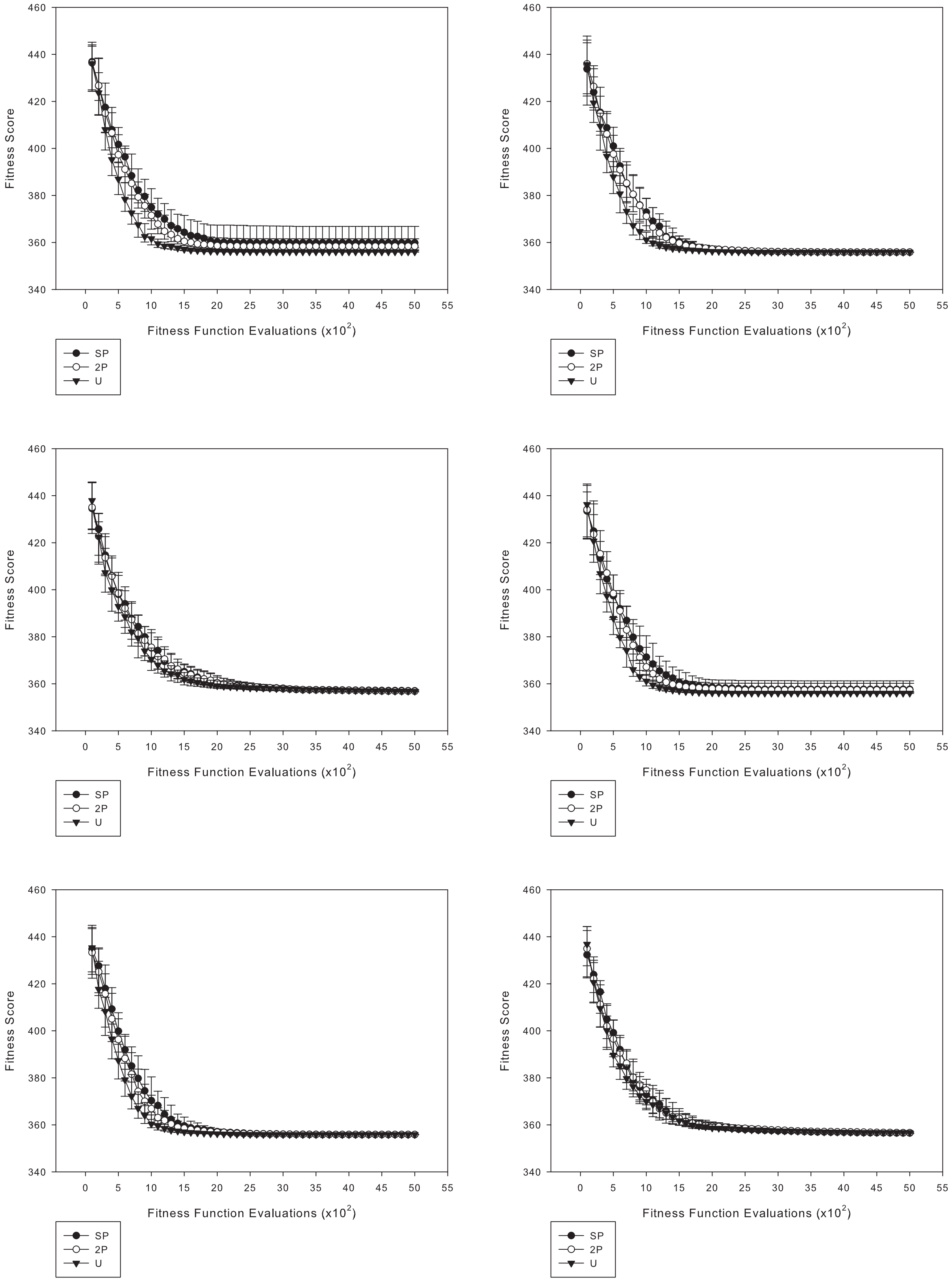


Figure B.4: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

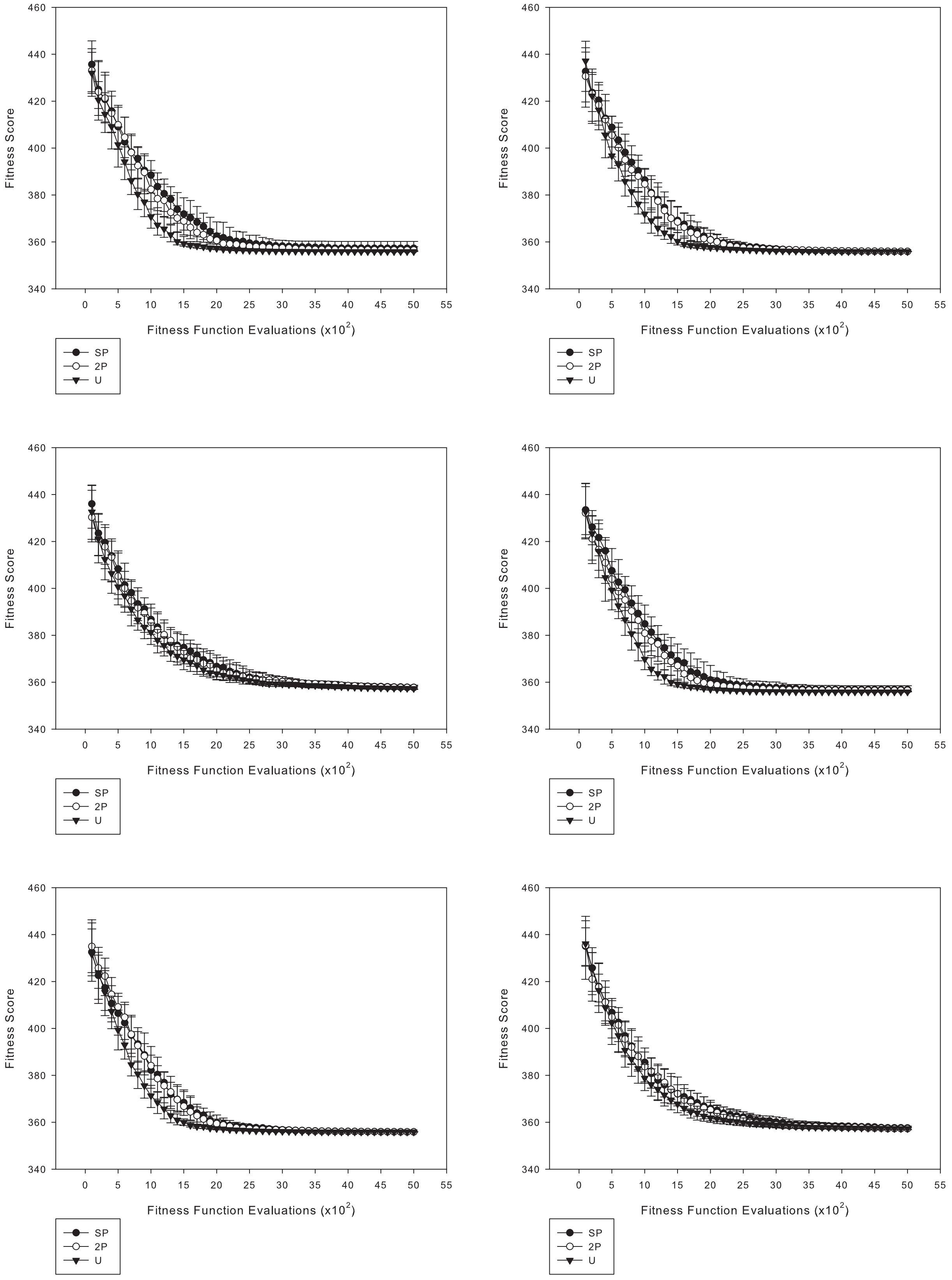
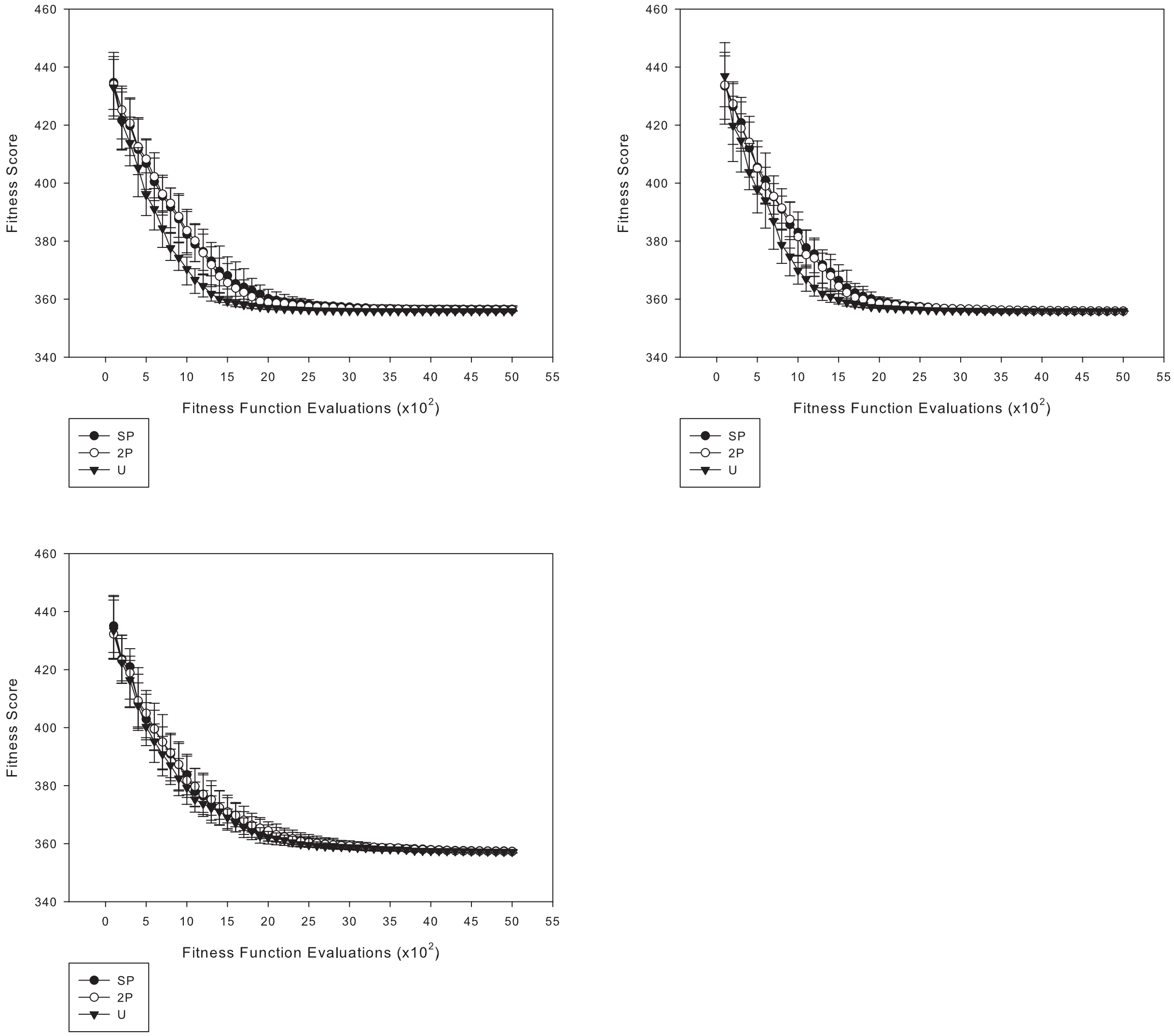


Figure B.5: $N = 150$, p_c 1.0, with p_m 0, 0.005



B.0.2 5 penalty points per intervention

Figure B.6: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

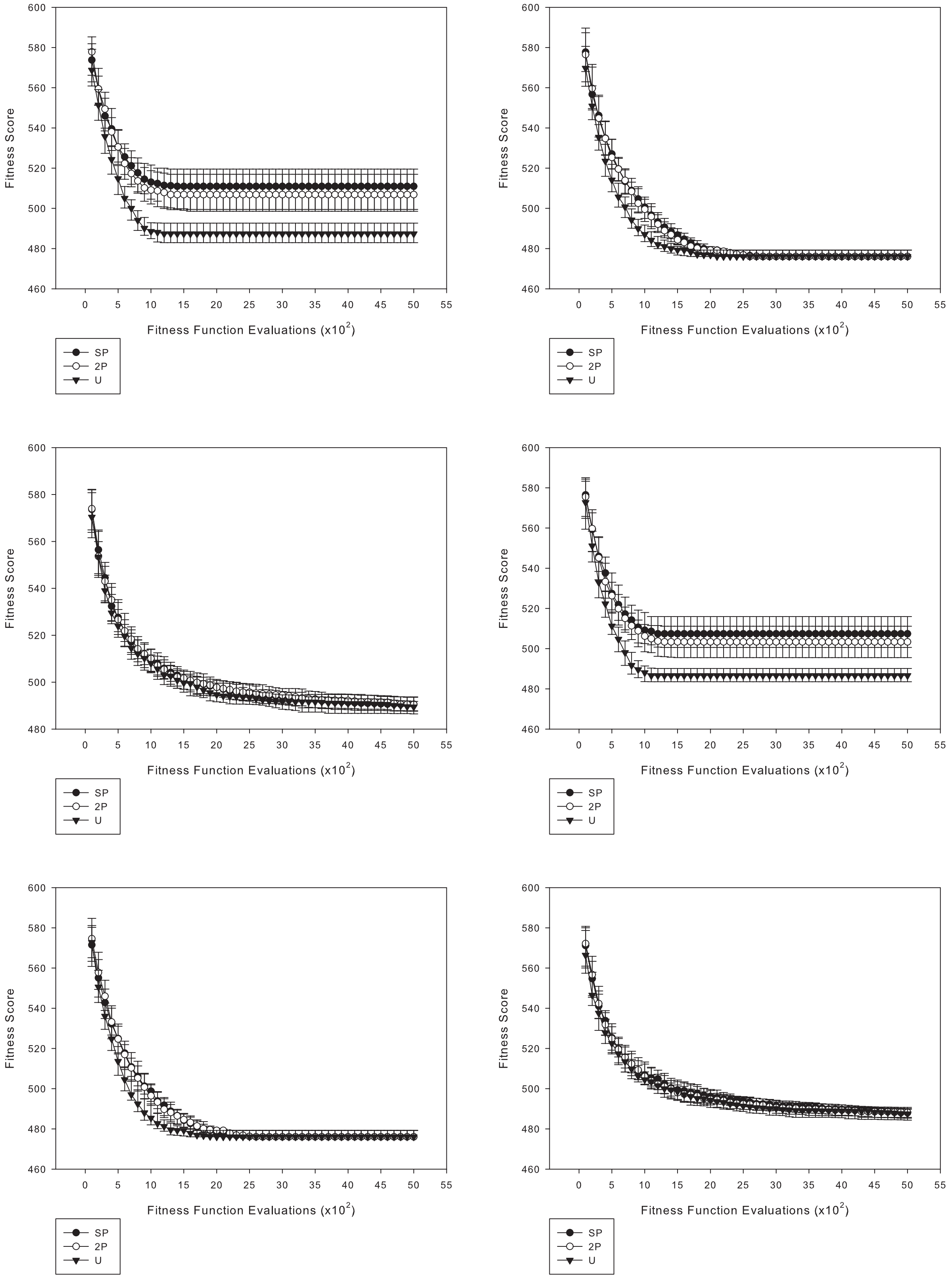


Figure B.7: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

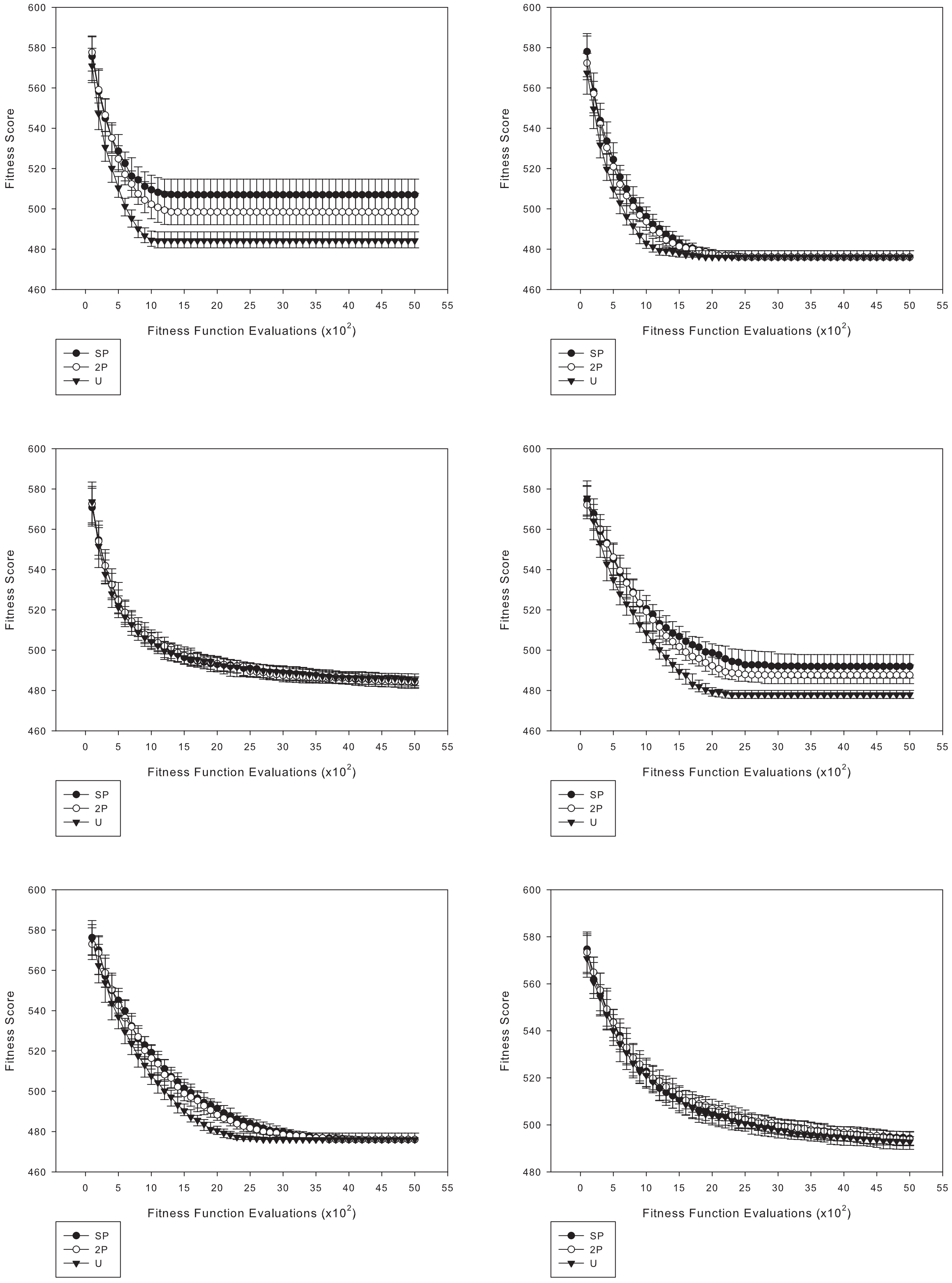


Figure B.8: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

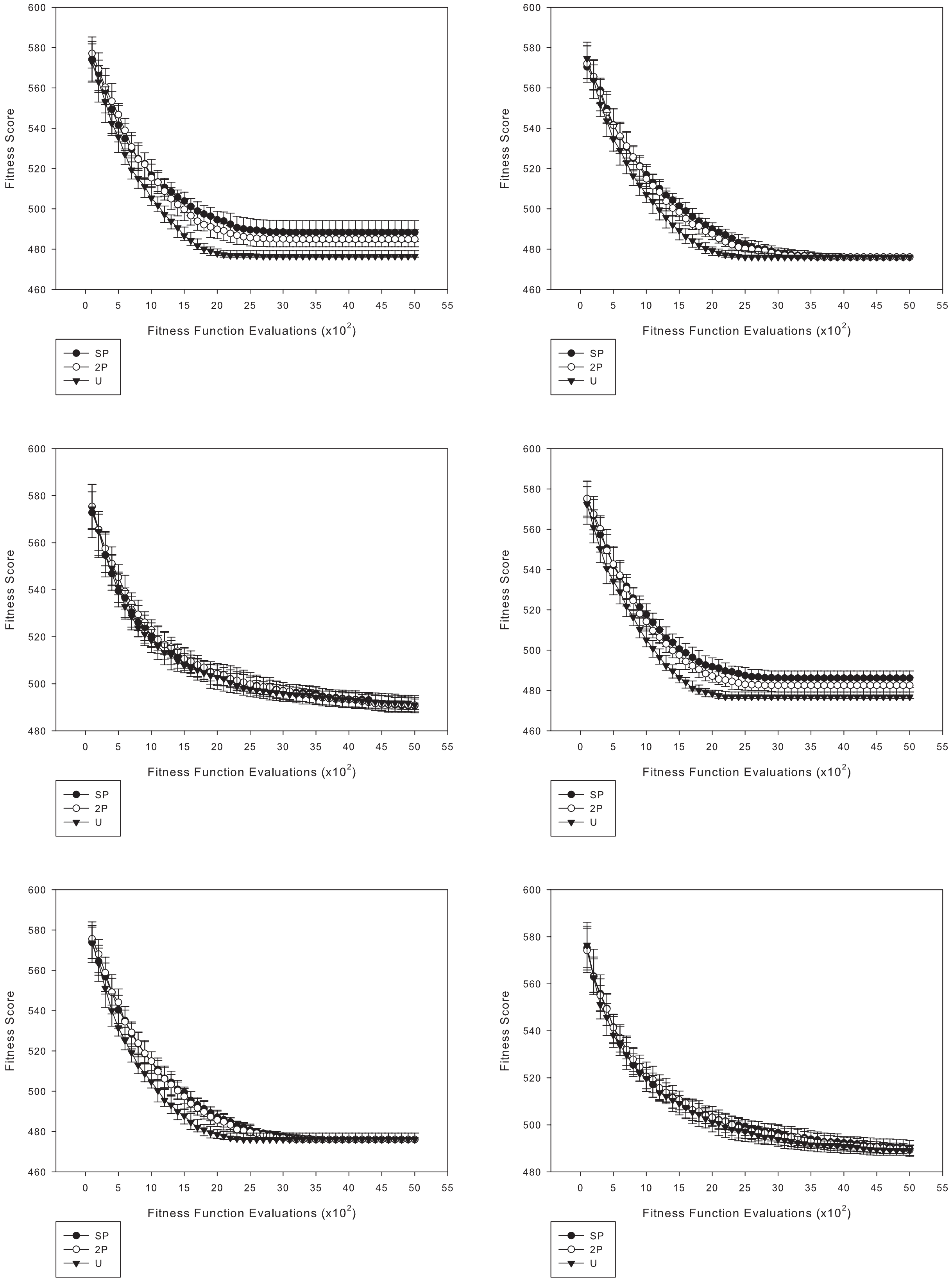


Figure B.9: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

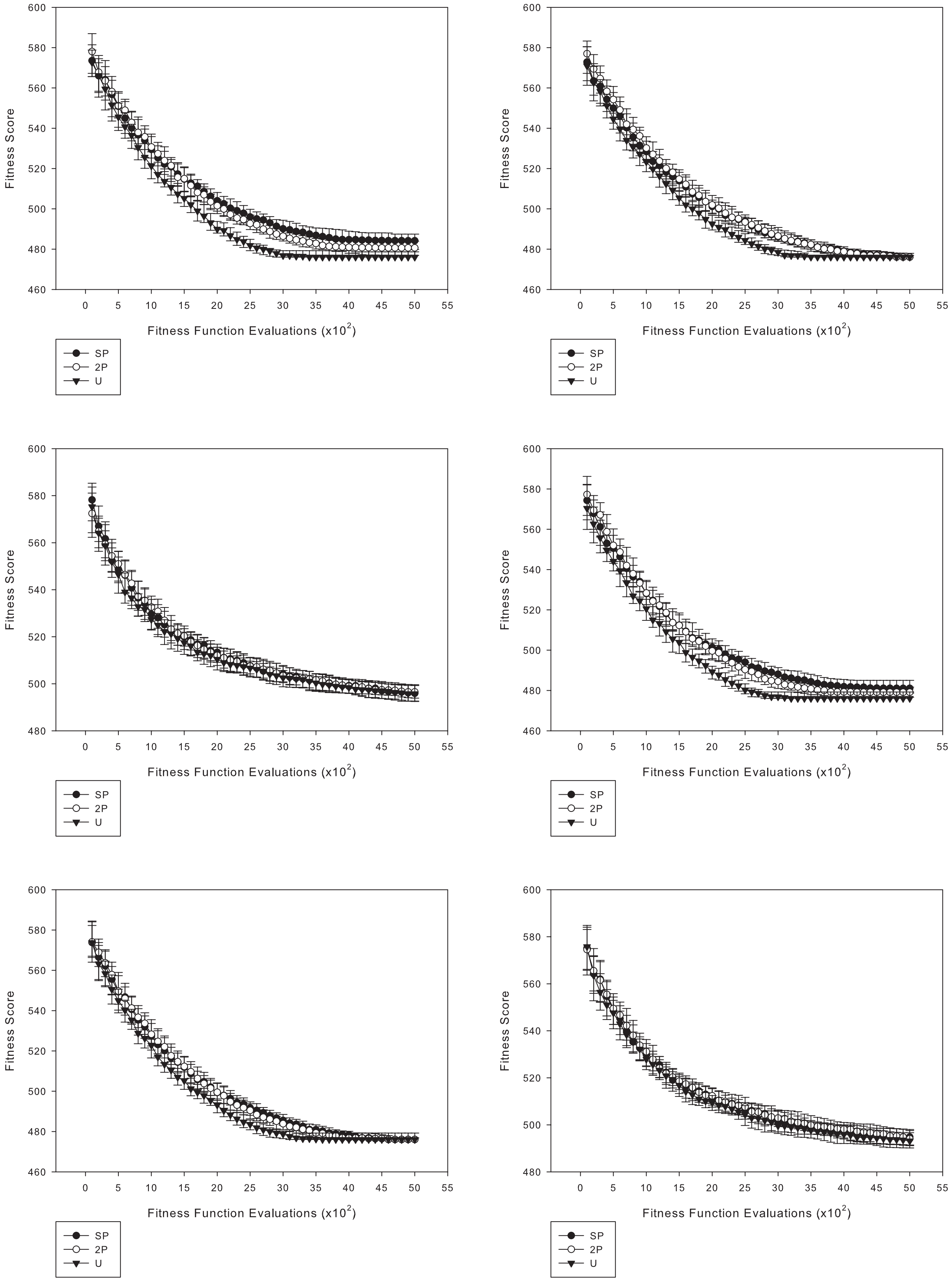
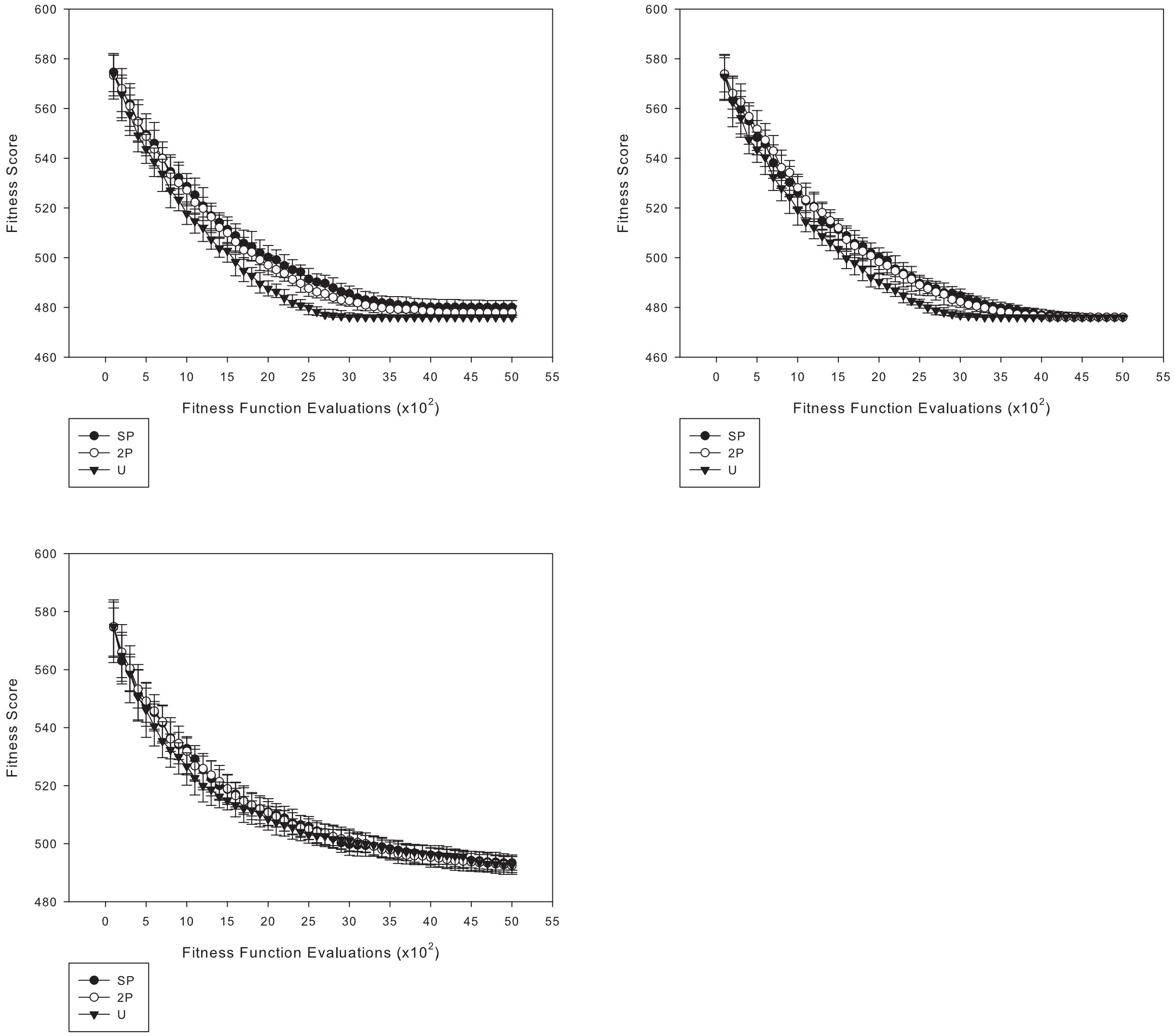


Figure B.10: $N = 150$, p_c 1.0, with p_m 0, 0.005



B.0.3 20 penalty points per intervention

Figure B.11: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

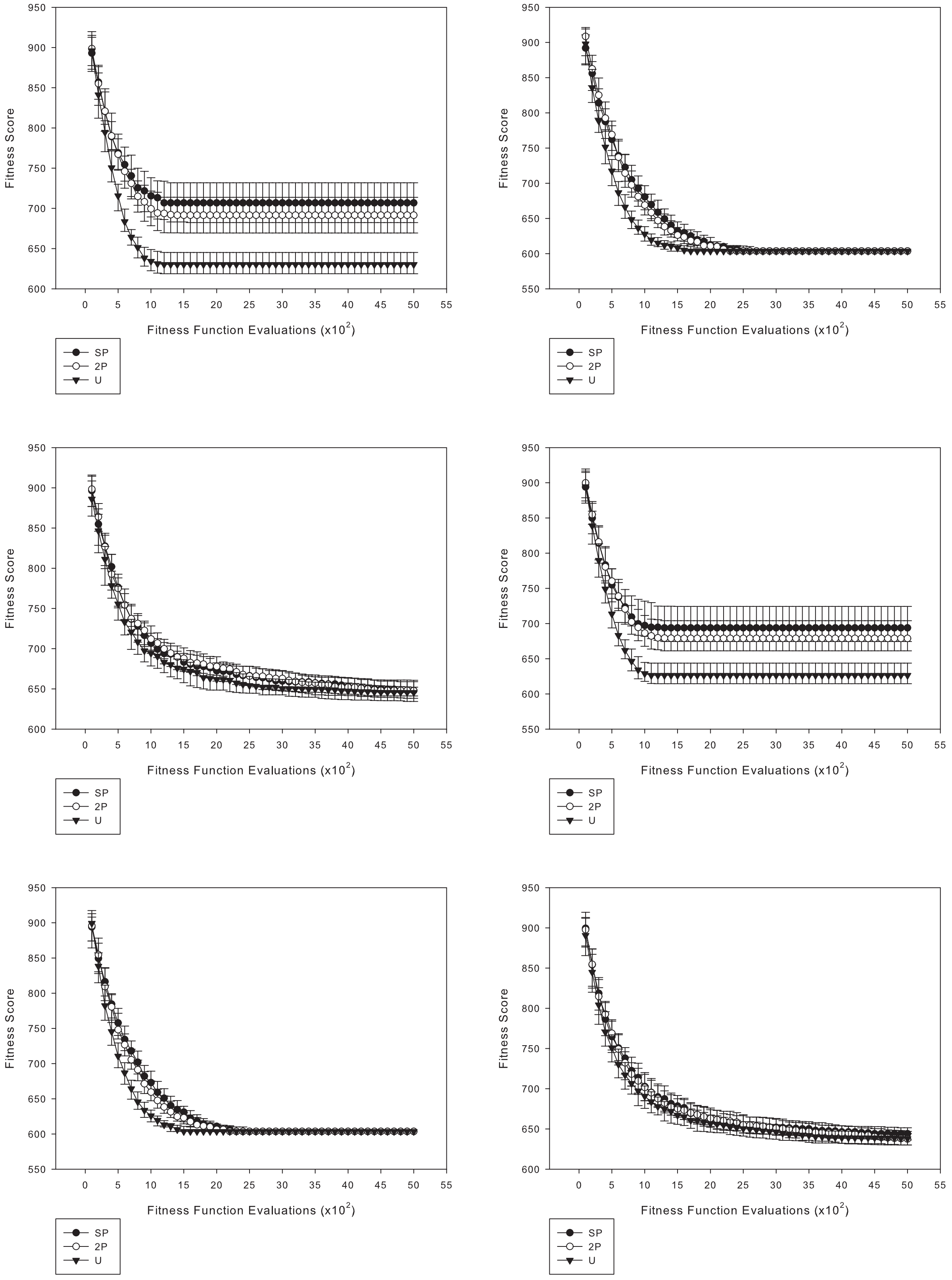


Figure B.12: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

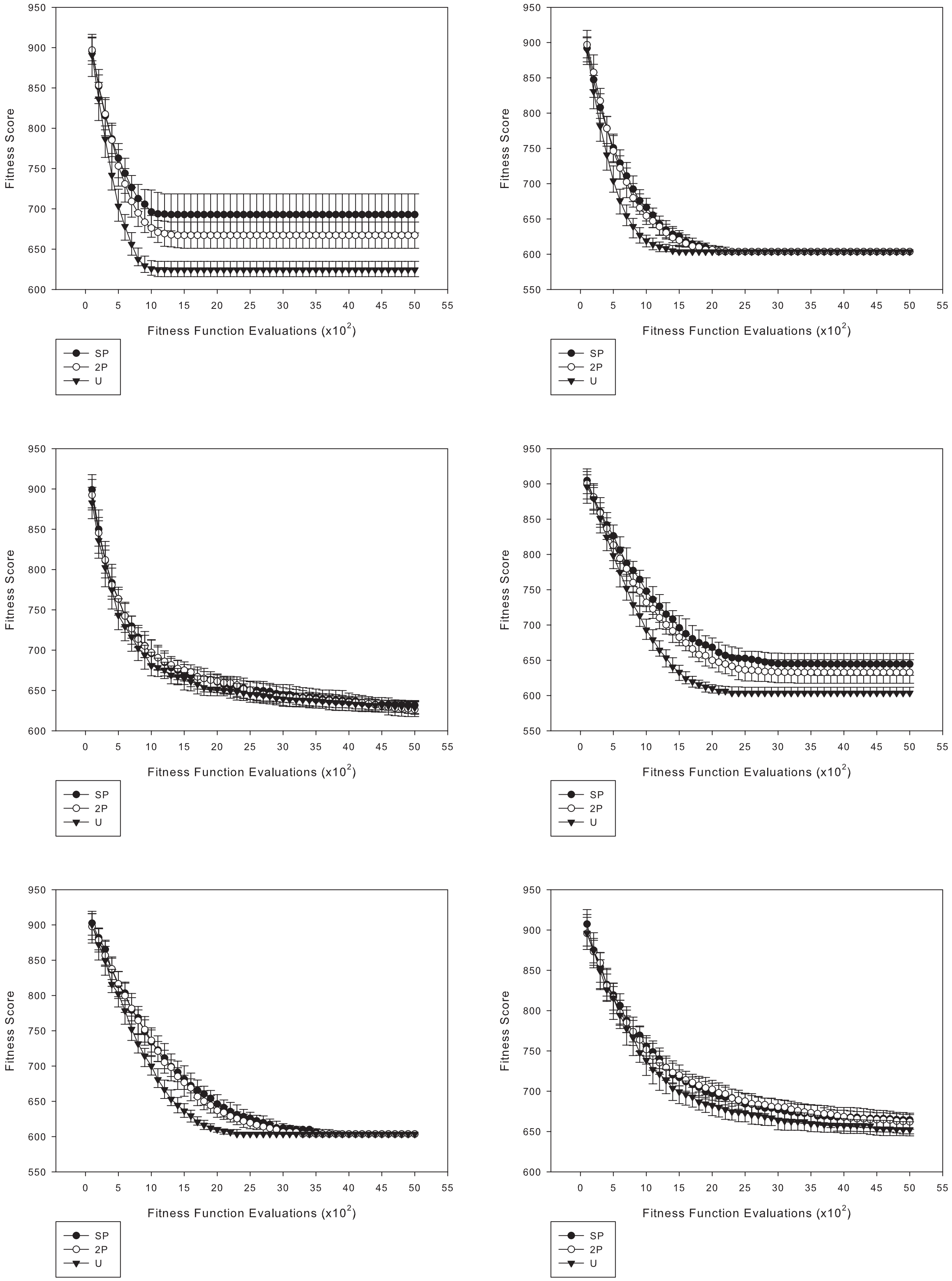


Figure B.13: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

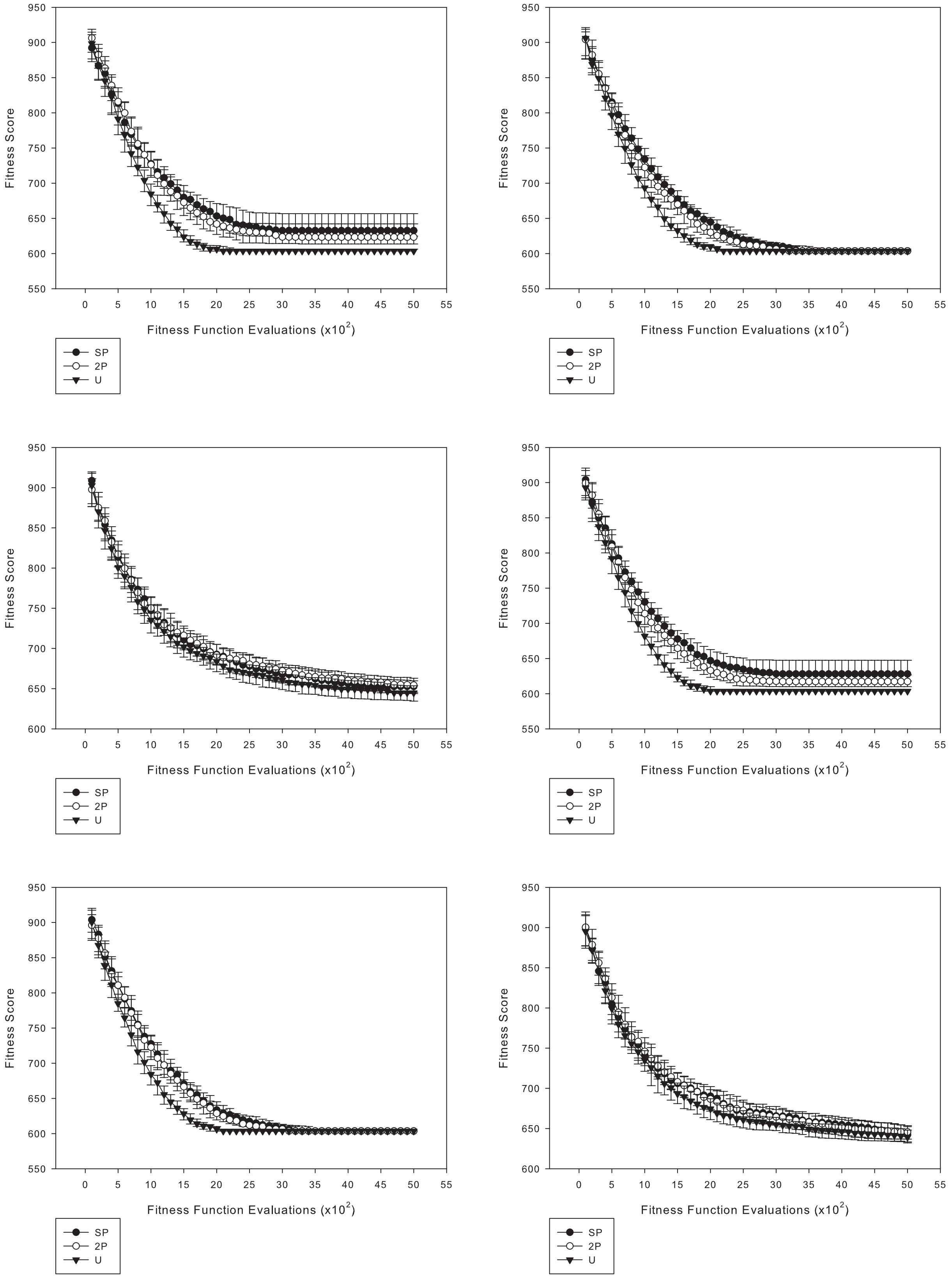


Figure B.14: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

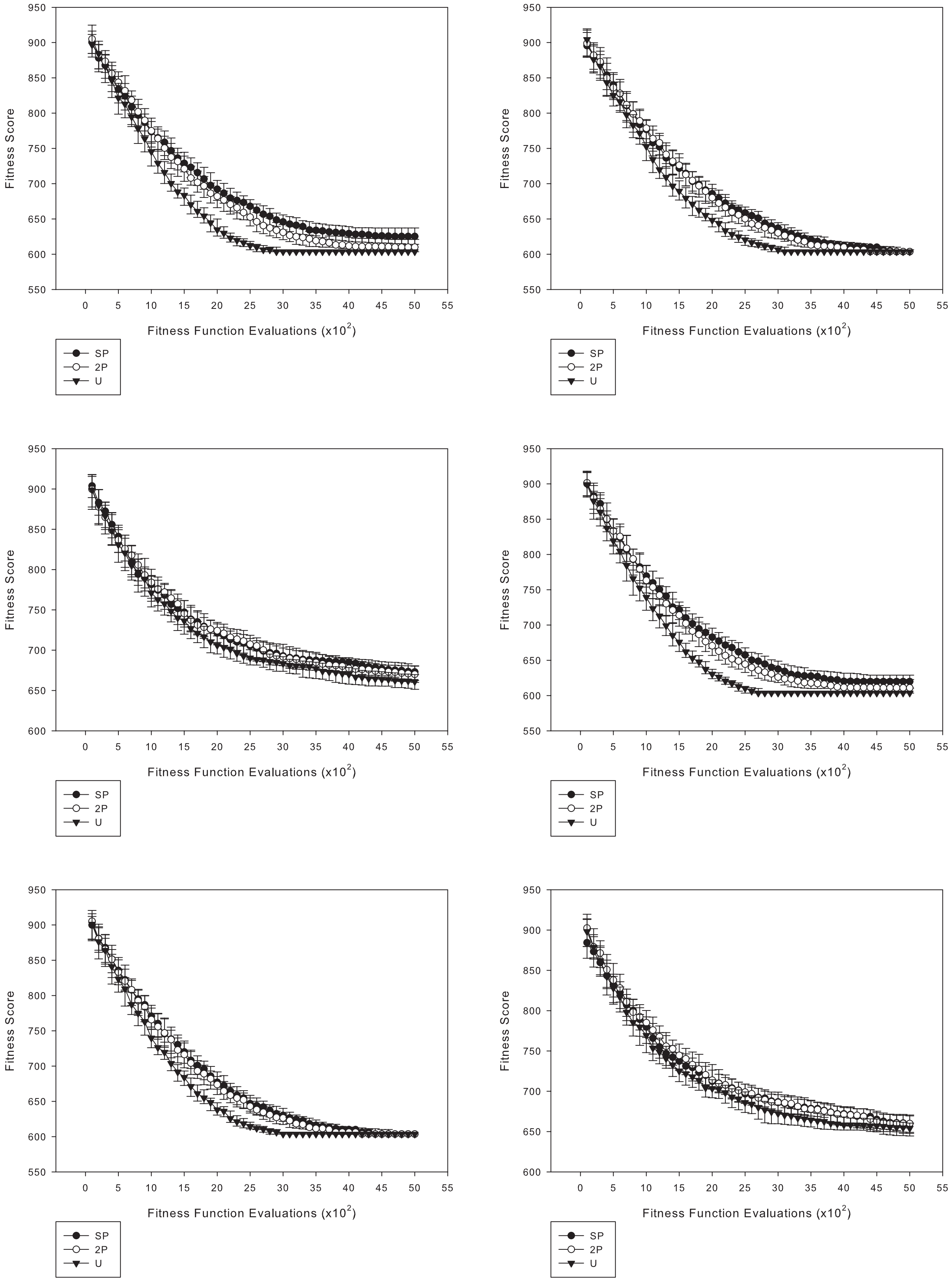
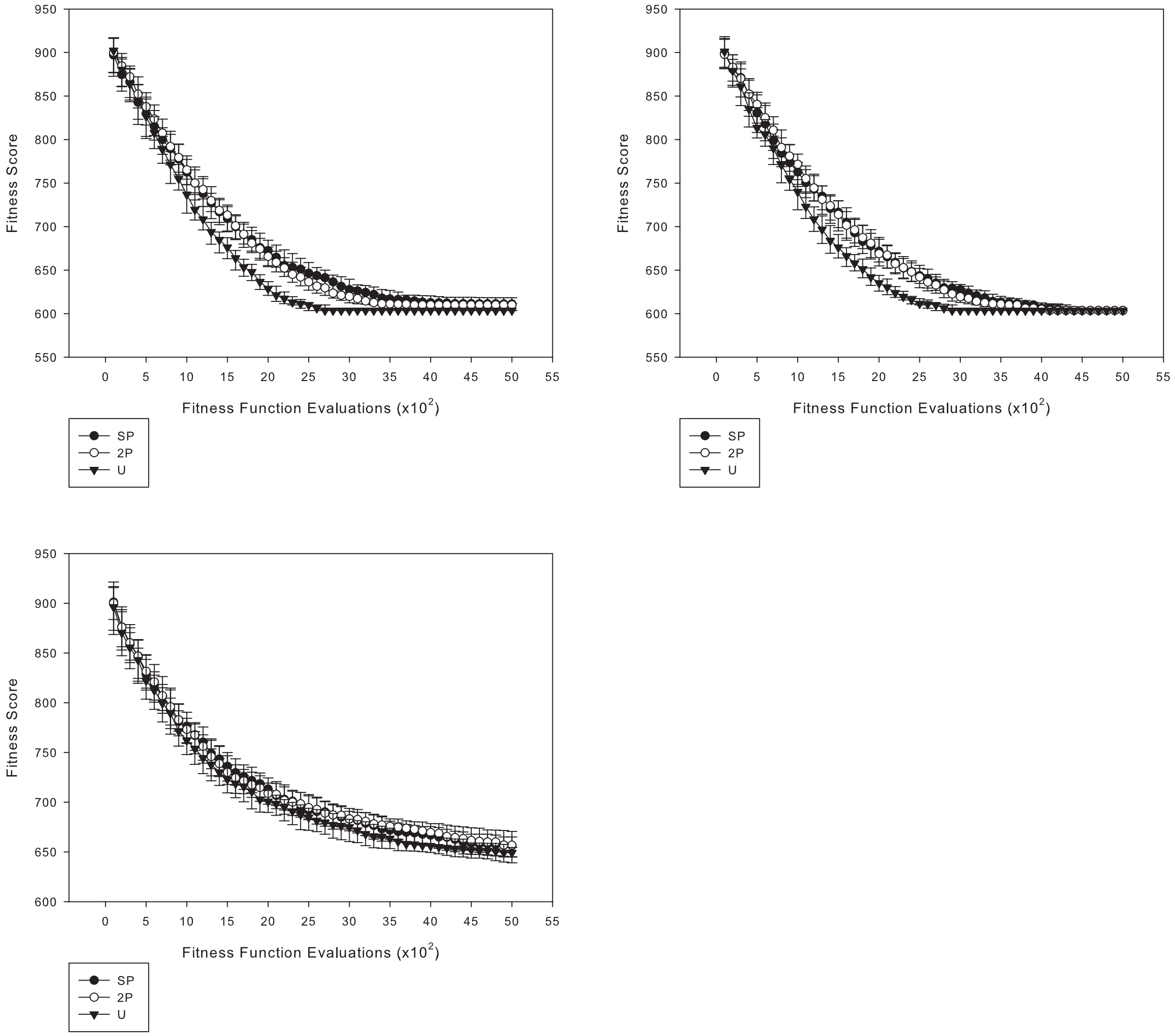


Figure B.15: $N = 150$, p_c 1.0, with p_m 0, 0.005



B.0.4 35 penalty points per intervention

Figure B.16: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

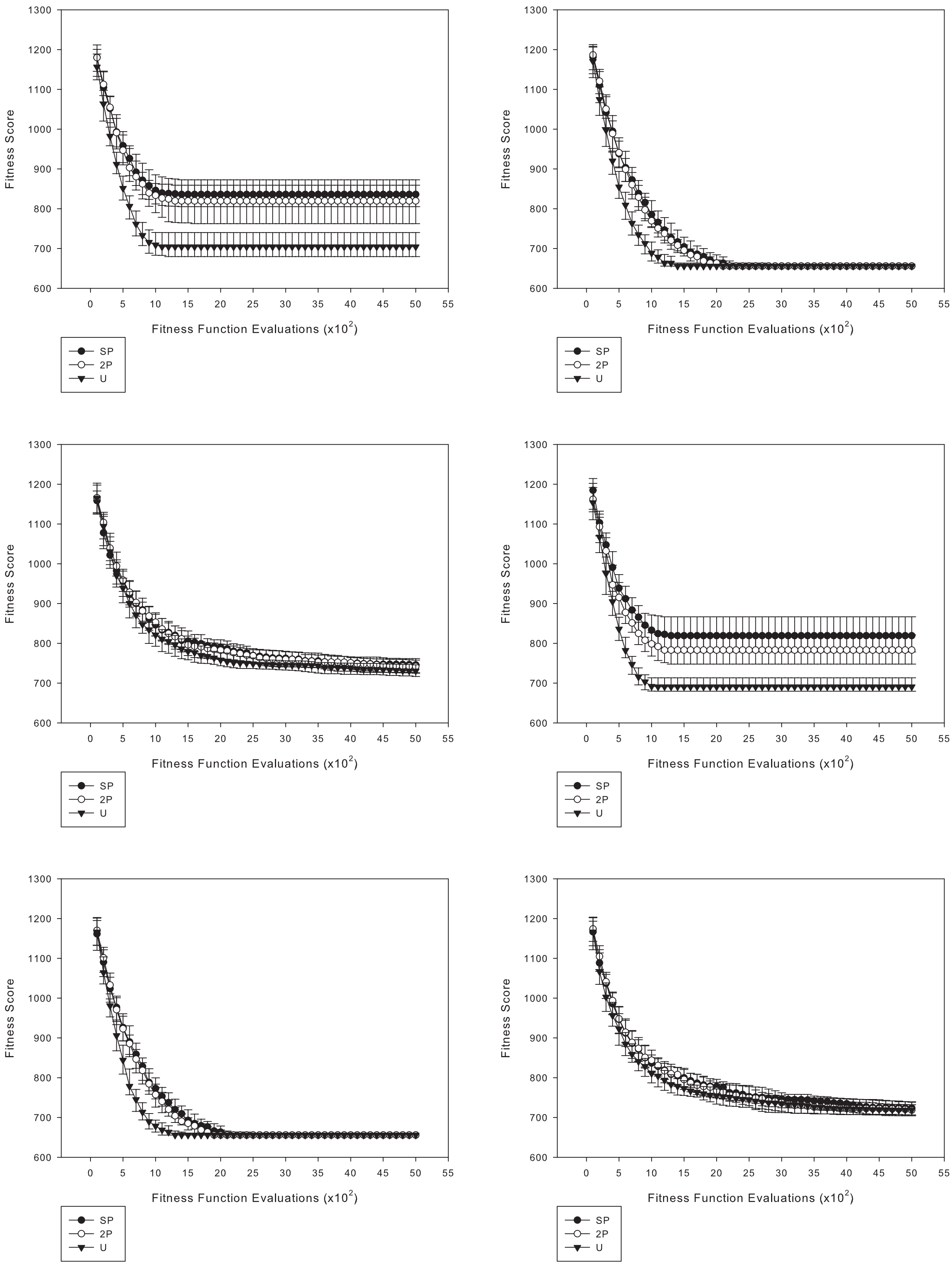


Figure B.17: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

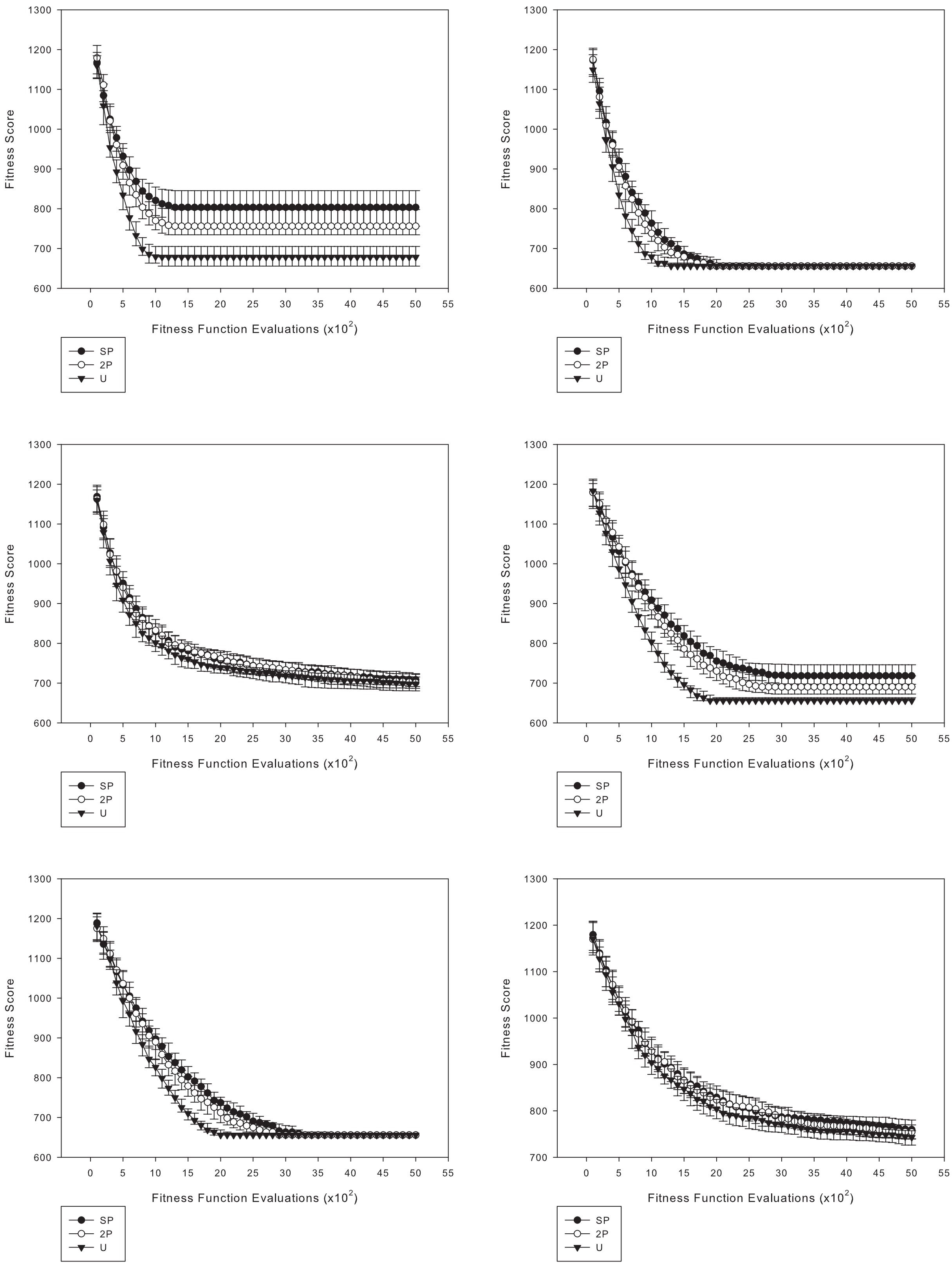


Figure B.18: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

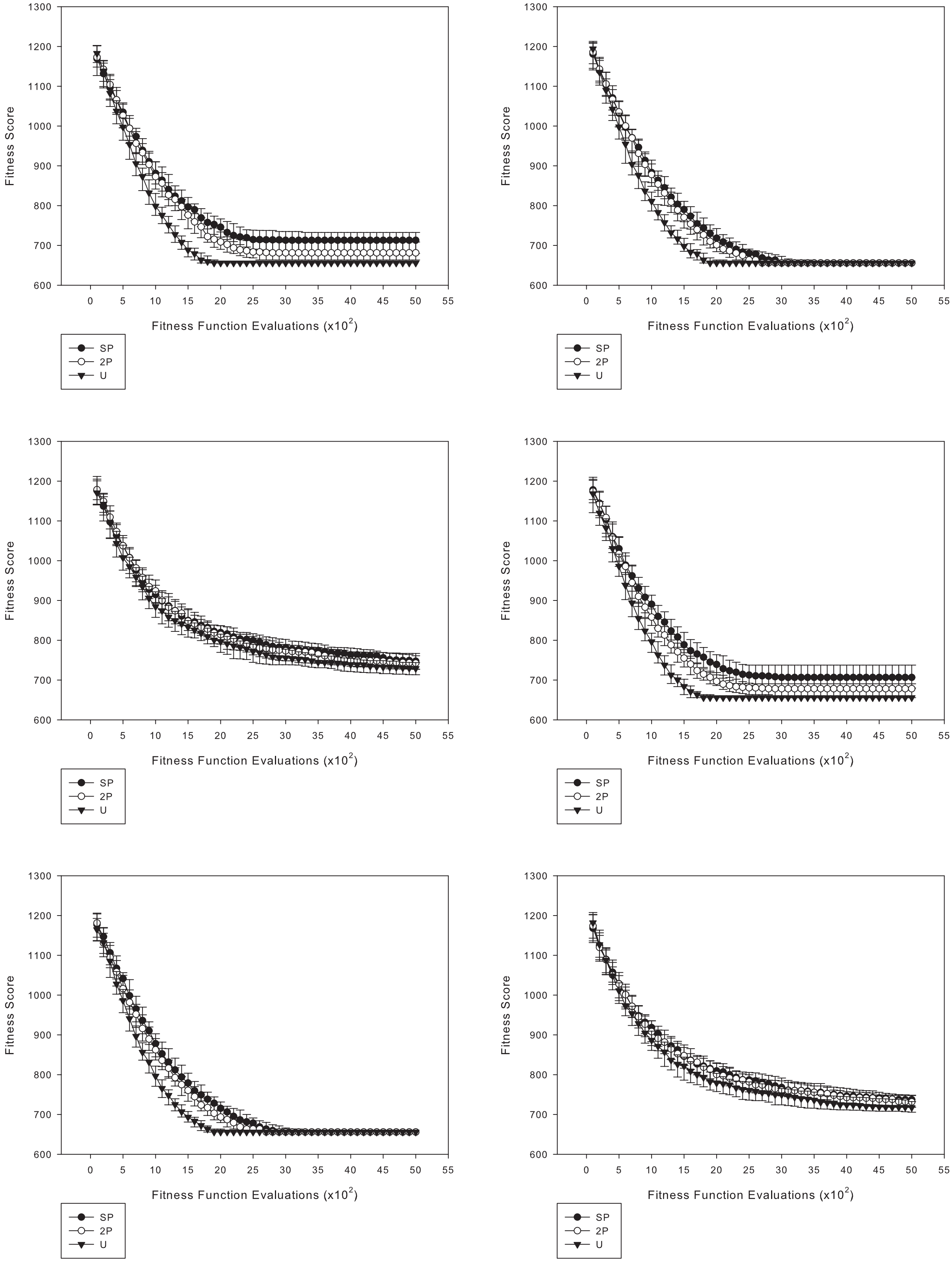


Figure B.19: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

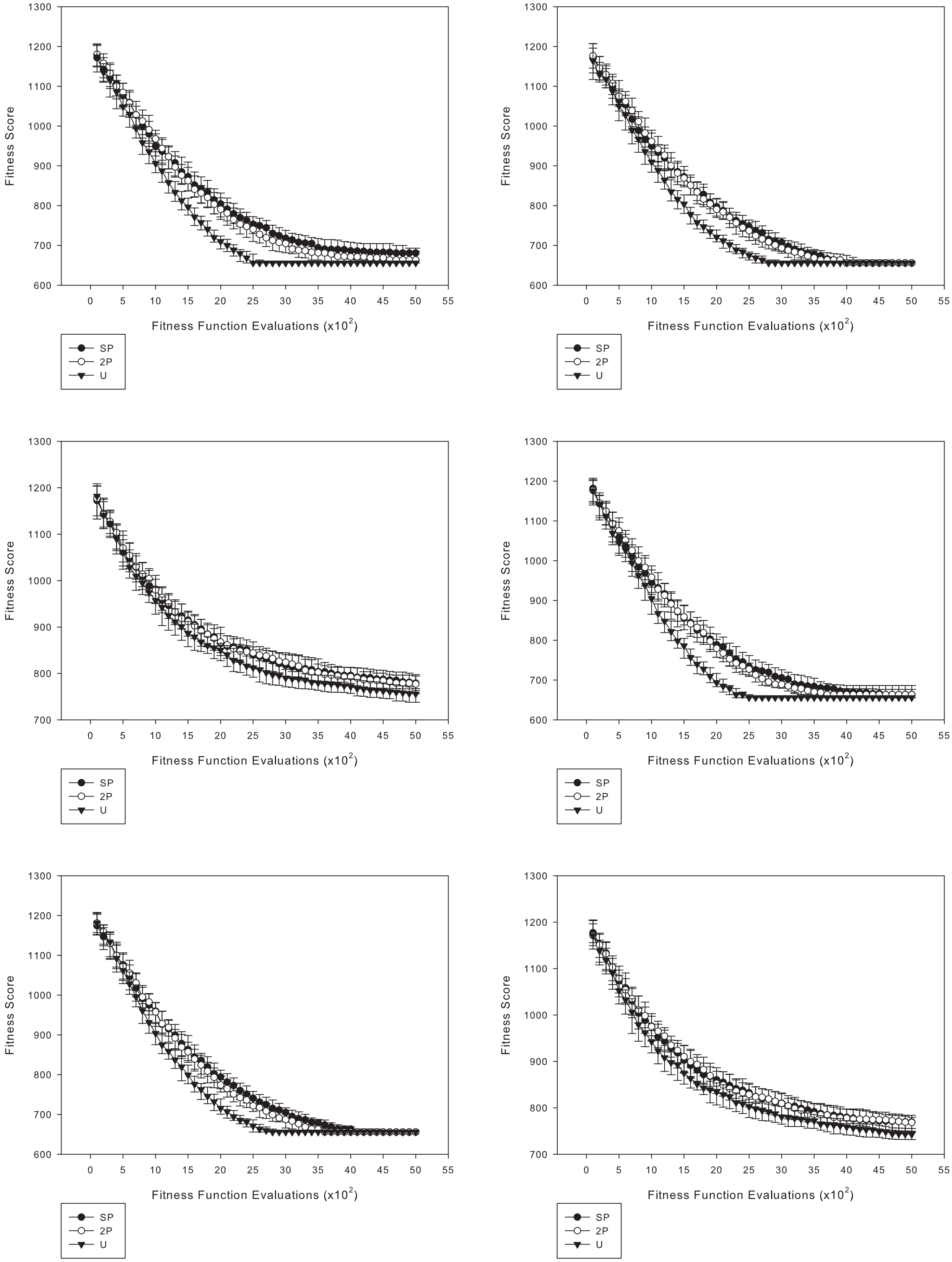
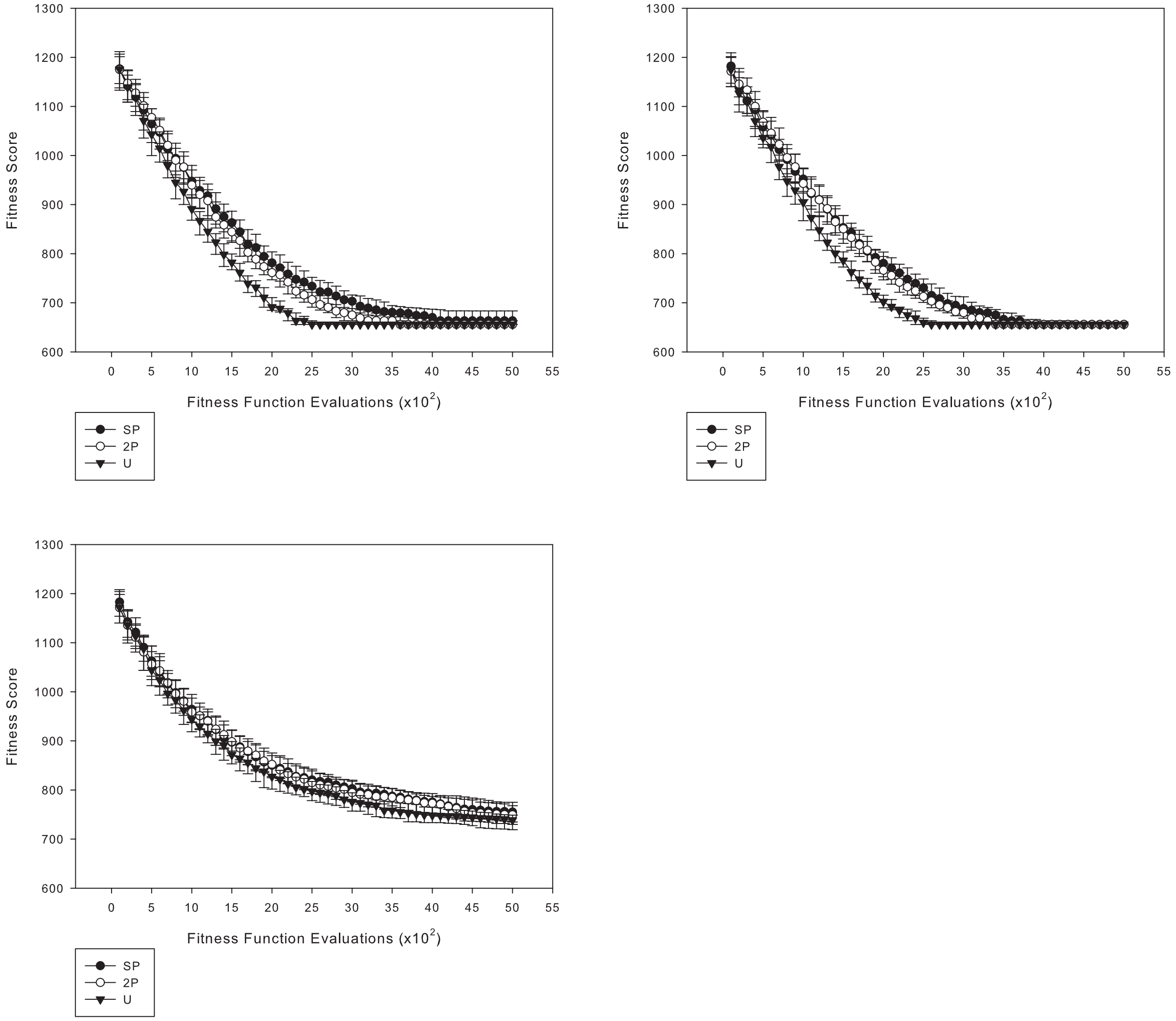


Figure B.20: $N = 150$, p_c 1.0, with p_m 0, 0.005



B.0.5 50 penalty points per intervention

Figure B.21: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

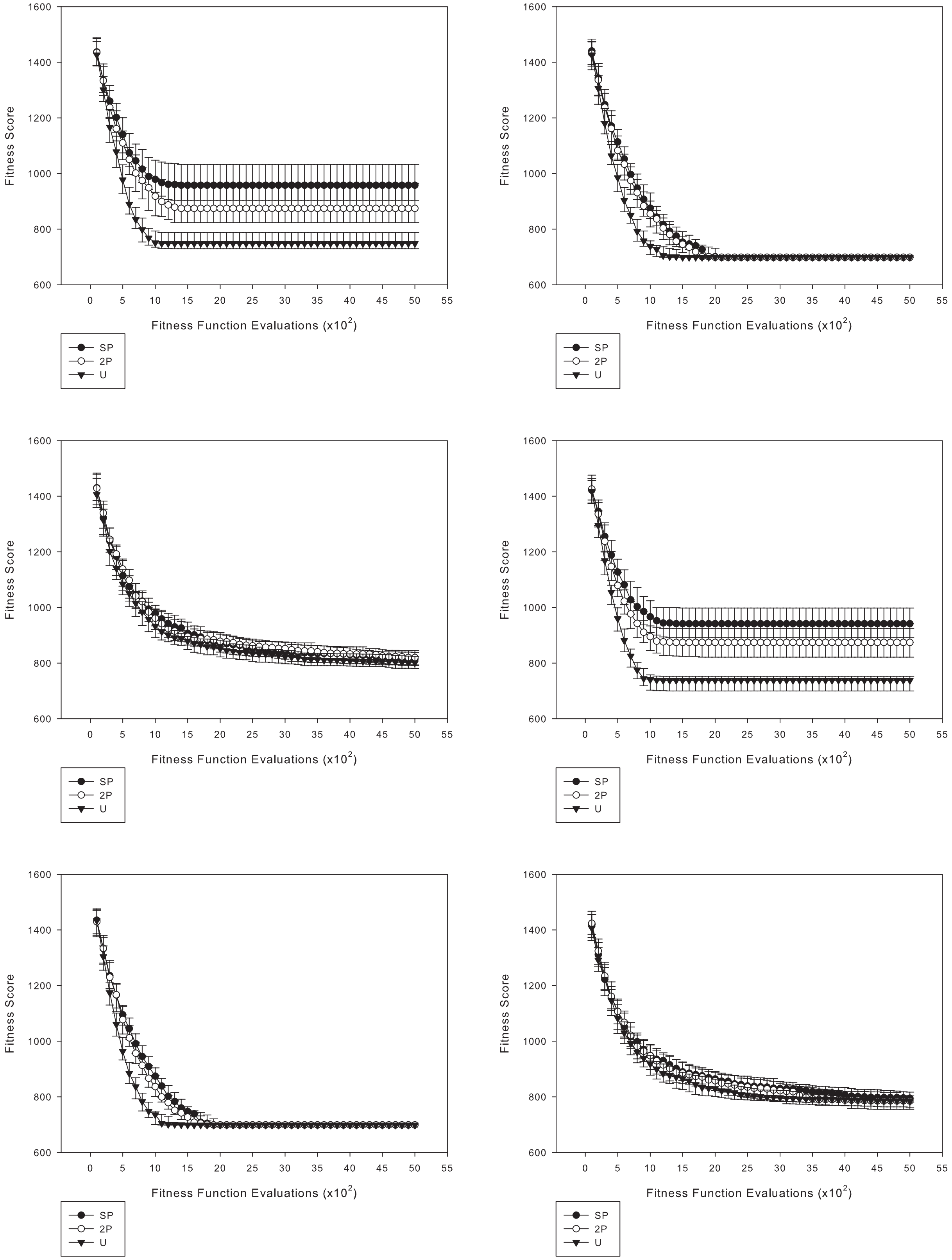


Figure B.22: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

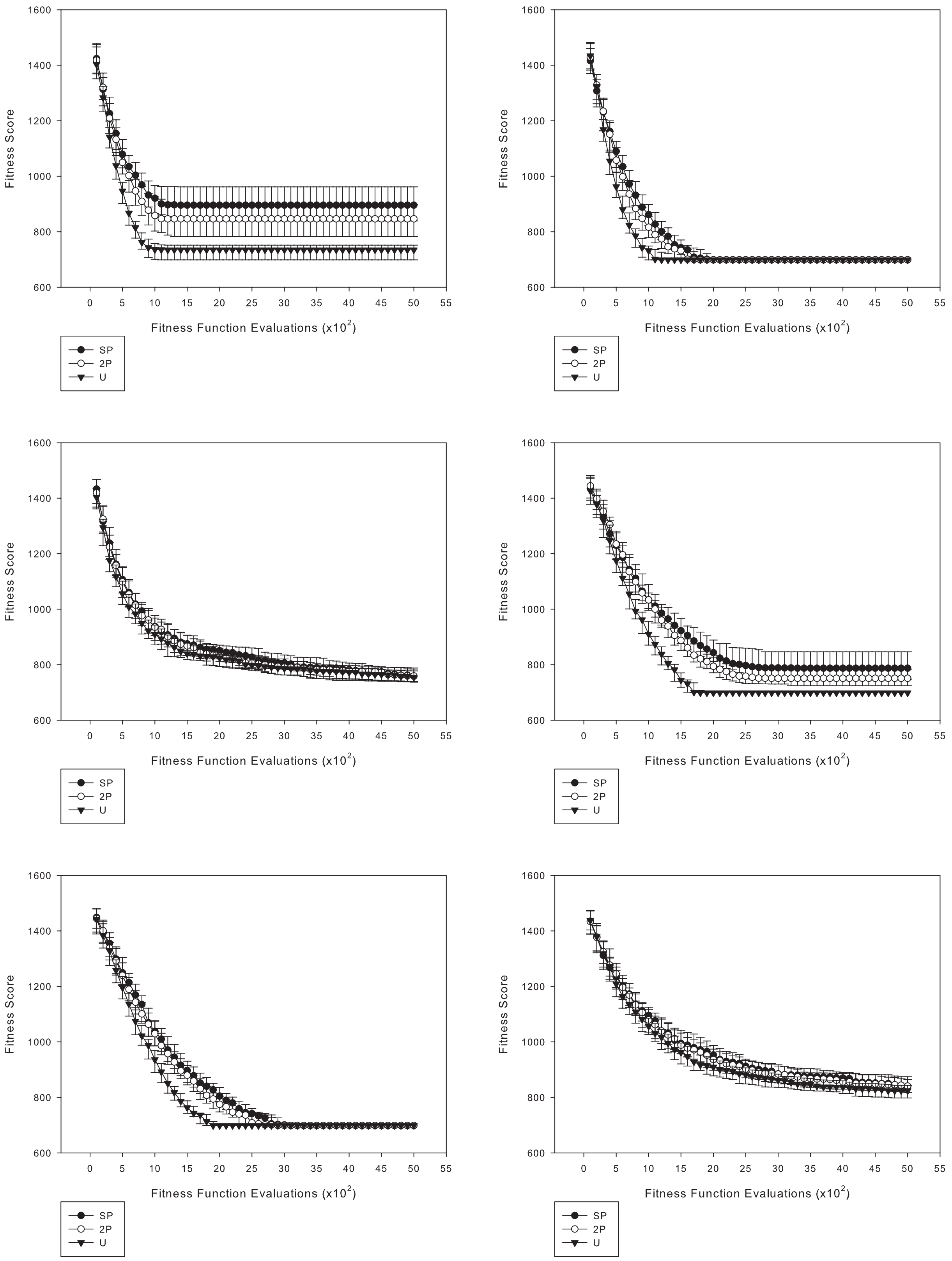


Figure B.23: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

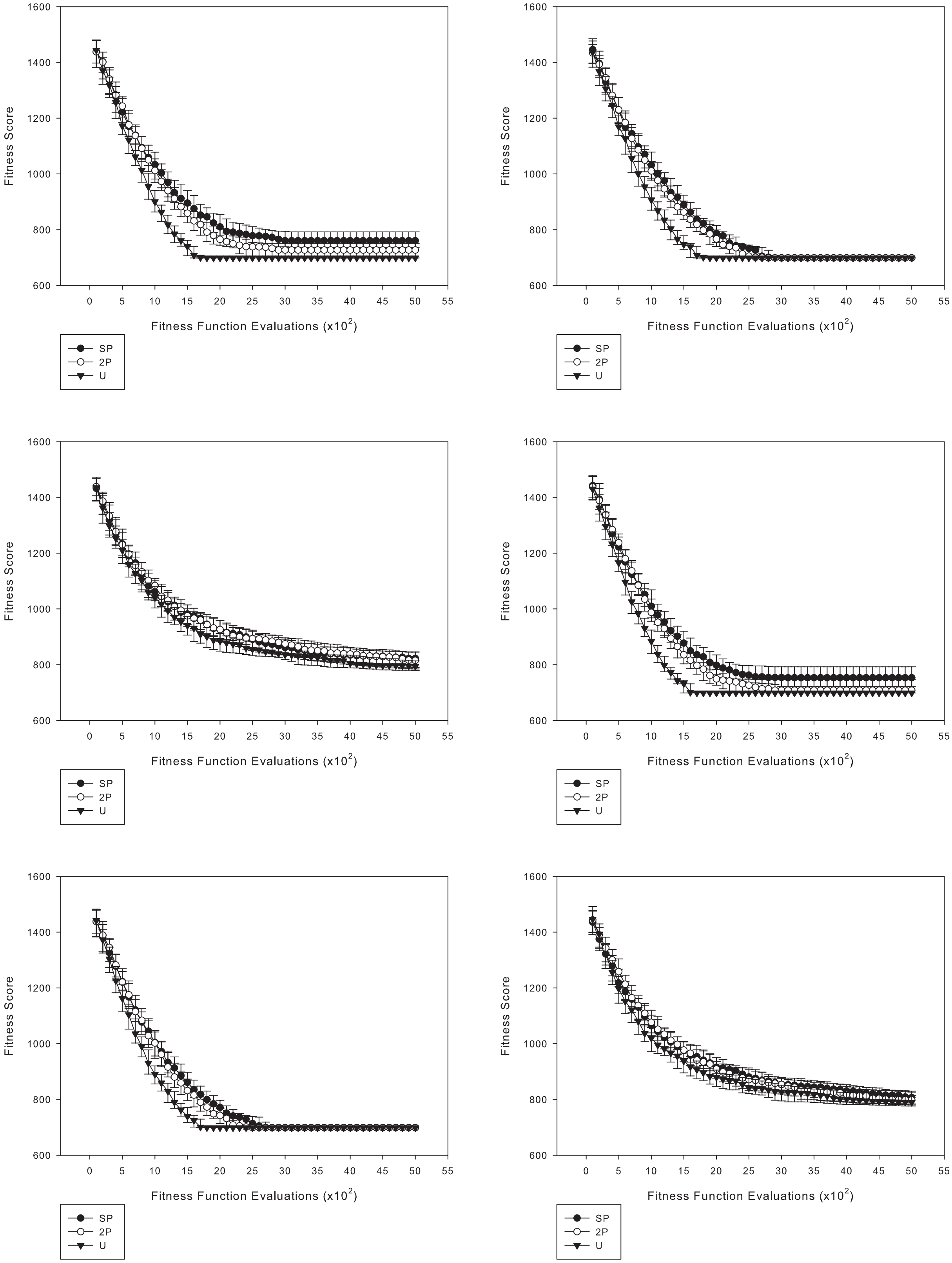


Figure B.24: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

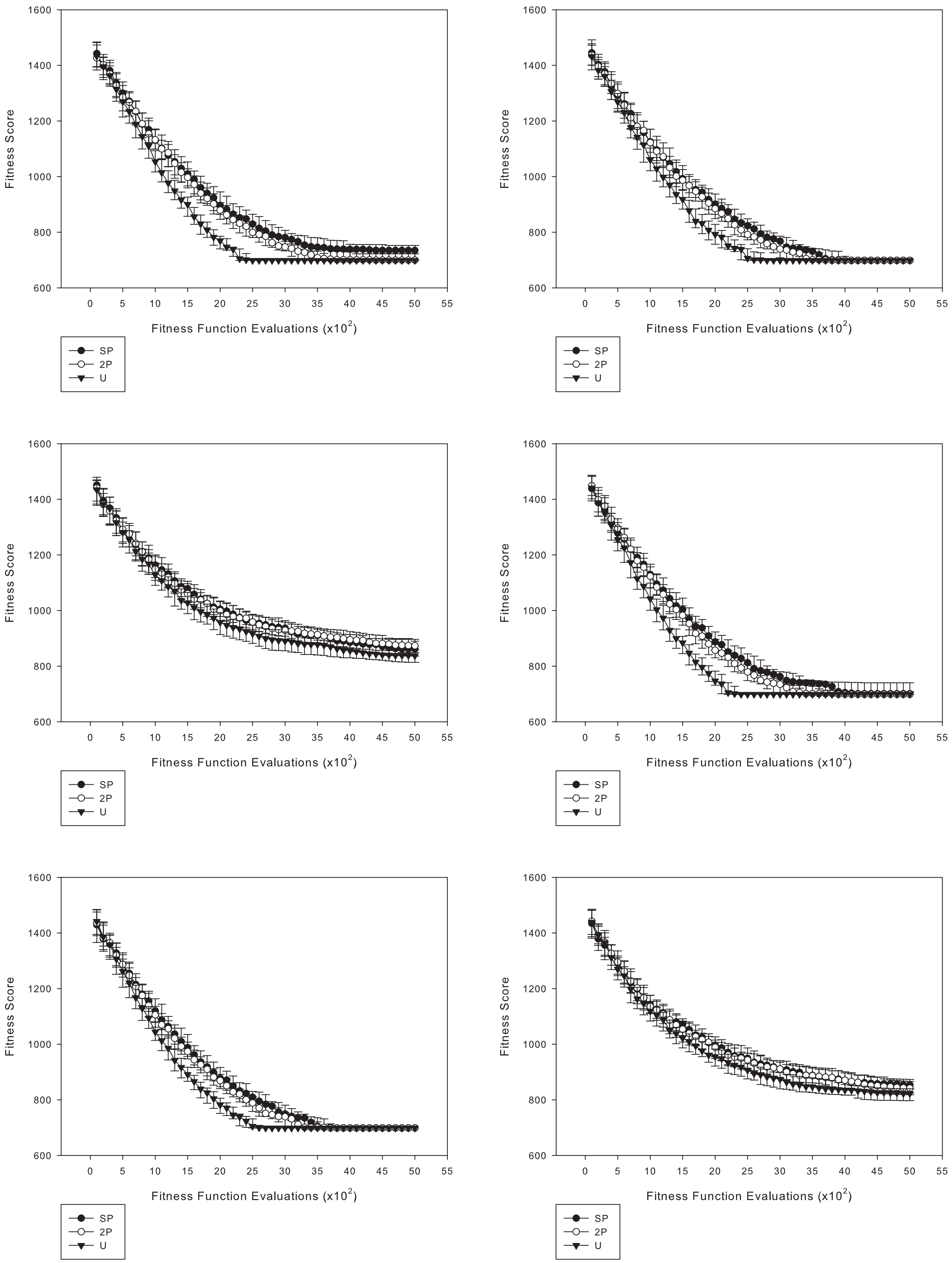
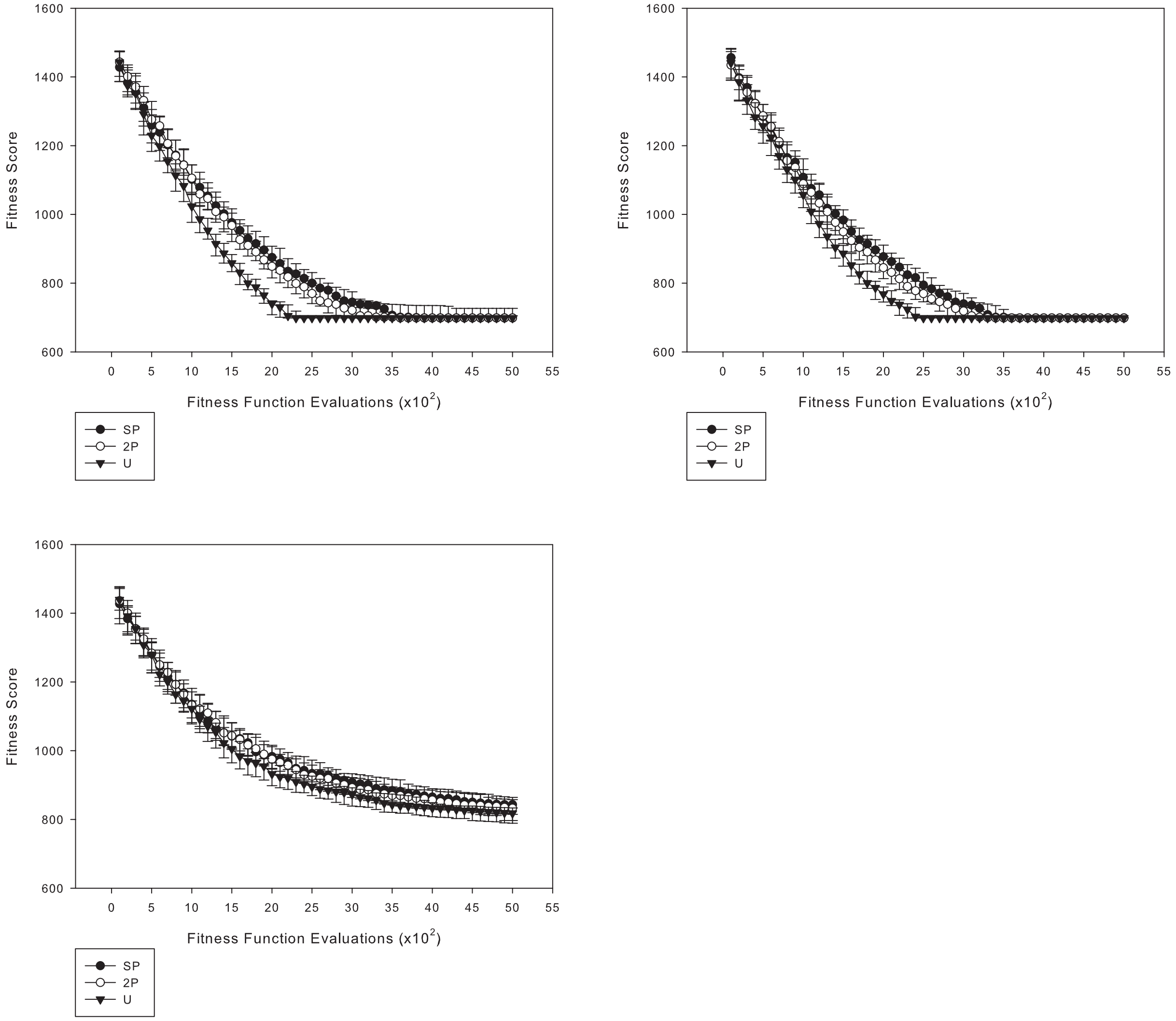


Figure B.25: $N = 150$, p_c 1.0, with p_m 0, 0.005



Appendix C

Bio-control scheduling graphs for UC, CalEB, TInSSel and DUC

C.0.6 0 penalty points per intervention

Figure C.1: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

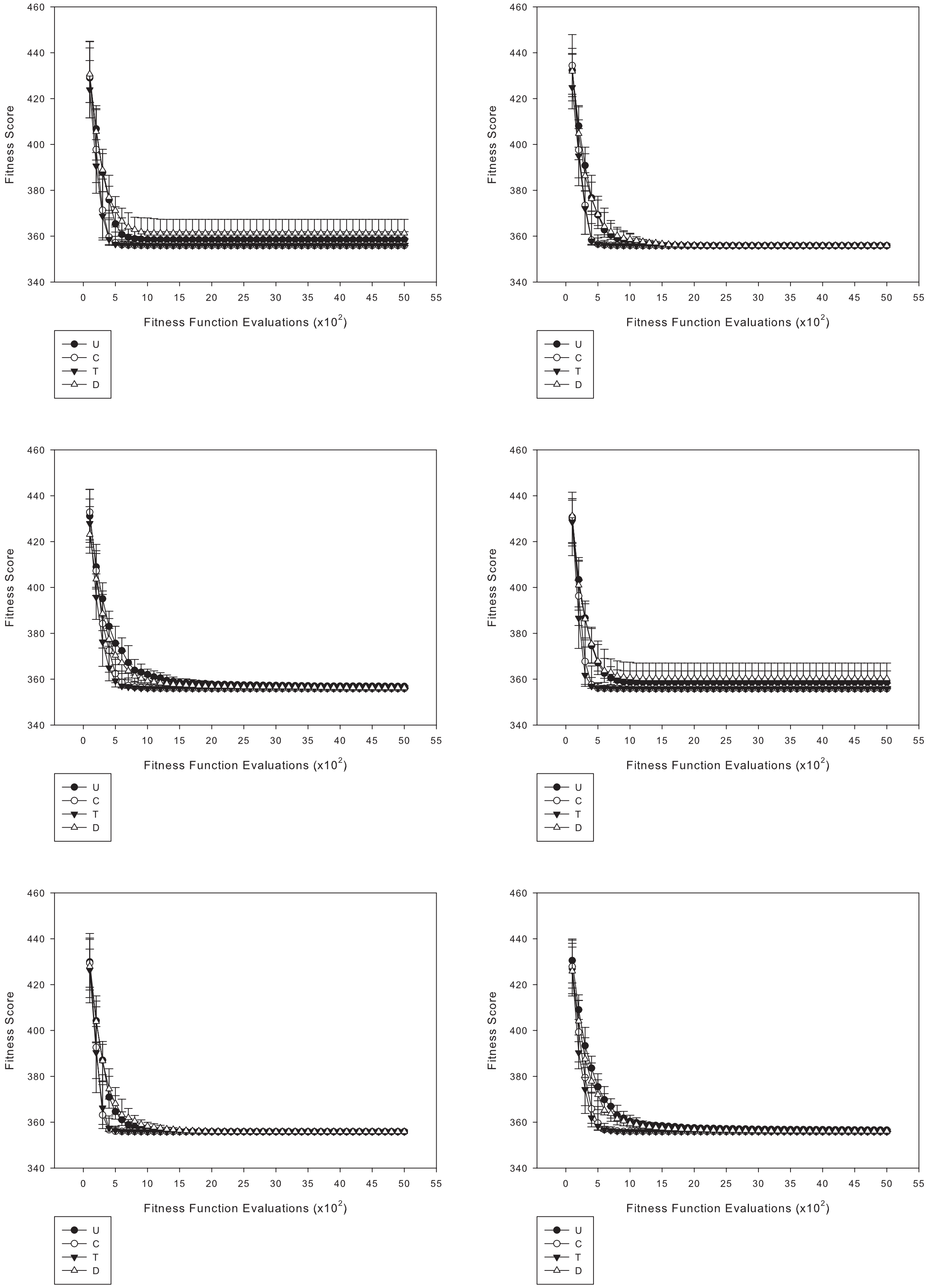


Figure C.2: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

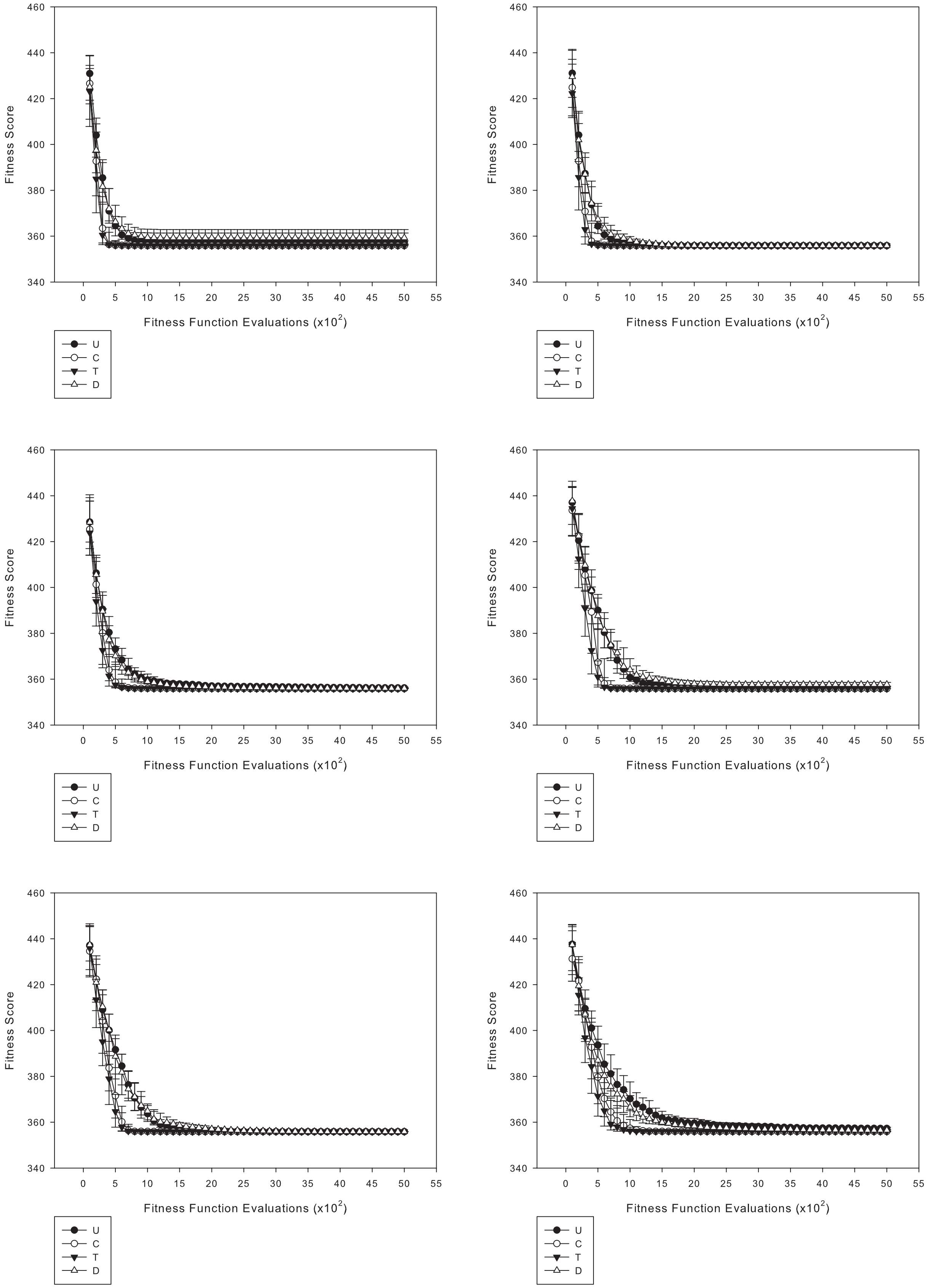


Figure C.3: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

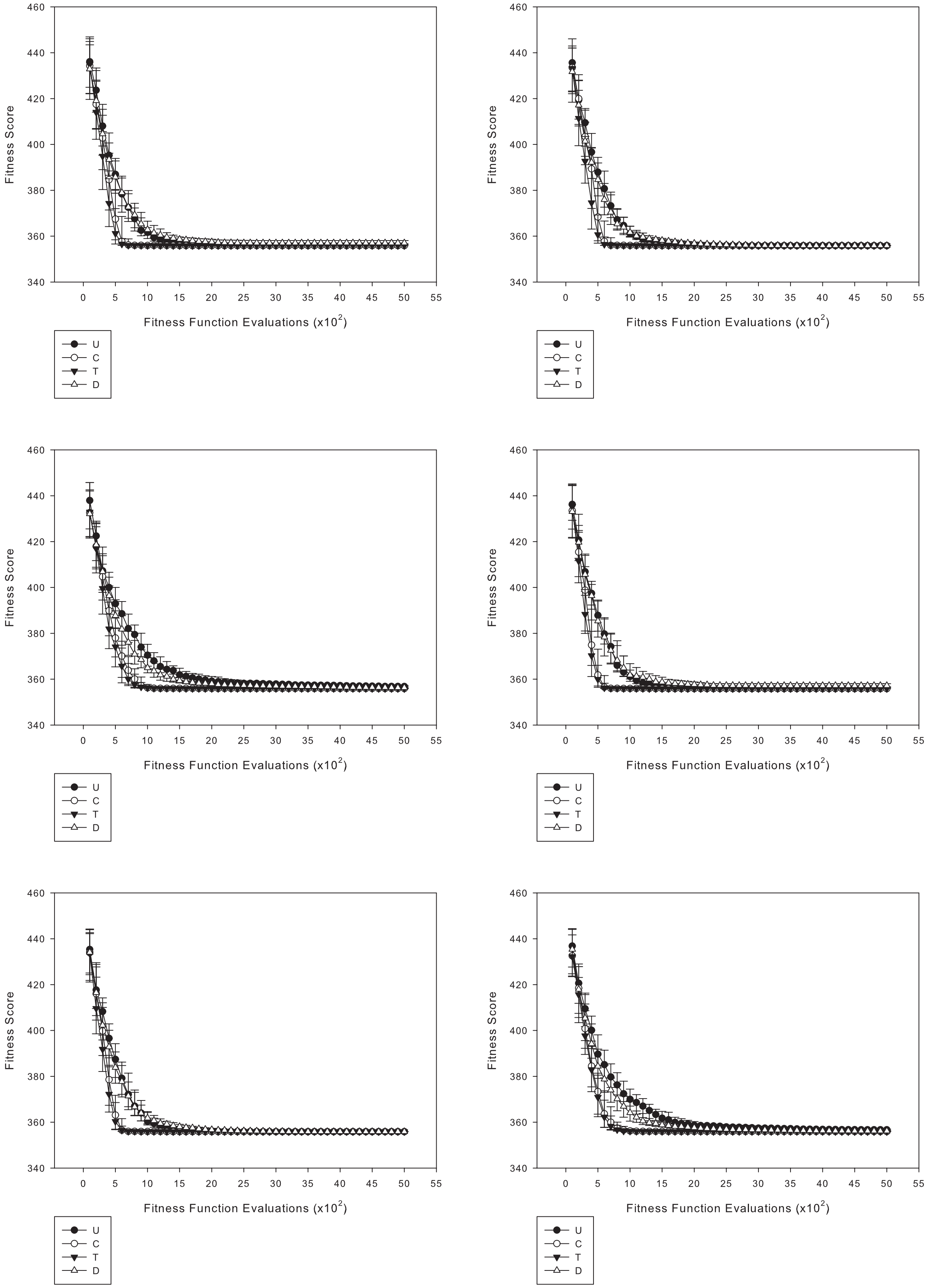


Figure C.4: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

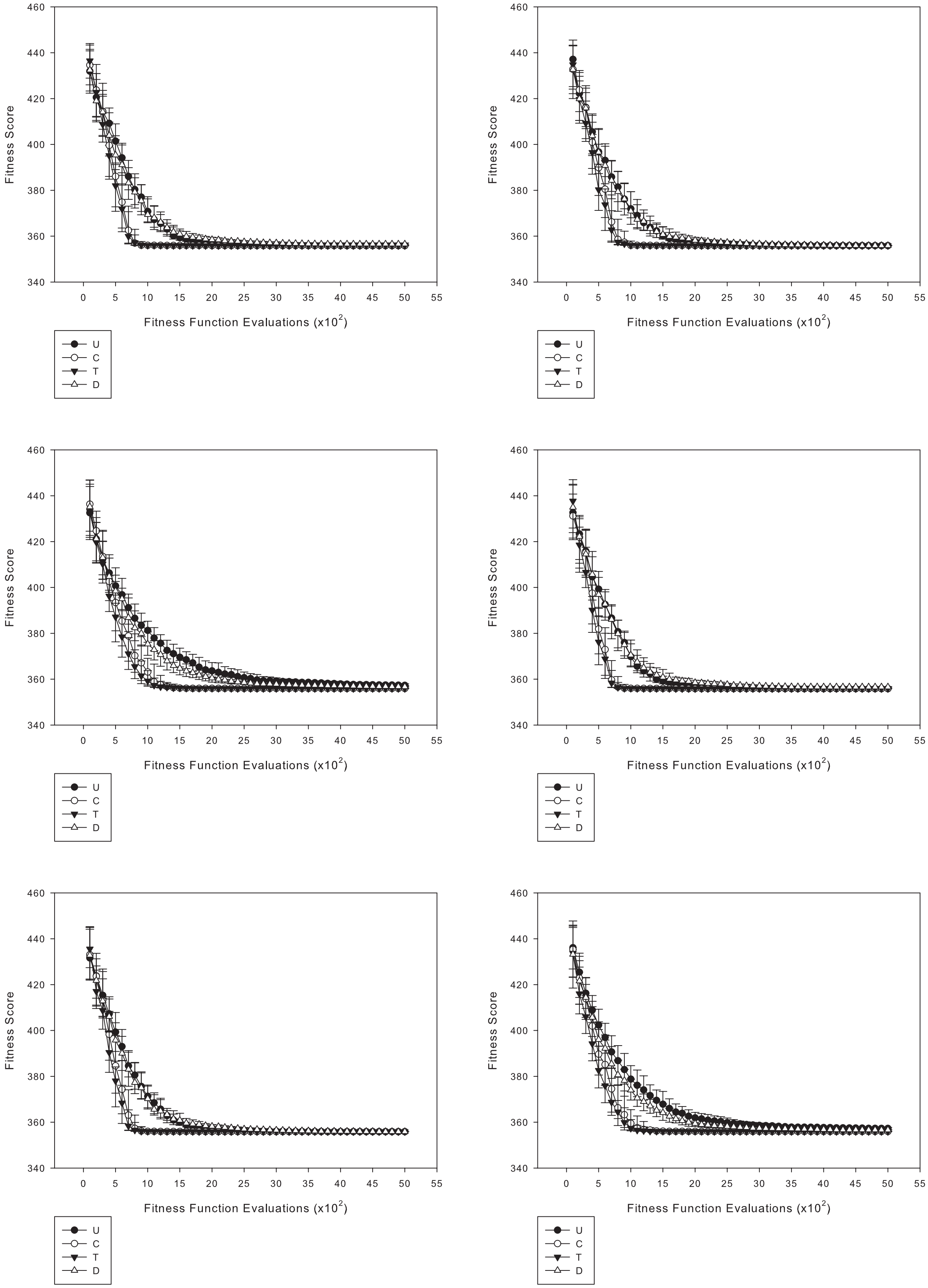
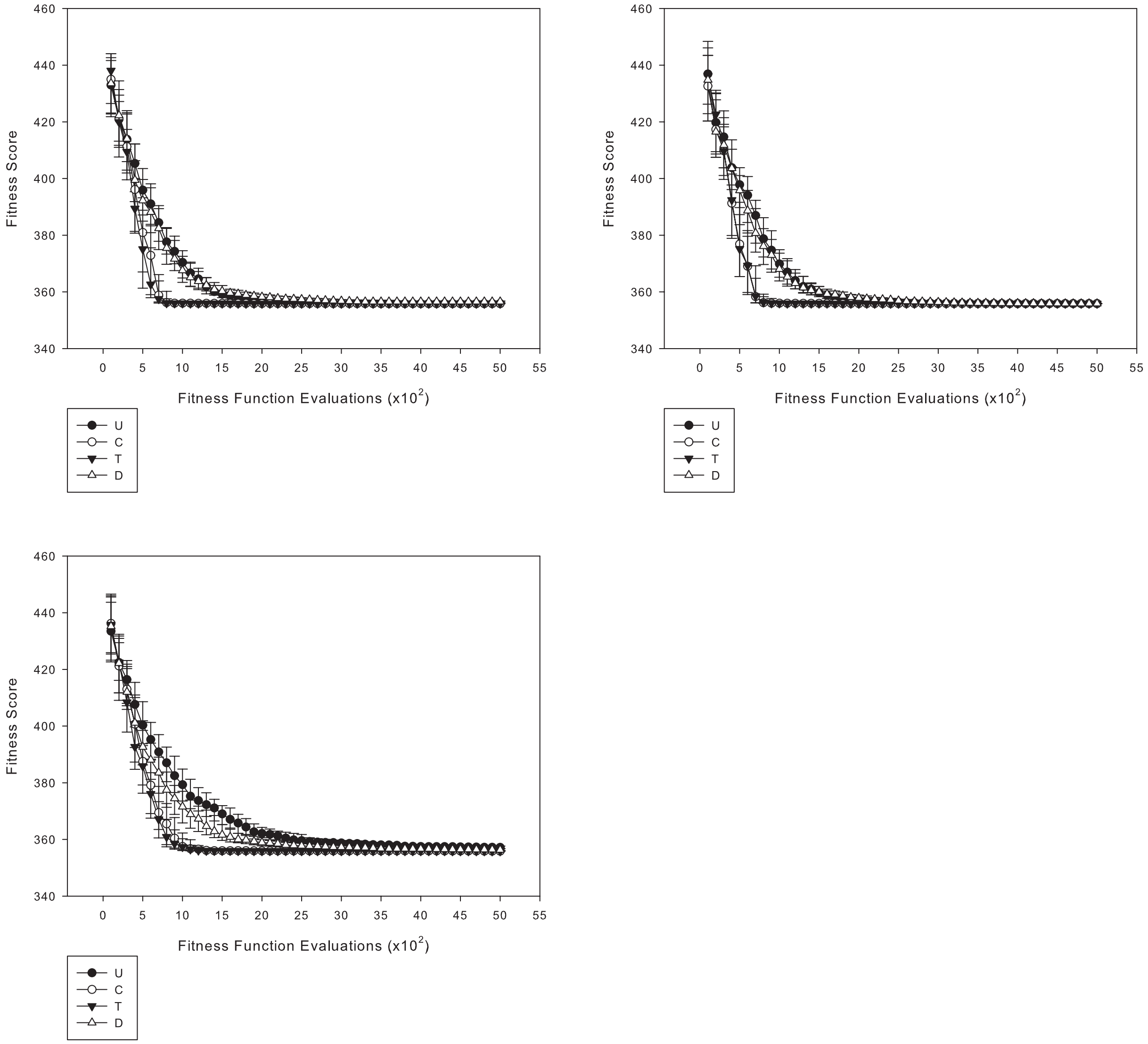


Figure C.5: $N = 150$, p_c 1.0, with p_m 0, 0.005



C.0.7 5 penalty points per intervention

Figure C.6: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

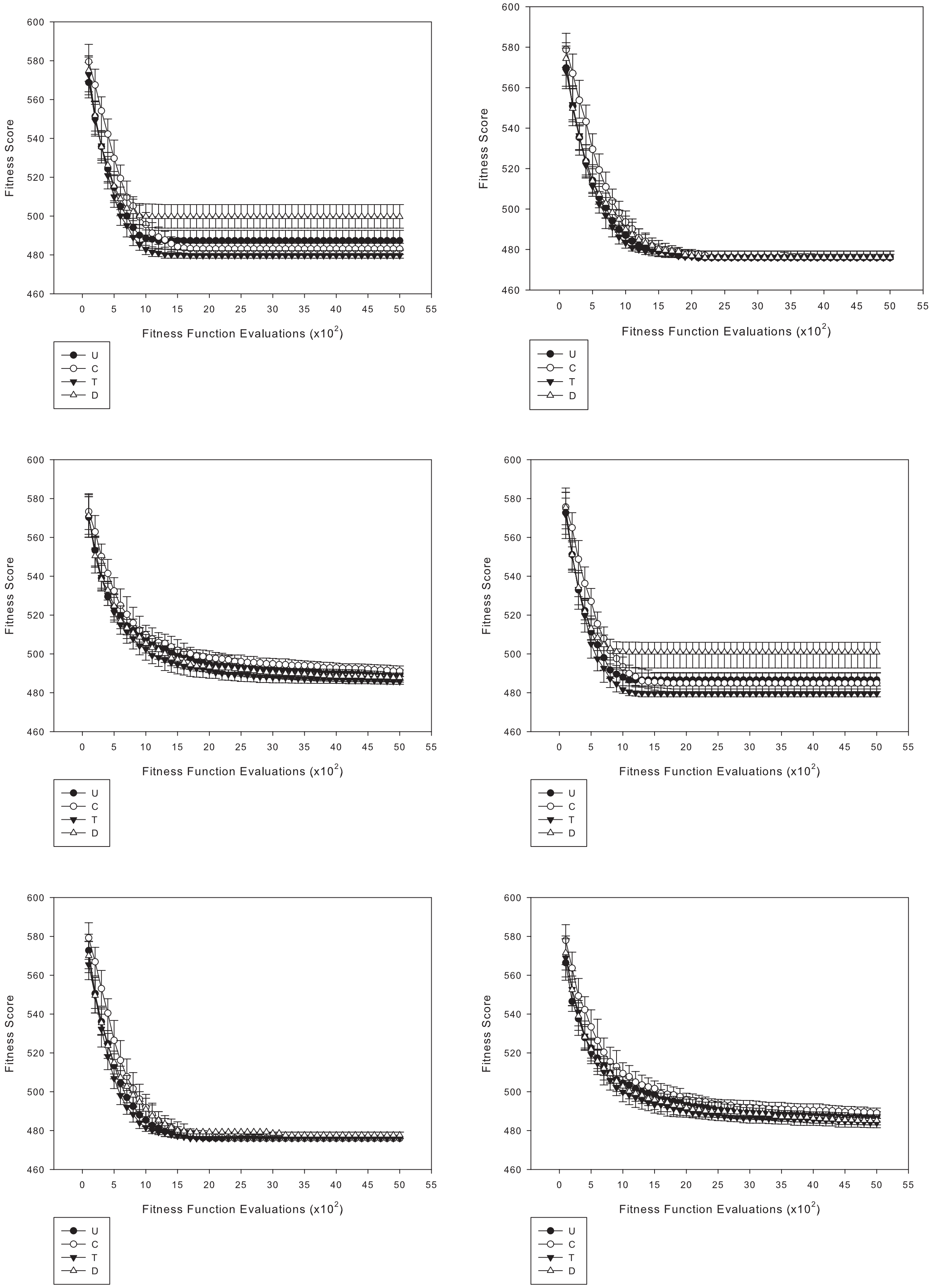


Figure C.7: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

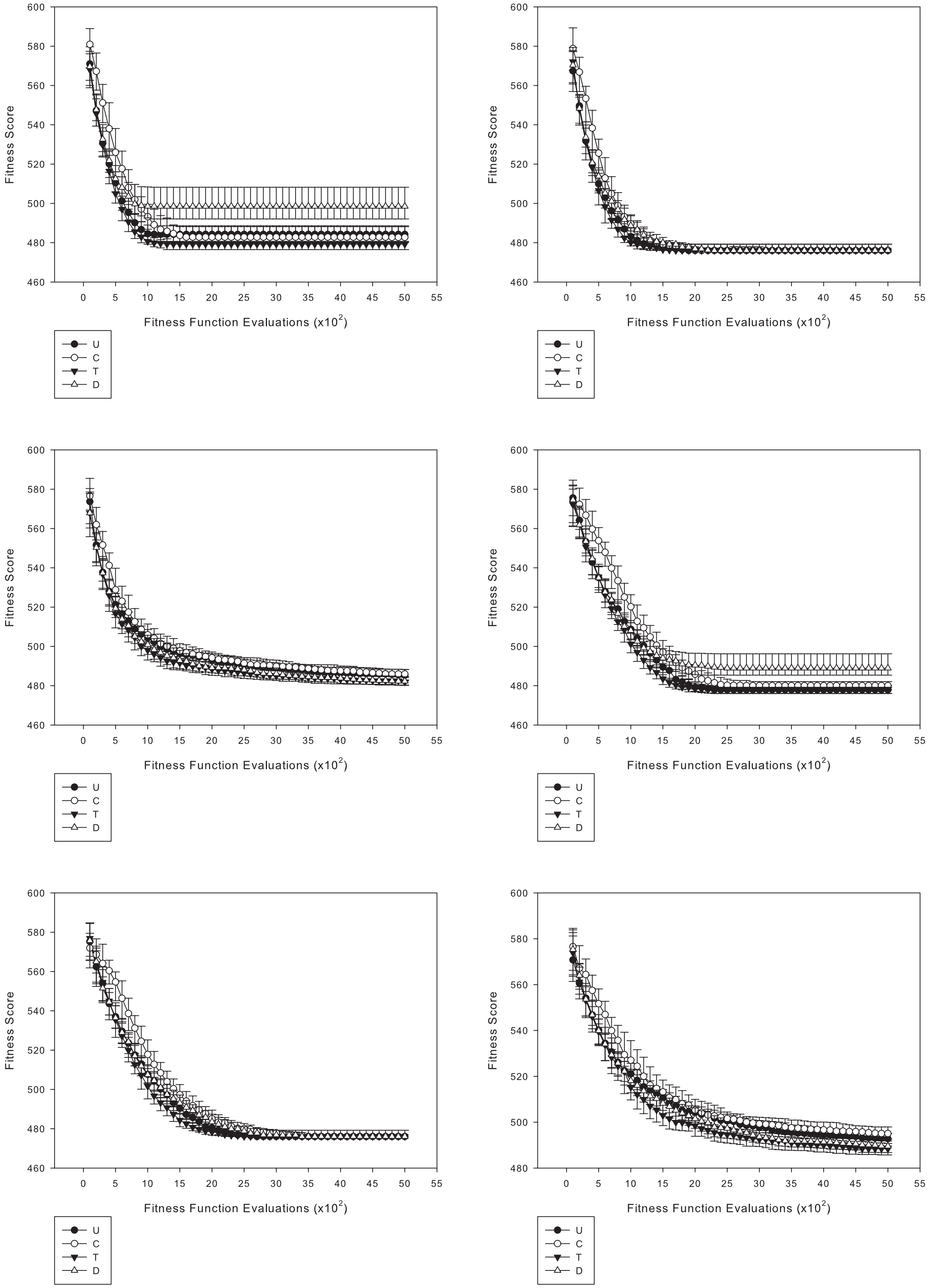


Figure C.8: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

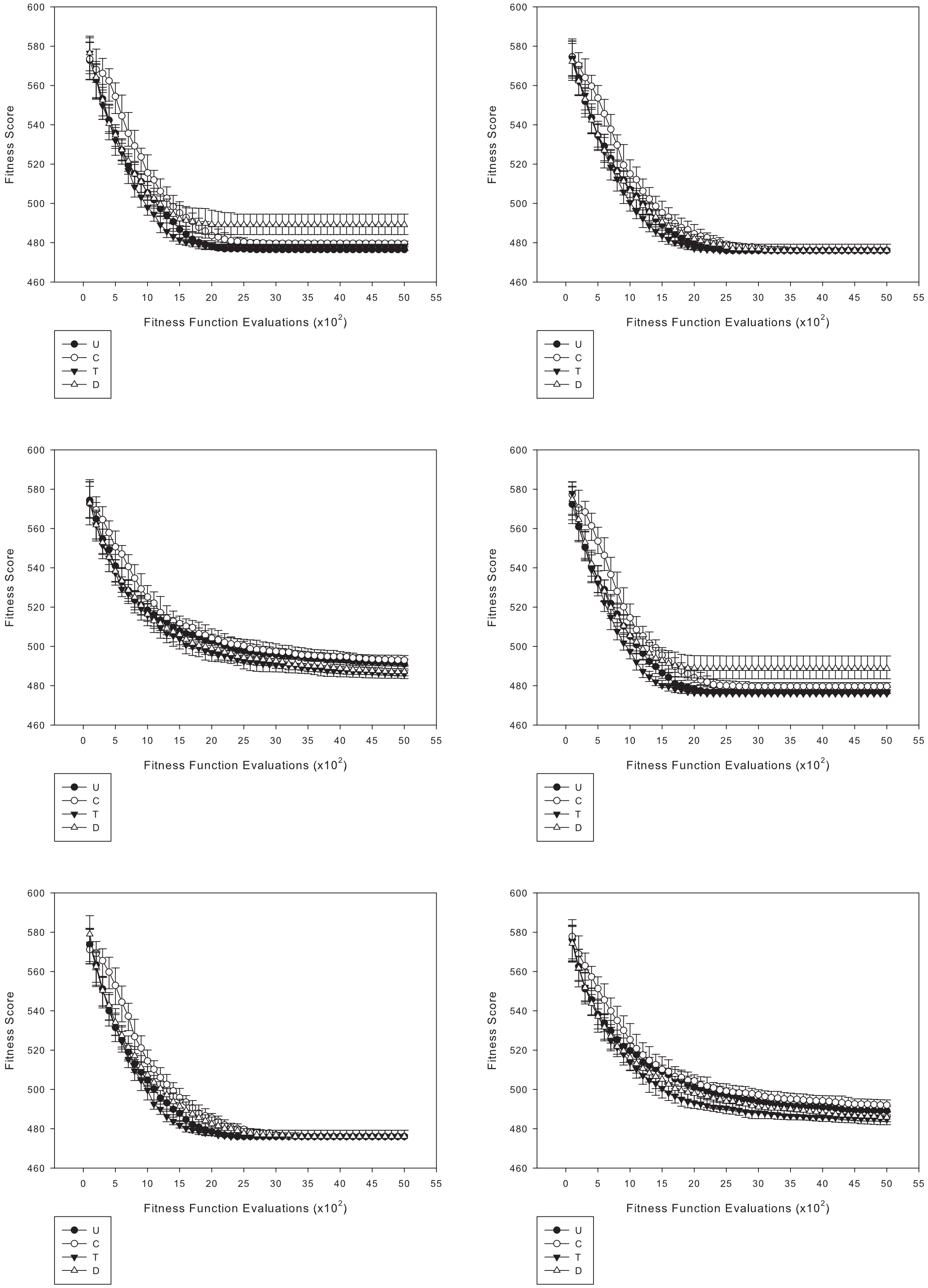


Figure C.9: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

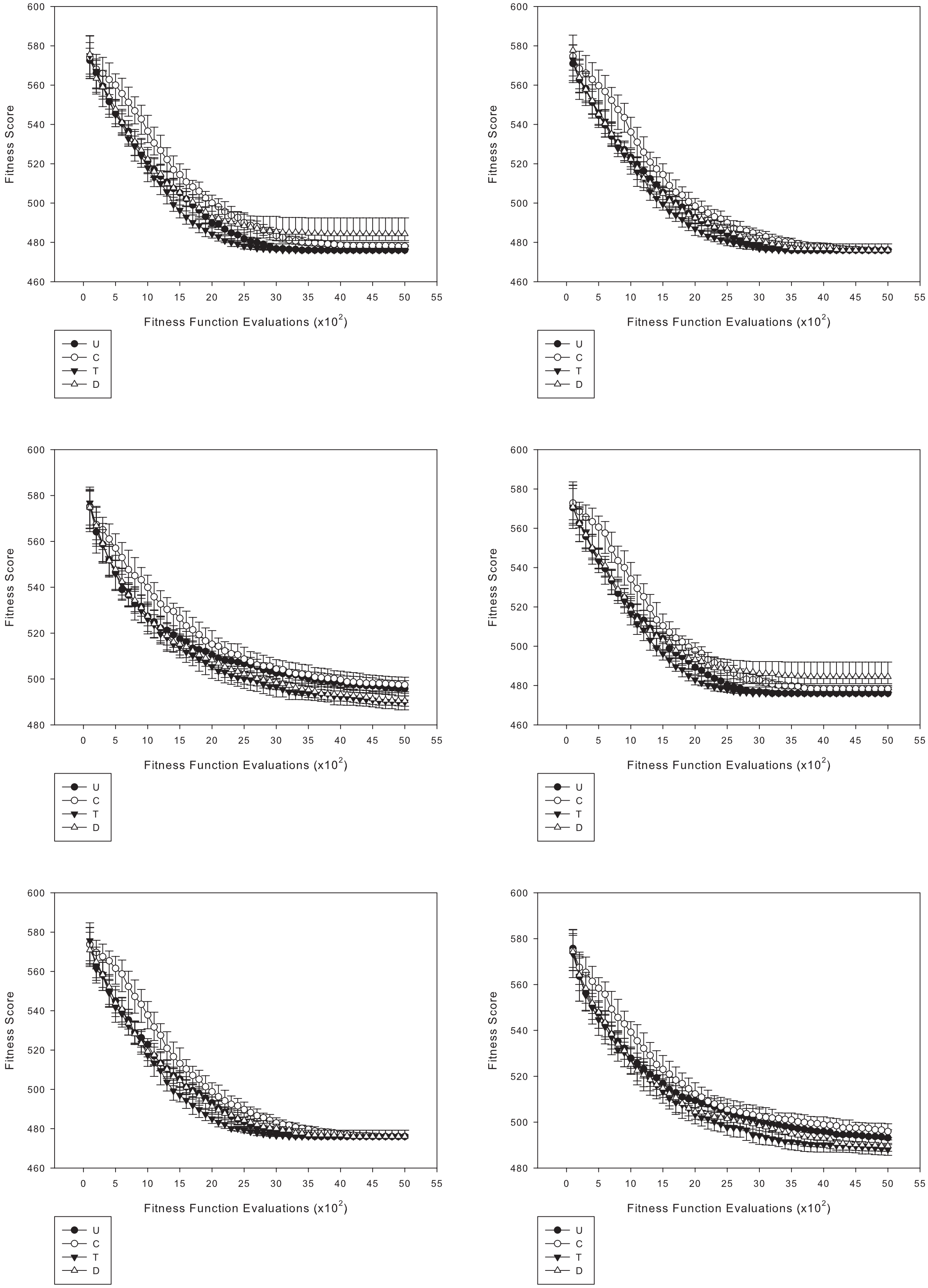
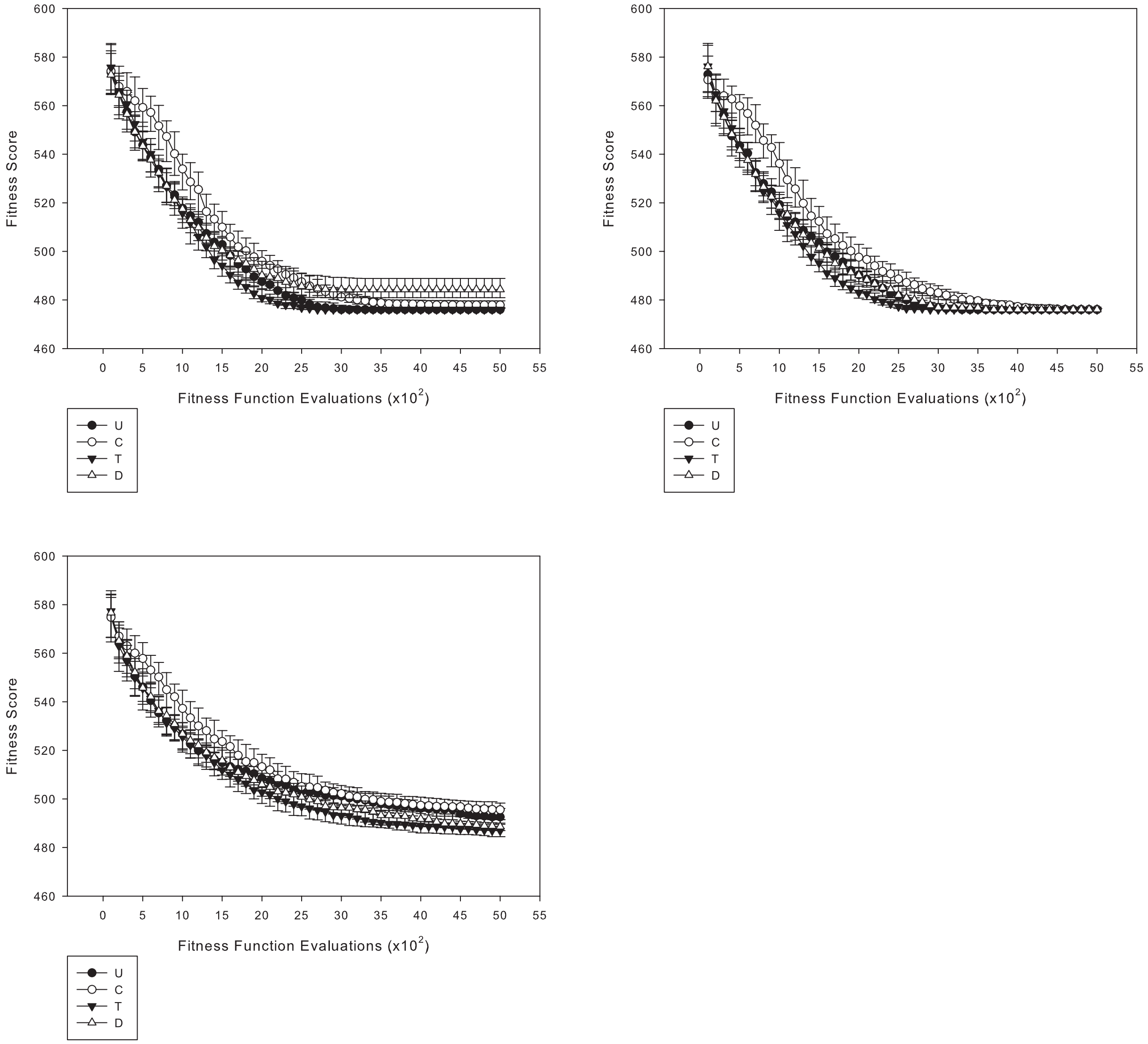


Figure C.10: $N = 150$, p_c 1.0, with p_m 0, 0.005



C.0.8 20 penalty points per intervention

Figure C.11: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

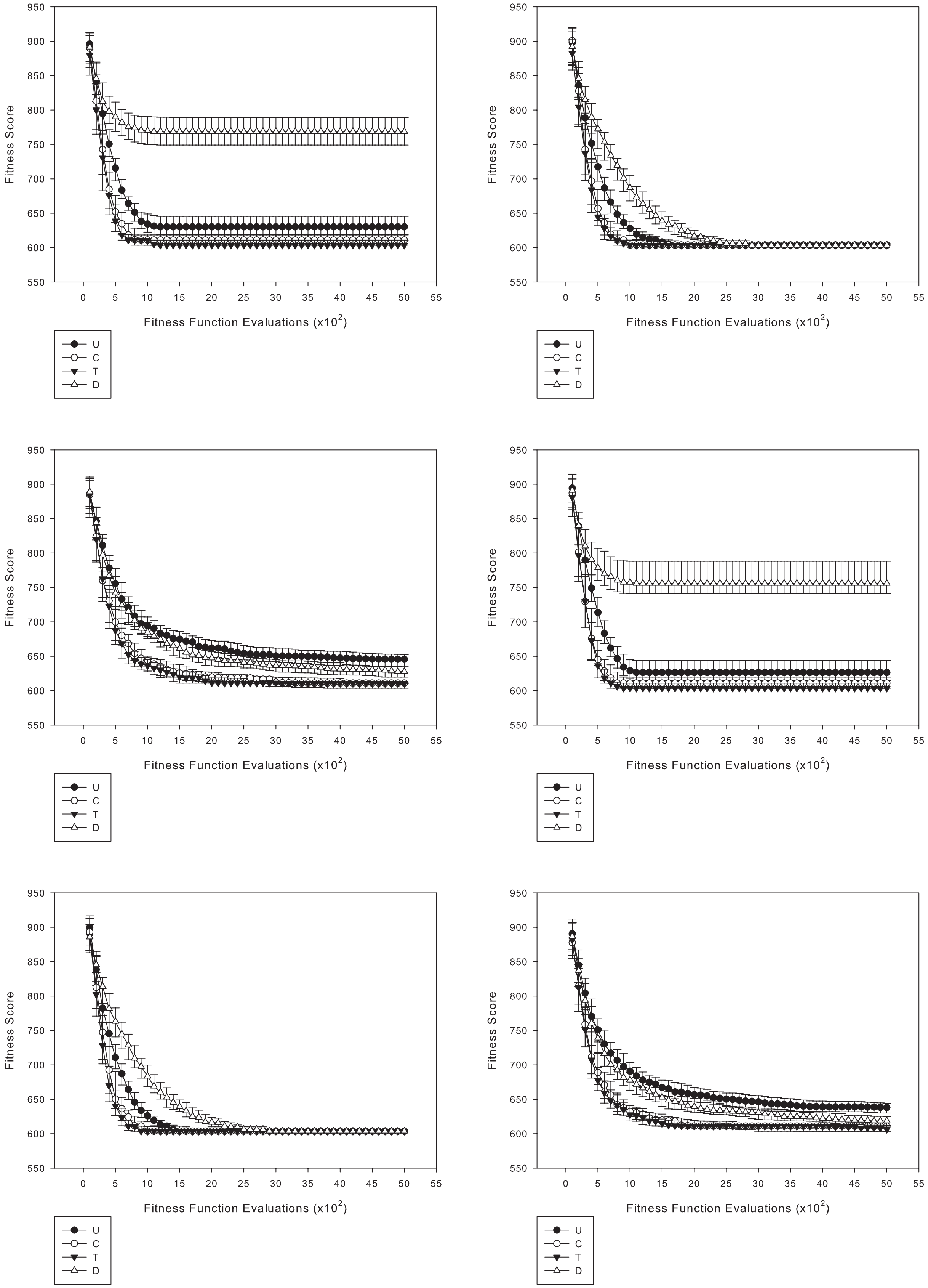


Figure C.12: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

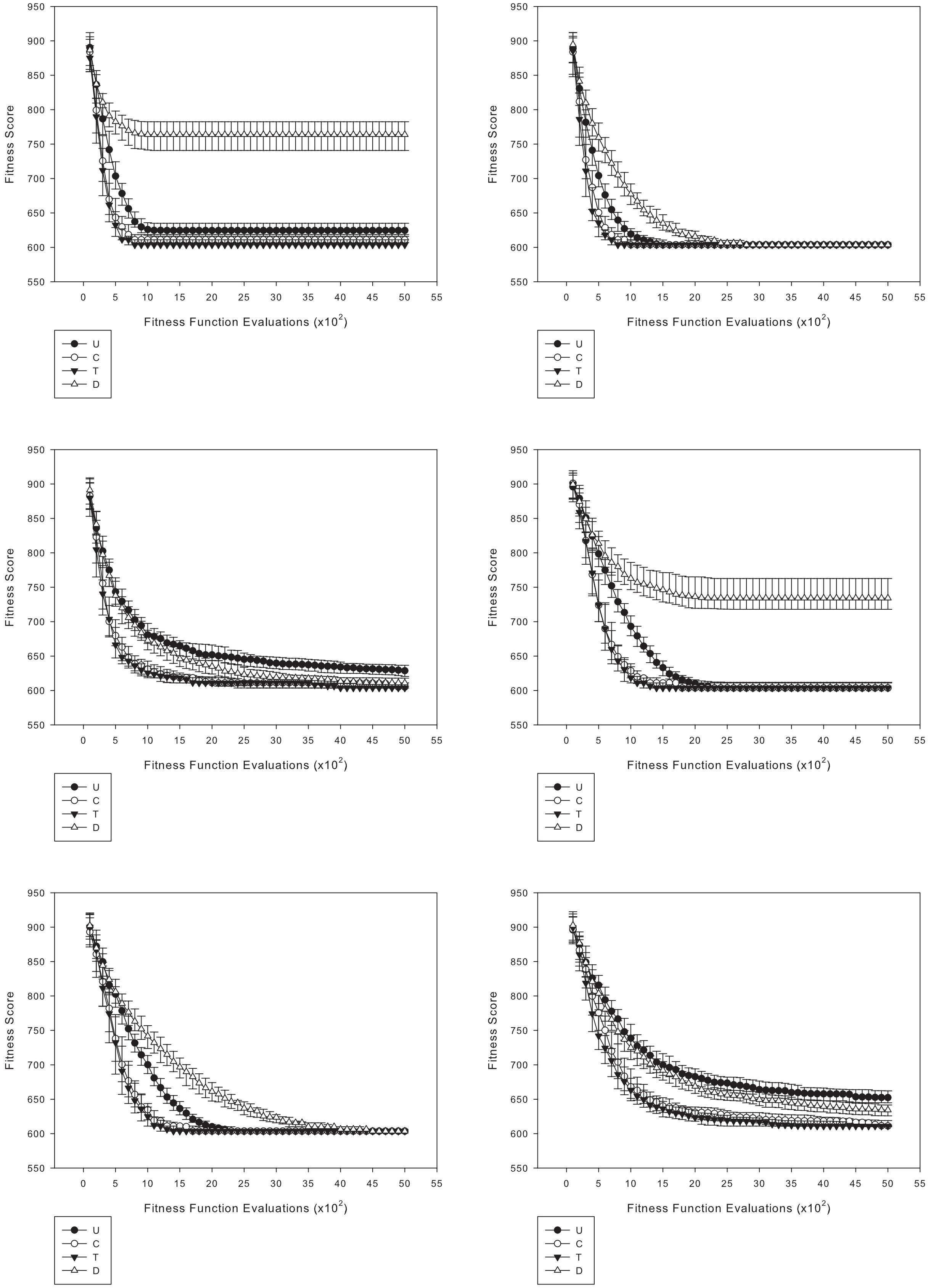


Figure C.13: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

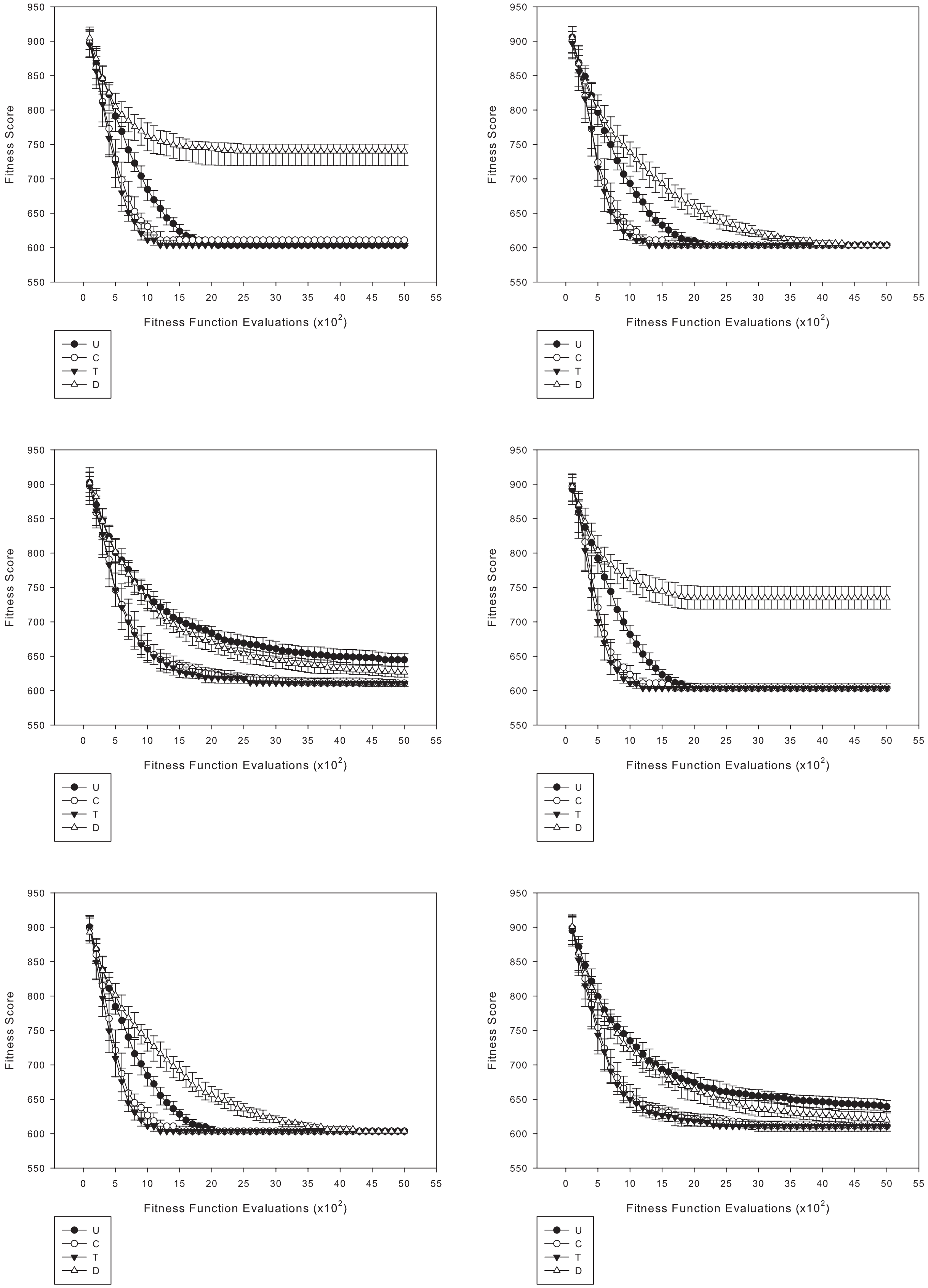


Figure C.14: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

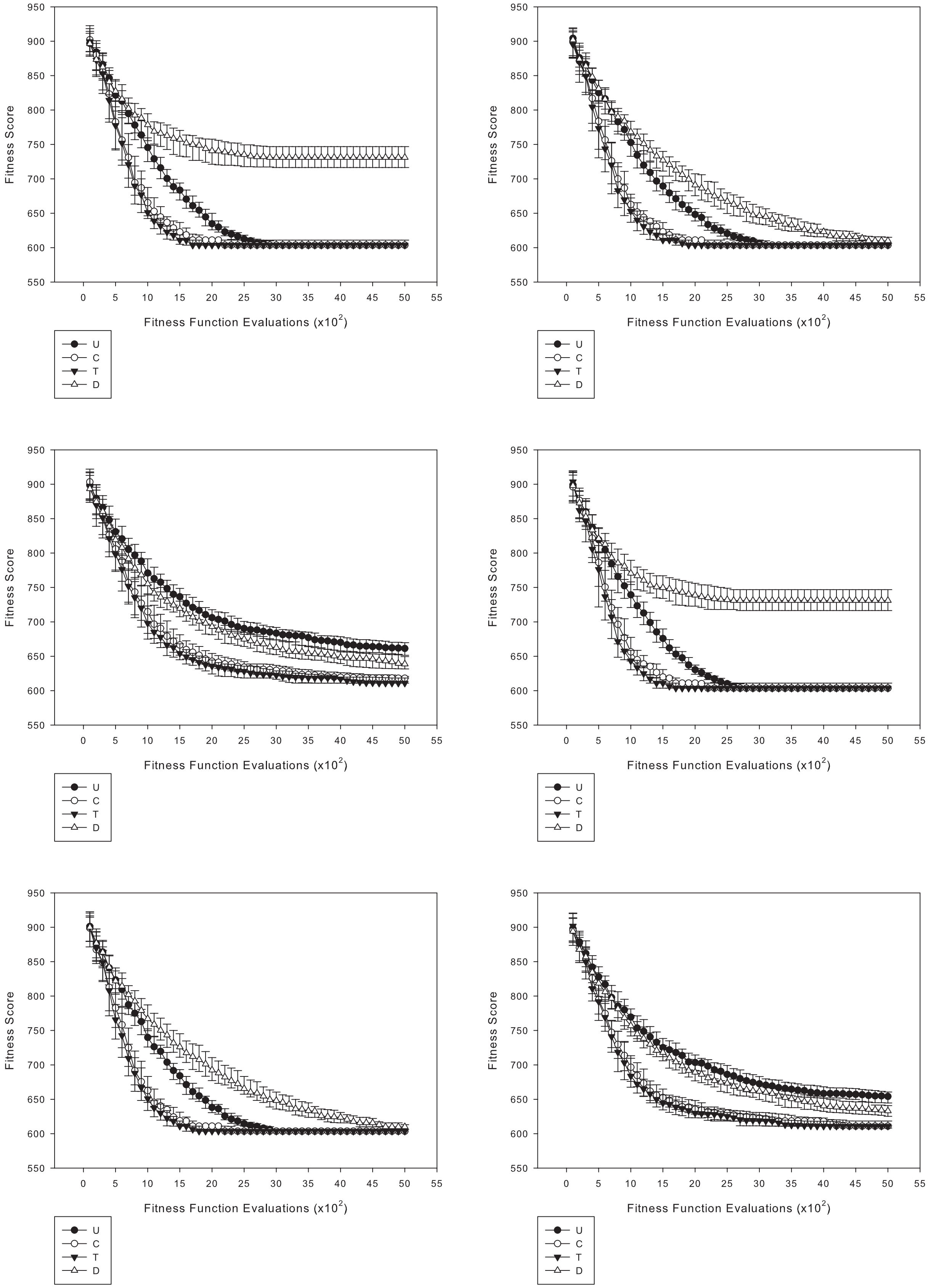
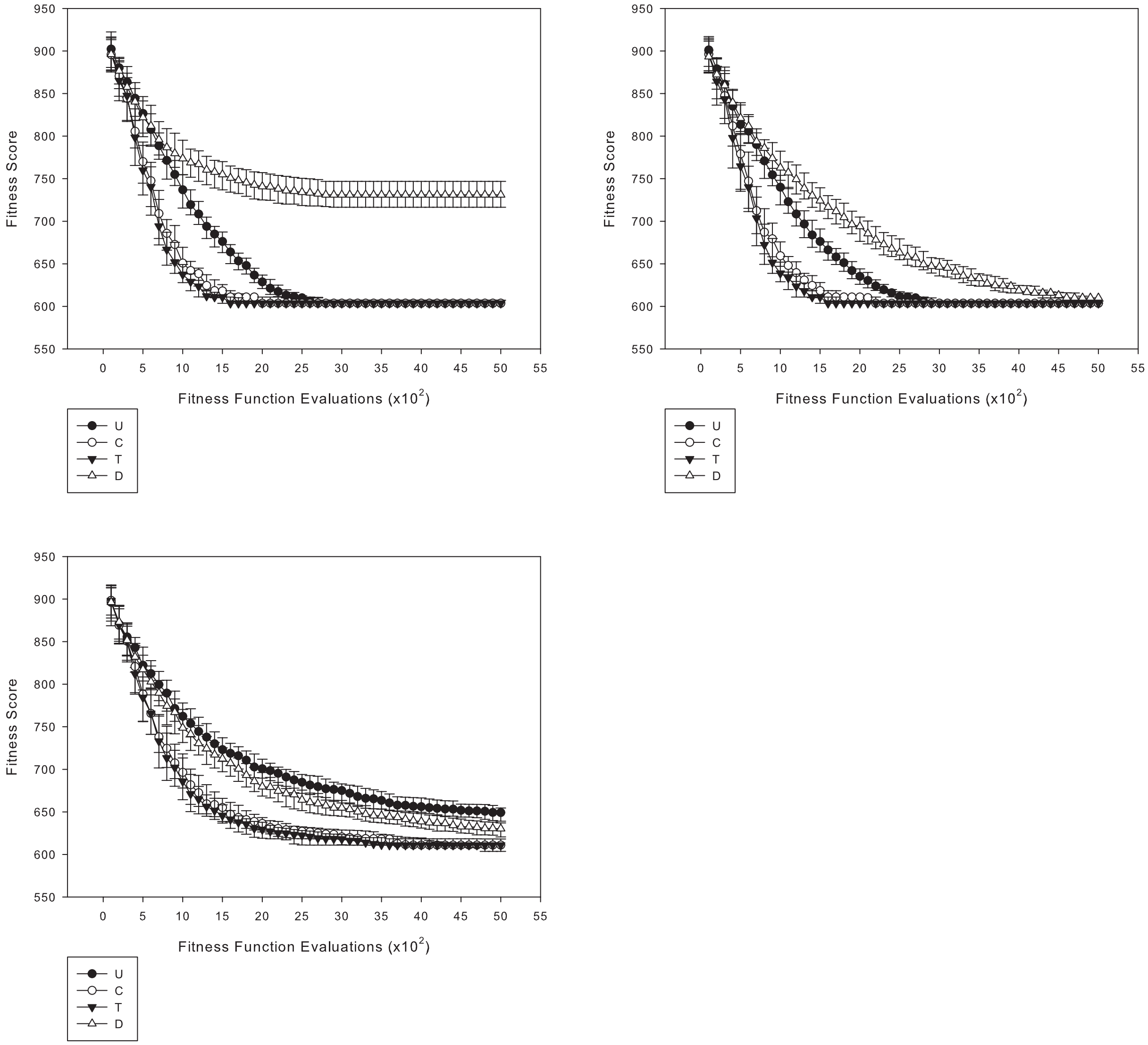


Figure C.15: $N = 150$, p_c 1.0, with p_m 0, 0.005



C.0.9 35 penalty points per intervention

Figure C.16: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

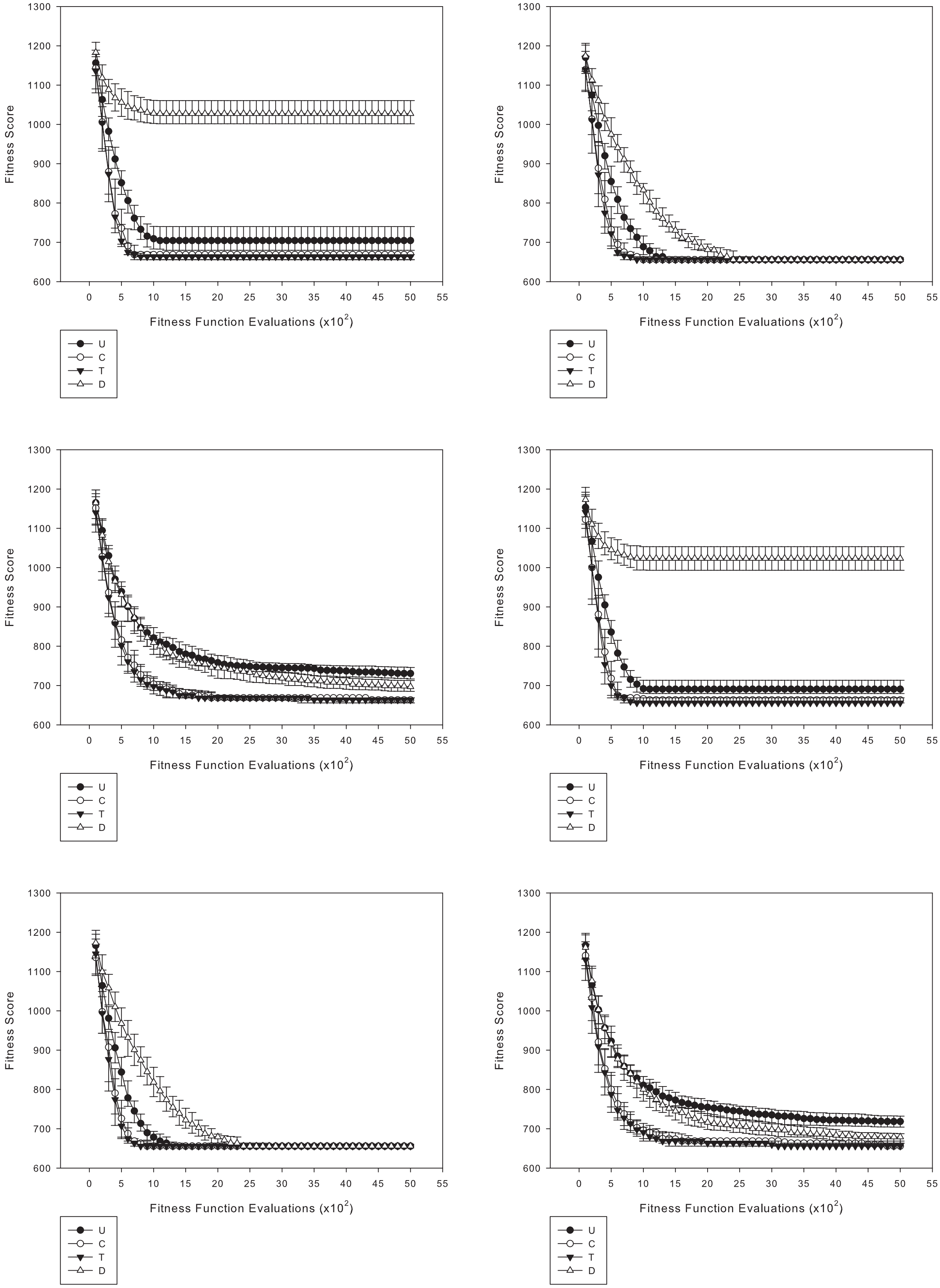


Figure C.17: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

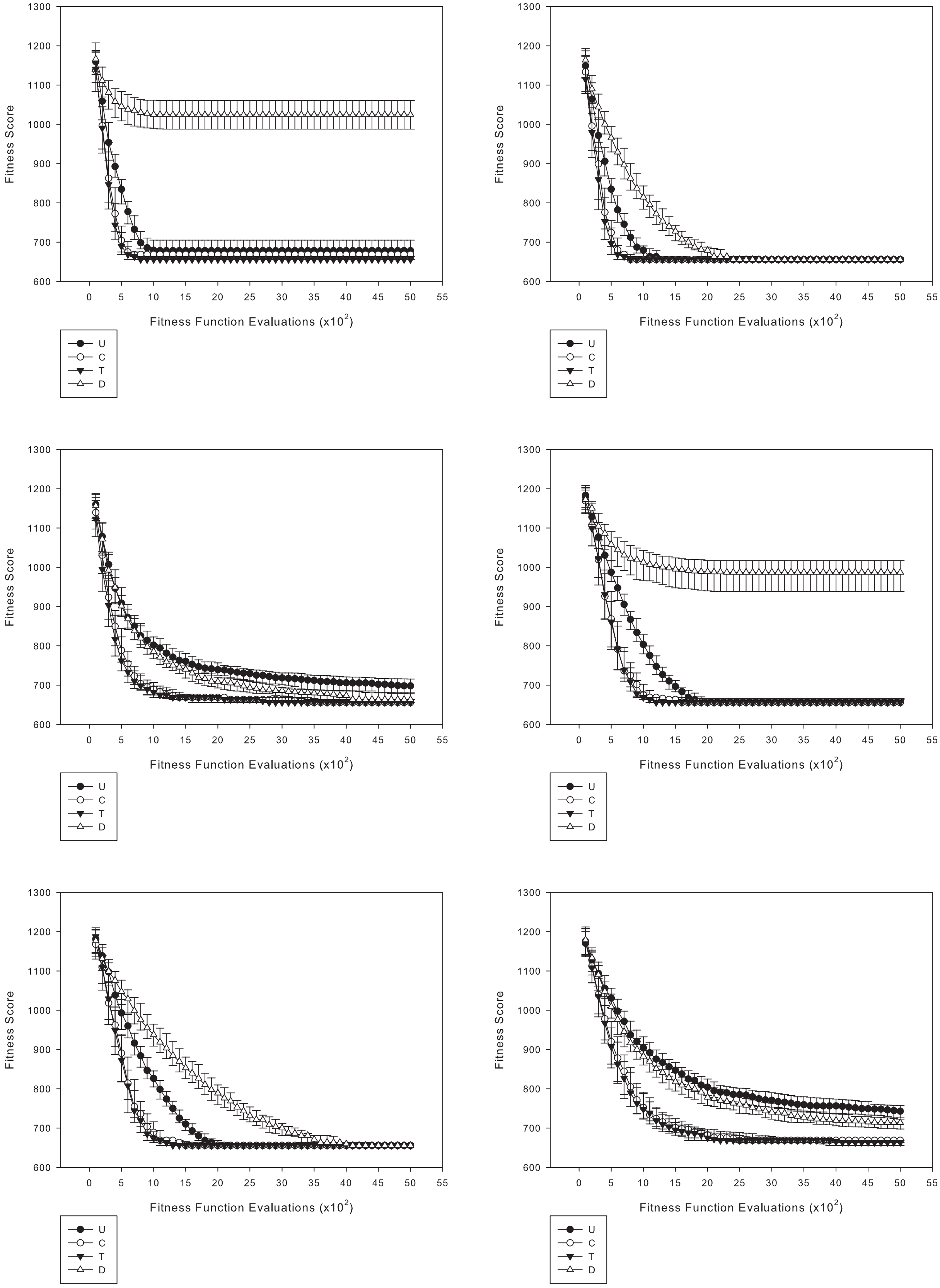


Figure C.18: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

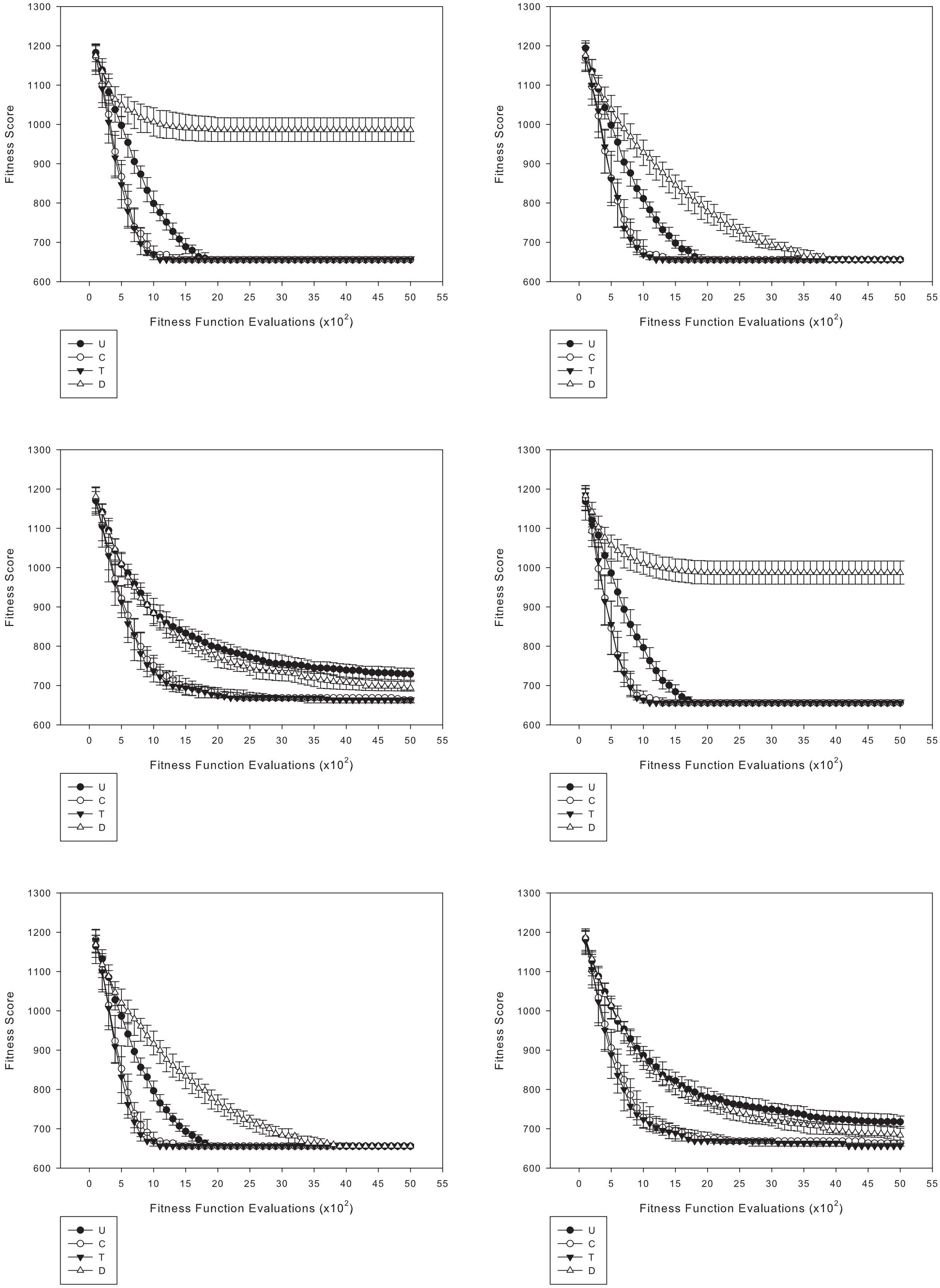


Figure C.19: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

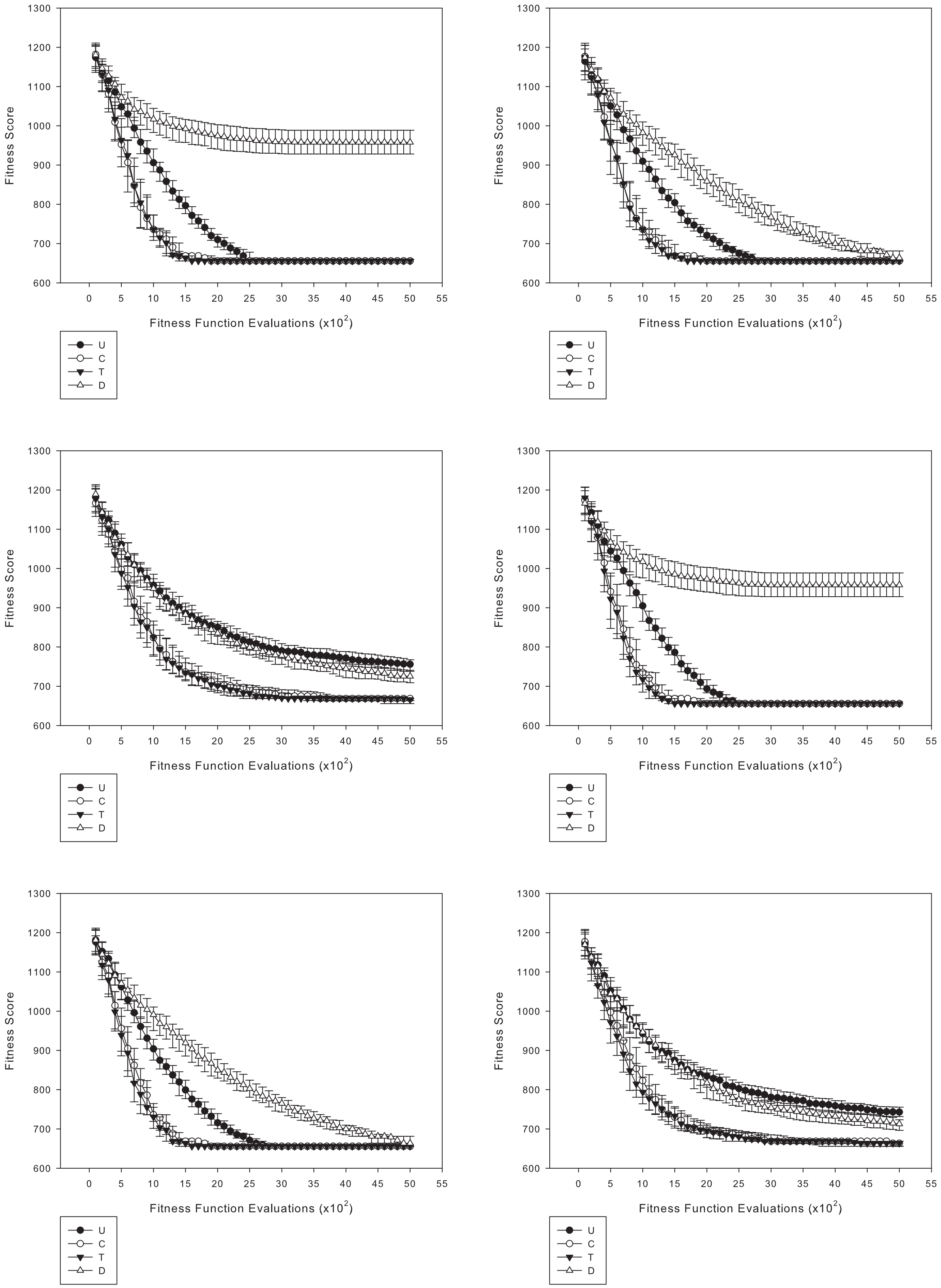
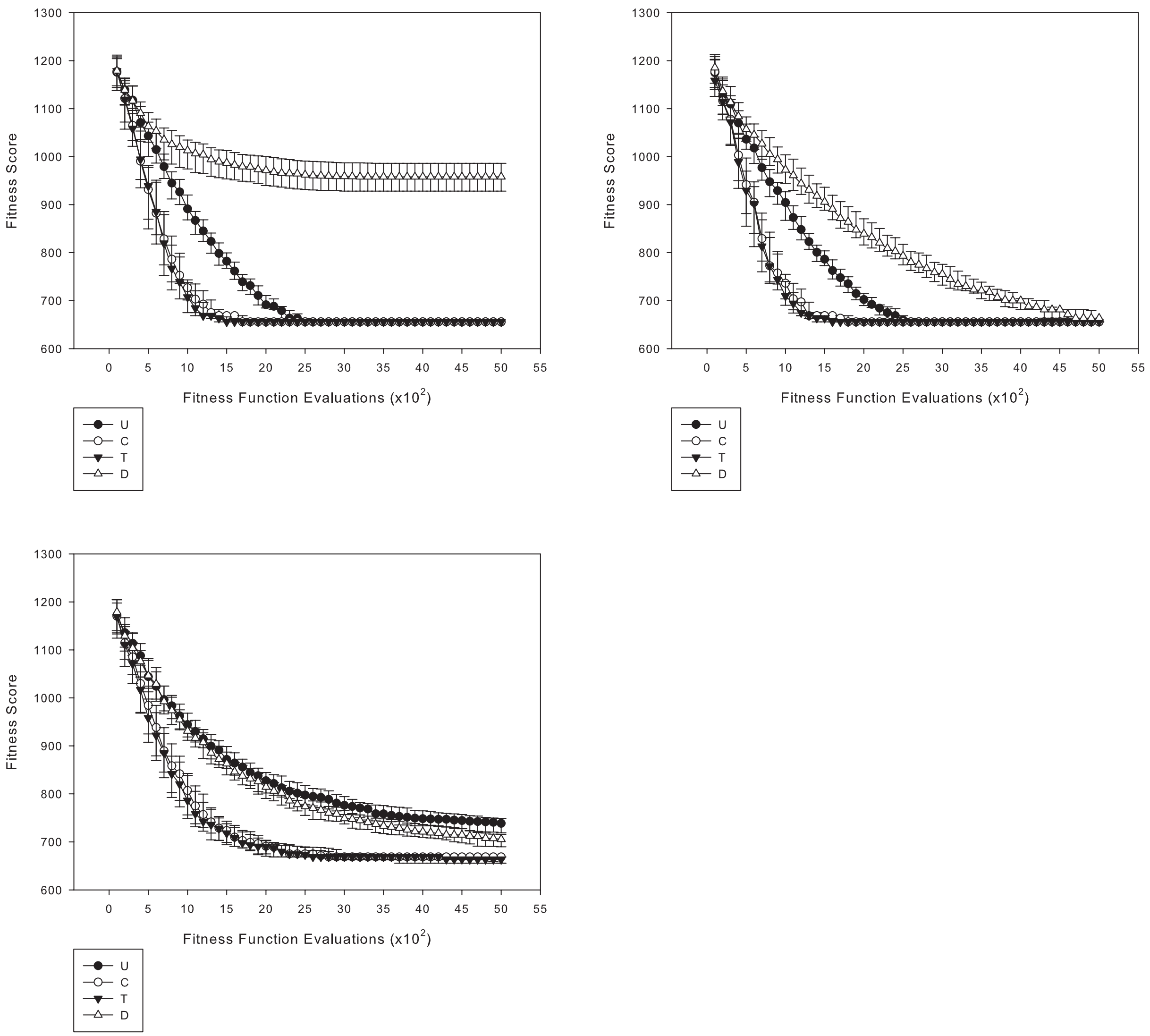


Figure C.20: $N = 150$, p_c 1.0, with p_m 0, 0.005



C.0.10 50 penalty points per intervention

Figure C.21: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

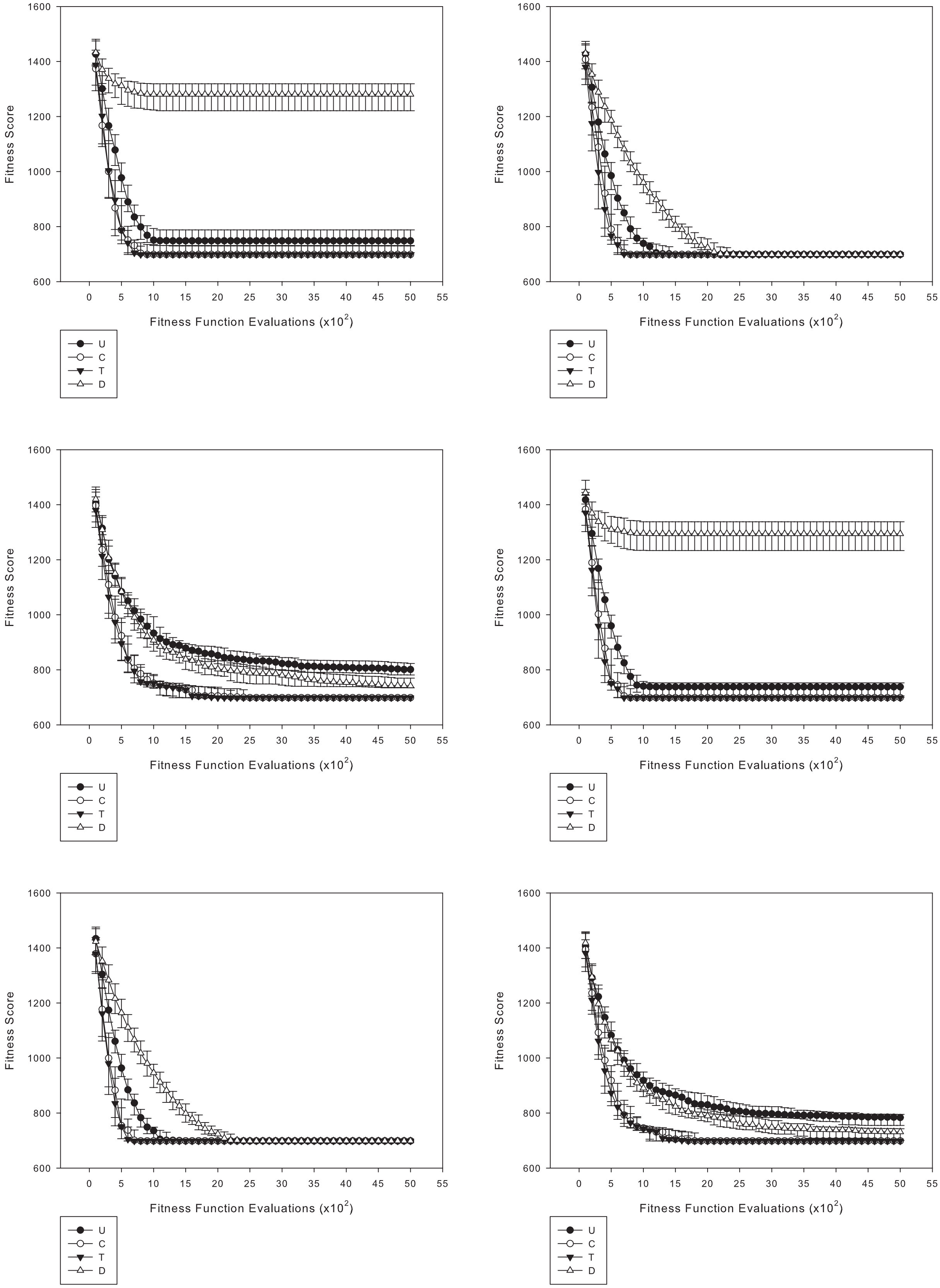


Figure C.22: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

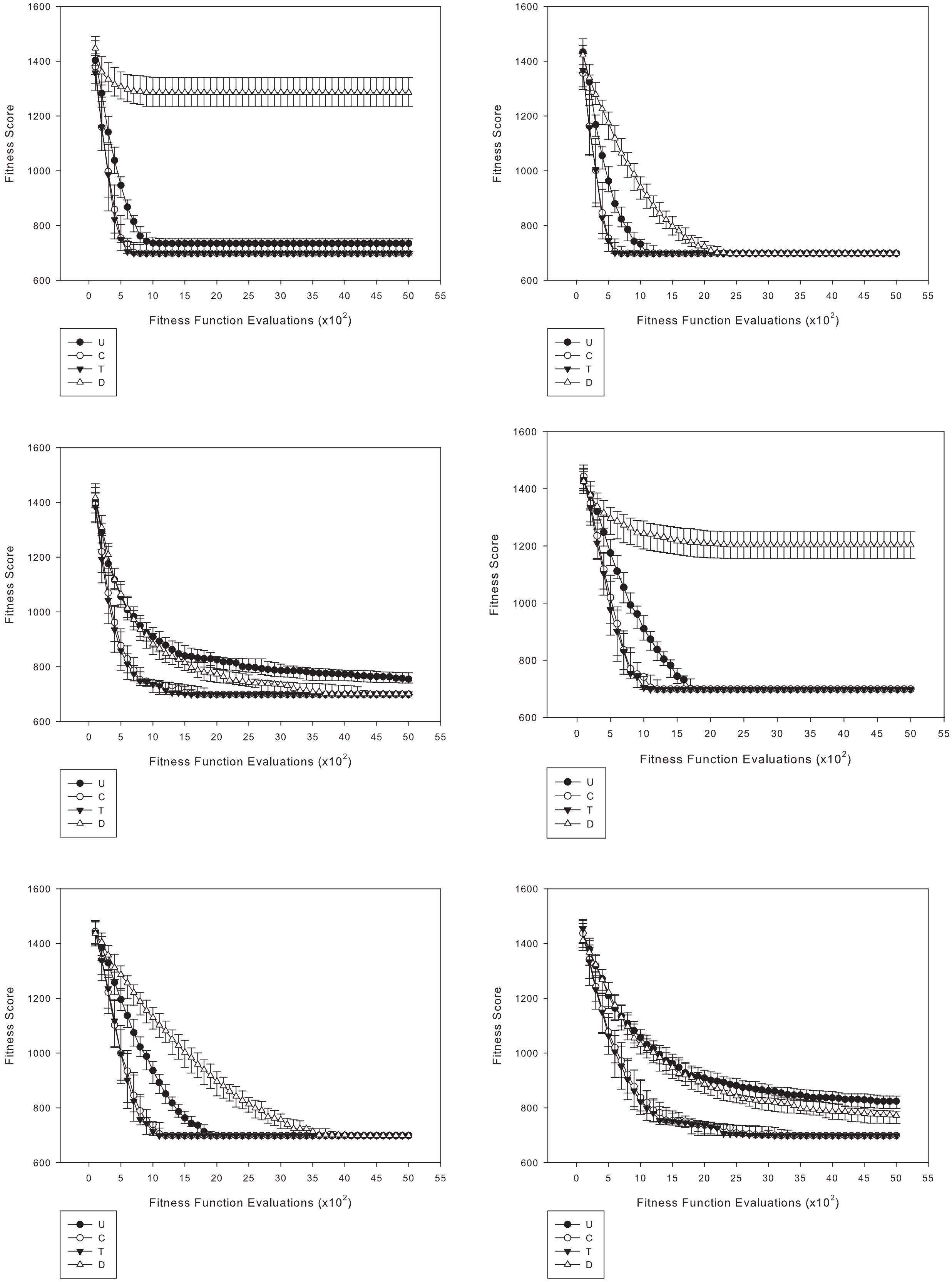


Figure C.23: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

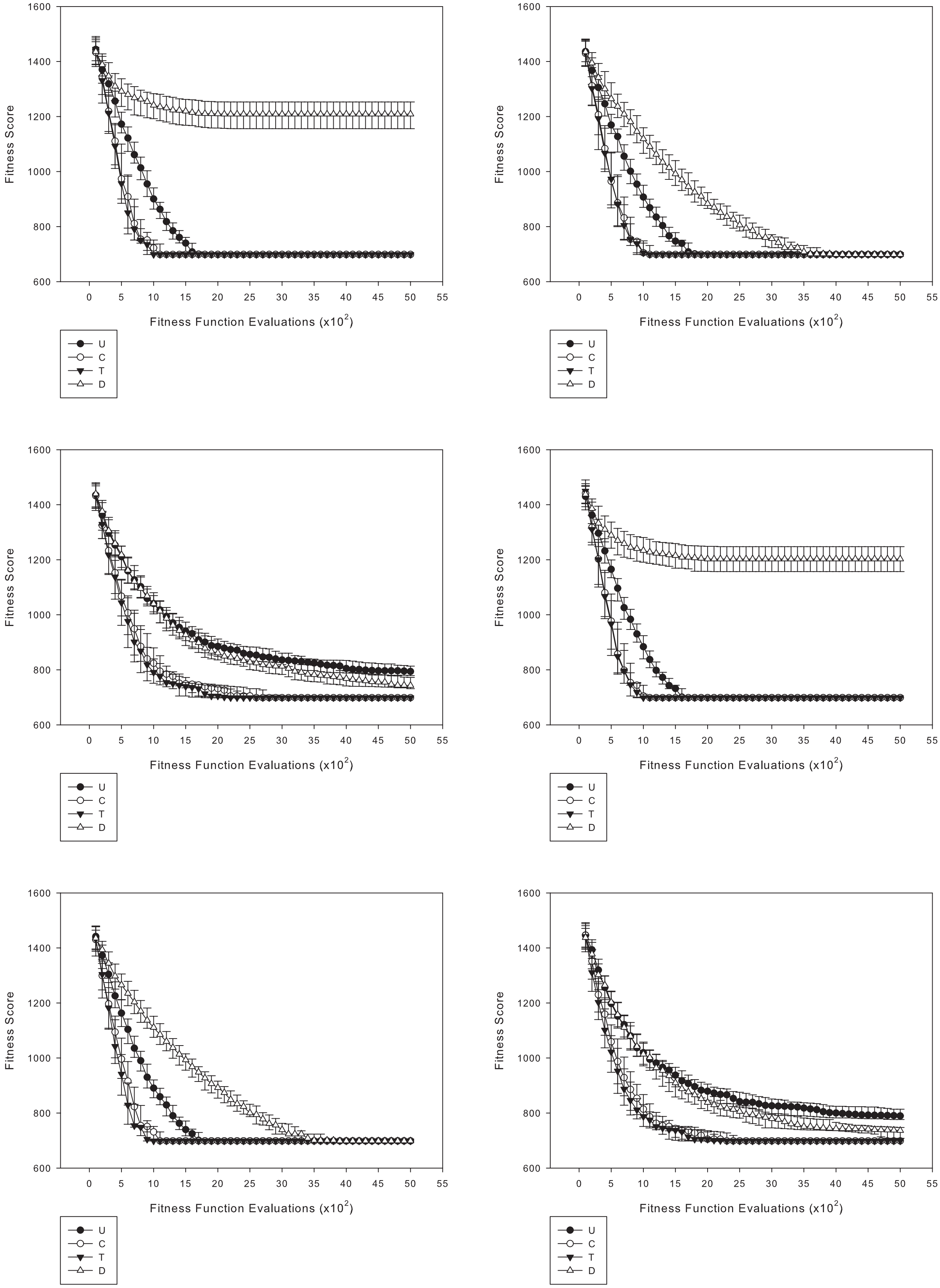


Figure C.24: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

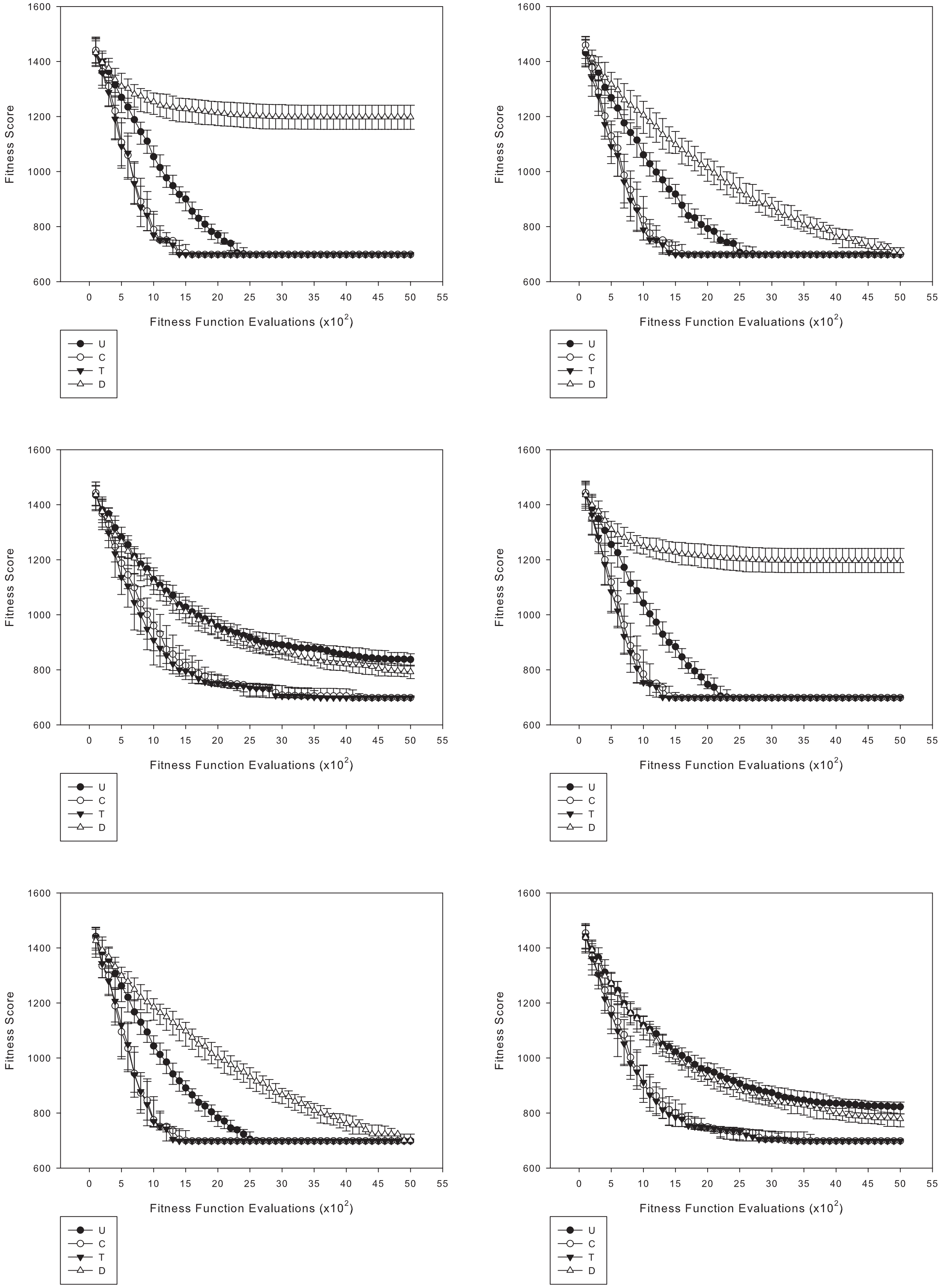
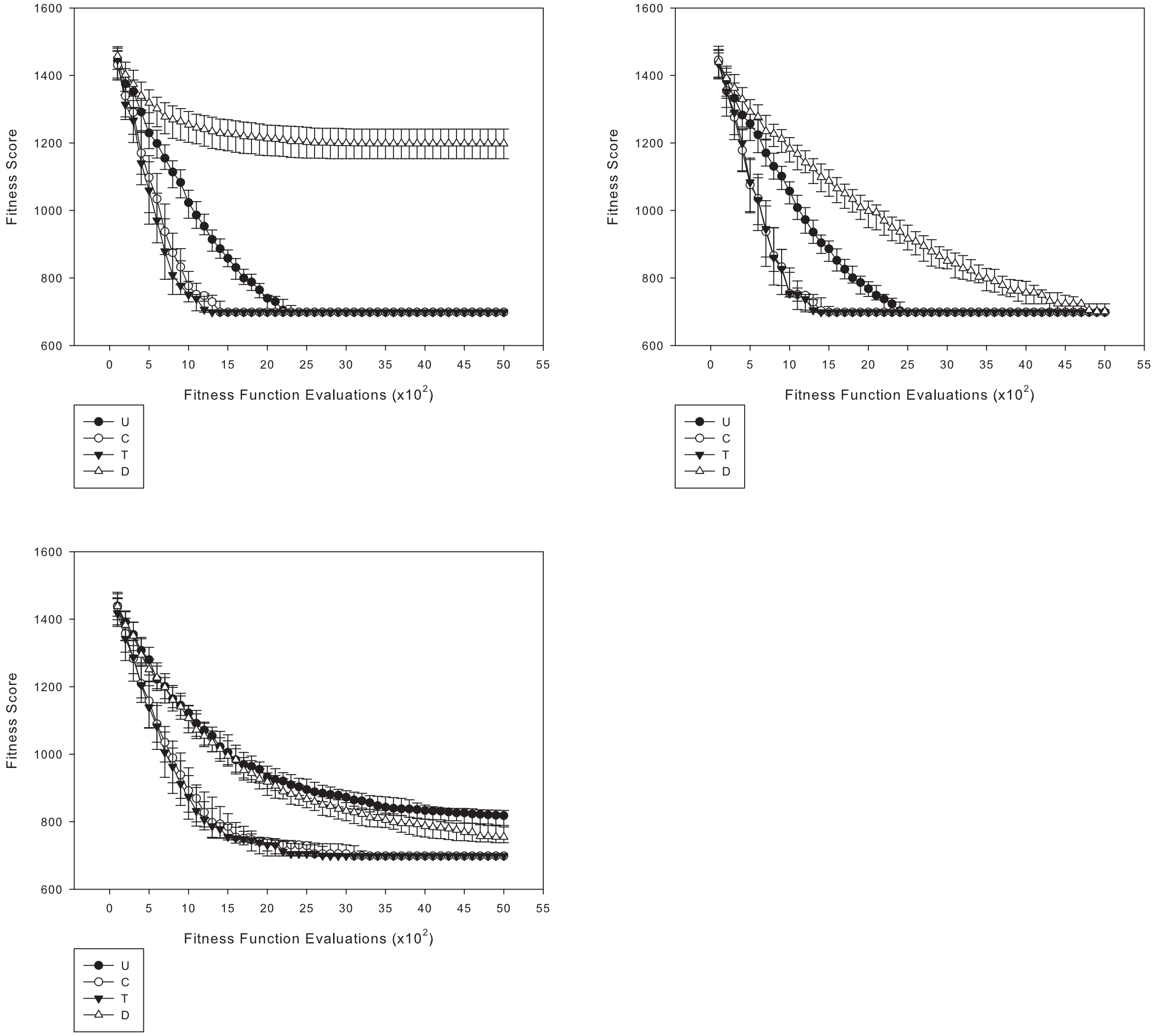


Figure C.25: $N = 150$, p_c 1.0, with p_m 0, 0.005



Appendix D

Bio-control scheduling graphs for UC, CalEB, TInSSel and FDC

D.0.11 0 penalty points per intervention

Figure D.1: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

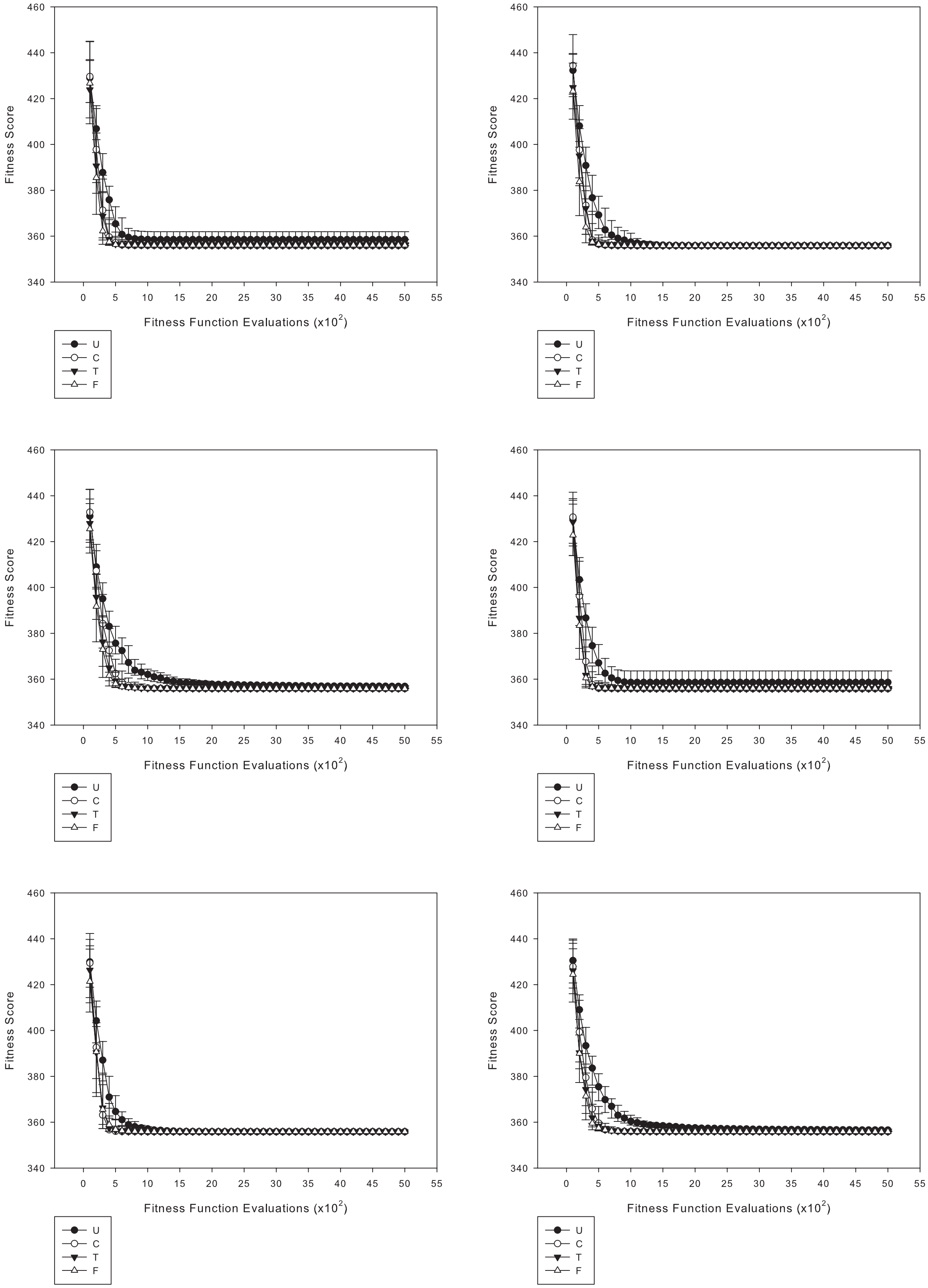


Figure D.2: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

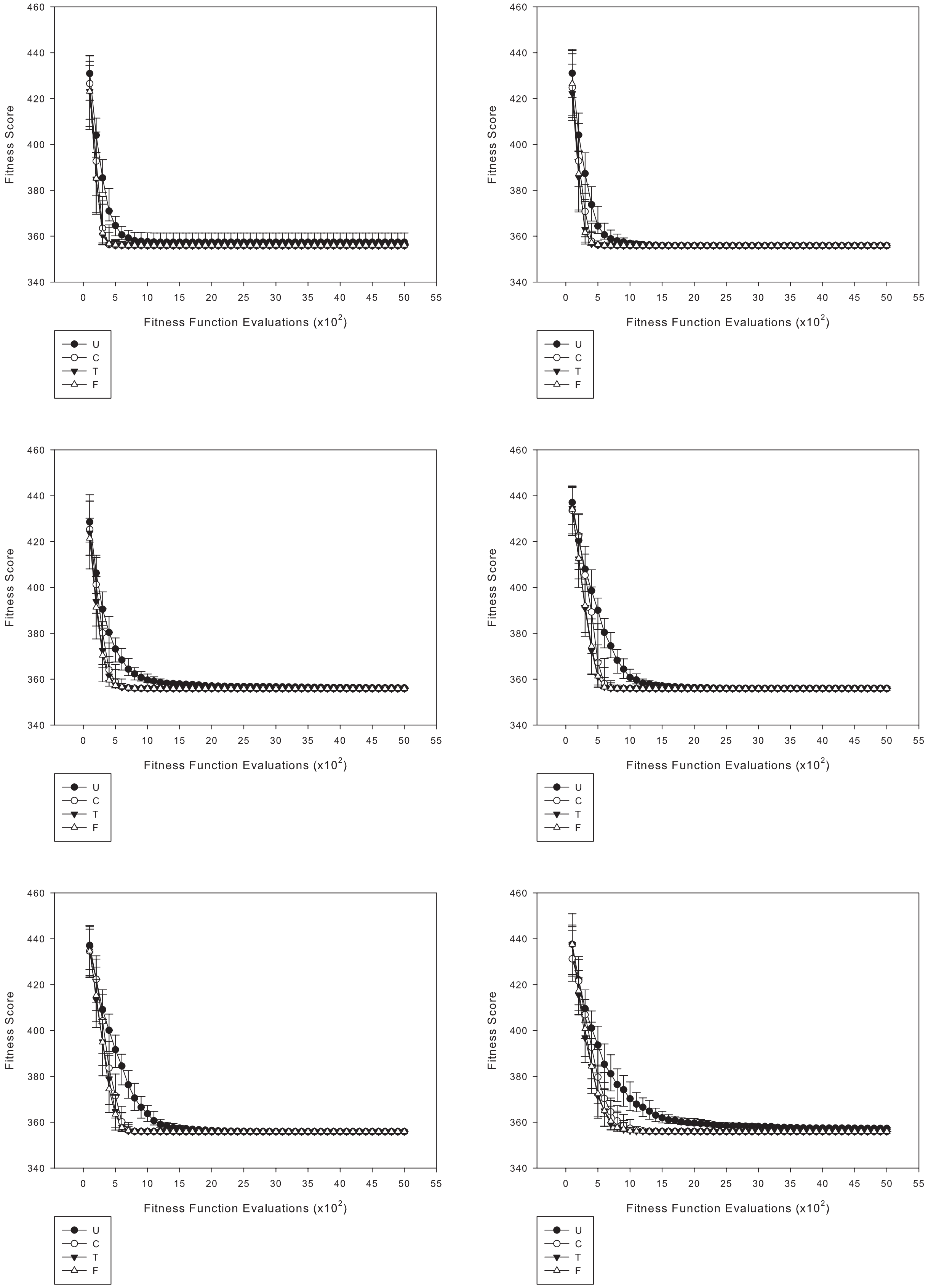


Figure D.3: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

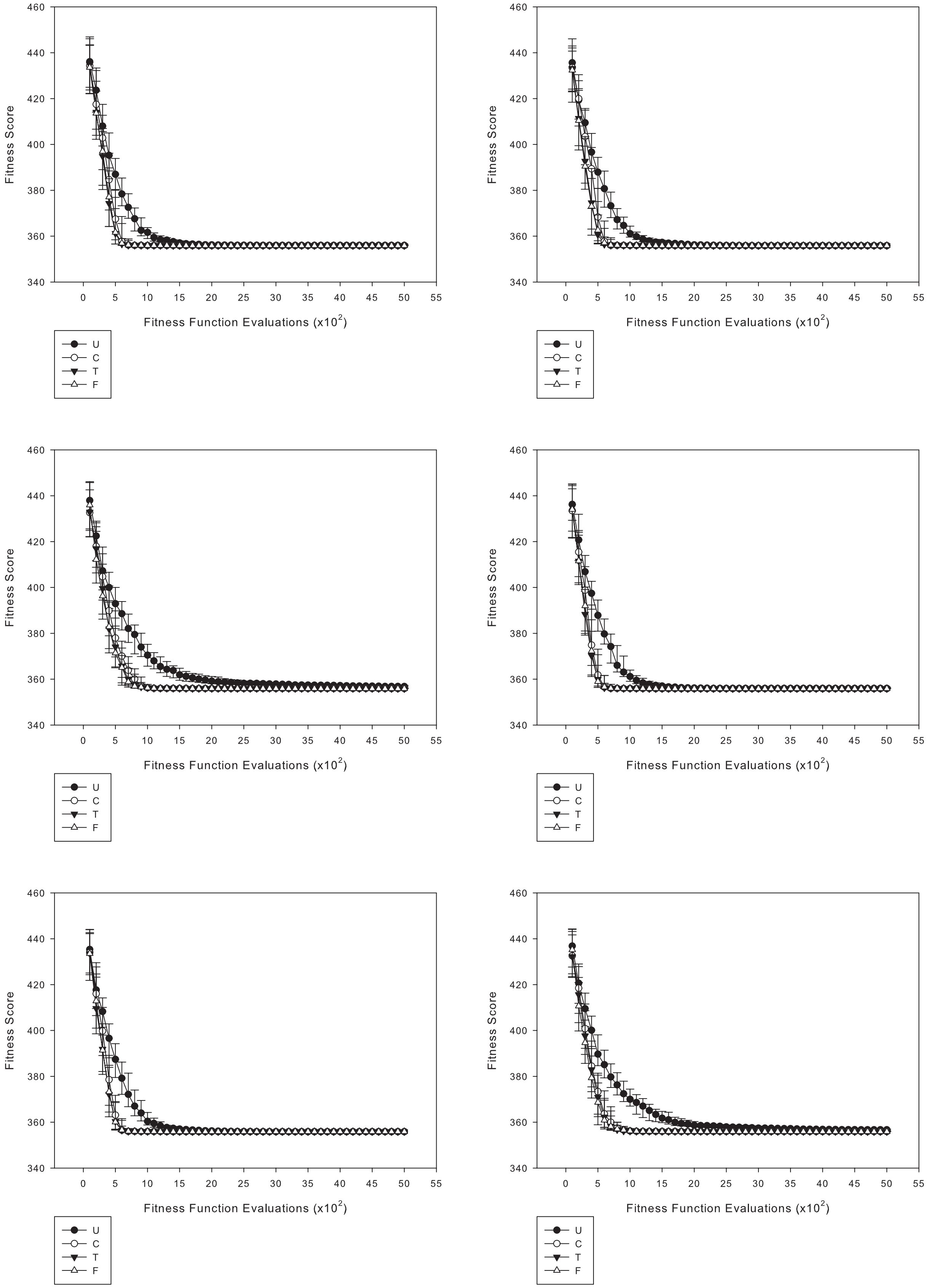


Figure D.4: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

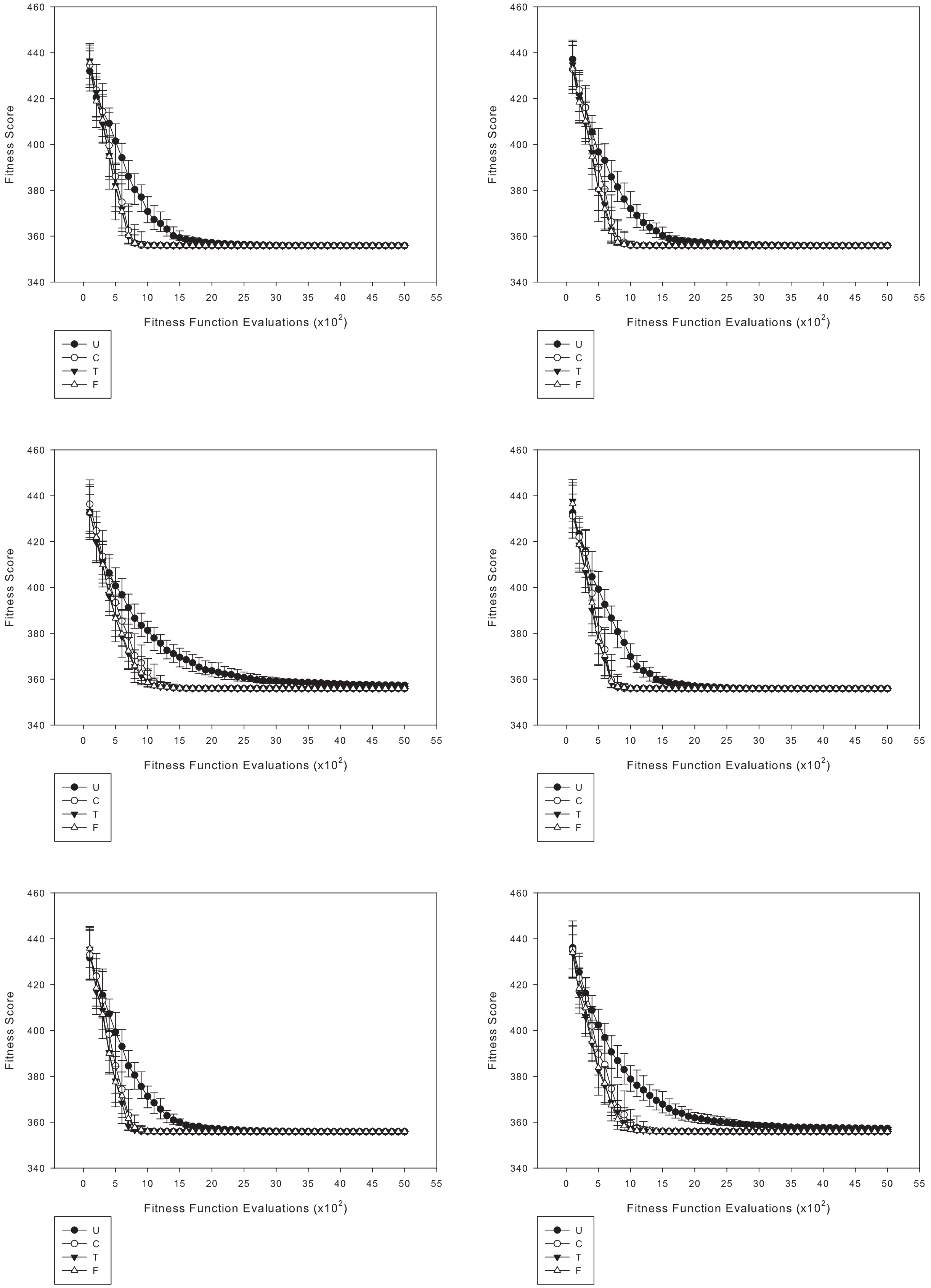
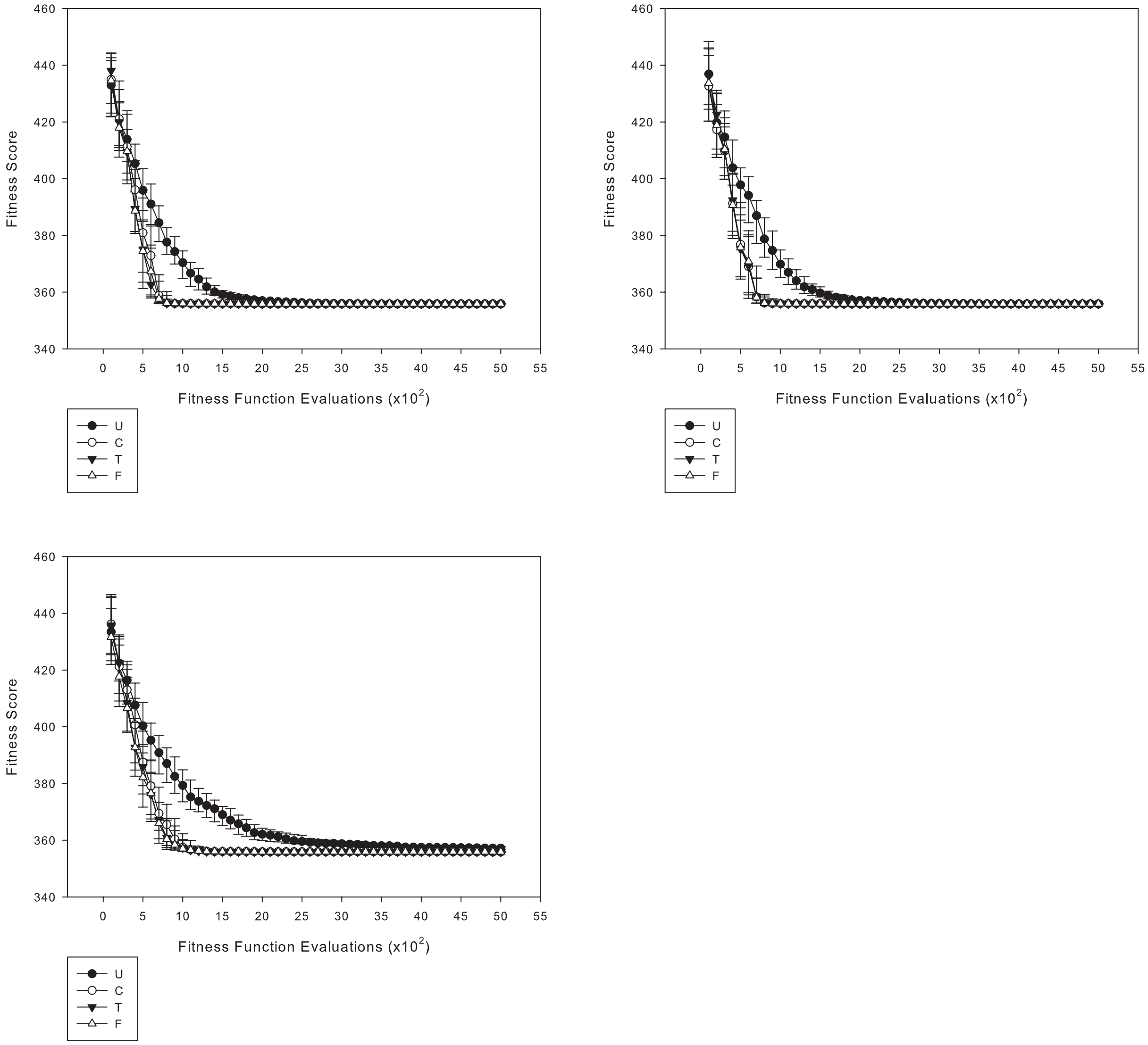


Figure D.5: $N = 150$, p_c 1.0, with p_m 0, 0.005



D.0.12 5 penalty points per intervention

Figure D.6: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

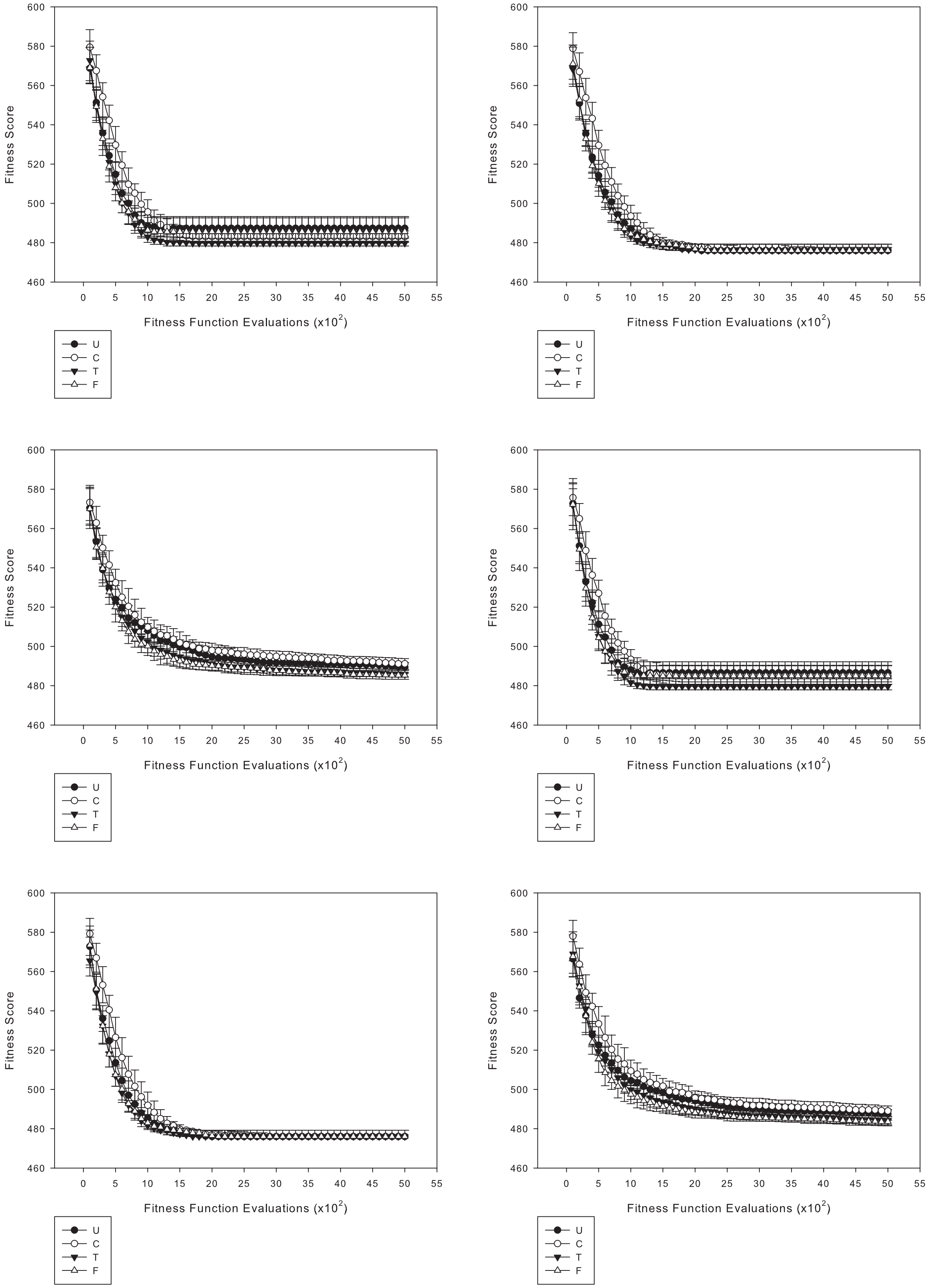


Figure D.7: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

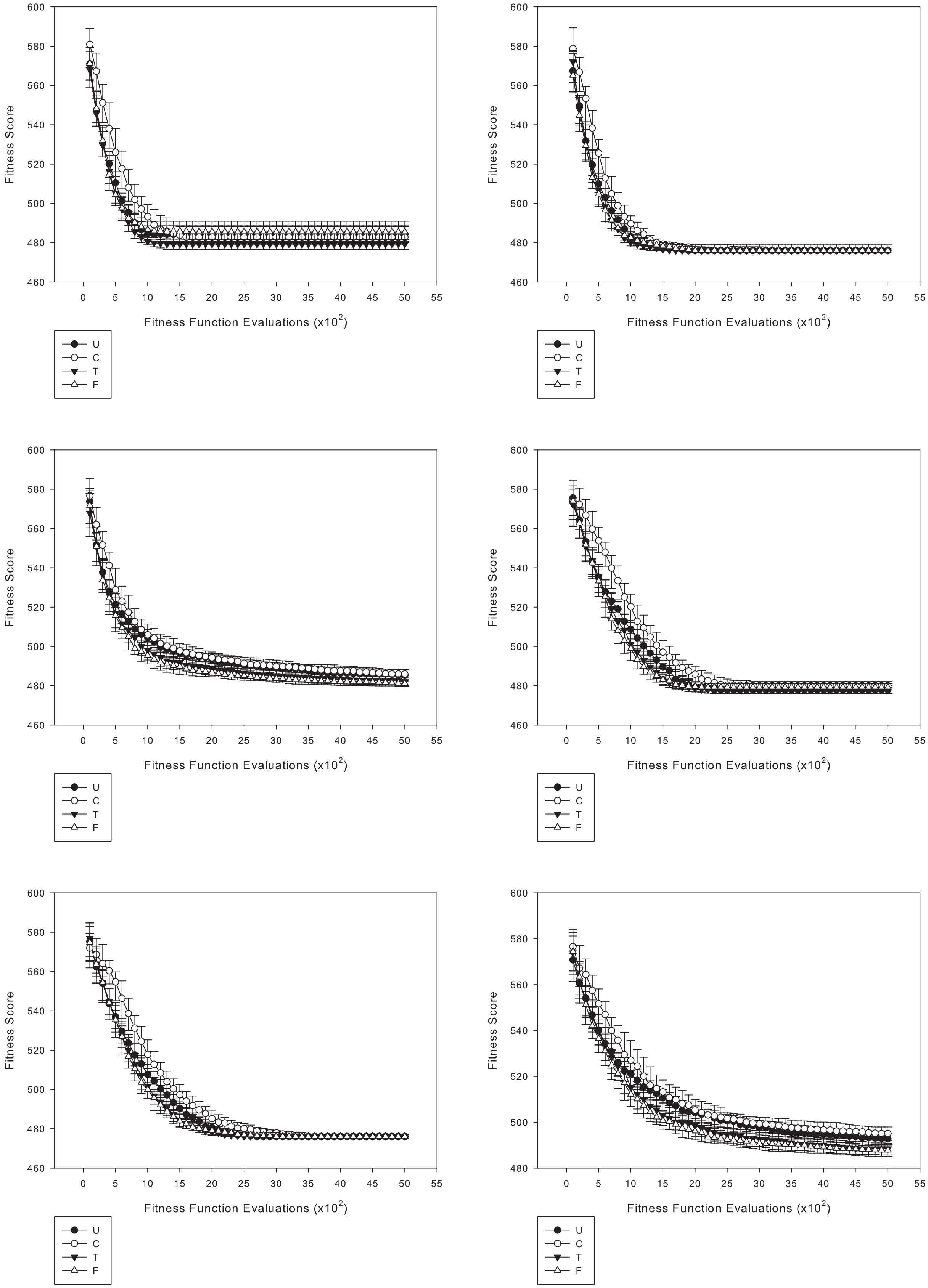


Figure D.8: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

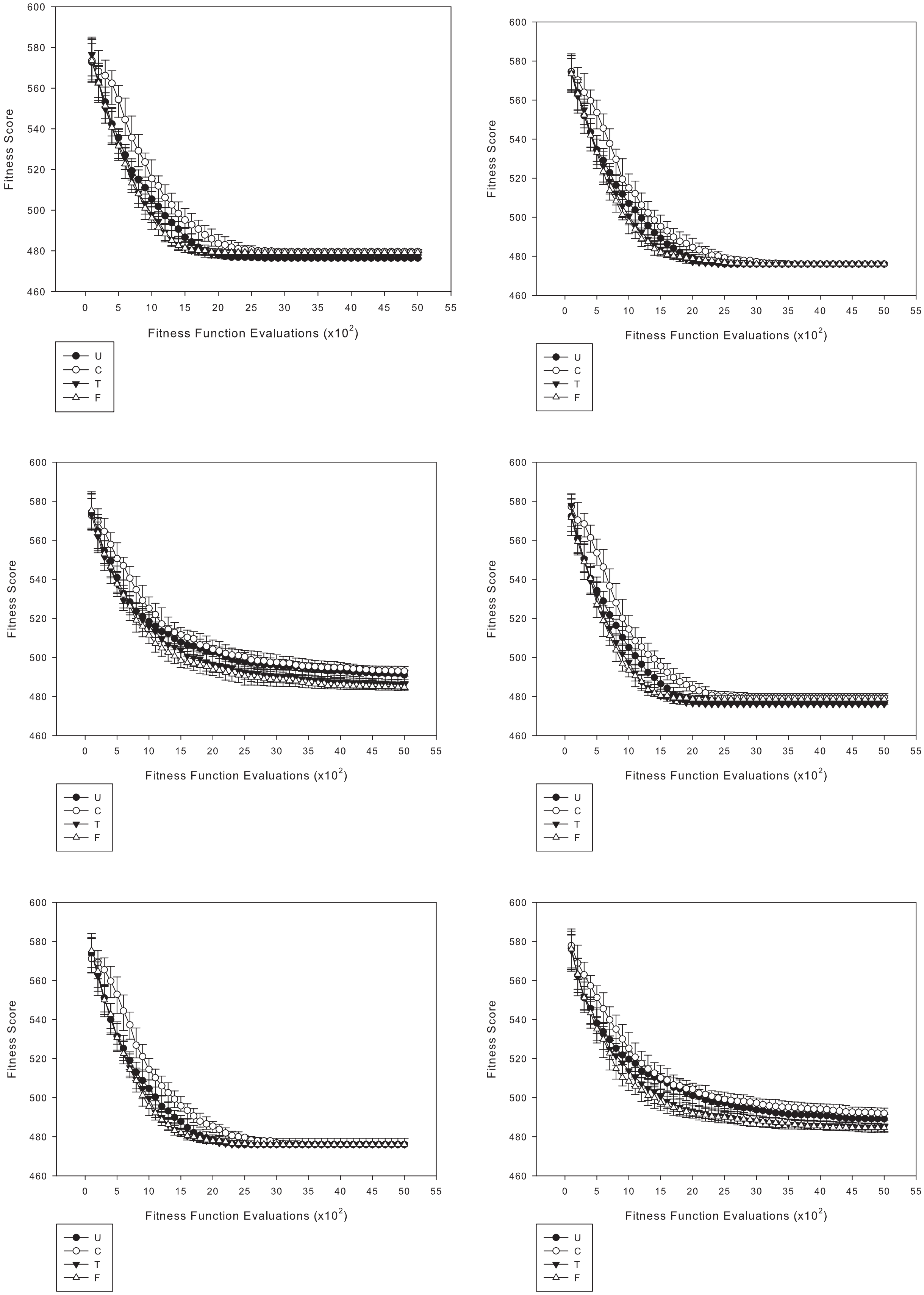


Figure D.9: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

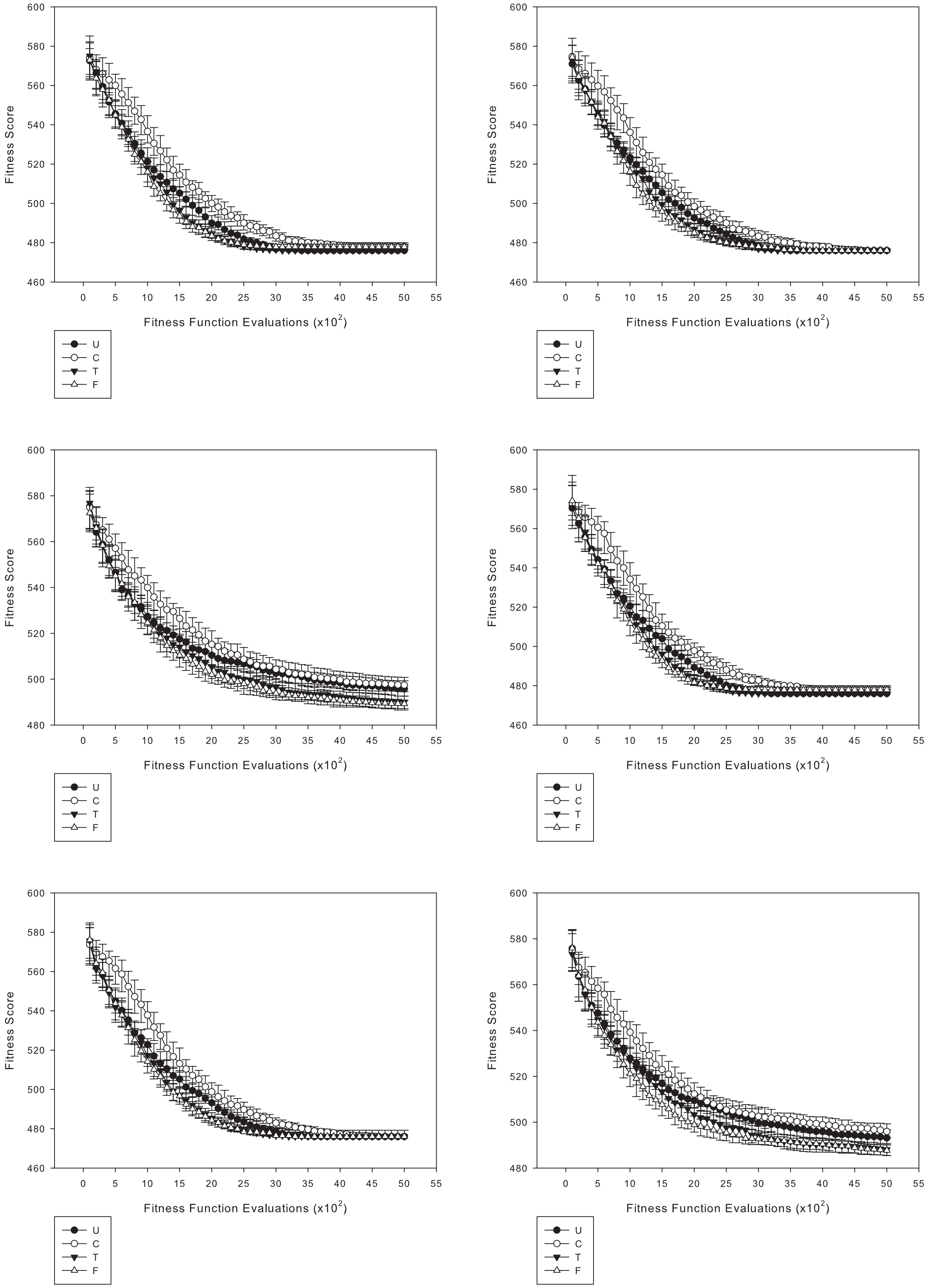
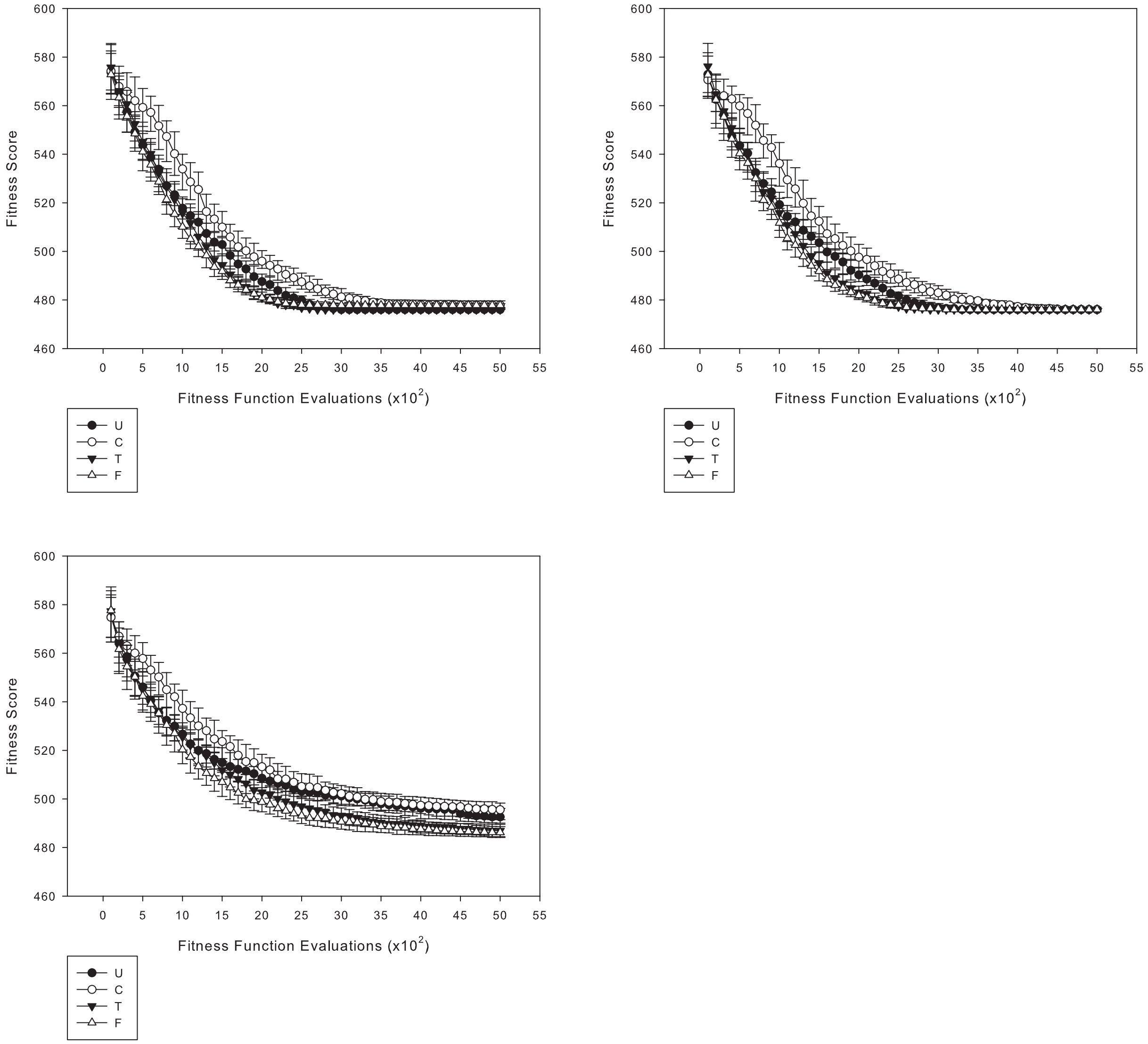


Figure D.10: $N = 150$, p_c 1.0, with p_m 0, 0.005



D.0.13 20 penalty points per intervention

Figure D.11: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

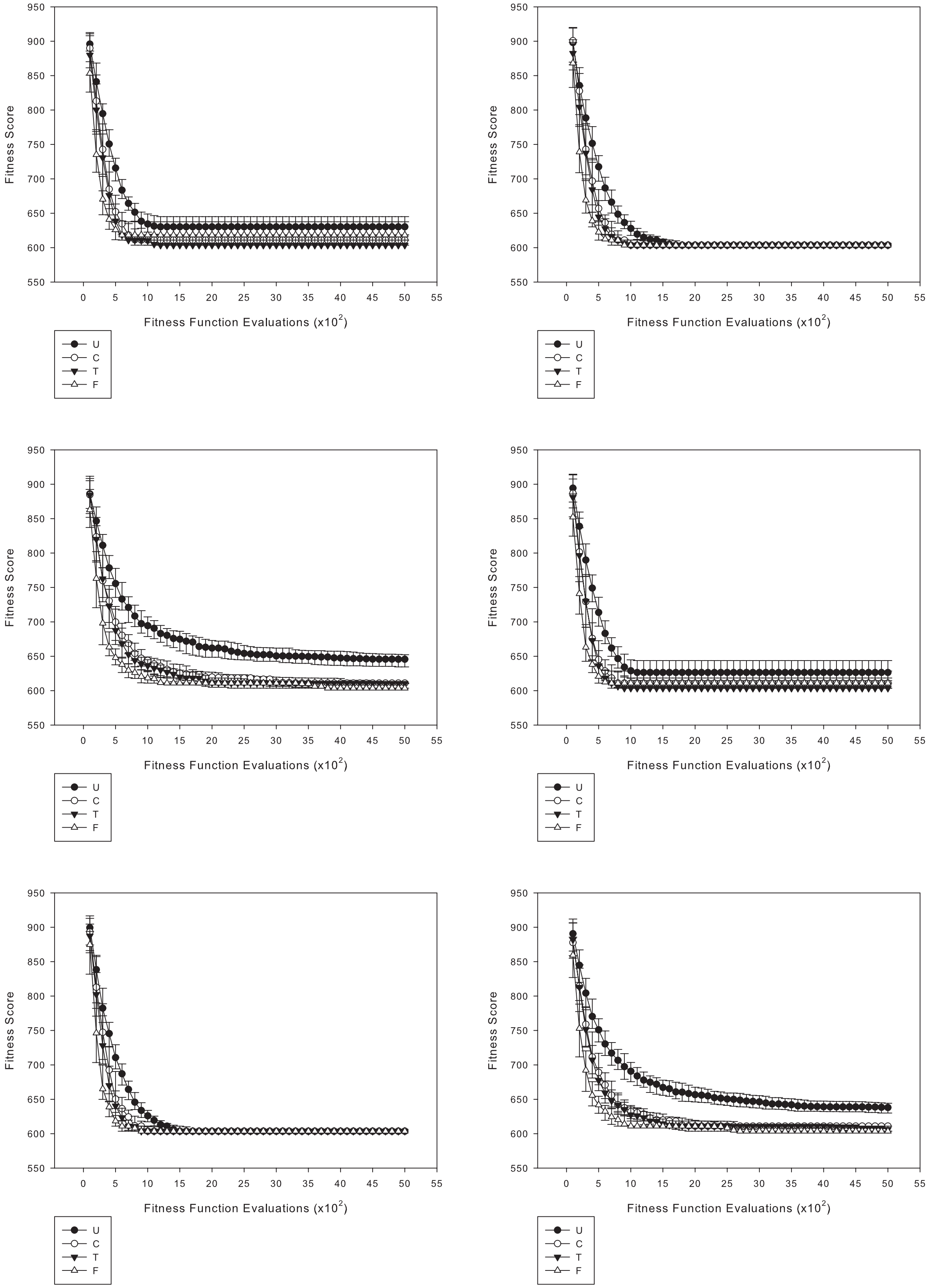


Figure D.12: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

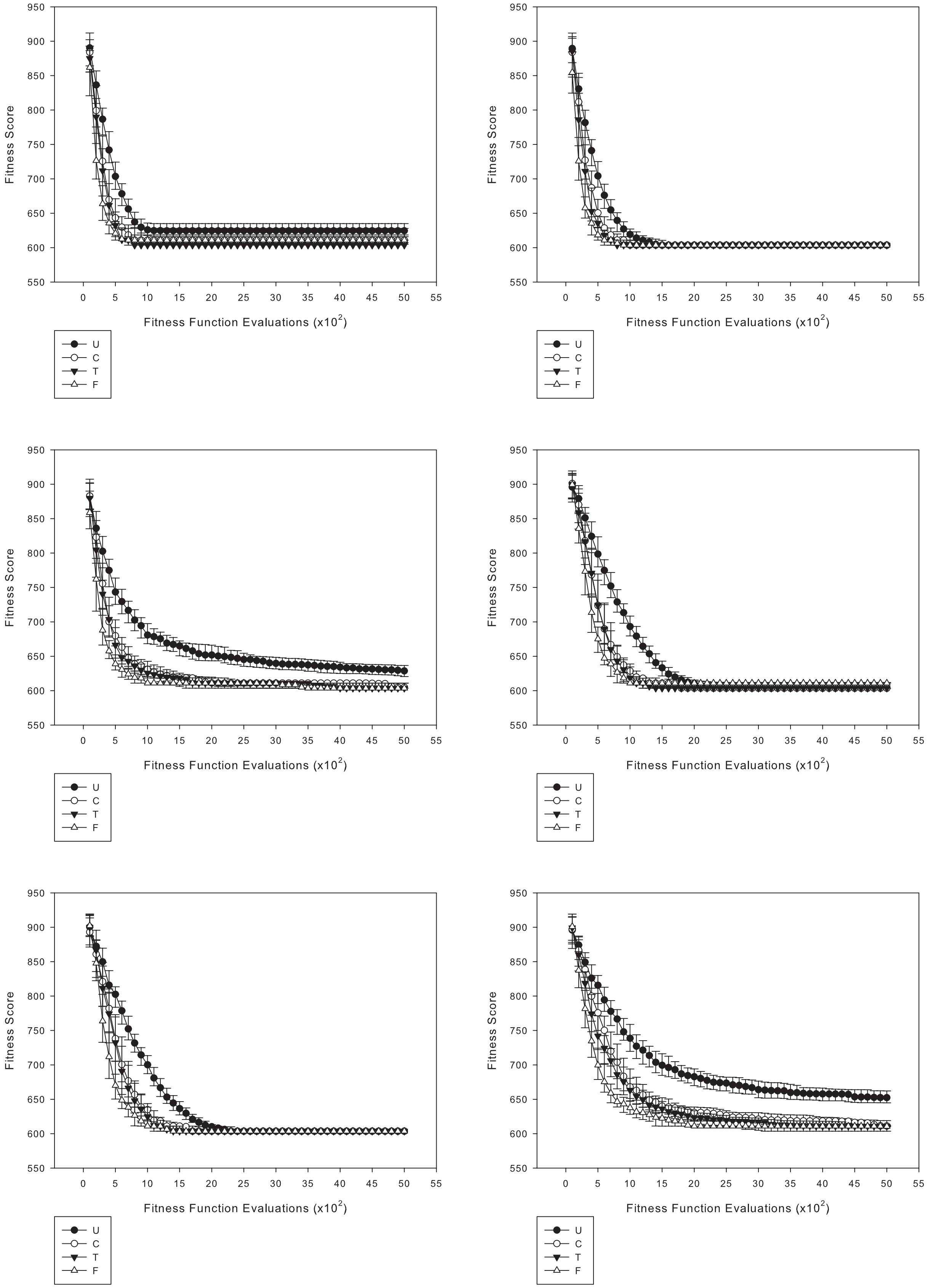


Figure D.13: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

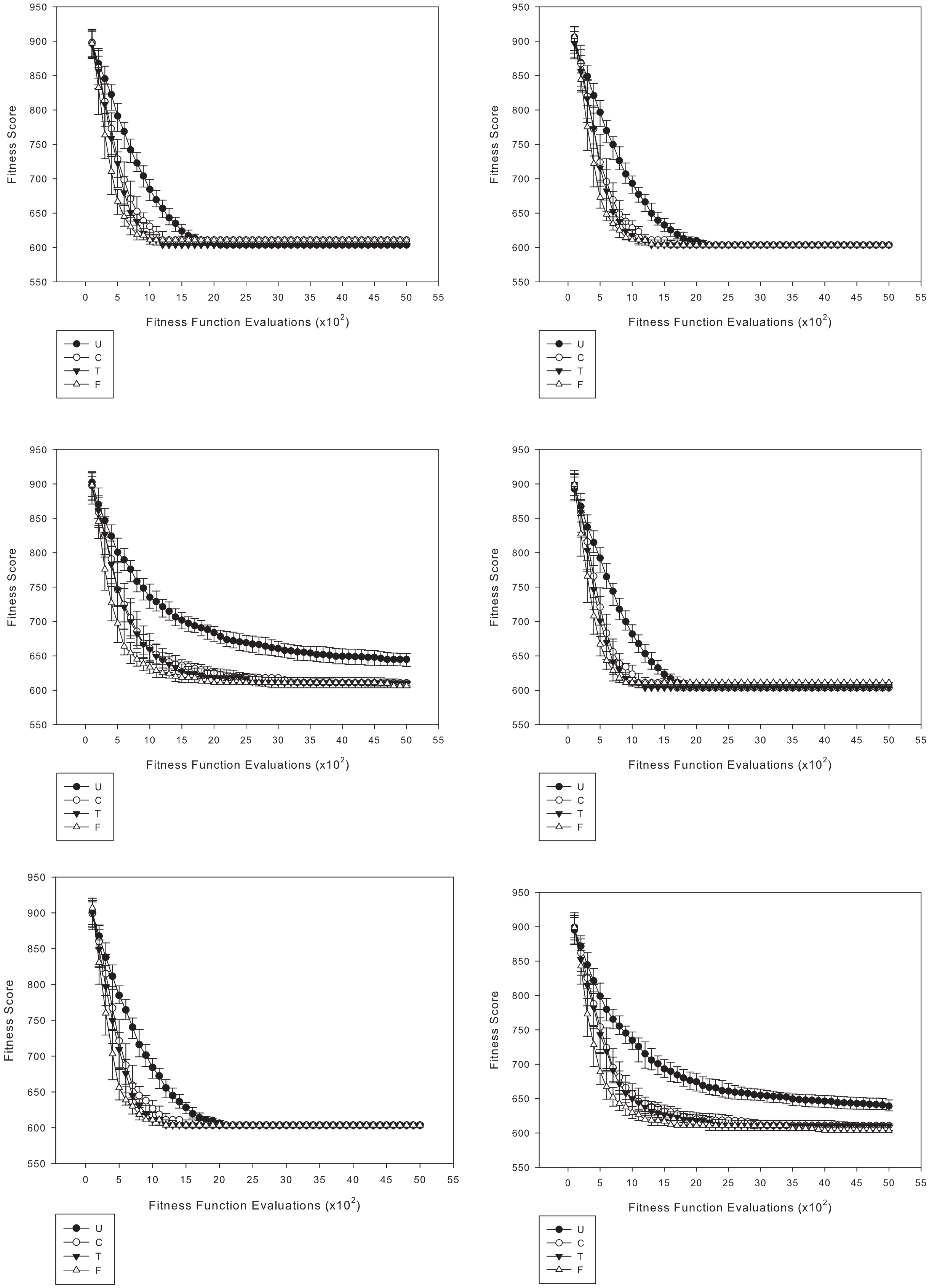


Figure D.14: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

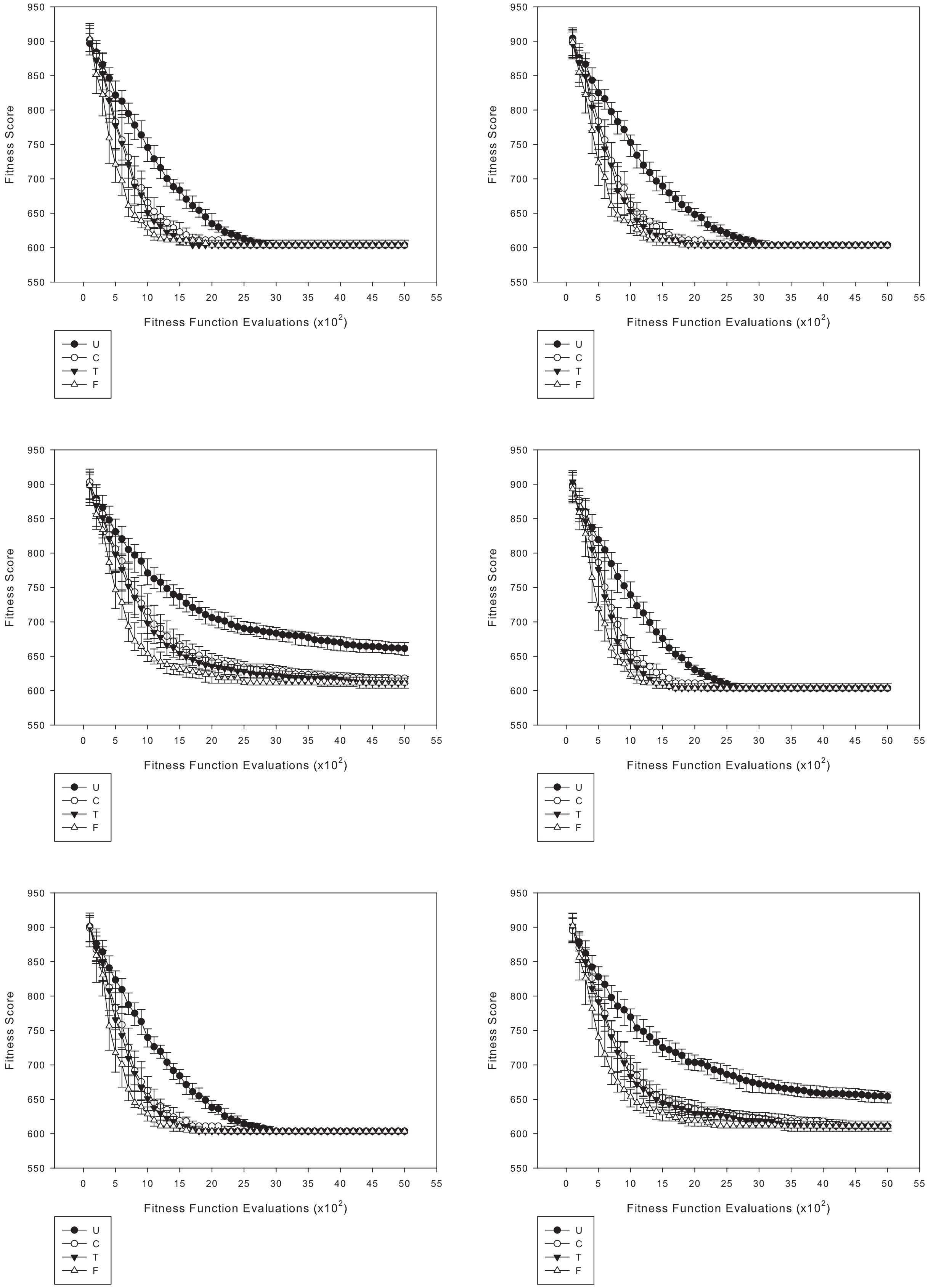
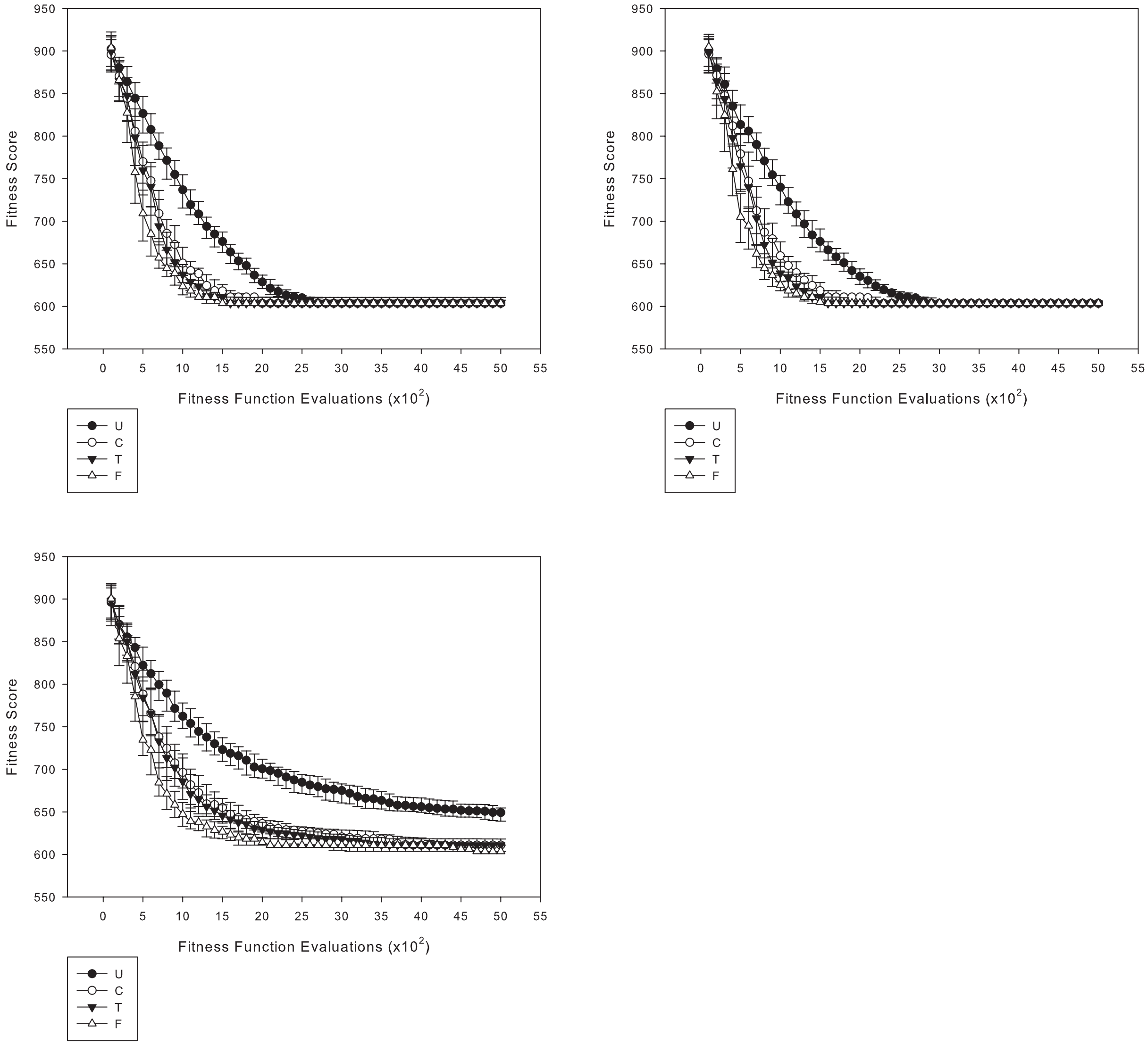


Figure D.15: $N = 150$, p_c 1.0, with p_m 0, 0.005



D.0.14 35 penalty points per intervention

Figure D.16: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

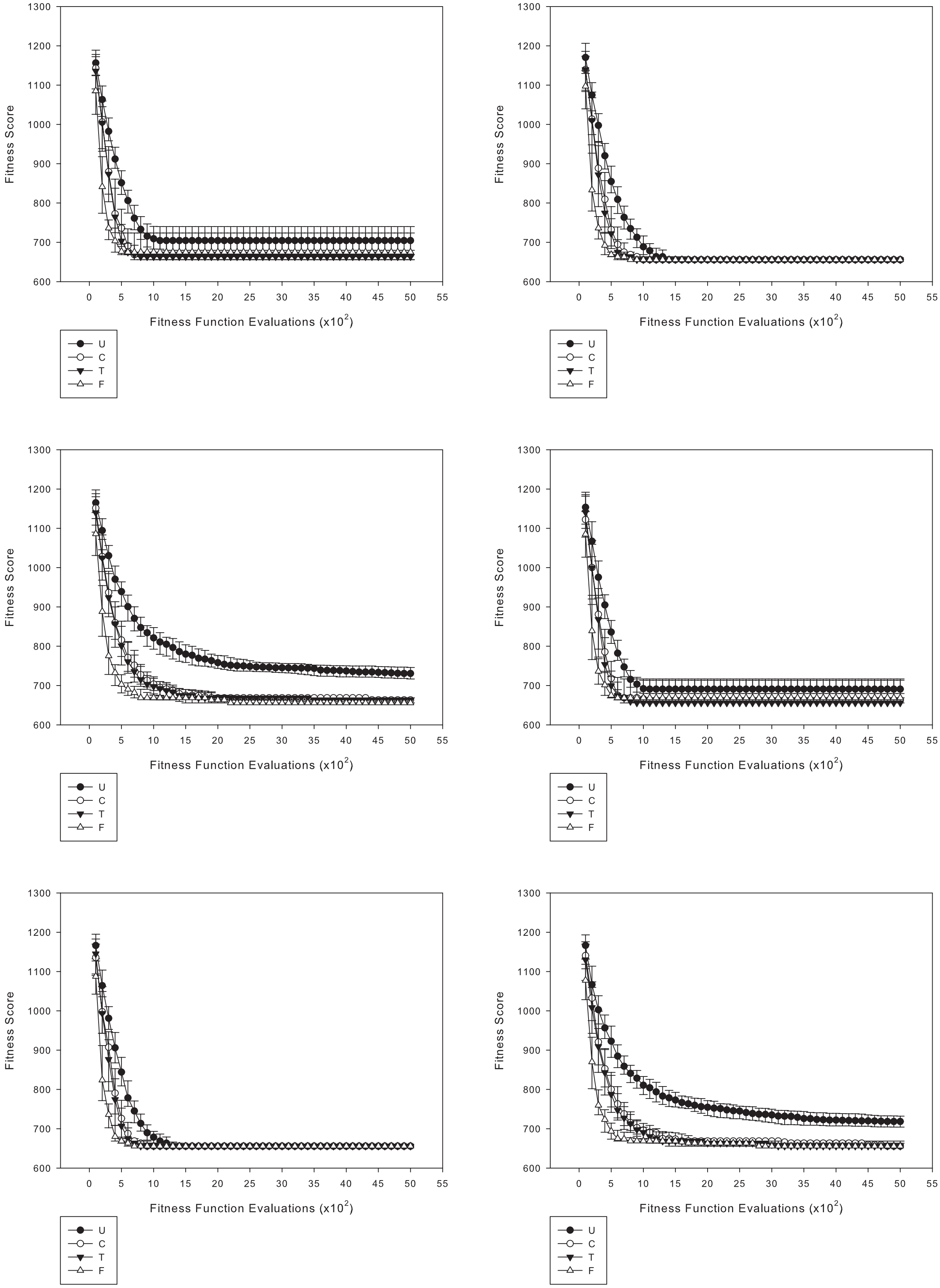


Figure D.17: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

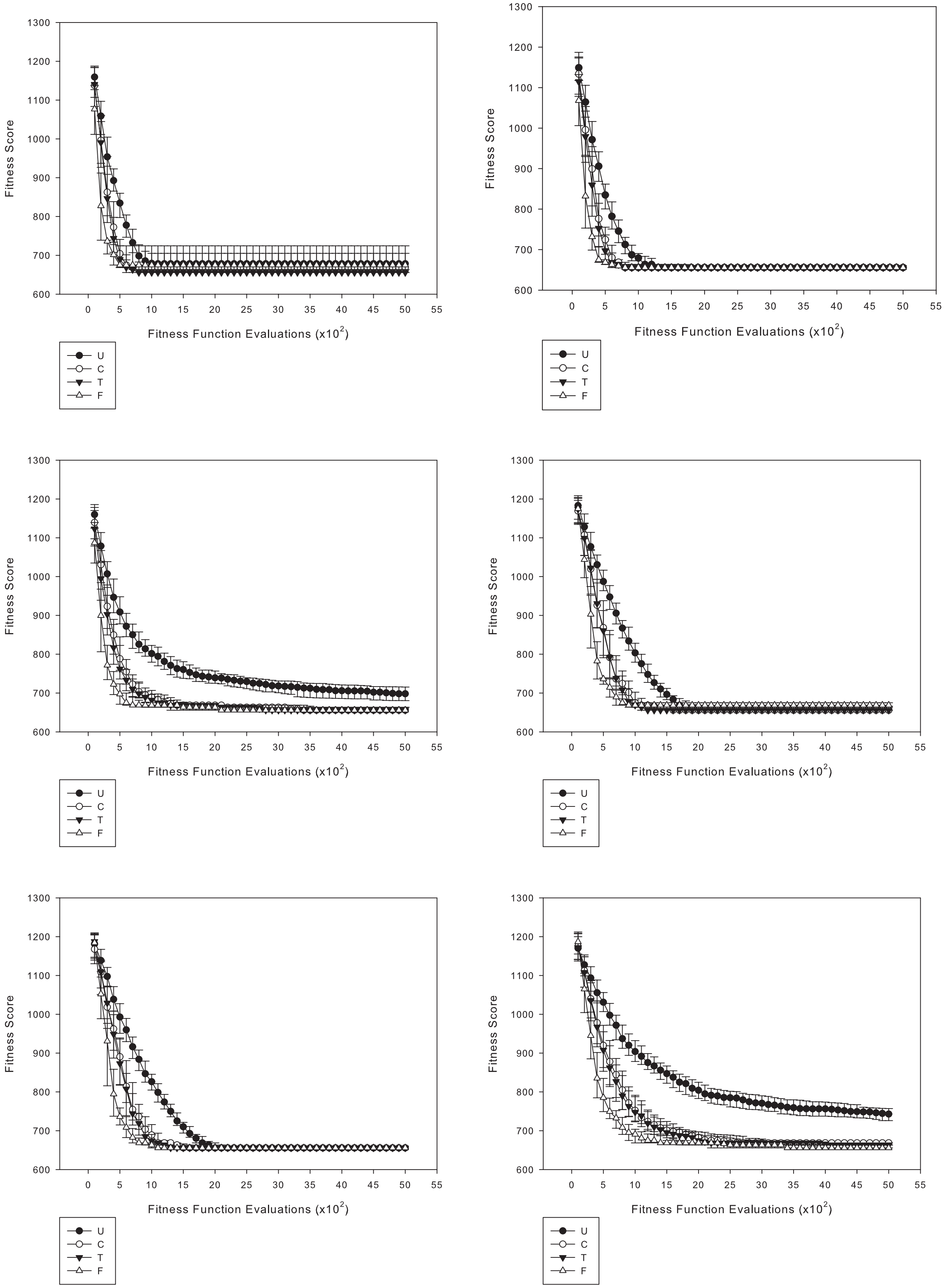


Figure D.18: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

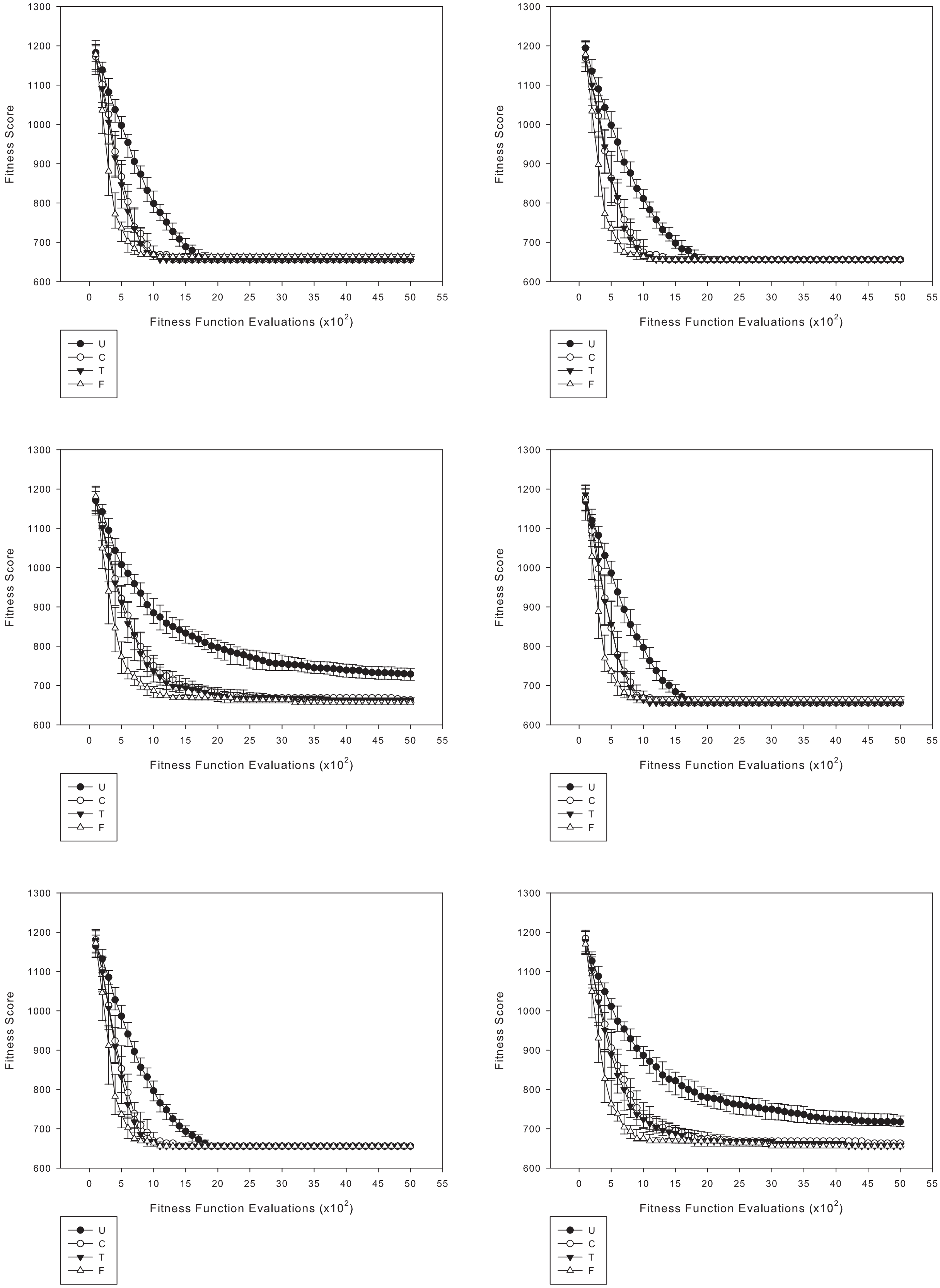


Figure D.19: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

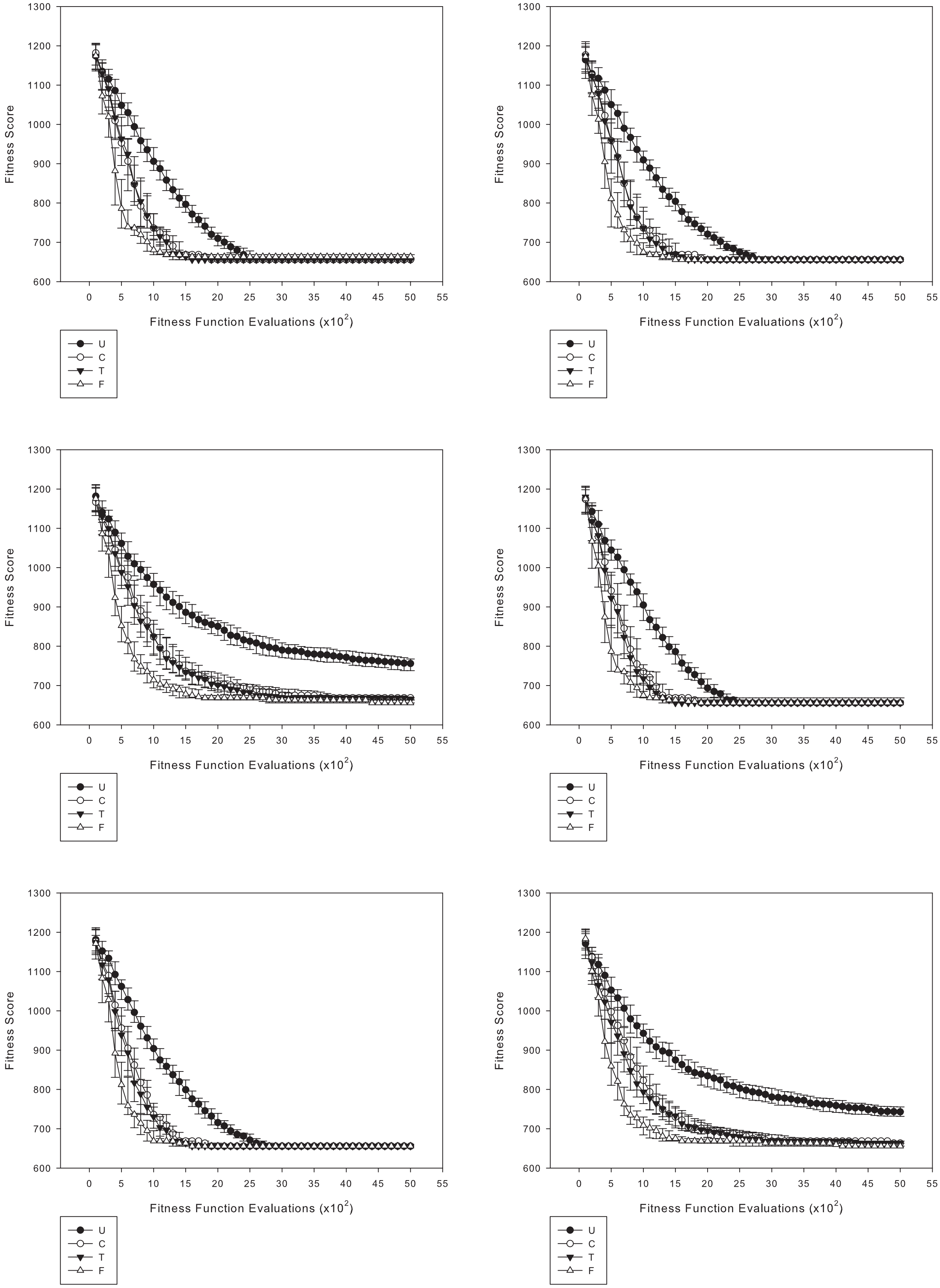
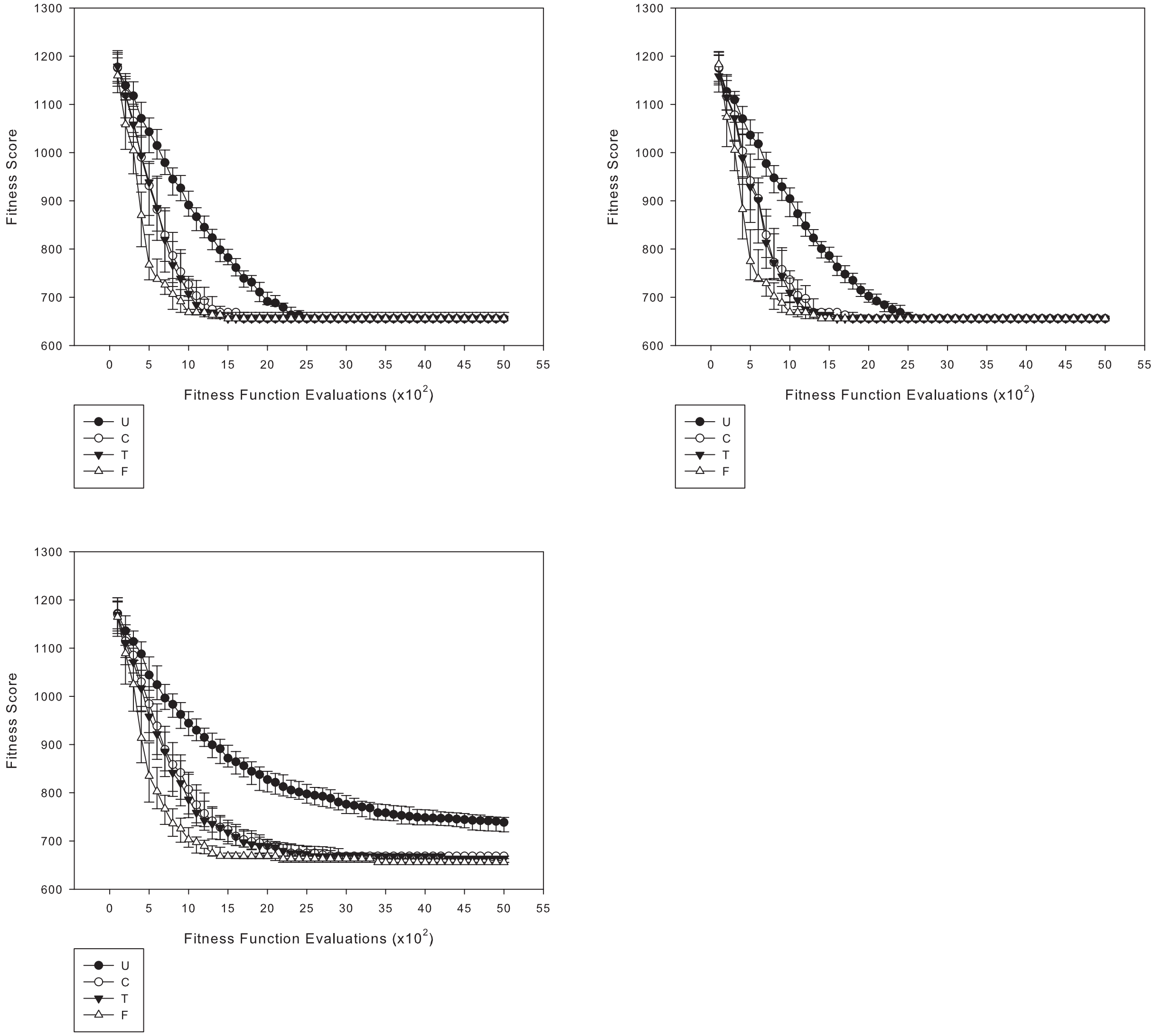


Figure D.20: $N = 150$, p_c 1.0, with p_m 0, 0.005



D.0.15 50 penalty points per intervention

Figure D.21: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 50$, p_c 0.9, with p_m 0, 0.005, 0.05

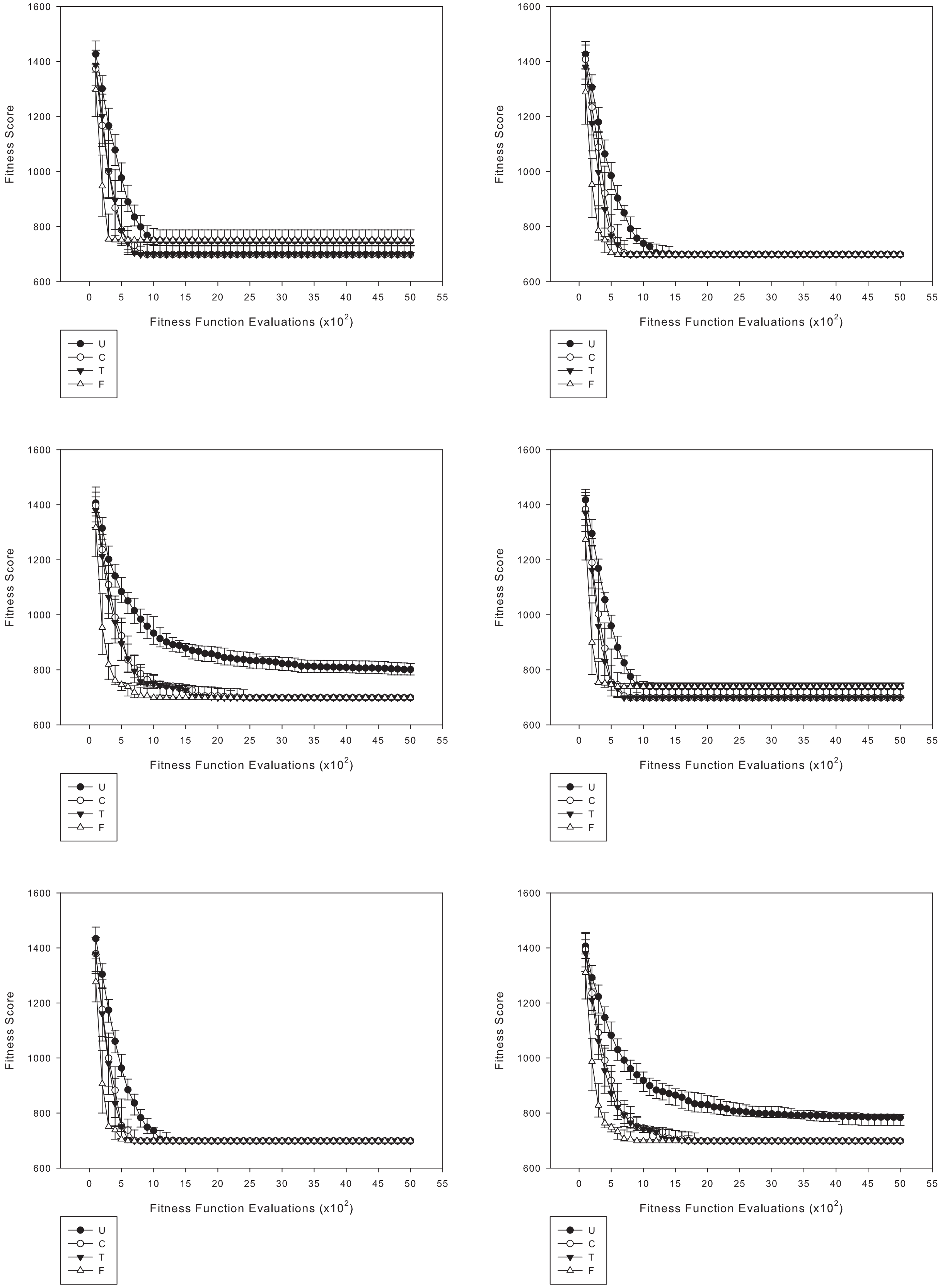


Figure D.22: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 0.8, with p_m 0, 0.005, 0.05

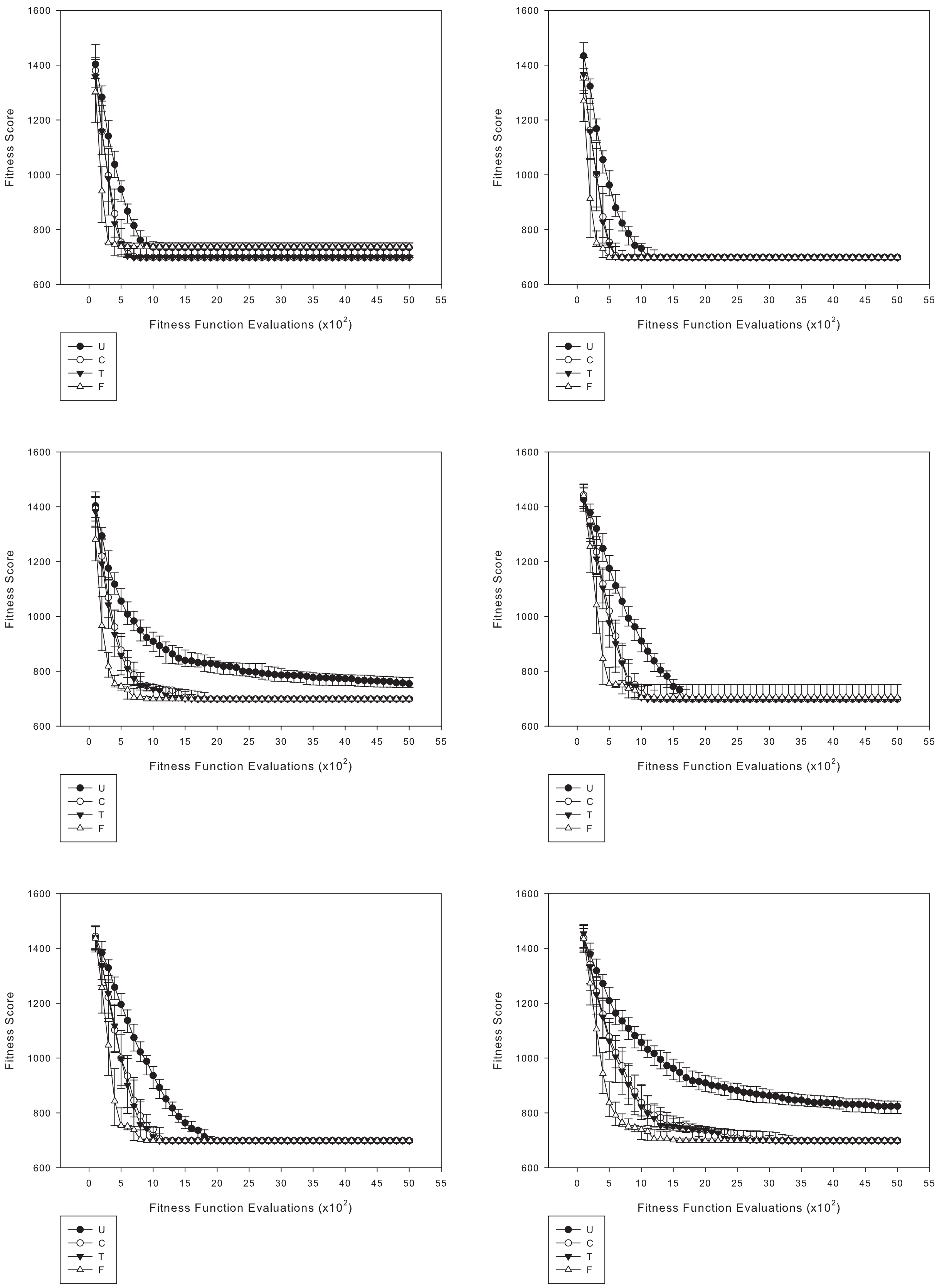


Figure D.23: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by $N = 100$, p_c 1.0, with p_m 0, 0.005, 0.05

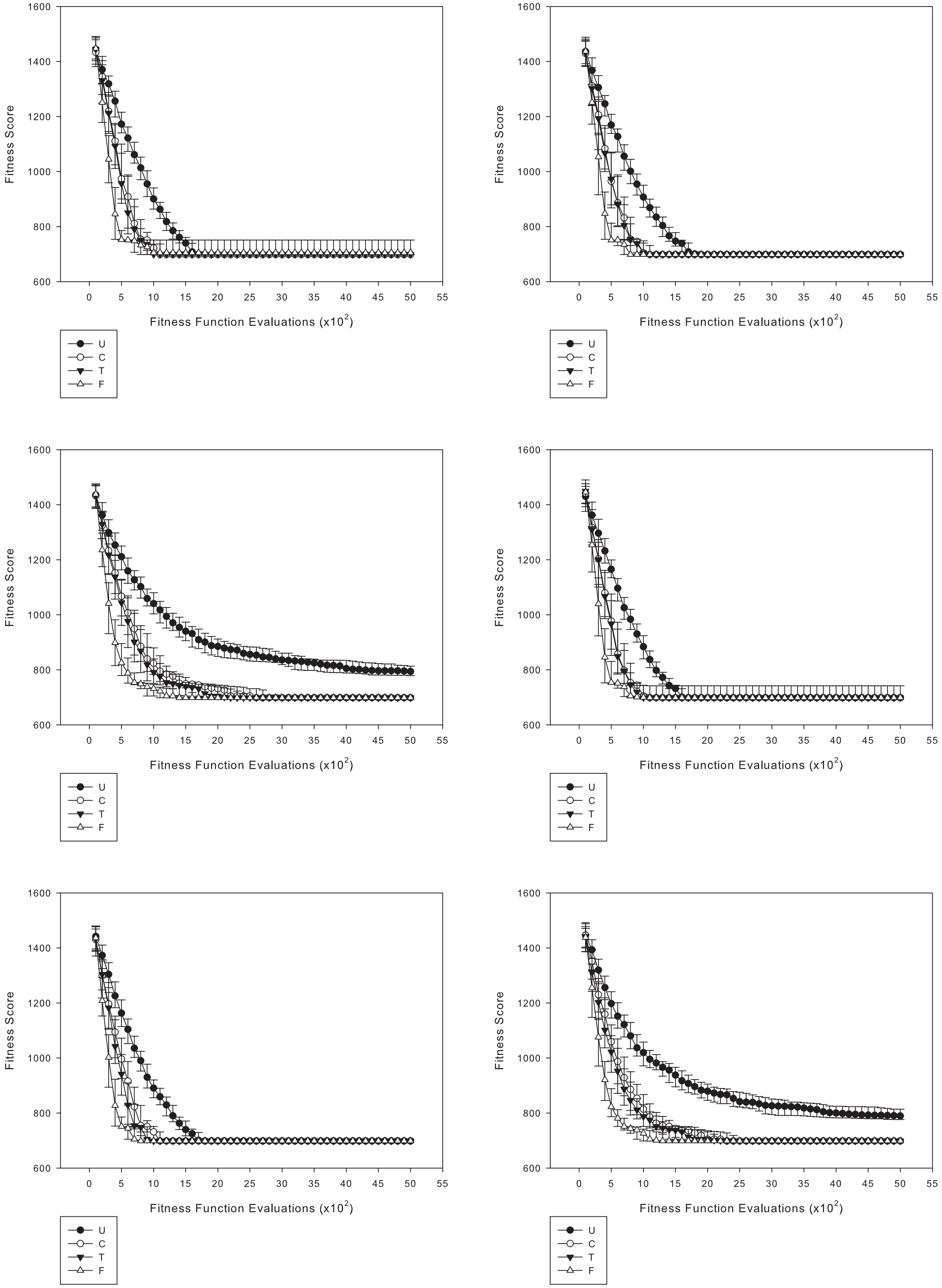


Figure D.24: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by $N = 150$, p_c 0.9, with p_m 0, 0.005, 0.05

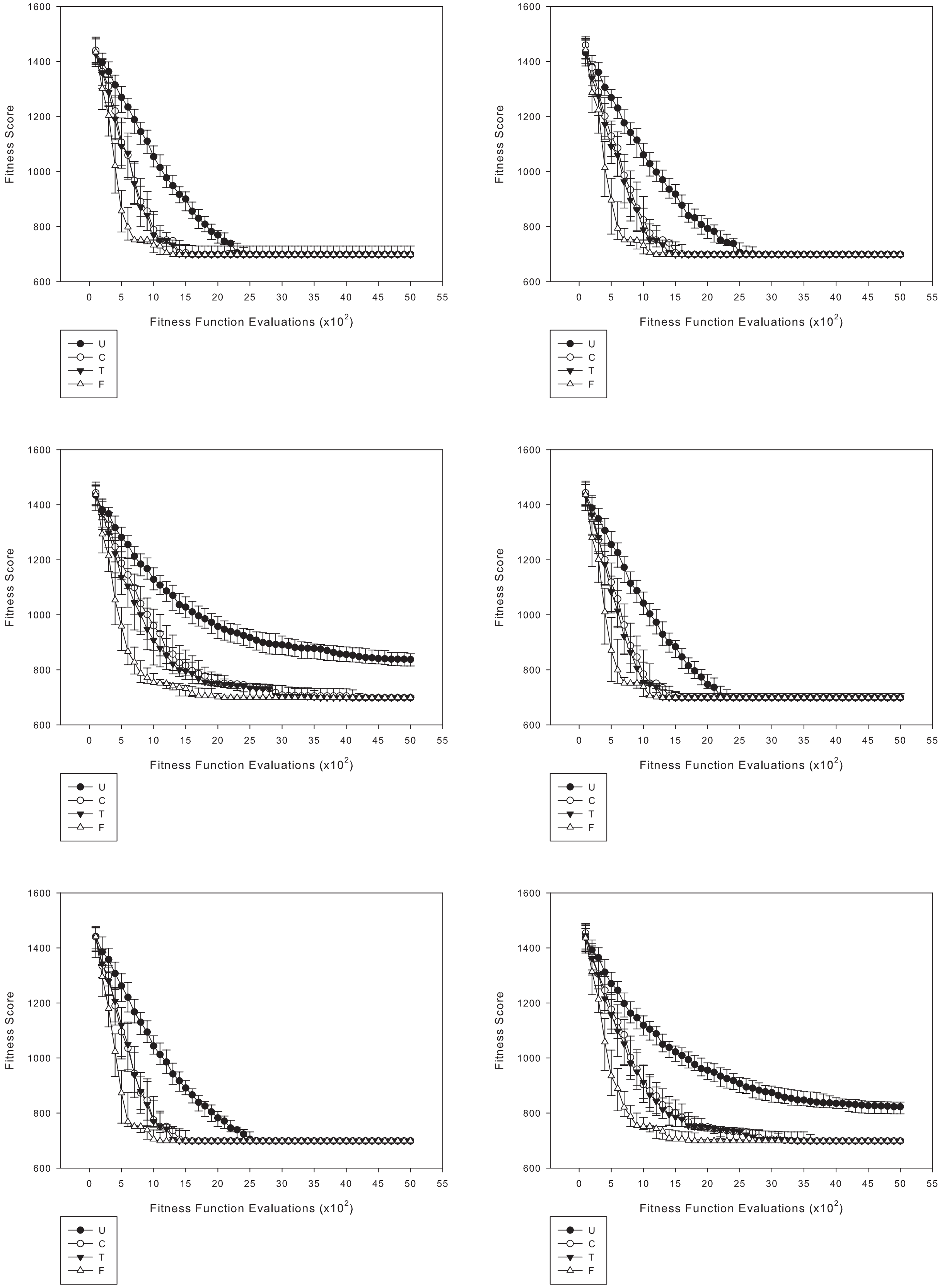
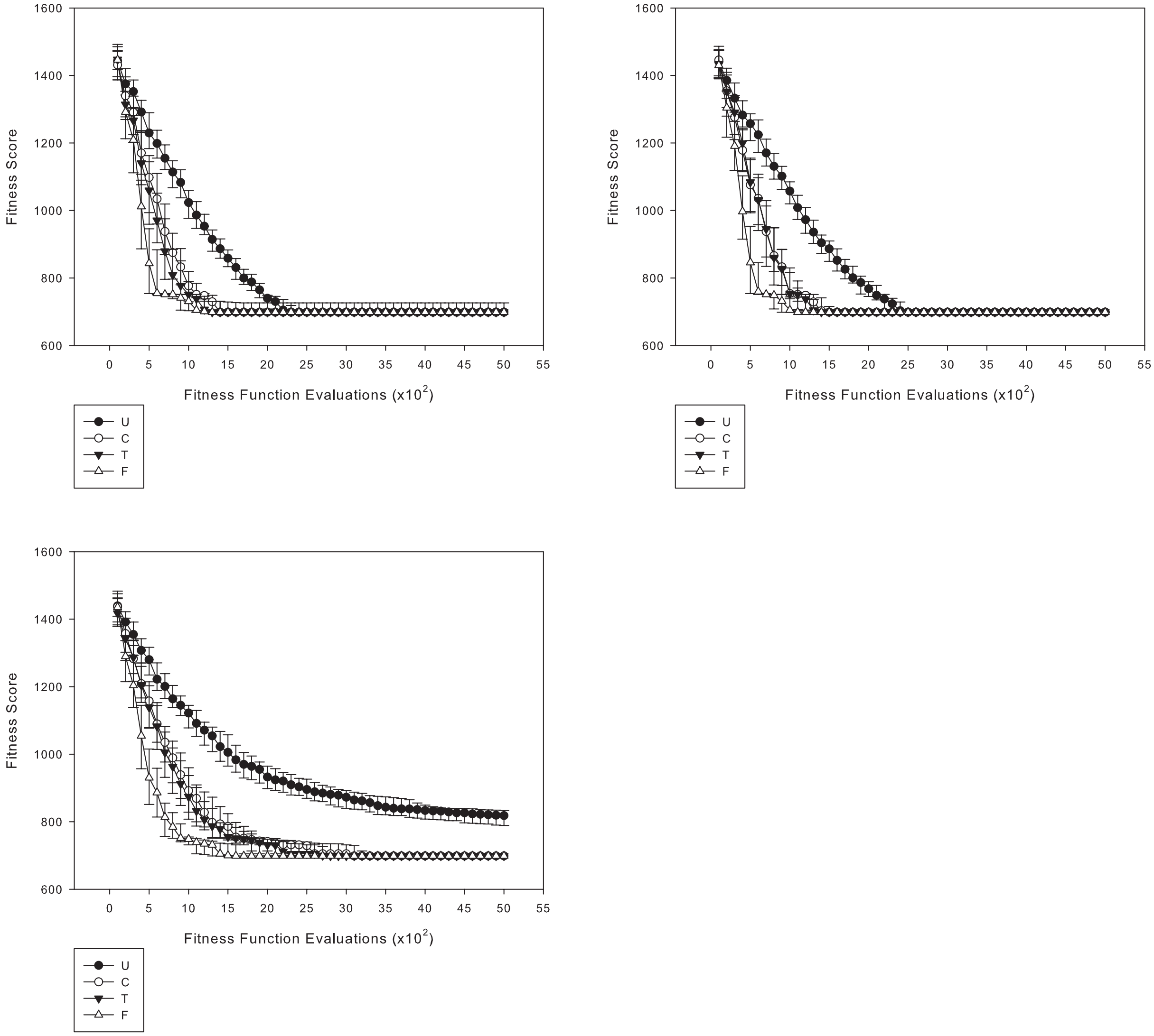


Figure D.25: $N = 150$, p_c 1.0, with p_m 0, 0.005



Appendix E

Kruskal-Wallis one-way analysis of variance for UC, CoFFB, TInSSel and DUC for bio-control optimisation

E.1 KW values for comparison between UC, CoFFB, TInSSel and DUC for the bio-control problem

E.1.1 0 penalty points per infeasible

[illegible]

E.1.2 5 penalty points per infeasible

[illegible]

E.1.3 20 penalty points per infeasible

[illegible]

E.1.4 35 penalty points per infeasible

[illegible]

E.1.5 50 penalty points per infeasible

[illegible]

E.2 KW values for comparison between UC, CoFFB, TInSSel and FDC for the bio-control problem

E.2.1 0 penalty points per infeasible

[illegible]

E.2.2 5 penalty points per infeasible

[illegible]

E.2.3 20 penalty points per infeasible

[illegible]

E.2.4 35 penalty points per infeasible

[illegible]

E.2.5 50 penalty points per infeasible

[illegible]

Appendix F

Single drug cancer chemotherapy scheduling graphs for SPC, 2PC and UC

Figure F.1: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

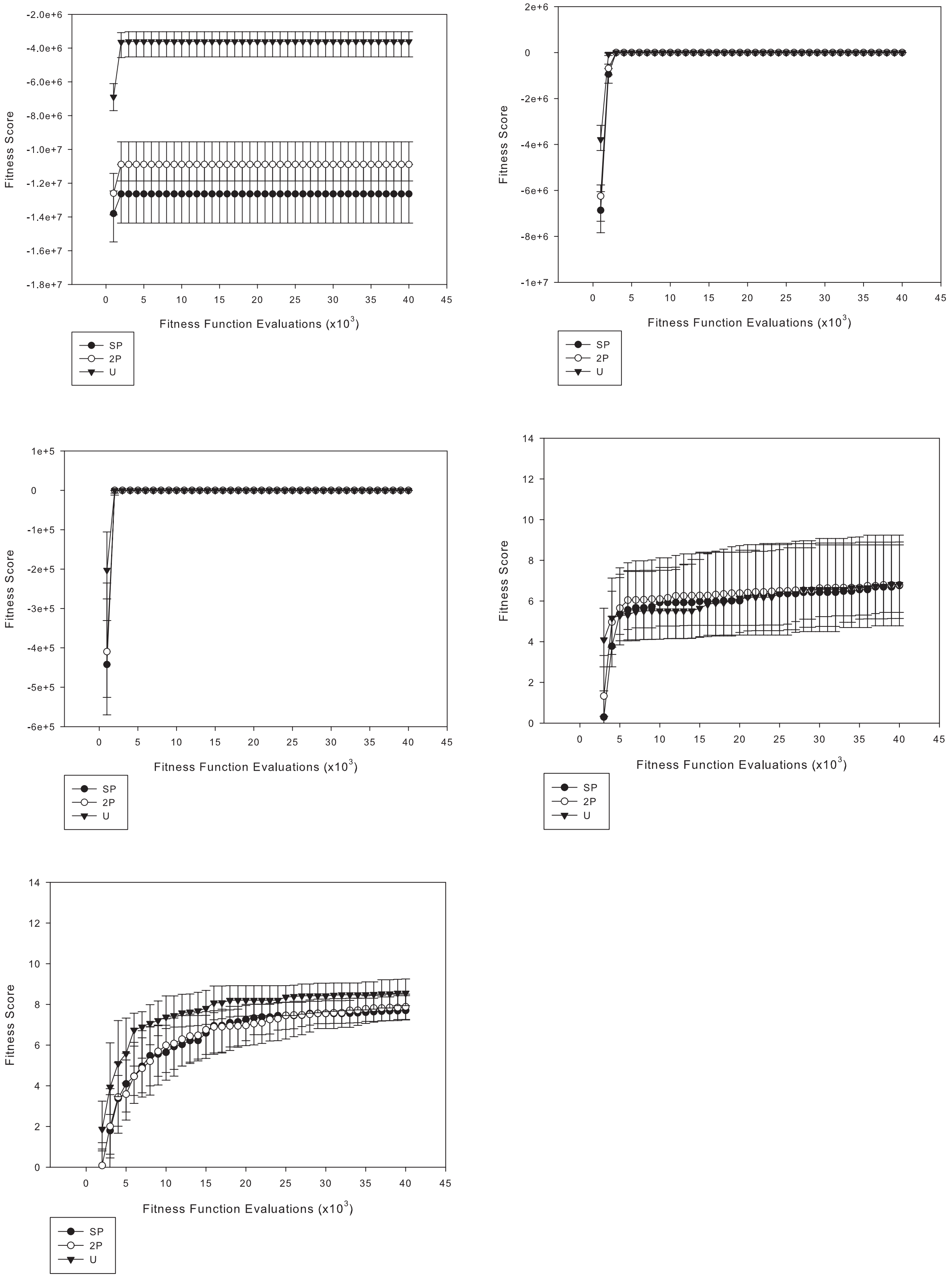


Figure F.2: $N = 50$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

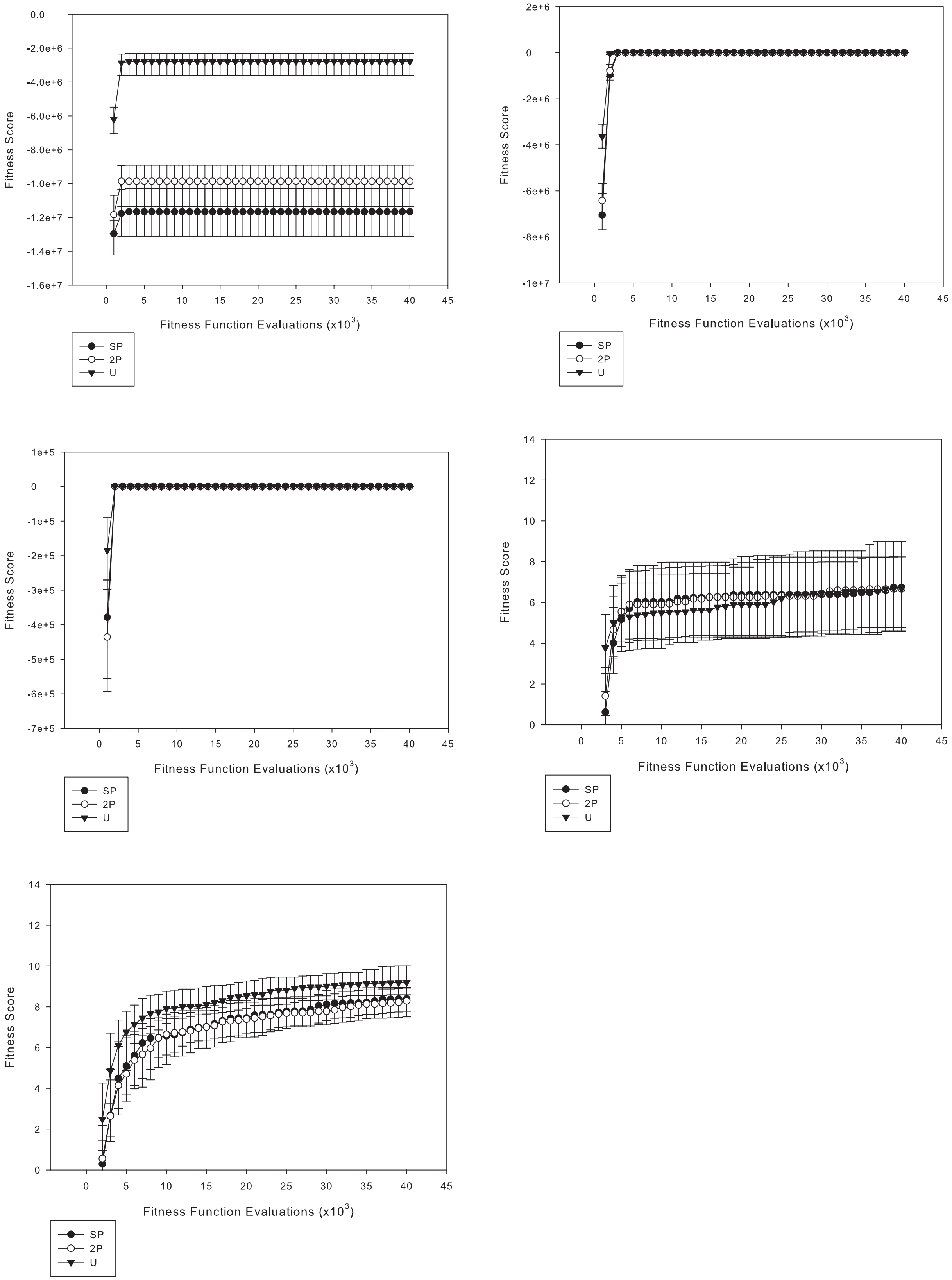


Figure F.3: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

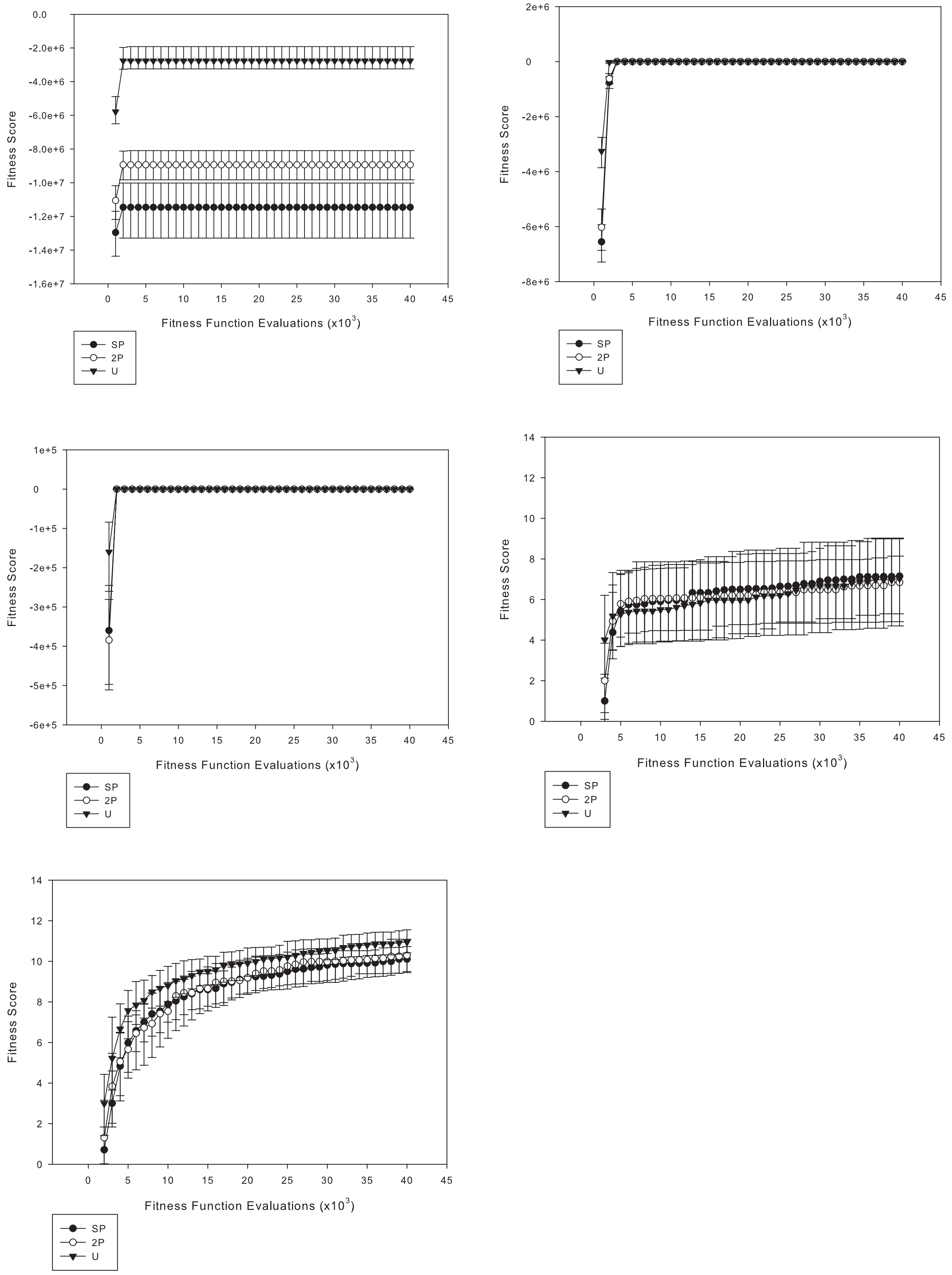


Figure F.4: $N = 100$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

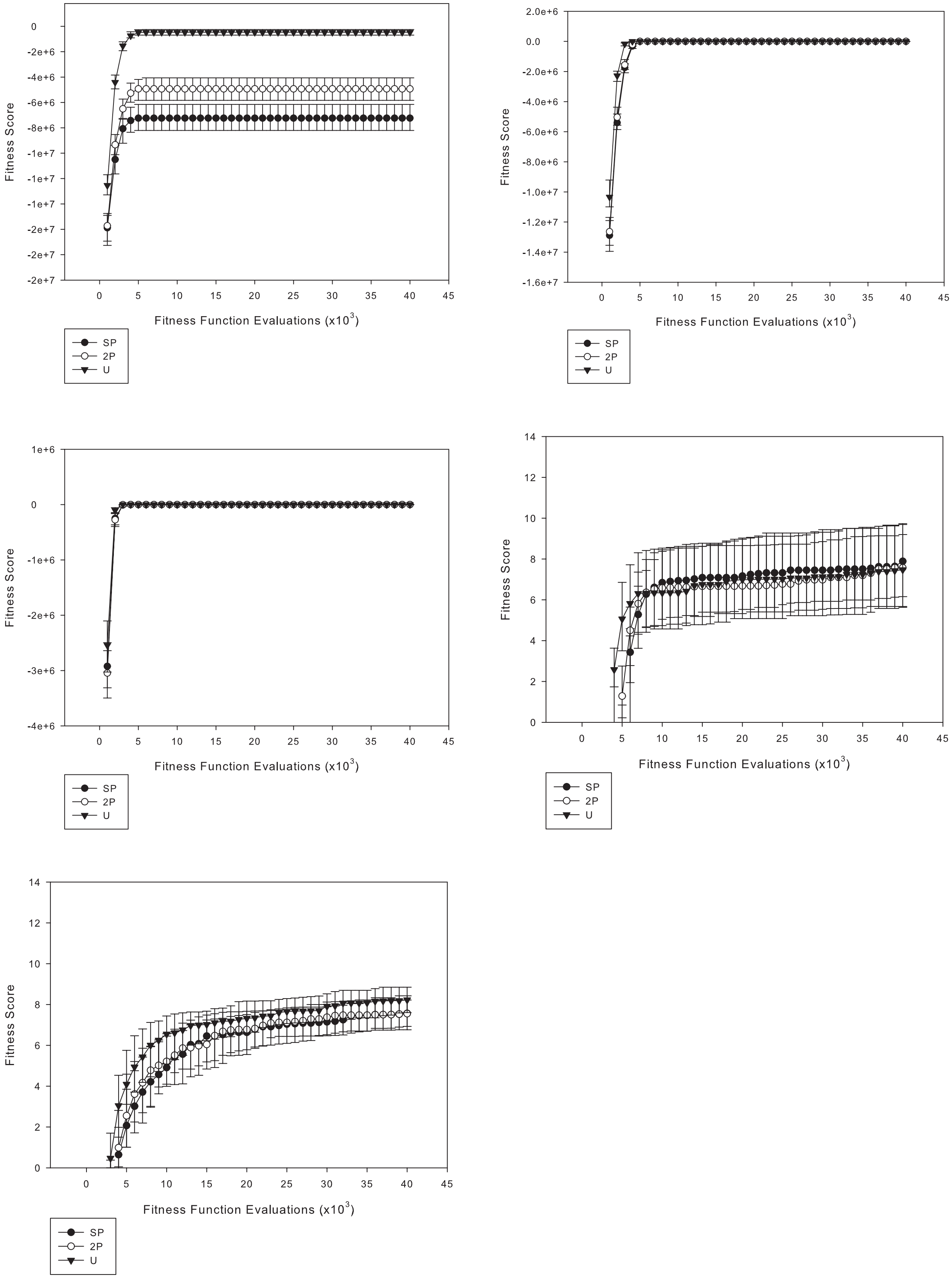


Figure F.5: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

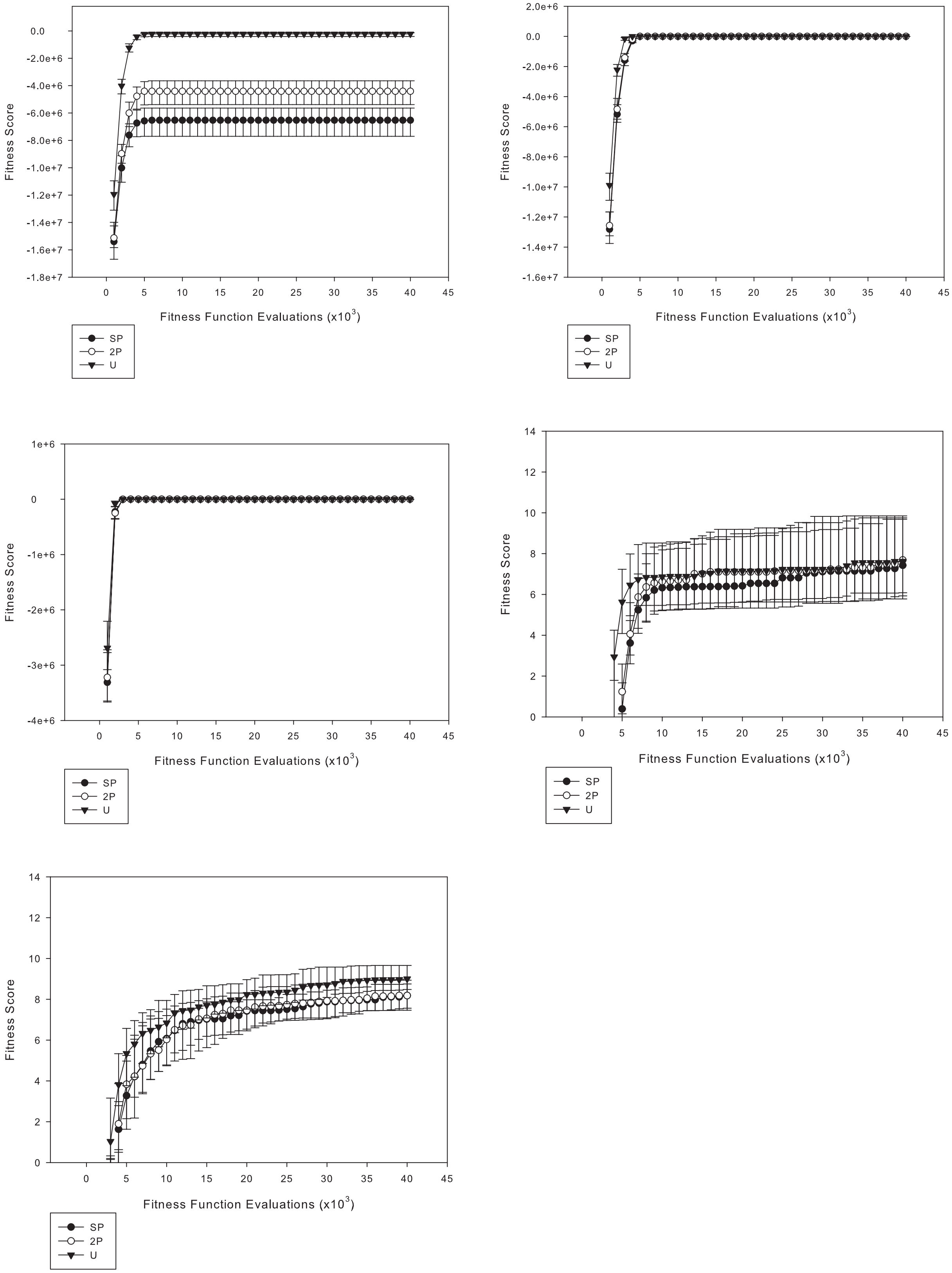


Figure F.6: $N = 100$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

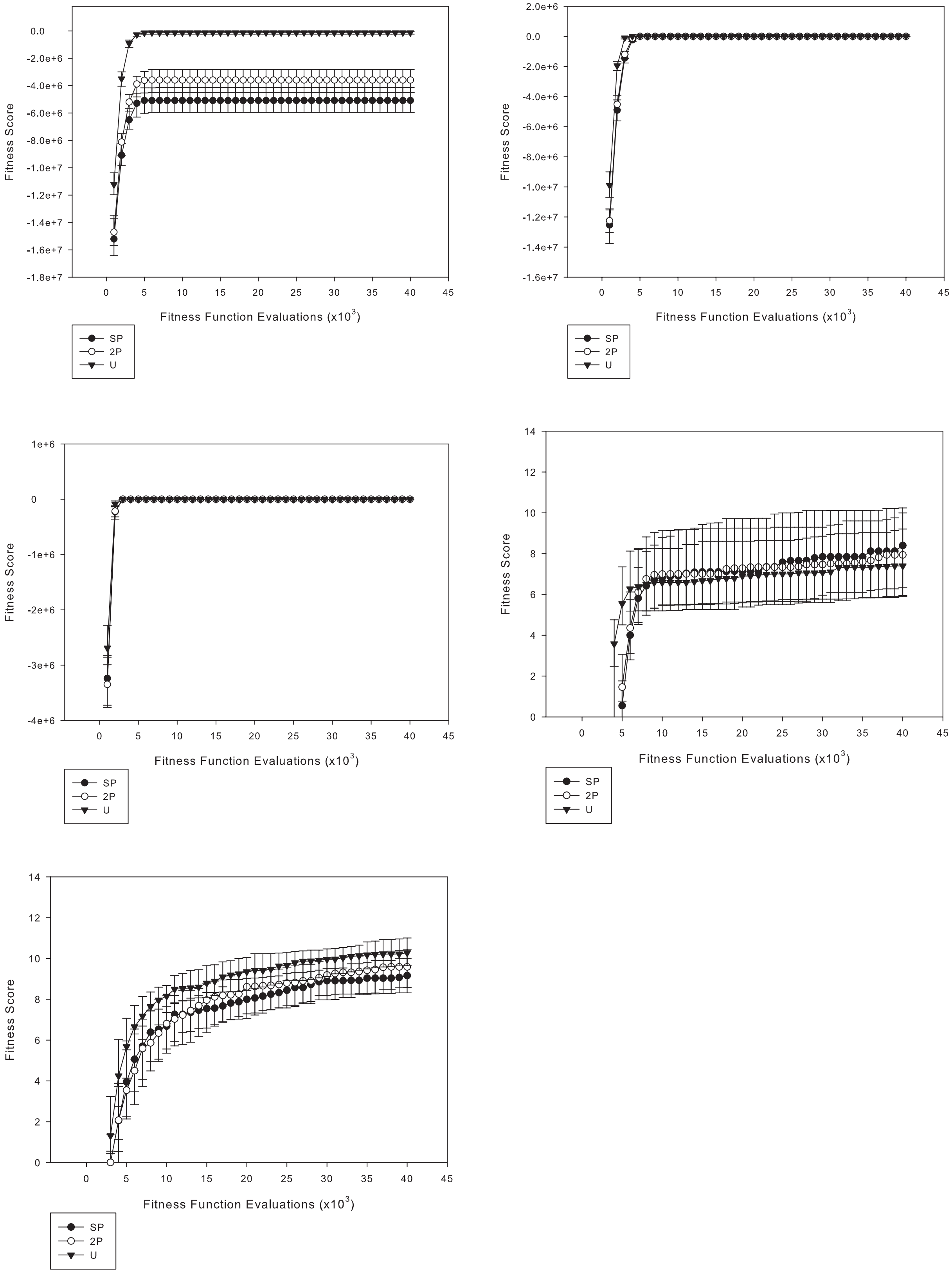


Figure F.7: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

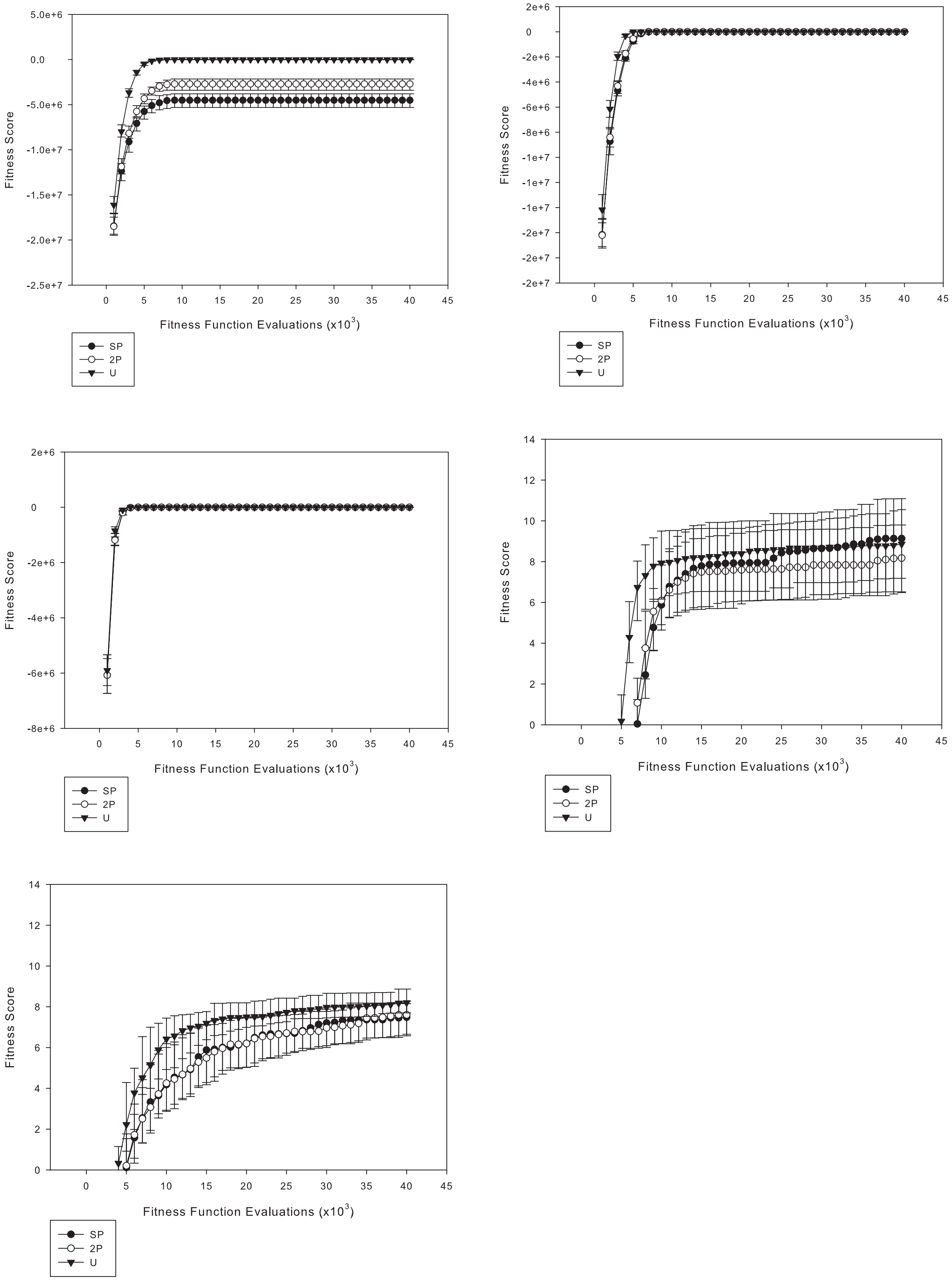


Figure F.8: $N = 150$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

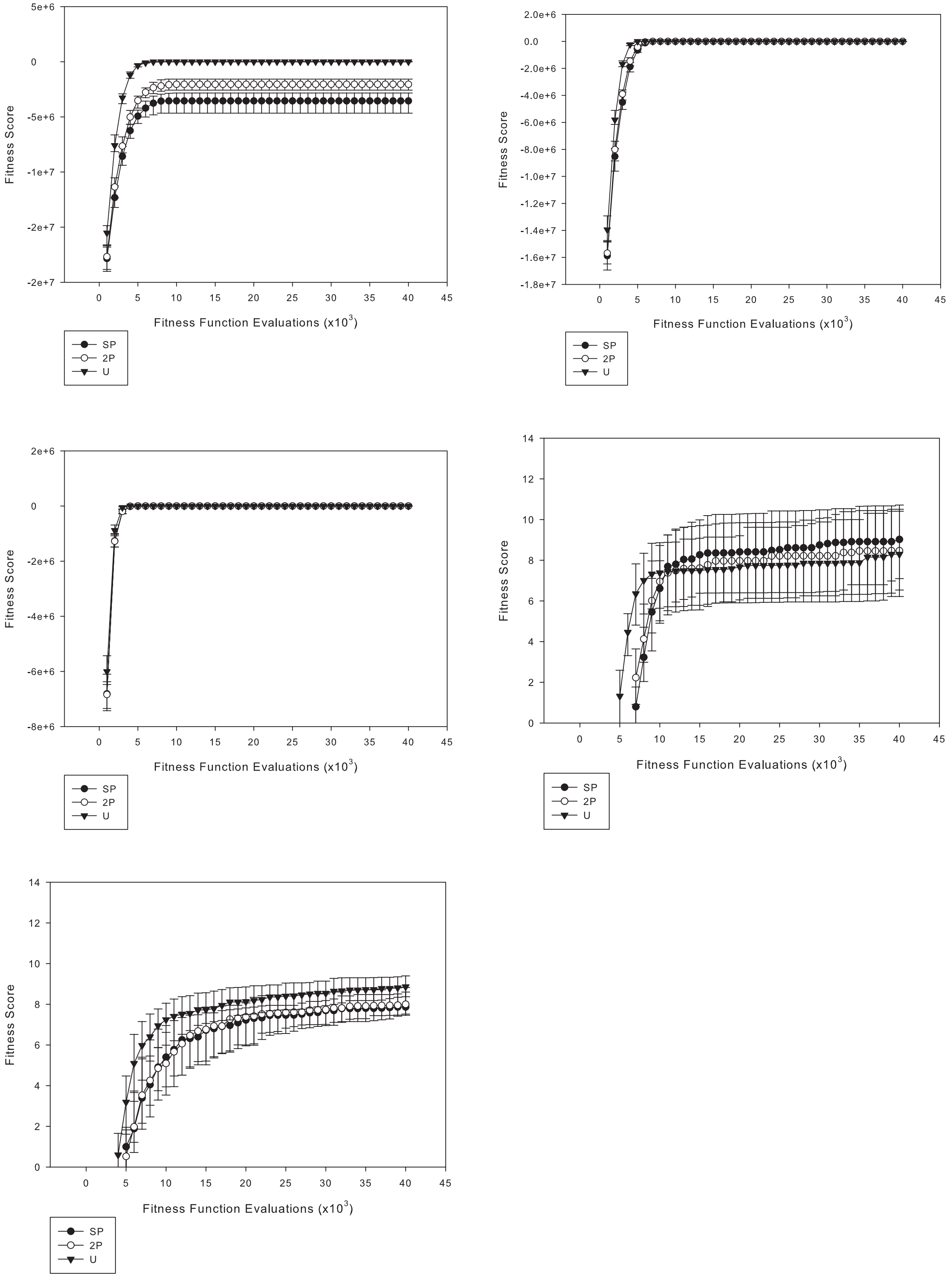
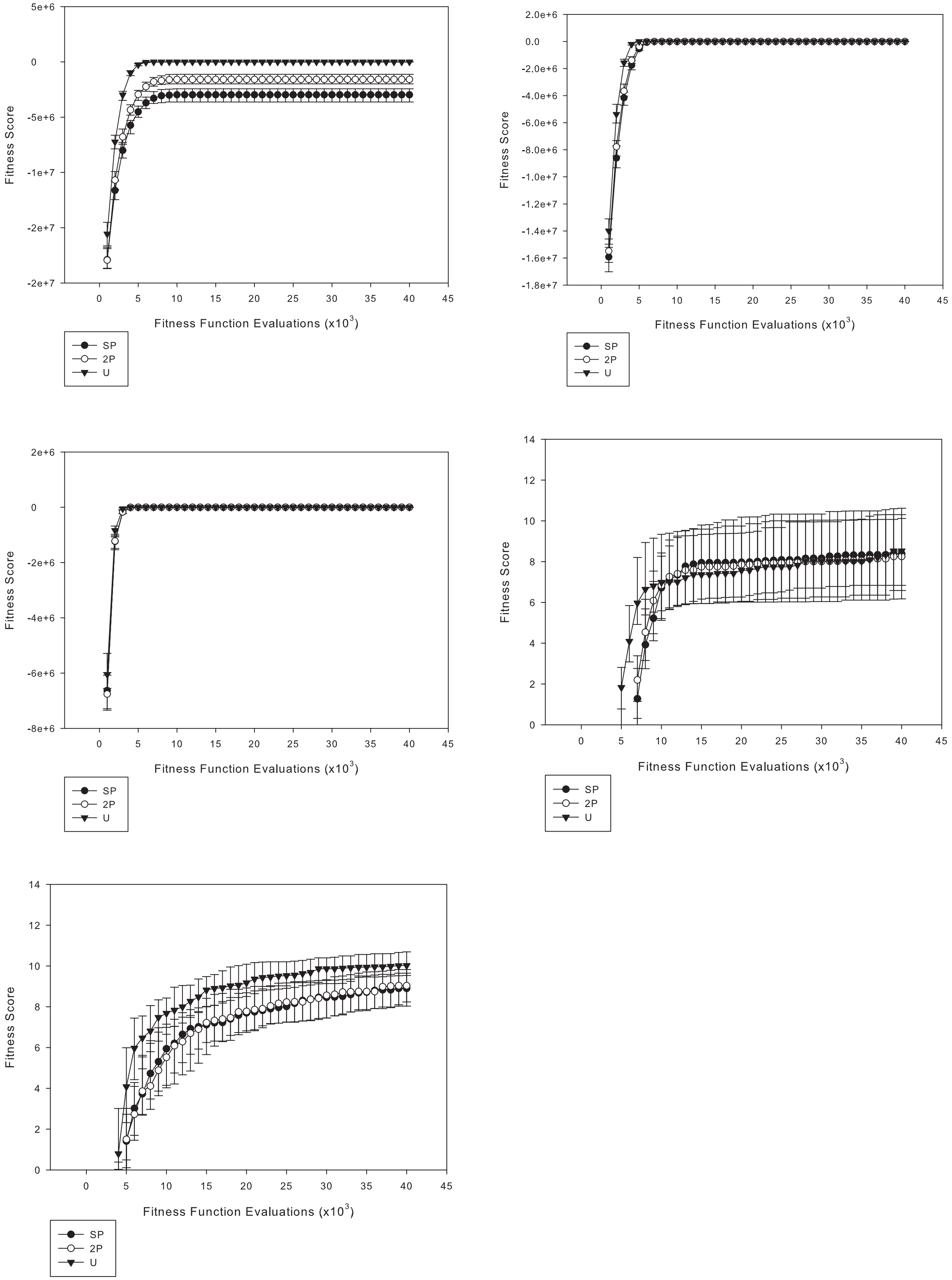


Figure F.9: $N = 150$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05



Appendix G

Single drug cancer chemotherapy scheduling graphs for UC, CalEB, TInSSel and DUC

Figure G.1: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

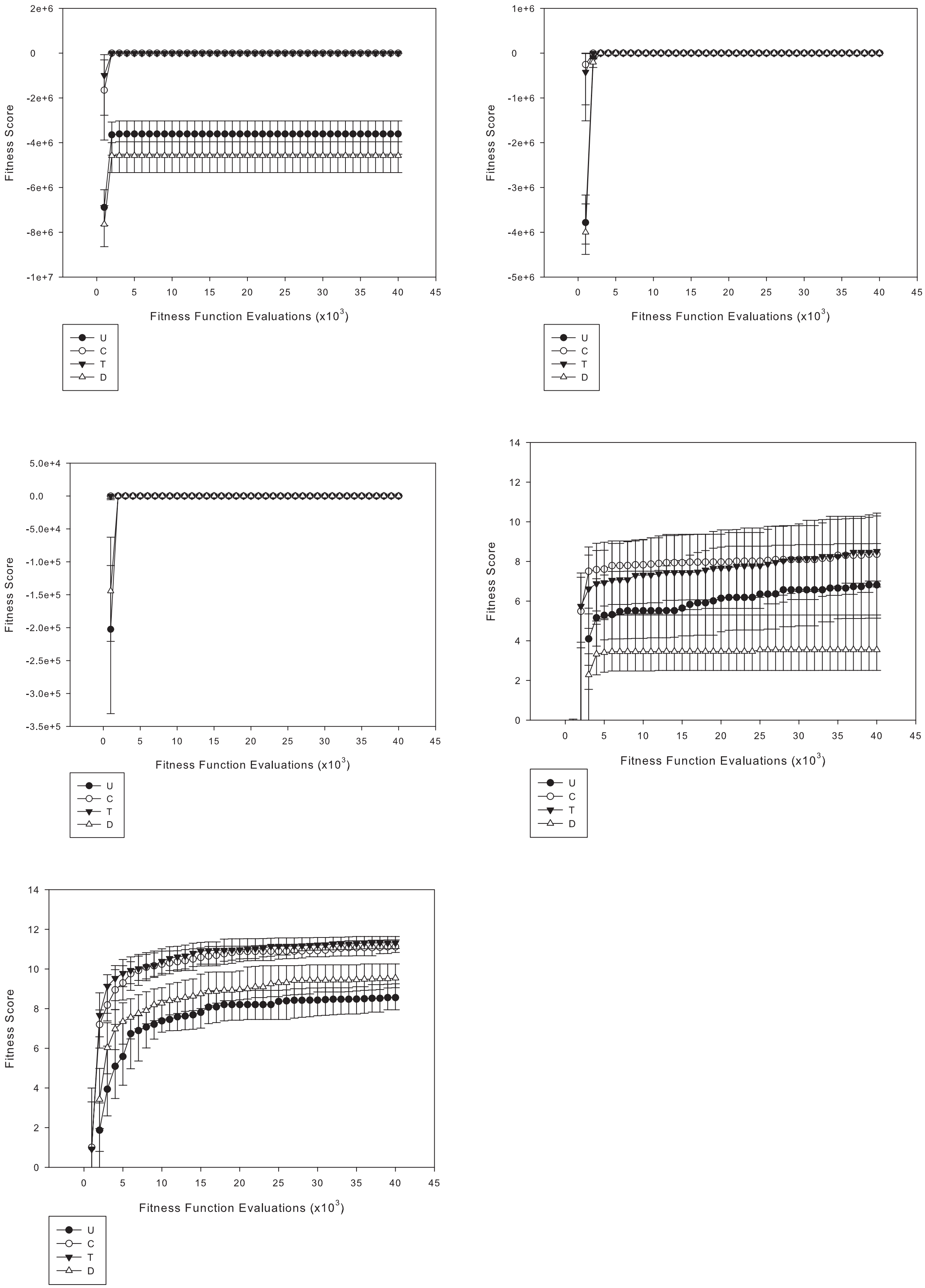


Figure G.2: $N = 50$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

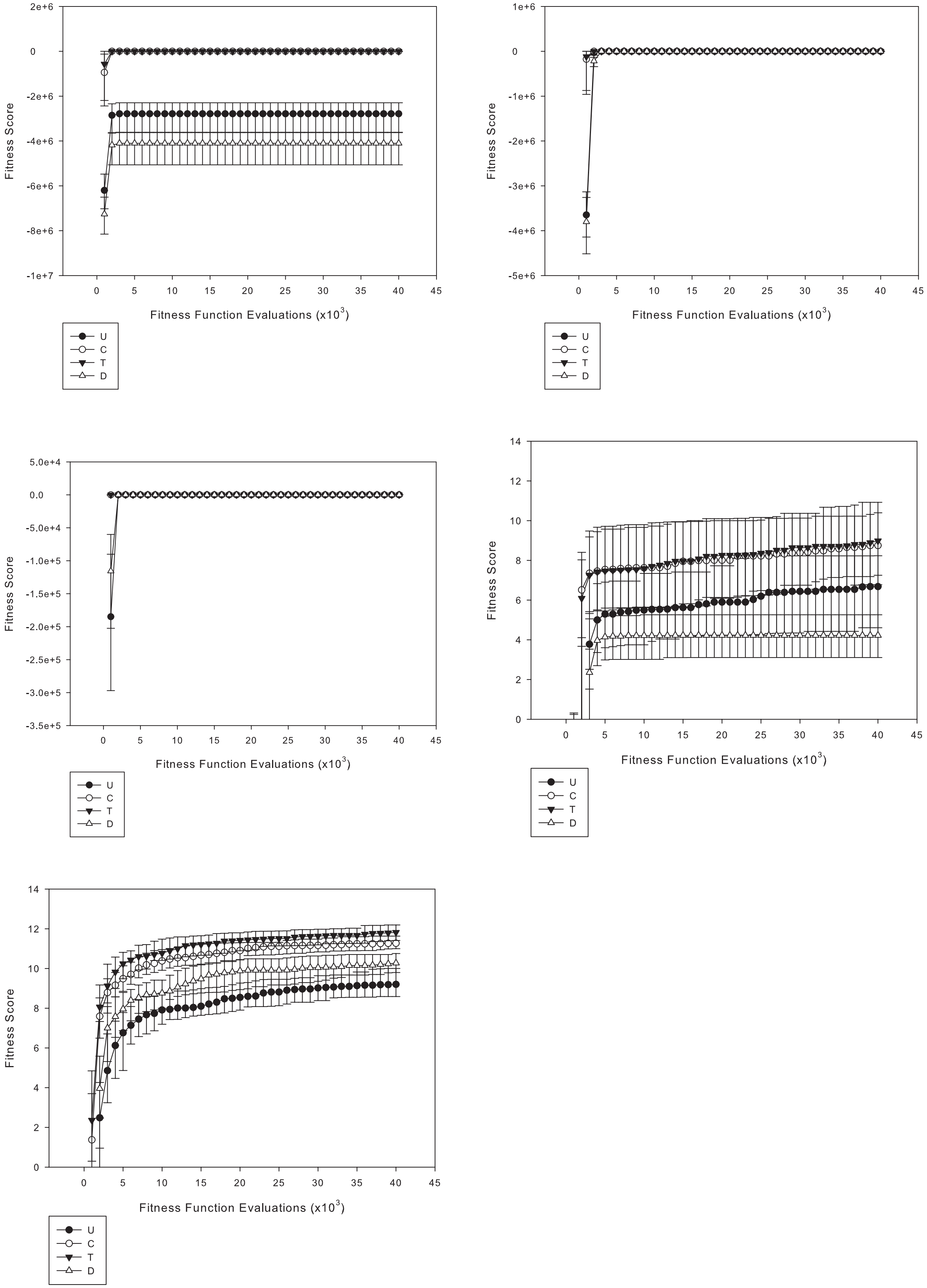


Figure G.3: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

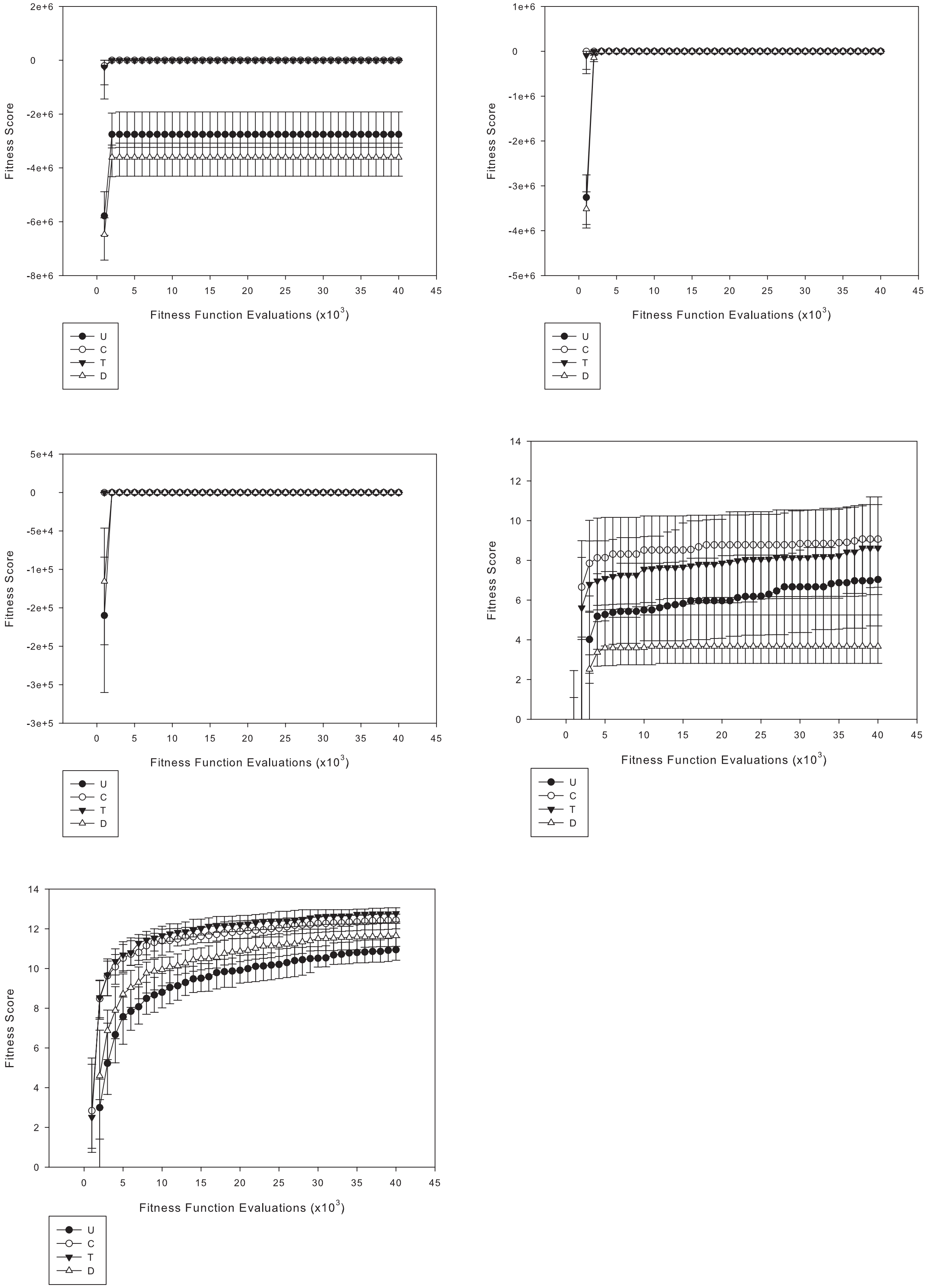


Figure G.4: $N = 100$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

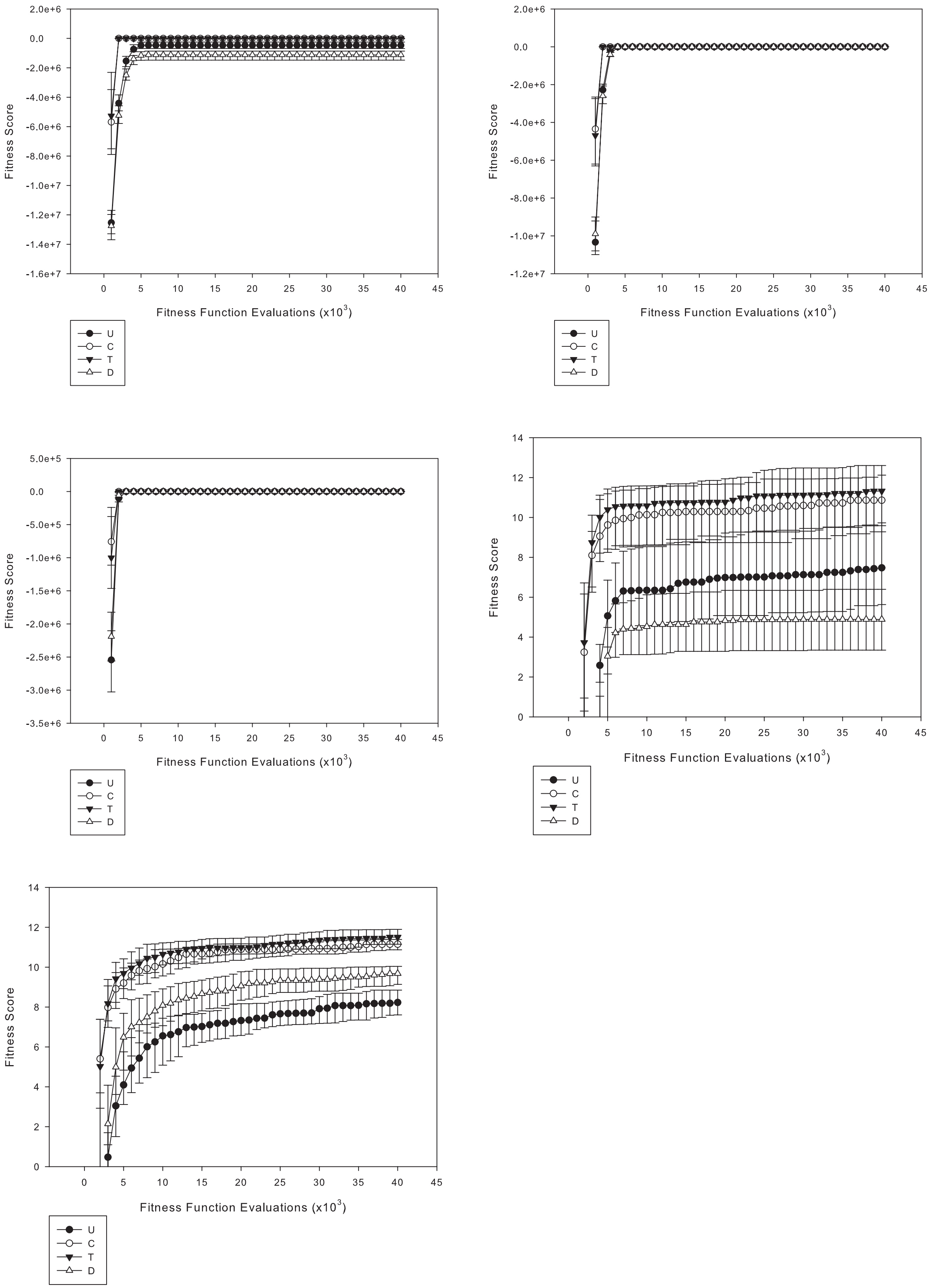


Figure G.5: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

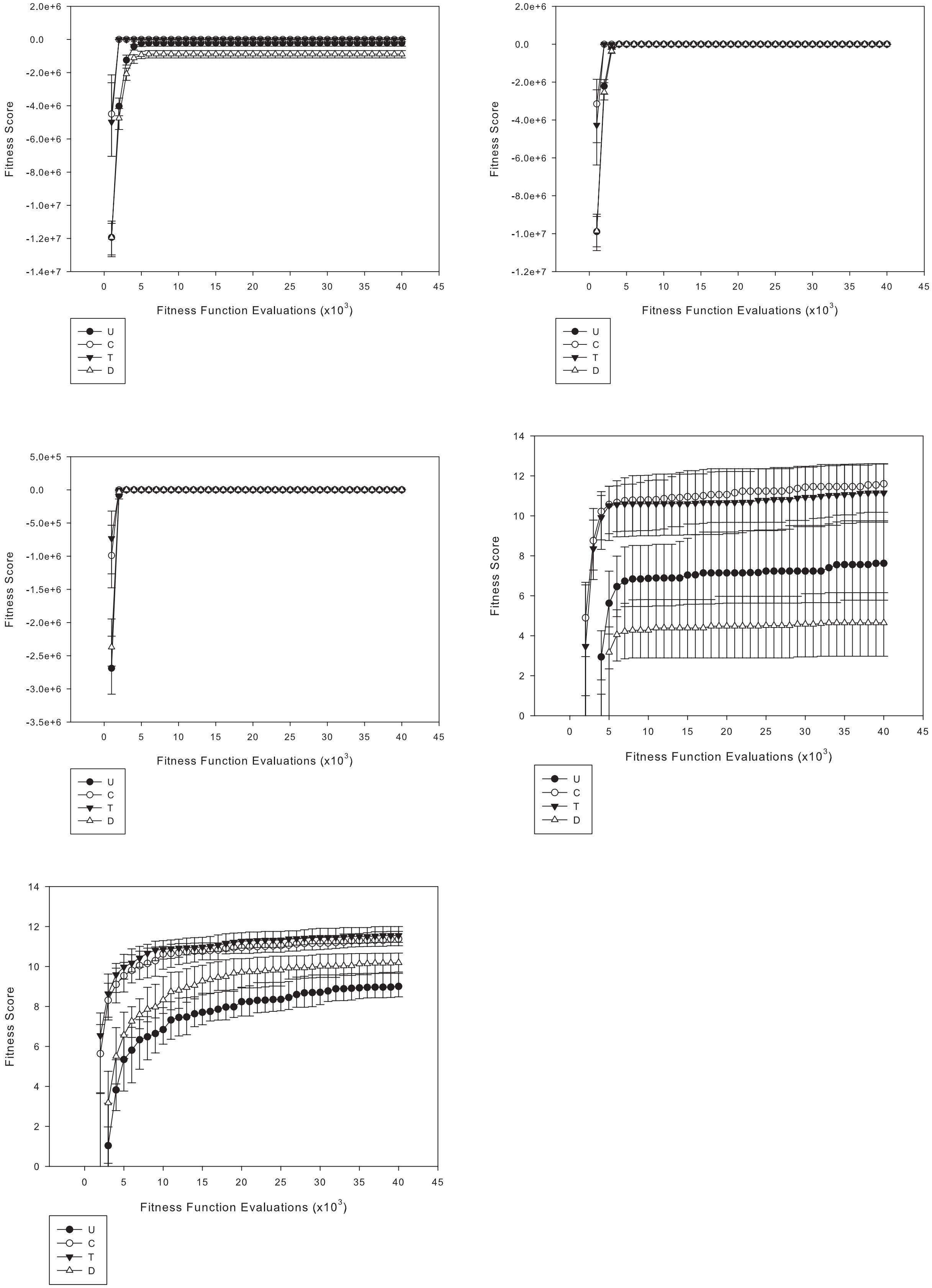


Figure G.6: $N = 100$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

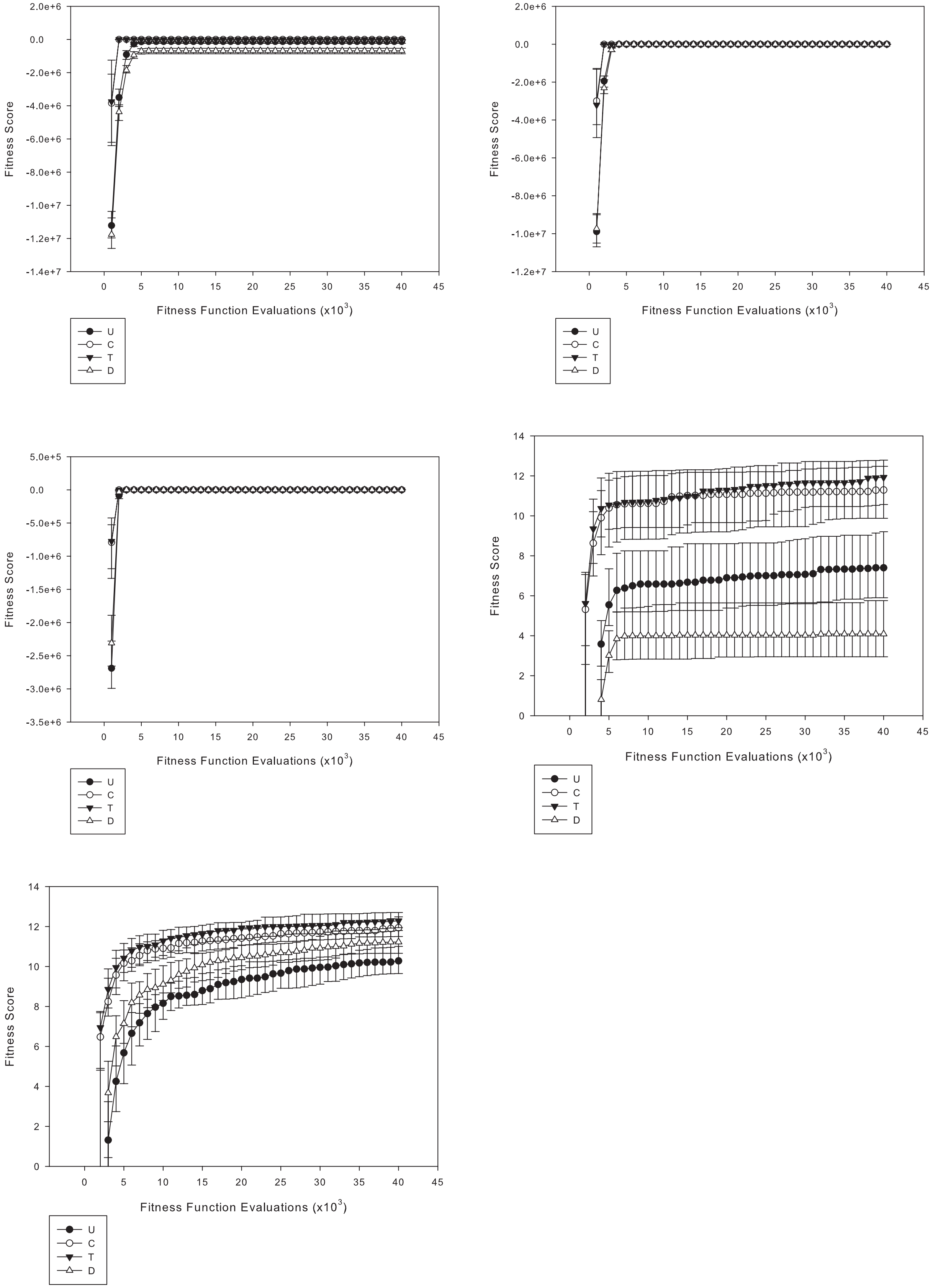


Figure G.7: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

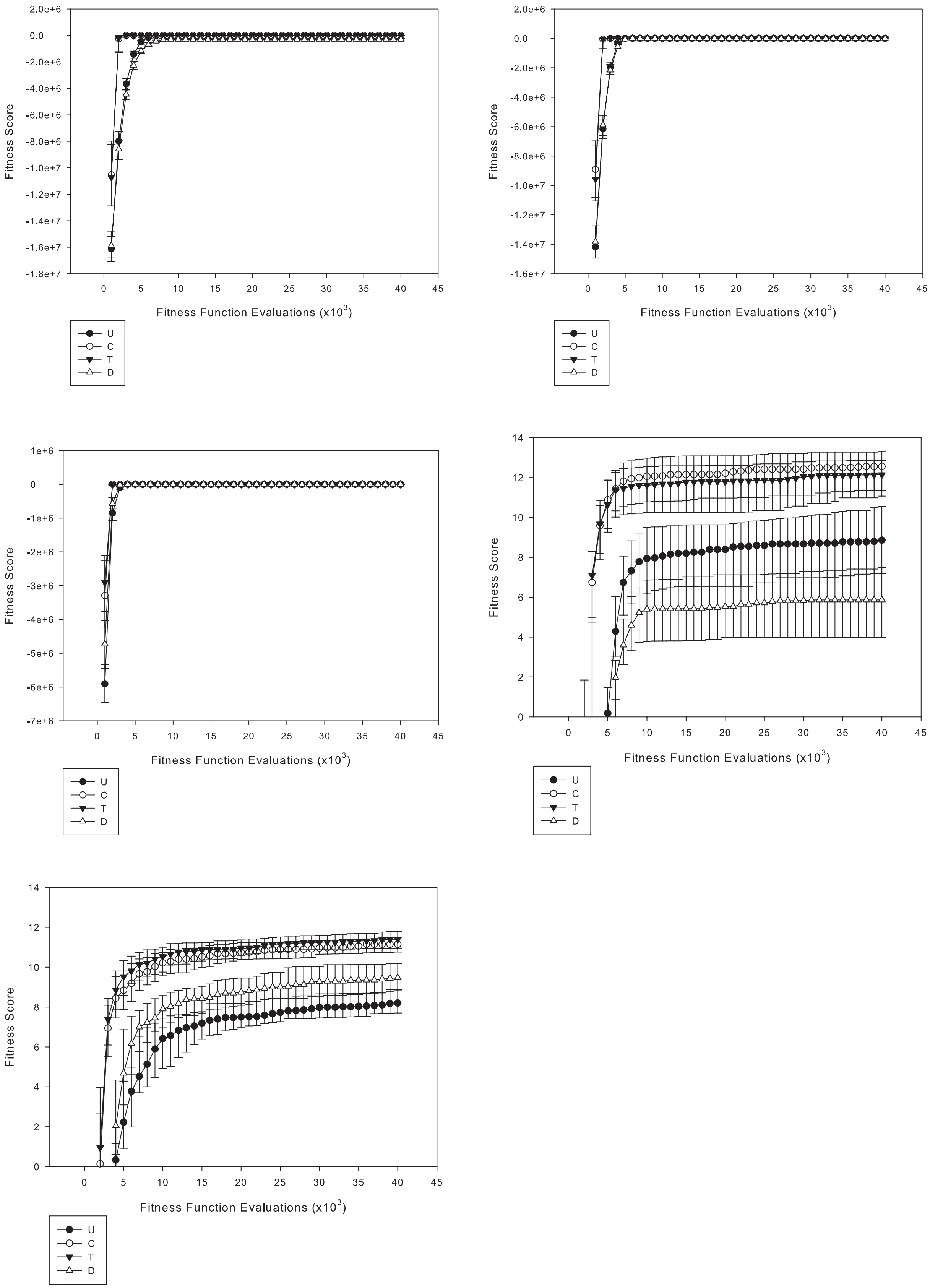


Figure G.8: $N = 150$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

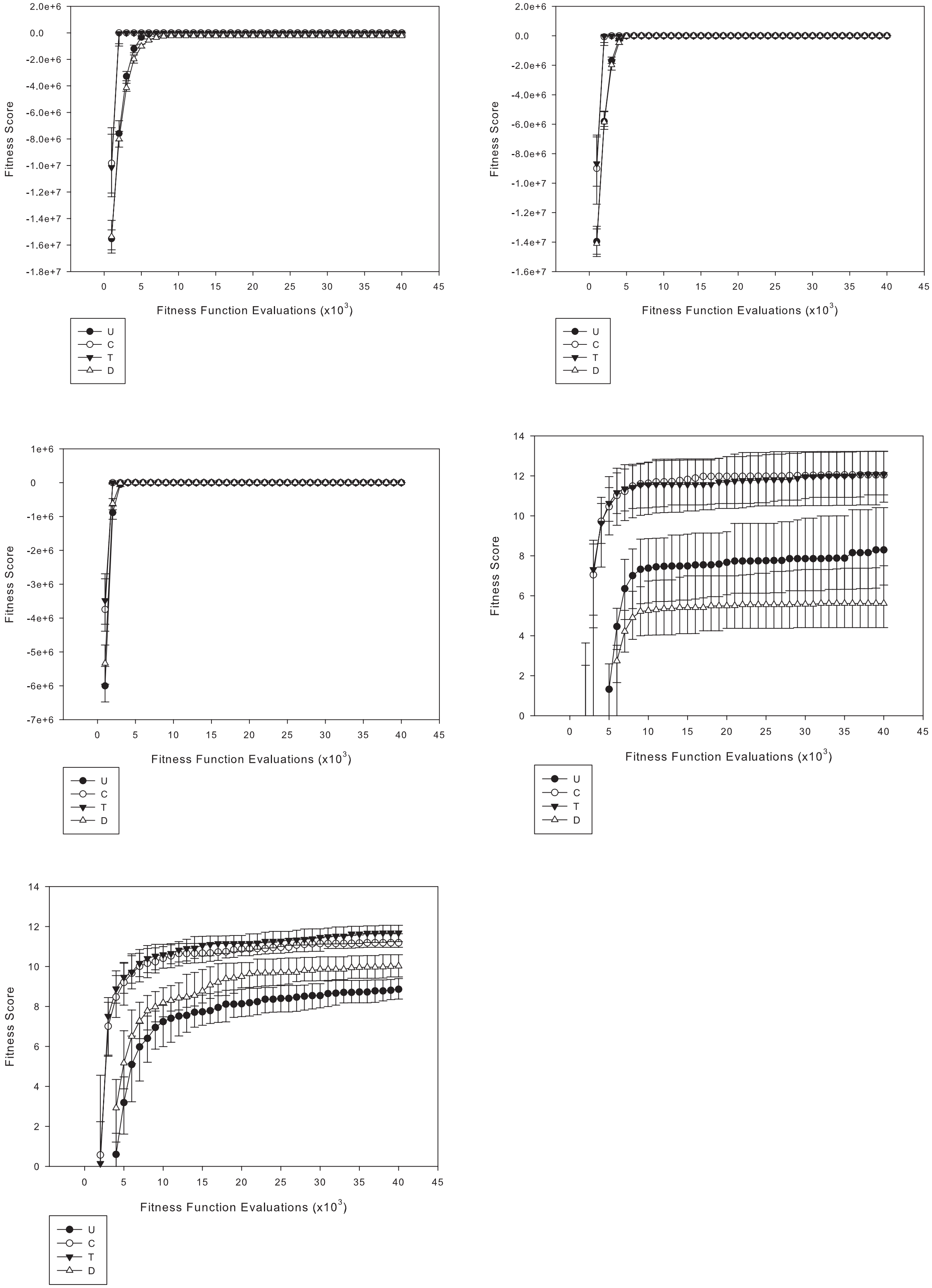
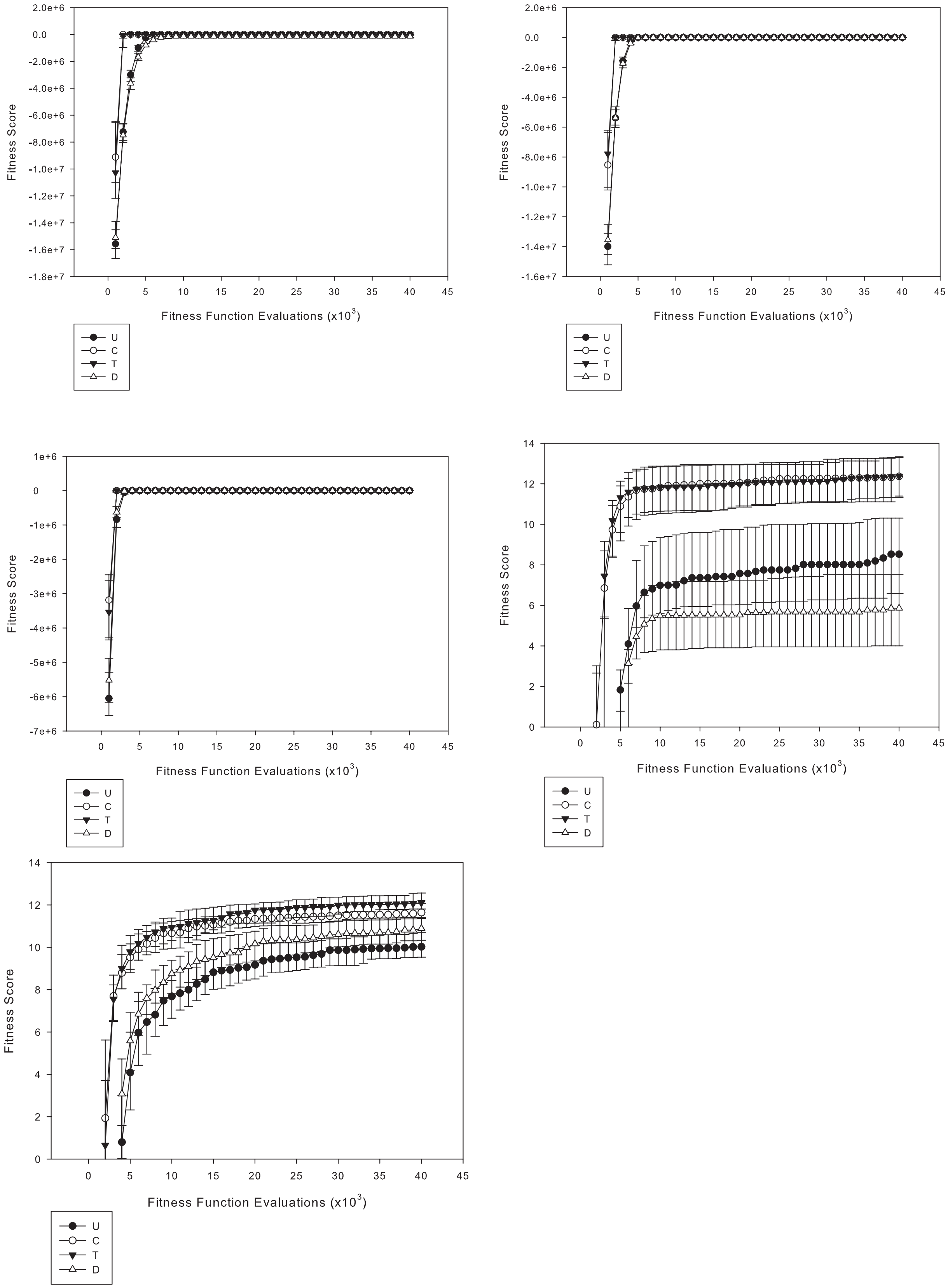


Figure G.9: $N = 150$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05



Appendix H

Single drug cancer chemotherapy scheduling graphs for UC, CalEB, TInSSel and FDC

Figure H.1: $N = 50$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

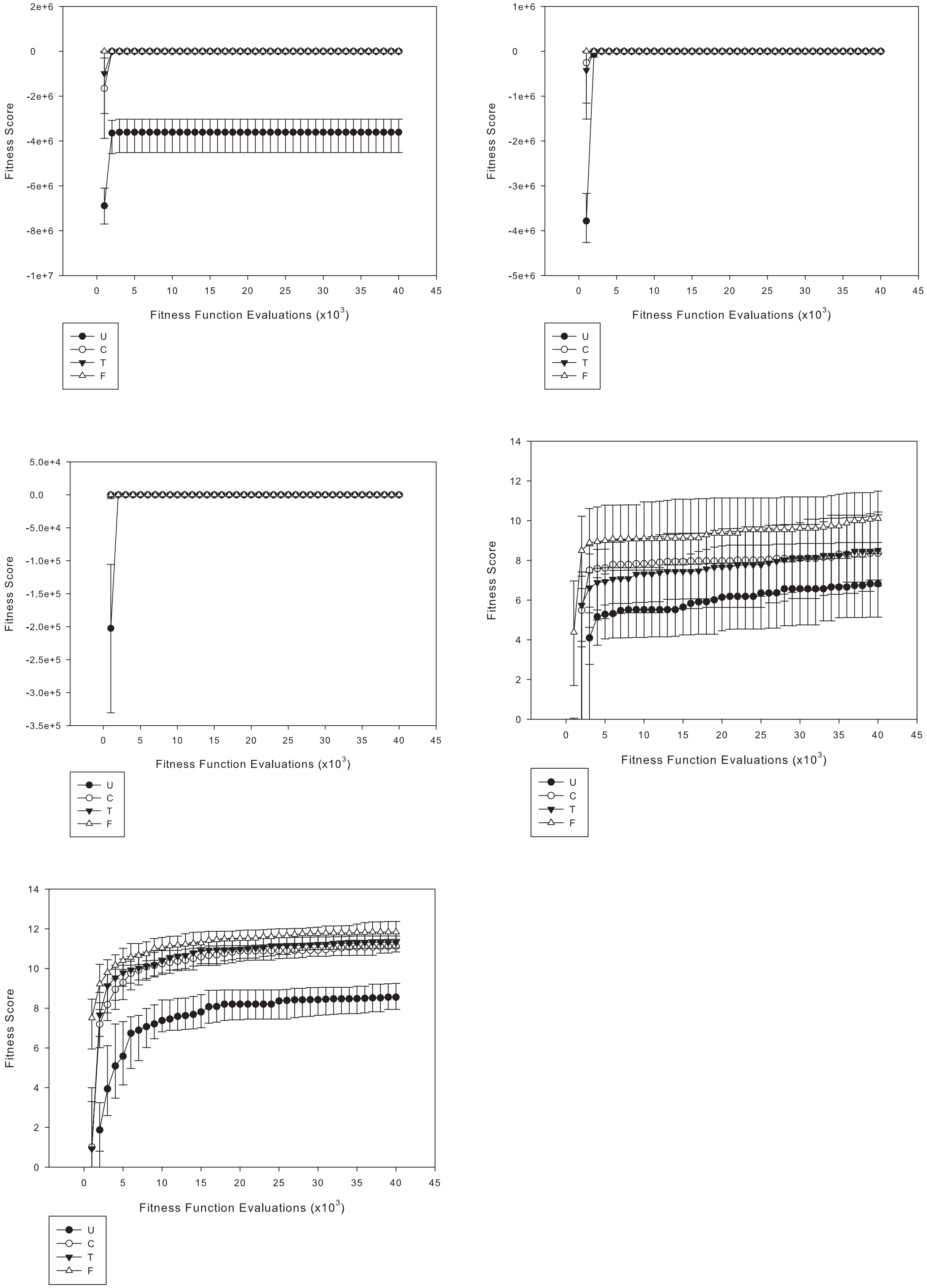


Figure H.2: $N = 50$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

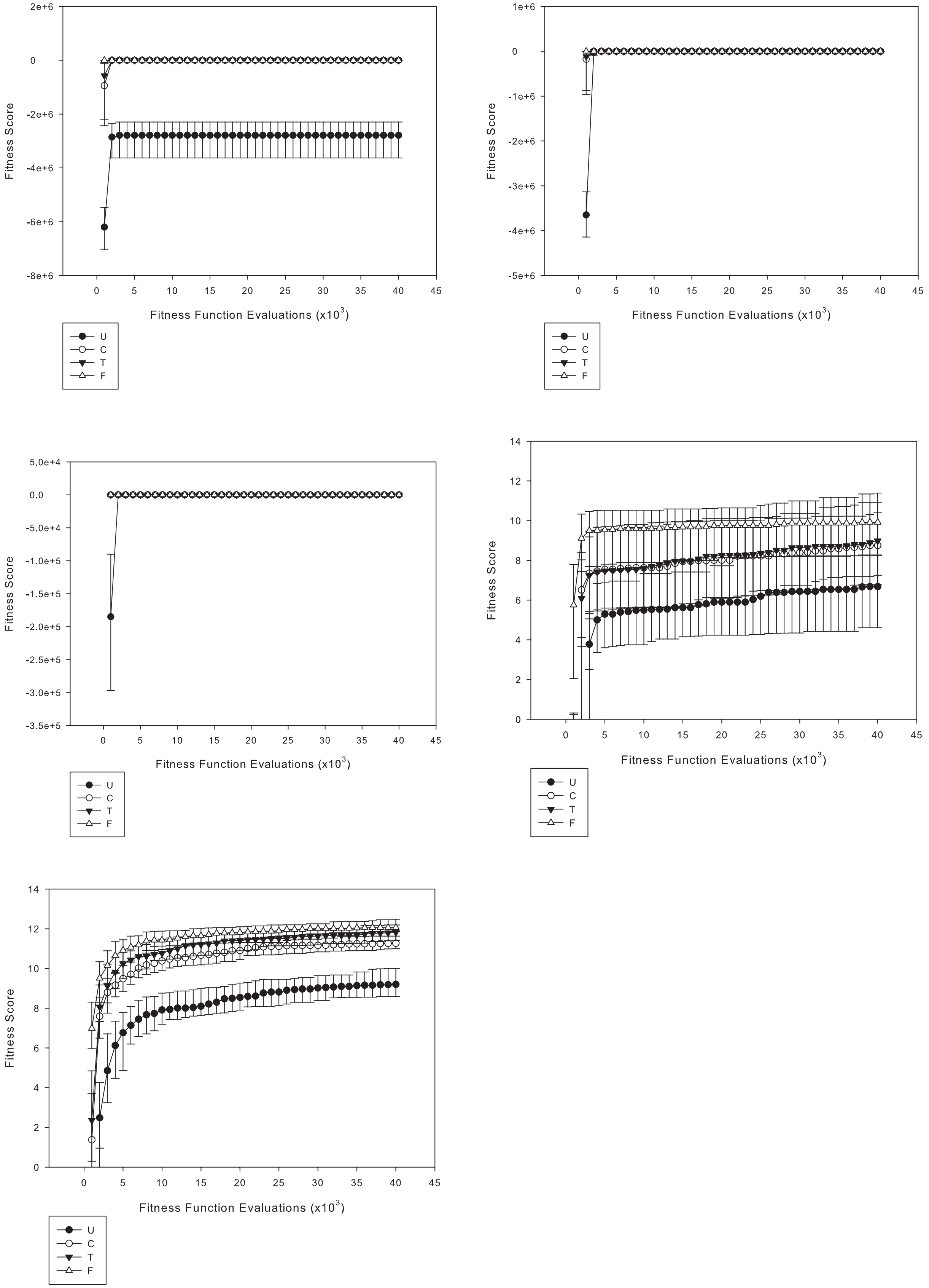


Figure H.3: $N = 50$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

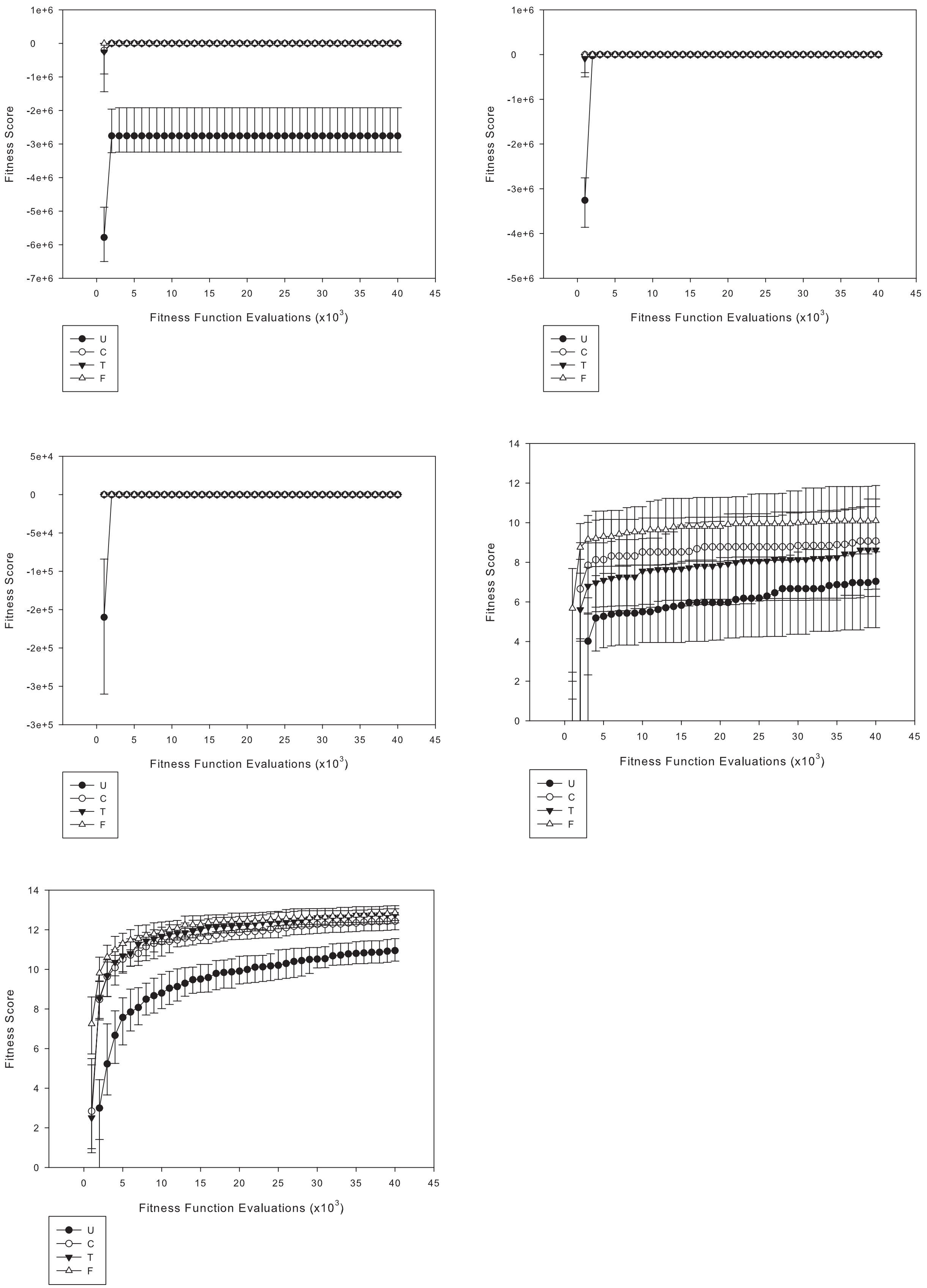


Figure H.4: $N = 100$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

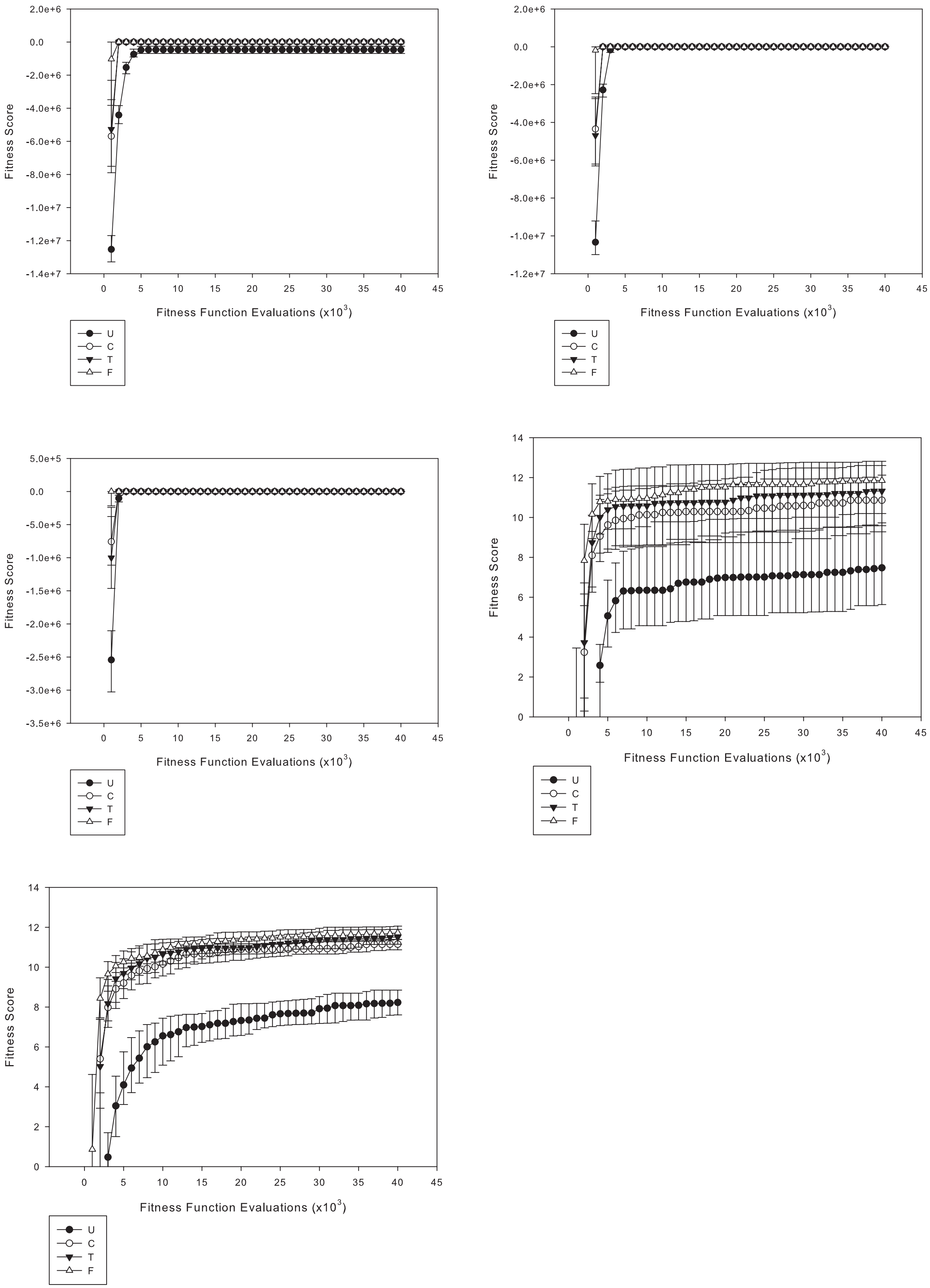


Figure H.5: $N = 100$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

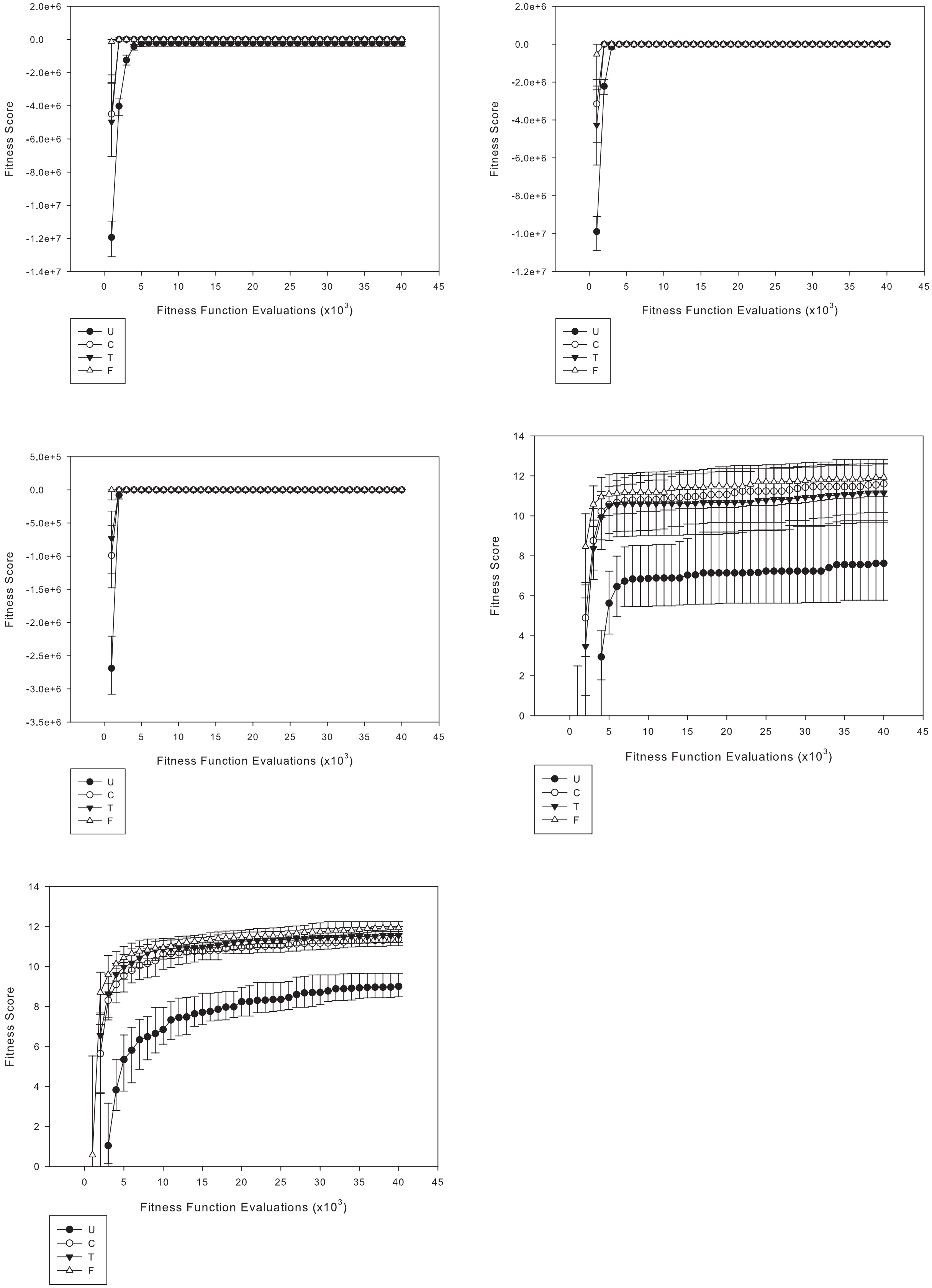


Figure H.6: $N = 100$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

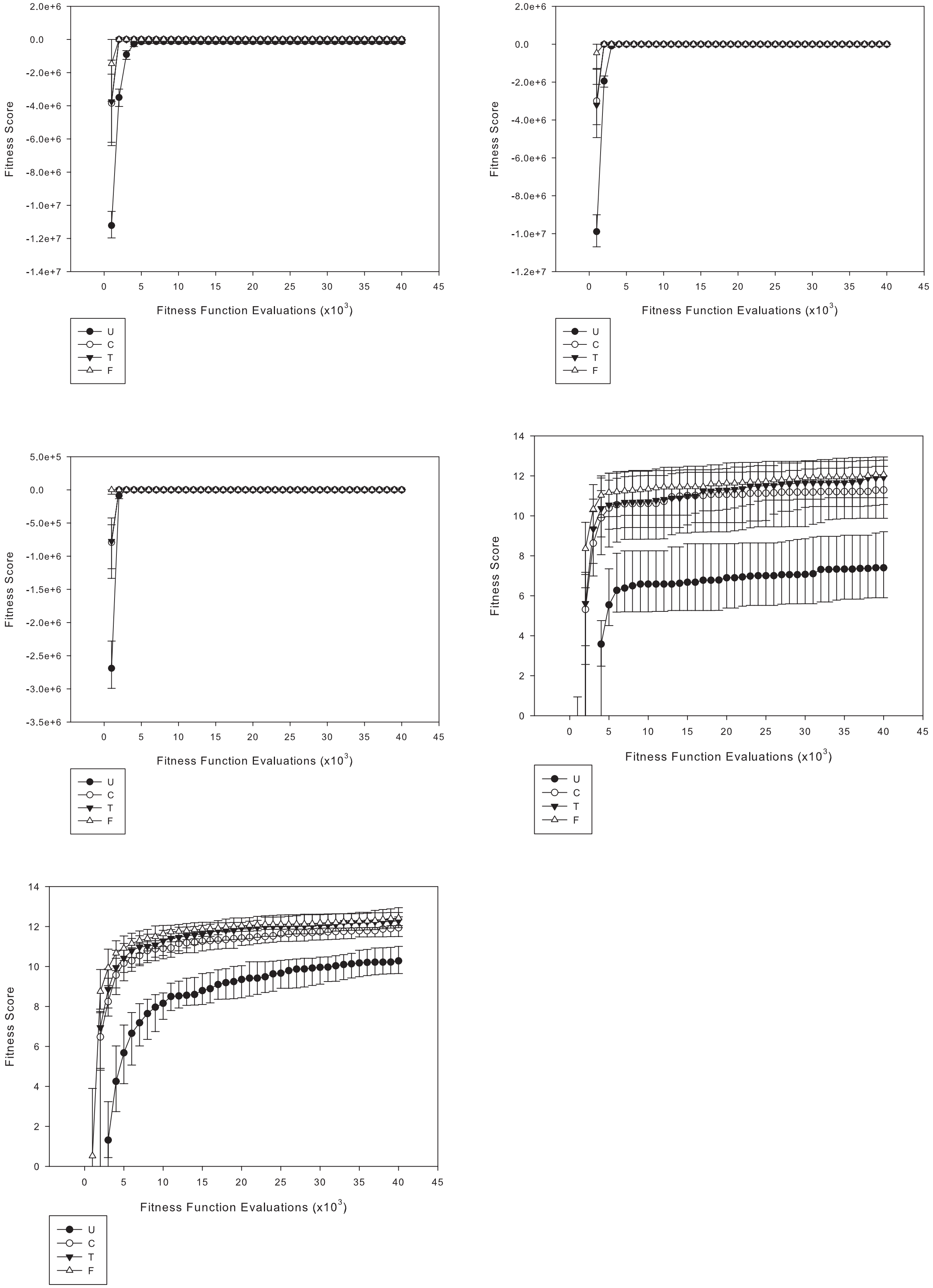


Figure H.7: $N = 150$, p_c 0.8, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

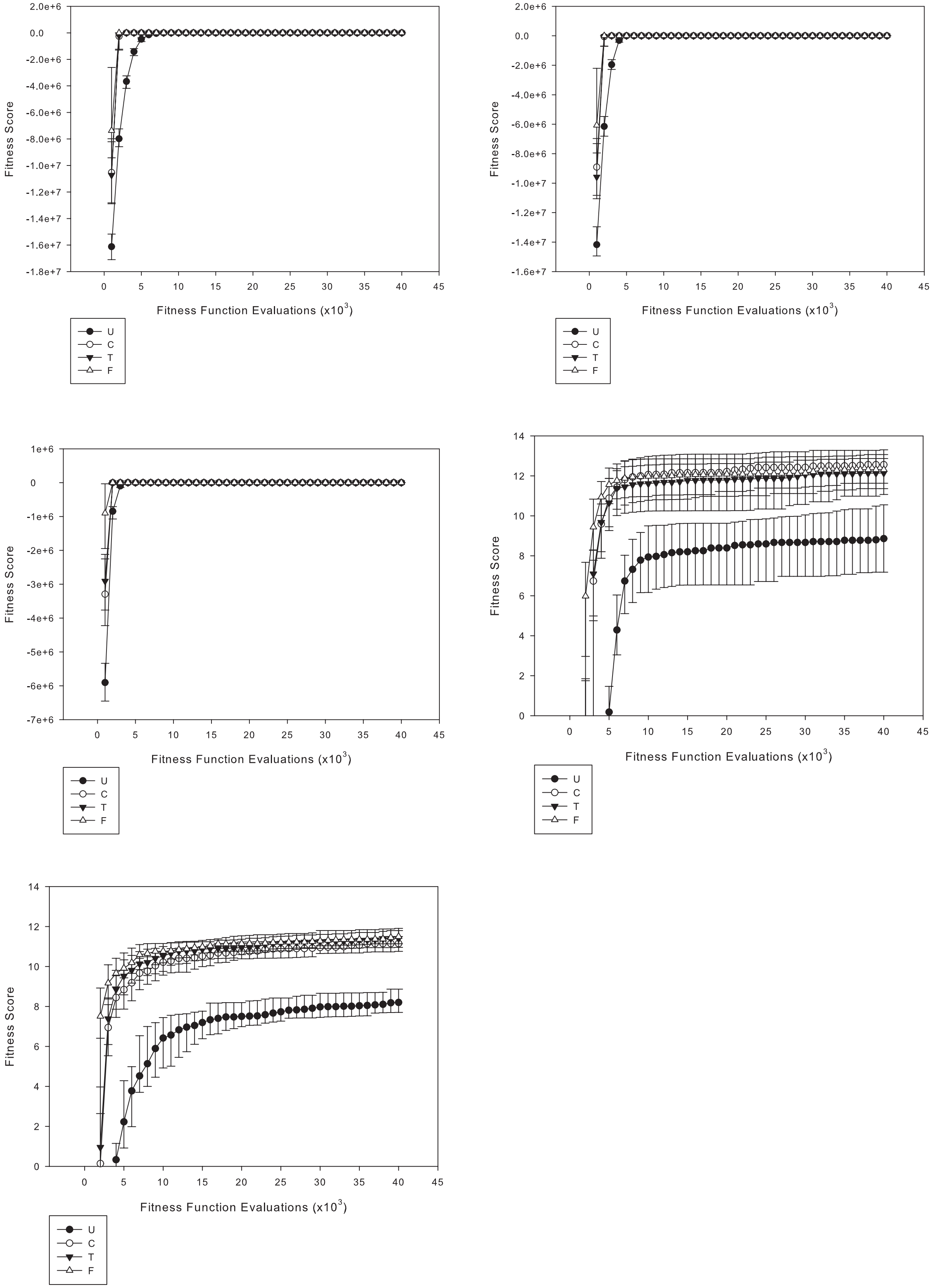


Figure H.8: $N = 150$, p_c 0.9, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05

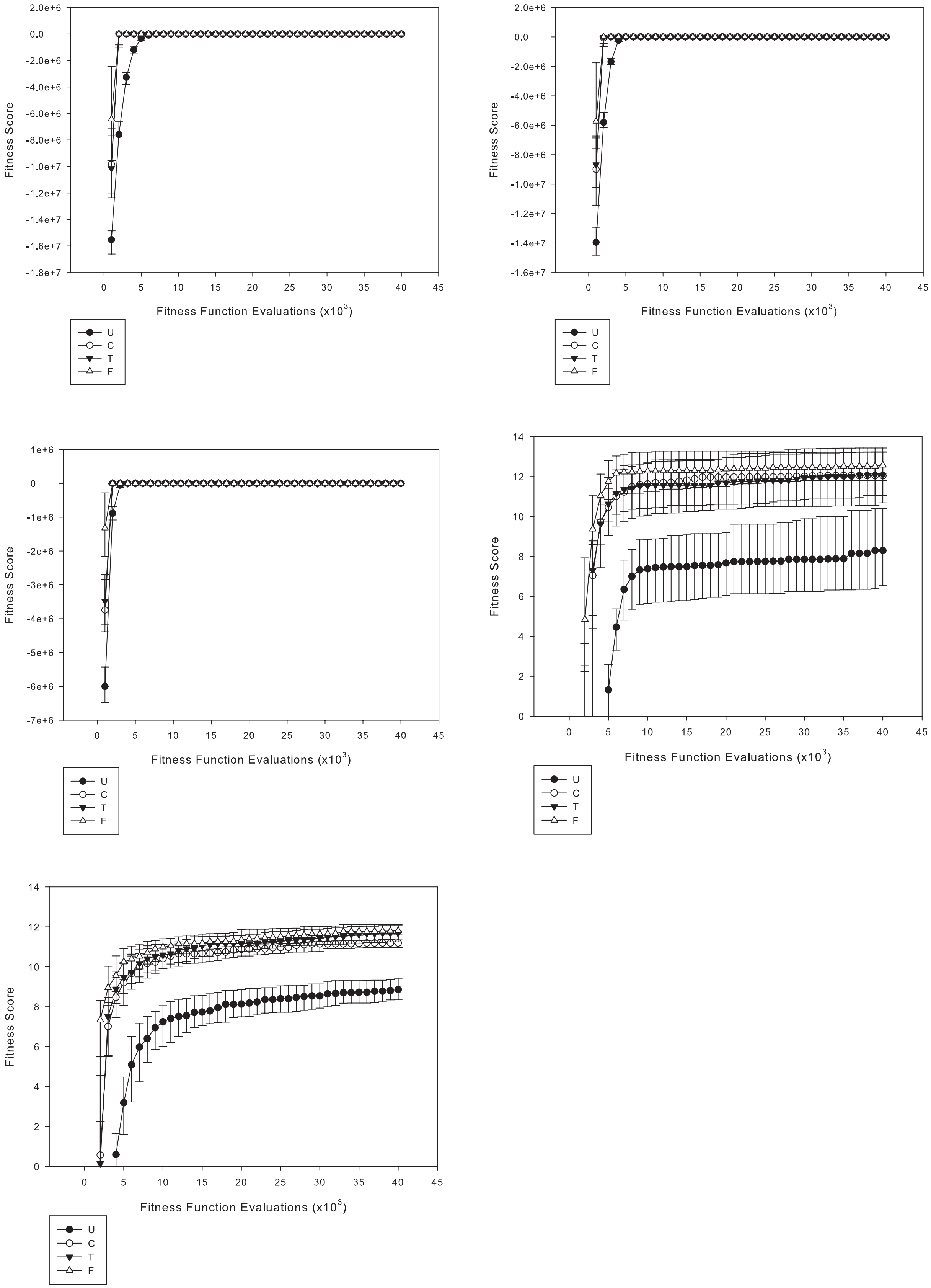
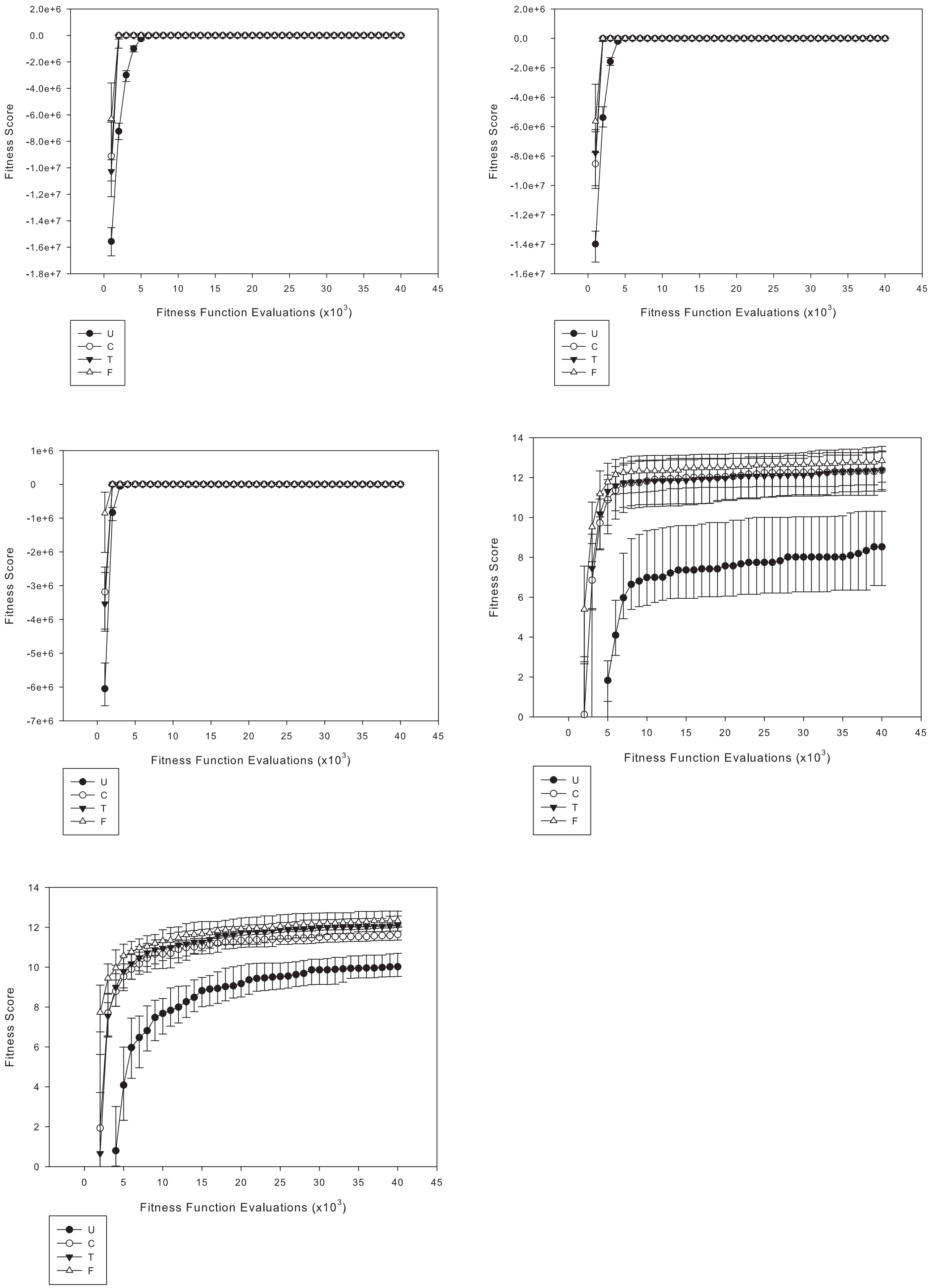


Figure H.9: $N = 150$, p_c 1.0, with p_m 0, 0.005 and 0.05 followed by magnified versions of p_m 0.005 and 0.05



Appendix I

Kruskal-Wallis one-way analysis of variance for UC, CalEB, TInSSel and DUC for chemotherapy optimisation

I.1 KW values for comparisons between UC, CalEB, TInSSel and DUC for the chemotherapy problem

[illegible]

I.2 KW values for comparisons between UC, CalEB, TInSSel and FDC for the chemotherapy problem

[illegible]

Appendix J

Statistical differences between approaches

J.1 Statistical differences between UC, CalEB, TInSSel and DUC for bio-control scheduling

J.1.1 0 penalty points per intervention

Section E.1.1 shows the Kruskal-Wallis test values for each of the approaches with an intervention penalty of 0 points per intervention. When the AS value is less than 0.01, this shows that a statistical difference exists between at least two of the crossover approaches at the 99% confidence level. Tables J.1 to J.3 detail which crossover approaches are statistically significantly different for mutation levels of 0, 0.005 and 0.05 respectively.

Table J.1: Mutation rate of 0

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200		T				
300	C	T		T		T
400	C	T		T	C	T
500	C	T		T	C	T
600	C	T			C	T
700	C	T			C	T
800	C	T			C	T
900	C	T			C	T
1000	C	T			C	T
1100	C	T			C	T
1200	C	T			C	T
1300	C	T	U		C	T
1400	C	T	U		C	T
1500	C	T	U		C	T
1600	C	T	U		C	T
1700	C	T	U		C	T
1800	C	T	U		C	T
1900	C	T	U		C	T
2000	C	T	U		C	T
2100	C	T	U		C	T
2200	C	T	U		C	T
2300	C	T	U		C	T
2400	C	T	U		C	T
2500	C	T	U		C	T

Continued on Next Page…

Table J.1 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
2600	C	T	U		C	T
2700	C	T	U		C	T
2800	C	T	U		C	T
2900	C	T	U		C	T
3000	C	T	U		C	T
3100	C	T	U		C	T
3200	C	T	U		C	T
3300	C	T	U		C	T
3400	C	T	U		C	T
3500	C	T	U		C	T
3600	C	T	U		C	T
3700	C	T	U		C	T
3800	C	T	U		C	T
3900	C	T	U		C	T
4000	C	T	U		C	T
4100	C	T	U		C	T
4200	C	T	U		C	T
4300	C	T	U		C	T
4400	C	T	U		C	T
4500	C	T	U		C	T
4600	C	T	U		C	T
4700	C	T	U		C	T
4800	C	T	U		C	T
4900	C	T	U		C	T
5000	C	T	U		C	T

Table J.2: Mutation rate of 0.005

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200		T		T		
300		T		T		T
400	C	T		T		T
500	C	T		T	C	T
600	C	T			C	T
700	C	T			C	T
800	C	T			C	T
900	C	T			C	T
1000	C	T			C	T
1100	C	T			C	T
1200	C	T			C	T
1300	C	T			C	T
1400	C	T			C	T
1500	C	T			C	T
1600	C	T			C	T
1700	C	T			C	T
1800	C	T			C	T
1900	C	T	U		C	T

Continued on Next Page…

Table J.2 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
2000	C	T	U		C	T
2100	C	T	U		C	T
2200	C	T	U		C	T
2300	C	T	U		C	T
2400	C	T	U		C	T
2500	C	T	U		C	T
2600	C	T	U		C	T
2700	C	T	U		C	T
2800	C	T	U		C	T
2900	C	T	U		C	T
3000	C	T	U		C	T
3100	C	T	U		C	T
3200	C	T	U		C	T
3300	C	T	U		C	T
3400	C	T	U		C	T
3500	C	T	U		C	T
3600	C	T	U		C	T
3700	C	T	U		C	T
3800	C	T	U		C	T
3900	C	T	U		C	T
4000	C	T	U		C	T
4100	C	T	U		C	T
4200	C	T	U		C	T
4300	C	T	U		C	T
4400	C	T	U		C	T
4500	C	T	U		C	T
4600			U		C	T
4700			U		C	T
4800			U		C	T
4900			U		C	T
5000			U		C	T

Table J.3: Mutation rate of 0.05

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200						
300		T				T
400	C	T		T	C	T
500	C	T			C	T
600	C	T	D		C	T
700	C	T	D		C	T
800	C	T	D		C	T
900	C	T	D		C	T
1000	C	T	D		C	T
1100	C	T			C	T
1200	C	T			C	T
1300	C	T			C	T

Continued on Next Page…

Table J.3 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
1400	C	T	D		C	T
1500	C	T			C	T
1600	C	T			C	T
1700	C	T	D		C	T
1800	C	T	D		C	T
1900	C	T	D		C	T
2000	C	T	D		C	T
2100	C	T	D		C	T
2200	C	T	D		C	T
2300	C	T	D		C	T
2400	C	T	D		C	T
2500	C	T	D		C	T
2600	C	T	D		C	T
2700	C	T	D		C	T
2800	C	T	D		C	T
2900	C	T	D		C	T
3000	C	T	D		C	T
3100	C	T	D		C	T
3200	C	T	D		C	T
3300	C	T	D		C	T
3400	C	T	D		C	T
3500	C	T	D		C	T
3600	C	T	D		C	T
3700	C	T	D		C	T
3800	C	T	D		C	T
3900	C	T	D		C	T
4000	C	T	D		C	T
4100	C	T	D		C	T
4200	C	T	D		C	T
4300	C	T	D		C	T
4400	C	T	D		C	T
4500	C	T	D		C	T
4600	C	T	D		C	T
4700	C	T	D		C	T
4800	C	T	D		C	T
4900	C	T	D		C	T
5000	C	T	D		C	T

J.1.2 5 penalty points per intervention

Section E.1.2 shows the Kruskal-Wallis test values for each of the approaches with an intervention penalty of 5 points per intervention. When the AS value is less than 0.01, this shows that a statistical difference exists between at least two of the crossover approaches at the 99% confidence level. Tables J.4 to J.6 detail which crossover approaches are statistically significantly different for mutation levels of 0, 0.005 and 0.05 respectively.

Table J.4: Mutation rate of 0

FFE(s)	**U/C**	**U/T**	**U/D**	**C/T**	**C/D**	**T/D**
100						
200	U			T	D	
300	U			T	D	
400	U			T	D	
500	U			T	D	
600	U			T	D	
700	U			T	D	T
800	U	T		T	D	T
900	U	T		T	D	T
1000	U	T		T	D	T
1100	U	T		T	D	T
1200	U	T		T	D	T
1300	U	T	U	T	D	T
1400	U	T	U	T	D	T
1500	U	T	U	T		T
1600	U	T	U	T		T
1700	U	T	U	T		T
1800	U		U	T		T
1900	U		U	T		T
2000	U		U	T		T
2100	U		U	T	C	T
2200	U		U	T	C	T
2300	U		U	T	C	T
2400	U		U	T	C	T
2500	U		U	T	C	T
2600	U		U	T	C	T
2700	U		U	T	C	T
2800	U		U	T	C	T
2900	U		U	T	C	T
3000	U		U	T	C	T
3100	U		U	T	C	T
3200	U		U	T	C	T
3300	U		U	T	C	T
3400	U		U	T	C	T
3500	U		U	T	C	T
3600	U		U	T	C	T
3700	U		U	T	C	T
3800	U		U	T	C	T
3900	U		U	T	C	T
4000	U		U	T	C	T
4100	U		U	T	C	T
4200	U		U	T	C	T
4300	U		U	T	C	T
4400	U		U	T	C	T
4500	U		U	T	C	T
4600	U		U	T	C	T
4700	U		U	T	C	T
4800	U		U	T	C	T
4900	U		U	T	C	T
5000	U		U	T	C	T

Table J.5: Mutation rate of 0.005

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200	U			T	D	
300	U			T	D	
400	U			T	D	
500	U			T	D	
600	U			T	D	
700	U			T	D	
800	U	T		T	D	
900	U	T		T	D	T
1000	U	T		T	D	T
1100	U	T		T	D	T
1200	U	T		T	D	T
1300	U	T		T	D	T
1400	U	T		T	D	T
1500	U	T		T	D	T
1600	U	T		T	D	T
1700	U	T		T	D	T
1800	U	T	U	T	D	T
1900	U	T	U	T	D	T
2000	U	T	U	T	D	T
2100	U		U	T	D	T
2200	U		U	T	D	T
2300	U		U	T	D	T
2400	U		U	T	D	T
2500	U		U	T	D	T
2600	U		U	T		T
2700	U		U	T		T
2800	U		U	T		T
2900	U		U	T		T
3000	U		U	T		T
3100	U			T		T
3200				T		T
3300				T		T
3400						T
3500						T
3600						T
3700						T
3800						
3900						
4000						
4100						
4200						
4300						
4400						
4500						
4600						
4700						
4800					C	
4900					C	

Continued on Next Page…

Table J.5 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
5000					C	

Table J.6: Mutation rate of 0.05

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200	U			T	D	
300	U			T	D	
400	U			T	D	
500	U			T	D	
600	U			T	D	
700	U			T	D	
800	U			T	D	
900	U			T	D	
1000	U			T	D	
1100	U			T	D	
1200	U			T	D	
1300	U	T		T	D	
1400		T		T	D	
1500		T		T	D	
1600		T		T	D	
1700		T	D	T	D	
1800		T	D	T	D	
1900		T	D	T	D	
2000		T	D	T	D	
2100		T	D	T	D	
2200		T	D	T	D	
2300		T	D	T	D	
2400		T	D	T	D	
2500		T	D	T	D	
2600		T	D	T	D	
2700		T	D	T	D	
2800		T	D	T	D	
2900		T	D	T	D	T
3000		T	D	T	D	
3100		T	D	T	D	
3200		T	D	T	D	
3300		T	D	T	D	
3400		T	D	T	D	
3500		T	D	T	D	
3600		T	D	T	D	
3700		T	D	T	D	
3800		T	D	T	D	
3900		T	D	T	D	
4000		T	D	T	D	
4100		T	D	T	D	
4200		T	D	T	D	
4300		T	D	T	D	

Continued on Next Page…

Table J.6 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
4400		T	D	T	D	
4500		T	D	T	D	
4600		T	D	T	D	
4700		T	D	T	D	
4800		T	D	T	D	
4900		T	D	T	D	
5000		T	D	T	D	

J.1.3 20 penalty points per intervention

Section E.1.3 shows the Kruskal-Wallis test values for each of the approaches with an intervention penalty of 20 points per intervention. When the AS value is less than 0.01, this shows that a statistical difference exists between at least two of the crossover approaches at the 99% confidence level. Tables J.7 to J.9 detail which crossover approaches are statistically significantly different for mutation levels of 0, 0.005 and 0.05 respectively.

Table J.7: Mutation rate of 0

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200					C	T
300	C	T			C	T
400	C	T			C	T
500	C	T			C	T
600	C	T	U		C	T
700	C	T	U		C	T
800	C	T	U		C	T
900	C	T	U	T	C	T
1000	C	T	U	T	C	T
1100	C	T	U	T	C	T
1200	C	T	U	T	C	T
1300	C	T	U	T	C	T
1400	C	T	U		C	T
1500	C	T	U		C	T
1600	C	T	U		C	T
1700	C	T	U		C	T
1800		T	U	T	C	T
1900		T	U	T	C	T
2000		T	U	T	C	T
2100			U	T	C	T
2200			U	T	C	T
2300			U	T	C	T
2400			U	T	C	T
2500			U	T	C	T
2600			U	T	C	T
2700			U	T	C	T
2800			U	T	C	T
2900			U	T	C	T
3000			U	T	C	T

Continued on Next Page…

Table J.7 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
3100			U	T	C	T
3200			U	T	C	T
3300			U	T	C	T
3400			U	T	C	T
3500			U	T	C	T
3600			U	T	C	T
3700			U	T	C	T
3800			U	T	C	T
3900			U	T	C	T
4000			U	T	C	T
4100			U	T	C	T
4200			U	T	C	T
4300			U	T	C	T
4400			U	T	C	T
4500			U	T	C	T
4600			U	T	C	T
4700			U	T	C	T
4800			U	T	C	T
4900			U	T	C	T
5000			U	T	C	T

Table J.8: Mutation rate of 0.005

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200		T				
300	C	T			C	T
400	C	T			C	T
500	C	T			C	T
600	C	T			C	T
700	C	T	U		C	T
800	C	T	U		C	T
900	C	T	U		C	T
1000	C	T	U		C	T
1100	C	T	U	T	C	T
1200	C	T	U	T	C	T
1300	C	T	U	T	C	T
1400	C	T	U	T	C	T
1500	C	T	U		C	T
1600	C	T	U		C	T
1700	C	T	U		C	T
1800	C	T	U		C	T
1900	C	T	U		C	T
2000	C	T	U		C	T
2100	C	T	U		C	T
2200	C	T	U		C	T
2300			U		C	T
2400			U		C	T

Continued on Next Page…

Table J.8 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
2500			U		C	T
2600			U		C	T
2700			U		C	T
2800			U		C	T
2900			U		C	T
3000			U		C	T
3100			U		C	T
3200			U		C	T
3300			U		C	T
3400			U		C	T
3500			U		C	T
3600			U		C	T
3700			U		C	T
3800			U		C	T
3900			U		C	T
4000			U		C	T
4100			U		C	T
4200			U		C	T
4300			U		C	T
4400			U		C	T
4500			U		C	T
4600			U		C	T
4700			U		C	T
4800						
4900						
5000						

Table J.9: Mutation rate of 0.05

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200					C	T
300	C	T			C	T
400	C	T			C	T
500	C	T			C	T
600	C	T			C	T
700	C	T			C	T
800	C	T			C	T
900	C	T			C	T
1000	C	T			C	T
1100	C	T			C	T
1200	C	T			C	T
1300	C	T			C	T
1400	C	T			C	T
1500	C	T			C	T
1600	C	T			C	T
1700	C	T			C	T
1800	C	T	D		C	T

Continued on Next Page…

Table J.9 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
1900	C	T	D		C	T
2000	C	T			C	T
2100	C	T			C	T
2200	C	T			C	T
2300	C	T			C	T
2400	C	T	D		C	T
2500	C	T	D		C	T
2600	C	T	D		C	T
2700	C	T	D		C	T
2800	C	T	D		C	T
2900	C	T	D		C	T
3000	C	T	D		C	T
3100	C	T	D		C	T
3200	C	T	D		C	T
3300	C	T	D		C	T
3400	C	T	D		C	T
3500	C	T	D		C	T
3600	C	T	D		C	T
3700	C	T	D		C	T
3800	C	T	D		C	T
3900	C	T	D		C	T
4000	C	T	D		C	T
4100	C	T	D		C	T
4200	C	T	D		C	T
4300	C	T	D		C	T
4400	C	T	D		C	T
4500	C	T	D		C	T
4600	C	T	D		C	T
4700	C	T	D		C	T
4800	C	T	D		C	T
4900	C	T	D		C	T
5000	C	T	D		C	T

J.1.4 35 penalty points per intervention

Section E.1.4 shows the Kruskal-Wallis test values for each of the approaches with an intervention penalty of 35 points per intervention. When the AS value is less than 0.01, this shows that a statistical difference exists between at least two of the crossover approaches at the 99% confidence level. Tables J.10 to J.12 detail which crossover approaches are statistically significantly different for mutation levels of 0, 0.005 and 0.05 respectively.

Table J.10: Mutation rate of 0

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200	C	T			C	T
300	C	T			C	T
400	C	T	U		C	T
500	C	T	U		C	T

Continued on Next Page…

Table J.10 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
600	C	T	U		C	T
700	C	T	U		C	T
800	C	T	U		C	T
900	C	T	U		C	T
1000	C	T	U		C	T
1100	C	T	U		C	T
1200	C	T	U		C	T
1300	C	T	U		C	T
1400	C	T	U		C	T
1500	C	T	U		C	T
1600	C	T	U		C	T
1700	C	T	U		C	T
1800		T	U		C	T
1900			U		C	T
2000			U		C	T
2100			U		C	T
2200			U		C	T
2300			U		C	T
2400			U		C	T
2500			U		C	T
2600			U		C	T
2700			U		C	T
2800			U		C	T
2900			U		C	T
3000			U		C	T
3100			U		C	T
3200			U		C	T
3300			U		C	T
3400			U		C	T
3500			U		C	T
3600			U		C	T
3700			U		C	T
3800			U		C	T
3900			U		C	T
4000			U		C	T
4100			U		C	T
4200			U		C	T
4300			U		C	T
4400			U		C	T
4500			U		C	T
4600			U		C	T
4700			U		C	T
4800			U		C	T
4900			U		C	T
5000			U		C	T

Table J.11: Mutation rate of 0.005

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200	C	T			C	T
300	C	T			C	T
400	C	T			C	T
500	C	T	U		C	T
600	C	T	U		C	T
700	C	T	U		C	T
800	C	T	U		C	T
900	C	T	U		C	T
1000	C	T	U		C	T
1100	C	T	U		C	T
1200	C	T	U		C	T
1300	C	T	U		C	T
1400	C	T	U		C	T
1500	C	T	U		C	T
1600	C	T	U		C	T
1700	C	T	U		C	T
1800	C	T	U		C	T
1900	C	T	U		C	T
2000			U		C	T
2100			U		C	T
2200			U		C	T
2300			U		C	T
2400			U		C	T
2500			U		C	T
2600			U		C	T
2700			U		C	T
2800			U		C	T
2900			U		C	T
3000			U		C	T
3100			U		C	T
3200			U		C	T
3300			U		C	T
3400			U		C	T
3500			U		C	T
3600			U		C	T
3700			U		C	T
3800			U		C	T
3900			U		C	T
4000			U		C	T
4100			U		C	T
4200					C	T
4300						
4400						
4500						
4600						
4700						
4800						
4900						
5000						

Table J.12: Mutation rate of 0.05

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200	C	T			C	T
300	C	T			C	T
400	C	T			C	T
500	C	T			C	T
600	C	T			C	T
700	C	T			C	T
800	C	T			C	T
900	C	T			C	T
1000	C	T			C	T
1100	C	T			C	T
1200	C	T			C	T
1300	C	T			C	T
1400	C	T			C	T
1500	C	T			C	T
1600	C	T			C	T
1700	C	T			C	T
1800	C	T			C	T
1900	C	T			C	T
2000	C	T			C	T
2100	C	T			C	T
2200	C	T			C	T
2300	C	T	D		C	T
2400	C	T	D		C	T
2500	C	T	D		C	T
2600	C	T	D		C	T
2700	C	T	D		C	T
2800	C	T	D		C	T
2900	C	T	D		C	T
3000	C	T	D		C	T
3100	C	T	D		C	T
3200	C	T	D		C	T
3300	C	T	D		C	T
3400	C	T	D		C	T
3500	C	T	D		C	T
3600	C	T	D		C	T
3700	C	T	D		C	T
3800	C	T	D		C	T
3900	C	T	D		C	T
4000	C	T	D		C	T
4100	C	T	D		C	T
4200	C	T	D		C	T
4300	C	T	D		C	T
4400	C	T	D		C	T
4500	C	T	D		C	T
4600	C	T	D		C	T
4700	C	T	D		C	T
4800	C	T	D		C	T
4900	C	T	D		C	T

Continued on Next Page…

Table J.12 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
5000	C	T	D		C	T

J.1.5 50 penalty points per intervention

Section E.1.5 shows the Kruskal-Wallis test values for each of the approaches with an intervention penalty of 50 points per intervention. When the AS value is less than 0.01, this shows that a statistical difference exists between at least two of the crossover approaches at the 99% confidence level. Tables J.13 to J.15 detail which crossover approaches are statistically significantly different for mutation levels of 0, 0.005 and 0.05 respectively.

Table J.13: Mutation rate of 0

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200		T			C	T
300	C	T			C	T
400	C	T	U		C	T
500	C	T	U		C	T
600	C	T	U		C	T
700	C	T	U		C	T
800	C	T	U		C	T
900	C	T	U		C	T
1000	C	T	U		C	T
1100	C	T	U		C	T
1200	C	T	U		C	T
1300	C	T	U		C	T
1400	C	T	U		C	T
1500	C	T	U		C	T
1600	C	T	U		C	T
1700	C	T	U		C	T
1800			U		C	T
1900			U		C	T
2000			U		C	T
2100			U		C	T
2200			U		C	T
2300			U		C	T
2400			U		C	T
2500			U		C	T
2600			U		C	T
2700			U		C	T
2800			U		C	T
2900			U		C	T
3000			U		C	T
3100			U		C	T
3200			U		C	T
3300			U		C	T
3400			U		C	T
3500			U		C	T
3600			U		C	T

Continued on Next Page. . .

Table J.13 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
3700			U		C	T
3800			U		C	T
3900			U		C	T
4000			U		C	T
4100			U		C	T
4200			U		C	T
4300			U		C	T
4400			U		C	T
4500			U		C	T
4600			U		C	T
4700			U		C	T
4800			U		C	T
4900			U		C	T
5000			U		C	T

Table J.14: Mutation rate of 0.005

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200	C	T			C	T
300	C	T	U		C	T
400	C	T	U		C	T
500	C	T	U		C	T
600	C	T	U		C	T
700	C	T	U		C	T
800	C	T	U		C	T
900	C	T	U		C	T
1000	C	T	U		C	T
1100	C	T	U		C	T
1200	C	T	U		C	T
1300	C	T	U		C	T
1400	C	T	U		C	T
1500	C	T	U		C	T
1600	C	T	U		C	T
1700	C	T	U		C	T
1800	C	T	U		C	T
1900	C	T	U		C	T
2000			U		C	T
2100			U		C	T
2200			U		C	T
2300			U		C	T
2400			U		C	T
2500			U		C	T
2600			U		C	T
2700			U		C	T
2800			U		C	T
2900			U		C	T
3000			U		C	T

Continued on Next Page…

Table J.14 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
3100			U		C	T
3200			U		C	T
3300			U		C	T
3400			U		C	T
3500			U		C	T
3600			U		C	T
3700			U		C	T
3800			U		C	T
3900			U		C	T
4000			U		C	T
4100			U		C	T
4200			U		C	T
4300			U		C	T
4400			U		C	T
4500			U		C	T
4600						
4700						
4800						
4900						
5000						

Table J.15: Mutation rate of 0.05

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
100						
200		T			C	T
300	C	T			C	T
400	C	T			C	T
500	C	T			C	T
600	C	T			C	T
700	C	T			C	T
800	C	T			C	T
900	C	T			C	T
1000	C	T			C	T
1100	C	T			C	T
1200	C	T			C	T
1300	C	T			C	T
1400	C	T			C	T
1500	C	T			C	T
1600	C	T			C	T
1700	C	T			C	T
1800	C	T			C	T
1900	C	T			C	T
2000	C	T			C	T
2100	C	T			C	T
2200	C	T			C	T
2300	C	T			C	T
2400	C	T			C	T

Continued on Next Page…

Table J.15 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
2500	C	T			C	T
2600	C	T			C	T
2700	C	T			C	T
2800	C	T			C	T
2900	C	T			C	T
3000	C	T			C	T
3100	C	T			C	T
3200	C	T	D		C	T
3300	C	T	D		C	T
3400	C	T	D		C	T
3500	C	T	D		C	T
3600	C	T	D		C	T
3700	C	T	D		C	T
3800	C	T	D		C	T
3900	C	T	D		C	T
4000	C	T	D		C	T
4100	C	T	D		C	T
4200	C	T	D		C	T
4300	C	T	D		C	T
4400	C	T	D		C	T
4500	C	T	D		C	T
4600	C	T	D		C	T
4700	C	T	D		C	T
4800	C	T	D		C	T
4900	C	T	D		C	T
5000	C	T	D		C	T

J.2 Statistical differences between UC, CalEB, TInSSel and DUC for chemotherapy scheduling

Section I.1 shows the Kruskal-Wallis test values for each of the crossover approaches. When the AS value is less than 0.01, this shows that a statistical difference exists between at least two of the crossover approaches at the 99% confidence level. Tables J.16 to J.18 detail which crossover approaches are statistically significantly different for mutation levels of 0, 0.005 and 0.05 respectively.

Table J.16: Mutation rate of 0

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
1000	C	T			C	T
2000	C	T			C	T
3000	C	T	U		C	T
4000	C	T	U		C	T
5000	C	T	U		C	T
6000	C	T	U		C	T
7000	C	T	U		C	T
8000	C	T	U		C	T
9000	C	T	U		C	T

Continued on Next Page. . .

Table J.16 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
10000	C	T	U		C	T
11000	C	T	U		C	T
12000	C	T	U		C	T
13000	C	T	U		C	T
14000	C	T	U		C	T
15000	C	T	U		C	T
16000	C	T	U		C	T
17000	C	T	U		C	T
18000	C	T	U		C	T
19000	C	T	U		C	T
20000	C	T	U		C	T
21000	C	T	U		C	T
22000	C	T	U		C	T
23000	C	T	U		C	T
24000	C	T	U		C	T
25000	C	T	U		C	T
26000	C	T	U		C	T
27000	C	T	U		C	T
28000	C	T	U		C	T
29000	C	T	U		C	T
30000	C	T	U		C	T
31000	C	T	U		C	T
32000	C	T	U		C	T
33000	C	T	U		C	T
34000	C	T	U		C	T
35000	C	T	U		C	T
36000	C	T	U		C	T
37000	C	T	U		C	T
38000	C	T	U		C	T
39000	C	T	U		C	T
40000	C	T	U		C	T

Table J.17: Mutation rate of 0.005

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
1000	C	T			C	T
2000	C	T			C	T
3000	C	T	U		C	T
4000	C	T	U		C	T
5000	C	T	U		C	T
6000	C	T	U		C	T
7000	C	T	U		C	T
8000	C	T	U		C	T
9000	C	T	U		C	T
10000	C	T	U		C	T
11000	C	T	U		C	T
12000	C	T	U		C	T
13000	C	T	U		C	T

Continued on Next Page…

Table J.17 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
14000	C	T	U		C	T
15000	C	T	U		C	T
16000	C	T	U		C	T
17000	C	T	U		C	T
18000	C	T	U		C	T
19000	C	T	U		C	T
20000	C	T	U		C	T
21000	C	T	U		C	T
22000	C	T	U		C	T
23000	C	T	U		C	T
24000	C	T	U		C	T
25000	C	T	U		C	T
26000	C	T	U		C	T
27000	C	T	U		C	T
28000	C	T	U		C	T
29000	C	T	U		C	T
30000	C	T	U		C	T
31000	C	T	U		C	T
32000	C	T	U		C	T
33000	C	T	U		C	T
34000	C	T	U		C	T
35000	C	T	U		C	T
36000	C	T	U		C	T
37000	C	T	U		C	T
38000	C	T	U		C	T
39000	C	T	U		C	T
40000	C	T	U		C	T

Table J.18: Mutation rate of 0.05

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
1000	C	T			C	T
2000	C	T			C	T
3000	C	T	D		C	T
4000	C	T	D		C	T
5000	C	T	D		C	T
6000	C	T	D		C	T
7000	C	T	D		C	T
8000	C	T	D		C	T
9000	C	T	D		C	T
10000	C	T	D		C	T
11000	C	T	D		C	T
12000	C	T	D		C	T
13000	C	T	D		C	T
14000	C	T	D		C	T
15000	C	T	D		C	T
16000	C	T	D		C	T
17000	C	T	D		C	T

Continued on Next Page…

Table J.18 – Continued

FFE(s)	U/C	U/T	U/D	C/T	C/D	T/D
18000	C	T	D		C	T
19000	C	T	D		C	T
20000	C	T	D		C	T
21000	C	T	D		C	T
22000	C	T	D		C	T
23000	C	T	D		C	T
24000	C	T	D		C	T
25000	C	T	D		C	T
26000	C	T	D		C	T
27000	C	T	D		C	T
28000	C	T	D		C	T
29000	C	T	D		C	T
30000	C	T	D		C	T
31000	C	T	D		C	T
32000	C	T	D		C	T
33000	C	T	D		C	T
34000	C	T	D		C	T
35000	C	T	D		C	T
36000	C	T	D		C	T
37000	C	T	D		C	T
38000	C	T	D		C	T
39000	C	T	D		C	T
40000	C	T	D		C	T

J.3 Statistical differences between UC, CalEB, TInSSel and FDC for biocontrol scheduling

J.3.1 0 penalty points per intervention

Section E.2.1 shows the Kruskal-Wallis test values for each of the approaches with an intervention penalty of 0 points per intervention. When the AS value is less than 0.01, this shows that a statistical difference exists between at least two of the crossover approaches at the 99% confidence level. Tables J.19 to J.21 detail which crossover approaches are statistically significantly different for mutation levels of 0, 0.005 and 0.05 respectively.

Table J.19: Mutation rate of 0

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
100						
200		T	F		F	
300	C	T	F	T		
400	C	T	F	T	F	
500	C	T	F			
600	C	T	F			
700	C	T	F			
800	C	T	F			
900	C	T	F			
1000	C	T	F			

Continued on Next Page…

Table J.19 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
1100	C	T	F			
1200	C	T	F			
1300	C	T	F			
1400	C	T	F			
1500	C	T	F			
1600	C	T	F			T
1700	C	T	F			T
1800	C	T	F		C	T
1900	C	T	F		C	T
2000	C	T	F		C	T
2100	C	T	F		C	T
2200	C	T	F		C	T
2300	C	T	F		C	T
2400	C	T	F		C	T
2500	C	T	F		C	T
2600	C	T	F		C	T
2700	C	T	F		C	T
2800	C	T	F		C	T
2900	C	T	F		C	T
3000	C	T	F		C	T
3100	C	T	F		C	T
3200	C	T	F		C	T
3300	C	T	F		C	T
3400	C	T	F		C	T
3500	C	T	F		C	T
3600	C	T	F		C	T
3700	C	T	F		C	T
3800	C	T	F		C	T
3900	C	T	F		C	T
4000	C	T	F		C	T
4100	C	T	F		C	T
4200	C	T	F		C	T
4300	C	T	F		C	T
4400	C	T	F		C	T
4500	C	T	F		C	T
4600	C	T	F		C	T
4700	C	T	F		C	T
4800	C	T	F		C	T
4900	C	T	F		C	T
5000	C	T	F		C	T

Table J.20: Mutation rate of 0.005

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
100						
200		T	F	T	F	
300		T	F	T	F	
400	C	T	F	T	F	

Continued on Next Page…

Table J.20 – Continued

FFE(s)	**U/C**	**U/T**	**U/F**	**C/T**	**C/F**	**T/F**
500	C	T	F	T		
600	C	T	F			
700	C	T	F			
800	C	T	F			
900	C	T	F			
1000	C	T	F			
1100	C	T	F			
1200	C	T	F			
1300	C	T	F			
1400	C	T	F			
1500	C	T	F			
1600	C	T	F			
1700	C	T	F			
1800	C	T	F			
1900	C	T	F			
2000	C	T	F			
2100	C	T	F			
2200	C	T	F			
2300	C	T	F			
2400	C	T	F			T
2500	C	T	F			
2600	C	T	F			
2700	C	T	F			
2800	C	T	F			
2900	C	T	F			
3000	C	T	F			
3100	C	T	F			
3200	C	T	F			
3300	C	T	F			
3400	C	T	F			
3500	C	T	F			
3600	C	T	F			
3700	C	T	F			
3800	C	T	F			
3900	C	T	F			
4000	C	T	F			
4100	C	T	F			
4200	C	T	F			
4300	C	T	F			
4400	C	T	F			
4500	C	T				
4600	C	T				
4700	C	T				
4800						
4900						
5000						

Table J.21: Mutation rate of 0.05

FFE(s)	**U/C**	**U/T**	**U/F**	**C/T**	**C/F**	**T/F**
100						
200			F			
300		T	F		F	
400	C	T	F	T	F	
500	C	T	F		F	
600	C	T	F			
700	C	T	F			
800	C	T	F			
900	C	T	F			
1000	C	T	F			
1100	C	T	F			
1200	C	T	F			
1300	C	T	F			
1400	C	T	F			
1500	C	T	F			
1600	C	T	F			
1700	C	T	F			
1800	C	T	F			
1900	C	T	F			
2000	C	T	F			
2100	C	T	F			
2200	C	T	F			
2300	C	T	F			
2400	C	T	F			
2500	C	T	F			
2600	C	T	F			
2700	C	T	F			
2800	C	T	F			
2900	C	T	F			
3000	C	T	F			
3100	C	T	F			
3200	C	T	F			
3300	C	T	F			
3400	C	T	F			
3500	C	T	F			
3600	C	T	F			
3700	C	T	F			
3800	C	T	F			
3900	C	T	F			
4000	C	T	F			
4100	C	T	F			
4200	C	T	F			
4300	C	T	F			
4400	C	T	F			
4500	C	T	F			
4600	C	T	F			
4700	C	T	F			
4800	C	T	F			
4900	C	T	F			
5000	C	T	F			

J.3.2 5 penalty points per intervention

Section E.2.2 shows the Kruskal-Wallis test values for each of the approaches with an intervention penalty of 5 points per intervention. When the AS value is less than 0.01, this shows that a statistical difference exists between at least two of the crossover approaches at the 99% confidence level. Tables J.22 to J.24 detail which crossover approaches are statistically significantly different for mutation levels of 0, 0.005 and 0.05 respectively.

Table J.22: Mutation rate of 0

FFE(s)	**U/C**	**U/T**	**U/F**	**C/T**	**C/F**	**T/F**
100						
200	U			T	F	
300	U			T	F	
400	U			T	F	
500	U			T	F	
600	U			T	F	
700	U		F	T	F	
800	U	T	F	T	F	
900	U	T	F	T	F	
1000	U	T	F	T	F	
1100	U	T	F	T	F	
1200	U	T	F	T	F	
1300	U	T	F	T	F	
1400	U	T	F	T	F	
1500	U	T	F	T	F	
1600	U	T	F	T	F	
1700	U	T	F	T	F	
1800	U			T	F	
1900	U			T	F	
2000	U			T	F	
2100	U		U	T	F	T
2200	U		U	T	F	T
2300	U		U	T	F	T
2400	U		U	T	F	T
2500	U		U	T	F	T
2600	U		U	T		T
2700	U		U	T		T
2800	U		U	T		T
2900	U		U	T		T
3000	U		U	T		T
3100	U		U	T		T
3200	U		U	T		T
3300	U		U	T		T
3400	U		U	T		T
3500	U		U	T		T
3600	U		U	T		T
3700	U		U	T		T
3800	U		U	T		T
3900	U		U	T		T
4000	U		U	T		T
4100	U		U	T		T
4200	U		U	T		T

Continued on Next Page…

Table J.22 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
4300	U		U	T		T
4400	U		U	T		T
4500	U		U	T		T
4600	U		U	T		T
4700	U		U	T		T
4800	U		U	T		T
4900	U		U	T		T
5000	U		U	T		T

Table J.23: Mutation rate of 0.005

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
100						
200	U			T	F	
300	U			T	F	
400	U			T	F	
500	U			T	F	
600	U		F	T	F	
700	U		F	T	F	
800	U	T	F	T	F	
900	U	T	F	T	F	
1000	U	T	F	T	F	
1100	U	T	F	T	F	
1200	U	T	F	T	F	
1300	U	T	F	T	F	
1400	U	T	F	T	F	
1500	U	T	F	T	F	
1600	U	T	F	T	F	
1700	U	T	F	T	F	
1800	U	T	F	T	F	
1900	U	T		T	F	
2000	U	T		T	F	
2100	U	T		T	F	
2200	U			T	F	T
2300	U			T	F	T
2400	U			T	F	T
2500	U			T	F	T
2600	U			T	F	
2700	U			T	F	
2800	U			T	F	
2900	U			T	F	
3000	U			T	F	
3100	U			T	F	
3200				T	F	
3300				T	F	
3400				T		
3500						
3600						

Continued on Next Page. . .

Table J.23 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
3700						
3800						
3900						
4000						
4100						
4200						
4300						
4400						
4500						
4600						
4700						
4800						
4900						
5000						

Table J.24: Mutation rate of 0.05

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
100						
200	U			T	F	
300	U			T	F	
400	U			T	F	
500	U			T	F	
600	U			T	F	
700	U			T	F	
800	U		F	T	F	
900	U		F	T	F	
1000	U		F	T	F	F
1100	U		F	T	F	F
1200	U		F	T	F	F
1300		T	F	T	F	F
1400		T	F	T	F	F
1500		T	F	T	F	F
1600		T	F	T	F	F
1700		T	F	T	F	F
1800		T	F	T	F	F
1900		T	F	T	F	F
2000		T	F	T	F	F
2100		T	F	T	F	F
2200		T	F	T	F	F
2300		T	F	T	F	F
2400		T	F	T	F	F
2500		T	F	T	F	
2600		T	F	T	F	
2700		T	F	T	F	
2800		T	F	T	F	
2900		T	F	T	F	
3000		T	F	T	F	

Continued on Next Page…

Table J.24 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
3100		T	F	T	F	
3200		T	F	T	F	
3300		T	F	T	F	
3400		T	F	T	F	
3500		T	F	T	F	
3600		T	F	T	F	
3700		T	F	T	F	
3800		T	F	T	F	
3900		T	F	T	F	
4000		T	F	T	F	
4100		T	F	T	F	
4200		T	F	T	F	
4300		T	F	T	F	
4400		T	F	T	F	
4500		T	F	T	F	
4600		T	F	T	F	
4700		T	F	T	F	
4800		T	F	T	F	
4900		T	F	T	F	
5000		T	F	T	F	

J.3.3 20 penalty points per intervention

Section E.2.3 shows the Kruskal-Wallis test values for each of the approaches with an intervention penalty of 20 points per intervention. When the AS value is less than 0.01, this shows that a statistical difference exists between at least two of the crossover approaches at the 99% confidence level. Tables J.25 to J.27 detail which crossover approaches are statistically significantly different for mutation levels of 0, 0.005 and 0.05 respectively.

Table J.25: Mutation rate of 0

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
100						
200			F		F	F
300	C	T	F		F	F
400	C	T	F		F	F
500	C	T	F		F	F
600	C	T	F		F	F
700	C	T	F		F	F
800	C	T	F	T	F	F
900	C	T	F	T	F	
1000	C	T	F	T	F	
1100	C	T	F	T	F	
1200	C	T	F	T	F	
1300	C	T	F	T		
1400	C	T	F	T		T
1500	C	T	F	T		T
1600	C	T	F	T		T
1700	C	T	F	T		T

Continued on Next Page…

Table J.25 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
1800		T		T		T
1900		T		T		T
2000		T		T		T
2100				T		T
2200			U	T		T
2300			U	T		T
2400			U	T		T
2500			U	T		T
2600			U	T		T
2700			U	T		T
2800			U	T		T
2900			U	T		T
3000			U	T		T
3100			U	T		T
3200			U	T		T
3300			U	T		T
3400			U	T		T
3500			U	T		T
3600			U	T		T
3700			U	T		T
3800			U	T		T
3900			U	T		T
4000			U	T		T
4100			U	T		T
4200			U	T		T
4300			U	T		T
4400			U	T		T
4500			U	T		T
4600			U	T		T
4700			U	T		T
4800			U	T		T
4900			U	T		T
5000			U	T		T

Table J.26: Mutation rate of 0.005

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
100						
200		T	F		F	
300	C	T	F		F	F
400	C	T	F		F	F
500	C	T	F		F	F
600	C	T	F		F	F
700	C	T	F		F	F
800	C	T	F		F	F
900	C	T	F	T	F	F
1000	C	T	F	T	F	
1100	C	T	F	T	F	

Continued on Next Page…

Table J.26 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
1200	C	T	F	T	F	
1300	C	T	F	T	F	
1400	C	T	F	T	F	
1500	C	T	F	T		
1600	C	T	F	T		
1700	C	T	F	T		
1800	C	T	F			
1900	C	T	F			
2000	C	T	F			
2100	C	T	F			
2200	C	T	F			
2300	C	T	F			
2400		T	F			
2500						
2600						
2700						
2800						
2900						
3000						
3100						
3200						
3300						
3400						
3500						
3600						
3700						
3800						
3900						
4000						
4100						
4200						
4300						
4400						
4500						
4600						
4700						
4800						
4900						
5000						

Table J.27: Mutation rate of 0.05

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
100						
200			F			F
300	C	T	F		F	F
400	C	T	F		F	F
500	C	T	F		F	F

Continued on Next Page…

Table J.27 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
600	C	T	F		F	F
700	C	T	F		F	F
800	C	T	F		F	F
900	C	T	F		F	F
1000	C	T	F		F	F
1100	C	T	F		F	F
1200	C	T	F		F	F
1300	C	T	F		F	F
1400	C	T	F		F	F
1500	C	T	F		F	F
1600	C	T	F		F	F
1700	C	T	F		F	F
1800	C	T	F		F	F
1900	C	T	F		F	F
2000	C	T	F		F	F
2100	C	T	F		F	F
2200	C	T	F		F	F
2300	C	T	F		F	F
2400	C	T	F		F	F
2500	C	T	F		F	F
2600	C	T	F		F	F
2700	C	T	F		F	F
2800	C	T	F		F	F
2900	C	T	F		F	F
3000	C	T	F		F	F
3100	C	T	F		F	F
3200	C	T	F		F	F
3300	C	T	F		F	F
3400	C	T	F		F	F
3500	C	T	F		F	F
3600	C	T	F		F	F
3700	C	T	F		F	F
3800	C	T	F		F	F
3900	C	T	F		F	F
4000	C	T	F		F	
4100	C	T	F		F	
4200	C	T	F		F	F
4300	C	T	F		F	
4400	C	T	F		F	
4500	C	T	F		F	
4600	C	T	F		F	
4700	C	T	F		F	F
4800	C	T	F		F	F
4900	C	T	F		F	F
5000	C	T	F		F	

J.3.4 35 penalty points per intervention

Section E.2.4 shows the Kruskal-Wallis test values for each of the approaches with an intervention penalty of 35 points per intervention. When the AS value is less than 0.01, this shows that a statistical difference exists between at least two of the crossover approaches at the 99% confidence level. Tables J.28 to J.30 detail which crossover approaches are statistically significantly different for mutation levels of 0, 0.005 and 0.05 respectively.

Table J.28: Mutation rate of 0

FFE(s)	**U/C**	**U/T**	**U/F**	**C/T**	**C/F**	**T/F**
100						
200	C	T	F		F	F
300	C	T	F		F	F
400	C	T	F		F	F
500	C	T	F		F	F
600	C	T	F		F	F
700	C	T	F		F	F
800	C	T	F		F	F
900	C	T	F		F	
1000	C	T	F	T	F	
1100	C	T	F	T		
1200	C	T	F	T		T
1300	C	T	F			T
1400	C	T	F			T
1500	C	T	F		C	T
1600	C	T	F		C	T
1700	C	T			C	T
1800	C	T			C	T
1900		T	U		C	T
2000			U		C	T
2100			U		C	T
2200			U		C	T
2300			U		C	T
2400			U		C	T
2500			U		C	T
2600			U		C	T
2700			U		C	T
2800			U		C	T
2900			U		C	T
3000			U		C	T
3100			U		C	T
3200			U		C	T
3300			U		C	T
3400			U		C	T
3500			U		C	T
3600			U		C	T
3700			U		C	T
3800			U		C	T
3900			U		C	T
4000			U		C	T
4100			U		C	T
4200			U		C	T
4300			U		C	T
4400			U		C	T
4500			U		C	T
4600			U		C	T
4700			U		C	T
4800			U		C	T

Continued on Next Page…

Table J.28 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
4900			U		C	T
5000			U		C	T

Table J.29: Mutation rate of 0.005

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
100						
200	C	T	F		F	F
300	C	T	F		F	F
400	C	T	F		F	F
500	C	T	F		F	F
600	C	T	F		F	F
700	C	T	F		F	F
800	C	T	F		F	F
900	C	T	F		F	F
1000	C	T	F		F	F
1100	C	T	F	T	F	
1200	C	T	F	T	F	
1300	C	T	F		F	
1400	C	T	F			
1500	C	T	F			
1600	C	T	F			
1700	C	T	F			
1800	C	T	F			
1900	C	T	F			
2000		T	F			
2100		T				
2200						
2300						
2400						
2500						
2600						
2700						
2800						
2900						
3000						
3100						
3200						
3300						
3400						
3500						
3600						
3700						
3800						
3900						
4000						
4100						
4200						

Continued on Next Page…

Table J.29 – Continued

FFE(s)	**U/C**	**U/T**	**U/F**	**C/T**	**C/F**	**T/F**
4300						
4400						
4500						
4600						
4700						
4800						
4900						
5000						

Table J.30: Mutation rate of 0.05

FFE(s)	**U/C**	**U/T**	**U/F**	**C/T**	**C/F**	**T/F**
100						
200	C	T	F		F	F
300	C	T	F		F	F
400	C	T	F		F	F
500	C	T	F		F	F
600	C	T	F		F	F
700	C	T	F		F	F
800	C	T	F		F	F
900	C	T	F		F	F
1000	C	T	F		F	F
1100	C	T	F		F	F
1200	C	T	F		F	F
1300	C	T	F		F	F
1400	C	T	F		F	F
1500	C	T	F		F	F
1600	C	T	F		F	F
1700	C	T	F		F	F
1800	C	T	F		F	F
1900	C	T	F		F	F
2000	C	T	F		F	F
2100	C	T	F		F	F
2200	C	T	F		F	F
2300	C	T	F		F	F
2400	C	T	F		F	F
2500	C	T	F		F	F
2600	C	T	F		F	F
2700	C	T	F		F	F
2800	C	T	F		F	F
2900	C	T	F		F	F
3000	C	T	F		F	F
3100	C	T	F		F	F
3200	C	T	F		F	F
3300	C	T	F		F	F
3400	C	T	F		F	F
3500	C	T	F		F	F
3600	C	T	F		F	F

Continued on Next Page…

Table J.30 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
3700	C	T	F		F	
3800	C	T	F		F	F
3900	C	T	F		F	
4000	C	T	F		F	F
4100	C	T	F		F	F
4200	C	T	F		F	
4300	C	T	F		F	
4400	C	T	F		F	F
4500	C	T	F		F	F
4600	C	T	F		F	F
4700	C	T	F		F	F
4800	C	T	F		F	F
4900	C	T	F		F	F
5000	C	T	F		F	F

J.3.5 50 penalty points per intervention

Section E.2.5 shows the Kruskal-Wallis test values for each of the approaches with an intervention penalty of 50 points per intervention. When the AS value is less than 0.01, this shows that a statistical difference exists between at least two of the crossover approaches at the 99% confidence level. Tables J.31 to J.33 detail which crossover approaches are statistically significantly different for mutation levels of 0, 0.005 and 0.05 respectively.

Table J.31: Mutation rate of 0

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
100						
200		T	F		F	F
300	C	T	F		F	F
400	C	T	F		F	F
500	C	T	F		F	F
600	C	T	F		F	F
700	C	T	F		F	F
800	C	T	F		F	F
900	C	T	F		F	
1000	C	T	F			
1100	C	T	F			T
1200	C	T	F		C	T
1300	C	T	F		C	T
1400	C	T	F		C	T
1500	C	T	F		C	T
1600	C	T			C	T
1700	C	T			C	T
1800		T	U		C	T
1900			U		C	T
2000			U		C	T
2100			U		C	T
2200			U		C	T
2300			U		C	T

Continued on Next Page...

Table J.31 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
2400			U		C	T
2500			U		C	T
2600			U		C	T
2700			U		C	T
2800			U		C	T
2900			U		C	T
3000			U		C	T
3100			U		C	T
3200			U		C	T
3300			U		C	T
3400			U		C	T
3500			U		C	T
3600			U		C	T
3700			U		C	T
3800			U		C	T
3900			U		C	T
4000			U		C	T
4100			U		C	T
4200			U		C	T
4300			U		C	T
4400			U		C	T
4500			U		C	T
4600			U		C	T
4700			U		C	T
4800			U		C	T
4900			U		C	T
5000			U		C	T

Table J.32: Mutation rate of 0.005

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
100						
200	C	T	F		F	F
300	C	T	F		F	F
400	C	T	F		F	F
500	C	T	F		F	F
600	C	T	F		F	F
700	C	T	F		F	F
800	C	T	F		F	F
900	C	T	F		F	F
1000	C	T	F		F	F
1100	C	T	F			
1200	C	T	F			
1300	C	T	F			
1400	C	T	F			
1500	C	T	F			
1600	C	T	F			
1700	C	T	F			

Continued on Next Page…

Table J.32 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
1800	C	T	F			
1900	C	T	F			
2000	C	T	F			
2100						
2200						
2300						
2400						
2500						
2600						
2700						
2800						
2900						
3000						
3100						
3200						
3300						
3400						
3500						
3600						
3700						
3800						
3900						
4000						
4100						
4200						
4300						
4400						
4500						
4600						
4700						
4800						
4900						
5000						

Table J.33: Mutation rate of 0.05

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
100						
200			F		F	F
300	C	T	F		F	F
400	C	T	F		F	F
500	C	T	F		F	F
600	C	T	F		F	F
700	C	T	F		F	F
800	C	T	F		F	F
900	C	T	F		F	F
1000	C	T	F		F	F
1100	C	T	F		F	F

Continued on Next Page…

Table J.33 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
1200	C	T	F		F	F
1300	C	T	F		F	F
1400	C	T	F		F	F
1500	C	T	F		F	F
1600	C	T	F		F	F
1700	C	T	F		F	F
1800	C	T	F		F	F
1900	C	T	F		F	F
2000	C	T	F		F	F
2100	C	T	F		F	F
2200	C	T	F		F	F
2300	C	T	F		F	F
2400	C	T	F		F	F
2500	C	T	F		F	
2600	C	T	F		F	
2700	C	T	F		F	
2800	C	T	F		F	
2900	C	T	F		F	
3000	C	T	F		F	
3100	C	T	F		F	
3200	C	T	F			
3300	C	T	F			
3400	C	T	F			
3500	C	T	F			
3600	C	T	F			
3700	C	T	F			
3800	C	T	F			
3900	C	T	F			
4000	C	T	F			
4100	C	T	F			
4200	C	T	F			
4300	C	T	F			
4400	C	T	F			
4500	C	T	F			
4600	C	T	F			
4700	C	T	F			
4800	C	T	F			
4900	C	T	F			
5000	C	T	F			

J.4 Statistical differences between UC, CalEB, TInSSel and FDC for chemotherapy scheduling

Section I.2 shows the Kruskal-Wallis test values for each of the crossover approaches. When the AS value is less than 0.01, this shows that a statistical difference exists between at least two of the crossover approaches at the 99% confidence level. Tables J.34 to J.36 detail which crossover approaches are statistically significantly different for mutation levels of 0, 0.005 and 0.05 respectively.

Table J.34: Mutation rate of 0

FFE(s)	**U/C**	**U/T**	**U/F**	**C/T**	**C/F**	**T/F**
1000	C	T	F		F	F
2000	C	T	F			
3000	C	T	F			
4000	C	T	F			
5000	C	T	F			
6000	C	T	F			
7000	C	T	F			
8000	C	T	F			
9000	C	T	F			
10000	C	T	F			
11000	C	T	F			
12000	C	T	F			
13000	C	T	F			
14000	C	T	F			
15000	C	T	F			
16000	C	T	F			
17000	C	T	F			
18000	C	T	F			
19000	C	T	F			
20000	C	T	F			
21000	C	T	F			
22000	C	T	F			
23000	C	T	F			
24000	C	T	F			
25000	C	T	F			
26000	C	T	F			
27000	C	T	F			
28000	C	T	F			
29000	C	T	F			
30000	C	T	F			
31000	C	T	F			
32000	C	T	F			
33000	C	T	F			
34000	C	T	F			
35000	C	T	F			
36000	C	T	F			
37000	C	T	F			
38000	C	T	F			
39000	C	T	F			
40000	C	T	F			

Table J.35: Mutation rate of 0.005

FFE(s)	**U/C**	**U/T**	**U/F**	**C/T**	**C/F**	**T/F**
1000	C	T	F		F	F
2000	C	T	F		F	F
3000	C	T	F		F	F

Continued on Next Page. . .

Table J.35 – Continued

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
4000	C	T	F			F
5000	C	T	F			
6000	C	T	F			
7000	C	T	F			
8000	C	T	F			
9000	C	T	F			
10000	C	T	F			
11000	C	T	F			
12000	C	T	F			
13000	C	T	F			
14000	C	T	F			
15000	C	T	F			
16000	C	T	F			
17000	C	T	F			
18000	C	T	F			
19000	C	T	F			
20000	C	T	F			
21000	C	T	F			
22000	C	T	F			
23000	C	T	F			
24000	C	T	F			
25000	C	T	F			
26000	C	T	F			
27000	C	T	F			
28000	C	T	F			
29000	C	T	F			
30000	C	T	F			
31000	C	T	F			
32000	C	T	F			
33000	C	T	F			
34000	C	T	F			
35000	C	T	F			
36000	C	T	F			
37000	C	T	F			
38000	C	T	F			
39000	C	T	F			
40000	C	T	F			

Table J.36: Mutation rate of 0.05

FFE(s)	U/C	U/T	U/F	C/T	C/F	T/F
1000	C	T	F		F	F
2000	C	T	F		F	F
3000	C	T	F		F	F
4000	C	T	F		F	
5000	C	T	F		F	
6000	C	T	F		F	
7000	C	T	F		F	

Continued on Next Page…

Table J.36 – Continued

FFE(s)	**U/C**	**U/T**	**U/F**	**C/T**	**C/F**	**T/F**
8000	C	T	F		F	
9000	C	T	F		F	
10000	C	T	F		F	
11000	C	T	F		F	
12000	C	T	F		F	
13000	C	T	F		F	
14000	C	T	F		F	
15000	C	T	F		F	
16000	C	T	F		F	
17000	C	T	F		F	
18000	C	T	F		F	
19000	C	T	F		F	
20000	C	T	F		F	
21000	C	T	F		F	
22000	C	T	F		F	
23000	C	T	F		F	
24000	C	T	F		F	
25000	C	T	F		F	
26000	C	T	F		F	
27000	C	T	F		F	
28000	C	T	F		F	
29000	C	T	F		F	
30000	C	T	F		F	
31000	C	T	F		F	
32000	C	T	F		F	F
33000	C	T	F		F	
34000	C	T	F		F	
35000	C	T	F		F	F
36000	C	T	F		F	F
37000	C	T	F		F	
38000	C	T	F		F	F
39000	C	T	F		F	F
40000	C	T	F		F	F