AN APPLICATION OF GENETIC ALGORITHMS TO CHEMOTHERAPY TREATMENT

Andrei Petrovski

A thesis submitted in partial fulfilment of the requirements of
The Robert Gordon University for the degree of Doctor of Philosophy

December 1998

DECLARATION

While registered as a candidate for the Robert Gordon University's research degree, I have not been a registered candidate for another award of any other University.

This thesis is the result of my own work and as far as I am aware is completely original. Clear references are made throughout the thesis where the work of others has been used or the ideas of others have been developed.

At the same time, this thesis is a part of an on-going research project involving external collaboration with practising oncologists and aiming at the development of an interactive decision support system for intelligent drug trials and cancer treatment composition. All biomedical information and data were either provided or approved by the medical collaborators.

Parts of this thesis have been written up in the following papers, which are at various stages in the publication process:

- PETROVSKI A, McCALL J and FORREST E, 1998. An Application of Genetic Algorithms to Optimisation of Cancer Chemotherapy. International Journal of Mathematical Education in Science and Technology, **29**(3), pp. 377-388.

- McCALL J and PETROVSKI A, 1996. Searching for optimal strategies in cancer chemotherapy using genetic algorithms. Oxford University Press: Seventh IMA Conference on Mathematics in Medicine and Biology. Oxford, July 1996.

- PETROVSKI A and McCALL J, 1997. Optimising GA parameters using statistical approaches. Proceedings of the First International Workshop on Frontiers in Evolutionary Algorithms. North Carolina, USA, March 1997.

- PETROVSKI A, WILSON A and McCALL J, 1998. Statistical analysis of genetic algorithms and inference about optimal parameters. Technical Report RGU-SCMS-98/2, The Robert Gordon University, Aberdeen, Scotland, 1998.

- McCALL J and PETROVSKI A, 1999. A Decision Support System for Cancer Chemotherapy using Genetic Algorithms. Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation: Vienna, Austria. IOS Press, ISBN 90 5199 474 5, pp. 65-70.

ABSTRACT

The present work investigates methods for optimising cancer chemotherapy within the bounds of clinical acceptability and making this optimisation easily accessible to oncologists. Clinical oncologists wish to be able to improve existing treatment regimens in a systematic, effective and reliable way. In order to satisfy these requirements a novel approach to chemotherapy optimisation has been developed, which utilises Genetic Algorithms in an intelligent search process for good chemotherapy treatments.

The following chapters consequently address various issues related to this approach. Chapter 1 gives some biomedical background to the problem of cancer and its treatment. The complexity of the cancer phenomenon, as well as the multi-variable and multi-constrained nature of chemotherapy treatment, strongly support the use of mathematical modelling for predicting and controlling the development of cancer. Some existing mathematical models, which describe the proliferation process of cancerous cells and the effect of anti-cancer drugs on this process, are presented in Chapter 2. Having mentioned the control of cancer development, the relevance of optimisation and optimal control theory becomes evident for achieving the optimal treatment outcome subject to the constraints of cancer chemotherapy. A survey of traditional optimisation methods applicable to the problem under investigation is given in Chapter 3 with the conclusion that the constraints imposed on cancer chemotherapy and general non-linearity of the optimisation functionals associated with the objectives of cancer treatment often make these methods of optimisation ineffective. Contrariwise, Genetic Algorithms (GAs), featuring the methods of evolutionary search and optimisation, have recently demonstrated in many practical situations an ability to quickly discover useful solutions to highly-constrained, irregular and discontinuous problems that have been difficult to solve by traditional optimisation methods. Chapter 4 presents the essence of Genetic Algorithms, as well as their salient features and properties, and prepares the ground for the utilisation of Genetic Algorithms for optimising cancer chemotherapy treatment.

The particulars of chemotherapy optimisation using Genetic Algorithms are given in Chapter 5 and Chapter 6, which present the original work of this thesis. In Chapter 5 the optimisation problem of single-drug chemotherapy is formulated as a search task and solved by several numerical methods. The results obtained from different optimisation methods are used to assess the quality of the GA solution and the effectiveness of Genetic Algorithms as a whole. Also, in Chapter 5 a new approach to tuning GA factors is developed, whereby the optimisation performance of Genetic Algorithms can be significantly improved. This approach is based on statistical inference about the significance of GA factors and on regression analysis of the GA performance. Being less computationally intensive compared to the existing methods of GA factor adjusting, the newly developed approach often gives better tuning results.

Chapter 6 deals with the optimisation of multi-drug chemotherapy, which is a more practical and challenging problem. Its practicality can be explained by oncologists' preferences to administer anti-cancer drugs in various combinations in order to better cope with the occurrence of drug resistant cells. However, the imposition of strict toxicity constraints on combining various anti-cancer drugs together, makes the optimisation problem of multi-drug chemotherapy very difficult to solve, especially when complex treatment objectives are considered. Nevertheless, the experimental results of Chapter 6 demonstrate that this problem is tractable to Genetic Algorithms, which are capable of finding good chemotherapeutic regimens in different treatment situations. On the basis of these results a decision has been made to encapsulate Genetic Algorithms into an independent optimisation module and to embed this module into a more general and user-oriented environment – the Oncology Workbench. The particulars of this encapsulation and embedding are also given in Chapter 6.

Finally, Chapter 7 concludes the present work by summarising the contributions made to the knowledge of the subject treated and by outlining the directions for further investigations. The main contributions are: (1) a novel application of the Genetic Algorithm technique in the field of cancer chemotherapy optimisation, (2) the development of a statistical method for tuning the values of GA factors, and (3) the development of a robust and versatile optimisation utility for a clinically usable decision support system. The latter contribution of this thesis creates an opportunity to widen the application domain of Genetic Algorithms within the field of drug treatments and to allow more clinicians to benefit from utilising the GA optimisation.

ACKNOWLEDGEMENTS

During the work on this project I have received invaluable help from a number of people. Collectively, I am indebted to all my supervisors with whom I have been working for the last three and a half years. First of all, I would like to thank my director of studies John McCall for all the support, advice and critical feedback he provided during the course of my research. It has been a very fruitful and enjoyable experience to work with John and to have had many thought provoking discussions with him. Secondly, I would like to express my deep gratitude to Ernie Forrest, who introduced me to some of the theoretical and practical aspects of optimisation and optimal control. I will not forget how much help I received from Ernie during the initial period of my work on this project and it is my belief that I should also thank him for the opportunity to stay at The Robert Gordon University (RGU) in order to complete my PhD. Thanks are also due to the other members of the supervisory team – John Usher and Alex Wilson. I am grateful to John for offering his expertise in the field of mathematical modelling of cancer related processes and for his numerous helpful comments on the drafts of this thesis. Alex provided much appreciated help with all statistical aspects of the present work.

While working in RGU, I was fortunate to have collaborated with Jim Cassidy and Howard McLeod from the Department of Medicine and Therapeutics at Aberdeen University, who acted in an advisory capacity and provided assistance regarding all biomedical issues. Also I would like to extend my thanks to Ann Gavriel, the English language support tutor, who helped me a great deal to improve my knowledge of the language. Without the support, assistance and help of the people mentioned above this treatise would not have appeared and as such I would like to reiterate my thanks to them once more.

I wish to gratefully acknowledge the financial support given by the Russian Federation State Committee of Higher Education (grant #551/07.06.94) in the form of the Presidential scholarship for studying abroad. My arrival in the United Kingdom and starting the work on this project became possible only because the Committee provided all the necessary assistance. Moreover, a number of people in the Samara State Technical University, many more than I could name individually, have in some way contributed to the realisation of the opportunity for my study abroad. Of these people I would like to single out firstly Vitaly I. Batischev and Evgheny I. Tatarenko, who strongly encouraged me to participate in the competition for the Presidential scholarship, and secondly Anatoly N. Malyarov and Galina I. Bogolubova, who recommended RGU in the first place and who made all the necessary arrangements with the officials over here. With respect to further financial support received during the course of my study, I would also like to acknowledge a support grant from the Sir Richard Stapley Educational Trust.

Last but not least thanks must go to my mum and brother Serghei, who, although being far away, always provided moral support, much needed encouragement and who uncomplainingly coped with my long absence.

CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABBREVIATIONS

CM	Complex Method
GAs	Genetic Algorithms
HCM	Homeostatic Control Mechanism
HJ	Hooke and Jeeves Method
IR	Information Repository
LP	Linear Programming
MB	Mating Buffer
MTD	Maximum Tolerable Dose
OE	Optimisation Engine
OW	Oncology Workbench
PF	Penalty Function
PK/PD	Pharmacokinetic/Pharmacodynamic
PST	Patient Survival Time
RV	Results Viewer
SE	Simulation Engine
TE	Treatment Editor

CHAPTER 0

INTRODUCTION

The thrust of the present work is to find a *robust*, *effective* and *practically usable* method for optimising cancer chemotherapy treatment within the bounds of clinical acceptability. These characteristics of the optimisation method are pivotal for allowing oncologists to improve existing treatment regimens in a systematic and reliable way. A few words need to be said about what they actually mean in the context of cancer chemotherapy.

Firstly, robustness of an optimisation method provides the means to tackle multiple constraints of cancer chemotherapy and the paucity of information on exact values of tumour characteristics. The latter factor is particularly important since it is known that amongst cancer patients there is a large variation of tumour growth parameters and of tumour response to anti-cancer drugs. Secondly, if the method of chemotherapy optimisation is effective, it expedites the finding of better chemotherapy schedules in different treatment scenarios. The sort of information provided by the optimisation method should assist oncologists in either developing novel chemotherapy schedules *ab initio* or in modifying some of the existing ones, behind which there is a considerable clinical experience. The final requirement to the method of optimising chemotherapy treatments is its practical usability. Nowadays oncologists tend to develop chemotherapeutic regimens which consist of more than one anti-cancer agent. Therefore, the optimisation method must be able to deal with multi-drug therapies and to communicate with oncologists in a user-friendly fashion, i.e. without delving into details of mathematical modelling and optimisation.

The present thesis is devoted to the development of such a method for optimising cancer chemotherapy treatment. The author hopes that the context and the order of the chapters comprising the thesis will help the reader to see and to follow this development. The issues addressed in each chapter are briefly described below so that the aims of the thesis can be appreciated and the main contributions can be identified.

1. Biology and Medicine

Chapter 1 gives some biomedical background to the problem of cancer and its treatment. The following questions are addressed:

- what cancer is;
- what the objectives of cancer treatment are;
- what constraints are imposed on cancer treatment;

and finally

- why chemotherapy has been chosen in this thesis as a cancer treatment modality.

2. Mathematical Models of Tumour Kinetics

The complexity of the cancer phenomenon as well as the multi-variable and multi-constrained nature of chemotherapy treatment strongly support the use of mathematical modelling for predicting and controlling the development of cancer. In Chapter 2 a range of existing mathematical models are discussed which deal with such processes as tumour growth and pharmacodynamic effects of anti-cancer drugs. The purpose of this discussion is to select the most general mathematical model, which provides the base for further investigations. The generality of the selected model allows a better fit with a wide range of tumour sizes and therefore makes the model applicable to a larger number of cancer patients. Moreover, a general model has fewer uncertain parameters and this can make the following optimisation easier and more reliable.

3. Mathematical Optimisation of Chemotherapy

Having specified the model of tumour growth and its response to chemotherapy, the objectives of cancer treatment can be formulated mathematically and various methods of mathematical optimisation become applicable. Chapter 3 first gives a survey of mathematical optimisation methods and then shows how some of these methods are applied in the context of cancer chemotherapy. However, traditional optimisation methods have a number of limitations, which are identified in the conclusion of this chapter, and elaborated in Chapter 5.

4. Genetic Algorithms as a Method of Evolutionary Search and Optimisation

To overcome the limitations of mathematical optimisation, an evolutionary approach to developing and improving chemotherapy regimens is proposed in this thesis. This approach is featured by the method of Genetic Algorithms, which has recently demonstrated an ability to quickly discover useful solutions to highly-constrained, irregular and discontinuous problems in many practical situations. In Chapter 4 the evolutionary concept, the rationale and the salient features of Genetic Algorithms are described in order to prepare the ground for applying this method to the problem of chemotherapy optimisation. An application of Genetic Algorithms constitutes the essence and the main innovative aspect of the present work and is thoroughly explored in Chapters 5 and 6.

5. Single-Drug Chemotherapy Optimisation using Genetic Algorithms

The proposal of a novel optimisation method capable of dealing with a certain problem is not enough for justifying the utilisation of this method in practical situations. Firstly the new method has to be tested and compared with the existing optimisation techniques and that is what this chapter is devoted to. The test-bed based on the Linear Programming formulation of the problem of single-drug chemotherapy optimisation is used in the present work to assess the quality of the GA solution and the efficiency of Genetic Algorithms as a whole. In order to improve the efficiency of Genetic Algorithms in finding better chemotherapeutic regimens, a new approach to tuning the factors affecting GA performance is developed and analysed in this chapter and subsequently used in Chapter 6.

6. Multi-Drug Chemotherapy Optimisation using Genetic Algorithms

Chapter 6 applies Genetic Algorithms to the problem of multi-drug chemotherapy optimisation, which does not have a known solution. The experimental results of this chapter demonstrate that the latter problem is tractable to Genetic Algorithms which are capable of finding viable treatment regimens for different optimisation objectives. On the basis of these results a decision has been made to encapsulate Genetic Algorithms into an independent optimisation module and to embed this module into a more general and user-oriented environment – the Oncology Workbench. The aim of this encapsulation and embedding is to enhance the practical usability of Genetic Algorithms.

7. Conclusions

The final chapter concludes the present work by summarising the contributions made to the knowledge of the subject treated and by outlining the directions for further investigations. The main achievement of this thesis is the development of a robust, effective and practically usable method for optimising cancer chemotherapy. This method is based on the idea of evolving various treatment regimens to produce better chemotherapeutic schedules. The evolution is implemented by Genetic Algorithms, the medical application domain of which is not restricted to chemotherapy alone but is open to many other therapeutic treatments.

CHAPTER 1

BIOLOGY AND MEDICINE

Before a practical problem can be mathematically analysed or an attempt made to find its analytical or numerical solution it is necessary to gain an understanding of the actual problem. In order to do this for optimisation of cancer chemotherapy treatment a number of preliminary issues need to be addressed. Firstly, an understanding is required of the complexity of cancer in terms of why, despite many years of intensive research, treatment of the disease still retains a number of uncertain features. It is for this reason that novel approaches to the improvement of cancer treatment still need to be investigated and developed. Section 1.1 sheds some light on this matter by presenting a discussion on salient characteristics of cancerous diseases and on existing treatment modalities used in clinical practice.

Secondly, the knowledge of treatment goals, and of hurdles on the way to attain these goals, is vital for gaining insight into what can be achieved (and what cannot) by utilisation of existing modes of treatment – chemotherapy in particular. Hence, Section 1.2 concentrates on the discussion of cancer treatment objectives, whereas practical constraints of cancer chemotherapy will be given in Section 1.3. The details of these constraints outline the difficulties of drug schedule composition as well as disclose a need for a systematic approach to this process. In the discussion section, Section 1.4, it will be explained why this thesis focuses exclusively on chemotherapy as a cancer treatment modality to be mathematically modelled and optimised.

1.1. Complexity of the Cancer Phenomenon

Cancer belongs to the category of *neoplastic diseases*, in which inappropriate cell proliferation leads to an excess of cells (referred to as a tumour) causing disruption of normal tissue architecture (Wheldon, 1988). There is a distinction between benign and malignant tumours. Benign tumours are characterised by self-limiting growth and staying at a particular location. In

contrast, malignant tumours, viewed as common examples of cancer, have the potential to grow beyond a life-sustainable level, and malignant cells are prone to spread or metastasise to other organs and tissues. Depending on the tissue of origin, cancer is divided into three main categories: carcinoma, sarcoma and leukaemia. Although very different in their physical manifestations and properties, instances of these categories have something in common – all of them display failure of control by the body's Homeostatic Control Mechanism (HCM).

The HCM is responsible for maintaining a balanced number of cells by regulating the cell proliferation process. This process consists of four active phases: $\mathbf{G_1}$, RNA and protein synthesis; **S**, DNA synthesis; $\mathbf{G_2}$, synthesis of remaining RNA and protein; **M**, mitosis or cell division, and a resting phase $\mathbf{G_0}$ (Widnell and Pfenninger, 1990). Under normal circumstances the proliferation process is carefully controlled to ensure that there is no excess of cells. If however the control breaks down then the organism's well being is at risk.

There are many possible reasons for the cell starting to act in contravention to the HCM. The mutation theory (Varmus and Weinberg, 1993) suggests that the process of the HCM infringement usually starts during the division phase in the cell proliferation cycle. This process results in partial loss or alteration of the DNA code of the new-formed cell, which could stop 'being programmed' to obey the HCM. The DNA alteration may occur due to a viral intrusion or due to a transformation inside the cell caused by a physical, chemical or biological process. Because of these reasons the cellular de-differentiation may commence which means that the mutant cell loses its organ- or tissue-specific traits and does not function in accordance with its intent. Instead, the mutant cell utilises all consumed nutrition for self-reproduction (Varmus and Weinberg, 1993).

The cellular de-differentiation may result in two perilous outcomes. Firstly, the excessive self-reproduction of de-differentiated cells (some of which eventually become cancerous, i.e. disobeying the body's HCM) precipitates more rapid growth of these cells and facilitates their domination in the surrounding neighbourhood. Secondly, it has been found (Varmus and Weinberg, 1993) that the cell de-differentiation often causes a profound shift in the relationship between the cells' nuclei and cytoplasm with the result that the mutant cells contain increased amounts of DNA. The excess in DNA quantity allows the mutant cells to acquire unusual nuclei composition after undergoing mitosis (i.e. to become cancerous cells) or unusual properties (genetic instability for example), which may subsequently lead to resistance to drugs being applied for cancer treatment.

These abnormalities, i.e. genetic instability, an increased amount of DNA, the loss of tissue-specific traits, and the ability to continuously proliferate, allow cancerous cells to increase in number, leading first to illness and then to death of the individual. In several studies attempts have been made to quantify the risk of these abnormalities occurring. Wheldon (1988), for example, gives an estimate, obtained from monitoring familial retinoblastomas, that

$\sim 1.9 \times 10^{-7}$ mutations/gene occur every year. Varmus and Weinberg (1993) give a statistic that $\sim 7.3\%$ of people acquire some type of cancer due to these mutations. To complicate the matter, the majority of them happen at the cellular level, which makes it extremely difficult to control the development of the disease. This is because at present a cell body containing less than 10^9 cells is undetectable by medical equipment (Martin and Teo, 1994). Yet modification of the process of cell proliferation can occur, and therefore the course of some neoplastic diseases may be amended. Ideally, such modifications result in restoration of the cellular HCM and often they are accomplished by cancer treatment.

The three main modes of cancer treatment are surgery, radiotherapy and chemotherapy. The surgical removal of tumours is a common mode of treatment, applied in most cases of benign tumours as well as in some cases of malignant cancers, where it is thought that metastasis has not occurred. If metastasis has occurred the success of this approach depends on how far the tumour cells have spread beyond the primary site (Wheldon, 1988). Bearing in mind that the majority of tumours are diagnosed at a late stage, the spread of cancerous cells is often too extensive for surgical procedures alone to suffice. Two other approaches are the use of ionising radiation for cancer treatment (radiotherapy) and the use of anti-cancer drugs (chemotherapy). At sufficiently high doses radiotherapy can eliminate cancer cells on a small scale by sterilising them completely. Unfortunately, high radiotherapeutic doses prohibit application on larger scales since they irreparably damage the surrounding normal tissues as well. Hence surgery and radiotherapy are best used against localised tumours. The advantage of chemotherapy, on the other hand, is its efficacy against tumour cells which have spread far beyond the original site. However, the rules of selecting a treatment modality in accordance with tumour cell dispersion are becoming more flexible. In modern cancer treatment, these modalities (surgery, radio- and chemotherapy) are often used together in various combinations (Wheldon, 1988). The goal of cancer treatment is to achieve the objectives (described in the next section), which are exemplified for the case of chemotherapy.

1.2. Treatment Objectives

Oncologists commonly use two measurements to summarise how successful the administered treatment has been – tumour reduction and patient survival time (Henderson, 1997). Therefore, the objectives of chemotherapeutic treatment will be either to reduce the size of the tumour as much as possible (ideally to eradicate the tumour entirely) or to maximise the patients' survival time (PST). These objectives are achieved by utilising rather different strategies. On the one hand, tumour eradication involves the use of high-intensity chemotherapy. For small tumour masses containing no drug resistant cells this kind of treatment is able to destroy the tumour (Martin *et al*, 1992). On the other hand, when the number of cancerous cells exceeds a certain limit, the probability that drug resistant cells are already present in the tumour becomes too large and makes the eradication of the disease unlikely or impossible (Birkhead and Gregory, 1984). In the latter case a palliative treatment is preferable based on low-intensity chemotherapy, which maintains the tumour burden at the highest level the patient can sustain,

thereby slowing down the proliferation of drug resistant cells and ultimately prolonging the PST. Besides, in designing chemotherapeutic strategies one of the major concerns is to limit the damage to normal cells while trying to reduce the cancer cell population.

Figure 1.1 graphically illustrates a typical pattern of tumour development before, during and after treatment as well as the main treatment objectives. The variable $N(t)$ on the y-axis denotes the number of cells in the tumour and therefore represents the tumour size at time t. T_0 and T_{final} specify the treatment interval on the time scale (x-axis). $N_{\max}$ is the critical tumour size, exceeding which results in the individual's death. N_{cure} is a hypothetical tumour size which has to be attained in order to prevent the tumour from regrowing. In practice N_{cure} is often substituted by the minimum detectable level. It is assumed that if the tumour cannot be detected then it has been cured. More realistic assumption, however, is that if the number of cancerous cells in the tumour is small enough (< 1,000), then these cells may disappear due to natural processes (e.g. cell necrosis or apoptosis), leading to a complete tumour elimination. Otherwise, if this does not happen, the cancerous cells that survived after treatment will start proliferating again until the tumour attains its lethal size (the broken line on the Figure 1.1 represents this scenario).

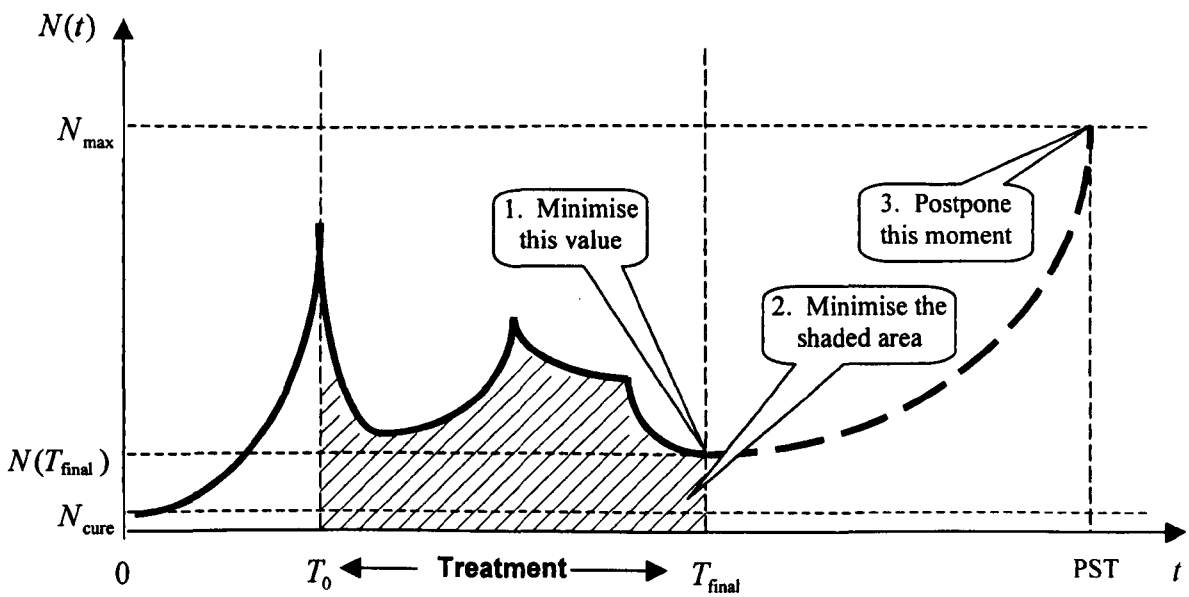


Figure 1.1. A gross pattern of tumour behaviour during its lifetime

The treatment goal of tumour size reduction may be implemented in two different ways. A simple way is to concentrate only on minimisation of the final tumour size $N(T_{\text{final}})$ after a fixed period of treatment (first objective). By implementing this strategy one can hope to drive the tumour down to N_{cure} thereby successfully accomplishing the cancer treatment. The second approach to tumour reduction is concerned not only with minimisation of $N(T_{\text{final}})$ but aims at minimising the overall tumour burden throughout the whole treatment (second objective). This

strategy attempts to keep the tumour burden to an absolute minimum at all times during the treatment interval and therefore aggressively seeks to destroy the tumour. Oncologists use this approach to treatment of cancer patients very often. The last objective is to prolong the PST.

As we said, the purpose of the graph presenting the tumour development pattern is to graphically illustrate the most common treatment objectives; more thorough mathematical formulation of these objectives will be given later. However, even without scrutinising the mathematical formulae, an observant look at Figure 1.1 may raise the question – what prevents oncologists from administering such an amount of anti-cancer drugs that will kill as many cancerous cells as required to reduce the tumour to a desirable size? To answer this question a number of constraints imposed on cancer treatment need to be introduced, a discussion of which is now provided.

1.3. Constraints of Cancer Chemotherapy

Chemotherapy is a systemic treatment; drug is delivered via the bloodstream and therefore affects all body tissues. This has advantages and disadvantages. If a tumour is sufficiently advanced it may release cancerous cells into surrounding tissue or the bloodstream (metastasis) giving rise to secondary tumours elsewhere in the body. Therefore chemotherapy has a chance of reaching secondary (possibly undetected) tumours as well as the intended primary target. However chemotherapeutic drugs are by their nature highly toxic and it is not surprising that different tissues may be dose-limiting for different drugs or drug combinations. A prime example of this is bone marrow, which can be significantly depleted during chemotherapy. Other organs may also be affected. High-dose treatment with Adriamycin, for instance, may damage the heart; Cyclophosphomide affects the bladder, and Cisplatinum causes kidney and peripheral nerve damage and injury to hearing. Many of these drugs also cause nausea and vomiting as well as detriment to the gastrointestinal tract (Wheldon, 1988).

Therefore the amount of drug which can be delivered is limited. The dosage restrictions are established experimentally and usually take the form of a maximum dosage $D_{\max}$ which can be delivered to a patient at any one time along with a cumulative dosage D_{cum} which should not be exceeded over the course of the whole treatment. However, the drug administered by injection or infusion has to circulate through the bloodstream from the point of entry to the tumour site. Along the way many processes occur, the net effect of which is not well understood. Thus, an alternative approach to drug dosage limitation is often used (Martin *et al*, 1990) whereby similar constraints are imposed on the concentration $c(t)$ of anti-cancer drug in the blood plasma. In the present work the latter approach will be adopted as more practical and independent from the drug delivery mode (McCall and Petrovski, 1996). In terms of the drug concentration $c(t)$ the first two constraints (i.e. toxicity constraints) of cancer chemotherapy may be formulated as follows.

1. **Maximum Instantaneous Concentration**

   There is a maximum permissible concentration, C_{max}, that must not be exceeded at any point during treatment.

2. **Maximum Cumulative Concentration**

   There is a maximum permissible cumulative concentration, C_{cum}, that must not be exceeded over the treatment period, $[T_0, T_{final}]$.

The next constraint is concerned with the phenomenon of drug resistance which is a frequent cause of chemotherapeutic failure in human cancers (Martin *et al*, 1994). It has been shown (Goldie and Coldman, 1979) that the probability of having drug resistant cells in the tumour, as well as their proportion to non-resistant cells, increases with increasing tumour size. In order to prevent the appearance of atypical cancer cells inherently resistant to the effects of anti-cancer drugs, it is often necessary to limit the tumour size $N(t)$ during treatment. Besides, an excessive tumour burden increases the likelihood of patient death due to inability of vital organs to function properly. Therefore there is a strong motivation for preventing the tumour from becoming too large, which is expressed in the form of the tumour size constraint.

3. **Maximum Tumour Size**

   There is a maximum permissible tumour size, N_{max}, that must not be exceeded at any point during treatment.

The last type of constraint is related to multiple drug chemotherapy and is meant to reduce the damage inflicted by administration of various drug combinations. Much of the art of successful chemotherapy depends on the full knowledge and anticipation of serious side effects caused by different anti-cancer agents or their combinations. Although it is extremely difficult to obtain quantitative measures of side effects, some empirical results are available in the form of organ- and drug-specific risk factors (Dearnaley *et al*, 1995), an example of which for the most commonly-used drugs is now provided:

TABLE 1.1. *Side effects of chemotherapeutic agents*

Anti-cancer drugs	*Side-effects*				
	Bone marrow	Kidney	Nausea, vomiting	Heart	Periph. nerves
Adriamycin	+++	-	++	++	-
Epirubicin	+++	-	++	+	-
Cisplatinum	++	+++	+++	–	+++

The entries in Table 1.1qualitatively indicate the likelihood that a given drug will cause a particular side effect seriously threatening the health of the patient. For instance, a minus means that a given drug does not have any serious effect, one plus represents a small chance of damaging a particular organ, two pluses – moderate chance, and three pluses symbolise that this drug has a sizeable chance of causing serious damage. Moreover, the number of pluses increases when two or more drugs are administered simultaneously.

It is rather difficult however to predict the exact side effects of multi-drug chemotherapy treatment. Many processes occur on the cellular level leading to various consequences of drug-drug interaction. For example, the inhibition of metabolising enzymes due to the effect of one drug in a multi-drug cocktail may cause a drastic change of toxic properties of other drugs in the cocktail. However, for the drugs that will be used throughout this thesis, interactions are known to be minor. This means that the risk factors in Table 1.1 are additive, i.e. the total number of pluses inflicted on a particular organ by a multi-drug chemotherapy schedule is equal to the sum of pluses produced by all constituent drugs.

Therefore, in order not to cause almost certain impairment of vital organs and in order to sustain reasonable living conditions for the cancer patient, a certain constraint ought to be introduced which will regulate how multi-drug chemotherapy schedules should be composed. For example, the combination of Adriamycin and Cisplatinum is preferable to the combination of Adriamycin and Epirubicin because the former never results in more than five pluses of risk for any side effect, whereas the latter accumulates six pluses on bone marrow. The introduction of a regulative constraint on composition of multi-drug chemotherapy schedules has been advised by the collaborating oncologists and is meant to prevent an extensive usage of anti-cancer agent combinations similar to the grouping of Adriamycin and Epirubicin. This constraint can be formulated in the form:

4. **Risk reduction of toxic side effects produced by multi-drug chemotherapy regimens**

   The ingredients of multi-agent chemotherapy should distribute the total damaging effect across several organs rather than stressing any particular organ to its limit.

In conclusion, it needs to be pointed out that although the four constraints formulated in this section do not seem to be very restrictive, the concurrent satisfaction of these constraints accomplished by achieving a specific treatment objective is a complex and demanding task. It is known, for example, that in order to minimise the final tumour burden $N(T_{\text{final}})$ (see Figure 1.1) a late treatment schedule should be applied which delivers all of the drug to the tumour near the end of the treatment period (Martin *et al*, 1990). This implies that no drug is given during the first half of the treatment period, and the tumour could grow unacceptably large, thereby violating the maximum tumour size constraint. This is not the only conflict between

treatment objectives and practical constraints – there are others, the resolution of which necessitates a deeper understanding of tumour growth and its response to chemotherapy. One way to acquire this understanding is to utilise the tool of mathematical modelling in order to predict and to control the course of the cancer disease when a chemotherapeutic treatment is used. The next chapter gives a survey of mathematical models related to cancer.

1.4. Discussion

Chemotherapy is the treatment modality which will be the focus of attention in this thesis. The reason for this is twofold. First of all, chemotherapy is often chosen as the first-line treatment (before any other modality is tried); sometimes chemotherapy is utilised for 'setting the scene', causing for example a tumour to decrease in size and thereby facilitating the later use of surgery or radiotherapy (Wheldon, 1988). This brings chemotherapy to the forefront of cancer treatment, making this modality frequently employed and thus necessitating its analysis and optimisation.

Secondly, chemotherapy, being a systemic treatment affecting the tumour indirectly by drug administration via the bloodstream, provides scope for mathematical modelling and optimisation. There are several mathematical models describing the development of tumours, the effect of anti-cancer drugs on this development and on other tissues and organs. The purpose of mathematical modelling therefore is to choose the most appropriate model and subsequently couple it with a suitable optimisation technique in order to improve the outcome of chemotherapeutic treatment.

However this course of actions is likely to face a number of difficulties. Due to the existence of a vast variety of anti-cancer drugs with different biochemical structure, molecular mode of action, pharmacology and toxic side effects, it is difficult to specify a general mathematical model which precisely describes the effect of each drug. This difficulty is caused by the fact that such models often need numerical values for a large number of parameters (Martin and Teo, 1994), accurate estimates of which are rarely available. Moreover, since compound regimens are often found to be more effective (Wheldon, 1988), it is usual nowadays to use cocktails of anti-cancer drugs and this makes the modelling of chemotherapy even more sophisticated.

Therefore, in order to make chemotherapy optimisation based on mathematical modelling practically useful it is necessary to meet two requirements. The first requirement is that the model has to be specified which fits as wide a range of tumour sizes as possible. This allows the model to be used in various clinical circumstances. Secondly, this model needs to be combined with a robust and efficient optimisation technique which does not heavily rely on the model's parameters, allows a certain degree of imprecision for these parameters, and is capable of working in the presence of nonlinear constraints of chemotherapeutic treatment. The goal of this thesis is to find such a combination of mathematical model and optimisation technique.

CHAPTER 2

MATHEMATICAL MODELS OF TUMOUR KINETICS

In the context of cancer treatment, mathematical modelling can be considered as a tool which if utilised effectively should provide us with an understanding of complex biological mechanisms and consequently be of value for oncologists. A good mathematical model is useful for a number of reasons. Firstly, it is often easier and cheaper to formulate a model and to use it in computer simulations than to carry out a laboratory experiment or a clinical trial. Secondly, far from all experiments can be performed on human beings for ethical and practical reasons. Finally, even when advanced state of a cancer does allow a particular clinical trial to proceed, the group of patients who satisfy the ethical criteria are not representative of typical cancer patients. This is because the standard approach is to test new drugs only when existing drugs have failed and perhaps modified the organism's responsiveness to other anti-cancer agents. Mathematical models, on the other hand, may provide us with a deeper insight into the current state of knowledge of tumour behaviour, not being concerned very much with ethical regulations and time consumed undertaking laboratory experiments or clinical trials.

The biological mechanisms involved in tumour growth and its response to treatment are exceedingly complex. Nevertheless, attempts have been made to model some of the more fundamental processes which are known to influence the treatment effectiveness. In this chapter the models will be discussed which deal with two such processes: tumour growth (Section 2.1 and 2.2) and tumour cell loss due to the administration of a chemotherapeutic agent (Section 2.3). In the last section (Section 2.4) other biological mechanisms will be mentioned for which some mathematical models have been developed.

2.1. Taxonomy of Mathematical Models of Tumour Growth

A variety of mathematical models have been utilised in an attempt to model tumour growth. Wheldon (1988) describes a mathematical model as a very exact form of a scientific theory which concentrates on the quantitative (rather than on philosophical) content of the theoretical assumptions made. However, if the assumptions are rather loosely expressed in the original theory – which is the case in the context of tumour growth (Wheldon, 1988) – then more than one mathematical model may be compatible with the theory.

In spite of the broad spectrum of growth patterns found, it is evident that some modes of growth occur more commonly than others. Typically a tumour will grow very rapidly in its initial stages with a growth rate proportional to the tumour size. However, as tumours become very large (beyond the point where they have developed their own vascular system and have become observable to the unaided eye) the growth rate is observed to decrease as tumour size increases. As may be seen from the description of the typical tumour growth pattern, the assumptions concerning the growth of tumours are defined rather loosely. It is not surprising therefore that various mathematical models have been proposed which mimic the observed behaviour. Marusic *et al* (1994) has classified them into three categories: *empirical*, *functional* and *structural*.

Empirical models are based on clinical observations confirming that growth results from two opposing processes – the increase in tumour volume and growth retardation. Due to a relatively simple form, empirical models often substantially facilitate numerical/computational solution of problems related to effective chemotherapy treatment. This is a big advantage as far as the current work is concerned, and therefore empirical models will be the main type of models which will be looked at.

Functional models stem from cell kinetics and are characterised by the cellular doubling time, the fraction of actively dividing cells (growth fraction), and the cell loss. The abnormal division rate of cancerous cells is often explained in functional terms by the process of gene amplification whereby a subsection of genetic material is duplicated. Harnevo and Agur (1992, 1993) have developed a mathematical model which describes the dynamics of gene amplification known to be a contributive factor for the defiance of HCM and for the occurrence of drug resistance. Speaking generally, however, the unifying aspect of all functional models is that they express an increase in the number of tumour cells as the difference between the cell gain (which is equal to the product of the proliferation rate and the growth fraction) and the random loss of cells . Also, functional models often allow for compartmental representation of tumours aimed at taking into account heterogeneity of the tumour cell population (Wheldon, 1988).

Structural models have been developed for description of tumour growth in structural terms. All such models assume that the tumour is a perfect sphere, the structure of which is composed of

a 'necrotic mantle' surrounded by a shell (crust) of proliferating cells (Conger and Ziskin, 1983). Supposing that such processes as proliferation, necrosis, diffusion, shedding, inhibition etc. obey spherical symmetry, the growth of a tumour can be conveniently described by its radius and that is precisely what structural models do (Landry *et al*, 1982; Wheldon, 1988).

The utilisation of mathematical models of tumour growth has a potential of being able to improve cancer chemotherapy (Swan, 1990). Each mathematical model involves a differential equation of some particular type and describes the dynamic course of development of the cancer cell population. Using this description, Webb (1992) proved that in certain cases of periodic chemotherapy treatment there is an advantage in choosing periods of shorter duration. Similarly, Panetta and Adam (1995) applied mathematical modelling of tumour growth to estimate the optimal parameter settings of acceptable dose and period of cycle-specific chemotherapy. On the other hand, by modelling and analysing the development of the normal cell population (especially bone marrow) it is possible to reduce the toxic effects of anti-cancer drugs (Agur *et al*, 1988).

Therefore, mathematical models describing the development of cell populations can play a pivotal role in optimising cancer treatment. Although all models are useful for gaining an understanding of, and further insights into, the biological mechanisms influencing cancer development, some of them are difficult to utilise in practical situations. For example, the functional and structural models require a large number of characteristics to be known or make biological assumptions that are not generally valid. These undermine their applicability in the first place and secondly complicate the procedure of coupling a mathematical model of these categories with any optimisation technique. Therefore, functional and structural models will be excluded from further consideration in this thesis and we will concentrate solely on empirical models.

2.2. Empirical Models of Tumour Growth

There is a large variance in tumour kinetics for different types of cancer cells. Yet, a particular tumour containing a large number of cells develops more predictably and hence is suitable for mathematical modelling (Martin and Teo, 1994). The use of empirical models provides a tool for determining general tumour growth patterns (Henderson, 1997). Steel (1977), Wheldon (1988) and Marusic *et al* (1994) review many mathematical models, all of which can be seen however as only an approximation of the extremely protean way in which tumours grow in vivo. The purpose of this section is to briefly discuss the empirical models that are most commonly used in cancer research.

Let us start with the introduction of two fundamental quantitative measures associated with the tumour and its growth. The most biologically meaningful measure of the tumour is the number of cells $N(t)$ it contains at time t. This number may be calculated from tumour volume

measurements since, according to (Sullivan and Salmon, 1972), there is an approximately linear relationship between tumour volume and the cell number.

Tumour growth, on the other hand, is most conveniently defined in terms of the doubling time τ of the tumour cell population. The tumour doubling time can be measured directly when the tumour exceeds the observable size of ~10^9 cells (about 1mm in diameter). This makes τ particularly suitable for empirical use. On the basis of the doubling time the following taxonomy of growth models can be employed:

- **one-phase models**: the variation of the doubling time is consistent throughout the growth of the tumour (i.e. τ is either constant or increases with tumour enlargement, but not both);
- **two-phase models**: the doubling time τ remains constant during the initial period of tumour growth but starts increasing after a certain point is reached.

In order to obtain mathematical expressions for the doubling time we first require appropriate equations governing the different types of tumour growth. For the present, attention will be restricted to continuous deterministic models described by a differential equation with general assumptions made about the way in which the tumour is growing. These assumptions are used to formulate the growth function $F(N)$: a positive-valued, continuous and monotonically increasing function that describes the increase per unit time in the tumour cell population. This leads to the following model of tumour growth:

$$\begin{aligned} &\dot{N}(t) = F(N) \\ &N(0) = N_0 \\ &\text{where } F(N) > 0 \text{ and } \frac{dF(N)}{dt} \geq 0. \end{aligned} \tag{2.1}$$

The growth function $F(N)$ can be mathematically expressed in a number of ways. The two-rate representation of the growth function in terms of the growth and the degradation rates has been suggested by von Bertalanffy (1957) and takes the form:

$$F(N) = \eta N^m - \mu N^n \tag{2.2}$$

As special cases, the growth function (2.2) yields the following well known growth equations (Marusic *et al*, 1994):

1) the exponential growth

$$m = 1,\ n = 0,\ \mu = 0: \qquad \dot{N}(t) = \eta N$$

2) the von Bertalanffy growth

$$m = \frac{2}{3},\ n = 1: \qquad \dot{N}(t) = \eta N^{\frac{2}{3}} - \mu N$$

3) the Verhulst (logistic) growth

$$m = 1,\ n = 2: \qquad \dot{N}(t) = \eta N - \mu N^2$$

4) the Gompertz growth

$$m \to 1,\ (n - m) \to 0: \qquad \dot{N}(t) = \eta N - \mu N \ln N$$

Another set of nested empirical models was proposed by Turner *et al* (1976) based on the assumption that the time derivative $\dot{N}(t)$ is proportional to the product of one function increasing with size and the other function decreasing with size $N(t)$. The corresponding genetic growth function reads as

$$F(N) = \frac{\beta}{k^n} N^{1-np} (k^n - N^n)^{1+p} \tag{2.3}$$

Special cases 1) – 4) can also be derived from the equation (2.3) subject to a proper choice of the parameter values.

Finally, Usher (1980) introduced the following Generalised growth function while investigating the radiotherapy treatment of cancer tumours:

$$F(N) = \frac{\lambda N}{\alpha}\left[1 - \left(\frac{N}{\Theta}\right)^{\alpha}\right] \tag{2.4}$$

where α, λ, Θ are growth characteristics of the tumour under investigation.

Again, appropriate limiting forms of (2.4) will yield the exponential, logistic and Gompertz growth models.

The variety of forms the growth function $F(N)$ can take brings forth the question – which of them is preferable? Unfortunately, there is no straightforward answer to this question – different models provide a better fit for different clinical data. However, general observations concerning the suitability of various mathematical models are available in the literature. For example, Steel (1977) reports that when the experimental data cover a narrow range of size $N(t)$ any of the equations described above will usually fit well. When however the data cover a wider range of sizes the Gompertz equation usually gives a better fit than any of the exponential, the logistic or the Bertalanffy equations. Wheldon (1988) also confirms that the Gompertz model is the most accepted amongst a plethora of empirical models of tumour growth. However, a disadvantage of the Gompertzian is that it does not allow for an exponential growth during the initial stage of tumour development; the Gompertz model displays retardation throughout. This makes it difficult to find an early-stage Gompertzian approximation of tumour growth which does not have an implausibly short initial doubling time.

To overcome this difficulty two-phase growth models have been proposed which combine the latent and macroscopic phases of tumour growth. One of the most commonly used two-phase models is the 'Gomp-Ex' model, described by Wheldon (1988). The 'Gomp-Ex' model suggests that tumours initially follow an exponential growth pattern which is replaced by a Gompertzian pattern after some critical cell number N_C has been reached:

$$F(N) = \begin{cases} \lambda N(t) & N \le N_C \\ \left[\lambda - \beta \ln\left(\frac{N(t)}{N_C}\right)\right] \cdot N(t) & N \ge N_C \end{cases} \qquad (2.5)$$

where N_C is the tumour size at which the transition between growth modes occurs (possibly $\sim 10^9$ cells in the case of human tumours).

Having applied the Generalised growth function (2.4) to the 'Gomp-Ex' model, Usher (1994) suggested the following 'Generalised-Ex' model:

$$F(N) = \begin{cases} \lambda N(t) & N \le N_C \\ \left[\lambda + \frac{\lambda_1}{\alpha}\left(1 - \left(\frac{N(t)}{N_C}\right)^{\alpha}\right)\right] \cdot N(t) & N \ge N_C \end{cases} \qquad (2.6)$$

which extends the two-phase approach to tumour growth modelling and therefore results in a more general description of tumour growth kinetics.

Although more accurate, two-phase models involve extra computation for numerical optimisation as well as requiring a number of additional parameters to be known. Given the fact that most tumours are diagnosed and treated at the macroscopic stage (which is far beyond N_C), little can be gained from utilising two-phase models. Thus, the Gompertz growth function will be used hereafter to model the way in which untreated tumours develop and to estimate the characteristics of this development (including the doubling time of the tumour cell population).

However, we have not addressed so far the most important problem from the practical point of view, i.e. how chemotherapeutic treatment will perturb the uncontrolled growth of tumours. The simplest approach to model this perturbation is to incorporate a cell-loss function $L(t)$ into the differential equation (2.1) governing the process of tumour development. The following section gives details on general forms of the function $L(t)$, on the meaning of its components as well as on the issue of how the numerical values of these components can be estimated.

2.3. Empirical Models of Cell Loss

The exposure of a tumour cell population to an anti-cancer drug results in retardation of the tumour growth rate and ideally leads to reduction of the number of cells the tumour contains. Mathematically the effect of anti-cancer drugs can be expressed in the form of a cell-loss function $L(t)$ which modifies the differential equation of tumour growth in the following way:

$$\dot{N}(t) = F(N(t)) - L(t)$$
$$N(0) = N_0 \tag{2.7}$$

where $F(N(t))$ and $L(t)$ represent the general forms of a tumour growth and a cell loss function respectively.

The rate of cell loss $L(t)$ is proportional to the cell number $N(t)$ as well as to the drug concentration $\upsilon(t)$ at the tumour site (Martin and Teo, 1994). Hence, the cell-loss function may be represented as the following product

$$L(t) = L(\upsilon)N(t)$$

where the coefficient $L(\upsilon)$ denotes the proportion of cells killed by the anti-cancer drug at the concentration level υ. The two most common forms of this coefficient are (Murray, 1990)

- **linear**:

$$L(\upsilon) = k \cdot \upsilon \tag{2.8}$$

where k is a quantity representing the effectiveness of a given anti - cancer agent

- **saturated**:

$$L(\upsilon) = \frac{k_1 \cdot \upsilon}{k_2 + \upsilon} \tag{2.9}$$

where k_1 and k_2 are parameters to be estimated.

The saturated model of the cell-kill rate appears to suit better those situations where increasing drug resistance can be expected to decrease the cell loss caused by the anti-cancer drug (Swan, 1990). However, the current work focuses solely on treating tumours consisting of drug-sensitive cells only. Thus, the hypothesis of a linear relationship between the rate of cell loss and the drug concentration $\upsilon(t)$ is expected to be valid. This allows us to use hereafter the linear model of the cell-loss function.

The problem with using the equation (2.8) is that information on the magnitude of $\upsilon(t)$ is rarely available. As has been pointed out in Chapter 1, the process of drug circulation through the bloodstream from the point of entry to the tumour site is not well understood and therefore difficult to model. Thus it is conventional (in the absence of a better approximation) to assume that the concentration $\upsilon(t)$ is equivalent to the concentration of anti-cancer drug in the plasma, $c(t)$. In other words, the tumour will be subject to the action of whatever concentration of drug is present in the blood at a given time.

There are two advantages to this approach. Firstly, it is easier to model the concentration $c(t)$ in the blood in terms of the drug delivery function $D(t)$. Having assumed that the bloodstream

is a single compartment with some volume V, the concentration $c(t)$ can be described by the following differential equation (Collins and Dedrick, 1982)

$$\dot{c}(t) = \frac{D(t)}{V} - \delta \cdot c(t) \tag{2.10}$$

where δ is a drug elimination constant

Secondly, by focusing on the actual concentration of anti-cancer drugs in the bloodstream we are modelling chemotherapy in terms of what the tumour actually experiences irrespective of the delivery mode by which the treatment regimen is administered. The concentration level $c(t)$ of anti-cancer drugs in the plasma provides a target regimen for clinicians to aim at regardless of the mode of drug delivery.

The function $c(t)$ over the treatment interval $[T_0, T_{\text{final}}]$ will be hereafter referred to as a *treatment regimen*. A regimen may take the form of a closed-form expression for c in terms of t or, more realistically, can be specified as the following vector

$$\mathbf{c}(t) = (C_i) \text{ where } C_i = C(t_i) \text{ for times } T_0 = t_0 < t_1 < \ldots < t_n = T_{\text{final}}$$

Without loss of generality we may assume that the time instants t_i are equally spaced by an amount Δt – otherwise one can introduce surplus time instants with appropriate drug doses:

$$t_i = i \cdot \Delta t \text{ for } i = \overline{0,n} \text{ and } c(t) \text{ is constant on the interval } [t_i, t_{i+1})$$

Since concentration levels are difficult to measure *in vivo* and there is a limit to the amount of measurements which can be tolerated, little can be gained from using a continuous function $c(t)$. In practice it will be approximated by a sequence of values C_i. Therefore throughout this thesis a discrete form $\mathbf{c}(t) = (C_i)$ of anti-cancer drug concentration in the blood will represent a particular treatment regimen.

Together with an appropriate tumour growth function $F(N(t))$, a discrete representation of a linear cell-loss function $L(t)$ yields the following differential equation governing the response of the tumour to chemotherapy treatment.

$$\dot{N}(t) = F(N(t)) - \kappa\left(\sum_{i=1}^{n} C_i \{H(t-t_{i-1}) - H(t-t_i)\}\right) N(t),\ N(0) = N_0 \tag{2.11}$$

where κ is the efficacy of the anti-cancer drug and
$H(t)$ is the Heaviside step function

Consequently, the problem of achieving the treatment objectives of cancer chemotherapy specified in Chapter 1 is equivalent to finding the drug dosage regimen $c(t)$ which controls the tumour kinetics described by the equation (2.11) in the optimal way. By optimal it is meant the way that allows achieving the best treatment results subject to satisfaction of multiple constraints imposed on cancer chemotherapy. Problems of this sort are dealt with in special branches of mathematics – Mathematical Programming and particularly in Optimal Control

theory, application of which in the context of cancer chemotherapy comprise the topic of the next chapter.

2.4. Discussion

In this chapter a mathematical description has been given of two fundamental processes defining the tumour kinetics during chemotherapeutic treatment, viz. tumour growth and tumour response to chemotherapy. Empirical models of these processes have been presented. Also, it has been shown that one way to improve the results of chemotherapeutic treatment is to utilise these empirical models in order to determine the optimal regimen of drug administration. The next chapter will give the particulars on how such a regimen can be found; however, a few things remain to be said here on mathematical modelling of tumour kinetics.

First of all, the mathematical models discussed above assume uniformity of the tumour cell population, i.e. all cancer cells are viewed as having the same properties. In reality, however, tumours are by and large heterogeneous with respect to almost every measurable biological property (Wheldon, 1988). This cellular heterogeneity may include the effect of clonal resistance, which is caused by an appearance of clones of atypical cancer cells inherently resistant to the effects of a given anti-cancer agent. The phenomenon of drug resistance often leads to treatment failure – therefore, it is desirable in practice to undertake precautions in order to minimise the risk of drug resistance occurrence. Mathematical models developed by Goldie and Coldman (1979, 1982), Birkhead and Gregory (1984), Usher and Henderson (1997), Agur *et al* (1988), Panetta and Adam (1995) and Webb (1992) may assist in this task. Furthermore, there have been attempts made to incorporate the theory of non-linear mechanics for modelling cellular properties. The understanding of the mechanical properties of cells is important because these properties determine the pattern of cell distribution and establishment of distant metastases (Chaplain and Sleeman, 1993). Also, mechanical pressure within solid tumours can be viewed as a drug-repelling mechanism, which is responsible for stopping the anti-cancer agent from reaching the cells inside tumours and therefore leads to the occurrence of drug resistance.

Secondly, even without clonal or 'mechanical' resistance of the tumour cells, the effect of the anti-cancer drug varies during the treatment period. This variation is known as the phenomenon of *kinetic resistance*, which simply means that in order to maintain tumour cell depopulation the drug concentration has to become larger as the tumour shrinks (Wheldon, 1988). To defy kinetic resistance Norton and Simon (1986) suggested a chemotherapy regimen, known as a late-intensity schedule, based on mathematical models of tumour kinetics. This was another area where mathematical results have had a real impact on clinical therapy.

Finally, the success of chemotherapy to a great degree depends on drug dosimetry or pharmacokinetics. In contrast to radiotherapy where the dose delivered to a particular region can usually be calculated reasonably accurately, there is no corresponding unambiguous

meaning of the chemotherapeutic dose. Determining the drug concentration as a function of time for each tissue (not only for the tumour) is the subject of pharmacokinetics wherein mathematical models often play a useful role. Pharmacokinetic (PK) models describe the transition of drug through the body and are based on kinetic parameters which govern this process. However, the PK parameters are often found to significantly differ from one patient to another, which undermines the applicability of pharmacokinetic models for treating real cancer patients (Wheldon, 1988; Panetta and Adam, 1995).

In general, the usefulness of mathematical modelling in the cancer related domain strongly depends on both biological realism of the assumptions made to formulate the models and on knowledge of the models' parameters. The latter factor particularly makes it difficult to obtain authentic models of the tumour kinetics during treatment, especially in cases when some kind of resistance is present or personalised pharmacokinetic properties need to be taken into account. Although the issues concerning drug resistance and pharmacokinetics lie beyond the scope of this thesis and therefore will not be pursued further, they ought to be mentioned in the present context. The reason for this is that these issues impose serious limitations on reliability of mathematical modelling in cancer treatment, and these limitations must be borne in mind.

CHAPTER 3

MATHEMATICAL OPTIMISATION OF CHEMOTHERAPY

It is a difficult task to determine an effective treatment regimen which optimises the beneficial effects of chemotherapy while limiting the adverse effects of anti-cancer drugs on the well being of the cancer patient. This difficulty is mainly caused by the fact that successful chemotherapy to a great degree depends on the type of chemotherapeutic drugs chosen, on the dosages and the timing of drug administration, and on the mode of administration. With so many variables, the amount of experimentation involved in determining the best chemotherapeutic regimen from a purely empirical approach is prohibitive in both cost and time. It is desirable therefore to make use of the techniques of mathematical optimisation in order to help clinicians to determine optimal treatment regimens.

In general, optimisation is the act of obtaining the best results under given circumstances. A huge variety of problems fall into the optimisation category, which makes it infeasible to find a universal method for solving all optimisation problems efficiently. Hence a plethora of optimum seeking methods, also known as mathematical programming techniques, have been developed for solving different types of optimisation problems.

Mathematically, a general optimisation problem can be stated as follows: Optimise the objective functional

$$J(\mathbf{c}) \quad \text{over the vector } \mathbf{c} = (c_1(t), c_2(t), \ldots, c_n(t))^{\mathrm{T}} \in R^n \tag{3.1}$$

subject to the constraints

$$\begin{cases} g_j(\mathbf{c}) = 0 & j = 1,2,\ldots,E \\ g_j(\mathbf{c}) \geq 0 & j = E+1,\ldots,S \end{cases} \tag{3.2}$$

where $\mathbf{c}$ is an n-dimensional vector of decision variables $c_i(t), i = \overline{1,n}$; E and S are the number of equality constraints and the total number of constraints respectively. The decision variables, often referred to as control factors, represent the quantities which can be controlled within the limits specified by (3.2) to attain the minimum or the maximum value of $J(\mathbf{c})$.

The objective functional $J(\mathbf{c})$ represents a criterion for comparing different control vectors of decision variables and for selecting the best one. The choice of the objective functional is governed by the nature of the optimisation problem given. In some situations, there may be more than one criterion to be satisfied simultaneously. With multiple objectives there often arises a possibility of conflict, which is usually handled either by using the concept of Pareto optimality or by constructing a composite objective functional as a linear combination of the conflicting objectives. Pareto optimality of a particular solution means that the solution satisfies the condition – any solution different from the optimum results either in no change to each optimisation objective, or it causes at least one objective to deteriorate in value. Multi-criteria optimisation problems will be discussed in more detail later in this chapter.

The general frame of an optimisation (or a mathematical programming) problem expressed in (3.1)-(3.2) can be used to formulate the problem of finding the best chemotherapeutic regimen: Given a performance criterion (any quantitative measure of the cancer treatment objectives), find the drugs, dosages and times of drug administration which will achieve the optimal value of the performance criterion without violating the treatment constraints. These constraints include limitations on drug dosages, on the tumour size and on toxic side effects produced by multiple drug administration.

Thus, the task formulated above belongs to a broad spectrum of optimisation problems. In this chapter a description of conventional mathematical methods applicable to chemotherapy optimisation will be provided, which are based on Mathematical Programming and on Optimal Control theory in particular. Before giving this description, a general classification of optimisation problems and an overview of optimisation techniques to tackle them will be given in Section 3.1. Also, it will be shown how optimal control problems fit into the introduced classification and why problems of the optimal control category are so important as far as cancer chemotherapy optimisation is concerned. In Section 3.2 we will give the particulars on how the optimal control approach has been utilised to help oncologists to improve chemotherapeutic treatment. Section 3.3 focuses on applications of other mathematical programming methods to cancer chemotherapy. And finally, in Section 3.4 a short overview of all conventional methods of optimisation will be given with an explanation of why their usage in the cancer chemotherapy context has not become widespread.

3.1. Classification of Optimisation Problems

Optimisation problems can be categorised in several ways (Rao, 1978) as described below.

i. *Classification based on the presence of constraints.* There are constrained and unconstrained optimisation problems.

ii. *Classification based on the nature of decision (control) variables.* The control variables may be static $(C_i \mid i = \overline{1,n})$ or may be continuous functions of some parameter $(c_i(t) \mid i = \overline{1,n})$.

iii. *Classification based on the physical structure of the problem.* Depending on the physical structure, optimisation problems are grouped into optimal control and nonoptimal control categories. An optimal control problem is usually characterised by two types of variables, viz. the control $\mathbf{c}$ and the state $\mathbf{x}$ vectors. The control vector governs the evolution of the system from one stage to the next, and the state vector specifies the system state at any stage. Explicitly, the optimal control problem is a mathematical programming problem involving a number of stages (specified by the state vector), where each stage evolves from the previous stage in a predescribed (by the control vector) manner.

The state vector can be discrete (when the number of stages is finite) or continuous. An optimal control problem with a continuous state vector may be formulated as follows: Optimise an objective functional of the form

$$J(\mathbf{c}) = \phi_0(\mathbf{x}(T \mid \mathbf{c})) + \int_0^T L_0(t, \mathbf{x}(t \mid \mathbf{c}), \mathbf{c})dt \tag{3.3}$$

subject to :

$$\dot{\mathbf{x}}(t) = f(t, \mathbf{x}, \mathbf{c})$$
$$\mathbf{x}(0) = \mathbf{x}_0(\mathbf{c}(0)) \tag{3.4}$$

where $\mathbf{x} = [x_1(t), x_2(t), \ldots, x_m(t)]^{\mathrm{T}} \in R^m$ is the state vector described by the vector - valued function $f = [f_1, f_2, \ldots, f_m]^{\mathrm{T}} \in R^m$;

$\mathbf{c} = [c_1(t), c_2(t), \ldots, c_n(t)]^{\mathrm{T}} \in R^n$ is the control vector;

ϕ_0 is a scalar - valued function of the terminal condition.

and the set of equality and inequality constraints:

$$g_i(\mathbf{c}) = \phi_i(\mathbf{x}(\tau_i \mid c)) + \int_0^{\tau_i} L_i(t, \mathbf{x}(t \mid \mathbf{c}), \mathbf{c})dt = 0 \quad \text{for } i = 1,2,\ldots,E \tag{3.5}$$

$$g_i(\mathbf{c}) = \phi_i(\mathbf{x}(\tau_i \mid \mathbf{c})) + \int_0^{\tau_i} L_i(t, \mathbf{x}(t \mid \mathbf{c}), \mathbf{c})dt \geq 0 \quad \text{for } i = E+1,\ldots,S \tag{3.6}$$

The differential equation (3.4) is defined on the interval $[0,T]$. For some problems the terminal time T is fixed, for others T is itself a variable parameter of the optimal control problem. The state vector $\mathbf{x}(t)$ uniquely specifies the state of the system, i.e. a set of quantities that can be measured but not directly altered. In cancer chemotherapy, for example, the state is usually related to the tumour volume. Combined together all state vectors form the *solution* (or state) *space*. Although directly uncontrollable, the state vector submits to indirect control through (3.4) by the control vector $\mathbf{c}(t)$. Usually control vectors are restricted to lie in some subset of R^n, and the elements of this subset are called *admissible* (or feasible) control vectors.

The solution of the equation (3.4) corresponding to a particular control $\mathbf{c}(t)$ is called a trajectory $\mathbf{x}(\bullet \mid \mathbf{c})$. The control $\mathbf{c}(t)$ is used to prevent, or at least to postpone, trajectories of the dynamical system from ending up in an undesirable region of the solution space. The efficiency of a control in implementing this task is usually measured by the objective functional $J(\mathbf{c})$. The equality and inequality constraints expressed in their canonical forms (3.5) and (3.6) are dependent upon the state and control vectors evaluated at single points in time τ_i, which are called characteristic times. In standard optimal control theory, there is at most one such time associated with each constraint. However, Martin and Teo (1994) developed an approach to deal with multiple characteristic time constraints.

iv. *Classification based on the nature of equations involved.* This classification is based upon the nature of expressions for the objective functional (3.1) and the constraints (3.2). Depending on these expressions, optimisation problems are sorted into linear, nonlinear, geometric, and quadratic programming categories. This classification is extremely useful from the computational point of view since there exist specific methods developed solely for efficient solution of a particular category of problems. The category of the optimisation problem will, in many cases, dictate the type of solution procedures to be adopted in solving the problem.

If the objective functional and all constraints in Equations (3.1)-(3.2) are linear functions of the control variables $(C_i \mid i = \overline{1,n})$, then the mathematical programming problem is called a *linear programming* (LP) problem and often is stated in the following standard form.

$$\text{Find } \mathbf{c} = (C_1, C_2, \ldots, C_n)^{\mathrm{T}} \text{ which optimises } J(\mathbf{c}) = \sum_{i=1}^{n} q_i C_i \tag{3.7}$$

subject to the constraints:

$$\begin{cases} g_j(\mathbf{c}) = b_j - \sum_{i=1}^{n} a_{ji} C_i = 0 & j = 1, \ldots, E \\ g_j(\mathbf{c}) = C_j \geq 0 & j = E+1, \ldots, E+n \end{cases} \tag{3.8}$$

where q_i, a_{ji} and b_j are constants.

If any of the expressions representing the objective of optimisation (3.1) or the optimisation constraints (3.2) are nonlinear, the problem is referred to as a *nonlinear programming* (NLP) problem. The NLP category is the most general and includes as special cases geometric and quadratic programming problems. A *geometric programming* problem is the one in which the objective functional and the constraints are expressed as the sums of the form:

$$\sum_{i=1}^{N} a_i \left[\prod_{j=1}^{n} C_j^{p_{ij}} \right], \quad C_j > 0 \quad \text{for all } j = \overline{1,n}$$

where a_i is a constant, $a_i > 0$;

N denotes the number of terms in the objective functional $J(\mathbf{c})$ or in the jth constraint $g_j(\mathbf{c})$.

A *quadratic programming* problem is an NLP problem with a quadratic objective functional

$$J(\mathbf{c}) = q_0 + \sum_{i=1}^{n} q_i C_i + \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} C_i C_j$$

where q_0, q_i, Q_{ij} are constants.

and linear constraints

$$\sum_{i=1}^{n} a_{ji} C_i = b_j \qquad j = 1,2,\ldots,S-n$$

$$C_i \geq 0 \qquad i = 1,\ldots,n$$

where a_{ji} and b_j are constants.

v. *Classification based on permissible values of the control vector.* If some or all of the control variables $c_i(t), i = \overline{1,n}$ of an optimisation problem are restricted to only integer (or discrete) values, then the problem is called an *integer programming* problem.

vi. *Classification based on deterministic nature of variables involved.* This is the last criterion according to which optimisation problems are classified either as *deterministic* or as *stochastic programming* problems. A distinctive feature of stochastic optimisation problems is that some of the problems' parameters are probabilistic (known with a certain probability or characterised by a probability distribution).

Various techniques are available for the solution of optimisation problems classified in this section. For example, to find unconstrained minima or maxima of a differentiable objective functional, the classical methods of differential calculus can be used, which are based on determining zero points of the gradients $\frac{\partial J(\mathbf{c})}{\partial c_i(t)} = 0, \; i = \overline{1,n}$. When the objective function is expressed as an integral, the methods of calculus of variations can be utilised (Burghes and Graham, 1980). If an optimisation problem deals with controlling a dynamic system, which is described by the state equation (3.4), then the methods of Optimal Control are well suited to solve the problem.

The techniques of linear, nonlinear, geometric, integer or stochastic programming can be used to solve the particular class of problems indicated by the name of the technique. These are all numerical methods whereby an approximate solution is sought by proceeding in an iterative manner from a starting solution. In addition, for optimisation of multi-stage decision problems the technique of dynamic programming may be adopted. When applicable, the dynamic programming technique represents or decomposes a multi-stage decision problem as a sequence of single stage decisions, each of which is easier to solve than the original problem.

Unsurprisingly, the rich arsenal of mathematical programming techniques provides a number of optimisation tools for solution of the cancer chemotherapy optimisation problem. The next two sections will particularise the application of Optimal Control theory, LP and NLP methods to optimisation of cancer chemotherapy. We will start with the Optimal Control techniques.

3.2. Optimal Control of Cancer Chemotherapy

As we said in the previous section, the aim of an optimal control problem is to find a control vector $(c_i(t) \mid i = \overline{1,n})$ that optimises the objective functional (3.3) subject to the state equation (3.4), which governs the evolution of the system to be optimised, and the constraints (3.5)-(3.6), which limit the values of some or all variables involved. A theoretical approach to finding such a control vector is based on the method of calculus of variations and on the Pontryagin Maximum principle, which necessitate the following definitions to be made.

Definition 3.1. The *augmented objective functional* J^* is an objective of an optimal control problem which incorporates (via Lagrange multipliers $\mathbf{p}^{\mathrm{T}} = [(p_1, p_2, \ldots, p_S)]^{\mathrm{T}}$) the status of the system, viz.

$$J^* = \phi_0(\mathbf{x}(T \mid \mathbf{c})) + \int_0^T \left\{ L_0(t, \mathbf{x}(t \mid \mathbf{c}), \mathbf{c}) + [\mathbf{p}(t)]^{\mathrm{T}} \left[f(t, \mathbf{x}(t), \mathbf{c}(t)) - \dot{\mathbf{x}}(t) \right] \right\} dt \qquad (3.9)$$

Definition 3.2. The *Hamiltonian* is the function of the form

$$H(t, \mathbf{x}, \mathbf{c}, \mathbf{p}) = L_0(t, \mathbf{x}, \mathbf{c}) - [\mathbf{p}(t)]^{\mathrm{T}} \cdot f(t, \mathbf{x}, \mathbf{c}) \qquad (3.10)$$

Applying Euler equations (Burghes and Graham, 1980) to the augmented objective functional J^*, we obtain the system of differential equations for finding the optimal control and the trajectory corresponding to it:

$$\begin{cases} [\dot{\mathbf{p}}(t)]^{\mathrm{T}} = -\dfrac{\partial H}{\partial \mathbf{x}} \\ \dfrac{\partial H}{\partial \mathbf{c}} = 0 \end{cases} \qquad (3.11)$$

with boundary conditions $[\mathbf{p}(T)]^{\mathrm{T}} = \dfrac{\partial \phi_0}{\partial \mathbf{x}}$.

If the control vector which solves the equation $\frac{\partial H}{\partial \mathbf{c}} = 0$ is not an admissible control (as often is the case in the theory of bounded control), then the solution to the optimal control problem (3.3)-(3.6) is specified by the following proposition.

Proposition 3.1. **(Pontryagin's Maximum Principle)** The control $\mathbf{c}(t)$ that satisfies the constraints (3.5) and (3.6), optimising at the same time the value of the Hamiltonian function (3.10), will be the solution to the optimal control problem (3.3) – (3.6).

Let us now discuss how the theory of Optimal Control has been used in the context of cancer chemotherapy optimisation. As was stated in Section 2.3, the most general form of the tumour kinetics may be represented as follows:

$$\dot{N}(t) = F(N) - L(t, N, \mathbf{c}), \quad N(0) = N_0 \tag{3.12}$$

where $F(N)$ and $L(t, N, \mathbf{c})$ are the tumour growth and the cell loss function respectively.

The differential equation (3.12) plays the role of the state equation. It governs tumour development during chemotherapeutic treatment and defines the solution space of treatment outcomes. Each trajectory $N(t \mid \mathbf{c}(t))$ of the solution space represents how the tumour reacts to the treatment regimen expressed as a control vector $\mathbf{c}(t)$.

The effect of the control vector $\mathbf{c}(t)$ is measured by a performance index associated in one way or another with the objective of optimisation. It is known that anti-cancer drugs affect not only cancerous but normal cells as well. In order to restrain the toxic side effects of anti-cancer drugs two types of optimal control models have been used. The first type directly incorporates a measure of how the normal-cell population is affected by drugs into the objective functional and therefore does not explicitly involve the toxicity constraints. The second type of models constrains normal-cell toxicity by imposing constraints on drug schedules that can be used. We will examine these model types in turn.

3.2.1. Unconstrained Optimal Control of Cancer Chemotherapy

Swan and Vincent (1977) pioneered this approach to optimisation of cancer chemotherapy by attempting to minimise the cumulative drug toxicity while guiding the tumour size to a specified level at the end of the treatment period. They used the following performance index

$$J(\mathbf{c}) = \int_0^T c(t)dt \tag{3.13}$$

which needed to be minimised. One obvious limitation of this model is that it is not clear how the tumour burden at the end of the therapy can be specified ahead of time.

An alternative to Swan and Vincent's model is to simultaneously minimise the size of the tumour and the amount of drug administered. Hence, Swan (1990) proposed to modify the performance index (3.13) as follows:

$$J(\mathbf{c}) = \int_0^T \left[\left(N(t) - N_d \right)^2 + \rho \cdot c^2(t) \right] dt \tag{3.14}$$

where N_d is the smallest detectable tumour size.

Since clinicians cannot measure the tumour size below N_d, they believe that one should treat the tumour only to the level of N_d. (NOTE. There is a significant difference between the smallest detectable tumour size N_d and the tumour size N_{cure} which has to be attained to prevent the tumour from regrowing. If the former has the order of magnitude $\sim 10^9$ cells, the latter is usually $\leq 10^3$ cells.) Since it is impossible to observe what is happening to the tumour smaller than N_d, the size N_{cure} is often substituted by N_d and the objective of practical optimisation of cancer treatment is to make the difference $(N(t) - N_d)$ as small as possible. The second term in (3.14) weighs the cost of control and represents a penalty for using too much drug. By increasing the positive constant ρ, this penalty can be made arbitrarily large. Then, the problem of cancer chemotherapy optimisation can be stated as follows: Given the state equation (3.12) find the optimal control $\mathbf{c}_{opt}(t)$ which minimises the objective (3.14).

This problem has been solved for the logistic model of tumour kinetics with the result that the optimal control $\mathbf{c}_{opt}(t)$ must be an increasing function whose rate of increase slows down rapidly as time progresses (Swan, 1990). Furthermore, it has been found that for the case of two identical tumours of different initial sizes at diagnosis, the optimal control does not require an increase in initial drug dosage in order to deal with the larger tumour. This finding is in accordance with clinical experience.

A totally different approach to incorporating the normal-cell toxicity into the optimisation objective was developed by Zietz and Nicolini (1979). Their idea was based upon the concept of multiple criteria optimisation. The authors used a multi-criteria objective functional to maximise the normal-cell population $N_1(T)$ and to minimise the tumour cell population $N_2(T)$ by the end of the treatment period $[0,T]$. In order to describe the kinetics of these populations the Gompertz model was adopted, which resulted in the following performance indexes:

$$J_1(\mathbf{c}) = \int_0^T N_1(t)\left[-\lambda_1 \cdot \ln\left(\frac{N_1(t)}{\Theta_1}\right) - \kappa_1 \cdot c(t)\right] dt \tag{3.15}$$

$$J_2(\mathbf{c}) = \int_0^T N_2(t)\left[-\lambda_2 \cdot \ln\left(\frac{N_2(t)}{\Theta_2}\right) - \kappa_2 \cdot c(t)\right] dt \tag{3.16}$$

The control variable in (3.15) and (3.16) is the drug concentration $c(t)$, whose mission was to maximise the two-objective functional $J(\mathbf{c}) = J_1(\mathbf{c}) - J_2(\mathbf{c})$. Although Zietz and Nicolini (1979) numerically solved this optimal control problem, all of the controls they found were of the open-loop type and consisted either of no therapy, continuous therapy, or continuous therapy followed by rest. The disadvantage of the latter approach is that it is often difficult to establish the cells of which body organ or compartment mostly contribute to the normal-cell population.

Therefore, another type of optimal control model has been developed which does not explicitly incorporate the measure of the normal-cell population into the objective functional. Instead, the optimisation models of the second type impose certain constraints on the state and the control vectors, thereby limiting the number of feasible solutions to the optimisation problem under investigation.

3.2.2. Constrained Optimal Control of Cancer Chemotherapy

This approach has been favoured by a number of researchers, particularly by Murray (1990a,b) and Martin *et al* (1990, 1992). Although Murray (1990a,b) still considers the normal-cell population $N_1(t)$ as one of the toxicity limits, his objective functional has the form:

$$\text{Minimise } J(\mathbf{c}) = \int_0^T f(t, N_2(t))dt \tag{3.17}$$

where $f(t, N_2(t))$ is a positive valued weighting function of the tumour cell population $N_2(t)$, and $[0,T]$ is a fixed treatment interval.

The function $f(t, N_2(t))$ can be used to penalise undesirable values of the state variable $N_2(t)$ by increasing their contribution to the objective functional. The optimisation objective (3.17) in Murray's studies had to be minimised subject to the following state equations:

$$\begin{cases} \dot{N}_1(t) = N_1(t)\left[-\lambda_1 \cdot \ln\left(\dfrac{N_1(t)}{\Theta_1}\right) - \kappa_1 \cdot c(t)\right], & N_1(0) = N_1^0 \\ \dot{N}_2(t) = N_2(t)\left[-\lambda_2 \cdot \ln\left(\dfrac{N_2(t)}{\Theta_2}\right) - \kappa_2 \cdot c(t)\right], & N_2(0) = N_2^0 \end{cases} \tag{3.18}$$

and the toxicity constraints:

$$
\begin{aligned}
& c(t) \in [0, C_{\max}] \\
& \int_0^T c(t)\, dt \le C_{\text{cum}} \\
& N_1(t) \ge N_1^{\min} \qquad \text{for all } t \in [0,T]
\end{aligned}
\tag{3.19}
$$

where $C_{\max}$ and C_{cum} are the maximum and the cumulative drug concentration limits introduced in Section 1.3; κ_1 and κ_2 are the cell-kill rates for the normal and the tumour cells respectively.

The optimisation problem (3.17)-(3.19) has been solved in Murray (1990b) for a number of test cases. The optimal control essentially turned out to be one of using the bolus application of drug to drive the tumour cell population down and then using continuous infusion. However, for a certain time before, during or after drug administration nothing should be done, i.e. zero control is used.

The timing of the nonzero portion is determined by the integrand of the objective functional (3.17). If $f(t, N_2(t)) = \dot{N}_2(t)$, that is if the aim of treatment is to minimise the tumour cell population at the final time, then one should leave the nonzero portion of control as late as possible. If the integrand is $\ln[N_2(t)]$ then the reverse is true; drug administration should commence immediately from the start of the treatment period. And in the last test case, when $f(t, N_2(t)) = e^{\rho t} \ln[N_2(t)]$, where ρ is a scaling factor, one gets the range of placement of the nonzero portion of control between the above two extremes:

$$
c_{\text{opt}}(t) = \begin{cases} 0, & 0 \le t < t_1 \\ C_{\max}, & t_1 \le t < t_2 \\ 0, & t_2 \le t < t_3 \\ C_s, & t_3 \le t < t_4 \\ 0, & t_4 \le t \le T \end{cases}
\tag{3.20}
$$

where C_S is a control that holds the normal population at the predescribed level.

The estimations of t_1, t_2, t_3, t_4 and their dependence on the type of the objective functional (3.17) are given in Murray (1990b).

Thus, the flexible structure of the optimal control $c_{\text{opt}}(t)$, which holds essentially regardless of the type of cell-loss and tumour growth functions, makes allowance for various clinical situations. When the main concern is to minimise the final number of tumour cells, the drug administration schedule is biased towards the end of treatment, i.e. it delays the time t_1. Conversely, in practical cases when oncologists try to avoid the development of new subpopulations (some of which might turn out to be drug resistant), treatment is commenced as

soon as possible. This brings t_1 closer to 0 and adjusts the other switching times t_2, t_3, t_4 accordingly.

Another optimal control model of the constraint-based type has been developed by Martin *et al* (1990). These authors entirely deviate from considering the normal-cell population and focus solely on the tumour expressed by the function $N(t)$ representing the number of cancerous cells. Furthermore, the authors investigate the situation where drug is administered according to a discrete dosage program in which there are n doses each of size C_i, $i = \overline{1,n}$, given at times $t_1, t_2, \ldots, t_n$.

Having adopted the Gompertz model of tumour growth and having assumed a linear relationship between drug concentration and cell loss, Martin *et al* (1990) formulated the following optimal control problem:

$$\text{Minimise } J(\mathbf{c}) = N(T) \quad \text{over the control vector } \mathbf{c} = (C_i \mid i = \overline{1,n}) \tag{3.21}$$

where T is a fixed length of the treatment interval.

subject to the state equations: -

$$\begin{cases} \dot{N}(t) = N(t)\left[\lambda \ln\left(\dfrac{\Theta}{N(t)}\right) - k(\upsilon(t) - \upsilon_{\text{th}})H(\upsilon(t) - \upsilon_{\text{th}})\right], \; N(0) = N_0 \\ \upsilon(t) = \displaystyle\sum_{i=1}^{n} C_i \exp[-\delta(t - t_i)]H(t - t_i) \end{cases} \tag{3.22}$$

where $H(t)$ is the Heaviside step function -

$$H(t - t_i) = \begin{cases} 0 & t < t_i \\ 1 & t \geq t_i \end{cases}$$

and three inequality constraints in their canonical form (3.6):

$$\begin{cases} g_1(\mathbf{c}) = \upsilon_{\max} - \upsilon(t) \geq 0 & \text{for all } t \in [0,T] \\ g_2(\mathbf{c}) = \upsilon_{\text{cum}} - \displaystyle\int_0^T \upsilon(t)dt \geq 0 & \\ g_3(\mathbf{c}) = N_{\max} - N(t) \geq 0 & \text{for all } t \in [0,T] \end{cases} \tag{3.23}$$

As may be seen from the equation (3.22), an exponential decay of drug concentration has been taken into account and characterised by the parameter δ. Moreover, it has been assumed that no tumour cells are killed if the drug concentration falls below a therapeutic drug threshold υ_{th}.

The inequalities (3.23) formulate the first three constraints of cancer chemotherapy introduced in Section 1.3, viz. maximum instantaneous concentration, maximum cumulative concentration and maximum tumour size. Since Martin *et al* (1990) deal only with single-drug treatments, the last constraint of risk reduction of toxic side effects induced by multiple drug administration has not been considered.

In the present work we will employ the Martin *et al* model as a basis for further investigations. The reason for choosing this particular model is threefold. Firstly, Martin *et al* (1990) adopt a discrete dosage program of drug administration, which makes it easier to apply the results of their optimisation in practice. Secondly, by concentrating exclusively on tumour cells, it is possible to substantially reduce the number of uncertain parameters characterising the normal cell population. The redundant parameters include such quantities as the minimum level and the asymptotic limit of the normal cell population, the growth rate of normal cells, and the kill rate of normal cells by anti-cancer drugs. Also, the model (3.21)-(3.23) is not concerned with the problem of what particular normal tissue is the most sensitive to application of a given drug and how to mathematically represent this sensitivity. It is apparent, however, that most of the above mentioned characteristics are implicitly included in the toxicity limits $\upsilon_{\max}, \upsilon_{\text{cum}}$ and $N_{\max}$, which specify the constraints (3.23). Nevertheless, the estimations of these limits are available from experimental studies, which spare us from the necessity to determine them analytically.

The last reason that supports the choice of the problem (3.21)-(3.23) as a basis for our optimal control model of cancer chemotherapy is that Martin *et al* (1990) developed a systematic approach to the solution of the optimal control problems of this type. Their approach is based on the production of certain gradients of the objective functional (3.21) and the constraints (3.23), and yields numerical solutions to a range of optimisation problems of cancer chemotherapy. These problems may have different objective functionals, utilise various models of tumour cell kinetics, and may even incorporate drug resistance (Martin and Teo, 1994).

An important practical implication of the optimal control found for the problem (3.21)-(3.23) is that the conventional method for treating cancer is not necessarily the best. Conventional therapy is defined as a treatment that attempts to keep the tumour burden to an absolute minimum at all times. The optimal regimen constructed by the model (3.21)-(3.23), on the other hand, differs from the conventional regimen in the respect that, though chemotherapy commences immediately, the bulk of the treatment is delayed as long as possible. This result coincides with the results of the Murray's studies (3.20) and is true regardless of the effectiveness of drugs used, provided that intermediate tumour size can be controlled.

However, two possible limitations of the Martin's *et al* model should be kept in mind. The first involves drug resistance, which may severely undermine the optimality of the late-stage chemotherapy. It has been shown by Goldie and Coldman (1979) that the probability of having drug resistant cell populations increases with increasing tumour size. Thus, under the optimal control regimen developed by Martin *et al* the tumour is more likely to acquire drug resistance compared with the conventional therapy. Secondly, since the optimal control problem was solved numerically using a fixed set of parameters, there is the risk that the conclusions about the optimal regimen are parameter dependent. However, detailed information on such parameters as the precise kill rates of drugs, the tumour growth rate, or the probability of drug

resistance occurrence is not always available. Information paucity becomes especially apparent when one attempts to optimise multi-drug chemotherapy regimens wherein the complexity of drug-drug interaction and highly variable patient tolerance leaves little chance for quantitatively characterising toxicity boundaries. Let us now examine what other methods of Mathematical Programming may offer to counterbalance the latter difficulty.

3.3. Mathematical Programming of Cancer Chemotherapy

As has already been pointed out, a distinctive feature of optimal control problems is that they deal with the processes governed and controlled by the state equation. Since chemotherapeutic treatment of cancer involves controlling the tumour, which is governed by the equation describing tumour cell kinetics, it is not surprising that the methods of optimal control are very suited for optimisation of cancer chemotherapy.

However optimal control methods are not the only techniques of Mathematical Programming capable of dealing with this problem. Under certain assumptions, it is possible to generalise the basis model of cancer chemotherapy optimisation (3.21)-(3.23) and present it as a conventional optimisation problem (3.1)-(3.2). In order to do this we need to incorporate the state equation (3.22) into one of the constraints (3.23). Then, other mathematical programming methods may become pertinent, and even more appropriate. In this section we will show how the approaches based on Linear and Nonlinear Programming can be applied to determine optimal regimens of anti-cancer drug administration.

3.3.1. Linear Programming

Without loss of generality we may assume that for single-agent chemotherapy the control vector $\mathbf{c}(t)$ representing the concentration of drug in the blood stream is a step function over n equal time intervals (as was suggested in Section 2.3)

$c(t) = (C_i, i = \overline{1,n})$, C_i are the constant drug concentration levels on the intervals $[t_{i-1}, t_i)$, where $T_0 = t_0 < t_1 < \ldots < t_n = T_{\text{final}}$.

By means of this approach for modelling drug concentration we ignore drug delivery and concentration decay mechanisms. Although it is apparent that drug concentration decay does occur due to drug absorption, metabolism and elimination, the length of the time intervals $[t_{i-1}, t_i)$ can be made arbitrary small. Then, the change in the drug concentration level during each time interval becomes negligible and can be well approximated by a constant value C_i.

Consider now the optimisation problem with the objective to minimise the final tumour size by the end of treatment subject to the constraints (3.23) and under the assumption of the

Gompertzian model of tumour growth. The Gompertzian model provides the following tumour kinetics

$$\dot{N}(t) = \left[\lambda \ln\left(\frac{\Theta}{N(t)} \right) - \kappa c(t) \right] N(t), \; N(0) = N_0 \tag{3.24}$$

Having assumed that $c(t)$ is a step function

$$c(t) = \sum_{i=1}^{n} C_i \{ H(t - t_{i-1}) - H(t - t_i) \} \tag{3.25}$$

where $H(t)$ is the Heaviside step function,

and using the substitution $y(t) = \ln\left(\frac{\Theta}{N(t)} \right)$, we can simplify the equation (3.24) as

$$\dot{y}(t) + \lambda y(t) = \kappa c(t)$$

to yield the analytical solution

$$y(t_p) = e^{-\lambda t_p} \left\{ y_0 + \frac{\kappa}{\lambda} \left(e^{\lambda \Delta t} - 1 \right) \sum_{i=1}^{p} C_i e^{\lambda t_{i-1}} \right\} \tag{3.26}$$

where $p = 0,1,2,\ldots,n.$

Making the following changes in notation:

$$K_p = \frac{\kappa}{\lambda} e^{-\lambda \cdot t_p} \left(e^{\lambda \cdot \Delta t} - 1 \right)$$

$$z_p = \frac{y(t_p) - e^{-\lambda \cdot t_p} \cdot y_0}{K_p} = \sum_{i=1}^{p} C_i \cdot e^{\lambda \cdot t_{i-1}} = \sum_{i=1}^{p} q_i \cdot C_i, \text{ and}$$

$$b_p = z_{\min}^{(p)} = \frac{y_{\min} - e^{-\lambda \cdot t_p} \cdot y_0}{K_p}$$

we are now in the position to formulate the LP problem in a similar fashion to the standard formulation (3.7)-(3.8):

$$\begin{aligned} \underset{\mathbf{c}=(C_i)}{\text{maximise}} \quad & J(\mathbf{c}) = z_n = \sum_{i=1}^{n} q_i \cdot C_i \qquad \text{subject to} \\ & 0 \le C_i \le C_{\max} \qquad i = \overline{1,n} \\ & \sum_{i=1}^{n} C_i \le C_{\text{cum}} \\ & z_p = \sum_{i=1}^{p} C_i \cdot e^{\lambda \cdot t_{i-1}} \ge b_p \qquad p = \overline{1,n}. \end{aligned} \tag{3.27}$$

Note that the third constraint only ensures that the tumour remains below maximum allowable size at the interval endpoints, whereas to satisfy the maximum tumour size constraints we need $y(t) \ge y_{\min} \quad \forall t \in [0,T]$. The following lemma proves the equivalence of these conditions.

Lemma 3.1. Given $y(t) = \ln\left(\frac{\Theta}{N(t)}\right)$ and a treatment regimen $c(t)$ in the form (3.25), let t_p and t_{p+1} be such that

$$y(t_p) \geq y_{\min} \text{ and } y(t_{p+1}) \geq y_{\min}.$$

Then, $y(s) \geq y_{\min}$ for $\forall s \in [t_p, t_{p+1}]$.

Proof: The Gompertz tumour growth model, with the given substitute $y(t) = \ln\left(\frac{\Theta}{N(t)}\right)$ yields the following analytical expression for the tumour under chemotherapeutic treatment $c(t)$.

$$y(t) = y_0 e^{-\lambda \cdot t} + \kappa \int_{T_0}^{t} e^{\lambda \cdot (\tau - t)} c(\tau) d\tau$$

Since the control function $c(t)$ is a step-function on the interval $[T_0, T_{\text{final}}]$, we may write

$$y(s) \cdot e^{\lambda \cdot s} - y_0 = \int_{T_0}^{s} \kappa \sum_{i=1}^{p} e^{\lambda \cdot \tau} C_i \cdot \{H(\tau - t_{i-1}) - H(\tau - t_i)\} d\tau$$

$$= \int_{T_0}^{t_p} \kappa \sum_{i=1}^{p} e^{\lambda \cdot \tau} C_i \cdot \{H(\tau - t_{i-1}) - H(\tau - t_i)\} d\tau + \int_{t_p}^{s} \kappa \sum_{i=1}^{p} e^{\lambda \cdot \tau} C_i \cdot \{H(\tau - t_{i-1}) - H(\tau - t_i)\} d\tau$$

$$= \frac{\kappa\left(e^{\lambda \cdot \Delta t} - 1\right)}{\lambda} \sum_{i=1}^{p} C_i \cdot e^{\lambda \cdot t_{i-1}} \cdot + \int_{t_p}^{s} \kappa \cdot C_{p+1} e^{\lambda \cdot \tau} d\tau.$$

where $s \in [t_p, t_{p+1}]$. After simplification the last expression becomes

$$y(s)e^{\lambda s} - y_0 = \frac{\kappa}{\lambda}(e^{\lambda \Delta t} - 1)\sum_{i=1}^{p} C_i e^{\lambda \cdot t_{i-1}} + \frac{C_{p+1}\kappa}{\lambda}\left(e^{\lambda s} - e^{\lambda t_p}\right)$$

$$y(s) = e^{-\lambda s}[y_0 + \frac{\kappa}{\lambda}(e^{\lambda \Delta t} - 1)\sum_{i=1}^{p} C_i e^{\lambda t_{i-1}}] + \frac{C_{p+1}\kappa}{\lambda}\left(1 - e^{-\lambda(s - t_p)}\right)$$

$$= e^{-\lambda(s-t_p)} y(t_p) + \frac{C_{p+1}\kappa}{\lambda}\left(1 - e^{-\lambda(s-t_p)}\right)$$

Hence, we obtain

$$y(s) = \left(y(t_p) - \frac{\kappa \cdot C_{p+1}}{\lambda}\right) \cdot e^{-\lambda(s-t_p)} + \frac{\kappa \cdot C_{p+1}}{\lambda}$$

Therefore, the function $y(s)$ is continuous on the interval $[t_p, t_{p+1}]$ and differentiable on the interval (t_p, t_{p+1}). It has no turning points because

$$y'(s) = -\lambda \cdot e^{-\lambda \cdot (s-t_p)}\left(y(t_p) - \frac{\kappa \cdot C_{p+1}}{\lambda}\right) \neq 0.$$

This means that minima occur at the endpoints and

$$y(s) \geq y_{\min} \Leftrightarrow N(s) \leq N_{\max} \quad \text{for } \forall s \in [t_p, t_{p+1}]$$

∎

The LP approach to solving the optimisation problem of cancer chemotherapy has both advantages and disadvantages. A very strong advantage of Linear Programming is that this technique will always find a global optimum, where one exists, i.e. when the feasibility region is nonempty and convex. This is done by exhaustive exploration of the vertices of the feasibility region, one of which will be, according to the theory of Linear Programming, the global optimum. Under the assumptions made, the optimal treatment strategy for the LP problem (3.27) will have the following form;

$$c_{\text{opt}}(t) = \begin{cases} 0 & 0 < t < \tau_1 \\ c_{\text{hold}}(t) & \tau_1 \le t < \tau_2 \\ C_{\max} & \tau_2 \le t \le T \end{cases} \tag{3.28}$$

where $c_{\text{hold}}(t)$ is the level of drug concentration which maintains the tumour at the maximum permissible size $N_{\max}$; τ_1 and τ_2 are characteristic times of the problem. τ_1 is the time when an untreated tumour with the initial size N_0 attains the size $N_{\max}$. τ_2 is the starting time of the highly intensive therapy which delivers the drug at the maximum permissible rate $C_{\max}$ and lasts until the end of treatment. An analytical expression for τ_1 can be obtained using the equation (3.24) and reads as

$$\tau_1 = -\frac{1}{\lambda}\ln\left[\log_{\frac{\Theta}{N_0}}\left(\frac{\Theta}{N_{\max}}\right)\right]$$

Also, if the function $c_{\text{hold}}(t)$ can be approximated by a constant C_{hold}, then the time τ_2 has an explicit form:

$$\tau_2 = \frac{C_{\max}T_{\text{final}} - C_{\text{hold}}\tau_1 - C_{\text{cum}}}{C_{\max} - C_{\text{hold}}}$$

The feasibility region of the LP problem (3.27) is nonempty iff the following is true

$$\begin{cases} c_{\text{hold}}(t) \le C_{\max} & \forall t \in [\tau_1, \tau_2] \\ \int_{\tau_1}^{T_{\text{final}}} c_{\text{hold}}(t)dt \le C_{\text{cum}} \end{cases} \tag{3.29}$$

Therefore, if an optimisation problem can be formulated as an LP problem and if the problem's constraints allow for feasible solutions, then the technique of Linear Programming will unfailingly find a global optimum. However, far from all optimisation problems can be formulated in this way. For example, if we change the optimisation objective, modify the model of tumour kinetics or add additional nonlinear constraints (real life scenarios often necessitate at least some of these), then the optimisation problem of cancer chemotherapy becomes intractable by the linear programming method. An alternative to the LP optimisation technique is Nonlinear Programming, a discussion on which is now provided.

3.3.2. Nonlinear Programming

In many optimisation problems the objective of optimisation and/or the constraints cannot be written explicitly in terms of the control variables. In such cases, one has to resort to the nonlinear programming methods of optimisation for numerical solution of the problem.

The basic philosophy of these methods is to produce a sequence of improved approximations to the optimum according to the following scheme.

i. Start with an initial trial point $\mathbf{C}_1$.

ii. Find a suitable direction $\mathbf{S}_i$ ($i = 1$ to start with) which points towards the optimum.

iii. Find an appropriate step length ζ_i^* for movement along the direction $\mathbf{S}_i$.

iv. Obtain the new approximation $\mathbf{C}_{i+1}$ as

$$\mathbf{C}_{i+1} = \mathbf{C}_i + \zeta_i^* \mathbf{S}_i \tag{3.30}$$

v. Test whether $\mathbf{C}_{i+1}$ is optimum. If $\mathbf{C}_{i+1}$ is optimum stop the procedure. Otherwise, set $i = i + 1$ and repeat step ii onwards.

The iterative procedure (3.30) is valid for both constrained and unconstrained optimisation problems, and the efficiency of this procedure depends on the appropriate choices of ζ_i^* and $\mathbf{S}_i$. In order to find an optimal step length ζ_i^* one-dimensional optimisation methods may be utilised. This is because the problem of finding ζ_i^* boils down to finding the value $\zeta_i = \zeta_i^*$, which optimises $J(\mathbf{C}_{i+1}) = J(\mathbf{C}_i + \zeta_i \mathbf{S}_i) = J(\zeta_i)$ for fixed values of $\mathbf{C}_i$ and $\mathbf{S}_i$. A thorough classification and description of one-dimensional methods is given in (Rao, 1978). Here, we will focus on the other aspect of the iteration procedure (3.30) concerned with finding the right direction $\mathbf{S}_i$, along which to seek the optimum.

Two categories of nonlinear programming methods are available for the solution of constrained optimisation problems, viz. direct methods and indirect methods. In the direct methods the constraints are handled in an explicit manner, whereas the indirect methods incorporate the constraints into the optimisation objective and proceed afterwards with solving an unconstrained problem. Figure 3.1 gives a classification of constrained optimisation techniques.

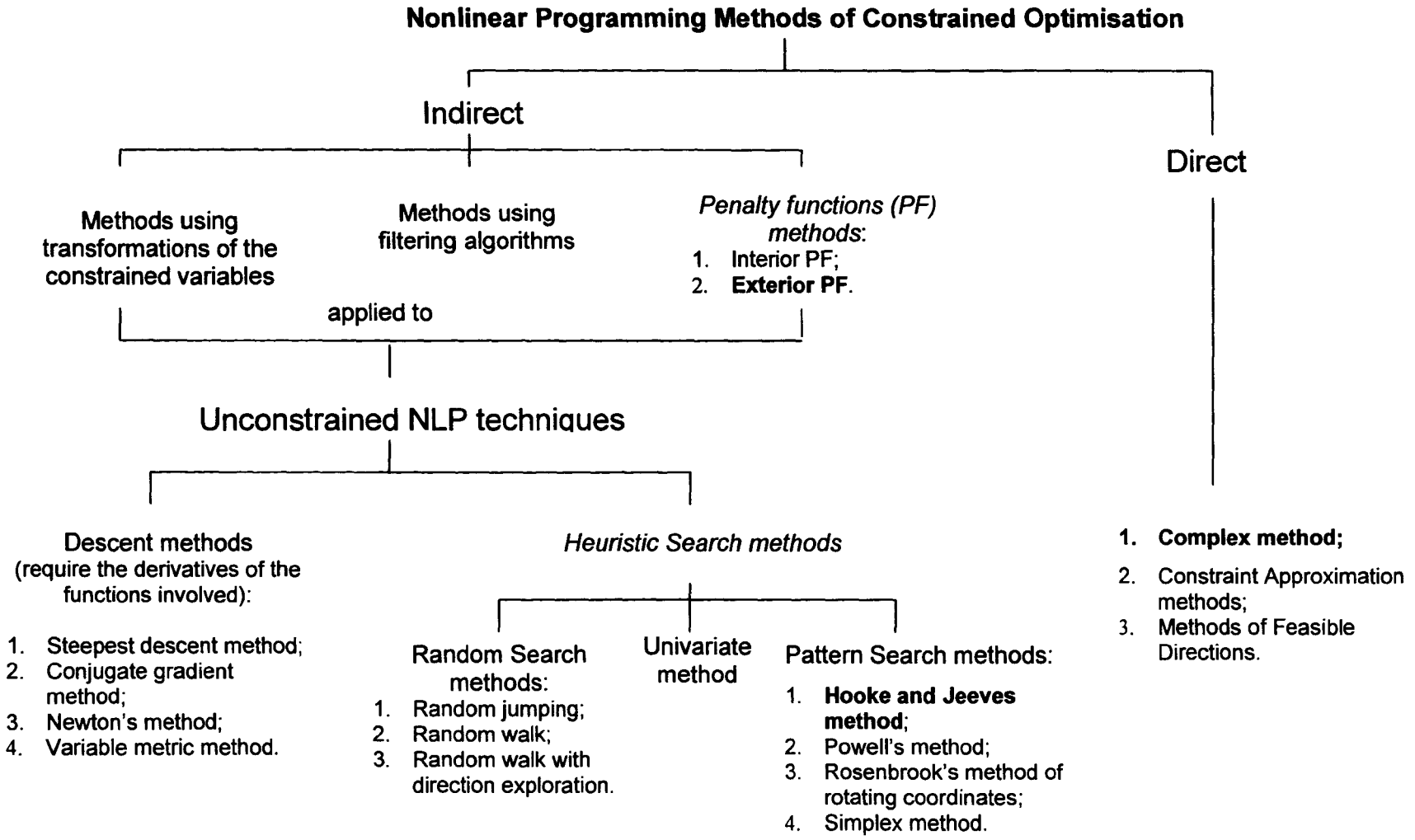


Figure 3.1. Taxonomy of the methods of constrained NLP optimisation

In the present work two well-established NLP techniques, viz. the Hooke and Jeeves method (HJ) and the Complex method (CM), have been selected and applied to cancer chemotherapy optimisation. The particulars of these methods will be provided shortly, but before that we need to justify the choice which has been made.

CM has been chosen to represent the category of direct methods since it is more versatile in comparison with the methods of constraint approximations and feasible directions. In the constraint approximation methods, the nonlinear objective functional and the constraints are linearised about some point and the problem is solved as a sequence of approximating LP problems. But in many inherently nonlinear situations, suitable approximations are often not available. The methods of feasible directions, on the other hand, produce an improving succession of feasible directions $\mathbf{S}_i$, along which at least a small step can be taken without leaving the feasibility domain. Therefore, the latter methods are confined to the feasible region, which makes them unsuitable in situations when the knowledge of the location of the feasible region is unavailable.

Amongst the category of indirect methods we will have to discard the methods that rely on supplementary operations with the objective functional or the constraints. By supplementary operations we mean either variable transformation or differentiation of the functions defining the optimisation objective/constraints. The reason for this is that firstly, for the optimisation problem of cancer chemotherapy, it is extremely difficult to find such a transformation of the control variables that will implicitly incorporate all problem constraints. Secondly, the gradient approaches based on differentiation of the objective functional $J(\mathbf{c})$ and the constraints $g_j(\mathbf{c}), j = \overline{1,S}$ make the results of optimisation dependent on the parameters of $J(\mathbf{c})$ and $g_j(\mathbf{c})$. Parameter dependence was the main disadvantage of the optimal control approach discussed in Section 3.2 and the original intent of mathematical programming techniques was to circumvent this disadvantage. Hence, we will exclude the methods of variable transformation and the descent methods altogether from the class of candidate NLP techniques suitable for cancer chemotherapy optimisation.

The category of indirect NLP methods will be represented by the method of penalty functions applied to heuristic search methods. Although the filtering algorithms developed by Fletcher and Leyffer (1998) also provide a robust approach (which is readily implemented and exhibits rapid convergence in practice), they do require the second derivatives of $J(\mathbf{c})$ and $g_j(\mathbf{c}), j = 1,S$ to be available. Amongst the heuristic methods of unconstrained optimisation only the pattern search methods will be considered. The random search methods, although very robust against discontinuity and multimodality of the objective functional, seriously lack

efficiency, which is indispensable for finding solutions in vast search spaces. The same is true for the univariate method, which permits changing only one control variable at a time.

Within the class of pattern search methods, our choice of an appropriate technique for optimisation of cancer chemotherapy will be narrowed down to the Hooke and Jeeves method. This can be explained by the relative simplicity of the method complemented by its versatility. Other pattern search methods have been discarded due to various reasons. The Powell's method is best suited for optimisation of quadratic objective functionals; the Rosenbrock's method of rotating coordinates is simply a further development of HJ, and finally, the Simplex method is an unconstrained version of the Complex method, which will be covered anyway.

This finalises our justification of why the Complex method and the penalty function approach to unconstrained NLP optimisation represented by the Hooke and Jeeves method, have been selected in the current work to investigate the capability of Nonlinear Programming in the context of cancer chemotherapy optimisation. Now the particulars of the chosen methods will be provided, starting with the Complex method.

3.3.2.1. Complex Method

The Complex method has been designed to solve constrained optimisation problems of the type: Optimise the objective functional

$$J(\mathbf{c}) = J(C_1, C_2, \ldots, C_n)$$

subject to the explicit constraints

$$l_i \le C_i \le u_i \qquad i = 1,2,\ldots,n$$

and also the implicit constrains

$$g_j(\mathbf{c}) \le b_j \qquad j = 1,2,\ldots,m.$$

The l_i and u_i are lower and upper bounds for the control variables. The optimisation is implemented by operating with a simplex, a definition of which follows.

Definition 3.3. A *simplex* is a geometric figure formed by a set of $k+1$ points in $k-$ dimensional space.

The basic idea of CM is to compare the values of the objective functional at the $k+1$ vertices of a simplex and move this simplex gradually towards the optimum point according to the following iterative process.

i. Initiate a feasible simplex (all vertices are feasible solutions).
ii. Find the vertex $\mathbf{C}_w$ with the worst value of the objective functional and form the centroid $\mathbf{C}_0$ of the other k vertices.

iii. Try to move away from $\mathbf{C}_w$ (this is how the search direction $\mathbf{S}_i$ is determined (see 3.30)) and so form a new solution point $\mathbf{C}_h$ by using one of three operations known as reflection, expansion and contraction.

iv. Test if $\mathbf{C}_h$ is feasible. If not apply the feasibility repair procedures (Bunday and Garside, 1987) until a feasible point is obtained.

v. If $J(\mathbf{C}_h)$ is better than the value of the objective functional at any of the remaining k vertices of the initial simplex, then replace the worst vertex by $\mathbf{C}_h$ and repeat the step ii onwards.

vi. Test whether method has converged.

A detailed description of CM, including Pascal code for its implementation, can be found in Bunday and Garside (1987). The Complex method has been utilised in the present work to solve the optimisation problem of minimising the tumour size by the end of chemotherapeutic treatment. We will postpone the discussion of the results of this optimisation until Chapter 5, where a systematic comparison will be made of CM with alternative optimisation techniques. Now let us examine how the method of Hooke and Jeeves differs from CM. But before that the method of penalty functions needs to be introduced, which allows the use of unconstrained optimisation in the context of constrained Nonlinear Programming.

3.3.2.2. Penalty Functions as a Constraint Handling Technique

For decades penalty functions have been a crucial part of constrained optimisation. There exist two basic types of penalty functions (PFs): exterior PFs, which penalise infeasible solutions, and interior PFs penalising feasible solutions (Smith and Coit, 1997). Interior penalty functions are applied to ensure that a given constraint is active (i.e. tight) in order to find an optimal solution lying on the boundary between feasibility and unfeasibility. Although potentially interesting, the idea of interior PF application is rarely used for multiple constraint problems.

Exterior PFs, on the other hand, will be extensively used throughout this thesis. These penalty functions allow three degrees of severity (Smith and Coit, 1997):

- barrier PFs in which no infeasible solution is considered;
- partial PFs in which a penalty is applied near the feasibility boundary;
- global PFs that are applied throughout the infeasible region.

It can be difficult to find a penalty function which immaculately converts an unconstrained optimisation problem into the constrained one. Many of the solutions with structure similar to that of the optimum solution will be infeasible. Therefore, restricting the search to only feasible solutions or imposing very severe penalties make it difficult to find promising directions towards the optimum. Conversely, if the penalty is not severe enough, then much of the search time will be used to explore territories far from the feasible region.

There are two approaches to penalise infeasible solutions. The simplest method takes into consideration only whether the constraints are satisfied or not. It is inferior to the second approach based on some distance metric d_j from the feasible region corresponding to the constraint $g_j(\mathbf{c}) \geq 0$ (Smith and Coit, 1997). In the presence of only inequality constraints $g_j(\mathbf{c}) \geq 0, j = \overline{1,m}$, distance-based PFs modify the objective of unconstrained optimisation $J(\mathbf{c})$ as follows:

$$\tilde{J}(\mathbf{c}) = J(\mathbf{c}) - \sum_{j=1}^{m} P_j(t) d_j^k \tag{3.31}$$

where P_j are penalty multipliers;

$$d_j^k = \delta_j \cdot g_j(\mathbf{c});$$

$$\delta_j = \begin{cases} 1 & \text{if constraint } j \text{ is violated} \\ 0 & \text{if constraint } j \text{ is satisfied.} \end{cases}$$

As can be seen from the equation (3.31), the distance measure d_j may be raised to the power k in order to either amplify or to reduce the penalising effect. Moreover, the penalty multipliers $P_j(t)$ define the PF type, which can be one of the follows:

- Static PFs: all penalty multipliers are constants;
- Dynamic PFs: some or all penalty multipliers are predefined functions of time;
- Adaptive PFs: some or all multipliers P_j undergo changes in accordance with the success (or lack of it) of the search implemented by a given constrained NLP technique.

In the present work static PFs will be used to express the maximum and cumulative dose limitations of the anti-cancer drug. The following distance measures correspond to these constraints:

$$d_1 = \sum_{i=1}^{n} \max^2 \{C_i - C_{\max}, 0\} \tag{3.32}$$

$$d_2 = \max\left\{\sum_{i=1}^{n} C_i - C_{\text{cum}}, 0\right\} \tag{3.33}$$

associated with constant penalty multipliers P_1 and P_2. The maximum tumour size constraint, on the other hand, will be incorporated into the optimisation objective by means of a dynamic PF. Since in many practical cases the entire eradication of the tumour is unachievable, one must give preference to those treatment regimens under which the tumour size constraint is violated later in the course of treatment. This may be achieved by variation of a penalty multiplier $P_3(t)$ over time, gradually decreasing the magnitude of penalty as the treatment period progresses. One possible method of penalty depreciation is to introduce a dynamic

exponential factor $e^{-\beta \cdot t}$ into a distance-based PF $P_3(t) = P_3 e^{-\beta \cdot t}$. The distance from the feasible region related to satisfaction of the tumour size constraint is measured as follows:

$$d_3 = \max\{N(t) - N_{\max}, 0\} \tag{3.34}$$

Having specified how the constraints of chemotherapeutic treatment can be included into the objective functional $\tilde{J}(\mathbf{c})$, we can now proceed with the unconstrained NLP optimisation of $\tilde{J}(\mathbf{c})$. Two concluding remarks however need to be made. Firstly, all penalty functions require to some degree the values of user-specified constants $P_j, j = \overline{1,m}$ defining the exploitation characteristics of the search. A systematic determination of the values for these constants is a problematic issue, which is currently under investigation. Secondly, throughout the present work we will adhere to the assumption that multiple constraints can be linearly combined to yield an appropriate penalty function. Intuitively, this seems to be a simplification of a general case since in many practical situations there is an interaction between constraints. Thus a real penalty is likely to increase more than linearly with the number of constraints violated. However the results of the current work will show that it is reasonable to assume in the present context that constraint violation incurs independent penalties on the objective functional. Now we are in the position to discuss the particulars of the Hooke and Jeeves method.

3.3.2.3. Hooke and Jeeves method

The pattern search method of Hooke and Jeeves is a sequential technique each step of which consists of two kinds of moves, viz. the exploratory and the pattern moves. The first kind is included to explore the local behaviour of the objective functional and the second is aimed at taking advantage of the pattern direction. The general procedure is described below.

i. Start with an arbitrarily chosen solution point $\mathbf{C}_1 = (C_1^1, C_2^1, , C_n^1) \in R^n$, called the starting base point, and prescribe lengths ζ_i in each of coordinate directions $\mathbf{e}_i, i = \overline{1,n}$.

ii. Compute $\tilde{J}(\mathbf{C}_k)$ ($k = 1$ to start with).

iii. Each control variable C_i^k is now changed in turn by adding the step length ζ_i. Thus first we evaluate $\tilde{J}(\mathbf{C}_k + \zeta_1 \mathbf{e}_1)$. If this improves the value of $\tilde{J}(\mathbf{c})$, then replace $\mathbf{C}_k$ by $(\mathbf{C}_k + \zeta_1 \mathbf{e}_1)$. If not, evaluate $\tilde{J}(\mathbf{C}_k - \zeta_1 \mathbf{e}_1)$ and replace $\mathbf{C}_k$ by $\mathbf{C}_k - \zeta_1 \mathbf{e}_1$ if improvement is achieved. If neither step gives an improvement, leave $\mathbf{C}_k$ unchanged and consider changes in the direction of $\mathbf{e}_2$, i.e. find $\tilde{J}(\mathbf{C}_k + \zeta_2 \mathbf{e}_2)$. When all control variables are examined we will have a new base point $\mathbf{C}_{k+1}$.

iv. If $\mathbf{C}_{k+1} = \mathbf{C}_k$, the exploration is repeated about the same base point $\mathbf{C}_k$ but with reduced step lengths ζ_i.

v. If $\mathbf{C}_{k+1} \neq \mathbf{C}_k$ we make a pattern move in the direction $\mathbf{S}_k = \mathbf{C}_{k+1} - \mathbf{C}_k$, i.e. $\mathbf{C}_{k+2} = \mathbf{C}_k + 2(\mathbf{C}_{k+1} - \mathbf{C}_k)$ since that move has already led to an improvement of $\tilde{J}(\mathbf{c})$.

vi. If $\tilde{J}(\mathbf{C}_{k+2})$ is better than $\tilde{J}(\mathbf{C}_{k+1})$ then a new base point $\mathbf{C}_{k+2}$ has been reached. In this case repeat step v.

vii. Otherwise abandon the pattern move from $\mathbf{C}_{k+1}$ and continue with exploration about $\mathbf{C}_{k+1}$ until the convergence criterion is met.

The Pascal code implementing this iterative procedure can be found in Bunday and Garside (1987). Similar to CM, the Hooke and Jeeves method has been applied to the problem of tumour size minimisation in the presence of the toxicity and the tumour size constraints. A comparative study of the results of this application needs to wait until we introduce an optimisation technique using Genetic Algorithms, utilisation of which for the optimisation of chemotherapeutic treatment constitutes the essence of the present work. One fact of that study however has to be mentioned here. It has been found that, although very efficient in optimising a unimodal objective functional subject to *a priori* knowledge of the feasible region location, the NLP optimisation methods become brittle when there is no feasible solution to start with. The situation worsens in cases when the optimisation objective is multimodal, i.e. there are a number of local optima surrounding the global one. In such cases the NLP methods tend to either converge to one of the local optima or fail to find any feasible solution at all. This severely undermines their usage for optimising cancer chemotherapy treatment.

3.4. Discussion

In this chapter the main mathematical approaches to optimisation of cancer chemotherapy have been discussed. These approaches utilise the theory of Optimal Control (based on the calculus of variations and on the Pontryagin Maximum Principle) and various methods of Mathematical Programming. Factors which determine what method is the most appropriate include the number of control and state variables necessary to describe the process, the linearity or nonlinearity of the objective functional, and the number and the form of the constraints.

The major deterrent to using the optimal control approach is the difficulty of solving the nonlinear differential equations resulting from the Euler equations and the Pontryagin's necessary conditions for optimality. Although a number of numerical solutions have been obtained based on the optimal control approach, these results are often parameter dependent and model specific.

Mathematical programming methods are more flexible in that respect. They can: (1) readily accommodate a large number of control variables; (2) avoid, in contrast to the optimal control methods, the problem of solving a large system of nonlinear differential equations, and (3) can be already implemented as standardised software packages. Besides, different methods of

Mathematical Programming have additional specific advantages. Linear Programming, for example, guarantees that the global optimum will always be found. Nonlinear programming methods provide an effective tool for optimisation of complex objective functionals subject to nonlinear constraints. Dynamic Programming is well suited to situations where the optimisation problem can be divided into a number of stages and a recursion relation between adjacent stages can be found. (Such situations might occur when discrete dose chemotherapy is used and the time interval of administration of each dose can be considered as one stage of a multistage process suitable for dynamic programming optimisation). Stochastic Programming deals with optimisation of problems which contain variables defined by probability distributions. (One possible application of Stochastic Programming is the optimisation of chemotherapeutic treatment when drug resistance might occur with a given probability.)

However, although potentially promising, the methods of Mathematical Programming have not found a widespread usage in the area of cancer chemotherapy optimisation. There have been no attempts reported in the literature to apply the dynamic and stochastic programming methods to optimise cancer chemotherapy treatment. Linear Programming can be used only in a small number of cases and under certain assumptions which are often unrealistic. Nonlinear programming methods perform well only when they are given a relatively simple optimisation objective and a good initial approximation of the problem's solution. In cases, however, when the optimal solution lies somewhere on the boundary between the feasible and infeasible regions and the location of the feasible region itself is unknown, the performance of NLP optimisation methods was unsatisfactory. According to Martin *et al* (1990, 1992), Murray (1990a,b) and Costa *et al* (1992), the best chemotherapeutic treatment strategies will always be at the brink of feasibility where one of the constraints is just about to be violated. Inability of the NLP methods to handle such cases means that they are deficient in precisely those situations when they are most needed.

Therefore, to provide clinical oncologists with a systematic, effective and reliable way to optimise cancer chemotherapy treatment, some novel approach has to be developed which is different from those of Optimal Control and Mathematical Programming. One possibility of such an approach emerges from the field of Evolutionary Computation (EC). This field encompasses different methods, often referred to as evolutionary algorithms, which draw inspiration from natural evolving systems to build problem solving algorithms. Evolutionary algorithms have demonstrated in recent years an ability to quickly discover useful solutions to problems that have been difficult to solve using classical optimisation methods (Fogel, 1997). In particular, Genetic Algorithms (GAs) have proved for a number of optimisation problems to be an effective alternative to methods of conventional optimisation. The next chapter provides an overview of Genetic Algorithms with the emphasis on how this method of evolutionary computation can be used to effectively search through large solution spaces.

CHAPTER 4

GENETIC ALGORITHMS AS A METHOD OF EVOLUTIONARY SEARCH AND OPTIMISATION

The original goal of Genetic Algorithms was to import into computer systems the mechanism of natural adaptation (Holland, 1975), where adaptation was envisaged as a 'process whereby a structure is progressively modified to give better performance in its environment'. We may assume that by structure he means a potential solution to a given optimisation problem and by environment he means the entire solution space. Then, the process of adaptation can be interpreted as a search for better solutions during which an optimisation of the search objective takes place.

The distinctive attributes that make Genetic Algorithms potentially suitable for dealing with problems intractable to mathematical methods of optimisation described in Chapter 3 may be summarised as follows:

- GAs implement multidirectional search by sustaining a population of candidate solutions;
- GAs explore the search space using stochastic processes rather than deterministic rules;
- GAs exploit the valuable information obtained so far by being biased towards the selection of "good" solutions and by utilisation of implicit parallelism;
- GAs require very little from the objective functional – the objective functional must unambiguously define the payoff of each solution, but may be multimodal, discontinuous and may allow a certain degree of imprecision.

Multidirectional search has two advantages. Firstly, it obviously increases the efficiency of the search by looking for the optimum in many directions simultaneously. Secondly, if the search in one direction gets stuck at a local optimum, there is still a chance to find the absolute optimum by approaching it from another direction. In addition to that, 'stochastic wandering' through the search space unburdens Genetic Algorithms from strong dependence on additional properties of the objective functional such as the gradient. Implicit parallelism, the concept of which will be

introduced later in this chapter, enables the algorithm to process much larger amounts of information than would have been possible using enumerative or purely random schemes and therefore also contributes to the increase of search efficiency.

The last advantage, i.e. the robustness of GAs with respect to the type of the objective functional, raises Genetic Algorithms to the position of a very versatile method, which can be applied to a large number of real life optimisation tasks. This versatility can be explained by the fact that GAs are not concerned with finding the optimal solution *per se*. Instead, their main goal is to find better solutions than those that are already known.

To achieve this goal Genetic Algorithms employ the evolutionary concept described in Section 4.1. The utilisation of such concept makes the finding of better solutions possible due to the rationale specified in Section 4.2. Section 4.3 focuses on implementation features of Genetic Algorithms. The efficiency of the GA method in performing its task can be improved by various enhancement techniques classified in Section 4.4. However, despite their remarkable robustness and versatility, Genetic Algorithms, similar to all other optimisation methods, have some limitations and drawbacks. These will be discussed in the last section (Section 4.5) of this chapter with the conclusion in favour of GA utilisation in the current work.

4.1. Evolutionary Concept of Genetic Algorithms

Genetic Algorithms, being a class of evolutionary algorithms, simulate a natural evolution process. A large number of GA implementations have been developed recently with a unifying aspect that all of them involve two main components: a population of strings encoding candidate solutions for the target problem and heuristics that manipulate these strings in search of an optimal solution (Lucasius and Kateman, 1993). A general form of the genetic algorithm structure can be presented as follows (Schwefel, 1995):

Step 0: *Initialisation*

> An initial population $P(0)$ (usually randomly generated) contains λ individuals and each of them is characterised by its genotype consisting of m genes. These genes determine an individual's fitness, i.e. adaptation capacity of an individual in the target problem environment. Each individual's genotype, or the encoded form of a potential solution, is represented by a digital string (referred to as a chromosome). Also, during this step a fitness function is specified, which assigns a numerical value to all individuals comprising the population. (The definitions of genes, chromosomes and fitness function will be given in the next section).

Step 1: *Selection*

Two parents from the population P are chosen for mating and copied to the mating buffer, MB, with probabilities proportional to their fitnesses, measured either by their contribution to the mean value of the fitness function in the current population (proportional selection) or by their ranks (e.g. linear ranking selection). The linear ranking scheme arranges all individuals in increasing order, assigns a predefined fitness maximum to the best individual and calculates the fitness of each subsequent individual in the population by subtraction of a certain value from the fitness of its predecessor. (The benefits of this scheme will be explained in detail later in Section 4.3). Thus, better individuals in the population are given more chances to get into the mating buffer MB and, therefore, more opportunities to produce offspring (reproduction with emphasis).

Step 2: *Recombination*

Next the individuals in the mating buffer MB undergo the genetic (structural) operators producing offspring MB'. These genetic operators are:

Crossover

Two offspring are produced by recombination (with a given probability p_c) of two parental genotypes by means of crossover. The application of the crossover operator involves dividing the parental genotypes into several pieces and systematic interchanging of these pieces between the parents to produce offspring genotypes.

Mutation

The genotypes of offspring produced from the 'crossover mill', undergo further modification via mutation, applied with a given probability p_m, to individual genes, either reversing them or randomly assigning a particular value to them.

After the new offspring have been created via the genetic operators, the two populations P and MB' must be merged to create a new population P'. Since most GAs maintain a fixed-sized population consisting of λ individuals, a total of λ candidates need to be selected from the parent and child populations. There are many strategies for accomplishing this task while imposing a selection bias towards "better" individuals either on the mating, or on the population, reassembling stages (Deb, 1997a).

This sequence of computations enables the population of candidate solutions to find "better" solutions in the search space by exploiting the advantages of the evolution process, which has two essential components: selection and reproduction. Selection can be viewed as a filter that determines, by competition, which individuals within a population survive to reproduce. Reproduction, which includes crossover and mutation, introduces an innovative aspect into the

evolution process through the genotype alteration of candidate solutions and, thus, provides a route for creation of increasingly successful individuals in the population. Finally, if we take into account the limitation of living resources, then only the fittest individuals, i.e. those most adapted to the target problem environment, will survive over time.

Therefore, the mechanism of evolution seems well suited to some of the most pressing computational problems, which require searching through a huge number of possibilities. Such search problems can often benefit from the effective use of parallelism, whereby many different possibilities can be explored simultaneously in an efficient way (Mitchell, 1996). This parallelism deserves more attention and is discussed in detail in the next section.

4.2. The Rationale for Genetic Algorithms

Although the description and implementation of Genetic Algorithms are fairly straightforward, their functioning is far from obvious. There still exist many open questions concerning how GAs work and what particular class of problems they are best suited for (Mitchell, 1996). Some work has been done on the theoretical foundation of GAs (Holland, 1975; Goldberg, 1989; Rawlings, 1991; Whitley, 1993; Whitley and Vose, 1995); yet controversy, even on basic issues, still remains (Beyer, 1997). However, to claim that results of a particular work are scientifically valid, one needs to back assertions by a rationale, justifying the data obtained. The results of the current work are consistent with the theory of schemas and building blocks, developed by John Holland (1975) and David Goldberg (1989); so, this theory is used as the explanatory rationale. In order to proceed a number of definitions need to be made.

Definition 4.1. A *search space* S is a set of objects over which search is to be conducted.

Let $A_1, A_2, \ldots, A_m$ be arbitrary finite sets, and let $I = A_1 \times A_2 \times \ldots \times A_m$.

Definition 4.2. The set I is called a *representation space* of candidate solutions defined on a Cartesian product of m finite sets $A_i, i = \overline{1,m}$ referred to as *alphabets* or *allele sets*. The elements of I are called *chromosomes* (or genotypes), whereas the elements of A_i are called *genes*.

Originally, the alphabets A_i were represented by the binary set $\{0,1\}$, which is still used in many GA implementations. In such cases chromosomes are referred to as bitstrings.

Definition 4.3. A function $g : I \to S$ which matches vectors in I to the solution space S is called an *encoding function*. Combined together, g and I form a representation of S.

Definition 4.4. (Schema) Let $I = A_1 \times A_2 \times \ldots \times A_m$ be a representation space. For each allele set A_i, define the extended allele set $A_i^+ = A_i \cup \{*\}$ where $*$ symbolises any element of A_i. Then a *schema* is any member of the set:

$$\Xi = A_1^+ \times A_2^+ \times \ldots \times A_m^+,$$

In other words, a schema is a chromosome in which any subset of the alleles may be replaced with the symbol $*$. Employing Goldberg's notation (Goldberg, 1989), the letter H is used to denote schemas because they can be regarded as hyperplanes in the representation space I. Thus, a schema $H = a_1 a_2 a_3 \ldots a_m$ describes a set of chromosomes which have the same alleles as H at all the positions i where $a_i \neq *$, i.e.

$$H = \{x \in I \mid \forall i \in \{1,2,\ldots,m\} : (a_i = x_i \text{ or } a_i = *)\}$$

Definition 4.5. The positions at which a_i is not $*$ are called the *defining positions* of a schema.

Definition 4.6. The *order*, $o(H)$, of a schema H is the number of defining positions it contains. The *defining length*, $\delta(H)$, of a schema H is the distance between its first and last defining positions.

The members of a schema are usually referred to as *instances*. If the representation space I is composed of only binary allele sets, then any chromosome of length m is an instance of 2^m schemas. Thus any GA population of λ chromosomes contains instances of between 2^m (if all bitstrings are identical) and $2^m \cdot \lambda$ (if all bitstrings are distinct) schemas. The importance of this fact closely relates to the concept of *implicit parallelism* of Genetic Algorithms. Implicit parallelism means that while explicitly evaluating λ chromosomes in the population, GAs in fact implicitly estimate the average fitness of a much larger number of all schemas, instances of which are present in the population. Having mentioned the schema fitness concept, we need to define it more rigorously. To do so we will need the following:

Definition 4.7. (Fitness function) Let S be a search space, $J : S \to R$ be an optimisation objective functional and let $g : I \to S$ be a representation of S. Then any function $f : I \to R$ with the property that $f(x)$ attains its maximum for the values of x which optimise $J(g(x))$, will be called a *fitness function*.

It is apparent that not all instances of any given schema H are evaluated during the evolution search implemented by Genetic Algorithms. The reason for this is that at any generation t, GAs operate with a finite population $P(t)$, which in general contains only a small portion of all instances of H. Thus, to estimate the fitness of H, the observed fitness of the schema is defined by

$$\hat{f}_H(t) = \frac{1}{N_H(t)} \sum_{x \in H \cap P(t)} f(x) \tag{4.1}$$

where $N_H(t)$ is the number of occurrences of the schema H in the current population $P(t)$. Although the average fitnesses of schemas are not explicitly calculated and stored in memory, Genetic Algorithms operate in such a way that all the averages are taken into account in terms of the increase or decrease in the number of instances of given schemas in the population (Mitchell, 1996). The number of instances $N_H(t)$ of a given schema H in the GA population at generation t is regulated by the following fundamental theorem of the Genetic Algorithm theory.

Theorem 4.1. (Schema Theorem) Let H be any schema over a representation space I being searched by a traditional genetic algorithm. Then

$$\left\langle N_H(t+1) \middle| N_H(t) \right\rangle \geq N_H(t) \frac{\hat{f}_H(t)}{\bar{f}(t)} \left[1 - D_c(H)\right]\left[1 - D_m(H)\right] \tag{4.2}$$

where $\left\langle A \middle| B \right\rangle$ is the conditional expectation of A given B;

$\hat{f}_H(t)$ is the observed fitness defined (4.1);

$\bar{f}(t)$ is the average fitness of the entire population at generation t;

$D_c(H)$ and $D_m(H)$ are the upper bounds on the disruptive effects on schema membership of the chosen crossover and mutation operators respectively.

A traditional genetic algorithm is one which uses fitness-proportional selection, crossover and mutation as generic operators, and generational update (i.e. replacement of the entire GA population with newly formed offspring at every evolution cycle). All these concepts will be explained in detail in the next section. The proof of the Theorem 4.1 is given in (Radcliffe, 1997).

In most cases it is difficult to determine $D_c(H)$ and $D_m(H)$ analytically. However, for a *simple genetic algorithm*, in which one-point crossover and bitwise mutation are used and applied with the probabilities p_c and p_m respectively, the upper bounds have been found as follows

$$D_c(H) = p_c \frac{\delta(H)}{m-1}$$
$$D_m(H) = p_m o(H)$$

where $\delta(H)$ and $o(H)$ are the defining length and the order of the schema H, m is the number of genes. One-point crossover means that parental genotypes are divided only into two pieces; bitwise mutation alters each gene's value with independent probability p_m.

The Theorem 4.1 is often pointed to as evidence of the implicit schema processing implemented by Genetic Algorithms. This is due to the fact that the theorem (potentially) gives a description of the dynamics of each schema instantiated in the population (Radcliffe, 1997). Also, the Schema Theorem advocates the idea that GAs achieve success through juxtaposition of short,

low-order, high-performance schemas, referred to as building blocks. The last statement is often formulated as the **Building Block Hypothesis** (Goldberg, 1989).

Proposition 4.1. (Building Block Hypothesis) Short, low-order and highly fit schemas are sampled, recombined and resampled to form chromosomes of potentially higher fitness.

The concept of a schema provides a way of decomposing a complex search problem into a hierarchy of progressively simpler problems. At first, GAs address the simpler problems and obtain a number of building blocks, which can be thought of as solutions to one of several noncompeting subproblems comprising the problem under investigation. (The subproblems are noncompeting in the sense that a single candidate solution may combine two or more building blocks.) Then, Genetic Algorithms combine the building blocks into more complex solutions until eventually the complete problem is solved.

In mathematical terms the decomposability of a problem into subproblems is referred to as *linear separability*, and models satisfying this condition are known as *additive* (Radcliffe, 1997). In the context of Genetic Algorithms, the biological term *epistasis* is used to describe a range of nonadditive phenomena (Holland, 1975). Epistasis means interaction between individual genes or groups of genes in chromosomes and determines the potential for crossover to succeed in assembling chromosomes representing good solutions through recombining useful building blocks. If the level of epistasis is too high, then the building blocks become long enough to prevent Genetic Algorithms from succeeding in finding better solutions. Therefore, the GA rationale can be summarised as follows: Given an acceptable level of epistasis, Genetic Algorithms implement efficient search through the solution space and find better solutions due to implicit parallelism of the search and the ability to bring together short and highly fit schemas (building blocks). In the next section we will consider the special features of Genetic Algorithms allowing such an efficient search to be implemented.

4.3. The Features of Genetic Algorithms

In addition to the theoretical aspects described in the previous section, each genetic algorithm application needs its own practical aspects. This section is dedicated to these aspects. We start with the most problem specific property – the fitness function.

4.3.1. Fitness function specification

From the Definition 4.7 it can be seen that the fitness function $f : I \rightarrow R$ returns a numerical value for each chromosome in the representation space I. This value is related to the quality of the solution represented by a given chromosome and is calculated using the objective functional J, on the basis of which the solution quality is judged.

The general rule in constructing a fitness function is that it should represent the actual "goodness" of a potential solution, i.e. good solutions should have better fitness values. However, "goodness" is not always a useful measure for guiding a genetic search. For example, in many optimisation problems with constraints most points in the search space are infeasible, i.e. represent invalid solutions to the target problem. Therefore, their "goodness" has zero value. For a GA to be effective in this case, we must invent a fitness function where the fitness of each individual solution reflects its ability to guide a genetic search towards the feasible and, ultimately, the optimal region in the solution space. But in order to do this we need to know *a priori* the location of the feasible region, giving good fitness values to nearby solutions and poor values to those far away. In reality, however, this information is rarely available. There are several methods whereby it is possible to overcome this problem. Cramer (1985), for example, suggests that better results can be obtained if a few meaningful subgoals are invented and the solutions that achieve them rewarded. Another approach, also used in the present work, is to apply penalty functions that represent how poor a candidate solution is or how violently it breaks the constraints of the target problem. Richardson *et al* (1989) suggest the construction of a penalty function from the 'expected completion cost' point of view, i.e. how much will it cost to turn a particular infeasible solution into a valid one. Other methods for fitness function construction exist, leaving at the same time considerable scope for further work in this area.

After the fitness function has been constructed, the fitness values of all chromosomes in the population can be calculated. Since the initial population has a random composition, there is a wide spread of individual fitnesses. However, after a while particular strings begin to predominate and the range of fitness values reduces. This variation in fitness range throughout a GA run often leads to problems of *premature convergence* and *slow finishing* (Beasley *et al*, 1993). Premature convergence occurs when comparatively highly fit (but not optimal) individuals rapidly dominate the population. This causes elimination of the crossover effect since swapping the parts of identical strings cannot produce anything new. Only mutation performs exploration of the search space and this is carried out in a slow and random fashion. The opposite problem to premature convergence is slow finishing, which means that after many generations the population will largely converge but may still not have precisely located the global optimum. The difference in fitness values of the best and average individuals in the population will not be large enough to lead GAs towards the optimum.

In order to combat these two problems several methods can be employed to expand the effective fitness range. The first method is *fitness scaling*, where the maximum number of chances to be selected for reproduction allocated to a particular string is set to a certain value. This is achieved by subtracting a suitable value from the raw fitness score, then dividing it by the average of the adjusted fitness values (Beasley *et al*, 1993). The second method is *fitness windowing*, used by Grefenstette (1986). It is very similar to fitness scaling, except that the amount to be subtracted is the minimum value of solution fitness encountered in the last w populations. The windowing size w varies depending on the target problem. The last method,

essentially different from the previous methods, is *fitness ranking*. The ranking scheme overcomes the reliance on extreme individuals because all strings are sorted in order of non-scaled fitness and then reproductive fitness values are assigned according to either linear or exponential rank, allowing intermediate strings to be regularly spread out. Several experiments have shown that fitness ranking is superior to fitness scaling (Baker 1985; Davis 1989). This conclusion is also supported by the present work.

4.3.2. Selection Techniques

Selection is one of the main operators used in evolutionary algorithms. The primary objective of selection is to emphasise better solutions in a population (Deb, 1997a). Good and bad solutions in a population are distinguished on the basis of fitness and the essential idea of evolutionary selection is that a solution with a better fitness must have a higher probability to be chosen for reproduction.

There are two major factors that determine the effectiveness and efficiency of the evolution process during a genetic search: population diversity and selective pressure. High selection pressure often leads to insufficient population diversity and, as a result of that, to premature convergence of Genetic Algorithms. To strike the balance between these two factors an appropriate selection operator needs to be chosen.

There is a wide spectrum of selection operators, which can be divided into two groups (Beasley *et al*, 1993) – deterministic and stochastic operators. The former operators sort the population according to fitness and deterministically choose the best few solutions. Stochastic operators, on the other hand, assign a probability of selection to each solution according to its fitness and operate using this probability distribution. Since in a finite initial population the best few individuals may represent a suboptimal region, the deterministic selection operators by consistently emphasising these seemingly good solutions may finally converge to a wrong solution. In similar situations stochastic operators maintain diversity in the population, therefore widening the search, via occasional choices of not-so-good solutions. For this reason we will consider stochastic selection methods only, since the problem of cancer treatment optimisation, which is the essence of the current work, is prone (as will be shown later) to having many suboptimal solutions militating against the use of deterministic methods.

The most prominent stochastic selection operator – the proportionate operator – assigns the expected number of copies to a particular solution in proportion to its fitness. The simplest form of the proportionate selection scheme is known as roulette-wheel selection, a short description of which is now provided.

Suppose a population contains λ chromosomes, $x_1, x_2, ..., x_\lambda$, with fitnesses $f_1, f_2, ..., f_\lambda$ respectively. We may assume that all fitnesses fall in the interval (0, $F_{\max}$]. We then select λ' times with replacement from the population according to the following scheme:

1. Calculate the total fitness $f = \sum_{i=1}^{\lambda} f_i$.
2. Calculate the proportion $p_i = \frac{f_i}{f}$ of total fitness for each chromosome x_i.
3. Divide the unit interval [0,1] into λ subintervals $[t_0, t_1], (t_1, t_2], ..., (t_{\lambda-1}, t_\lambda]$ where $t_0 = 0$ and
$$t_i = \sum_{k=1}^{i} p_k \ , \ 1 \le i \le \lambda$$
4. (Repeat λ' times) Calculate a random number, r, in [0,1]; r will lie in a subinterval containing precisely one t_i. Select chromosome x_i for reproduction.

Note that $[t_{i-1}, t_i]$ has length $p_i, 1 \le i \le \lambda$, so that [0,1] is partitioned proportionately according to the relative fitness of the λ chromosomes in the population. This process results in composition of a mating buffer, MB, which is a subset of the original population and consists of the $\lambda'(\le \lambda)$ chromosomes selected for reproduction. Since the solutions are marked proportionally to their fitness, a solution with a higher fitness is likely to receive more copies than an inferior solution.

The limitations of the proportional selection scheme, however, is that negative values for a solution's fitness are not allowed and minimisation problems cannot be handled directly (they must at first be transformed to equivalent maximisation problems). Also if a population contains a solution whose fitness substantially exceeds those of the rest of the solutions, this 'supersolution' very soon will dominate the population, which will inevitably lose its diversity and converge prematurely. Another difficulty may arise when most of the population members have more or less the same fitness and the proportional selection operator cannot make a distinction between different solutions and thus results in almost a random selection. However, the last two problems can be avoided by using a fitness ranking scheme, where every solution is linearly mapped between a lower and an upper bound before making a roulette wheel.

Another approach to stochastic selection is the use of a tournament selection operator, which has no restriction on negative fitness values and is very suitable for a parallel implementation (Deb, 1997a). There are several variants. In the simplest of them, binary tournament selection, a pair of strings is picked at random from the population, and whichever has better fitness is copied into the mating buffer for the next generation. This process is repeated until the mating buffer is full. In some applications larger tournaments are used where the best from n $(n > 2)$ randomly chosen strings is copied into the mating buffer. Also, additional complexity may be

included into tournament selection by choosing the best string only with a certain probability p (obviously p should satisfy $0.5 < p < 1.0$); using lower values of p would result in a decrease in selection pressure.

The proportional and the tournament selection keep the selection pressure more or less constant during a GA run. But often different amounts of selection pressure are needed at different times in a run. For example, early on it might be good to allow less fit chromosomes to be selected at the rate close to that of fitter chromosomes, thereby maintaining a lot of variation in the population. Later, however, an increase in selection pressure might be useful to strongly emphasise highly fit chromosomes and to speed up the convergence to the optimal solution. One approach to regulate the selection pressure is to use the Boltzmann selection operator (similar to simulated annealing), in which a continuously varying "temperature" controls the rate of selection according to a pre-set schedule (Mitchell, 1996).

The last aspect of GA selection worth mentioning here is generation overlapping. Most GAs replace the entire population with newly formed offspring. Some of these offspring will be identical to their parents, but there is no direct path (avoiding the selection procedure) from one population to another. There exist population formation schemes, however, which use elitism to a certain degree, that is a portion of the previous generation is retained in the new one. This inherited portion is usually called a generation gap, and this gap allows Genetic Algorithms to preserve the best solutions found so far. The population replacing schemes are usually characterised by a low value of a generation gap. In steady-state selection, on the other hand, only a few individuals are replaced in each generation and therefore a generation gap has a larger magnitude. In the present work it has been decided to utilise the elitist strategy that fixes a potential source of loss (due to a randomised search) by copying the single best member of each generation into the succeeding generation.

4.3.3. Genetic operators

Two essential procedures comprise every efficient search: exploration and exploitation of the search space. Biased selection, however achieved, may be regarded as the exploitative part, which is necessary but not sufficient for efficient search. Without the explorative part no variation is imposed on the population and this causes depletion of population diversity. In order to counterbalance this depletion some exploration heuristics are required, which lead to better performing members but do not waste too much useful information already collected in the population (Lucasius and Kateman, 1993). To allow GAs to carry out such exploration the following principle must hold.

Proposition 4.2. (Principle of Maximal Preservation) Given the amount of variation to be imposed on the population, this should be done in such a way that there is a maximal preservation of information which has accumulated in the population during past generations and which is important for the purposes of the search task.

The two most commonly used genetic operators implementing exploration in accordance with the Principle of Maximal Preservation are mutation and crossover. Let us discuss them in turn.

4.3.3.1. Mutation

Looking at the Principle of Maximal Preservation, we can make a proposition that if we change a tiny amount of genes in chromosomes leaving all others alone, then a small variation will be imposed on the population with very little disruption of assembled information. The best known mechanism for producing variations is mutation, where one allele of a chromosome is randomly replaced by another (Eshelman, 1997). If a binary representation is used, then mutation is achieved by 'flipping' bits at random.

Mutation is traditionally viewed as an operator responsible for reintroducing irreversibly lost genes. This means that if the population consists of chromosomes all having the same value of a particular gene then it is impossible, using selection and crossover only, to explore the region in the representation space wherein the value of this gene is different. It is generally believed that mutation by itself cannot solve a complex optimisation problem (Beasley *et al*, 1993). However, in combination with the crossover operator, mutation is able to tackle very complex tasks and is one of the main sources (together with a random start) of raw ideas for evolutionary exploration.

4.3.3.2. Crossover

The intuitive idea behind crossover is easy to state: given two individuals who are highly fit, but for different reasons, ideally what one would like to do is to create a new individual that combines the best features from each (the Building Block Hypothesis). Since we do not know *a priori* which features account for the good performance, the best we can do is to recombine features at random using a different number of cross points.

The number of cross points varies from one (Holland's classical recombination, denoted **1X**) to $m-1$, where m is the length of a solution string (uniform crossover). There are, of course, intermediate options, with a special name of multi-point crossover **MX**. One often-used version of it is 2-point crossover. According to Spears and DeJong (1991), it achieves a compromise between striving for minimal Holland's schema disruption and the theoretical argument that no disruption occurs for an even number of breakpoints, falling down into gaps between each of

the defining positions of schemas (Lucasius and Kateman, 1993). Figure 4.1 illustrates the 2-point crossover operation:

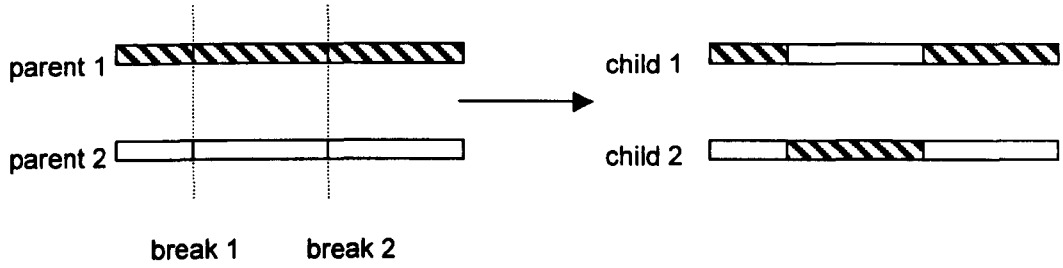


Figure 4.1. Two-point crossover

The success or failure of a particular crossover operator depends in complicated ways on the fitness function, encoding and other details of Genetic Algorithms. There are many aspects of the crossover operator (positional bias, disruption potential, crossover probability) as well as of other GA features (fitness function, selection scheme and mutation), which can be used to improve the efficiency of the GA search. The next section gives a survey of different techniques used for GA performance enhancement.

4.4. Enhancement of Genetic Algorithms

In order to achieve a better performance of Genetic Algorithms a number of decisions need to be made, viz. what types of GA operators and which control factor settings are likely to produce the desired effect in a short computation time. Control factors quantitatively characterise various GA features and generally include the probabilities of crossover and mutation, characteristics of the selection scheme, various parameters of the fitness function and some other aspects of genetic operators. The following subsections discuss the techniques which help to make the decisions on how to choose the GA operator types and control factor settings easier. These techniques can be classified into three groups which will be addressed in turn.

4.4.1. Enhancement of GAs via parameterisation

This subsection focuses on the issue of selecting numerical values for GA control factors. Since GAs have many factors, it is often very difficult to select appropriate values in order to draw out a maximum capability of the GA method. In recent years a number of new techniques have been proposed which optimise in one way or another the values of GA factors. Lucasius and Kateman (1993) and Hatta *et al* (1997) give short surveys of these techniques. The following lines of approach are most commonly used in practice.

4.4.1.1. *Ad hoc and skillful optimisation*

In this approach GA practitioners either choose GA factors according to the guidelines suggested in the literature or make guesses about rough estimates of the optimal factor setting, i.e. 'guestimates', prompted by their own experience. According to Michalewicz (1994), 'it seems that finding good values for the GA factors is still more an art than a science'. This observation highlights the fact that very often in practice, skillful hand optimisation is the fastest and most reliable way for obtaining high performing GA factor configurations because various shortcomings generally prevent other approaches from widespread usage (Lucasius and Kateman, 1993). However, ad hoc and skillful GA factor optimisation does not have any underlying rationale, which facilitates a straightforward transition from one problem of GA factor adjustment to another. This means that if, for example, the objective function or encoding scheme have been changed for a particular task, then the routine of GA factors tuning has to be repeated all over again requiring the same amount of the practitioner's effort.

4.4.1.2. *Meta-optimisation*

If one wants to introduce an automated approach to the optimisation of GA factors– why not use Genetic Algorithms themselves; after all, they are designed for optimisation purposes? Grefenstette (1986) pioneered this approach, having introduced the concept of "meta-level GAs", which operate on a population of GA factor sets. Base-level GAs in Grefenstette's approach perform an optimisation of a test suite of objective functions introduced by DeJong (1975). Each individual in the meta-GA population encoded six GA factors: population size, mutation and crossover rates, generation gap, windowing size and the type of selection. The six-dimensional GA factor space was discretised, and only a small number of different values was permitted for each factor. Grefenstette managed to obtain factor settings that slightly outperform those that DeJong had found by skillful optimisation. Another attempt was made by Bramlette (1991) to use Genetic Algorithms for meta-optimisation of "subject" GAs with non-binary representation schemes.

Pham (1994) repeated Grefenstette's approach as a preliminary step towards so-called competitive evolution. In his method, several populations, each with different GA operator variants and factor settings, evolve simultaneously. At regular intervals, the populations' performances are compared, and only the population with the highest improvement rate in the recent past is allowed to evolve for a few more steps until another comparison is made. The competitive evolution approach is similar to a parallel (island) population model, which is a method of optimising the GA configuration.

Concluding the description of meta-evolutionary approaches, we can say that the usage of Genetic Algorithms for GA factor adjustment is less computationally intensive in comparison with an exhaustive search. Nonetheless, the meta-optimisation procedure permits only a limited number of GA factors to be taken into account in order to complete its task within reasonable

time constraints. Also, it concentrates only on the point estimation of GA factors, not giving the overall picture of the influence of different factors on the GA performance.

4.4.1.3. Optimisation by systematic search

An example of systematic search is a full factorial design (or grid search) in the set of GA factors. The main merit of this strategy is its guaranteed success, i.e. the best set of GA factors will ultimately be found. Moreover, the full grid search facilitates the identification of the GA response surface, which defines the performance of Genetic Algorithms as a continuous function of GA factors. It is obvious that for continuous GA factors a search grid of any resolution will leave some settings of GA factors unevaluated. The response surface, on the other hand, provides global information on the quality of all factor settings subject to a certain level of statistical confidence.

One obvious disadvantage of full factorial experimentation, however, is that the computability of this task reduces very rapidly with the number of control factors. For example, Schaffer *et al* (1989) spent more than one CPU year, using a fairly fast computer, to carry out a full factorial search for a relatively small set of GA factors. Therefore, in order to utilise the merits of systematic search and also to be able to accomplish the factor optimisation task in reasonable time, a compromise to an exhaustive grid search ought to be found. A fractional factorial design can be a good alternative here.

In fractional factorial designs a substantial reduction in the number of experiments is achieved by assuming that GA factors are either independent of each other or not significantly interacting in a statistical sense. In general, such assumptions are not always justified because some GA factors do interact. However in some cases the effect of this may be negligible. On the basis of the fractional factorial approach to designing statistical experiments with the GA control factors a novel method of GA enhancement will be developed in Chapter 5. Also, it will be shown that this novel method is often more advantageous in comparison with other techniques of GA factor optimisation.

4.4.1.4. Dynamic optimisation

The last approach to enhancement of Genetic Algorithms via parameterisation is dynamic adaptation. Adaptive tuning is the method which determines the values of GA factors dynamically during the execution of a program. It has been noticed (Mitchell, 1996), that the optimum population size, crossover and mutation rates as well as selection pressure are likely to undergo some changes during a GA run. To monitor these changes the adaptation of factor values to the ongoing search in real time has been suggested. It can be achieved either by a pre-programmed schedule or by self-adaptation.

The idea behind a pre-programmed adjusting strategy is to mimic a cooling schedule of simulated annealing (Lucasius and Kateman, 1993). Davis (1989) and Syswerda (1991) found that linear variations in crossover and mutation rates could be advantageous when crossover decreases during a GA run accompanied by an increase in mutation. However, such fixed schedules do not monitor closely enough the state of the GA population. These schedules keep changing some factors when there may be no need to do so. Booker (1987) adopted the scheme in which the crossover rate alters depending on the spread of fitness – for a more or less converged population the rate of crossover reduces, automatically increasing the rate of mutation and offering the latter more chances to explore the solution space. A serious drawback of pre-programmed schedules is that they replace the GA factor set by a large, and probably more complex, space of candidate adjustment strategies.

Self-adaptation, on the other hand, appears to be more flexible. It was most noticeably pioneered by Davis (1989). He monitors the fitness of each heuristic operator on the basis of the operator's success in producing good offspring. Each operator starts with the same initial fitness. Every time an operator is chosen to create a new candidate solution, which replaces a low-fitness member of the population, this operator increases its significance weight, consequently altering the probability to be applied next time. A weighting figure is allocated to each heuristic operator based on its performance over the past 50 matings. Therefore, during a course of a GA run operator probabilities vary in an adaptive, problem-specific way (Beasley *et al*, 1993).

There are other approaches to self-adaptation. Tuson and Ross (1996), for example, have investigated the *coevolutionary* approach. In this approach the factors that control the evolutionary search process are encoded into GA chromosomes. That is, the representation space I is given by

$$I = I_X \times I_S$$

where I_X denotes the set of object variables (i.e. representation of solutions) and I_S denotes the set of GA factors; thus, the control factors of Genetic Algorithms are allowed to evolve as part of the solution process. The general observation however is that, due to a large variation in GA performance, dynamic optimisation leads to reliable results only when a limited number of control factors are taken into account (Lucasius and Kateman, 1993). Furthermore, experience confirms that the approaches, which separate the factor adaptation mechanism from the main genetic algorithm, show a better performance (Tuson and Ross, 1996).

4.4.2. Enhancement of GAs via configuration

A general form of the GA structure was presented in Section 4.1 and included three component procedures: initialisation, selection, and recombination. The particularization of these procedures is referred to as a GA configuration. In other words, a GA configuration defines a specific implementation within the family of Genetic Algorithms. Having only briefly discussed

the vast variety of initialisation and selection techniques and taking into account the fact that the number of their combinations is even larger, it becomes evident that the family of Genetic Algorithms is excessive. Due to this configuration flexibility, the GA methodology has a versatile ability to adapt to very divergent problems (Lucasius and Kateman, 1993).

However, configurational flexibility can be used not only for adaptation of Genetic Algorithms to a given search or optimisation task, but for enhancement of the GA performance as well. A rich arsenal of tools that can significantly enhance the efficiency of Genetic Algorithms include but are not limited to:

1. Different representation schemes such as:
    - the method of diploidy and dominance (Goldberg, 1989);
    - niche exploration strategies (Beasley *et al*, 1993);
    - sharing and crowding schemes.
2. Various genetic operators, for example:
    - inversion;
    - duplication and deletion (Holland, 1992);
    - segregation and translocation.
3. Novel evolutionary methods such as:
    - messy GAs (Goldberg *et al*, 1991),
    - migration and mating restrictions.

A comprehensive account of all the aforementioned routines may be found in the literature; here we will briefly discuss only one configurational technique which will be utilised in the current work and involves a restarting facility.

Using Markov chain analysis, Ghannadian *et al* (1996) analytically derived an expression for the expected time required by GAs to reach an optimal solution. The results obtained can be used to safely restart Genetic Algorithms whenever they have been working longer than the expected time. By systematically applying the restart procedure we may reach an optimal solution faster than if we waited for the results of long, unsuccessful GA runs. Needless to say however, the method of random restart has its own limitations. One of the most obvious is the apparent difficulty in precise estimation of the expected time for real-life problems.

In conclusion of this subsection it needs to be pointed out that despite structural simplicity of Genetic Algorithms, the overall evolution process is mechanically very complex due to mutual effects of operators, procedures, and GA parameters on each other. Having realised that, it is not difficult to understand why the search for an adequate GA configuration still remains an innovative and creative endeavour, rarely amenable to analytical approach. The best configuration strategies actually appear to be those based on intuitive insights obtained from empirical experience (Lucasius and Kateman, 1993).

4.4.3. Enhancement of GAs via hybridisation

In their canonical form Genetic Algorithms implement a blind search by randomised recombination of encoded solutions and by exploiting the fitness function value in order to determine preferable trials in the next generation (Goldberg, 1989). On the one hand, this makes the GA search robust; but, on the other hand, the fact that Genetic Algorithms do not utilise the domain-specific knowledge very often puts them at a disadvantage. If knowledge-augmented operator and/or hybrid techniques are used, the efficiency of GAs is likely to improve (Deb, 1997b).

Therefore, whenever problem specific information exists, it is natural to consider a hybridisation of Genetic Algorithms with existing optimisation methods. These methods range from exact algorithms discussed in Chapter 3 to meta-heuristic algorithms, for instance simulated annealing and tabu search (Ibaraki, 1997). The resultant hybrid algorithm is likely to do better than just a simple GA since it incorporates a good deal of domain-specific knowledge. At the same time, it almost certainly outperforms the domain-based heuristic on its own, because the good features of this heuristic will be exploited in the hybrid system together with all merits of evolutionary search (Davis, 1991). In short, hybridisation enhances the GA performance by achieving a substantial reduction in computational time and results in more accurate or precise end-solutions (Lucasius and Kateman, 1993). Figure 4.2 presents different approaches to the hybridisation procedure, explanation of which follows.

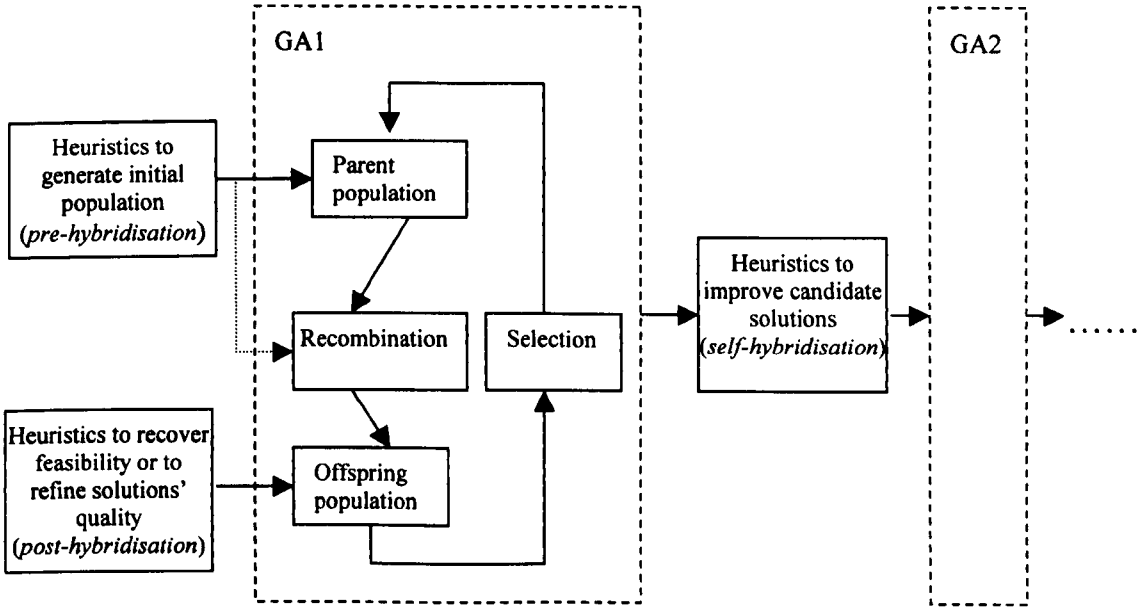


Figure 4.2. Genetic Algorithms and their hybridisation

4.4.3.1. *Pre-hybridisation*

In some practical situations an inaccurate, yet not entirely useless, estimate of the solution to a given problem can be obtained from some knowledge source. Thus the initial GA population can be constructed in such a way that it contains a few copies of a solution representing the initial estimate, which is expected to facilitate convergence speed. Such an approach comprises the essence of pre-hybridisation, which may also be involved in the encoding procedure or in the amendment of recombination operators.

4.4.3.2. *Post-hybridisation*

The purpose of post-hybridisation is to refine the end-solution found by Genetic Algorithms, whenever the precision of the GA search is comparatively poor (Lucasius and Kateman, 1993). For precision improvement a local search technique is used; thus, GAs may in this case be regarded as a hill-finder and the local search technique subsequently invoked as a hill-climber. Traditional optimisation techniques may serve the role of a hill-climber very well. The only problem is that they usually require an estimate solution to start with. Given such a starting solution many mathematical programming methods can find a better solution in a more efficient way.

In many practical situations, however, a feasible starting solution is unknown and traditional optimisation techniques on their own may fail to find one. Fortunately, robustness of the GA search often helps in locating the feasible region in the solution space (McCall and Petrovski, 1996). After that, well-established hill-climbing techniques (such, for example, as the Complex method or the method of Hooke and Jeeves), which heavily rely on the structure of a problem domain and therefore are capable of accomplishing the search much faster, can be utilised for precision improvement.

4.4.3.3. *Self-hybridisation*

Lucasius and Kateman (1993) use the term self-hybridisation in the sense of hybridisation of one genetic algorithm with another, differently configured. The result of such iterated reconfiguration is a chain of genetic algorithms each of which – apart from the very last one – passes its results to a successor for further improvement.

One example of self-hybridisation is a combination of Genetic Algorithms with Delta-Coding, introduced by Whitley *et al* (1991). In Delta-Coding, instead of encoding solutions as GA chromosomes, one encodes modifications or 'delta changes' to a given solution. Therefore, when the GA population converges, the numeric representation is remapped so that the solution ranges are centered around the best value found so far, and the algorithm is restarted. Whitley suggested that a conventional GA could be run once and when the best solution is obtained it is passed onto the following delta-coding algorithm for further improvement. In the next chapter

such self-hybridisation will be used to find optimal modifications of cancer treatment strategies obtained either from GAs or supplied by oncologists.

4.5. Discussion

In this chapter a survey of salient GA features has been presented. In essence, Genetic Algorithms are based on a conceptually simple application of the principle of competition for limited resources and adaptation to surrounding environment. Despite structural simplicity of Genetic Algorithms, various recombination, selection, and fitness scaling routines provide significant versatility and configurational flexibility, which result in the efficient and robust nature of the GA search. Moreover, the search effectiveness and accuracy can be substantially improved by a wide range of parameterisation, configuration and hybridisation techniques.

Needless to say however that, similar to all optimisation techniques, Genetic Algorithms have their own limitations and disadvantages. The limitations mainly come from improper choice of GA factors such as the population size, the probabilities of recombination operators, and selection pressure. GAs are not expected to work on arbitrary problems with arbitrary factor settings. Thus, to implement an efficacious search with the help of Genetic Algorithms, it is often necessary to adjust GA factors in an appropriate manner using either some guidelines from the literature or utilising specific GA factor optimisation techniques described in the previous section.

Secondly, Genetic Algorithms belong to the class of randomised techniques. This implies that it is always likely that different results will occur after repeating the same experiment. The difference in the outcome may be very large and this large intrinsic variation makes it difficult to predict, model and analyse the behaviour of Genetic Algorithms. Moreover, the performance of most evolutionary algorithms, GAs in particular, largely depends on the quality of the chosen random number generator. With a biased random number generator, the randomness of the GA operators will be lost and, although it is not a limitation of the GA method *per se*, the performance of Genetic Algorithms may be disappointing. To ensure that the random number generator used is adequate, a designated statistical test has been carried out in the course of the present work. The results of this test will be mentioned in the next chapter.

Another prominent weakness of GAs is the problem of deception. Deception occurs when two schemas H_1 and H_2 have the property that the average fitness of H_1 is greater than the average fitness of H_2 even though H_2 includes a solution that has a greater fitness than any member of H_1. In practice it means that the lower-order building blocks lead GAs away from the global optimum. Depending on the excess in the fitness average of the deceptive schema over that of the proper schema, all deceptive problems are categorised into two types: partial and full deceptive. Fully deceptive problems are very difficult to solve using Genetic Algorithms.

The difficulties associated with problem's deceptiveness as well as with high epistasis (i.e. gene interaction) which has been mentioned earlier, may be tackled in two ways (Beasley *et al*, 1993). The first approach is based on alteration of the encoding procedure. Vose and Liepins (1991) pointed out that in principle an appropriate encoding procedure can transform a highly epistatic (deceptive) problem to a problem of low epistasis (deceptiveness). So, it is always possible to reduce the level of epistasis or deception. Unfortunately, for some problems the effort involved in developing such an encoding scheme is comparable in difficulty with solving the original problem itself.

The second approach, on the other hand, employs theoretical analysis of GA performance to answer the question whether a given problem is solvable by Genetic Algorithms. A number of theoretical methods have been developed to analyse the performance of GAs and can be divided into two categories, viz. analytical and stochastic. Strictly speaking, however, all outcomes of GA runs are probabilistic in nature. Therefore, in the context of GA performance analysis, by "analytical" is meant methods which are aimed at providing an understanding of the nature of the search space in term of GA structures, i.e. digital representations of candidate solutions and Holland's schemas. For example, the method of a dual solution space developed by Battle and Vose (1993) suggests a way whereby it is possible to transform some of fully deceptive problems into partly deceptive ones and find their solutions. A second analytical method of stable fixed points in the search space (Vose and Liepins, 1991; Vose and Wright, 1995) analyses the points of convergence of the evolutionary operator $\Gamma : S \to S$, which is defined on the search space S and represents the process of the GA search. Stability of the convergence points determines whether they represent global or local optima and thus whether a sufficiently large GA population will find the global optima or not.

Stochastic methods of GA performance analysis are mainly concerned with GA dynamics, i.e. how a finite GA population evolves in time in the presence of population sampling errors (Mitchell, 1996). This evolution may be analysed, for instance, using Markov chains (Nix and Vose, 1991; Vose, 1993) or using the statistical-mechanics approach (Prugel-Bennett and Shapiro, 1994). In some cases stochastic methods are capable of making a very accurate prediction of how a GA population will transform from one generation to another and therefore, together with analytical methods, may help in resolving the difficulties caused by limitations of the Genetic Algorithm method.

This leaves us with an encouraging conclusion that none of the known weaknesses can completely obstruct Genetic Algorithms from successful performance –there is always one way or another to overcome the difficulties caused by a particular GA deficiency. Given all the advantages of evolutionary optimisation, it is reasonable to assume that Genetic Algorithms can outperform the methods of conventional optimisation, or at least give similar results subject to increased robustness. To verify this assumption we will apply the GA method to the problem of chemotherapy treatment optimisation. This task constitutes the essence and the main innovative aspect of the present work and will be thoroughly explored in the following two chapters.

CHAPTER 5

SINGLE-DRUG CHEMOTHERAPY OPTIMISATION USING GENETIC ALGORITHMS

This chapter starts the discussion on the innovative aspects of the present work and on the original contributions to knowledge which have been made. The major objective of this thesis is to apply the GA technique to cancer chemotherapy in order to either develop optimal treatment regimens *ab initio*, or to improve those that already exist. In this chapter we will concentrate on the optimisation of single-drug treatments.

The problem of single-drug chemotherapy optimisation can be solved by different methods. In Chapter 3 we showed that if the objective of treatment is to minimise the final tumour size, then the problem of chemotherapy optimisation becomes tractable to the linear programming method. Linear Programming guarantees that the optimal treatment regimen will be found. This can then be used for comparison with regimens obtained from Genetic Algorithms and those obtained from other competitive optimisation techniques. The latter methods will be represented by the Complex method and the Hooke and Jeeves method for reasons specified in Section 3.3.

The intent of this comparison is to assess: (1) how effective the Genetic Algorithm technique is in solving the multi-constrained optimisation problem of chemotherapy improvement, (2) whether its effectiveness is better than the effectiveness of the conventional optimisation methods, and (3) whether the GA effectiveness can be enhanced. The sections of this chapter consequently address these issues. Firstly, it will be shown in Section 5.1 how the optimal control problem of cancer chemotherapy can be formulated as a search problem suitable for Genetic Algorithms. Two approaches to the problem formulation will be considered. In the first approach, referred to as a Genetic Algorithm strategy, the goal of the GA search will be to find good treatment regimens which achieve a specified treatment objective and satisfy all the constraints. The second approach, referred to as a Regime Modification (RM) strategy, will

focus on finding the best adjustment schemes for improving already existing regimens. Also, in Section 5.1 we will particularise the most essential implementation features of Genetic Algorithms. Section 5.2 presents the comparative results of the GA and the NLP optimisation. The comparison is performed on the LP test-bed for a number of optimisation scenarios. The issue of GA efficiency will be addressed in Section 5.3 where a novel approach to GA factor tuning will be developed. The final section of this chapter, Section 5.4, will summarise the results achieved and the conclusions which can be drawn from these results.

5.1. Transforming Cancer Treatment Optimisation into a GA Problem

As was shown in the previous chapter, in order to convert the problem of cancer chemotherapy optimisation into a search task for GAs we need to specify the following components: representation of the solution space, the fitness function and the evolutionary GA procedures of initialisation, selection and recombination. Firstly we address the representation component.

5.1.1. Representation of single-drug treatment regimens

In the present work the approach, which has been adopted for chemotherapy scheduling, assumes discrete drug dosage administration over the treatment period. The acceptability of this approach has been vindicated earlier (Section 2.3 and Section 3.2); moreover, the optimisation of cancer chemotherapy has been confined to the problem of deciding which control vector $\mathbf{c} = (C_i)$, $i = \overline{1,n}$, representing a drug concentration profile, is the most suitable and effective.

Genetic Algorithms can solve this problem by implementing a multi-directional, implicitly parallel search through the solution space S of chemotherapeutic regimens. The representation space I of these solutions may be expressed as a Cartesian product

$$I = A_1 \times A_2 \times \ldots \times A_n \tag{5.1}$$

where $A_i, i = \overline{1,n}$ are the allele sets encoding the concentration levels $C_i, i = \overline{1,n}$ of a given anti-cancer drug in the blood plasma. In the present work two encoding schemes have been adopted. The first scheme, referred to as the Genetic Algorithms strategy, uses a 4-bit representation for each concentration C_i. Thus all allele sets $A_i, i = \overline{1,n}$ are identical and consist of 16 elements

$$A_i = \{0000,0001,0010,0011,\ldots,1110,1111\} \quad \forall i = \overline{1,n}$$

Such representation allows each concentration C_i to take an integer value in the range 0 to 15 and any treatment regimen $\mathbf{c} = (C_i), i = \overline{1,n}$ can be represented as a $4n$-bit chromosome $x \in I_{GA}$:

$$x = \left\{a_1 a_2 a_3 \ldots a_{4n} : a_k \in \{0,1\} \ \forall k = \overline{1,4n}\right\} \tag{5.2}$$

This approach can, if required, be applied in greater generality. Any desired concentration range $[0, C_{\max}]$ can be encoded in this way to any required degree of precision, ε, by dividing the range into 2^p-1 subintervals where $\frac{C_{\max}}{2^p - 1} < \varepsilon$. The quantity $\Delta C = \frac{C_{\max}}{2^p - 1}$ represents the smallest change in concentration level detectable by the algorithm. ΔC is referred to as a concentration unit and plays an important role in the context of GA optimisation. Similarly one can in principle divide the treatment period into as many treatment intervals as required. As this precision increases however the search space becomes exponentially larger. Specifically, with n treatment intervals and 2^p concentration levels there are 2^{np} possible treatment regimens from which to choose. However it is difficult to control drug concentration with high precision and also, treatment intervals do not in practice become arbitrarily short. In this chapter we will examine how Genetic Algorithms perform the optimisation of a single-drug treatment, which consists of 10 treatment intervals and allows the drug concentration during each interval to have 16 distinct values. This assumption giving 2^{40} possible treatment regimens is not unrealistic.

In the second encoding scheme, referred to as the Regimen Modification strategy, each chromosome $x \in I_{\mathrm{RM}}$ represents a modification of a given treatment regimen $\mathbf{c}_0 = (C_i^0), i = \overline{1,n}$. The representation space I_{RM} has the same form as (5.1), but the allele sets $A_i, i = \overline{1,n}$ are different and consist of three elements

$$A_i = \{0,1,-1\} \qquad \forall i = \overline{1,n}.$$

The elements of the allele sets represent either no change or an alteration of $\pm \Delta C$ to the associated level C_i^0. ΔC can take different values; in the present work it has been set to 1. Thus the Regimen Modification strategy yields the following form of a chromosome $x \in I_{\mathrm{RM}}$

$$x = \left\{ a_1 a_2 a_3 \ldots a_n : a_k \in \{0,1,-1\} \;\; \forall k = \overline{1,n} \right\} \qquad (5.3).$$

Having specified how chemotherapy treatment regimens can be represented, let us now proceed to the issue of their evaluation.

5.1.2. Evaluation of single-drug treatment regimens

While discussing the major concepts of Genetic Algorithms in the previous chapter, the importance of the right choice of the GA fitness function repeatedly arose. First of all, this function must be strongly related to the objective of optimisation. Secondly, all constraints of the target problem have to be taken into account in such a way that solutions that satisfy all constraints have greater fitness than those which break any constraints. Moreover, it is also desirable if solutions, which only moderately violate the problem's constraints can be distinguished by their fitness from those which break constraints to a greater degree. Finally, since the fitness function is typically calculated a large number of times it should be as simple as possible.

As was shown in Section 2.2, the Gompertz growth model, being relatively simple, is the most suitable model for the description of tumour kinetics during treatment. The fitness function $f(x), x \in I$ therefore will be specified on the basis of this model and will express the effectiveness of a given treatment regimen, encoded as a chromosome x, in achieving a particular treatment objective. In this chapter we will focus on minimising the final tumour size $N(T_{\text{final}})$ after a fixed-length treatment period $[T_0, T_{\text{final}}]$. This objective of chemotherapeutic treatment will allow us to use the LP method to obtain the optimal treatment regimen, on the basis of which the quality of the GA optimisation will be assessed.

Thus, the search task for Genetic Algorithms can now be formulated as an optimal control problem similar to the problem (3.21)-(3.23):

$$\text{Find a treatment regimen } \mathbf{c} = (C_i \mid i = \overline{1,n}) \text{ which minimises } N(T_{\text{final}}) \tag{5.4}$$

subject to the state equation

$$\dot{N}(t) = \left[\lambda \ln\left(\frac{\Theta}{N(t)}\right) - \kappa \sum_{i=1}^{n} C_i \{H(t-t_{i-1}) - H(t-t_i)\}\right] N(t), \; N(0) = N_0 \tag{5.5}$$

and the inequality constraints

$$\begin{cases} g_1(\mathbf{c}) = C_{\max} - C_i \geq 0 & \text{for all } i = \overline{1,n} \\ g_2(\mathbf{c}) = C_{\text{cum}} - \sum_{i=1}^{n} C_i \geq 0 & \\ g_3(\mathbf{c}) = N_{\max} - N(t_i) \geq 0 & \text{for all } i = \overline{1,n} \end{cases} \tag{5.6}$$

(N.B. The last constraint utilises the result of Lemma 3.1.) The substitution $y(t) = \ln\left(\frac{\Theta}{N(t)}\right)$ simplifies the state equation (5.5) as:

$$\dot{y}(t) + \lambda y(t) = \kappa \sum_{i=1}^{n} C_i \{H(t-t_{i-1}) - H(t-t_i)\}$$

and yields its analytical solution

$$y(t_p) = e^{-\lambda t_p}\left\{y_0 + \frac{\kappa}{\lambda}\left(e^{\lambda \Delta t} - 1\right)\sum_{i=1}^{p} C_i e^{\lambda t_{i-1}}\right\} \tag{5.7}$$

where $p = 0,1,2,\ldots,n.$

Remembering that $N(T_{\text{final}})$ is minimised when $y(T_{\text{final}})$ is maximised, we can now formulate the objective functional for the problem (5.4)-(5.6):

$$\text{Maximise } J(\mathbf{c}) = \frac{y(T_{\text{final}}) - y_0 e^{-\lambda T_{\text{final}}}}{\frac{\kappa}{\lambda}\left(e^{\lambda \Delta t} - 1\right)} = \sum_{i=1}^{n} C_i e^{\lambda(t_{i-1} - T_{\text{final}})} \tag{5.8}$$

$$\text{where } y_0 = \ln\left(\frac{\Theta}{N_0}\right)$$

To obtain a GA fitness function from the optimisation functional (5.8), the constraints (5.6) need to be taken into account. The incorporation of the constraints into the fitness function is implemented via the distance-based penalty functions of the form:

$$p_j(g_j(\mathbf{c})) = P_j(t) d_j^k, j = \overline{1,m}$$

where d_j is the distance between a given solution $\mathbf{c}$ and the feasible region corresponding to the constraint $g_j(\mathbf{c})$; k is the power to which d_j is raised to regulate the penalising effect; $P_j(t)$ are the penalty multipliers and m is the number of constraints. In Section 3.3 it has been decided to use constant penalty multipliers with the toxicity constraints ($g_1(\mathbf{c})$ and $g_2(\mathbf{c})$) and an exponentially decreasing penalty multiplier with the tumour size constraint $g_3(\mathbf{c})$ (see (5.6)). The distance metrics were also defined there (see (3.32)-(3.34)). Thus using (5.8), we can now build an augmented objective functional which incorporates the constraints (5.6) of single-drug chemotherapy treatment:

$$\begin{aligned} \tilde{J}(\mathbf{c}) = \sum_{i=1}^{n} C_i e^{\lambda(t_{i-1} - T_{\text{final}})} - P_1 \sum_{i=1}^{n} \max^2\{C_i - C_{\max}, 0\} \\ - P_2 \max^2\left\{\sum_{i=1}^{n} C_i - C_{\text{cum}}, 0\right\} - P_3 \sum_{i=1}^{n} e^{-\beta \cdot t_i} \max^2\{y_{\min} - y(t_i), 0\} \end{aligned} \tag{5.9}$$

where $y_{\min} = \ln\left(\frac{\Theta}{N_{\max}}\right)$, and $y(t_i)$ can be determined from the equation (5.7).

The encoding schemes of the Genetic Algorithm (5.2) and the Regimen Modification (5.3) strategies provide the following representations of the concentration levels C_i:

$$\text{GA}: \qquad C_i = \Delta C \sum_{k=1}^{4} 2^{4-k} a_{4(i-1)+k}, \quad i = \overline{1,n} \tag{5.10}$$

$$\text{RM}: \qquad C_i = C_i^0 + a_i \Delta C, \quad i = \overline{1,n} \tag{5.11}$$

The substitutions of (5.10) and (5.11) into the augmented objective functional (5.9) yield the fitness functions $f_{\text{GA}}(x)$, $x \in I_{\text{GA}}$ and $f_{\text{RM}}(x)$, $x \in I_{\text{RM}}$ respectively, the full expressions of which are given in Appendix A1. These functions are used for evaluation of candidate treatment regimens encoded as chromosomes in the representation spaces I_{GA} and I_{RM}.

5.1.3. Evolutionary procedures of GAs for single-drug treatments

Having encoded different treatment regimens as chromosomes and having specified the fitness function to evaluate them, we now need to particularise the evolutionary procedures (population initialisation, chromosome selection and recombination) in order to start the GA search. Initialisation is usually implemented at random (*random start*). However, as we mentioned in Section 4.4.3 on the hybridisation of Genetic Algorithms, sometimes it is possible to obtain a few satisfactory solutions to the problem before the GA search begins. These solutions can be seeded into an initial GA population (the technique known as pre-hybridisation) and the GA search gets a *feasible start*. In the next section we will examine how the starting condition affects the GA performance.

The selection procedure adopted in the present work is based on combining roulette-wheel selection with linear fitness normalisation (see Section 4.3.2). Fitness normalisation is used to protect the method of roulette-wheel selection against misleading guidance of 'super'-chromosomes, which do not represent optimal solutions but whose fitness is vastly greater than those of the rest of the chromosomes in the GA population. The presence of 'super'-chromosomes often leads to the problem of premature convergence; fitness normalisation schemes are used to circumvent this problem by expanding the effective range of fitness in the GA population. The effective fitness range is expanded by sorting all chromosomes in order of raw fitness and then assigning normalised fitness to each chromosome according to its rank. The linear normalisation scheme is characterised by two constants, viz. the *normalisation maximum*, i.e. the maximum fitness value assigned to the best chromosome, and the *normalisation slope*, which determines how fast the normalised fitness of a chromosome decreases with a decrease in the chromosome's rank. These constants can be varied to regulate the selection pressure of Genetic Algorithms. Also, due to a randomised selection of chromosomes for recombination, it is possible that the best chromosome might not be chosen. To prevent a potential source of loss, an elitist strategy of chromosome selection is used whereby the single best chromosome from each generation is copied into the succeeding generation. Finally, the procedure of chromosome recombination is composed of two genetic operators – two-point crossover and bit-wise mutation.

In order to exemplify the search process for better chemotherapy regimens using Genetic Algorithms let us assume that we are trying to optimise a 10-dose chemotherapeutic schedule of Adriamycin with MTD $C_{\max} = 75 \ \frac{\text{mg}}{\text{m}^2}$, maximum cumulative dose $C_{\text{cum}} = 550 \ \frac{\text{mg}}{\text{m}^2}$, and the effectiveness value $\kappa = 5.605 \cdot 10^{-3}$. The 4-bit representation scheme (see Section 5.1.1) allows each concentration C_i to take an integer value in the range 0 to 15 concentration units, which makes the value of one unit equal to $\Delta C = \frac{75}{5} = 5 \ \frac{\text{mg}}{\text{m}^2}$. Thus the maximum cumulative

dose for Adriamycin is limited to $\frac{550}{5} = 110$ concentration units. The mapping between a randomly generated GA chromosome and the corresponding treatment schedule can be seen from the following table:

TABLE 5.1. *Chromosome-treatment correspondence*

GA chromosome	0101	1000	1001	0011	0101	1011	1110	1001	0111	1111
Concentration units	5	8	9	3	5	11	14	9	7	15
Treatment regimen $\frac{mg}{m^2}$	25	40	45	15	25	55	70	45	35	75

The number of concentration units in Table 5.1 is an integer representation of the binary code of the given GA chromosome (divided in ten 4-bit portions for the reader's convenience). Knowing the numerical value of one concentration unit ΔC, it is possible to specify the values of the drug doses $C_i, i = \overline{1,10}$, which constitute the actual treatment regimen corresponding to our GA chromosome. Substituting these doses into the formula (5.9), we obtain a quantitative characteristic (fitness value) of the chromosome's quality and as such the quantification of the merit of the corresponding treatment regimen. On the basis of their fitness values, all chromosomes in the GA population will be assigned the probabilities of being selected for reproduction – good chromosomes will get a better chance of being passed into the next generation. Then the roulette-wheel selection determines which chromosomes will constitute the next generation. Each selected chromosome undergoes with certain probabilities the recombination procedures of mutation and crossover. Figure 5.1 illustrates the mutation process of a chemotherapy schedule:

Old chromosome	0101	1000	1001	0011	0101	1011	1110	1001	0111	1111
Old treatment	**25**	40	45	**15**	25	55	**70**	45	35	75

↓

Mutation

↓

New chromosome	0111	1000	1001	0001	0101	1011	1100	1001	0111	1111
New treatment	**35**	40	45	**5**	25	55	**60**	45	35	75

Figure 5.1. Schedule mutation

In Figure 5.1 the chromosome alleles which have undergone mutation are shown in bold. The resultant changes in dose values are also emphasised.

To illustrate the effect of crossover we will need two chromosomes, and the 2-point crossover operator will be used in order to make the illustration simple and concise at the same time. The crossing points are chosen in such a way that the first one falls in between the encoding of two constituent doses, whereas the second point splits the encoding of a single dose. Figure 5.2 illustrates the features of crossover recombination.

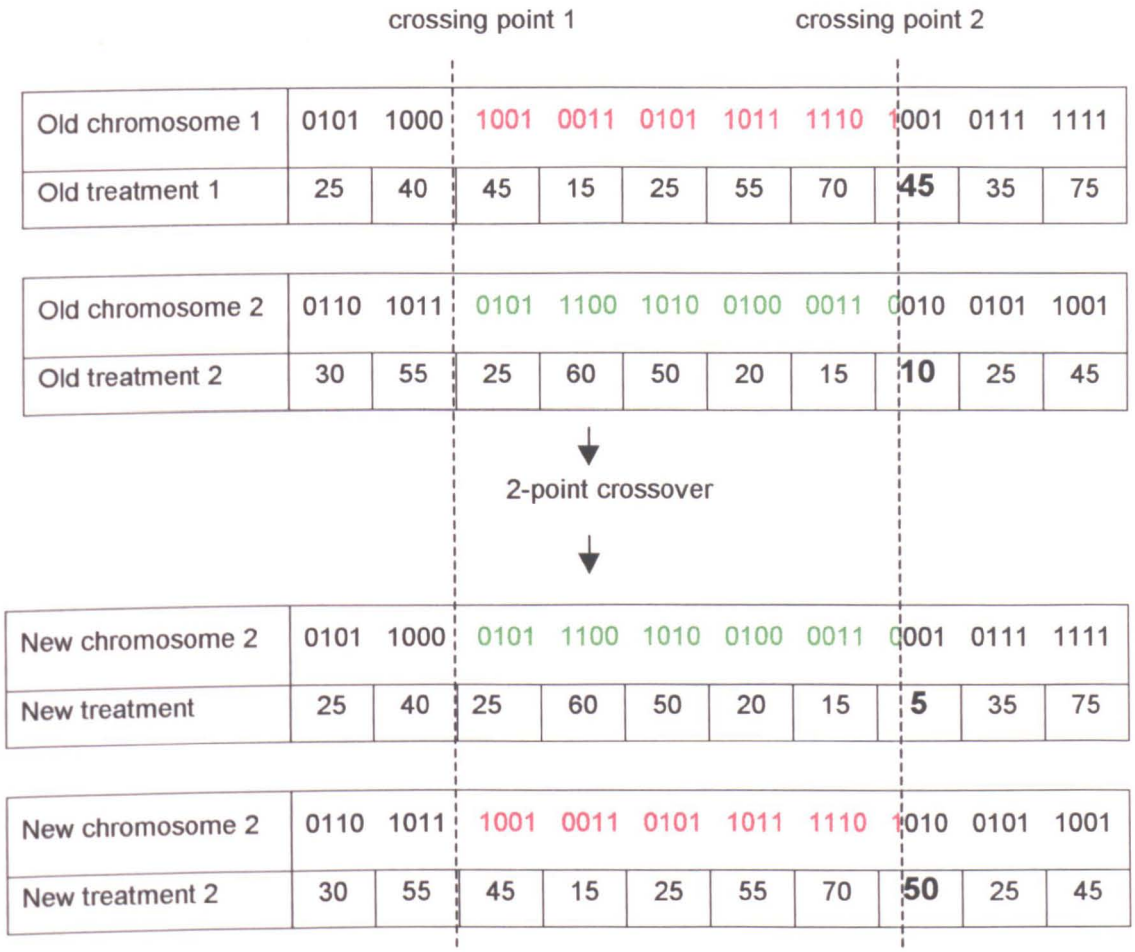


Figure 5.2. Crossover of two chemotherapy schedules

As may be seen from this figure, the effect of crossover on chemotherapy regimens is twofold. Firstly, the main task of the crossover operator in the evolutionary search of Genetic Algorithms is to assemble better solutions from promising portions (building blocks) of the current solutions. Figure 5.2 depicts this assembling process of new chemotherapy regimens. Secondly, by allowing a crossing point to split the representation of a single dose, it is possible to achieve the effect of mutation of the dose in question (see the effect of the 2nd crossing point). Thus both genetic operators, i.e. mutation and crossover, are responsible for introducing and exploring new values of drug doses which constitute chemotherapy treatment. Moreover, together with biased selection, the crossover operator is actively involved in the exploitation of useful

information accumulated during the GA search in the form of the portions of chemotherapy schedules accountable for the treatment success. By assembling these portions together in one treatment regimen the objective of chemotherapy optimisation can be achieved.

The crossover and mutation operators are applied with the probabilities p_c and p_m respectively. For conducting the comparative study, the results of which are presented in the next section, these probabilities were assigned the following values $p_c = 0.5$ and $p_m = 0.15$. These values have been found empirically by the author during his formative experiments with Genetic Algorithms. A more rational approach to specification of the crossover and mutation probabilities, as well as other GA factors, will be developed later in this chapter (see Section 5.3). However, what needs to be said here is that by adjusting these probabilities one may achieve the equivalence of different GA implementations. For instance, if we introduce a constraint on the crossover operator which will allow crossover to only juxtapose different portions of various treatment regimens without the ability to modify the values of drug doses, then the GA implementation will differ from the previously described. Yet one may argue that by tuning the values of the mutation and crossover probabilities, the effects of these operators on the performance of Genetic Algorithms can be made essentially the same. One practical implication of this conclusion is that different implementations of GA evolutionary procedures achieve essentially the same result as long as the right balance between the exploitative and explorative features of GA search is maintained.

Finally, before we embark upon the description of what has been achieved in the course of the present work, a few words need to be said about the implementation of Genetic Algorithms. The GA programs for optimising single-drug chemotherapy treatments and for comparing Genetic Algorithms with other heuristic optimisation techniques were written in Pascal and run on both Windows (WIN32 and Windows NT) and Unix (SunOS 4.1.4 and Solaris 2.4) platforms. The GA program for optimising multi-drug chemotherapies were written in the C++ language in order to ease the process of embedding the GA optimisation component into a more general decision support system implemented in JAVA. To develop, edit and debug the GA code the author used the Borland C++ (v. 5.02) environment. The latest version of the code is ~600 lines long and the size of its executable file is 85K. A Kolmogorov-Smirnov test has been carried out to ensure the randomness of the numbers produced by the random number generator, which is used by the GA program. The results of this test confirm the appropriateness of the random number generator ($\left| F_{OBS}(x) - F_{EXP}(x) \right|_{max} = 0.12$ which is significantly less than the rejection number $F_{CRIT,\alpha=0.05} = 0.26$) and complete the preparation work for commencing the GA optimisation of chemotherapy treatments. In the next section the single-drug case of this optimisation will be addressed.

5.2. GA Optimisation of single-drug treatments

When the objective of optimisation is to minimise tumour size by the end of treatment, the optimisation problem of single-drug chemotherapy is amenable to the solution by the optimal control method (Martin *et al*, 1990) and the linear programming technique (see Section 3.3.1). Martin *et al* (1990) found the optimal single-drug regimens for two case studies, which were based on the model (3.22) of tumour kinetics. In the first case, the partly constrained problem (where only toxicity constraints on the maximum instantaneous and the cumulative doses were present) was addressed. The second case was concerned with the fully constrained problem (where the tumour size constraint was also considered). In the present work, the partly constrained problem was used as the first test for Genetic Algorithms. The purpose of this test was to ascertain how quickly and reliably GAs can find an optimal solution in a relatively simple case. The result of the GA optimisation was compared with that of the Martin's *et al* study and turned out to be very satisfactory (Petrovski *et al*, 1998a).

The performance of Genetic Algorithms on the fully-constrained problem has been studied on the test-bed provided by the linear programming formulation of the single-drug chemotherapy optimisation problem. As was shown in Section 3.3.1, if an optimisation problem can be formulated as an LP problem with a non-empty feasibility region, then the method of Linear Programming will guarantee the finding of the optimal solution to this problem. The global optimum obtained from the LP technique is used as a target for other optimisation methods to attain. The purpose of the current section is to show how Genetic Algorithms perform this task in comparison with the Complex and the Hooke and Jeeves methods, which have been selected (see Section 3.3) to represent the traditional methods of non-linear constrained optimisation.

The comparison will be made for a number of situations, which, being practically meaningful, serve to illustrate the strengths and weaknesses of the methods. We distinguish between situations where some initial information is supplied to the implementation algorithms in terms of a feasible, but not optimal, solution and the situation where no information is supplied. Moreover, in cases where a feasible start is given to the algorithms, differences in performance occur when the problem's parameters are altered so that the set of feasible solutions increases or decreases. The analysis of this performance will be provided later; now let us describe the experimental results obtained from the optimisation algorithms started at random.

5.2.1. Random start

The efficiency of numerical optimisation methods is dependent to some degree on the starting condition. This dependence varies from one method to another; a general rule applied here is that the methods, which are less dependent on the starting condition of optimisation, are considered to be more robust. The first experimental trial shall therefore ascertain how different optimisation techniques can handle the paucity of *a priori* knowledge about the location of the desirable region. The simplest way to do this is to give the implementation algorithms a random

start and to examine how far they can progress towards the optimal solution. In all cases where there is a non-deterministic outcome, the implementation algorithms are run several times and the best results obtained by each optimisation method are compared.

The optimal solution will be produced by the LP method applied to the following single-drug chemotherapy problem:

> Minimise the size of the tumour after administration of 10 discrete doses, resulting in a piece-wise constant concentration profile $\mathbf{c} = (C_i), i = \overline{1,10}$ subject to the restricting conditions
>
> $$\begin{cases} N_0 = N_{\max} \\ \sum_{i=1}^{10} C_{\text{hold}}(t_i) = C_{\text{cum}} \end{cases} \tag{5.12}$$
>
> where $0 \le C_i \le C_{\max} = 15, \ \forall i = \overline{1,10}$ and C_{hold} is the drug concentration necessary to maintain the tumour at the maximum permissible level. N.B. The maximum instantaneous concentration $C_{\max}$ is equal to 15 units due to a 4-bit representation of GA chromosomes (see the previous section).

Since $N_0 = N_{\max}$ we are forced to start chemotherapeutic treatment without any delay because the tumour is growing and without an immediate drug delivery it will exceed $N_{\max}$. The rate of tumour growth λ and the effectiveness of the drug κ have been chosen in such a way that the concentration of 10 units ($C_{\text{hold}} = 10$) will be sufficient to hold the tumour at the level $N_{\max} = N_0$. Every dose which results in concentration less than 10 units will be insufficient to satisfy the tumour size constraint $N(t) \le N_{\max}$. The maximum cumulative dose C_{cum} is chosen to permit the administration of only necessary drug doses (plus a couple of extra drug units to allow for the rounding error of the execution program). This means that the overall drug allowance C_{cum} will be slightly more than 100 units, which will have to be tightly disseminated among the concentrations $C_i, i = \overline{1,10}$.

Such choice of the parameters N_0 and C_{cum} is aimed at allowing the optimisation problem under investigation to have only a handful of feasible solutions. The task of the numerical optimisation techniques (GAs, CM and HJ) is to find a feasible solution given a random start. This means that the initial population of Genetic Algorithms is randomly generated. Similarly the initial complex $\mathbf{C}_k, k = \overline{1,11}$ for CM is randomly generated and a random initial vector $\mathbf{C}_1$ is generated for HJ. (NOTE. The Regimen Modification strategy was excluded from this series of experiments since the rationale behind RM is that it will be used to optimally improve known feasible regimens, as opposed to randomly generated ones which may be infeasible).

Under these conditions in general CM and HJ performed poorly, in all cases generating final solutions lying in the infeasible region. Various values of the penalty coefficients P_2 and P_3 for

the C_{cum} and N_{max} constraints respectively were used in successive runs. (NOTE. The penalty coefficient P_1 is reserved for the maximum instantaneous dose constraint. The GA strategy of single-drug chemotherapy optimisation implicitly takes this constraint into account through the encoding procedure. However other algorithms will need an explicit formulation of the C_{max} constraint and the penalty P_1 is reserved for this purpose.) Best results were obtained for both CM and HJ when one of the penalty coefficients was set to a very low value (10 as opposed to the high value of 10^6). This approximates the two simpler one-constraint problems, defined by relaxing either the N_{max} or C_{cum} constraint. In these cases HJ and CM arrived at solutions which satisfied the more strictly enforced constraint. However the second constraint was violated in all cases. When the values of the penalty coefficients were close together CM and HJ consistently produced final solutions which failed to satisfy either constraint. GAs proved robust in the sense that, under all conditions, a feasible solution was produced. Table 5.2 compares the best results obtained by each algorithm under a random start against the optimal LP solution on the basis of the ratio of the final and the initial tumour sizes:

TABLE 5.2. *Algorithms' performance given a random start*

Algorithm	$\frac{N(T_{final})}{N_0}$	C_{cum} constraint	N_{max} constraint
LP	0.88	Satisfied	Satisfied
GA	0.95	Constraint satisfied for all values of P_2 and P_3	Constraint satisfied for all values of P_2 and P_3
CM	0.96	Constraint violated P_2=10	Constraint satisfied P_3=10^6
HJ	1.03	Constraint satisfied P_2=10^6	Constraint violated. P_3=10

5.2.2. Feasible start

The second situation to consider is the case where the searching algorithms are given initial information in the form of a feasible solution. In this situation, irrespective of penalty coefficient values, all algorithms end up with feasible solutions. However the relative performance of the algorithms depends on the size of the set of feasible solutions. This set, referred to as the feasible region in the solution space S, represents a search domain for the algorithms; the larger its size the more solutions are present to choose from. One way to vary the size of the feasible region is through the alteration of the initial tumour size N_0 subject to the same values of the problem parameters and constraint boundaries. Two experimental situations will be examined here, the particulars of which are now provided.

First we consider the situation when $N_0 = N_{\max}$, i.e. the problem is identical to that addressed in the previous subsection. The number of feasible solutions to this problem is very small due to the tightness of the cumulative constraint boundary C_{cum}. One feasible solution is $C_i = C_{\text{hold}}$ (equal to 10 units) for $\forall i = \overline{1,10}$. This solution was used as an initial vector $\mathbf{C}_1$ to start HJ and as one of the initial vertices for CM; in the case of GAs, the initial population was seeded with a few copies of this solution. Also, the RM optimisation algorithm was used in the experiments with feasible starting conditions and was given the solution $C_i = C_{\text{hold}}, \forall i = \overline{1,10}$ as a starting regimen for further improvement. The results obtained from the comparison of LP, GAs, RM, CM and HJ on the first experimental problem are presented in Table 5.3 and Figure 5.3.

TABLE 5.3. *Algorithms' performance with* $N_0 = N_{\max}$

Algorithm	$\frac{N(T_{\text{final}})}{N_0}$
LP	0.88
RM	0.88
CM	0.9
GA	0.95
HJ	0.95

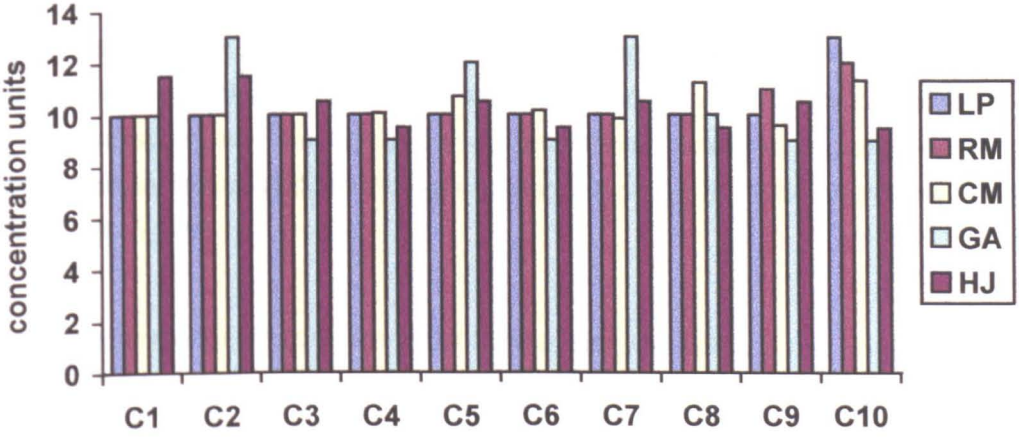


Figure 5.3. Regimens resulting from feasible start with $N_0 = N_{\max}$

These results indicate that the Regimen Modification strategy is the best performer with respect to minimisation of the final tumour size. The concentration profile obtained from RM is very close to optimal and differs from LP by only one delta change. Table 5.4 gives an extended summary of the comparison between LP, RM and GA based on the results presented in Table 5.3 and in Figure 5.3:

TABLE 5.4. *Analysis of optimal solutions for the case* $N_0 = N_{max}$

	Treatment interval	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
LP	Concentr. units	10	10	10	10	10	10	10	10	10	13
	Drug dose, $\frac{mg}{m^2}$	50	50	50	50	50	50	50	50	50	65
	Final tumour reduction =12%										
RM	Concentr. units	10	10	10	10	10	10	10	10	11	12
	Drug dose, $\frac{mg}{m^2}$	50	50	50	50	50	50	50	50	55	60
	Final tumour reduction = 12%										
GA	Concentr. units	10	13	9	9	12	9	13	10	9	9
	Drug dose, $\frac{mg}{m^2}$	50	65	45	45	60	45	65	50	45	45
	Final tumour reduction = 5%										

Analysing the results obtained from the traditional optimisation methods, we can see that CM performs creditably, whereas HJ, similarly to GAs, finds a treatment regimen which while feasible is far from optimal. It was noticeable that GAs did not appear to be significantly assisted by the given solution in terms of the best solution eventually produced. Also, GAs, as well as RM, were much slower compared to the other methods. They needed 15-25 minutes to complete the search compared to less than a minute of computational time required by LP, CM and HJ. Furthermore, the time spent by Genetic Algorithms in search for the best solution significantly varied depending on the values of the GA factors such as the probabilities of mutation and crossover, the overall number of generations, selection pressure, etc. The specification of these factors and the analysis of their effects on the GA performance will be discussed in Section 5.3.2.

In the second experimental situation the initial tumour size was set to a half of the maximum limit, i.e. $N_0 = 0.5N_{max}$. Thus the tumour was allowed to double in size and still be within the acceptable boundaries. This might be realistic if the tumour is very small or perhaps suspected but not observed. Since the drug allowance remains the same as in the previous experimental situation, there is a higher number of feasible solutions. When $N_0 = 0.5N_{max}$, it is possible to postpone the commencement of treatment, thus allowing delivery of the late treatment doses at a higher rate in order to minimise the final tumour size. The results of this minimisation performed by the contesting methods are tabulated in Table 5.5 whereas the best treatment regimens obtained form each optimisation method are presented in Figure 5.4.

TABLE 5.5. *Algorithms' performance with* $N_0 = 0.5N_{\max}$

Algorithm	$\frac{N(T_{\text{final}})}{N_0}$
LP	0.65
RM	0.66
GA	0.7
CM	0.72
HJ	0.95

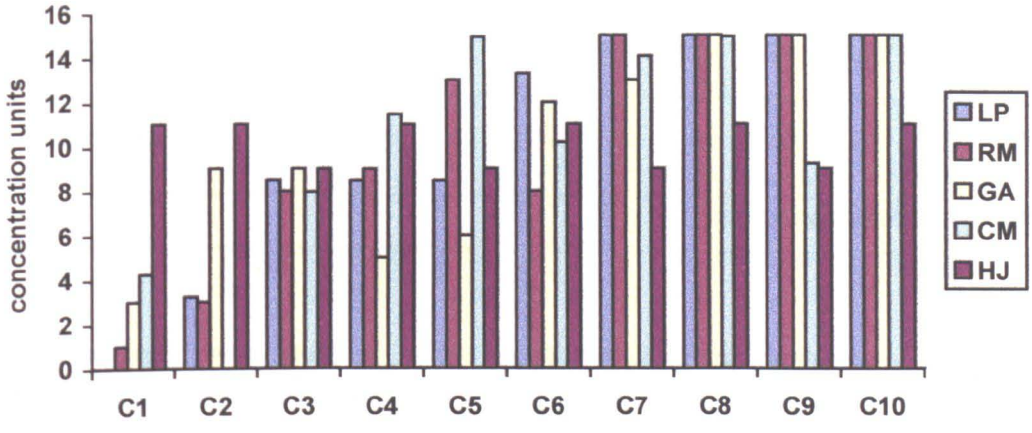


Figure 5.4. Regimens resulting from feasible start with $N_0 = 0.5N_{\max}$

The latter experiment shows that RM is still the best method as far as minimisation of tumor size is concerned. The relative performance of GAs and CM change around compared to the previous experiment; HJ performs very poorly, only achieving the same tumour size reduction as in the first case. Similarly to the previous experiment, we summarise the comparative results of LP, RM and GA in Table 5.6 (overleaf).

TABLE 5.6. *Analysis of optimal solutions for the case* $N_0 = 0.5N_{\max}$

	Treatment interval	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
LP	Concentr. units	0	3	8	8	8	13	15	15	15	15
	Drug dose, $\frac{\text{mg}}{\text{m}^2}$	0	15	40	40	40	65	75	75	75	75
	Final tumour reduction = 35%										
RM	Concentr. units	1	3	8	8	13	8	15	15	15	15
	Drug dose, $\frac{\text{mg}}{\text{m}^2}$	5	15	40	40	65	40	75	75	75	75
	Final tumour reduction = 34%										
GA	Concentr. units	3	9	9	5	6	12	13	15	15	15
	Drug dose, $\frac{\text{mg}}{\text{m}^2}$	15	45	45	25	30	60	65	75	75	75
	Final tumour reduction = 30%										

To complete the experimental study of Genetic Algorithms within the framework of single-drug chemotherapy optimisation, we will consider an optimisation problem which is not tractable to Linear Programming but is solvable by Genetic Algorithms. In the next section one such problem is addressed.

5.2.3. Prolongation of the patient survival time

In cases when the tumour is deemed incurable the main objective of chemotherapy treatment is to maintain a reasonable quality of life for as long as possible. One way to mathematically express this objective is to maximise PST:

$$\underset{\mathbf{c}}{\text{maximise}} \quad T \quad \text{such that} \quad N(T) = N_{\max}$$

subject to the state equation (5.5) and the constraints (5.6). As has been shown in Section 5.1.2, after the substitution $y = \ln\left(\frac{\Theta}{N}\right)$ the state equation (5.5) becomes:

$$y(t) = e^{-\lambda t}\left\{y_0 + \frac{\kappa}{\lambda}\left(e^{\lambda\Delta t} - 1\right)\sum_{i=1}^{n} C_i e^{\lambda t_{i-1}}\right\}.$$

Since the condition $N(T) = N_{\max}$ is equivalent to $y(T) = y_{\min} = \ln\left(\frac{\Theta}{N_{\max}}\right)$, we can express T as follows:

$$y_{\min} = e^{-\lambda T}\left\{ y_0 + \frac{\kappa}{\lambda}\left(e^{\lambda\Delta t} - 1\right)\sum_{i=1}^{n} C_i e^{\lambda t_{i-1}} \right\}$$

$$e^{\lambda T} = \frac{y_o}{y_{\min}} + \frac{\kappa}{\lambda \cdot y_{\min}}\left(e^{\lambda\Delta t} - 1\right)\sum_{i=1}^{n} C_i e^{\lambda t_{i-1}}$$

$$T = \frac{1}{\lambda}\ln\left[\frac{y_o}{y_{\min}} + \frac{\kappa}{\lambda \cdot y_{\min}}\left(e^{\lambda\Delta t} - 1\right)\sum_{i=1}^{n} C_i e^{\lambda t_{i-1}} \right]$$

Excluding from the last expression all the terms that do not depend on the control variables $C_i, i = \overline{1,10}$, we obtain the following non-linear objective functional:

$$\text{maximise } J(\mathbf{c}) = \ln\left[\sum_{i=1}^{n} C_i e^{\lambda t_{i-1}} \right]$$

Thus the problem of PST prolongation is not tractable to the linear programming approach. On the other hand, all what is required to apply Genetic Algorithms to it is to change the first term in (5.9) from $\sum_{i=1}^{n} C_i e^{\lambda(t_{i-1} - T_{\text{final}})}$ to $\ln\left[\sum_{i=1}^{n} C_i e^{\lambda t_{i-1}} \right]$. Having made this change in the GA program used in the experiments with random start (see Section 5.2.1) and having run the program for 5000 generations, the following solution has been obtained:

TABLE 5.7. *Optimal single-drug treatment with respect to PST prolongation*

Treatment interval	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Concentration units	15	12	8	3	8	6	6	13	6	15
Drug dose, $\frac{\text{mg}}{\text{m}^2}$	75	60	40	15	40	30	30	65	30	75
Patient survival time = 33 treatment intervals										

The treatment regimen presented in Table 5.7 does not break any of the constraints (5.6) and achieves the PST value of 33 treatment intervals. Recall that a treatment interval is the period during which the concentration of drug in the blood plasma remains the same. In the present work we will not specify the duration of treatment intervals in order to make the GA approach to chemotherapy optimisation as general as possible. However, to give the reader an idea of how long it may be, we can say that practicing oncologists usually use weekly treatment intervals in their chemotherapy schedules (Dearnaley *et al*, 1995).

Therefore, combining all presented experimental cases together, one can see that the methods of evolutionary optimisation (GAs and RM) either outperform the conventional numerical optimisation methods with respect to robustness and effectiveness, or give comparable results. The only drawback of the former methods is that they tend to take more time to perform an optimisation task (in the experimental studies presented above, RM and GAs needed approximately 10-20 minutes of computational time as opposed to less than a minute for CM

and HJ). However, the performance of Genetic Algorithms can be significantly enhanced, resulting in noticeable reduction of the computational time required by GAs or RM to find a reasonably good solution. The next section presents a novel approach to the enhancement of Genetic Algorithms.

5.3. Enhancement of GA Performance

Having decided to use Genetic Algorithms for solving a search/optimisation problem, one is faced with the task of specifying the setting of GA factors which are likely to produce the best results. The experiments presented in Section 5.2 illustrated that the proper choice of these factors is crucial to the success of evolutionary optimisation. Moreover, bearing in mind that real-life chemotherapy treatment usually utilises more than one drug and therefore makes the optimisation problem under investigation more complex, the dependence of the GA efficiency on the choice of the factor setting becomes more marked. Thus, before attempting to solve the multi-drug chemotherapy optimisation problem using the evolutionary approach, we need to examine mechanisms for tuning the values of GA factors.

These mechanisms may be based on: (1) systematic evaluation of various factor values; (2) adopting the factor setting applied in similar application scenarios; (3) theoretical analysis of GA factors. The optimal values of GA factors are determined by the properties of the representation space I, by the size of the GA population and by the types of the evolutionary procedures used. Due to a vast diversity of representation spaces and due to a large number of feasible evolutionary procedures, a universally valid method of GA factor tuning does not exist (Freisleben, 1997). Also, the large number of possibilities precludes an exhaustive search of the space of operators and operator probabilities thereby advocating different approaches to GA factor adjustment. Several proposals and empirical studies have been made in the literature to establish these approaches. We discussed them in detail in Section 4.3.1, having indicated their merits and shortcomings. The outcome of that discussion was that the majority of GA factor tuning methods concentrate on selecting a champion from a limited set of factor settings. Such selection is incomplete and does not guarantee the global optimality of factor values.

Therefore, in this section a new technique for GA factor tuning will be introduced, which has a potential to gain information about the global behaviour of Genetic Algorithms and thus about the optimality of GA factors. This technique is based on regression modelling of GA performance in terms of the factors that significantly affect this performance. The regression modelling is implemented using the methods of experimental design and statistical inference, application of which in the present context comprises one of the innovative aspects of this thesis. The details and the benefits of the statistical approach to GA factor tuning are explained in the following subsections. These subsection are aimed at: (1) finding the most appropriate measure of GA performance (Subsection 5.3.1); (2) developing a systematic approach based on statistical inference for tuning the values of GA factors (Subsection 5.3.2); (3) confirming that

the new approach is statistically reliable and is superior to the known methods of GA factorisation (Subsection 5.3.3). Firstly we will specify the measures of GA performance.

5.3.1. Measures of GA performance

In order to analyse the performance of an optimisation method, a certain measure of its efficiency needs to be introduced. In the case of Genetic Algorithms, such efficiency will be characterised by the number of GA generations, hereafter denoted as Ψ, which are required to find a feasible solution. A feasible solution will be sought to the LP test-bed problem defined in the previous section (see (5.12)).

There are a number of reasons why Ψ is the most convenient characteristic of GA efficiency. First of all, as the experiments with a random start illustrated (Section 5.2.1), to find the feasible region in the solution space is often the most difficult part of the search for optimum. The ability to consistently accomplish this task puts Genetic Algorithms in a better position compared to CM and HJ; once the feasible region is found, the difference in performance of the selected methods becomes less radical. However, another important issue arises here, that is how fast GAs can possibly be in finding a feasible solution. The number of generations Ψ specifically addresses this issue and as such adequately characterises the efficiency of GAs. Secondly, from the implementation point of view, it is a very straightforward task to detect the presence of a feasible solution in the GA population and count the number of prior generations which led to the appearance of this solution. Moreover, once a feasible solution has been found, thereby specifying Ψ, the program running Genetic Algorithms can be stopped. This substantially reduces the computational time required to run a large number of statistical experiments. These reasons justify the choice of Ψ as a characteristic of GA efficiency.

Due to the fact that Genetic Algorithms implement the search in a randomised fashion, the number of generations Ψ is a random variable. Random variables are described by probability distributions; each variable has its own distribution. However, it is often possible to approximate the distribution of a particular random variable by a known statistical distribution which is chosen depending on: (1) the nature of the problem; (2) the underlying assumptions associated with the distribution; (3) the shape of the graph obtained from plotting the available data; and (4) the convenience and simplicity afforded by the approximating distribution. To find a suitable approximation means to be able to adjust the parameters of a known distribution in such a way that the distribution curve will fit the experimental data obtained for a given random variable. The adjustable parameters for some common distributions are as follows: for the uniform distribution – lower and upper endpoints; for the normal distribution – the mean value and the standard deviation; for the exponential distribution – the mean; for the Gamma, the Beta and the Weibull distributions – the shape α and the scale β parameters. If certain parameter values of a particular common distribution provide the approximating curve which satisfies an

appropriate goodness-of-fit test, then a given random variable is said to have this particular distribution and may be characterised by its parameters.

To obtain a distribution list for the random variable Ψ 50 independent runs were performed of the GA program, which implements the search for a feasible solution to the problem (5.12). These runs produced a positively skewed distribution, for approximation of which many common statistical distributions have been tried. The attempt to fit the Weibull distribution produced the best result. Figure 5.5 gives a graphical illustration of the quality of this approximation.

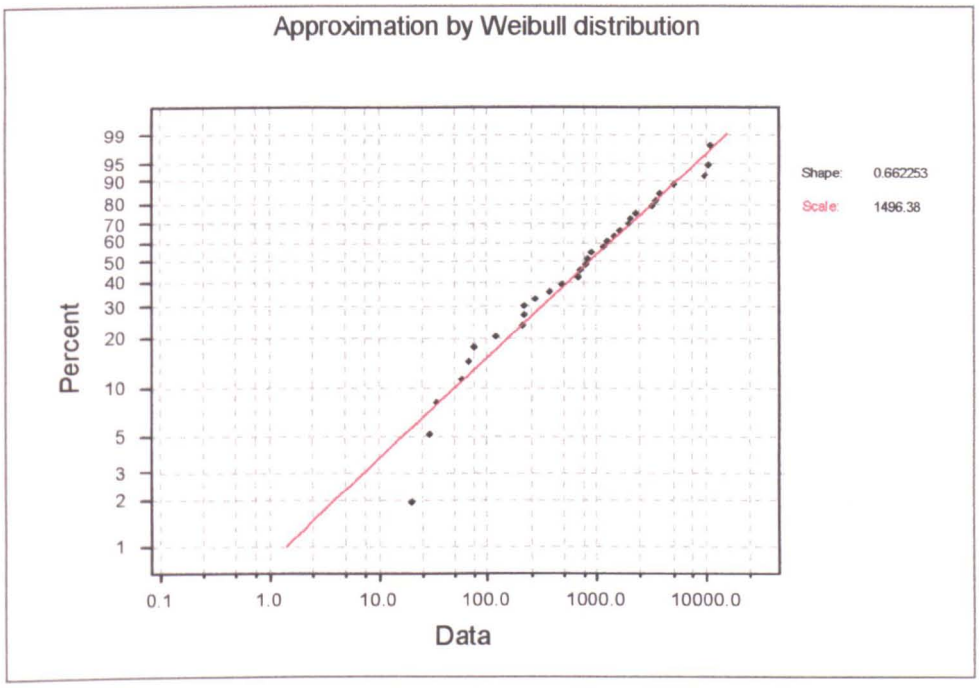


Figure 5.5. Distribution of the measuring variable Ψ

Quantitatively the approximation of the distribution is characterised by the Kolmogorov-Smirnov goodness-of-fit test. The maximum difference between the observed value of the cumulative distribution function for Ψ and the expected value is **max** $|F_{OBS}(x) - F_{EXP}(x)| = 0.0512$, which is significantly less than the rejection number ($F_{CRIT} = 0.352$). Therefore, the Kolmogorov-Smirnov test is safely passed and the parameter values of the Weibull distribution (the shape parameter $\alpha = 0.6623$ and the scale parameter $\beta = 1496.38$) can be used to characterise the random variable Ψ. The scale parameter β is especially suitable for this purpose since it is closely related to the mean value of Ψ (Devore, 1995), i.e. $\mu_\Psi = \beta \cdot \Gamma\left(1 + \alpha^{-1}\right)$ where Γ is the Gamma function. (For more detailed description of the Weibull distribution see Appendix B0.) The mean value μ_Ψ indicates the average number of generations required for a given GA program to find a feasible solution and hence characterises the efficiency of Genetic Algorithms.

Yet it is not very convenient to use the mean value of Ψ since available statistical packages do not directly estimate μ_{Ψ}. Instead, they provide the value of β, which hereafter will be used as the main measure of GA efficiency.

Although the Weibull distribution fits the experimental data very well, it is not the most convenient distribution to deal with. Some of the facilities of statistical inference are valid only under the assumption that the approximating distribution is normal. The logarithmic transformation of the random variable Ψ, $\log(\Psi)$, makes the normal distribution acceptable to approximate the experimental data since the normality test of the later distribution is passed as can be seen from Figure 5.6 (all data fall into the 95% confidence interval which is outlined by the blue lines).

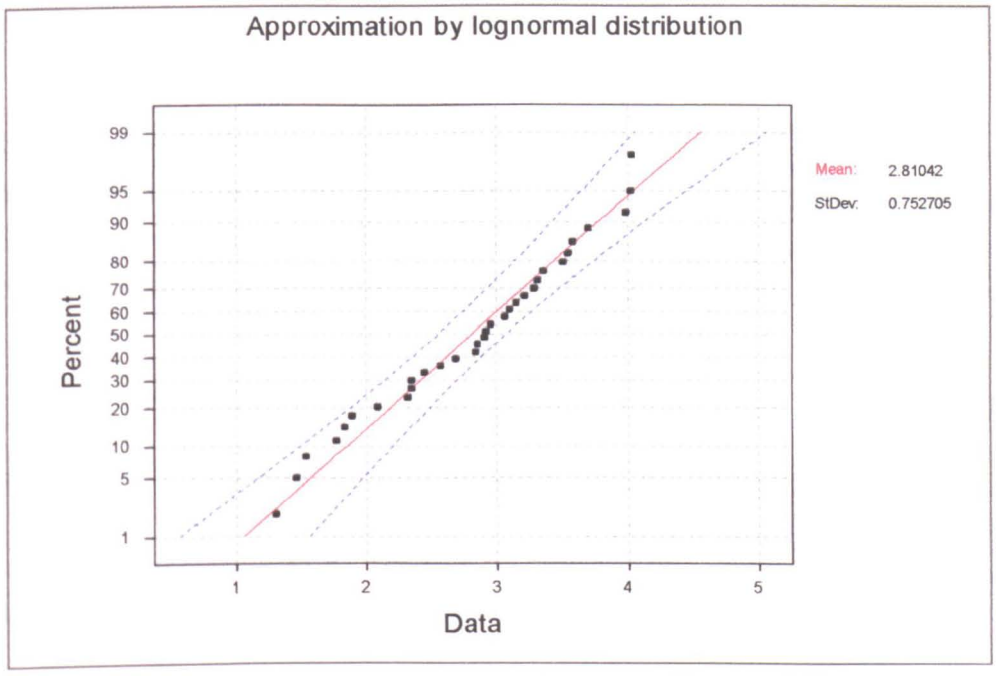


Figure 5.6. Distribution of Ψ after the logarithmic transformation

The mean value $\mu_{\log(\Psi)}$ of the transformed random variable can be used as an alternative measure of GA efficiency whenever the normality of distribution is required.

Having established the performance measure, we can now try to find an expression for this measure in terms of GA factors ϕ_i (such as the probabilities of mutation and crossover, the overall number of generations, selection pressure, etc):

$$\beta = \beta(\phi_i),\ i = \overline{1,l} \tag{5.13}$$

where l is the number of the GA factors which significantly affect the performance. To ascertain which factors are significant and to particularise the relation (5.13) a systematic approach will be employed based on statistical analysis and described in detail in the next section.

5.3.2. Statistical analysis of GA performance

The ultimate goal of the statistical analysis presented here will be to estimate the optimal factor values ϕ_i^{opt} that minimise the performance measure (5.13) of Genetic Algorithms. (The performance measure needs to be minimised since it is directly related to the number of generations Ψ, the lesser values of which are preferable.) This will be done in three steps. First of all a screening experiment will be conducted reducing the number l of the GA factors which need to be included into the model (5.13). Only significant factors, variation of which noticeably (in a statistical sense) affect the performance, will remain in the model.

On the basis of the significant factors a second-order regression model will be obtained of the form:

$$\beta = a_0 + \sum_{i=1}^{l} a_i \phi_i + \sum_{1 \le i \le j \le l} b_{ij} \phi_i \phi_j \tag{5.14}$$

where $a_0, a_i, b_{ij} \in R$ for $\forall i, j$. The regression model is a result of a central composite experiment, which comprises the second step of the statistical analysis. Thirdly, and this will be the last step, the standard calculus techniques will be applied to the regression model (5.14) to specify the optimal values of significant GA factors. These values are the solutions of the following equations:

$$\frac{\partial \beta}{\partial \phi_i} = 0 \quad \forall i = \overline{1,l} \tag{5.15}$$

More detailed description of the steps leading towards the optimal setting of GA factors are now provided. (NOTE. All experiments involved in the present analysis were designed, implemented and interpreted using the MINITAB statistical package, a number of outputs from which will be included in the text.)

5.3.2.1. Screening experiment

The number of GA factors which might affect the performance of Genetic Algorithms is quite large. In order to ascertain how many of them really affect the performance, and therefore should be included in the model (5.13), a preliminary experiment will be carried out which utilises a *factorial design*. Factorial designs, i.e. designs where all factors are crossed, are more advantageous and efficient compared to designs in which only one experimental factor is varied at a time (Johnson and Bhattacharyya, 1992). Factorial designs allow the study of multiple factors in the same experiment and the assessment of the manner in which these factors interact. If no interactions are present, then the effect of each factor can be evaluated with the same efficiency as if the whole experiment had been devoted entirely to that factor (Wadsworth, 1998). However, factorial designs with even a moderate number of factors necessitate a large number of experimental trials. Therefore, factorial designs are usually restricted to those with all

factors at only two or three levels. Such designs are referred to as 2^k and 3^k designs, respectively.

The interpretation of experimental results obtained from factorial designs is based on the *analysis of variance* (ANOVA), a detailed description of which is given in Johnson and Bhattacharyya (1992). ANOVA allows one to test the hypothesis that the variation in the performance caused by changing the level of a particular factor is statistically different from the variation due to random errors. If this is the case, then the factor causing such variation is considered to be significant and will be included into the model (5.13). Otherwise, if the factor's effect is comparable with random errors associated with performance measurements, the factor will be excluded from further consideration.

The merit of the factorial design is that it arranges experimental trials in such a way that all main factors, as well as their interactions, can be easily estimated. However, by assuming that some of the interactions do not exist, it is possible to construct *fractional factorial designs* requiring fewer experimental trials. This may be especially useful when each trial necessitates a large amount of replicates as in the case, for example, of obtaining the Weibull scale parameter β characterising the GA performance. To validate this parameter, each trial in a factorial experiment is repeated 25 times.

By requiring fewer trials, fractional factorial designs facilitate less tedious statistical experimentation. For example, consider the case of four factors ϕ_1, ϕ_2, ϕ_3 and ϕ_4 each at two levels. A complete 2^4 factorial design would necessitate 16 runs, while using a one-half fraction of a 2^4 design (denoted as 2^{4-1}), the four factors can be studied in only eight runs. A 2^{4-1} fractional factorial design is constructed as follows. Starting with a 2^3 design, one assumes that the $\phi_1\phi_2\phi_3$ interaction does not exist and uses it to measure the effect of a fourth factor ϕ_4. The effect of ϕ_4 is said to be confounded with the $\phi_1\phi_2\phi_3$ interaction, and what is actually being measured by ANOVA in this case is the effect of ϕ_4 plus the effect of $\phi_1\phi_2\phi_3$. The assumption that the interaction $\phi_1\phi_2\phi_3$ produces no effect allows us to attribute the result obtained from ANOVA to the effect of the fourth factor ϕ_4. The relation $\phi_4 = \phi_1\phi_2\phi_3$ is called a generator for the fractional factorial design and is used to identify other confounding patterns (or alias structure of the design). Thus, a one-half fraction of a 2^k design requires only half as many experiments as a complete factorial design. Smaller fractions (2^{k-2}, 2^{k-3} etc.) necessitate an even fewer number of experimental trials, but to obtain these fractions more than one generator is needed. This means that the measures of high-order interactions will be lost; however in many practical situations such sacrifice is quite acceptable (Box, Hunter and Hunter, 1978).

In this section we will use a fractional factorial approach to designing the experiment aimed at the evaluation of GA factor significance. Seven factors will be examined:

- the penalty coefficient for breaking $N_{\max}$;
- the penalty coefficient for breaking C_{cum};
- the probability of mutation;
- the probability of crossover;
- fitness normalisation maximum;
- fitness normalisation slope;
- the number of cross-points for crossover.

For examining the effects of these factors on GA performance, a screening experiment has been carried out utilising a 2^{7-2} fractional factorial design. Usually, when factorial designs are used in industrial-type experiments, the levels of experimental factors are determined by the characteristics of the production process (e.g. the highest and lowest permissible levels of temperature, pressure, viscosity etc.). While experimenting with GA factors however such guidelines are unavailable and the designer of Genetic Algorithms has no option other than to make guesses about the appropriate levels of GA factors. In our screening experiment the following high and low levels for the aforementioned GA factors have been used:

TABLE 5.8. *Levels of GA factors (one drug)*

GA factors	*Variable*	*Low level*	*High level*
Penalty for exceeding $N_{\max}$	ϕ_1	100	500
Penalty for exceeding C_{cum}	ϕ_2	15	95
Probability of mutation	ϕ_3	0.05	0.20
Probability of crossover	ϕ_4	0.25	0.50
Fitness normalisation maximum	ϕ_5	100	500
Fitness normalisation slope	ϕ_6	1	10
Number of cross-points for crossover	ϕ_7	2	20

The design and the results of the screening experiment are listed in Appendix B1.1. (In the design matrix 0 represents the low level of a GA factor and 1 represents the high level.) For each experimental set of factors, 25 replicate runs of the GA program were carried out. (Recall that the experiment is based on the GA optimisation of the problem (5.12).) The results obtained for each setting of GA factors are approximated by the Weibull distribution. The scale parameter β of the approximating distribution characterises the performance of GAs under the factor setting which yields such performance.

The analysis of the screening experiment is presented in Figure 5.7, where the significance of each GA factor is shown in the form of a histogram. The length of each column is proportional to the significance of the GA factor corresponding to this column. The fact that a column stretches over a certain threshold (depicted by the dotted line which corresponds to the critical value of the t-statistics and is calculated by MINITAB) indicates the significance of the corresponding GA factor with respect to its effect on the performance measure.

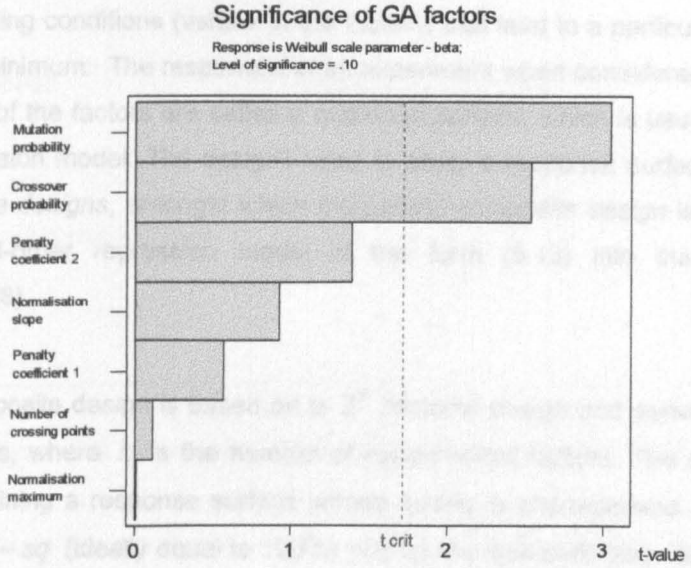


Figure 5.7. MINITAB analysis of the screening experiment (one drug)

As can be seen from Figure 5.7, only two GA factors, viz. the probabilities of mutation (ϕ_3) and crossover (ϕ_4), significantly affect the performance of Genetic Algorithms. In terms of the chemotherapy optimisation problem this means that the speed of finding a feasible treatment regimen is affected only by the frequencies of modifying the constituent doses and of exchanging various portions of different regimens. The magnitude of penalties for breaking the constraints or the selection pressure towards better treatment regimens in the current GA population do not significantly influence the search speed (as long as these penalties and pressure are present). The effects of 'insignificant' factors are indistinguishable from the effect which might be caused by random errors of performance measurements; thus, the other factors will be excluded from further analysis. The significant factors ϕ_3 and ϕ_4, on the other hand, will be examined more thoroughly in the next section, where a regression model of the form (5.14) will be constructed in terms of these factors.

The last remark of this subsection is that all GA factors $\phi_i, i = \overline{1,7}$, which have been examined here, belong to the category of controllable factors. This means that the designer of Genetic Algorithms has a direct control over them and can choose their values. However there is another category of factors, which also affect the performance of GAs but the values of which are predefined by the optimisation problem and as such lie beyond the designer's control. For example, the rate of tumour growth λ in the objective functional (5.8) may be considered as an

uncontrollable factor. The important corollary of this subsection is that the effects of uncontrollable factors can also be assessed by the factorial approach to experimental studies. (More details on this aspect of the screening experiments are given in Petrovski *et al* (1998b).)

5.3.2.2. Central composite experiment

Often when studying continuous factors, after finding their effects to be significant, one is interested in finding conditions (values of the factors) that lead to a particular response, usually a maximum or minimum. The responses of an experiment when considered as a function of the possible values of the factors are called a *response surface*, which is usually expressed in the form of a regression model. The designs used to study a response surface are referred to as *response surface designs*, amongst which the *central composite design* is commonly used for fitting a second-order regression model of the form (5.13) into the experimental data (Wadsworth, 1998).

The central composite design is based on a 2^k factorial design and derived from it by adding $2k$ design points, where k is the number of experimental factors. The additional points are required for obtaining a response surface whose quality is characterised by the coefficient of determination $R-sq$ (ideally equal to 100%) and by the lack-of-fit ratio. In MINITAB this ratio is measured by the $p-\text{value}$ which denotes the probability of getting such differences between the observed experimental data and the responses predicted by the regression model. If $p-\text{value}$ is greater than 0.05 then the regression model is considered to be adequate and can be used for making inferences about the experimental factors.

In the previous section we established that only two GA factors, ϕ_3 and ϕ_4, significantly affect the performance of Genetic Algorithms. Now the effects of these factors will be examined more closely by carrying out a central composite experiment. The performance measure is again obtained by repeating each experimental run 25 times and fitting a Weibull curve into the resultant observations. The scale parameter β of the fitted curve is used as the response attributed to each setting of experimental factors. The design matrix and the results of the central composite experiment are listed in Appendix B1.2.

The analysis of these results yields the following regression model, which particularises the equation (5.14):

$$\beta = 1723 - 5874\phi_3 - 3731\phi_4 + 7565\phi_3^2 + 2281\phi_4^2 + 4683\phi_3\phi_4 \qquad (5.16)$$

The lack-of-fit ratio calculated for this model by MINITAB is expressed as $p = 0.207$ and $R-sq = 88.4\%$. A graphical representation of the regression model is shown in Figure 5.8.

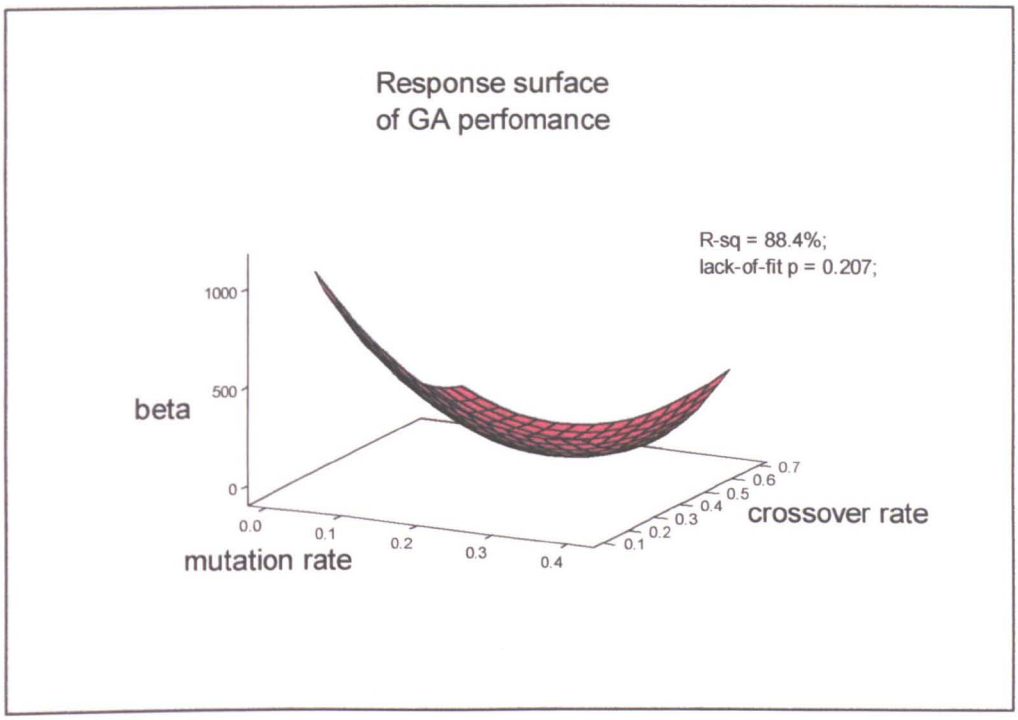


Figure 5.8. Response surface of GA performance (one drug)

As can be seen from this graph, the curvature of the response surface indicates that within the experimental ranges of factors ϕ_3 and ϕ_4 there is a certain setting of these factors at which the performance measure β attains its minimum. Small values of β imply that Genetic Algorithms require a small number of generations Ψ to find a feasible solution to the problem under investigation and hence perform an efficient search. The goal of undertaken regression analysis therefore is to find the optimal values of ϕ_3 and ϕ_4 which minimise the performance measure β of Genetic Algorithms. The last step that needs to be taken to attain this goal is the solution of the system of differential equations (5.15), which after particularisation of the regression model (5.16) acquire the following form:

$$\begin{cases} \dfrac{\partial \beta}{\partial \phi_3} = -5874 + 15130\phi_3 + 4683\phi_4 = 0 \\ \dfrac{\partial \beta}{\partial \phi_4} = -3731 + 4562\phi_4 + 4683\phi_3 = 0 \end{cases} \tag{5.17}$$

This system yields the optimal values for the significant GA factors which represent the probabilities of mutation (ϕ_3) and crossover (ϕ_4):

$$\begin{cases} \phi_3^{opt} = 0.1981 \\ \phi_4^{opt} = 0.6146 \end{cases} \tag{5.18}$$

To verify that the factor estimates obtained from the statistical analysis really improve the efficiency of the GA search, a confirmation experiment has been carried out. In the confirmation run of the GA program the mutation and crossover probabilities were set at their optimal levels, whereas the values of other GA factors listed in Table 5.8 were chosen arbitrarily from the

specified in that table ranges. 25 replicates of the random variable Ψ were obtained during the confirmation experiment and a Weibull curve was fitted into the resultant data. The estimate of the Weibull scale parameter for the confirmation data is $\beta = 27.6893$, which is better than any result of the screening and the central composite experiments. In order to ascertain whether it would have been possible to obtain the same improvement of GA performance had we used alternative factor tuning methods, a comparative study has been undertaken, the results of which are presented in the next section.

5.3.3. Statistical approach vs. conventional methods

In the previous section we have demonstrated that with the help of the statistical analysis of GA factors it is possible to improve the efficiency of Genetic Algorithms. However we need to substantiate the benefits of employing the statistical approach introduced when compared with other methods for adjusting GA factors. A short survey of these latter methods was given in Section 4.4.1 where it was indicated that in addition to ad hoc and skilful factor adjustment techniques, there exist systematic approaches which can be used for comparison.

5.3.3.1. Selection of a contestant method

In Section 4.4.1 three main groups of systematic approaches to GA factor optimisation were presented, viz. meta-optimisation, optimisation by factorial experiments and dynamic optimisation. The statistical methodology of GA factor tuning introduced in Section 5.2 is based on factorial experimentation and therefore belongs to the second group of approaches. Thus, to make an unbiased comparison, a contestant method ought to represent either dynamic optimisation approaches or meta-optimisation.

Dynamic optimisation (adaptation) of GA factors is a group of methods which determine the values of GA factors dynamically during the execution of the program implementing Genetic Algorithms. As was pointed out in Section 4.4.1.4, these methods tune the values of GA factors using either pre-programmed schedules or self-adaptation strategies (e.g. by monitoring the effectiveness of each evolutionary operator (Davis, 1989) or by using a co-evolutionary strategy (Tuson and Ross, 1996)). Although potentially promising, the dynamic optimisation of GA factors heavily relies on designer intuition about the best set-up of adaptation procedures. According to Beasley *et al* (1993), the methods of dynamic factor tuning replace an optimisation task defined on the set of candidate GA factors by the optimisation operating on a larger and more complex space of candidate adaptation strategies. The experiments with changing the starting condition of the GA search and with modifying the constraints of cancer chemotherapy (implemented in Section 5.2) have shown that the dynamics of the problem of chemotherapeutic treatment optimisation changes with the set-up of the problem. Since the techniques of dynamic GA factor tuning often necessitate rethinking of the adaptation strategy when the problem's set-up has been changed, these techniques do not suit very well for the task of GA performance enhancement in the context of cancer chemotherapy optimisation.

Meta-optimisation, on the other hand, is more directly comparable to the method of statistical inference. The statistical approach implements the modelling of the performance of Genetic Algorithms in terms of GA factors and then finds the factor values which optimise the performance model. The meta-optimisation methods, operating on a population of GA factor settings, carries out a heuristic search through the space of all possible settings of GA factors and tries to find the best setting in order to enhance the GA performance. Because of this conceptual similarity between the meta-GA and the statistical approaches to GA factor tuning, the meta-GA method has been chosen as a contestant method for the comparison.

5.3.3.2. Meta-GA optimisation

To tune the factors of Genetic Algorithms using Genetic Algorithms themselves has been a fascinating idea for quite a long time. Empirical studies in this area were originated by DeJong (1975) when he devised a suite of five test functions presenting the scale of optimisation difficulty for the gradient techniques. Using DeJong's suite as a test-bed for GA optimisation, Grefenstette (1989) suggested a robust approach to determining the good values of GA factors whereby the factor settings themselves were encoded as chromosomes and evolved within a meta-GA population. Schaffer *et al* (1989) expanded the test function suite in an attempt to make the meta-GA approach more versatile and robust.

To assess the impact of GA factors on the efficiency of Genetic Algorithms two performance measures, *online* and *offline* indexes, have been proposed (Grefenstette, 1989). The online index, attributed to a particular factor setting, is simply the average fitness of all base-level chromosomes tested by the genetic algorithm with this factor setting. This measure would be appropriate in situations where each test of base-level solutions must be taken into account (Schaffer *et al*, 1989); the online index penalises GA factor settings that tend to test many poor solutions before locating the good ones. Thus, to do well on this performance measure, a setting of GA factors must provide the search with the ability to quickly find the region where the best solutions are situated.

The offline index, on the other hand, characterises a given GA factor setting by monitoring only the fitness of the best base-level solution found by the genetic algorithm with this factor setting. This measure does not penalise genetic algorithms which explore poor regions of the search space on the way to better solutions. In this chapter we are concerned with the ability of Genetic Algorithms to locate, as well as with their efficiency in doing so, any feasible solution to the optimisation problem of cancer chemotherapy. The offline index is a more suitable measure in this case, since if the best solution in a GA population satisfies the feasibility criteria, then the optimisation goal is attained and there is no need to consider the quality of other solutions in the population. Therefore, the offline index will hereafter be used to evaluate the quality of a meta-GA solution.

There are a number of forms which the offline index may take. The standard approach to the definition of this index is based on comparing the fitness of the best solution found by GAs with that of the best solution obtained by a random search (Grefenstette, 1989). Alternatively, the number of generations Ψ can be used as an offline index of GA performance for the reasons discussed in Section 5.3.1. Since the latter approach makes the comparison of meta-GAs with the statistical method more straightforward, we will adopt it in the comparative study which is now presented.

5.3.3.3. Comparison between meta-GAs and the statistical method

Earlier in this chapter we established that the performance of Genetic Algorithms, optimising single-drug chemotherapeutic treatments, is affected by two factors, viz. by the probabilities of mutation (p_{m}) and crossover (p_{c}). The utilisation of the statistical approach to tuning these probabilities yielded the optimal setting (5.18). Let us now examine how meta-GAs perform the same task of finding the optimal values for these factors.

The solution space S for the meta-GA method will be $S=[0,0.5]\times[0,1]$, where $[0,0.5]$ and $[0,1]$ are the ranges of acceptable values for the probabilities of mutation and crossover respectively. The representation space I of meta-GAs is defined as

$$I=\left\{a_1a_2a_3\ldots a_{14} : a_k \in (0,1) \;\; \forall k=\overline{1,14}\right\}$$

where the first seven genes $a_1a_2\ldots a_7$ encode p_{m} and the last seven genes $a_8a_9\ldots a_{14}$ encode p_{c}. Since the use of a 7-bit encoding allows these GA factors to have 128 distinct values (including 0), the mapping between the representation and the solution spaces of meta-GAs can be expressed by the following relations:

$$\begin{cases} p_{\mathrm{m}}=\sum_{k=1}^{7}\dfrac{2^{7-k}}{2\cdot 127}\cdot a_k \\ p_{\mathrm{c}}=\sum_{k=8}^{14}\dfrac{2^{14-k}}{127}\cdot a_k \end{cases} \tag{5.19}$$

To run the meta-GA program itself the standard GA factor set, suggested by Grefenstette (1989), and a standard GA program configuration were used: that is, population size = 50, crossover probability = 0.6, mutation probability = 0.001, the meta-GA program uses generational replacement, windowing fitness scaling and the elitist selection strategy (see Chapter 4). After evaluating 1000 settings of the GA factors under examination, the meta-GA program found the following best combination:

$$\begin{cases} p_{\mathrm{m}}^{\mathrm{opt}}=0.109 \\ p_{\mathrm{c}}^{\mathrm{opt}}=0.867 \end{cases} \tag{5.20}$$

Having obtained these values, a confirmation experiment was run similar to that carried out for the optimal set (5.18). The results of the confirmation experiments with the optimal factor settings (5.18) and (5.20) are listed together in Appendix B1.3. A general observation, which can be made from their comparison, is that the values of GA factors obtained from the statistical analysis seem to be better tuned. To statistically prove this fact we need to employ a hypothesis test on the inequality of the samples' means. The most commonly used test for this purpose is a two-sample one-tailed t-test. However, the t-test assumes the normality of samples' distributions. Therefore the logarithmic transformation of the confirmation data has been performed before testing the hypothesis that the means of the two confirmation samples are not equal. On a logarithmic scale, the distributions of the results of the confirmation experiments are normal (see Section 5.3.1) and Figure 5.9 presents the boxplots of the experimental data.

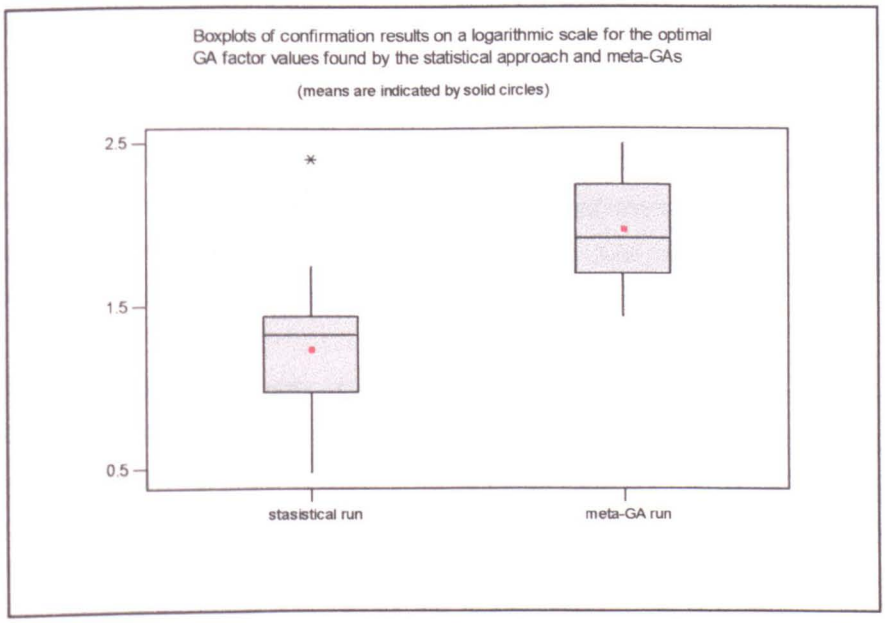


Figure 5.9. Comparison of the confirmation results

The application of a one-tailed t-test to these data confirms that at a statistical level of 95% the confirmation results obtained from the setting (5.18) are better compared to those yielded by the setting (5.20). Quantitatively this can be expressed by the value of the t-statistic for the confirmation results on a logarithmic scale: $t = 6.78 > (t_{0.1\%} = 3.277)$. Alternatively, Wilcoxon Confidence Intervals for the medians on the original data scale can be used, which are detailed in the following table:

TABLE 5.9. *Wilcoxon Signed Rank Confidence Intervals*

	Median	Confidence Interval	Confidence level
statistical run	19.0	(14.0, 27.0)	95%
meta-GA run	108.5	(68.5, 162.0)	95%

The Mann-Whitney test of the data on the original scale confirms the conclusion drawn from the t-test (the value of the corresponding non-parametric statistic is $W = 370.0$, which shows a significant difference ($p < 0.0001$) between the confirmation runs). Hence, as may be seen from all these comparisons, the statistical tuning method, developed in this thesis, performs better optimisation of GA factors than the alternative methods of GA factor tuning.

5.4. Conclusions

In this chapter we demonstrated how evolutionary optimisation can be applied to the problem of single-drug chemotherapy. On the basis of the optimal control approach (Martin *et al*, 1990) to modelling and optimising chemotherapeutic treatment a search problem has been formulated suitable for solution by the methods of evolutionary computation. Amongst these methods two particular techniques, viz. Genetic Algorithms and Regimen Modification, have been chosen to implement the task of finding good chemotherapeutic regimens. (NOTE. There exist other methods of evolutionary optimisation such, for instance, as Neural Networks or Simulated Annealing. The choice of Genetic Algorithms in the present context can be explained by the fact that GAs are better suited for the problems requiring optimal parameterisation and/or scheduling (Baeck *et al*, 1997).) Appropriate solution representations and fitness functions for the specified evolutionary strategies have been found which take into account the objectives of treatment optimisation and the constraints of cancer chemotherapy.

To evaluate the quality of evolutionary optimisation, its robustness and efficiency in comparison with conventional optimisation techniques (CM and HJ), a test-bed was used based on the linear programming transformation of the optimisation problem under investigation. The treatment regimens obtained from GAs, RM, CM and HJ were compared on the optimality scale provided by LP. This comparison has shown first of all that in cases when the search starts from an arbitrary point in the solution space, Genetic Algorithms are able to find a solution which satisfies all constraints. Finding such a solution was a problematic issue for the traditional optimisation techniques, which consistently ended the search in the infeasible region of the solution space when a random start was given. The robustness of Genetic Algorithms with respect to the ability to find a feasible treatment regimen regardless of the starting condition of the search was the first important experimental result of the present work. This result illustrated the usefulness of Genetic Algorithms in the context of chemotherapy optimisation, especially for the development of novel treatment strategies *ab initio*. Secondly, in cases when the optimisation starts from a feasible solution, the methods of evolutionary computation (GAs and RM) have shown either superiority or, in the worst case, compatibility with traditional optimisation techniques. This means that whenever there is a treatment regimen approved by clinicians and therefore feasible, the utilisation of Genetic Algorithms or Regimen Modification for its improvement also can be more effective compared to the use of non-linear programming methods of constrained optimisation.

The disadvantage of evolutionary optimisation is that it is a time consuming procedure. However, by tuning certain factors which determine the properties of the evolution process it is possible to noticeably improve the speed of the GA search. In this chapter a novel approach to GA performance enhancement has been developed, based on statistical inference and regression analysis. This approach outperforms an alternative method of factor tuning and its application yields a substantial improvement in the search speed (the number of required GA generations to find a feasible chemotherapy regimen has been reduced from ~500 to ~20). Moreover, statistical modelling of the search speed provides the data, on the basis of which it is possible to predict the likely time of search completion. For example, the results of the confirmation experiment with the optimal factors found by the statistical approach indicate that the 95% confidence interval for the median number of GA generations required to find a feasible solution is $[14,27]$. Therefore, it might be useful for improvement of the search efficiency to restart Genetic Algorithms after a certain amount of generations (30 generations in our case) rather than to wait for the completion of long runs. The statistical approach to GA factor tuning provides the data necessary to determine the safe restarting time in each particular case.

However, the statistical approach is not a panacea for determining the optimal values of GA factors. Consider for instance the situation when due to either a large random variation in experimental responses or due to an inappropriate choice of factor domains the factorial experiment cannot reliably identify the significant factors; then the consequent regression analysis becomes meaningless. What one may attempt to do in that case is to modify the factor ranges using an intuitive approach and to repeat a factorial experiment. If this attempt fails, meaning that GA factors have no control over the search speed, the process of factor tuning should be abandoned. Still, the merit of the statistical approach is that it necessitates the evaluation of much fewer factor settings to establish the fact that GA performance is not affected by the factors' values.

As can be seen from all the conclusions made above, the problem of single-drug chemotherapy optimisation has provided a fruitful case study for the assessment of how useful and effective the method of Genetic Algorithms can be in the context of chemotherapeutic treatment of cancer. It has been verified in this chapter that the GA method implements a robust and effective search for good solutions to multi-constrained and non-linear optimisation problems. The Regimen Modification version of the method, also based on evolutionary search, turned out to be the most adequate technique for improving feasible but non-optimal solutions to optimisation problems of a similar sort. To evaluate the quality of the GA and RM optimisation, the LP test-bed with a known optimal solution was used in this chapter. The results of this evaluation have shown that the evolutionary methods can perform the required optimisation task very well. Having acquired the confidence in the ability of GAs and RM to handle multi-constraint non-linear optimisation problems, we now can apply these methods to situations where none of the known techniques of mathematical optimisation have been used or have led to a conclusive outcome. The next chapter, which is concerned with the optimisation of multi-drug chemotherapy, presents an example of such a situation.

CHAPTER 6

MULTI-DRUG CHEMOTHERAPY OPTIMISATION USING GENETIC ALGORITHMS

The optimisation of multi-drug chemotherapy is the culminating aspect of the present work. The material presented in the previous chapters of this thesis was necessary to prepare the ground and to make a number of essential contributions towards the development of a practical method whereby the effectiveness of multi-drug chemotherapeutic treatments can be improved. Having accomplished all these, we can now concentrate on the implementation issues of multi-drug chemotherapy optimisation.

In practice oncologists usually utilise multi-drug treatments due to the ability of such treatments to better deal with drug resistance (Goldie and Coldman, 1979; Birkhead and Gregory, 1984; Wheldon, 1988; Martin and Teo, 1994). This emphasises the importance of a reliable and practically usable approach to the optimisation of multi-drug chemotherapy. However, when a cocktail of anti-cancer drugs is administered to a cancer patient, the tumour kinetics under such treatment and the toxicity constraints become more complex. In particular, the fourth constraint of cancer chemotherapy (see Section 1.3), which is concerned with the risk reduction of toxic side effects produced by multiple drugs, now has to be taken into account. Finding a mathematical expression for this constraint is a non-trivial task which will be addressed in this chapter. However what is conceivable even without mathematical formulation is that in the presence of the toxicity constraint on multiple drug administration, Linear Programming cannot be applied to the problem under investigation without its substantial oversimplification. Moreover, if the goal of a multi-drug treatment is to minimise the overall tumour burden or to prolong the PST (see Section 1.2), then the objective of optimisation becomes a non-linear function of the control variables. As a result of this the LP technique cannot provide the optimal multi-drug regimen. Since in the previous chapter we showed that the methods of Genetic Algorithms and Regimen Modification show the best performance with respect to the robustness and the effectiveness of the search for better treatment regimens, the choice of these methods as optimisation tools for multi-drug treatments seems to be natural.

The following sections of this chapter are aimed at developing, exploring and utilising such optimisation tools. Section 6.1 addresses the particulars of how the optimisation problem of multi-drug chemotherapy can be transformed into a search problem suitable for Genetic Algorithms and how these particulars differ from those of the single-drug case. Furthermore, in Section 6.1.3 the statistical approach to GA factor tuning (discussed in Section 5.3) will be applied in the attempt to enhance the speed of the GA search for good multi-drug treatment regimens. Section 6.2 presents the resultant treatment regimens which have been found by Genetic Algorithms subject to different optimisation objectives of cancer chemotherapy. Having developed and explored a workable version of Genetic Algorithms for the optimisation of multi-drug chemotherapeutic treatment, this version can be encapsulated in an autonomous optimisation module and can be used in a more general framework. Section 6.3 specifies the framework, into which the GA optimisation module can be incorporated, and gives the details of this incorporating process. Finally, in Section 6.4 we will focus on implications of the results obtained in this chapter, as well as on the practicality of the optimisation module developed.

6.1. Multi-Drug Chemotherapy as an Optimisation Problem for GAs

The transformation of any practical problem into an optimisation problem for Genetic Algorithms involves specifying the following: (1) the encoding scheme, (2) the fitness function and (3) the evolutionary procedures of initialisation, selection and recombination. Let us start with the encoding scheme.

6.1.1. Representation of multi-drug treatments

In Section 5.1.1 we described in detail how a single-drug treatment can be encoded as a GA chromosome. A discrete dosage approach was adopted to represent treatment regimens and two encoding strategies, viz. Genetic Algorithms and Regimen Modification, were suggested and implemented. In the case of multi-drug treatments the representation approach and the encoding strategies remain the same, subject to slight modifications.

First of all, the vector of control variables now acquires the form $\mathbf{c} = (C_{ij}), i = 1, n, j = 1, d$, where d is the number of drugs used in the treatment and C_{ij} are the constants denoting the concentration of the j^{th} drug in the blood plasma during the i^{th} treatment interval. In the present work the maximum number of drugs in chemotherapeutic cocktails will be limited to 10 since, according to the collaborating oncologists, it is uncommon in practice to administer too many drugs simultaneously. The number of treatment intervals also will be equal to 10 for the sake of simplicity of simulation experiments. (Recall that during each treatment interval the concentration levels of all drugs in the blood are assumed to be constant.) However, should a practical situation necessitate a larger number of discrete multi-drug doses or anti-cancer drugs, the number of control variables C_{ij} can be increased.

All possible control vectors $\mathbf{c} = (C_{ij}), i = 1,n, j = 1,d$ constitute the solution space S for the problem of multi-drug chemotherapy optimisation. The representation space I of S can be expressed as a Cartesian product similar to that of (5.1)

$$I = A_1^1 \times A_1^2 \times \ldots \times A_1^d \times A_2^1 \times A_2^2 \times \ldots \times A_2^d \times \ldots \times A_n^1 \times A_n^2 \times \ldots \times A_n^d \tag{6.1}$$

In order to make the encoding procedure of single-drug treatments compatible with the multi-drug case, the allele sets A_i^j will utilise the same encoding schemes, viz. the 4-bit scheme

$$A_i^j = \left\{a_1 a_2 a_3 a_4 : a_k \in \{0,1\} \forall k = \overline{1,4}\right\}$$

and the sign scheme

$$A_i^j = \{0,1,-1\}.$$

These schemes allow one to implement the Genetic Algorithm and the Regimen Modification encoding strategies for representing multi-drug treatments. The former strategy yields the following form of a chromosome $x \in I_{\mathrm{GA}}$

$$x = \left\{a_1 a_2 a_3 \ldots a_{4nd} : a_k \in \{0,1\} \ \forall k = \overline{1,4nd}\right\} \tag{6.2}$$

whereas the latter strategy represents a chromosome $x \in I_{\mathrm{RM}}$ as

$$x = \left\{a_1 a_2 a_3 \ldots a_{nd} : a_k \in \{0,1,-1\} \ \forall k = \overline{1,nd}\right\} \tag{6.3}$$

The encoding functions, mapping the representation spaces I_{GA} and I_{RM} onto the solution space S, will have the following forms (see (5.10) and (5.11) for comparison):

$$\mathrm{GA}: \qquad C_{ij} = \Delta C_j \sum_{k=1}^{4} 2^{4-k} a_{4d(i-1)+4(j-1)+k}, \ \forall i = \overline{1,n}, j = \overline{1,d} \tag{6.4}$$

$$\mathrm{RM}: \qquad C_{ij} = C_{ij}^0 + a_{d(i-1)+j}\Delta C_j, \qquad \forall i = \overline{1,n}, j = \overline{1,d} \tag{6.5}$$

where C_{ij}^0 are the control variables associated with the treatment regimen which is undergoing the improvement process using RM; ΔC_j are the concentration units attributed to each drug. As may be seen from (6.4), the adopted GA encoding strategy uses the same number of gradations for all anti-cancer drugs. Recall that the concentration unit and the length p of a binary string representing the concentration level are related as

$$\Delta C = \frac{C_{\max} - C_{\mathrm{th}}}{2^p - 1}$$

where $C_{\max}$ is the maximum tolerable concentration of a given anti-cancer drug; C_{th} is the concentration level at which the drug ceases to have a therapeutic effect. In the case of single-drug chemotherapy, ΔC had a unique value. However, when multiple anti-cancer drugs are used with different maximum concentration levels, the meaning of the concentration unit becomes ambiguous.

There are two ways to resolve this ambiguity. One way is to fix the value of the concentration unit ΔC and to use bitstrings of variable length to encode concentrations of different drugs in accordance with their C_{max} level. The alternative way is to use the same length of bitstrings for all drugs (i.e. to have the same number of gradations regardless of C_{max}) and to adjust the value of ΔC for each drug accordingly. (NOTE. This dilemma is important only for the GA encoding strategy. This is because the GA strategy incorporates the C_{max} constraint via encoding and makes the length p of bitstrings and the concentration unit ΔC dependent on each other.) To make the GA implementation of multi-drug chemotherapy optimisation compatible with that of the single-drug case, the approach of fixing p will be adopted in the present work. Then, the concentration unit for each drug is gauged using the method of potency factors whereby a specific factor is assigned to every anti-cancer drug used for composition of cancer chemotherapy regimens (Henderson, 1997). A potency factor for each drug is calculated by normalising its Maximum Tolerable Dose (MTD) with that of Adriamycin. Adriamycin has been chosen as a standard since, according to Perry (1992), it "is perhaps the single most important drug in the chemotherapists' arsenal, used in a wide range of malignancies". The MTD is used since it is a common chemotherapeutic measure, which enables direct comparison between drugs and which has been directly established on human patients in Stage 1 trials.

Consider an example. Suppose we want to calculate a potency factor for Cisplatinum whose MTD is equal 120 mg/m^2. The MTD for Adriamycin is 75 mg/m^2, therefore the potency factor for Cisplatinum is:

$$\rho_{\text{cisplatinum}} = \frac{\text{MTD}_{\text{standard}}}{\text{MTD}_{\text{Cisplatinum}}} = \frac{75}{120} = \frac{5}{8} = 0.625$$

The potency factor of Adriamycin itself obviously is equal to 1. Since we have decided to use Adriamycin as a gauge, the standard concentration unit hence will be

$$\Delta C^* = \frac{75\ \text{mg/m}^2}{15} = 5\ \text{mg/m}^2.$$

However, a concentration unit for Cisplatinum, for instance, will be different and will be calculated as follows:

$$\Delta C_{\text{cisplatinnum}} = \frac{\Delta C^*}{\rho_{\text{cisplatinum}}} = \frac{5\ \text{mg/m}^2}{0.625} = 8\ \text{mg/m}^2$$

So if we decide to administer the maximum amount of Cisplatinum, that is 15 concentration units in our binary representation, then we deliver the dose ($15 * 8\ \text{mg/m}^2 = 120\ \text{mg/m}^2$), which is equal to MTD for this particular drug. The potency factors of other anti-cancer agents most frequently used in multi-drug cancer chemotherapy are tabulated in Appendix C.

Applying the concept of potency factors, we can now revise the representation functions (6.4) and (6.5) in order to settle the ambiguity of multiple drug encoding on the same scale:

$$\text{GA}: \qquad C_{ij} = \frac{\Delta C^*}{\rho_j} \sum_{k=1}^{4} 2^{4-k} a_{4d(i-1)+4(j-1)+k}, \quad \forall i = \overline{1,n}, j = \overline{1,d} \tag{6.6}$$

$$\text{RM}: \qquad C_{ij} = C_{ij}^0 + a_{d(i-1)+j} \frac{\Delta C^*}{\rho_j}, \qquad \forall i = \overline{1,n}, j = \overline{1,d} \tag{6.7}$$

The functions (6.6) and (6.7) map chromosomes in I_{GA} and I_{RM} respectively onto the space S, the elements of which are potential solutions to the problem of multi-drug chemotherapy optimisation. In order to define the fitness function for GAs we need to establish a way to evaluate the quality of these potential solutions. The next section addresses this issue.

6.1.2. Evaluation of multi-drug treatment regimens

As in assessing the quality of single-drug chemotherapy regimens (see Section 5.1.2), multi-drug treatments will be evaluated on the basis of the objective functional $J(\mathbf{c})$ which specifies a treatment goal. As was pointed out in Section 1.2, a cancer chemotherapy treatment may be either curative (the treatment goal is to reduce the tumour to a critical size N_{cure}) or palliative (the treatment goal is to prolong PST and maximise the patients' quality of life). In the present work we are mainly concerned with finding a curative treatment; therefore the primary objective of the multi-drug chemotherapy is to eradicate the tumour.

The objective of tumour eradication can be mathematically formulated using two approaches. The first approach, which has been already actively exploited in the previous chapter, results in the following form of the objective functional:

$$\begin{aligned} &\text{minimise } J_1(\mathbf{c}) = N(T_{\text{final}}) \\ &\text{where } [T_0, T_{\text{final}}] \text{ is the fixed-length treatment interval.} \end{aligned} \tag{6.8}$$

The alternative approach to tumour eradication takes into account the kinetics of tumour development and attempts to keep the tumour burden to an absolute minimum at all times during the treatment interval. The latter approach enacts the second objective of chemotherapeutic treatment and can be formulated as follows:

$$\text{minimise } J_2(\mathbf{c}) = \int_0^{T_{\text{final}}} (N(t) - N_{\text{cure}})^{2l+1} dt \qquad \text{where } l = 0,1,2,\ldots \tag{6.9}$$

Intuitively speaking, the objective functional (6.9) is a gradual transition from tumour size minimisation to prolongation of the patients' survival time. Since the integrand in (6.9) is raised to an odd power, the minimisation of $J_2(\mathbf{c})$ is equivalent to reducing the tumour size $N(t)$ below a certain limit N_{cure} and forcing the tumour burden to stay there for as long as possible. Thus, the tendency to increase the time duration, when a certain condition is satisfied, comes

into play, which is closely related to prolongation of PST. The patient survival time T itself is defined as the instance when the tumour reaches its maximum tolerable level $N(T) = N_{\max}$. Therefore, the objective of PST prolongation may be formulated as follows (Martin and Teo, 1994):

$$\text{maximise} \quad J_3(\mathbf{c}) = \int_0^T 1 dt \tag{6.10}$$

The objective functionals (6.8)-(6.10) are to be optimised subject to the following state equation (the multi-drug version of (5.5)):

$$\dot{N}(t) = N(t) \cdot \left[\lambda \ln\left(\frac{\Theta}{N(t)} \right) - \sum_{j=1}^{d} \kappa_j \sum_{i=1}^{n} C_{ij} \{ H(t - t_{i-1}) - H(t - t_i) \} \right] \tag{6.11}$$

where C_{ij} the control varibles denoting the concentrations of the anti-cancer drugs used;

κ_j the quantities representing the efficacy of the anti-cancer drugs;

and the extended set of constraints:

$$\begin{cases} g_1(\mathbf{c}) = C_{\max\, j} - C_{ij} \geq 0 & \forall i = \overline{1,n}, j = \overline{1,d} \\ g_2(\mathbf{c}) = C_{\text{cum}\, j} - \sum_{i=1}^{n} C_{ij} \geq 0 & \forall j = \overline{1,d} \\ g_3(\mathbf{c}) = N_{\max} - N(t_i) \geq 0 & \forall i = \overline{1,n} \\ g_4(\mathbf{c}) = C_{\text{s-eff}\, k} - \sum_{j=1}^{d} \eta_{kj} C_{ij} \geq 0 & \forall i = \overline{1,n}, k = \overline{1,\Omega}. \end{cases} \tag{6.12}$$

The first two constraints ($g_1(\mathbf{c})$ and $g_2(\mathbf{c})$) specify the boundaries for the maximum instantaneous dose $C_{\max\, j}$ and the maximum cumulative dose $C_{\text{cum}\, j}$ of each anti-cancer drug used. The third constraints, $g_3(\mathbf{c})$, limit the tumour size during treatment. The last constraints, $g_4(\mathbf{c})$, restrain toxic side effects of multi-drug chemotherapy. The factors η_{kj} represent the likelihood of damaging the k^{th} organ or tissue by administering the j^{th} drug. To qualitatively characterise these factors the system of pluses is used, exemplified in Table 1.1 and fully covered in Dearnaley *et al* (1995). After transforming them into a numerical format, the factors η_{kj} can be used to ensure that the administration of multiple drugs does not lead to an excess of the maximum tolerable value of side-effect toxicity $C_{\text{s-eff}\, k}$ attributed to each organ $k, k = \overline{1,\Omega}$ (see $g_4(\mathbf{c})$). The present study takes into account toxic side effects on five ($\Omega = 5$) vital organ and tissues, viz. on bone marrow, kidney, peripheral nerves, liver and heart. For the drugs, most commonly used in breast cancer treatments, the risk factors η_{kj} attributed to these organs and tissues are tabulated in Appendix C.

On the basis of whether a given regimen satisfies the constraints (6.12) or not, a distinction will be made between feasible and infeasible multi-drug treatment regimens. Feasible regimens will be evaluated simply by ascertaining what values of the objective functionals (6.8)-(6.10) they yield. The evaluation of infeasible regimens, on the other hand, involves penalising the values of the objective functionals by applying the distance-based penalty functions corresponding to each constraint $g_s(\mathbf{c}), s = \overline{1,4}$. The distance metrics from the feasible regions defined by the constraints $g_1(\mathbf{c})$ and $g_2(\mathbf{c})$ take the following forms for the multi-drug case:

$$\begin{cases} d_1 = \sum_{j=1}^{d} \sum_{i=1}^{n} \max^2 \{C_{ij} - C_{\max j}, 0\} \\ d_2 = \sum_{j=1}^{d} \max^2 \left\{ \sum_{i=1}^{n} C_{ij} - C_{\text{cum } j}, 0 \right\} \end{cases} \tag{6.13}$$

The distance from the feasible region defined by the maximum tumour size constraint $g_3(\mathbf{c})$ is measured in the same way as for single-drug treatments and uses the substitution $y(t) = \ln\left(\frac{\Theta}{N(t)}\right)$:

$$d_3 = \sum_{i=1}^{n} e^{-\beta \cdot t_i} \max^2 \{y_{\min} - y(t_i), 0\} \qquad \text{where } y_{\min} = \ln\left(\frac{\Theta}{N_{\max}}\right) \tag{6.14}$$

The measure of toxic side effects produced by multi-drug treatments has not been introduced previously and needs a more detailed explanation. In discussion with the collaborating oncologists it has been deemed reasonable to assume that the risk factors η_{kj} have an additive property and the maximum number of risk pluses, which any organ or tissue can sustain, is limited to five. Hence, if a particular multi-drug treatment accumulates more than five risk pluses on any vital organ or tissue, then this treatment will be considered infeasible and penalised in accordance with a degree of the constraint violation. This degree is measured by the following distance from the feasible region of $g_4(\mathbf{c})$:

$$d_4 = \sum_{i=1}^{n} \sum_{k=1}^{\Omega} \max^2 \left\{ \sum_{j=1}^{d} \eta_{kj} C_{ij} - C_{\text{s-eff } k}, 0 \right\} \tag{6.15}$$

where $C_{\text{s-eff } k}$ quantifies the maximum tolerable value (i.e. corresponding to five risk factors) of side-effect toxicity for the k^{th} organ.

Having defined the distance measures from the feasible regions attributed to the constraints $g_s(\mathbf{c}), s = \overline{1,4}$, we can now augment the objective functionals (6.8)-(6.10). After the substitution $y(t) = \ln\left(\frac{\Theta}{N(t)}\right)$, the state equation (6.11) yields the following analytical solution:

$$y(t_p) = y_0 e^{-\lambda \cdot t_p} + \frac{1}{\lambda} \cdot \left(e^{\lambda \cdot \Delta t} - 1\right) \cdot \sum_{j=1}^{d} \kappa_j \cdot \sum_{i=1}^{p} C_{ij} \cdot e^{\lambda \cdot (t_{i-1} - t_p)}, \; p = 0,1,2,\ldots,n \qquad (6.16)$$

If we omit the terms which do not contain the control variables, the augmented functional (6.8) becomes

$$\tilde{J}_1(\mathbf{c}) = \sum_{j=1}^{d} \kappa_j \cdot \sum_{i=1}^{n} C_{ij} \cdot e^{\lambda \cdot (t_{i-1} - T_{\text{final}})} - \sum_{s=1}^{4} P_s d_s \qquad (6.17)$$

where d_s are the distance measures defined in (6.13)-(6.15); P_s are the corresponding penalty coefficients. The augmented objective functional (6.9) of overall tumour minimisation takes the following form when $l = 0$:

$$\tilde{J}_2(\mathbf{c}) = \sum_{p=1}^{n} \sum_{j=1}^{d} \kappa_j \cdot \sum_{i=1}^{p} C_{ij} \cdot e^{\lambda \cdot (t_{i-1} - t_p)} - \sum_{s=1}^{4} P_s d_s \qquad (6.18)$$

In order to express the last treatment optimisation objective (6.10) in the augmented form, we firstly use (6.16) to determine the time T when $y(t)$ becomes equal to $y_{\min}$; secondly we form on the basis of this time an unconstrained optimisation objective, and finally we apply the penalty functions to generate the following objective:

$$\tilde{J}_3(\mathbf{c}) = \ln\left(\sum_{j=1}^{d} \kappa_j \cdot \sum_{i=1}^{n} C_{ij} \cdot e^{\lambda \cdot (t_{i-1} - T_{\text{final}})}\right) - \sum_{s=1}^{4} P_s d_s \qquad (6.19)$$

Thus, potential multi-drug regimens may be evaluated on the basis of the augmented functional (6.17)-(6.19) depending upon which particular treatment objective oncologists have in mind. The composition of these functionals with the mappings (6.6) and (6.7) produces the fitness functions for GA and RM respectively. The expressions for the fitness functions are listed in Appendix A2.

The last remark concerning evaluation of potential treatment regimens is related to the estimation of efficacy parameters κ_j. The effectiveness of each anti-cancer drug has to be calibrated so that known clinical trial results can be reproduced. The trial results detail the percentage of patients who either partially or completely responded to chemotherapy treatment. Henderson (1997) has developed a procedure for calibrating drug efficacy by analysing 100 tumours with normally distributed doubling times. This procedure takes into account the difference in patient response to a given anti-cancer drug and as such provides more reliable estimates of κ_j. Appendix C details the specific efficacy values which have been calculated for each drug being used in our studies. These values are listed along with maximum toxicity doses ($C_{\max j}$ and $C_{\text{cum}\,j}$) and potency factors ρ_j, all of which will be used in computational implementations of the GA optimisation of multi-drug treatments. The current section is aimed at specifying the particulars of these implementations. We have already addressed the encoding and the evaluation aspects; now the details on evolutionary procedures of Genetic Algorithms will be provided.

6.1.3. Evolutionary procedures of GAs for multi-drug treatments

All evolutionary procedures (i.e. population initialisation, chromosome selection and recombination) of Genetic Algorithms for multi-drug treatments are similar to those of the single-drug case. The GA population consists of 50 chromosomes initialised at random. The selection procedure is the combination of the roulette-wheel selection with a linear fitness normalisation scheme. The recombination procedure uses the crossover and mutation operators.

However, chromosomes representing multi-drug treatments are 10 times longer than their single-drug counterparts. An increased size of the chromosomes, as well as an additional constraint imposed on the optimisation task, might have deformed the search space of treatment regimens. If that is the case, then the optimal GA factor values obtained in Section 5.3.2 need to be readjusted. To do this the statistical approach to GA factor tuning, developed in the previous chapter, has been employed.

Firstly, a screening experiment utilising a 2^{8-3} fractional factorial design has been carried out with the following levels of the design factors:

TABLE 6.1. *Levels of GA factors (multiple drugs)*

GA factors	*Variable*	*Low level*	*High level*
Penalty for exceeding N_{max}	ϕ_1	100	500
Penalty for exceeding C_{cum}	ϕ_2	15	95
Penalty for exceeding $C_{s\text{-}eff}$	ϕ_3	75	750
Probability of mutation	ϕ_4	0.05	0.20
Probability of crossover	ϕ_5	0.25	0.50
Fitness normalisation maximum	ϕ_6	100	500
Fitness normalisation slope	ϕ_7	1	10
Number of cross-points for crossover	ϕ_8	2	200

Apart from the presence of an additional penalty coefficient, there is only one difference between the contents of Table 6.1 and Table 5.1. (Table 5.1 contains the levels of GA factors for the screening experiment in the case of single-drug chemotherapy optimisation.) This difference refers to the high level of the last GA factor, viz. the number of crossing points pertinent to the crossover operator. In the multi-drug case, where the chromosomes are much longer, the number of crossing points may be increased in order to examine how it affects the GA performance.

The results of the multi-drug screening experiment are listed in Appendix B2.1, whereas the analysis of these results is presented in Figure 6.1.

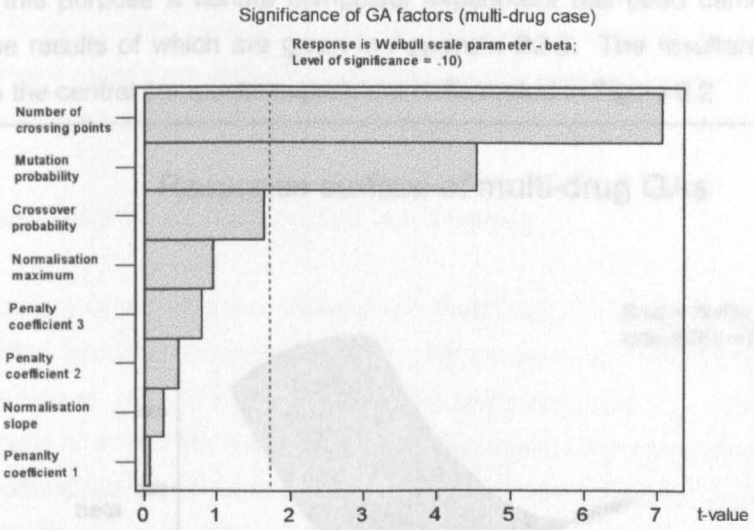


Figure 6.1. MINITAB analysis of the screening experiment (multiple drugs)

This analysis reveals that in the case of multi-drug treatment optimisation there are also two GA factors significantly affecting the performance measure (the scale parameter of the results' distribution). However, the set of significant factors is different – the probability of crossover, which was significant in the single-drug case, now gives way to the number of crossing points; the mutation probability still retains its significance. The significance of the number of crossing points for the crossover operator implies that in the case of multi-drug chemotherapy optimisation it is important to specify the right average length of treatment portions from which new treatment regimens are assembled. If this length is inappropriate, then the juxtaposition procedure implemented by crossover does not lead to treatment improvement.

Another feature of multi-drug chemotherapy optimisation is that it is possible to group single-drug doses within GA chromosomes either by the dose number or by the drug number. In the former case all the single-drug doses delivered during the first treatment interval are listed first, then all the doses delivered during the second interval and so on. In the latter case, we first list all doses of the first drug delivered over the whole treatment period, then all doses of the second drug and so on. As was mentioned in Section 5.1.3, the effect of crossover on chemotherapy regimens depends on the allowed positions for crossing points. If crossing points can be anywhere along the chromosome length, then crossover may lead to the alteration of single-drug doses in addition to rescheduling the administration of drugs. By selecting the second grouping strategy and by confining crossing points only to the places which separate the doses of the same drug, it is possible to eliminate the effects of dose alteration and inter-drug exchange. A number of experiments have been carried out to see the benefits of the latter implementation of the crossover operator. These experiments have shown that as long as the number of crossing points is specified appropriately and the trade-off between the exploration of the search space and the exploitation of the accumulated information is achieved, the implementation differences do not play a significant role.

Having completed a screening experiment and having singled out significant GA factors, the next step to tuning these factors is to perform a regression analysis using a response surface design. For this purpose a central composite experiment has been carried out, the design matrix and the results of which are given in Appendix B2.2. The resultant response surface obtained from the central composite experiment is illustrated in Figure 6.2.

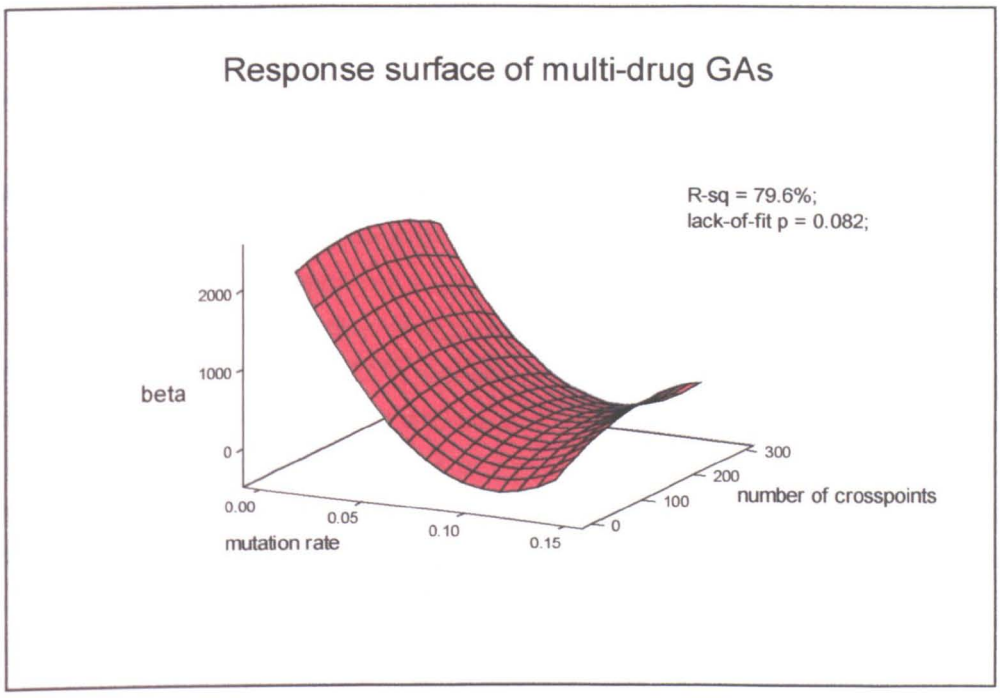


Figure 6.2. Response surface of the GA performance (multiple drugs)

The analysis of this response surface yields the following best setting of the significant GA factors in the specified domain:

$$\phi_4^{\text{opt}} = 0.062, \qquad \phi_8^{\text{opt}} = 2 \tag{6.20}$$

As may be seen from (6.20) the optimal mutation probability, which is a common GA significant factor for both the single-drug and the multi-drug optimisation problems, has decreased in value. One possible explanation for this is that only a certain number of genes need to undergo mutation to allow Genetic Algorithms to perform an efficient search through the solution space. Therefore, when the number of genes increases (chromosomes representing multi-drug treatments contain 10 times as many genes as their single-drug counterparts), the probability of mutation is bound to decrease in order to adhere to the Principle of Maximum Preservation (see Section 4.3.3).

The confirmation experiment testing the quality of the optimal setting (6.20) produces the data which can be summarised in the form of a numerical value of the performance measure ($\beta = 18.77$) and in the form of the 95% Wilcoxon confidence interval for the median value of Ψ: $[9,21.5]$. (Recall that the factors, which do not affect the GA performance, are assigned arbitrary values from the ranges specified in Table 6.1.) This summary indicates that the optimal factor setting for Genetic Algorithms tackling the multi-drug optimisation problem provides a reasonable level of search efficiency. Thus, the following GA factor set

$$\varphi_1 = 300,\ \varphi_2 = 50,\ \varphi_3 = 500,\ \varphi_4 = 0.062,\ \varphi_5 = 0.5,\ \varphi_6 = 500,\ \varphi_7 = 10,\ \varphi_8 = 2 \qquad (6.21)$$

will be used hereafter to run Genetic Algorithms for optimising multi-drug chemotherapy treatments. These factors completely specify the evolutionary procedures and the fitness function of Genetic Algorithms under investigation and prepare the ground for the experimental studies which comprise the material of the next section.

6.2. GA optimisation of multi-drug treatments

The binary representation of the concentration levels $C_{ij}, i = 1,n, j = 1,d$ adopted by the Genetic Algorithm encoding strategy (see (6.2)) allows each control variable C_{ij} to have 2^p distinct values, where p is the length of a binary string encoding C_{ij}. When the number of treatment intervals is n and the number of anti-cancer agents used in multi-drug chemotherapy is d, the solution space S contains 2^{pnd} possible treatment regimens. In the present work, a 4-bit representation has been adopted to encode the concentration levels of all drugs and we assume that the treatment period $[T_0, T_{\text{final}}]$ is divided into ten equal intervals, i.e. $n = 10$. Also, this study focuses around ten drugs ($d = 10$) commonly used in various treatments of breast cancer. These assumptions imply that the optimisation problem of multi-drug chemotherapy involves searching through the space of 2^{400} possible treatment regimens in order to find good drug administration regimens.

In this section the experimental results of GA optimisation are presented, where the optimisation is performed with respect to the three treatment objectives of multi-drug chemotherapy introduced in Chapter 1 and expressed as (6.8)-(6.10). For each optimisation objective Genetic Algorithms, operating a population of 50 chromosomes, run for 5000 generations and produced the regimens illustrated in Figures 6.3, 6.4 and 6.5. (NOTE. The program runs for 5000 generations because the statistical experiments with GA factors have shown that irrespective of the problem constraints and the settings of GA factor the improvement in the quality of the final GA solution after 5000 generations becomes negligible.)

Regimens A, B and C are designed to achieve specific treatment goals quantified by the objectives $\tilde{J}_1(\mathbf{c}), \tilde{J}_2(\mathbf{c})$ and $\tilde{J}_3(\mathbf{c})$ respectively. Furthermore, a subsequent application of the RM strategy to improve these regimens produces a desirable effect as may be seen from Table 6.2, which follows the graphs. The Regimen Modification program runs with the same setting of the GA factors as in (6.21) for 1000 generations. The modification of the best chemotherapy regimen found after 1000 generations does not lead to further significant improvement in treatment quality. Although the number of RM generations required to reach this saturation point can be lowered by readjusting the factor settings (6.21) to the Regimen Modification strategy, the effort involved in doing so was deemed unnecessary. The reason for this is that the RM program takes ~5-7 minutes to run 1000 generations, which on the GA time scale is not very long.

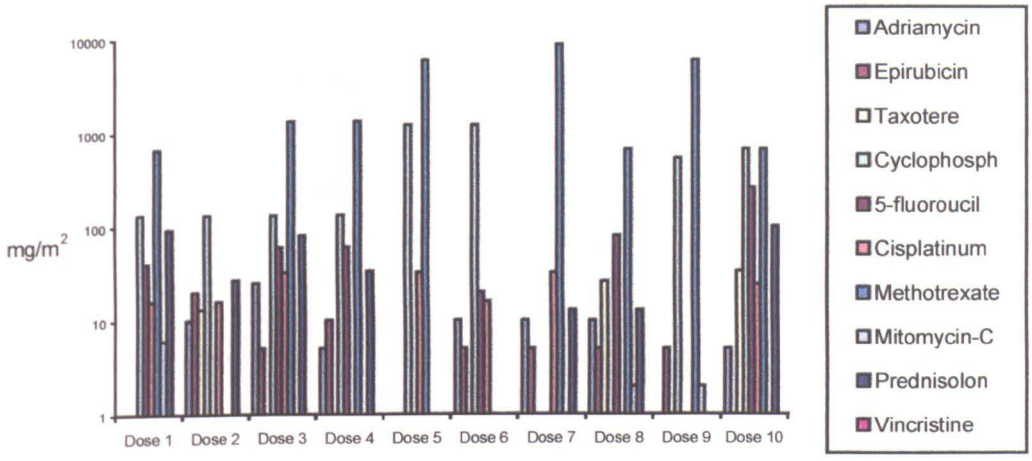


Figure 6.3. Treatment minimising tumour size (**Regimen A**)

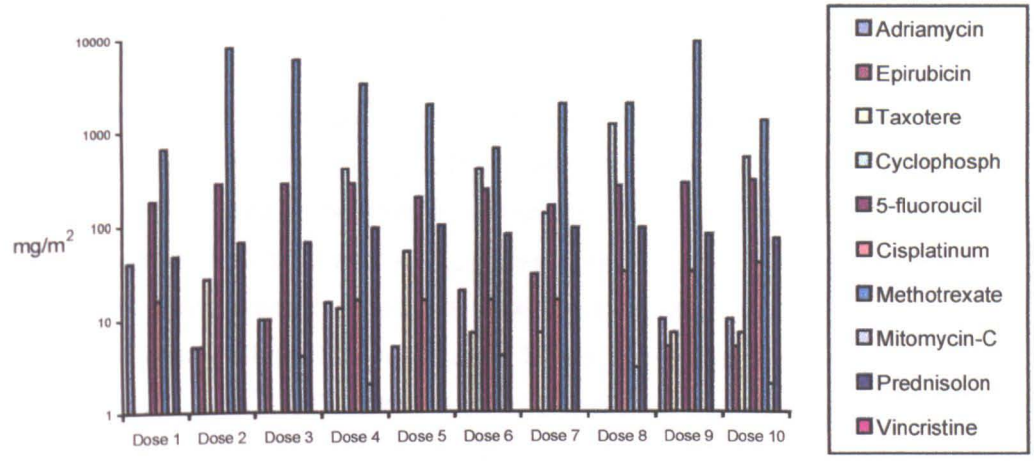


Figure 6.4. Treatment minimising tumour burden (**Regimen B**)

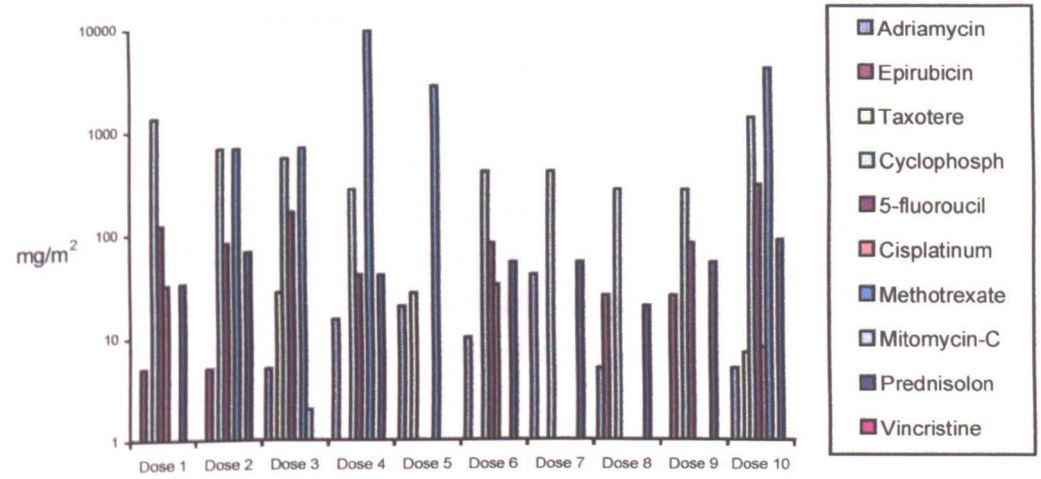


Figure 6.5. Treatment maximising survival time (**Regimen C**)

TABLE 6.2. *Characteristics of optimal treatments*

Optimal treatment	Stage of the optimisation process	$\widetilde{J}_1(\mathbf{c})$	$\widetilde{J}_2(\mathbf{c})$	$\widetilde{J}_3(\mathbf{c})$	$\frac{N(T_{\text{final}})}{N_0}$	$\frac{\sum_{i=1}^{10} N(t_i)}{10N_0}$	PST
Regimen A	before RM	1.2985	n/a	n/a	40.31%	82.77%	30 weeks
	after RM	1.3306	n/a	n/a	39.03%	82.48%	30 weeks
Regimen B	before RM	n/a	6.6432	n/a	42.67%	64.32%	28 weeks
	after RM	n/a	7.0683	n/a	40.75%	61.84%	29 weeks
Regimen C	before RM	n/a	n/a	0.3036	38.09%	82.66%	31 weeks
	after RM	n/a	n/a	0.3310	36.68%	79.45%	31 weeks

In Table 6.2 the third last column gives the value of the final tumour size $N(T_{\text{final}})$ as a percentage of the initial tumour size $N(T_0)$. The smaller this value the larger reduction of the tumour size has been achieved in the course of treatment. The second last column gives a more intricate measure of treatment success, expressed as a ratio of two numbers. The first number quantifies the shaded area in Figure 1.1. The second number is the area of a rectangle which has the height equal to the initial tumour size $N(T_0)$ and the width equal to the duration of the treatment period $T_{\text{final}} - T_0$. Therefore the ratio of these two numbers gives a measure of the overall tumour reduction. The last column shows the patient survival time.

As may be seen from Table 6.2, while attaining the intended effects, Regimens A, B and C perform reasonably well the optimisation of unintended treatment objectives. For instance, Regimen A, minimising the final tumour size, achieves approximately a 60% tumour reduction and at the same time prolongs PST up to 30 weeks. Regimen B brings down the overall tumour burden in the most effective way and also results in a reasonable reduction of $N(T_{\text{final}})$. Finally, Regimen C gives the best value of the patient survival time. Furthermore, applied to Regimen C, the RM optimisation strategy achieves the best improvement results. It reduces the final tumour size by an additional 2.22% (from 38.09% before RM to 36.68% after RM) and brings down the overall tumour burden by a further 3.4% (from 82.66% before RM to 79.45% after RM).

In order to assess the quality of the regimens found by Genetic Algorithms, we shall compare them with a couple of actual, known chemotherapy regimens. CAF (the combination of Cyclophosphomide, Adriamycin and 5-fluorouracil) and CMF (the combination of Cyclophosphomide, Methotrexate and 5-fluorouracil) are two multi-drug chemotherapy regimens

which are very often used for treating breast cancer. The listings of these regimens are given in Dearnaley *et al* (1995). Having evaluated CAF and CMF using the same measures as for Regimens A, B and C, we obtained the results presented in Table 6.3, which also shows how these results have been improved after applying the RM strategy to the chemotherapy regimens in question.

TABLE 6.3. *Characteristics of the known treatments CAF and CMF*

Treatment Regimen	Stage of the optimisation process	Reduction of the		PST
		final tumour size	overall tumour burden	
CAF	before RM	76.08%	75.85%	28 weeks
	after RM	63.26%	70.15%	28 weeks
CMF	before RM	78.34%	82.18%	26 weeks
	after RM	67.81%	79.43%	27 weeks

Comparing the figures in Table 6.2 and Table 6.3 we can see that the chemotherapy regimens found by Genetic Algorithms noticeably outperform CAF and CMF with respect to all treatment objectives. Furthermore, the application of RM to the latter regimens yields more substantial improvement in treatment quality than that of the regimens found by GAs. For the CAF therapy RM has managed to achieve additional 13.18% reduction of the final tumour size and as such strongly support the use of the evolutionary approach to improving known chemotherapy schedules.

Although potentially autonomous and self-controlled, the computational program implementing GA optimisation does not provide on its own a user-friendly environment for composition, evaluation and optimisation of different treatment regimens. However from the oncology standpoint, such an environment is highly desirable since it allows communication with the computer in a language appropriate to the application domain and to refrain from going into all technicalities of mathematical modelling and optimisation. Therefore, to facilitate the usage of the results of the present work in clinical optimisation of chemotherapeutic treatments, Genetic Algorithms need to be embedded into a more user-oriented software system. An example of such system is the Oncology Workbench, the prototype of which was developed by Boyle *et al* (1997). The Workbench was designed to be easily extensible, so that new components can be added to widen the system's functionality. Since the original version of the Workbench did not have an optimisation facility, the development of an optimisation component utilising Genetic Algorithms and the incorporation of this component into the system was a reasonable task to do. The details of how this task has been accomplished, as well as a brief description of the Workbench itself, are given in the following section.

6.3. Incorporating Genetic Algorithms into the Oncology Workbench

The Oncology Workbench has been designed to be a decision support system (Boyle *et al*, 1997) assisting the oncologists to efficiently evaluate various chemotherapeutic regimens proposed for cancer treatment and to effectively use existing anti-cancer drugs in novel combinations for cancer therapy. The requirement for a rational approach to the design of chemotherapeutic regimens is well established (Cassidy and McLeod, 1995). The intention of the Workbench therefore is to meet this requirement and to provide a convenient communication medium that allows the tools of mathematical modelling and optimisation to be delivered to the oncologists. The following sections give a more detailed description of how this medium is implemented and what role Genetic Algorithms play in it.

6.3.1. System architecture of the Oncology Workbench

The designers of the Workbench prototype put much effort in ensuring that the system is platform and location independent (Boyle *et al*, 1997). To the oncologists it is essential that the Workbench is available on a number of platforms and information can be accessed from a variety of different sites using a variety of means. Also, they wanted the system to be extensible so that other components (for example the GA optimisation module) can be easily fitted within the system architecture. Delivery of the Workbench over the WWW and using the JAVA language to build and link different components together enabled the designers to meet these requirements.

With JAVA it is possible to make the Workbench available through several Internet browsers – Netscape Navigator (v. 4.06) in particular. The common availability of the of web browsing tools allows the user to retrieve information about anti-cancer drugs, as well as to access remotely located modelling or optimisation modules, using a variety of different mechanisms. This makes the Oncology Workbench a convenient medium through which Genetic Algorithms can be brought to the forefront of modern cancer treatment.

6.3.2. Composition and functionality of the Oncology Workbench

Due to the multi-faceted nature of chemotherapy treatment it is unrealistic to expect that a fixed system design will satisfy every oncologist. A better methodology, therefore, is to encapsulate the whole functionality of the Oncology Workbench (OW) into smaller software components that can be flexibly interconnected in accordance with preferences of a particular professional. Such a methodology has been adopted and has resulted in the development of a system prototype containing the following components.

1) *Treatment Editor*

The treatment editor (TE) is an interactive interface allowing oncologists to graphically define a treatment regimen using available drugs in any combination and administered by employing permissible drug delivery modes. TE keeps track of toxicity information and advises oncologists on possible problems. It is achieved by utilising the $C_{\max}$ and C_{cum} indicators together with side-effect toxicity bars. The toxicity bars visually display on a qualitative colour scale the additive side effects induced by multiple drug administration (see Figure 6.6). Each toxicity bar corresponds to a particular side effect and gradually turns from white to black as the risk of causing this side effect increases. Figure 6.6 depicts the toxicity bars for such side effects as bone marrow depletion, nausea and kidney damage. The scrolling bar on the right side of these indicators allows examination of other toxic side effects. Therefore, the purpose of the aforementioned elements related to toxicity is to ensure that the user is warned about potential toxicity problems that might threaten the wellbeing of the patient.

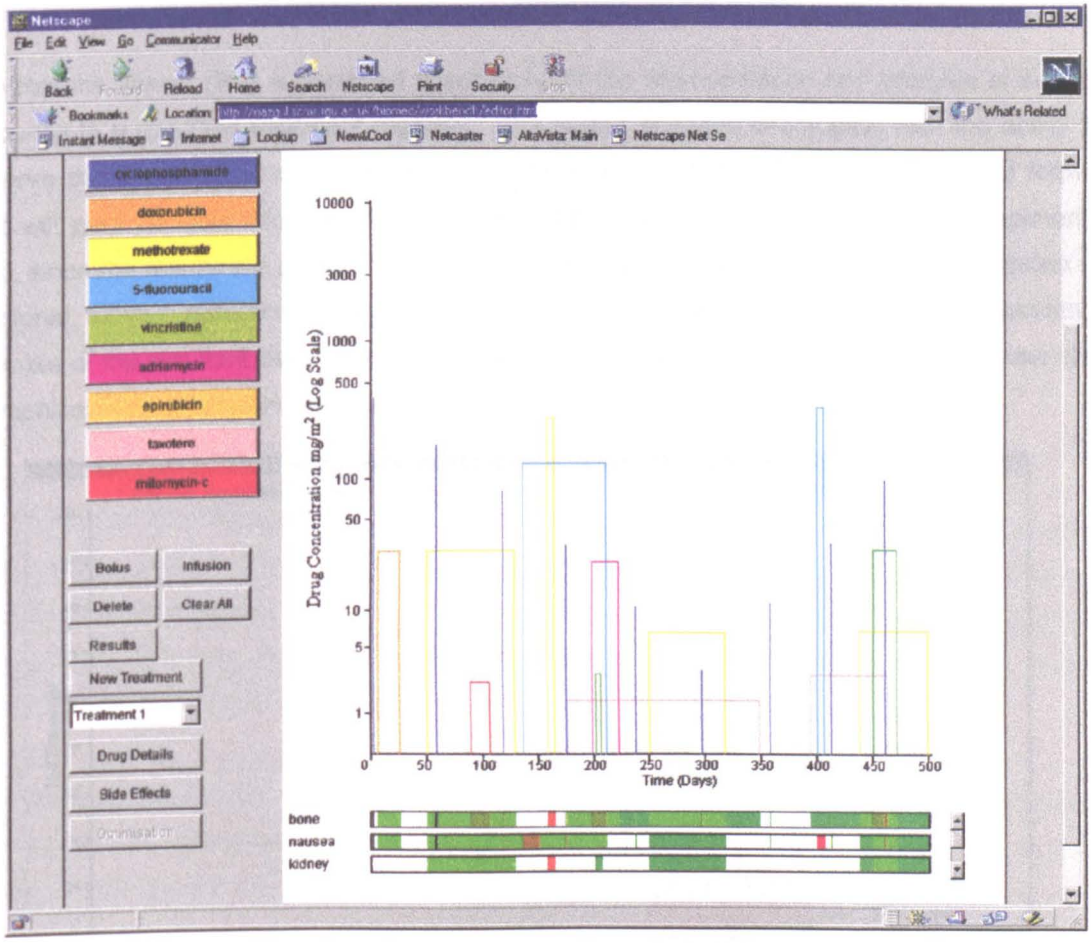


Figure 6.6. Treatment Editor Applet

When a treatment has been designed, the user faces a choice among three options. First of all, it is possible to utilise TE for sending the treatment data to a Results Viewer, which uses the outcome of a Simulation Engine to describe the resultant tumour behaviour. This description consists of tumour response statistics, a spreadsheet view and a graphical view of tumour development. Secondly, the treatment data can be sent to the Optimisation Engine, which will

return an optimal treatment for the drug combination used. Alternatively, the treatment profile can be stored for future reference, analysis or development.

2) Simulation Engine

Upon receiving the regimen composed in TE, the simulation engine of the OW retrieves appropriate information regarding the efficacy and toxicity of the anti-cancer drugs used. Then, it performs the simulation of the tumour response to this regimen over a normally distributed range of tumour growth parameters. To implement this task, the simulation engine uses the Gompertz tumour growth model with a linear cell-kill term incorporated (see Equation (5.5)). Also, the system has already been tested on alternative models and shown satisfactory performance (Henderson, 1997). After accomplishing the simulation, the results are sent to the Results Viewer for display.

3) *Results Viewer*

The Results Viewer (RV) is designed specifically for the representation and analysis of tumour response to a specified treatment regimen. The output provides oncologists with the ability to observe the resultant tumour size throughout treatment in both tabular and graphical forms. This will yield valuable information about tumour behaviour under various treatment regimens. Also, since the results are given in tabular form, it is straightforward to compute the statistics of fractional tumour reductions (Henderson *et al*, 1997). Figure 6.7 presents an illustrative example of the resultant data consisting of tumour response statistics, a spreadsheet view and a graphical view of tumour development.

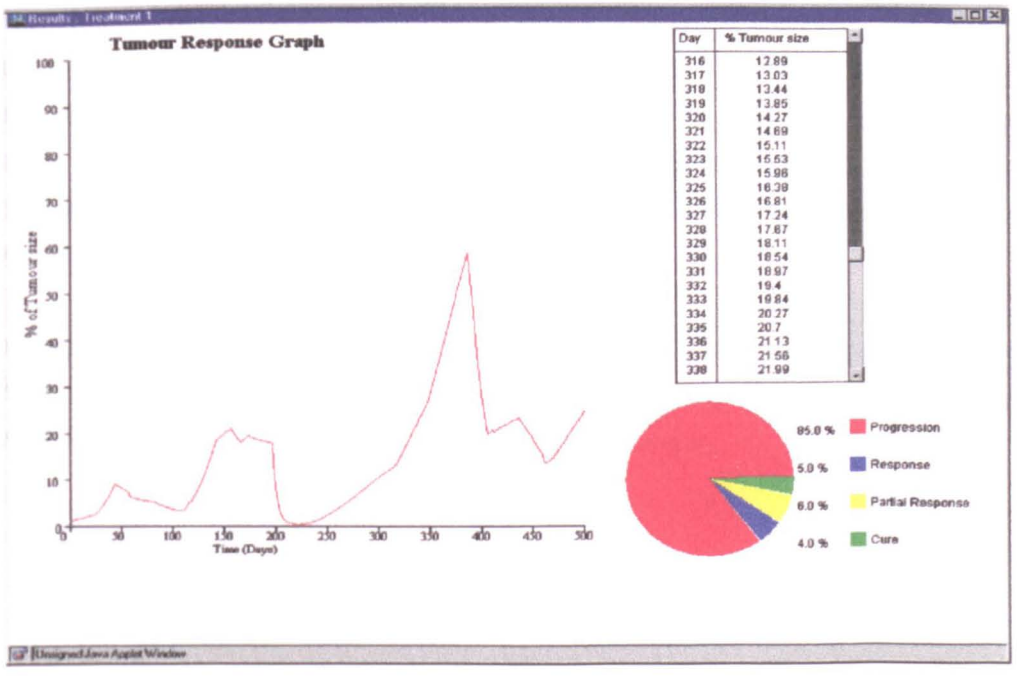


Figure 6.7. Results Viewer Applet

Finally, if all calculation procedures are performed in real time, the oncologist will be able to dynamically alter various parameters (e.g. dosages, drug effectiveness, growth rates), immediately seeing their effect on the treatment outcome. Also, comparative analysis may be performed simultaneously for different treatment strategies giving both qualitative and quantitative evaluations. Having completed this comparison, the oncologist may store its results in the Information Repository or may print them for future reference.

4) *Optimisation Engine*

The optimisation engine (OE) of the Workbench has the ability to suggest novel chemotherapeutic regimens which, being within the boundaries of clinical acceptability, are likely to produce best treatment outcomes. As a result of this, the oncologists can focus on the most promising treatment options, so that the costs, both human and financial, of the clinical research can be significantly reduced. However, there are too many combinations of anti-cancer drugs, as well as dosage values and timings, that have never been explored. Moreover, new anti-cancer agents are continually being developed. As the number of possible treatments increases exponentially, there is need for methods to intelligently search for those which seem promising. This is the area where the method of GA optimisation comes into play since the present study has vindicated the appropriateness and suitability of this method for finding good treatments in a robust and reliable manner. The development of an optimisation utility on the basis of Genetic Algorithms and the incorporation of this utility in the form of an optimisation module, comprise the author's contribution to the extension of the Oncology Workbench functionality and will be discussed in more detail in the next section.

5) *Information Repository*

The information repository (IR) allows the user to input, review, store and share information related to both simulated and real clinical data. This information covers the toxic side effects, the dosage limitations, the efficacy and the standard administration profiles of existing anti-cancer drugs. The role of the Information Repository is to supply necessary data to other components of the system.

Now, having introduced the constituent components, we can proceed with the description of the functionality of the Oncology Workbench. Figure 6.8 illustrates how different components of the Workbench interact and shows the information flow between them.

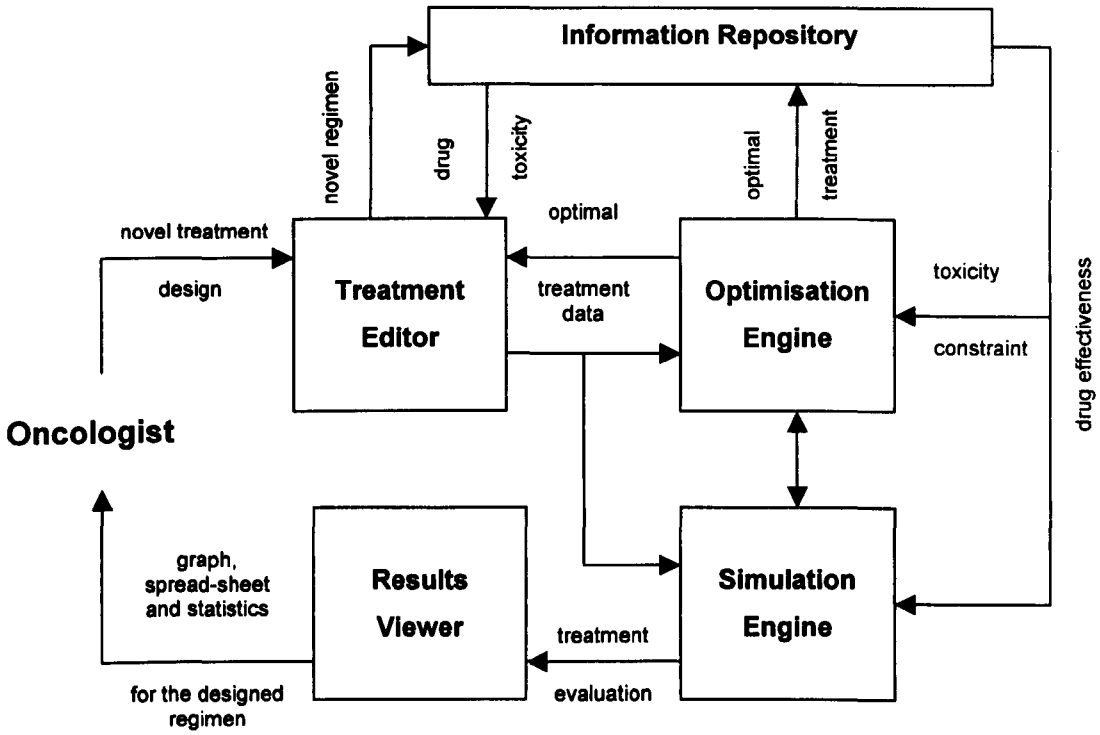


Figure 6.8. Interaction of the Oncology Workbench components

As may be seen from this diagram, the oncologist operating on the Workbench deals only with the Treatment Editor and the Results Viewer, which play a role of interface objects between the user and the system. A novel treatment regimen composed by the user with the help of TE is sent simultaneously to SE and to OE (optional). Having received treatment data, both engines call IR for additional information concerning the effectiveness and toxicity constraints of the anti-cancer drugs used. After obtaining this information, SE evaluates the newly composed treatment regimen and sends the results to RV. Optimisation engine, on the other hand, either advises the user via TE what the best possible treatment strategy would be subject to particular constraints, or stores this strategy in IR for future references.

Therefore, having composed a novel treatment regimen on the Workbench, the oncologist gets feedback from the system via two parallel channels. The main channel is the Results Viewer, which presents all the information related to tumour behaviour. In addition, the Treatment Editor itself has the online ability to monitor the levels of toxic side effects and represent them dynamically on coloured toxicity bars. Also, TE is used for displaying the optimal treatment regimen found by the Optimisation Engine using Genetic Algorithms. The next section gives the particulars of how this optimisation process is performed.

6.3.3. GA optimisation of chemotherapy treatments

From a practical standpoint all cancer chemotherapy treatments may be viewed as either curative or palliative (McCall and Petrovski, 1998). In a curative treatment, the aim is to drive down tumour size to a point where the tumour either disappears or other actions can be taken to remove it. The objectives (6.8)-(6.9) of minimising tumour size by the end of treatment and minimising the overall tumour burden correspond to this case. Palliative treatment, on the other hand, is applied where the tumour is deemed to be incurable. Here the objective is to maintain a reasonable quality of life as long as possible. One possible way to mathematically express the latter objective is to maximise PST in the form of (6.10).

The distinction between curative and palliative treatments is made because their objectives are understood to be in conflict with each other in the sense that regimens which tend to minimise tumour size are highly toxic and therefore have negative effect on the quality of life. Moreover, as smaller tumours tend to grow more rapidly, in the presence of drug resistance a severe regimen can result in the ultimate shortening of patient life because a drug resistant cell population accumulates to a fatal size earlier than it otherwise might have (Martin *et al*, 1992).

The primary objective of the developed optimisation engine inside the Workbench is to help the oncologists to find a curative treatment (McCall and Petrovski, 1998). Therefore, the first optimisation step is to apply the GA strategy to find treatment regimens which minimise the objectives (6.8)-(6.9). If the values of these objectives indicate that the best treatment found by GAs can be considered as curative, then the second step will be taken by applying the RM strategy to improve this treatment. Otherwise, the GA strategy will be applied again to find a good palliative treatment, which will be subsequently enhanced by RM. The optimisation process as a whole is illustrated in Figure 6.9.

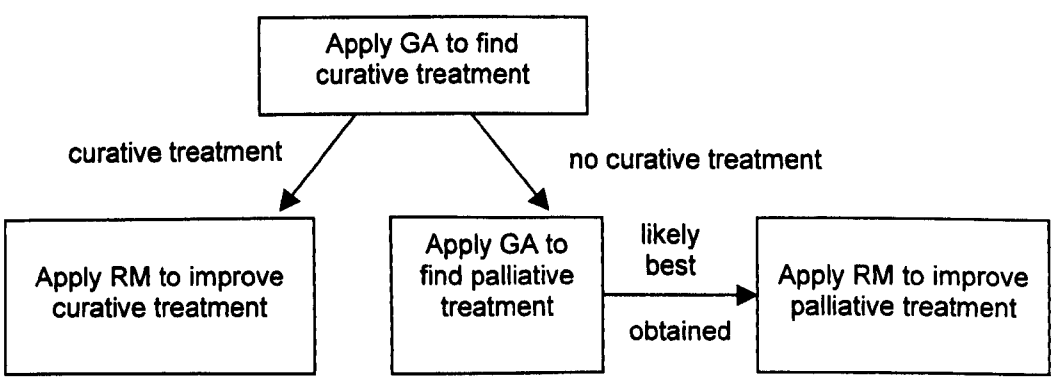


Figure 6.9. GA optimisation of chemotherapeutic treatments

To conclude this section devoted to the issue of GA usage in clinical practice, a few words ought to be said about how the developed OE is integrated with the rest of the Workbench. As we mentioned earlier, a request to activate the optimisation procedure is received from the Treatment Editor, which also specifies the drugs selected by the oncologist for treatment.

Subsequently the optimisation engine queries the Information Repository to obtain the characteristics of these drugs (e.g. maximum toxicity dosages, effectiveness and potency factors). Then, OE initiates the optimisation process (epitomised by Figure 6.9) using the simulation engine to evaluate candidate treatment regimens encountered during the GA search. The search is carried out until a predefined stopping condition is met and culminates in the form of the best chemotherapeutic treatment found by Genetic Algorithms. This treatment is passed back to the Treatment Editor and displayed for analysis or storing.

6.4. Discussion

In this chapter it has been shown how Genetic Algorithms can be applied to the optimisation of multi-drug chemotherapy. The majority of chemotherapy treatments used in clinical practice nowadays utilise multiple anti-cancer agents since drug cocktails tend to be more effective in achieving various treatment objectives. These objectives include not only the minimisation of tumour size by the end of treatment (thoroughly studied in Chapter 5), but the minimisation of the overall tumour burden and the prolongation of PST as well. The additional constraint on concurrent administration of multiple drugs and more complex optimisation objectives make the problem of multi-drug chemotherapy non-amenable to LP and extremely difficult for conventional methods of constrained optimisation. Genetic Algorithms, on the other hand, proved to be capable of dealing with this problem. Viable treatment strategies have been found by GAs for both curative and palliative treatment scenarios, thereby confirming the suitability and effectiveness of Genetic Algorithms for cancer chemotherapy optimisation.

On the basis of this conclusion, a decision has been made to encapsulate Genetic Algorithms into an independent optimisation module and to embed this module into a more general and user-oriented environment. The Oncology Workbench – the system aimed at helping the oncologists in the decision making activity and developed by Boyle *et al* (1997) – has been chosen as the most suitable environment for the embedding of Genetic Algorithms. The facilities provided by the Workbench may assist the oncologists in: (1) expediting clinical drug trials, (2) guiding more cost effective experimentation, and (3) suggesting optimal treatment regimens within the bounds of clinical acceptability. Moreover, the Oncology Workbench is an autonomous, versatile and customisable decision support system which can be also used for optimising drug treatments unrelated to cancer (e.g. cardiovascular or diabetic therapies). This creates an opportunity to widen the application domain of Genetic Algorithms within the field of drug treatments and to allow more clinicians to benefit from utilising the GA optimisation.

CHAPTER 7

CONCLUSIONS

As may be seen from the present treatise, the current work has a multi-faceted nature, i.e. several issues have been addressed in this thesis which arise from a number of disciplines such as: biology and medicine, non-linear control and mathematical modelling, optimisation and search techniques, stochastic analysis and computer science. We started with realising the complexity of the cancer phenomenon and showed why mathematical modelling of cancer development and its treatment can be of a great assistance here. While discussing the treatment of cancer, the relevance of optimisation and optimal control theory became evident for achieving the most effective usage of available anti-cancer drugs within the boundaries of clinical acceptability. However, the multi-constraint nature of cancer chemotherapy and general non-linearity of the optimisation functionals attributed to the objectives of cancer treatment often make the problem of cancer chemotherapy optimisation analytically intractable.

Therefore a change in approach to the optimisation of cancer chemotherapy is often necessary if a practical solution to the problem is to be found. In the present work a new approach has been opted for, whereby the optimisation of chemotherapeutic treatment as a whole has been substituted by a search for treatment regimens which are "better" than those already known. To implement this search, the GA technique has been chosen as the most robust and versatile method and has been embedded into a generic framework – the Oncology Workbench – which allows oncologists to interactively define, evaluate and optimise simulated drug treatments. The summary of key issues of the application of Genetic Algorithms to cancer chemotherapy optimisation and the outline of possible directions for future work are presented in the following two sections. Thus, Section 7.1 addresses the issues which, in being resolved, have displayed innovative aspects or have led to original contributions to existing knowledge. The final section of the thesis, Section 7.2, focuses on future work.

7.1. Summary of key issues

In the research programme for the present study the following aims were formulated as the PhD objectives:

1) to apply the Genetic Algorithm technique to mathematical models of tumour growth in order to improve existing chemotherapy treatment regimens;
2) to apply the GA technique, *ab initio*, to develop optimal treatment regimens;
3) to develop an optimisation utility for a clinically usable decision support system.

Firstly, the GA technique has been successfully applied to both scenarios of chemotherapy optimisation, i.e. when the optimal treatment strategy is obtained by improving existing treatment regimens and when the optimal regimen is developed from scratch. In the case of single-drug chemotherapy, Genetic Algorithms showed a good optimisation performance compared with that of traditional methods. More importantly, the GA approach has also yielded satisfactory results in the case of multi-drug chemotherapy, where none of the known mathematical optimisation methods were effective.

The second major contribution of the thesis is the development and incorporation of the GA optimisation utility into the Oncology Workbench. The aim of the Workbench is to overcome the paucity of objective tools for intelligent design and optimisation of chemotherapeutic regimens and to simplify for oncologists the decision making activity. It is self-evident that such a system needs the presence of a robust and effective utility responsible for executing all optimisation procedures, some of which are multi-constrained and intricate. The experimental results of the last two chapters unmistakably demonstrate that the evolutionary optimisation featured by Genetic Algorithms is suitable for this role, thereby justifying the use of the GA optimisation component within the Workbench.

In addition, a new GA factor tuning method has been developed, tested and analysed which can be used for enhancing the optimisation performance of Genetic Algorithms. This method is based on fractional factorial experiments with GA factors and on the statistical analysis of how these factors affect the performance. The above summarises the contribution made by the thesis to the existing knowledge of effective chemotherapy treatment and highlights various innovative aspects.

7.2. Future work

Although achieving its main objectives, this thesis brings forth a whole series of other issues. All the way through this work we maintained the assumptions that the tumour cell population is homogeneous and that the representation of the bloodstream as a single compartment allows one to express the drug concentration in the blood plasma in terms of the delivery dose without delving into pharmacological details. Also it has been assumed that the cell-kill rates of all drugs remain constant throughout treatment and that interactions between drugs within compound chemotherapeutic doses are negligible. Needless to say however, these

assumptions are not always valid. First of all, the cellular heterogeneity may lead to the appearance of drug resistant cancerous cells and, subsequently, to treatment failure. To take this heterogeneity into account, the Gompertz model of tumour growth needs to be replaced by a more accurate alternative (for example, by the Competition model developed by Usher and Henderson (1996)). This will be one route for future investigations.

Secondly, for a more precise estimation of the drug concentration in the blood and for a more realistic description of drug-drug interactions, one has to use the techniques of PK/PD modelling. The utilisation of the PK/PD analysis for optimising cancer chemotherapy treatment is another area where further work is required. Ideally, this will lead to the development of a PK/PD modelling component for the Oncology Workbench. The presence of such a component will enable oncologists to take a closer look at the processes occurring during chemotherapy treatment and to gain further insight into drug actions.

Thirdly, in the present work a number of quantitative measures were introduced to evaluate the quality of chemotherapeutic treatments and the goal of chemotherapy optimisation was to find such treatment regimens which result in the best values of these measures. Although clinically meaningful, the introduced measures are bound to the treatment objectives which assume that the treatment period is fixed. Fixing the length of treatment was necessary in the current work since all chemotherapeutic regimens were encoded as chromosomes with a predefined number of genes. In practical situations however the treatment duration often varies. If, for example, a curative treatment is found which reduces the tumour to a critical size at some point during treatment and maintains it at or below this value for a certain period, then further use of drugs might become unnecessary. This possibility arises due to the existence of other mechanisms (e.g. programmed cell death) capable of removing remaining tumour cells if the tumour is smaller than ~1000 cells. Thus, the ultimate treatment period might be shortened, yielding at the same time a more favorable treatment outcome. Alternatively, when the tumour is deemed to be incurable, a reasonable approach would be to palliate the suffering by extending the duration of treatment, thereby maintaining a tolerable quality of patient life until the tumour reaches its lethal size. Therefore, to endow Genetic Algorithms with the ability to dynamically adjust the length of chemotherapeutic treatments, the encoding and the recombination procedures need to be developed further. The future work directed towards this objective has the highest priority as far as the Oncology Workbench is concerned.

Finally, the techniques and various tools developed during the work on this project are by no means restricted to cancer chemotherapy optimisation alone. They can be adapted to other drug administration problems such as, for example, the optimisation of cardiovascular therapies or diabetic treatments. Therefore, the material of this thesis may be viewed as a foundation which other applications of effective and customisable treatment optimisation can be built up.

REFERENCES

1. BAECK T, FOGEL D and MICHALEWICZ Z (eds.), 1997. Handbook of Evolutionary Computation: Oxford University Press.
2. BAKER JE, 1985. Adaptive Selection Methods for Genetic Algorithms. Proceedings of the First International Conference on Genetic Algorithms: Hillsdale, NJ – Lawrence Erlbaum Associates.
3. BATTLE DL and VOSE MD, 1993. Isomorphisms of genetic algorithms. Artificial Intelligence, **60**, pp. 155-165.
4. BEASLEY D, BULL DR and MARTIN RR, 1993. An overview of genetic algorithms: Part 1, fundamentals and Part 2, research topics. University Computing, **15**, pp. 58-69 and pp. 170-181.
5. BEYER H-G, 1997. An alternative explanation for the manner in which genetic algorithms operate. BioSystems, **41**, pp. 1-15.
6. BIRKHEAD BG and GREGORY WM, 1984. A Mathematical Model of the Effects of Drug Resistance in Cancer Chemotherapy. Mathematical Biosciences, **72**, pp. 59-69.
7. BOOKER LB, 1987. Improving search in genetic algorithms. In: DAVIS L (ed.), 1987, Genetic Algorithms and Simulation Annealing: Pitman, pp. 61-73.
8. BOX G, HUNTER W, HUNTER JS, 1978. Statistics for experimenters. An introduction to design, data analysis and model building: New York – John Wiley & Sons.
9. BOYLE J, HENDERSON D, McCALL J, McLEOD H, USHER J, 1997. Exploring Novel Chemotherapy Treatments using the WWW. International Journal Of Medical Informatics, **47(1-2)**, pp. 107-114.
10. BRAMLETTE MF, 1991. Initialisation, mutation and selection methods in genetic algorithms for function optimisation. Proceedings of the Forth International Conference on Genetic Algorithms. Morgan Kaufmann, pp. 100-107.
11. BUNDAY BD and GARSIDE GR, 1987. Optimisation methods in Pascal: London – Edward Arnold.
12. BURGHES D and GRAHAM A, 1980. Introduction to Control Theory including Optimal Control: Ellis Horwood Series Mathematics and its Applications – John Wiley & Sons.
13. CASSIDY J and McLEOD HL, 1995. Is it possible to design a logical development plan for an anti-cancer drug. Pharmaceutical Medicine, **9**, pp. 95-103.
14. CHAPLAIN MAJ and SLEEMAN BD, 1993. Modelling the growth of solid tumours and incorporating the method for their classification using nonlinear elasticity theory. Journal of Mathematical Biology, **31**, pp.431-473.
15. COLLINS JM and DEDRICK RL, 1982. Pharmacokinetics of anti-cancer drugs. In: CHABNER BA (ed.), Pharmacologic Principles of Cancer Treatment: Philadelphia – Saunders, pp. 77-99.
16. CONGER AD and ZISKIN MC, 1983. Growth of mammalian Tumour Spheroids. Cancer Research, **43**, pp. 556-560.

17. COSTA MIS, BOLDRINI JL, BASSNEZI RC, 1992. Optimal chemical control of population developing drug resistance. IMA Journal of Mathematics Applied in Medicine & Biology, **9**, pp. 215-226.
18. CRAMER NL, 1985. A representation for the adaptive generation of simple sequential programs. Proceedings of an International Conference on Genetic Algorithms and Their Applications, pp.183-189.
19. DAVIS L, 1989. Adapting operator probabilities in genetic algorithms. Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann, pp. 61-69.
20. DAVIS L, 1991. Handbook of Genetic Algorithms: New York – Van Nostrand Reinfold.
21. DEARNALEY DP, JUDSON I and ROOT T, 1995. Handbook of adult cancer chemotherapy schedules. The Medicine Group (Education) Ltd., Oxfordshire.
22. DeJONG KA, 1975. An analysis of the behaviour of a class of genetic adaptive systems. Doctoral dissertation, University of Michigan.
23. DEB K, 1997a. Introduction to GA selection. In [1], Section C2.1.
24. DEB K, 1997b. Limitations of evolutionary computation methods. In [1], Section B2.9.
25. DEVORE JL, 1995. Probability and statistics for engineering and the science. 4th edn. Duxbury Press.
26. DRAIN D, 1997. Statistical Methods for Industrial Process Control: Chapman&Hall.
27. ESHELMAN LJ, 1997. Genetic Algorithms. In [1], Section B1.2.
28. FOGEL LJ, 1997. Future research in evolutionary computation. In [1], Section H1.2.
29. FREISLEBEN B and HARTFELDER M, 1993. Optimisation of Genetic Algorithms by Genetic Algorithms. The 1993 International Conference on Artificial Neural Nets and Genetic Algorithms: Vienna – Springer-Verolag, pp. 393-399.
30. FREISLEBEN B, 1997. Metaevolutionary approaches. In [1], Section C7.2.
31. GHANNADIAN F, ALFORD C and SHONKWILER R, 1996. Application of Random Restart to Genetic Algorithms. Information Science, **95**, pp. 81-102.
32. GOLDBERG DE, 1985. Alleles, loci, and the TSP. Proceedings of the First International Conference on Genetic Algorithms. Lawrence Erlbaum, pp. 154-159.
33. GOLDBERG DE, 1989. Genetic Algorithms in search, optimisation and machine learning: Reading, Mass. – Addison-Wesley.
34. GOLDBERG DE and DEB K, 1991. A comparative analysis of selection schemes used in genetic algorithms. In RAWLINGS G (ed.), Foundation of Genetic Algorithms. Morgan Kaufmann.
35. GOLDIE JH and COLDMAN AJ, 1979. A mathematical model for relating the drug sensitivity of tumours to their spontaneous mutation rate. Cancer Treatment Reports, **63**, pp. 1727-1733.
36. GOLDIE JH, COLDMAN AJ and GUDAUSKAS GA, 1982. Rationale for the use of alternating non-cross resistant chemotherapy. Cancer Treatment Reports, **66**, pp. 439-449.
37. GREFENSTETTE JJ, 1986. Optimisation of Control Parameters for Genetic Algorithms. IEEE Transactions on Systems, Man and Cybernetics, **16(1)**, pp.122-128.

38. HARNEVO LE and AGUR Z, 1991. The Dynamics of Gene Amplification Described as a Multitype Compartmental Model and as a Branching Process. Mathematical Biosciences, **103**, pp. 115-138.

39. HARNEVO LE and AGUR Z, 1993. Use of mathematical models for understanding the dynamics of gene amplification. Mutation Research, **292**, pp. 17-24.

40. HATTA K, MATSUDA K, WAKABAYASHI S and KOIDE T, 1997. On-the-Fly Crossover Adaptation of Genetic Algorithms. Proceedings of the Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications: University of Strathclyde – Glasgow, U.K. September 1997, pp. 197-202.

41. HENDERSON D, McCALL J, BOYLE J, 1997. Decision Support System for Designing Chemotherapeutic Regimens. Proceedings of the 14th International Congress on Health Care in the Information Highways: IOS Press, pp. 353-358.

42. HENDERSON D, 1997. Mathematical modelling in the scheduling of cancer chemotherapy treatment when drug resistance is present. Ph.D. thesis, The Robert Gordon University.

43. HOLLAND J, 1992. Adaptation in natural and artificial systems. 2nd edn. Massachusetts Institute of Technology.

44. IBARAKI T, 1997. Introduction to Combination of Genetic Algorithms with other Optimisation Methods. In [1], Section D3.1.

45. JOHNSON RA and BHATTACHARYYA GK, 1992. Statistics: Principles and Methods: John Wiley and Sons, 2nd edn.

46. LANDRY J, FREYER JP and SUTHERLAND RM, 1982. A model for the growth of multicellular spheroids. Cell Tissue Kinetics, **15**, pp. 585-.

47. LUCASIUS CB and KATEMAN G, 1993. Understanding and using genetic algorithms. Part 1. Concepts, properties and context. Chemometrics and Intelligent Laboratory Systems, **19**, pp.1-33.

48. LUCASIUS CB and KATEMAN G, 1994. Understanding and using genetic algorithms. Part 2. Representation, configuration and hybridisation. Chemometrics and Intelligent Laboratory Systems, **25**, pp. 99-145.

49. MARTIN RB, FISHER ME, MINCHIN RF and TEO KL, 1990. A mathematical model of cancer chemotherapy with an optimal selection of parameters. Mathematical Biosciences, **99**, pp. 205-230.

50. MARTIN RB, FISHER ME, MINCHIN RF and TEO KL, 1992. Optimal control of tumour size used to maximise survival time when cells are resistant to chemotherapy. Mathematical Biosciences, **110**, pp. 201-219.

51. MARTIN RB and TEO KL, 1994. Optimal control of drug administration in cancer chemotherapy: Singapore, New Jersey, London, Hong Kong – World Scientific.

52. MARUSIC M, BAJZER R, FREYER JP and VUK-PAVLOVIC S, 1994. Analysis of growth of multicellular tumour models by mathematical methods. Cell Proliferation, **27**, pp. 73-94.

53. McCALL JAW and PETROVSKI AV, 1996. Searching for optimal strategies in cancer chemotherapy using genetic algorithms. Presented at 7th IMA Conference on Mathematics in Medicine and Biology.

54. McCALL J and PETROVSKI A, 1999. A Decision Support System for Cancer Chemotherapy using Genetic Algorithms. Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation: Vienna, Austria. IOS Press, ISBN 90 5199 474 5, pp. 65-70.
55. MICHALEWICZ Z, 1992. Genetic Algorithms + Data Structure = Evolution programs. Springer-Verlag.
56. MITCHELL M, 1996. An introduction to Genetic Algorithms: Cambridge, Mass. – MIT Press.
57. MURRAY JM, 1990a. Optimal Control for a Cancer Chemotherapy Problem with General Growth and Loss Functions. Mathematical Biosciences, **98**, pp. 273-287.
58. MURRAY JM, 1990b. Some optimal control problems in cancer chemotherapy with a toxicity limit. Mathematical Biosciences, **100**, pp. 49-67.
59. NIX AE and VOSE MD, 1991. Modelling genetic algorithms with Markov chains. Annals of Mathematics and Artificial Intelligence, **5**, pp. 79-88.
60. NORTON I and SIMON R, 1986. Cancer treatment Reports, **70**, pp. 163-169.
61. PARRILL AL, 1996. Evolutionary and Genetic Methods in Drug Design. Drug Discovery Today, 1(12), pp. 514-521.
62. PHAM QT, Competitive evolution: a natural approach to operator selection. Progress in Evolutionary Computation (Lecture notes in Artificial Intelligence 956), pp. 49-60.
63. PERRY CP (ed.), 1992. The chemotherapy sourcebook. Williams and Williams: Maryland.
64. PETROVSKI A and McCALL J, 1997. Optimising GA parameters using statistical approaches. Proceedings of the First International Workshop on Frontiers in Evolutionary Algorithms. North Carolina, USA, March 1997.
65. PETROVSKI A, McCALL J and FORREST E, 1998a. An Application of Genetic Algorithms to Optimisation of Cancer Chemotherapy. International Journal of Mathematical Education in Science and Technology, **29**(3), pp. 377-388.
66. PETROVSKI A, WILSON A and McCALL J, 1998b. Statistical analysis of genetic algorithms and inference about optimal parameters. The paper is submitted for possible publication in the Journal of Applied Statistics.
67. PRUGEL-BENNET A and SHAPIRO JL, 1994. Analysis of Genetic Algorithms Using Statistical Mechanics. Physical Review Letters, **72(9)**, pp. 1305-1309.
68. RADCLIFFE NJ, 1997. Schema Processing. In [1], Section B2.5.
69. RAO SS, 1978. Optimisation: Theory and Applications: Wiley Eastern Ltd.
70. RAWLING G (ed.), 1991. Foundation of Genetic Algorithms. Morgan Kaufmann.
71. RICHARDSON JT, PALMER MR, LIEPINS G and HIILIARD M, 1989. Some Guidelines for Genetic Algorithms with Penalty Functions. Proceedings of the Third International Conference on Genetic Algorithms: Los Altos – Morgan Kaufmann Publishers.
72. SCHAFFER JD, CARUANA RA, ESHELMAN LJ and DAS R, 1989. A study of control parameters affecting online performance of genetic algorithms for function optimisation. Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann.

73. SCHWEFEL H-P, 1995. Evolution and Optimum Seeking: New York etc. – John Wiley & Sons, Inc.
74. SKIPPER HE, SCHABEL FM and WILCOX WS, 1964. Experimental evaluation of potential anti-cancer agents. XIII. On the criteria and kinetics associated with "curability" of experimental leukemia. Cancer Chemotherapy Reports, **61**, pp. 1307-1317.
75. SMITH AE and COIT DW, 1997. Penalty functions. In [1], Section C5.2.
76. SPEARS WM and DEJONG KA, 1991. On the virtues of parameterised uniform crossover. Proceedings of the Forth International Conference on Genetic Algorithms. Morgan Kaufmann.
77. STEEL GG, 1977. Growth Kinetics of Tumours: Oxford – Clarendon.
78. SULLIVAN PW and SALMON SE, 1972. Kinetics of tumour growth and regression in IgG multiple myeloma. Journal of Clinical Investigations, **51**, pp. 1697-1708.
79. SWAN GW and VINCENT TL, 1977. Optimal control analysis in the chemotherapy of IgG multiple myeloma. Bulletin of Mathematical Biology, 39, pp. 317-337.
80. SWAN GW, 1990. Role of Optimal Control theory in Cancer Chemotherapy. Mathematical BioScience, **101**, pp. 237-284.
81. SYSWERDA G, 1991. Schedule optimisation using genetic algorithms. In: DAVIS L (ed.), 1991, Handbook of Genetic Algorithms.
82. TUSON AL and ROSS P, 1996. Co-evolution of operator settings in genetic algorithms. Proceedings of 1996 AISB Workshop on Evolutionary Computing, Briton, UK, pp. 120-128.
83. USHER JR, 1994. Some Mathematical Models for Cancer Chemotherapy. Computers and Mathematics with Applications, **28(9)**, pp. 73-80.
84. USHER JR and HENDERSON D, 1996. Some drug-resistant models for cancer chemotherapy. Part 1: Cycle-nonspecific drugs. IMA Journal of Mathematics Applied in Medicine and Biology, **13**, pp. 99-126.
85. VARMUS H and WIENBERG, 1993. Genes and the biology of cancer: New York – Scientific American Library.
86. VOSE MD, 1991. Generalising the notion of schema in genetic algorithms. Artificial Intelligence, **50**, pp. 385-396.
87. VOSE MD and LIEPINS GE, 1991. Punctuated equilibria in genetic search. Complex Systems, **5**, pp. 31-44.
88. VOSE MD and WRIGHT AH, 1995. Stability of Vertex Fixed Points and Applications. Evolutionary Computation, **2(4)**, pp. 347-368.
89. WADSWORTH HM, 1998. Handbook of Statistical Methods for Engineers and Scientists: McGraw-hill, 2nd edn.
90. WHELDON TE, 1988. Mathematical models in cancer research: Bristol and Philadelphia – Adam Hilger.
91. WHITLEY LD, MATHIAS K and FITZHORN P, 1991. Delta Coding: An iterative search strategy for genetic algorithms. Proceedings of the Forth International Conference on Genetic Algorithms: Los Altos, CA – Morgan Kaufmann Publishers.
92. WHITLEY LD (ed.), 1993. Foundation of Genetic Algorithms 2. Morgan Kaufmann.

93. WHITLEY LD and VOSE MD (eds.), 1995. Foundation of Genetic Algorithms. Morgan Kaufmann.
94. WIDNELL CC and PFENNINGER KH, 1990. Essential cell biology. Williams & Williams.
95. ZEITZ S and NICOLINI C, 1979. Mathematical approaches to optimisation of cancer chemotherapy. Bulletin of Mathematical Biology, **41**, pp. 305-324.

APPENDIX A

FITNESS FUNCTIONS

A1. Single-drug chemotherapy

(See Sections 5.1.1 and 5.1.2)

Solution space: $S = \left\{\mathbf{c} : \mathbf{c} = (C_i), i = \overline{1,n}\right\}$

Representation spaces:

1. $I_{\mathrm{GA}} = \left\{a_1 a_2 a_3 \ldots a_{4n} : a_k \in \{0,1\} \ \forall k = \overline{1,4n}\right\}$
2. $I_{\mathrm{RM}} = \left\{a_1 a_2 a_3 \ldots a_n : a_k \in \{0,1,-1\} \ \forall k = \overline{1,n}\right\}$

Fitness functions:

Tumour size optimisation: $\underset{\mathbf{c}=(C_i)}{\text{maximise}} \ J(\mathbf{c}) = \sum_{i=1}^{n} C_i e^{\lambda(t_{i-1}-T_{\text{final}})}$

$$f_{\mathrm{GA}}(x \in I_{\mathrm{GA}}) = \sum_{i=1}^{n}\sum_{k=1}^{4} 2^{4-k} a_{4(i-1)+k} e^{\lambda(t_{i-1}-T_{\text{final}})} - P_2 \max^2\left\{\sum_{i=1}^{n}\sum_{k=1}^{4} 2^{4-k} a_{4(i-1)+k} - C_{\text{cum}}, 0\right\} - P_3 \sum_{i=1}^{n} e^{-\beta \cdot t_i} \max^2\{y_{\min} - y(t_i), 0\}$$

where $y(t_i) = e^{-\lambda t_i}\left\{y_0 + \frac{\kappa}{\lambda}\left(e^{\lambda \Delta t} - 1\right)\sum_{j=1}^{i}\sum_{k=1}^{4} 2^{4-k} a_{4(i-1)+k} e^{\lambda t_{j-1}}\right\}$

$$f_{\mathrm{RM}}(x \in I_{\mathrm{RM}}) = \sum_{i=1}^{n}(C_i^0 + a_i \Delta C) e^{\lambda(t_{i-1}-T_{\text{final}})} - P_1 \sum_{i=1}^{n} \max^2\{(C_i^0 + a_i \Delta C) - C_{\max}, 0\} - P_2 \max^2\left\{\sum_{i=1}^{n}(C_i^0 + a_i \Delta C) - C_{\text{cum}}, 0\right\} - P_3 \sum_{i=1}^{n} e^{-\beta \cdot t_i} \max^2\{y_{\min} - y(t_i), 0\}$$

where $\mathbf{c}^0 = (C_i^0)$ is the treatment regimen which undergoes modification using RM.

A2. Multi-drug chemotherapy

(See Sections 6.1.1 and 6.1.2)

Solution space: $S = \left\{\mathbf{c} : \mathbf{c} = \left(C_{ij}\right), i = \overline{1,n}, j = \overline{1,d}\right\}$

Representation spaces:

1. $I_{\mathrm{GA}} = \left\{a_1 a_2 a_3 \ldots a_{4nd} : a_k \in \{0,1\}\ \forall k = \overline{1,4nd}\right\}$
2. $I_{\mathrm{RM}} = \left\{a_1 a_2 a_3 \ldots a_{nd} : a_k \in \{0,1,-1\}\ \forall k = \overline{1,nd}\right\}.$

Fitness functions:

1. *Tumour size optimisation*: $\underset{\mathbf{c}=(C_{ij})}{\text{maximise}}\ J_1(\mathbf{c}) = \sum_{j=1}^{d} \kappa_j \sum_{i=1}^{n} C_{ij} e^{\lambda(t_{i-1} - T_{\text{final}})}$

$$f_{\mathrm{GA}}(x \in I_{\mathrm{GA}}) = \sum_{j=1}^{d} \kappa_j \sum_{i=1}^{n} \sum_{k=1}^{4} 2^{4-k} a_{4d(i-1)+4(j-1)+k} e^{\lambda(t_{i-1} - T_{\text{final}})}$$
$$- P_2 \sum_{j=1}^{d} \max^2\left\{\sum_{i=1}^{n} \sum_{k=1}^{4} 2^{4-k} a_{4d(i-1)+4(j-1)+k} - C_{\text{cum}\,j}, 0\right\}$$
$$- P_3 \sum_{i=1}^{n} e^{-\beta \cdot t_i} \max^2\{y_{\min} - y(t_i), 0\}$$
$$- P_4 \sum_{i=1}^{n} \sum_{l=1}^{\Omega} \max^2\left\{\sum_{j=1}^{d} \eta_{kj} \sum_{k=1}^{4} 2^{4-k} a_{4d(i-1)+4(j-1)+k} - C_{\text{s-eff}\,k}, 0\right\}$$

$$f_{\mathrm{RM}}(x \in I_{\mathrm{RM}}) = \sum_{j=1}^{d} \kappa_j \sum_{i=1}^{n} (C_{ij}^0 + a_{d(i-1)+j} \Delta C_j) e^{\lambda(t_{i-1} - T_{\text{final}})}$$
$$- P_1 \sum_{i=1}^{n} \max^2\{(C_{ij}^0 + a_{d(i-1)+j} \Delta C_j) - C_{\max\,j}, 0\}$$
$$- P_2 \sum_{j=1}^{d} \max^2\left\{\sum_{i=1}^{n} (C_{ij}^0 + a_{d(i-1)+j} \Delta C_j) - C_{\text{cum}\,j}, 0\right\}$$
$$- P_3 \sum_{i=1}^{n} e^{-\beta \cdot t_i} \max^2\{y_{\min} - y(t_i), 0\}$$
$$- P_4 \sum_{i=1}^{n} \sum_{l=1}^{\Omega} \max^2\left\{\sum_{j=1}^{d} \eta_{kj} (C_{ij}^0 + a_{d(i-1)+j} \Delta C_j) - C_{\text{s-eff}\,k}, 0\right\}$$

2. *Tumour burden optimisation*: $\underset{\mathbf{c}=(C_{ij})}{\text{maximise}}\ J_2(\mathbf{c}) = \sum_{p=1}^{n}\sum_{j=1}^{d}\kappa_j \sum_{i=1}^{p} C_{ij} e^{\lambda(t_{i-1}-t_p)}$

$$
\begin{aligned}
f_{\mathrm{GA}}(x \in I_{\mathrm{GA}}) = & \sum_{p=1}^{n}\sum_{j=1}^{d}\kappa_j \sum_{i=1}^{p} 2^{4-k} a_{4d(i-1)+4(j-1)+k} e^{\lambda(t_{i-1}-t_p)} \\
& - P_2 \sum_{j=1}^{d} \max^2\left\{ \sum_{i=1}^{n}\sum_{k=1}^{4} 2^{4-k} a_{4d(i-1)+4(j-1)+k} - C_{\mathrm{cum}\,j}, 0 \right\} \\
& - P_3 \sum_{i=1}^{n} e^{-\beta \cdot t_i} \max^2\{ y_{\min} - y(t_i), 0 \} \\
& - P_4 \sum_{i=1}^{n}\sum_{l=1}^{\Omega} \max^2\left\{ \sum_{j=1}^{d} \eta_{kj} \sum_{k=1}^{4} 2^{4-k} a_{4d(i-1)+4(j-1)+k} - C_{\text{s-eff}\,k}, 0 \right\}
\end{aligned}
$$

$$
\begin{aligned}
f_{\mathrm{RM}}(x \in I_{\mathrm{RM}}) = & \sum_{p=1}^{n}\sum_{j=1}^{d}\kappa_j \sum_{i=1}^{p} (C_{ij}^0 + a_{d(i-1)+j}\Delta C_j) e^{\lambda(t_{i-1}-t_p)} \\
& - P_1 \sum_{i=1}^{n} \max^2\{ (C_{ij}^0 + a_{d(i-1)+j}\Delta C_j) - C_{\max j}, 0 \} \\
& - P_2 \sum_{j=1}^{d} \max^2\left\{ \sum_{i=1}^{n} (C_{ij}^0 + a_{d(i-1)+j}\Delta C_j) - C_{\mathrm{cum}\,j}, 0 \right\} \\
& - P_3 \sum_{i=1}^{n} e^{-\beta \cdot t_i} \max^2\{ y_{\min} - y(t_i), 0 \} \\
& - P_4 \sum_{i=1}^{n}\sum_{l=1}^{\Omega} \max^2\left\{ \sum_{j=1}^{d} \eta_{kj} (C_{ij}^0 + a_{d(i-1)+j}\Delta C_j) - C_{\text{s-eff}\,k}, 0 \right\}
\end{aligned}
$$

3. *Survival time optimisation:* $\underset{\mathbf{c}=(C_{ij})}{\text{maximise}}\ J_3(\mathbf{c}) = \ln\left(\sum_{j=1}^{d}\kappa_j \sum_{i=1}^{n} C_{ij} e^{\lambda(t_{i-1}-T_{\mathrm{final}})} \right)$

$$
\begin{aligned}
f_{\mathrm{GA}}(x \in I_{\mathrm{GA}}) = & \ln\left(\sum_{j=1}^{d}\kappa_j \sum_{i=1}^{n}\sum_{k=1}^{4} 2^{4-k} a_{4d(i-1)+4(j-1)+k} e^{\lambda(t_{i-1}-T_{\mathrm{final}})} \right) \\
& - P_2 \sum_{j=1}^{d} \max^2\left\{ \sum_{i=1}^{n}\sum_{k=1}^{4} 2^{4-k} a_{4d(i-1)+4(j-1)+k} - C_{\mathrm{cum}\,j}, 0 \right\} \\
& - P_3 \sum_{i=1}^{n} e^{-\beta \cdot t_i} \max^2\{ y_{\min} - y(t_i), 0 \} \\
& - P_4 \sum_{i=1}^{n}\sum_{l=1}^{\Omega} \max^2\left\{ \sum_{j=1}^{d} \eta_{kj} \sum_{k=1}^{4} 2^{4-k} a_{4d(i-1)+4(j-1)+k} - C_{\text{s-eff}\,k}, 0 \right\}
\end{aligned}
$$

$$f_{\mathrm{RM}}(x \in I_{\mathrm{RM}}) = \ln\left(\sum_{j=1}^{d} \kappa_j \sum_{i=1}^{n} (C_{ij}^0 + a_{d(i-1)+j}\Delta C_j) e^{\lambda(t_{i-1} - T_{\mathrm{final}})} \right)$$
$$- P_1 \sum_{i=1}^{n} \max^2 \left\{ (C_{ij}^0 + a_{d(i-1)+j}\Delta C_j) - C_{\max j}, 0 \right\}$$
$$- P_2 \sum_{j=1}^{d} \max^2 \left\{ \sum_{i=1}^{n} (C_{ij}^0 + a_{d(i-1)+j}\Delta C_j) - C_{\mathrm{cum}\, j}, 0 \right\}$$
$$- P_3 \sum_{i=1}^{n} e^{-\beta q_i} \max^2 \left\{ y_{\min} - y(t_i), 0 \right\}$$
$$- P_4 \sum_{i=1}^{n} \sum_{l=1}^{\Omega} \max^2 \left\{ \sum_{j=1}^{d} \eta_{kj} (C_{ij}^0 + a_{d(i-1)+j}\Delta C_j) - C_{\text{s-eff}\, k}, 0 \right\}$$

APPENDIX B

STATISTICAL EXPERIMENTS

B0. Weibull distribution

(See Section 5.3.1)

Figure B0.1 presents a general form of the Weibull curve and shows how it depends on the distribution's parameters.

B1. Genetic Algorithms optimising single-drug treatments

(See Section 5.3.2)

Table B1.1 presents the results of the screening experiment aimed at pinpointing significant GA factors. The analysis of this experiment is given on page 84 in the form of a Pareto chart.

Table B1.2 presents the results of the central composite experiment aimed at obtaining a response surface for significant GA factors. The resultant response surface is illustrated on page 86.

Histogram B1.3 compares the results of the confirmation experiments with the optimal GA factor sets obtained from the statistical and the meta-GA factor tuning methods. Statistical analysis of these results is given on page 91.

B2. Genetic Algorithms optimising multi-drug treatments

(See Section 6.1.3)

Table B2.1 presents the results of the screening experiment with an extended set of GA factors attributed to the multi-drug case. The Pareto analysis of these results was performed on page 103.

Table B2.2 presents the results of the corresponding central composite experiment. These results were used to obtain the response surface illustrated on page 104.

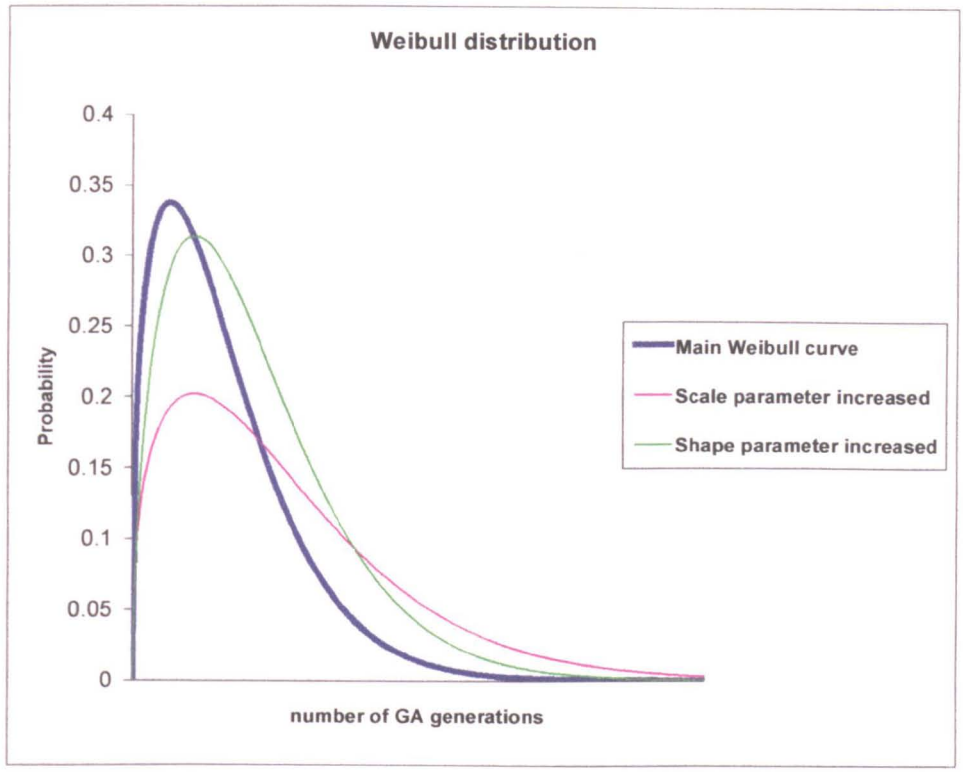


Figure B0.1. Properties of the Weibull distribution

The Weibull distribution is defined by two parameters, viz. the shape α and the scale β parameters. The probability density function of the Weibull distribution for the random variable Ψ (i.e. the number of GA generations required to find a feasible solution) is defined by

$$p(\Psi) = \alpha \cdot \beta \cdot \Psi^{\alpha-1} \exp\left(-\beta \cdot \Psi^{\alpha}\right), \qquad \Psi \geq 0, \quad \alpha \geq 0, \quad \beta \geq 0.$$

The shape parameter α affects the form of the Weibull curve (see the difference between the green and the blue lines in Figure B0.1). The scale parameter β determines the spread of the values (when β increases the range of possible values of Ψ increases too as illustrated by the red line). The Weibull distribution has a mean value

$$\mu_{\Psi} = \beta \cdot \Gamma\left(1 + \frac{1}{\alpha}\right), \qquad \text{where } \Gamma \text{ is the Gamma function.}$$

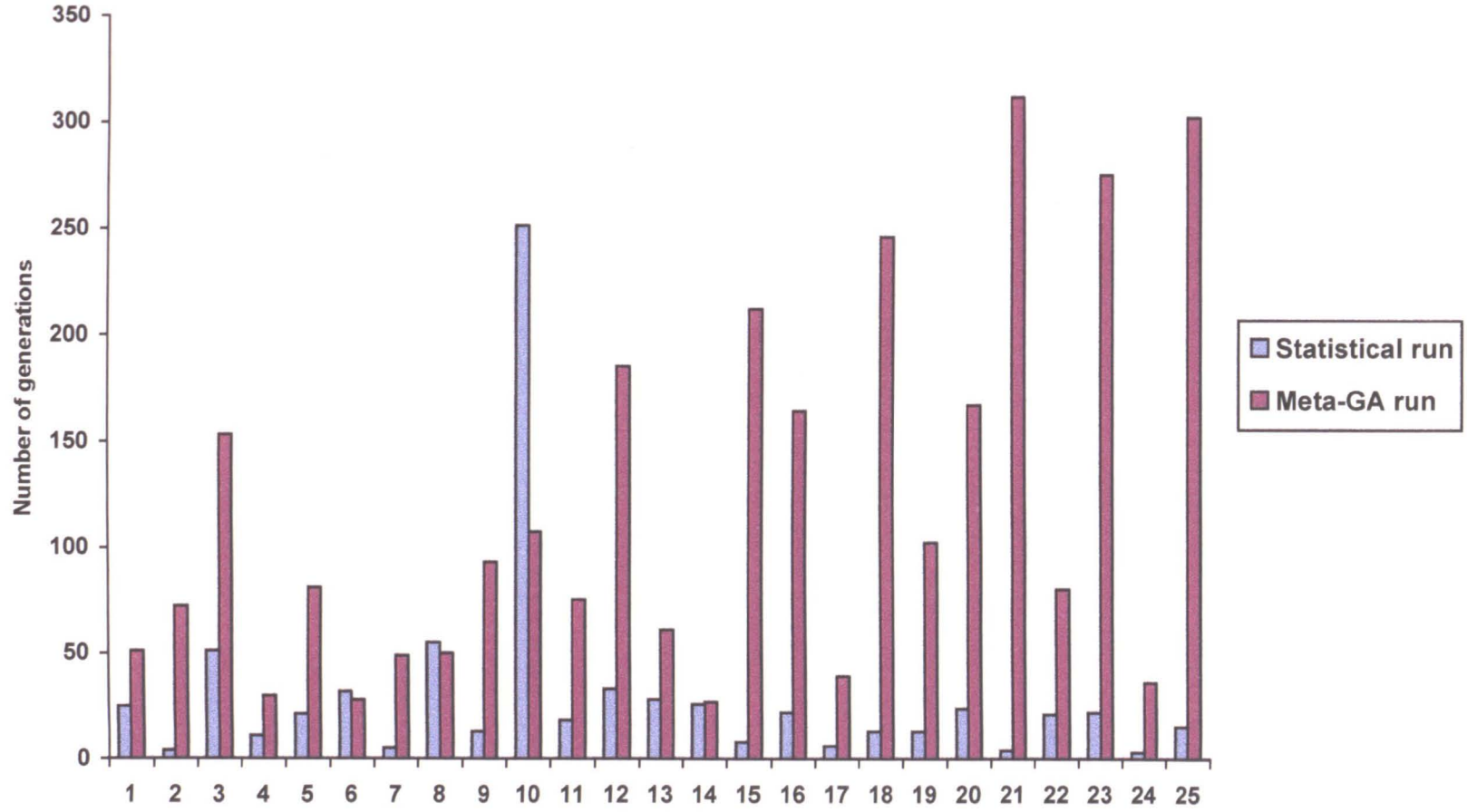
Therefore, by minimising β we minimise μ_{Ψ} and as such we improve the performance of Genetic Algorithms by requiring fewer generations to complete their search.

Levels of design factors							*Scale parameter* β
ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	
0	0	0	0	0	1	1	6199.10
1	0	0	0	0	0	0	3477.44
0	1	0	0	0	0	0	4141.60
1	1	0	0	0	1	1	5574.13
0	0	1	0	0	0	1	1152.94
1	0	1	0	0	1	0	54.13
0	1	1	0	0	1	0	1821.27
1	1	1	0	0	0	1	57.62
0	0	0	1	0	0	0	1219.10
1	0	0	1	0	1	1	70.35
0	1	0	1	0	1	1	3993.44
1	1	0	1	0	0	0	1110.72
0	0	1	1	0	1	0	1140.62
1	0	1	1	0	0	1	830.09
0	1	1	1	0	0	1	740.64
1	1	1	1	0	1	0	1918.33
0	0	0	0	1	1	0	4879.21
1	0	0	0	1	0	1	40.95
0	1	0	0	1	0	1	3698.61
1	1	0	0	1	1	0	3196.18
0	0	1	0	1	0	0	2729.73
1	0	1	0	1	1	1	1175.17
0	1	1	0	1	1	1	2996.95
1	1	1	0	1	0	0	41.58
0	0	0	1	1	0	1	29.40
1	0	0	1	1	1	0	43.74
0	1	0	1	1	1	0	914.14
1	1	0	1	1	0	1	2112.44
0	0	1	1	1	1	1	167.10
1	0	1	1	1	0	0	3426.00
0	1	1	1	1	0	0	1583.48
1	1	1	1	1	1	1	1950.74

Table B1.1. Results of the screening experiment (single-drug case)

Levels of significant factors for the statistic corresponding to the scale parameter β		*Scale parameter β*
ϕ_3	ϕ_4	
-1	-1	642.895
1	-1	277.990
-1	1	33.180
1	1	136.562
$-\alpha$	0	448.808
α	0	211.702
0	$-\alpha$	493.058
0	α	59.641
0	0	118.251
0	0	58.865
0	0	71.049
0	0	54.524
0	0	89.991

Table B1.2. Results of the central composite experiment (α = 1.414) (single-drug case)



Histogram B1.3. Confirmation experiments with optimal GA factor settings

Levels of design factors								*Scale parameter β*
ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	
0	0	0	1	0	0	1	0	394.22
1	0	0	0	1	1	1	0	64.16
0	0	1	1	0	1	1	1	41.58
0	1	0	1	1	1	0	0	56.85
0	0	1	0	0	1	0	0	557.71
0	1	1	0	0	0	1	1	38.10
0	1	0	1	0	1	0	1	40.44
1	1	1	0	0	1	0	1	81.41
1	1	1	0	1	1	0	0	116.10
1	1	0	1	1	0	1	0	64.34
1	1	1	1	0	1	1	0	87.77
0	0	0	0	1	0	0	0	41.55
1	0	1	1	1	0	0	0	54.12
0	0	1	1	1	1	1	0	34.33
1	1	0	1	0	0	1	1	51.00
1	0	0	1	0	1	0	0	121.15
0	0	0	0	0	0	0	1	32.62
1	1	0	0	1	0	0	1	375.80
0	0	1	0	1	1	0	1	338.06
1	0	0	1	1	1	0	1	70.53
1	0	1	0	0	0	1	0	50.73
1	0	1	0	1	0	1	1	139.27
1	1	0	0	0	0	0	0	71.04
0	1	1	1	0	0	0	0	324.33
0	1	1	0	1	0	1	0	338.06
0	1	1	1	1	0	0	1	49.47
0	0	0	1	1	0	1	1	150.16
1	1	1	1	1	1	1	1	40.30
0	1	0	0	0	1	1	0	245.97
1	0	0	0	0	1	1	1	33.13
0	1	0	0	1	1	1	1	105.62
1	0	1	1	0	0	0	1	244.93

Table B2.1. Results of the screening experiment (multi-drug case)

Levels of significant factors for the statistic corresponding to the scale parameter β		*Scale parameter β*
ϕ_4	ϕ_8	
-1	-1	1425.26
1	-1	54.33
-1	1	1278.99
1	1	175.39
$-\alpha$	0	2367.8
α	0	319.31
0	$-\alpha$	49.65
0	α	58.13
0	0	273.45
0	0	286.64
0	0	272.45
0	0	336.97
0	0	350.44

Table B2.2. Results of the central composite experiment (α = 1.414) (multi-drug case)

APPENDIX C

DRUG DETAILS

Table C.1 contains the characteristics of ten drugs most commonly used in various treatments for breast cancer. Risk factors η_{kj} express in a numerical form the likelihood that a given drug will cause a side effect on a particular organ. In the literature (Dearnaley *et al*, 1995) this likelihood is usually denoted by a number of pluses, which can be transformed into numbers as follows:

Likelihood of causing a side effect	Risk factor, η
-	0
+	1
++	2
+++	3

Maximum instantaneous and cumulative doses for each drug are measured in $\mathrm{mg/m^2}$ and obtained from the collaborating oncologists. The algorithm for calculating the potency factors ρ_j is explained in Section 6.1.1, the algorithm for estimating the efficacy κ_j of the drugs is given in (Henderson, 1997).

Drugs	*Side-effects* η_{kj}					Maximum instantaneous dose, $C_{\max j}$	Cumulative dose, $C_{\text{cum } j}$	Potency factor, ρ_j	Effectiveness, κ_j (*10^{-3})
	Bone marrow	Kidney	Periph. nerves	Liver	Heart				
Adriamycin	3	0	0	0	2	75	550	1	5.605
Epirubicin	3	0	0	0	1	75	700	1	4.484
Taxotere / Taxol	3	0	2	0	1	100/130	1000/1500	0.75/0.577	7.29
Cyclophosphamide	2	0	0	0	0	2000	10000	0.0375	3.9235
5 – fluorouracil	0	0	0	0	0	3000	30000	0.025	2.242
Cisplatinum	1	3	3	0	0	120	600	0.625	4.335
Methotrexate	1	1	0	1	0	10000	100000	0.0075	1.6815
Mitomycin – C	2	0	0	1	0	15	40	5	2.242
Prednisolon	0	0	0	0	0	100	1000	0.75	1.121
Vincristine	0	0	2	0	0	2	30	200	2.242

Table C.1. Drug profiles of the utilised anti-cancer agents