TWH working copy

A GA-BASED APPROACH TO ARMS RACES

by

Timothy Hackworth

A Thesis Submitted in Fulfilment of the Requirements for the Degree of

Doctor of Philosophy

in the

University of London

October 2003
Department of Computer Science
Birkbeck College

ABSTRACT

In this thesis we consider the use of Genetic Algorithms (GAs) to predict the outcome of arms races between nations. Clearly a means of predicting *if*, and more importantly, *when* an arms race is likely to escalate into open conflict would be of enormous social, political and economic benefit. It could even lead to some wars being averted altogether. Richardson developed a theory of arms races in the late 1940s but his methods needed a prohibitively large search space and hence no implementation of his theory was feasible until the advent of GAs. Richardson's arms races model is known to be an accurate predictor of escalating conflict but unfortunately does not make it at all obvious *when* unstable conditions will arise. To be of any practical use, therefore, we need a means of predicting the timing of instabilities. In 1991 Peng, Gáspár and Showalter developed a model of canard explosions which successfully predicts the timing of instabilities in certain industrial chemical processes. We demonstrate that ideas borrowed from the Peng, Gáspár and Showalter model can be applied equally successfully to Richardson's model.

In the tradition of the *proof of concept* paradigm, we look in detail at a snapshot (1993 to 1999) of the India versus Pakistan arms race, a volatile situation where minute differences in the average levels of percentage defence expenditure can make all the difference between stability and instability. We then validate our model by applying it to three recent long-running conflicts: the Middle East 1955 to 2000, India versus Pakistan 1955 to 2000, and Greece versus Turkey 1955 to 2000. We show that almost every real and potential instability in these three conflicts could have been predicted using our techniques.

We then apply the same techniques to a model of the stability of nuclear deterrence embodied in the Strategic Arms Limitation Talks (SALT) of 1972 between the United States and the then Soviet Union. It is shown that, once subjected to a Pareto sort, results obtained from our model agree with the figures calculated so laboriously by the Americans at that time. Additionally, our deterrence model can evaluate '*what if* **...**' scenarios previously regarded as infeasible.

We also consider the beneficial effects of introducing redundant genes into the chromosomes of the GAs and discretising their evaluation equations. Experimental evidence suggests that these two techniques lead to more robust GAs.

The thesis concludes by commenting on the viability and credibility of our results, and by making a number of general observations on the application of GAs to political subjects.

ABBREVIATIONS

The following abbreviations are used in the body of the thesis.

col	column
dom	dominant or dominated
EA	evolutionary algorithm
Eq(.), Eqs(.)	equation(.), equations(.)
ES	evolution strategy
ffn	a fitness function, i.e. an expression which calculates fitness
fit	fitness
GA	genetic algorithm
gen	generation, or generations run
normfit	normalised fitness, see ‘Linear Norm’ in the Glossary
ODE(s)	ordinary differential equation(s)
RNG	*random number generator*, normally the one embedded in a PC
random	random number(s)
running	running total of all the fitnesses or normfits
srl	serial number (normally of an individual chromosome)

A GA-BASED APPROACH TO ARMS RACES

Style

The nine chapters are numbered consecutively. Sections (sub-headings), figures, tables, equations and inequalities all bear the chapter number; for example, Section 5.2, Section 5.2.2, Section 5.2.2(4), Figure 5.3, Table 5.4 and Eqs(5.5) are all to be found in Chapter 5. Appendices are numbered correspondingly with the chapter number first, such as Appendix 6A, Appendix 6A.1, Figure 4A.3 and Eq(7B.3).

Texts in double quotation marks are actual quotations. Single quotation marks are used for emphasis and for indicating unconventional use, e.g. “A badly-handled crossover can result in an effective ‘mutation’ of the chromosome”. Italics are used for emphasis; many of the phrases in italics will be found in the Glossary at the end of the thesis, where there is also a Catalogue of Symbols. The `Courier` font is used for source code.

Contents

List of Figures and Tables

Catalogue of Algorithms used in the Thesis

Sixty-eight algorithms were written during this study. Those quoted in the thesis are listed here in the order that they appear (under 'Ref'). *LEGEND*: **ES**, evolutionary strategy: **GA**, genetic algorithm: **MC**, Monte Carlo: **NM**, numerical method (other than GA, MC or SA): **SA,** simulated annealing: **Gene**, number of genes: **Gen**, generations run: **GT**, Greece/Turkey: **IP**, India/Pakistan: **ME**, Middle East: **P**, Population: **Redund,** Redundancy trials. ***SUFFICES:*** **a**, prediction GA: **z**, modelling GA: **ab,** no redundancy: **aa**, 33% redundancy: **axx**, 50% redundancy: **PP**, 1955 to 2000. All GAs have population 100

Srl	Name	Role	Gene	Ref	Gen	Description and Principal Role
GA-0	GA-37ab	IP	10	Glossary	40	No redundancy. Control for redundancy trials
GA-1	GA-37aaa	Redund	15	2.17	40	Redundancy trials (33%)
GA-2	GA-41a	Nuclear	6	2.17	20	Nuclear prediction GA. Del probs variable
GA-3	GA-57aaa	Nuclear	15	2.17	40	Discretised evaluation equations. Robust
GA-4	GA-33aPP	ME	6	3.5	20	ME prediction GA.
GA-5	GA-37aa	IP	15	3.8	40	Improved version of GA-37a
GA-7	GA-37e	IP	15	3.5	40	Fractal basins for GA-37aa
GA-9	GA-37axx	Redund	20	4.4	40	Redundancy trials (50%)
GA-10	GA-33c	ME	6	4.10	20	Fractal basins for GA-33aPP (2)
GA-11	GA-33d	ME	6	4.10	20	Fractal basins for GA-33aPP (1)
GA-12	GA-33a	ME	6	4.11	20	Dominance Diagrams
GA-13	GA-37z	IP	15	5.52	10	IP modelling GA. To process IMF data
GA-6	GA-37a	IP	15	5.6	20	IP prediction GA for Chapter 5 (1993 to 1999)
NM-5	NM-37a	IP		5.13.1		Numerical Method 1
GA-26	GA-37aPP	IP	15	6.6	40	IP prediction GA. 33% redundant
MC-5	MC-37aa	IP		5.13.4		Monte Carlo, with discretised equations. P 100
SA-5	SA-37aay	IP		5.13.6		Simulated Annealing. P 2
GA-14	GA-50zPP	GT	15	6.3	40	GT modelling GA. To process IMF data
GA-15	GA50zPPT	GT	15	6.3.1	40	GT modelling GA with Cao *et al.*'s norm
GA-16	GA-50aPP	GT	15	6.3.1	40	GT prediction GA
GA-17	GA-33zPP	ME	6	6.5.1	10	ME modelling GA. To process IMF data
ES-1	ES-41a	Nuclear	6	7.8	60	Evolutionary Strategy. P 2
GA-19	GA-34a	Nuclear	6	7.9.2	20	Early nuclear trials. Not robust.
GA-18	GA-41b	Nuclear	6	7.11	20	Pareto-sort
GA-20	GA-41ae	Nuclear	6	7.11	20	1% elitism
GA-21	GA-41be	Nuclear	6	7.11	20	Pareto-sort with 1% elitism
GA-22	GA-41ax	Nuclear	9	7.11	20	33% redundancy
GA-23	GA-41bx	Nuclear	9	7.11	20	Pareto-sort, with 33% redundancy
GA-24	GA-41axe	Nuclear	9	7.11	20	33% redundancy and 1% elitism
GA-25	GA-41bxe	Nuclear	9	7.11	20	Pareto-sort with 33% redundancy & 1% elitism
NM-2	GA-41aLF	Nuclear		7.16.1-5		Gauss-Seidel and Jacobi iterations
MC-6	MC-41a	Nuclear	6	7.16.6	40	Monte Carlo, using mutation to perturb. P100
GA-27	GA-37abe	Epistasis	10	8.6	20	GA-37ab with gene loci varied (1)
GA-28	GA-37abe2	Epistasis	10	8.6	20	GA-37ab with gene loci varied (2)
GA-29	GA-38ab	Epistasis	10	8.7	100	de Jong /Naudts test
GA-30	GA-38aaa	Epistasis	15	8.7	100	de Jong /Naudts test
GA-31	GA-38axx	Epistasis	20	8.7	100	de Jong /Naudts test
GA-33	GA-38abt	Epistasis	10	8.9	100	de Jong /Naudts test (one intron only)
GA-32	GA-37abt	Redund	11	8.9	40	One intron only, at the fixed point of crossover
GA-34	GA-57aat	Nuclear	15	8.9	40	Discretised, one intron, fixed crossover.

Preface

In 1972 I was posted as a post-graduate student to the National Defence College, Latimer. It was the time of the Strategic Arms Limitation Talks (SALT) between the United States and the Soviet Union. Because of my background (I had previously worked in the Design Mathematics Section of the Atomic Weapons Research Establishment at Aldermaston) I was tasked with verifying the figures which the Americans had calculated to meet the SALT criteria. What holding of nuclear weapons would publicly assure deterrence (while quietly ensuring that the Americans would always hold the upper hand)? The Americans were known to have based their SALT calculations on the 1947 zero-sum game theory of von Neumann and Morgenstern, and I was encouraged to use some other method. I decided on the arms race theories of LF Richardson and, after some considerable difficulty, did agree the American figures, but there was little latitude or flexibility in the results. The calculations were so tedious and the computing facilities so primitive that I remember thinking that there must be a better way. From then on, my career went in a different direction, and Richardson's theories were put on the back burner. Interest revived in 1981 when I was appointed Defence Attaché at the British Embassy in Amman. Here, one lived among Jordanians (both Bedou[1] and Palestinians) who felt themselves surrounded by predators, and whose principal, everyday concern was survival. To them, the subject of this thesis is very real.

I have spent much of my life as an engineer. Many of my technical postings in the Army had involved the more mathematical aspects of computing, so when I left the Army in 1988 I was glad to join the paid staff of the British Computer Society (BCS) as its Technical Director. After retiring from full-time employment in 1992 I had time, inclination and the means to do many of the things I had always wanted to do. Among them was to revisit Richardson's theories, which have always intrigued me. By the mid-1990s, however, the situation was very different. The combination of 'number-crunching' PCs and workable Evolutionary Algorithms made it possible to handle problems requiring huge search spaces. The effect of applying this combination to Richardson's theories was dramatic, and answers to a wide range of nuclear scenarios became feasible. Furthermore, it became evident that the arms race theories were applicable to a whole raft of current situations and, once coupled with the canard explosion theory of Peng, Gáspár and Showalter, can even predict *when* instabilities are likely to occur. The ability to predict the timing of an outbreak of war

---

[1] The plural of "Bedouin", the nomadic desert tribesmen, is "Bedou"

is invaluable; not all arms races end in war, so efforts at mediation can be directed where they are most likely to be effective.

Despite remaining technical mentor for BCS mature candidates and an assessor and Representative for the Engineering Council, and a Liveryman of the Worshipful Company of Information Technologists, retirement found me living in an intellectual vacuum. It seemed natural, therefore, to attempt the discipline of a doctoral degree to re-dress the balance, partly for my own satisfaction ("*Engineers have an obdurate tenacity of vision*" [2]) and partly because I really do hope that some good will come from the results of my efforts. This is the motivation for my research.

I admit to sharing the outlook and philosophy of George Berkeley (1685-1753) [Berman, 2000; Luce, 1967]. Berkeley, an Irish mathematical philosopher and cleric, was described by Locke as a master of conceptual analysis and argument. Berkeley believed that experience forms the basis of all human knowledge and is the only thing that can be trusted. He regarded mathematics as a science of empty abstractions (despite being rather good at it), and he thought that language was so flawed that it could never be trusted, even when it appeared to be revealing a contradiction (thereby pre-empting Wittgenstein by some two hundred years). Berkeley's scientific experimental method – *subjective empiricism* – says, in effect, that when we set about solving a problem we should go for what we perceive in our imagination and experience, and look for a theoretical justification afterwards. Whoever said "do not attempt to create and analyse at the same time, for they are totally separate activities" was echoing Berkeley.

My sincere thanks go to my supervisors, Dr Xiaohui Liu (now a professor at Brunel) and, latterly (over the past 3½ years), to Professor George Loizou, Head of the School of Computer Science and Information Systems at Birkbeck and holder of the University Chair of the Mathematics of Computation, for their help and encouragement, for their painstaking attention to detail, for their constructive and (in the main) good-humoured criticism, and often for pointing me in the right direction.

In particular it was Professor Loizou who noticed the mathematical similarity between Peng *et al.*'s canard explosion model and Richardson's equations and suggested that the one might be applied to the other.

---

[2] David Jordan, Managing Director of Philips (UK) plc, Annual Mountbatten Lecture, IEE London, 27 November 2001

I would also like to thank a number of anonymous referees of various learned journals for their lengthy and perceptive comments and (in the main) for being tolerant of my sometimes iconoclastic views. My thanks are also due to Stan Kay, Nigel Morgan, Andy Smith, Ian Williams and Meirion Thomas of the British Computer Society for their unfailing patience whenever I had serious PC problems, to my fellow Engineering Council assessor Sid Dunn (who actually knew and had worked with LF Richardson at the Meteorological Office in Bracknell just after World War II), to my fellow Birkbeck students Stephen Swift and Allan Tucker, firstly for helping me out so readily when I was poorly in 1999 and, secondly, for making a number of practical suggestions about presenting my case to best effect. I must also thank Jason Crampton for taking such a rationalist approach (contrasting markedly with my empiricism), and Ito Wasito for showing me how to use MATLAB so painlessly. Lastly, I must again mention Dr Stephen Swift, now a Research Fellow at Brunel, to whom I am particularly grateful for reading through the complete draft, and Dr Steve Counsell, Lecturer at Birkbeck, for his unfailing help and encouragement during some dark days.

My interest in the mathematical treatment of topics such as arms races dates back to the mid-1950s when Professor Cyril Lambe[3] of the Royal Military College of Science, Shrivenham, inspired in me an enthusiasm for applied mathematics, one which, sadly, I did not have the mind-set to exploit. Since then, many people have influenced my approach and encouraged my efforts, including Dr Guy Scorgie of AWRE Aldermaston in the 1970s and, more recently, the Reverend Canon Dr Barry Thompson, now a Canon Emeritus of Windsor, the late Peter Begent the war historian and philosopher, Dr Roger Johnson, Reader in Computer Science at Birkbeck, and Dr James Levenick of the Computer Science Department at Williamette University, Salem, Oregon.

Finally I am enormously grateful to my long-suffering wife Jan who, despite seeing very little of me for the past six years, has quietly provided the permanence, the infrastructure, and the facility for concentration without which this thesis could never have been attempted.

TWH

---

[3] CG Lambe, author of a number of mathematical texts for engineers

CHAPTER 1

Introduction

1.1 An Overview

The thesis is primarily concerned with the application of *Genetic Algorithms* (GAs) to arms races between nations. In particular, we want to know whether there exists some means of using GAs to predict when (or under what conditions) conflicts between nations are likely to erupt into open war. Until a solution to this problem can be determined, there is no possibility of averting wars. The arguments are based on Richardson's *Theory of Arms Races* [Richardson, 1960b], originally developed in the late 1940s, but needing such a large search space that it was virtually unusable[4] until the advent of *heuristic search* which, in turn, allowed the development of *Evolutionary Algorithms* (EAs).

Richardson's equations bear a marked similarity to those developed by Bo Peng, Vilmar Gáspár and Kenneth Showalter [Peng *et al.*, 1991] to predict *canard explosions* in the manufacture of plastics, and some ideas about impending instability have been borrowed from their chaos-based analysis of limit cycles.

Throughout the thesis *it is assumed* that evolutionary algorithms have advantages over all other methods when large, 'hilly' search spaces have to be handled, and *it is assumed* that their subset, GAs, are best suited to handle Richardson's equations. Nowadays, there are certainly other methods of solving Richardson's equations, and some of them (e.g. non-evolutionary population-based hill-climbing methods) could well be more effective in some circumstances; however, our subject matter is how easily GAs adapt to handling arms races.

Very early in our studies it became evident that application non-specific GAs are neither efficient nor particularly useful for any practical purpose. We therefore built on existing ideas and developed *ab initio* our problem-specific GAs, ones which took into consideration the arms races domain, yet were sufficiently conventional to be properly described as *canonical*. The resulting GAs do not marry up precisely with any of the nineteen variants[5] of Holland-type GAs mentioned in Section 2.5.

---

[4] In 1983 Kirkpatrick *et al.* developed a *simulated annealing* method which we now use in Section 5.8.2 to solve a Richardson arms race problem. As far as is known, simulated annealing has never been used in this role before

[5] We use 'variant' rather than 'type'. We identify four generic *types* of GA **;** Holland (canonical), Falkenaeur (grouping), Whitley (steady-state) and Nishio *et al.* (interactive). (See Section 2.5.) Within each of these types there are many *variants*

Rather than compare the performance of GAs in the arms race role with other (non-evolutionary) methods, comparisons will – with two exceptions – be made between:

- The effects of changing various parameters in our GAs, including the *fitness function*
- The type of *representation* used (i.e. how the *genotype* encodes the *phenotype*)
- How the *evaluation equations* [Davis, 1991] are presented mathematically
- Redundant and non-redundant chromosomes

The two exceptions lie in Chapters 7 and 5, where we briefly invoke the Gauss-Seidel and Jacobi numerical methods [Buckingham, 1962], Monte Carlo techniques [Kalos and Whitlock, 1986] and simulated annealing techniques [Pham and Karaboga, 2001] to see how their results compare with those of our GAs.

GAs are now used to solve engineering problems and even economic ones. Biethahn and Volker-Nissen [1995] were among the first researchers to apply GAs to commercial management. Not so common, however, is their use for political or social problems. In this context we mention Ackley [1987], who developed a GA where the voters (population) express their satisfaction or dissatisfaction toward a ϑ–member government (ϑ is the length of the chromosome), summing positive and negative feedback from individual voters, Nishio *et al.* [1997] , who designed an interactive GA for the Tokyo Police aimed at helping witnesses identify the faces of suspect criminals, and Deboeck [1994] who employed a GA to forecast fluctuations in share prices on the financial markets.

More recently Bhattacharyya *et al*. [2002] have been using EAs to develop trading models for foreign exchange markets, while Aldawoodi and Perez [2003] have developed a GA-based technique to generate equations to monitor the migration of both people and seabirds. Our searches of EA literature for papers in similar fields to ours have revealed comparatively few, the most significant being Forrest and Mayer-Kress [1991] who used GAs to create early models of international security, Andreou and Zombanakis [2001] who used neural networks to calculate *relative security coefficients* to measure the impact of arms races between Greece and Turkey, Axelrod [1997] who studied the complexities of co-operation as part of the whole subject of conflict resolution, and Moore [2002] who considers the optimisation of missile counter-measures.

1.2 Arms Races: a Brief History

The ability to analyse stability and equilibrium is vital to all models of the real world and is based on principles established in the nineteenth century by Le Chatelier and Gibbs [Porter, 1994]. More recently, however, the credit for pioneering the application of mathematics to international politics belongs to the meteorologist *Lewis Fry Richardson DSc FRS* (1881 - 1953). His principal works are *The Statistics of Deadly Quarrels* [Richardson, 1960a] and *Arms and Insecurity* [Richardson, 1960b] both of which were published posthumously. The latter contains his *Theory of Arms Races* which underpins the thesis.

Richardson analysed every armed conflict from the Napoleonic Wars of the early 1800s to the First World War and concluded that all of them had resulted from an out-of-control arms race. Such arms races do not necessarily result in wars but, in his own words "...war is more likely than not".

Richardson's aim was to understand the long-term implications of national political and economic policies on global stability, for he was convinced that the understanding which grew from the systematic analysis of the events which were known to lead to war would contribute more to advancing the cause of peace than the intuitive and emotive reasoning of statesmen, politicians, soldiers and diplomats. His model shows how expenditure on arms changes over time in response to previous expenditures, perceived external threats, and certain economic factors. The model can be used as a tool to study how policies evolve, making it possible in theory to investigate the ramifications of various strategic policy decisions.

Richardson's model is essentially a non-linear dynamic system, the behaviour of which can be difficult to predict (as we show in Chapter 3) because of non-linear couplings between dependent variables. Additionally, in any realistic model, the number and range of possible parameter settings can be vast: the three-nation equations alone have fifteen real-valued parameters, giving rise to an unworkably large search space. In consequence, researchers have until recently avoided Richardson's model and looked for other methods.

It is not clear who originally coined the phrase 'arms race' but it is not a good analogy with, say, a running event on an athletics track. Is an arms race a sprint or an endurance race? Most running races tend to be (more or less) steady speed events, while arms races appear to accelerate without limit. There must, of course, be a limit in the real world; perhaps it is bankruptcy but, historically, no nation is known to have bankrupted itself in this way although it could be argued that Pakistan has recently come close to doing so. Additionally, as Dawkins [1983] has said, there is an essential asymmetry to evaluating arms races: a prey is running for its life, while a predator is running for its lunch!

1.3 Limitations of the Richardson Approach

The Richardson approach will only work where the chance of conflict is high and between known protagonists. Wars of independence, civil wars, guerrilla wars, *local* terrorist activity[6] and insurrections do not come into this category. Moreover, it is only worth trying the Richardson approach where there are reasons for suspecting major antagonisms. Chapter 5 considers India versus Pakistan; we could equally have tried India versus Goa, India versus China, or even India versus Bhutan, but it did not seem worth it because part of the skill in applying mathematics to any 'political' subject is to recognise where and when such effort will be cost-effective.

1.4 New Contributions and Benefits

The major contribution of this thesis has been the ability to forecast when wars are likely to break out. Based on statistical information (and the more up-to-date, the better), the ability of changes in the direction of curvature of the *limit cycle* to predict impending instabilities in the Richardson equations (Chapter 3), and the *implicit parallelism* inherent in GAs (Chapter 2), we use our model to forecast future conflicts and demonstrate the *proof of this concept* historically in Chapters 5 and 6. We believe that this is a major achievement inasmuch that *knowing* the date of an impending conflict (to within, say, a month) will make it easier for other nations to defuse the situation by diplomatic action (i.e. by threatening or cajoling the potential antagonists). Other features are also innovative, in particular:

- Merging the Richardson and Peng models (which were created at different times, for different purposes, and under different circumstances) into one coherent whole.
- The use of *integer-coded* GAs where the genotype uses, literally, the same numbers as the phenotype. Such direct mapping is comparatively unusual.
- GAs are mostly used for optimisation. Our use is unconventional in the sense that a GA is used as a modelling vehicle for Richardson's equations and almost the same GA is then employed as a predictor of impending instabilities. Creating a credible model and transferring rate-factors from the modelling GA to the prediction GA poses a problem because the transferred rate-factors must be, and be seen to be, *appropriate*.
- Making use of the contention of Vose [1999] that the mathematics of GAs are not invalidated by the representation used. This is contrary to the views of Holland [1975] and Goldberg [1989] who, even today, stick to their *schema* theorems.

---

[6] *International* terrorist activity, in contrast, can sometimes be considered in that it is often sponsored and financed by some third-party government

- The use of real-world data to *seed* some of the genes.
- Deliberately introducing *totally useless* redundant genes into a chromosome. At first sight this does nothing but, judiciously placed, *one* redundant gene can significantly reduce the disruption to the building-blocks caused by crossover. This makes the GA under consideration more *robust* (Chapter 8).

1.5 Structure of the Thesis

The thesis is set out as follows:

- Chapter 2 presents background material on GAs which is either referenced in the thesis or is necessary for an understanding of the results.
- Chapter 3 starts by presenting background material on Richardson's arms race model and its use. It then briefly discusses Peng *et al.*'s [1991] model of canard explosion theory and the techniques for predicting instability which they developed. It concludes by putting the two together and arguing the mathematical case for so doing.
- Chapter 4 discusses the GAs and the model used in the thesis. It also gives some experimental justification for the assumptions made. An Appendix to Chapter 4 shows a typical set of GA diagnostic tables, and some selected computer outputs.
- Chapter 5 looks at a snapshot of the India versus Pakistan situation (1993 to 1999) in some detail and shows how minute changes in average *defexp%* (see Glossary) can make all the difference between stability and instability. It also uses the 1993 to 1999 data as a *training-set* to predict an event in 2000. The chapter also looks at two alternative 'non-GA' methods (Monte Carlo and simulated annealing) and seeks to establish why GAs are so effective as predictors.
- Chapter 6 validates the model by showing that it works over long periods of time (1955 to 2000) in three dissimilar historical instances, Greece and Turkey, India and Pakistan, and the Middle East. It then considers the effects of *noise* on the predictions. The chapter concludes by attempting tentatively to predict some conflicts in the future.
- Chapter 7 applies Richardson's theory to nuclear deterrence, verifies the SALT calculations of 1972, and demonstrates how the flexibility of GAs allows situations of the '*what if ...*' variety to be resolved. GAs pertaining to nuclear deterrence are not robust, but it is shown that this can be ameliorated in two different ways:

  - by subjecting the results to a Pareto-sort
  - by discretising the evaluation equations.

- Chapter 8 explains the effects of redundancy in a GA’s chromosomes and considers how it can be beneficial in the case of our GAs.

- In Chapter 9 we present our overall conclusions and suggest ideas for further research.

CHAPTER 2

Background 1: Genetic Algorithms

2.1 Introduction: Soft Computing

Zadeh [2002] defines *soft computing* as follows: “Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation. Soft computing techniques can solve optimisation problems”. Alternatively, to quote Sanchez *et al.* [1997], “Soft computing encompasses all those modes of computing in which imprecision is traded for tractability, robustness, ease of implementation and lower solution cost. Soft computing, therefore, is primarily concerned with optimisation. Its components include all types of *evolutionary algorithms*, fuzzy logic, neural networks, probabilistic reasoning, chaos theory, and even machine learning”.

This chapter presents background material on evolutionary algorithms which is either referenced in the thesis or is necessary for an understanding of the techniques used and of the results obtained. It summarises the traditional Holland/Goldberg approach to the mathematics of GAs, but does not attempt to present a full theoretical background of either GA structure or GA operation.

2.2 Evolutionary Algorithms

Evolution is about search, a search for something better, an optimum. A search space is one filled with all possible solutions, and a point in that space defines a solution. EAs evolve solutions to problems, not just with one but with a whole population of candidate solutions at once. Bentley [1999] states that “evolution is not specified in an EA but it is an *emergent property* of the algorithm”.

For small spaces, classical search methods suffice, but for large spaces such methods, while still valid, are generally not easy – and sometimes impossible – to implement. Furthermore, classical computer systems may well excel at dealing with complicated data, but they are not designed for processing inaccurate, noisy or complex data [Yao, 1999]. EAs are stochastic processes whose search methods are based on a natural phenomenon; genetic inheritance, and the quest for survival as expounded by Darwin. The metaphor underpinning GAs [Davis and Steenstrup, 1987] is that of natural evolution, where each species faces a search for beneficial adaptations to a complex and changing environment. The ‘knowledge’ that each species gains is embodied in the construction of the chromosomes of its members.

The four principal classes of EAs are generally taken to be Genetic Algorithms (GAs), Evolution Strategies (ES), Evolutionary Programming (EP) and Genetic Programming. (GP)[1].

EAs manipulate populations (ranging from hundreds to thousands) of chromosomes which represent individuals, or ‘candidate solutions’. Some individuals are better (‘fitter’) than others and these are the ones which must be allowed, even encouraged, to reproduce. EAs are sometimes classified by data structure and by the number and nature of their *genetic operators* which ‘do’ things to those chromosomes. All EAs have one thing in common; the physical representation of the coding (the *genotype*) must in some way mirror or otherwise map onto the characteristics of the problem to be solved (the *phenotype*). More is said later about this mapping, which is known by its biological name *embryogeny.*

Evolutionary activity is the “spontaneous generation of innovative functional structures” [Zadeh, 2002]. As Baldwin said [see Cohen and Stewart, 1994], if learning helps survival then the organisms best able to learn will have the most surviving offspring. Some observers [Bedau and Packard, 1992], echoing Holland [1992], were more philosophical: “Adaptation is a tension between *exploration* (the search for new, useful adaptations) and *exploitation* (the use and propagation of the adaptations)”.

2.3 GAs: The Traditional View

John Holland invented and developed GAs at the University of Michigan in the 1960s and 1970s. GAs work by combining selection, recombination and mutation operators [Haataja, 1999]. The selection pressure drives the population towards *fitter* solutions. Individuals are selected for ‘parenthood’ – or, with elitism, for survival into the next generation – on the basis of their fitness.

All GAs, however, need to escape from local optima, and this is where *mutation* is effective, because it widens the search space and strays into new territory [Haupt and Haupt, 1999].

Coded solutions (genotypes) must be mapped onto actual solutions (phenotypes) before fitness can be determined. Phenotypes are collections of live parameters of real-world problems expressed in whatever manner is conventional and appropriate. A coded parameter is called a *gene* [Davis, 1991; Bentley, 1999]. There is a random element to the exploration of the search space but GAs are not parallel random search algorithms because a GA’s search

---

[1] Over the past few years these distinctions have become blurred

is unquestionably directed by selection towards areas which contain better solutions. Since there are many solutions in parallel (see *implicit parallelism* in Section 2.4), GAs are rarely misled by local optima [Bentley,1999].[2]

2.4 Schemas (The Holland Approach)

It is a central tenet of GAs that they work by discovering, emphasising, and recombining – via selection, mutation and crossover collectively – good ‘*schemas*’ or *building blocks* of solutions in a highly parallel fashion [Mitchell, 1996]. Schemas[3] do not actually exist, but they are a useful notional tool. In essence, good building blocks are combinations of bit values which confer higher fitness on the strings in which they are present, and hence lead in turn to better solutions.

A traditional schema is a fixed template describing a subset of strings with similarities at certain defined positions [Coley, 1999]. Strings which contain the same schemas contain (to some degree) similar information. (Using a ternary alphabet (0, 1, and the metasymbol #), 101001 and 111001 are both *instances* of the schema 1##001.) For any string of length L over {0, 1} there are 2^L possible schemas. In general, for an alphabet of cardinality (distinct characters) k, there are k^L possible schemas. For a population of N strings ($N > 1$) there are potentially Nk^L possible schemas, but the actual number is likely to be much less than this, for there may be duplicates. In either case, the number of schemas will always be less than or equal to Nk^L.

Fatal accidents can happen to strings of a GA. They can be broken up by crossover, attacked by mutation, or wilfully discarded by selection. Despite this, it is relatively easy to estimate how the number of instances of a particular schema might change during a GA run. This estimation throws light on why GAs can be so successful as optimisation vehicles.

If $\xi(S, g)$ is the number of strings matched by schema S within the population at generation g such that $\xi(S, g) > 0$, then, in general, it would seem reasonable to assume that if we consider strings of above average fitness, then $\xi(S, g+1) > \xi(S, g)$. If selection is founded on a fitness-proportional basis (e.g. by roulette wheel selection as in our GAs), the probability $p_i(g)$ of any individual, say i, at generation g being selected is

---

[2] Deceptive trap functions, which are specifically designed to mislead GAs, are outside the scope of the thesis

[3] Pedants talk about ‘schemata’; we prefer ‘schemas’

$$p_i(g) \approx \frac{f_i(g)}{\sum_{j=1}^{N} f_j(g)} \quad ,$$

where $f_i(g)$ stands for the fitness of the individual *i* at generation *g*.

If *eval* (*S, g*) is the average fitness of all strings in the population matched by schema *S*, then

$$\xi(S, g+1) \approx \frac{eval(S, g)}{F(g)} \xi(S, g) \ , \tag{2.1}$$

where *F*(*g)* is the average fitness of the entire population at generation *g*. Eq(2.1) means that the number of strings in the population that match schema *S* grows as the ratio of the average fitness of every string in the population matched by schema S to the average fitness of the entire population. Hence an 'above average' schema receives an increasing number of strings in the next generation, and vice versa.

Eq(2.1) is the *schema growth equation* (ignoring mutation and crossover) and shows that the number of instances of any schema *S* in the next generation, *g* + 1, depends on *eval*(*S, g*) in the current generation.

Assuming that a particular schema remains constantly above average fitness by $\beta F(g)$, then Eq(2.1) becomes

$$\xi(S, g+1) \approx (1+\beta)\xi(S, g) \quad \text{or} \quad \xi(S, g) \approx (1+\beta)^g \xi(S, 0) \quad \text{[Coley, 1999]} \, . \qquad .$$

This binomial progression suggests that better performing schemas will receive exponentially increasing numbers of trials in the next generation. Barring elitism, there are only *N* acts of selection per generation (i.e. one per individual member of the population), but the GA manages simultaneously to allocate exponentially increasing number of trials to a vast number (>> *N*) of schemas effortlessly. This is called *implicit parallelism,* and the reason for it is that any population of *N* binary strings contains instances of between 2^k and $N2^k$ different schemas. This means that, at any given generation, while the GA explicitly evaluates the fitness of the *N* strings of the population, it is in fact estimating the *average* fitness of a much larger number of schemas − Bertoni and Dorigo [1993] have estimated that at least N^3 schemas will be processed usefully. The GA behaves as if the increase and decrease in the numbers of given schemas in any population was actually reflected in the results − indeed, just as if it were calculating and storing these averages. We make extensive use of implicit parallelism later in the thesis; indeed, this is the principal reason why GAs outperform most other methods in our arms race application.

The disruption caused to schemas by *crossover* will depend on the schemas involved. Empiricism urges that at least half the schemas will remain intact. Rationalism insists that the probability of a schema S being destroyed by crossover is less than

$$\frac{d(S)}{L-1}, \qquad .$$

where $d(S)$ specifies the *defining length*, the distance between the first and last non-meta-symbol within the string, e.g. if S = #1#0# then $d(S)$ is 2 while if $S = 11001$ then $d(S) = 5$. Given a crossover probability p_c, the chance of the survival of a bit into the next generation is bounded below by $1 - p_c \frac{d(S)}{L-1}$.

Given a mutation probability p_m, the probability of a single bit surviving a single mutation is $1 - p_m$. The probability of the whole schema surviving intact is $(1-p_m)^{o(S)}$, where $o(S)$ denotes the number of (non-metasymbol) bits in schema S.

Holland [1975] showed that, by expanding but ignoring lesser terms, the combined effect of selection, crossover and mutation can be expressed in the single growth equation

$$\xi(S, g+1) \geq \xi(S, g) \frac{eval(S, g)}{F(g)} \left(1 - p_c \frac{d(S)}{L-1} - p_m o(S)\right). \tag{2.2}$$

The point to note is that above-average, short-length, low-order schemas (called *building blocks*) will be sampled at an exponentially increasing rate [Davidor and Schwefel, 1992].

The building block hypothesis [Goldberg, 1989] contends that GAs evolve good solutions by combining these fit, low-order schemas to form better schemas. It is for this reason that recombination (crossover) is generally taken to be the GAs' strength**;** namely, the ability to recombine good schemas to form instances of equally-good or even better schemas.

One important aspect of the building block hypothesis is that these short, high-average fitness schemas tend to be combined to best effect by single-point crossover. Multiple crossovers, however, have their uses; the *Breeder Algorithm* [Mühlenbein, 1994] emulates the activities of farmers and stock breeders and allows only a small number of (the fittest) animals to breed. In an effort to get the best of both worlds, another form of crossover, *uniform cross-over* [Syswerda, 1989], was developed, engineering the recombination of genes with the aim of making the best arrangement of schemas. Holland suggested that alternative repres-entations in a GA could be compared by calculating the number of schemas processed by the algorithm [Holland, 1975]. It has now become apparent [Fogel, 1999] that in order to max-

imise intrinsic parallelism Holland wanted representations chosen with the fewest "detectors with a range of many attributes". In other words, alphabets of low cardinality were to be favoured because they generated more schemas which Holland thought, in turn, would give a "larger information flow" to a GA's reproductive plans. It was for this reason that he chose to use a binary representation in his algorithms. This decision has dogged GAs ever since, for it became a sort of gospel which still has its devotees today despite its many shortcomings.

The reason for focusing here on schemas and on binary coding is partly that it is traditional, and partly that, until comparatively recently, the theoretical background of GAs was argued in these terms. For some years, however, practical GAs have made increasing use of real number[4] coding in that they are more compact, and thus the resulting GAs are easier to handle. Moreover, Fogel and Ghozeil [1997] have shown that there can be no intrinsic advantage to any choice of cardinality nor, indeed, to any particular choice of representation.

More recently, taking a quite different viewpoint, Vose [1999] has shown that the fundamental mathematical arguments behind (and the validity of) GAs are not upset no matter what coding is used. Vose focuses on the *Simple GA* (SGA) as an evolutionary system rather than as a search or optimisation technique. He starts by defining a broad class of *Random Heuristic Search* (RHS) algorithms and demonstrates that a GA is a special case of an RHS, one in which all the transition rules are first-order Markov chains.

In the meantime Cantú-Paz [2000] has pointed out that GAs do not actually manipulate schemas or building blocks explicitly or, indeed, at all. Schemas are abstract devices, artifices which are used to construct models which attempt to explain how GAs work. They can do this (up to a point) by assuming a binary representation, but tend to come unstuck when alphabets of higher cardinality are used. In short, schemas have their uses but do not give a complete picture of how a GA works.

2.5 Types, Variants and Other Arrangements of GAs

There is no generally-accepted taxonomy, but we would make a distinction between four quite different *types* of GA; the conventional (Holland-type) GA – generally styled the c*anonical* or *simple* GA – Whitley's s*teady-state* GA [Whitley, 1988], Nishio's *interactive* GA [Nishio *et al.,* 1997], and Faulkenaeur's g*rouping* GA [Faulkenauer, 1997]. Within these four types there are many *variants,* of which we studied nineteen in detail and allowed

---

[4] Real numbers in the sense of decimal or integer; anything other than binary or hexadecimal

them to influence the development of our arms race GAs. Nevertheless, our prime concern here is with the canonical GA. A canonical GA is a standard one and is generally accepted as being of the Holland-type, i.e. operating on a population of chromosomes with mutation and crossover operators, and subjecting the population to selection processes whereby the fitter chromosomes reproduce at the expense of the others. ‘Canonical’ does not imply any particular representation; in particular, in the context of the thesis, the word should *not* be exclusively associated with binary representation.

2.6 GAs – A More Detailed Treatment

A GA is an adaptive algorithm for solving problems using computational models of natural evolutionary systems. A GA imbues a computer with a life-like adaptive capability to learn and to control the GA’s environment. It can handle very large search spaces (of many parameters, some of which can be interacting); it can deal with discontinuous search spaces, and it can keep parameters correlated [Sanchez *et al.,* 1997].

Each chromosome can be thought of as a point in the search space of candidate solutions. A GA tries efficiently to find a solution to a problem in a large space of such solutions. First, it evaluates the worth of each chromosome being produced, and then it uses this knowledge to bias the selection process towards reproducing the better chromosomes (i.e. those with good fitnesses) at the expense of the poorer ones [Davis, 1991]. The fitness of a chromosome depends on how well that chromosome solves the problem at hand. A GA knows nothing about the problem being solved, but is a general method of solving ‘search for solutions’ problems. Put formally, a GA is an e*mergent computation* – one in which the actions of simple components with limited information and communication give rise to global information processing. Evolution operates on chromosomes rather than on the systems which the chromosomes encode.

Note. In the development of the thesis we have used *years* (time) and *generations* (GA-parlance) interchangeably.

2.7 Selection: Pairing

There are several ways [Haupt and Haupt, 1998] of pairing a matrix of chromosomes:

- *Pairing from top to bottom of the chromosome list*, e.g. Chromosome 1 with Chromosome 2, Chromosome 3 with Chromosome 4, etc.

- *Random Pairing.* Chromosomes are selected at random from the chromosomes in pairs, e.g. Chromosome 13 with Chromosome 57, Chromosome 26 with Chromosome 83, etc.

- *Weighted Random Pairing.* Probabilities are assigned to chromosomes according to their fitness, the fittest being the one of least cost. The better chromosomes have therefore a much better chance of pairing than the weak ones, although none are entirely excluded. An RNG decides which chromosome is selected. This is generally known as *roulette wheel weighting* [Baker, 1985]. There are two types of this weighting:
  - *Rank weighting.* As evaluation proceeds, a running total is kept of all the fitnesses so far computed. Its accumulated total at the end of the chromosome list is called *total_fitness.* A number, <`random(total_fitness)`>, is then randomly generated for each chromosome. Starting at the top of the list, the first chromosome whose *total_fitness* is greater than the corresponding random number is selected for pairing. The selection process continues in pairs. If a chromosome is paired with itself, another one is chosen.
  - *Cost Weighting.* As for rank weighting, except that fitnesses are linearly normalised first. In the thesis, the best fitness is assigned a new value, *normfit,* of 500, the second best 495, the third best 490, and so forth, and *total fitness* is the sum of the *normfits*. This has advantages if there are large numerical disparities in the fitnesses; otherwise superfit chromosomes get *too big* a reproductive advantage. This method is used in all the GAs in the thesis. 'Bigness' is a matter of choice; we originally introduced a linear norm when we discovered that in one GA an over-dominant individual chromosome was being selected 42 times for parenthood in a population of 100.

- *Tournament Selection.* Only an elite subset comprising the fittest chromosomes are allowed to pair, and the others get no chance (and hence die out, being replaced by the offspring of the elite subset). This is what happens in horse-breeding and cattle farming. The *Breeder Algorithm* [Mühlenbein, 1994] is based on this. Care has to be taken that the elite subset is not too small or the chromosomes will rapidly degenerate (in a GA this means that individual genes will tend to zero)[5]. Tournament selection is in widespread use in theoretical work on GAs [e.g. Cantú-Paz, 2000], but is far less common elsewhere.

---

[5] A very recognisable condition, known as 'degenerative convergence', see Section 2.13

2.8 Mutation and Crossover

Mitchell [1996] summarises the concepts of selection, mutation and crossover thus**:** "*Selection* increasingly focuses the search on subsets of the search space which feature schemas with observed above-average fitness, whereas *crossover* puts high-fitness 'building blocks' together on the same string in order to create strings of increasingly higher fitness. *Mutation* plays the role of an insurance policy, making sure that genetic diversity is never irrevocably lost at any locus".

2.8.1. Mutation

Schrödinger [1954] defines mutation as an isomeric change[6] in some part of the gene's mole-cule. Once two of our electronic chromosomes have been selected for reproduction they are laid side by side, element by element, in the algorithm. A random number then decides whether or not each element is to be mutated; if so, then the element is changed in some way. The mutation rate (more pedantically *the mutation probability per locus*) is usually very small, typically 0.8%, but in terms of the mechanics of the GA it is significant in that:

- By widening the search space, it expands the chromosome in ways not hitherto feasible.
- It helps prevent premature convergence.

Table 2.1 Mutating a chromosome. Mutation rate 20%

Original chromosome	8 4 6 2 3 8 1 6 5 7		.
Mutate by adding 5 where shown	5 5	'carry' if necessary.	.
Mutated chromosome	8 **9** 6 2 3 8 1 **7** **0** 7		.

One can start with one mutation rate and progressively *reduce* it at runtime, since too much mutation is undesirable once convergence approaches; moreover, we have known excessive mutation to make a GA unstable.

2.8.2 Crossover

Mating is the creation of one or more offspring from the selected parents. In a sense, the GA is exploring the cost surface, making use of the schemas present in the chromosomes. A 'crossover point' is selected; in many applications this point would have been chosen randomly between elements 2 and $L - 1$, but for our purposes we chose typically 55% to 65% of L, figures originally recommended by Davis [1991] which we found satisfactory for our application. Parent 1 passes all the elements beyond that point to Parent 2, but keeps the

---

[6] In chemistry, a molecule is changed isomerically if the product consists of the same atoms in a different arrangement. In biology it means a different element in the same locus (see Glossary). Either way, a mutation is a quantum jump, in the sense understood by quantum physicists

remainder. Parent 2 does the same to Parent 1. The two resulting chromosomes (the 'offspring') now contain a sizeable number of elements of both parents but nevertheless, like human children, both are now individuals in their own right. This is called 'single-point' crossover.

Table 2.2 Single-point crossover at 60%, counting from the right [7]

	Chromosome 1	4 7 5 6 |2 3 6 0 1 8	.
	Chromosome 2	3 5 8 9 |2 9 5 1 5 4	.
become			
	Chromosome 1*A*	4 7 5 6 |2 9 5 1 5 4	.
	Chromosome 2*A*	3 5 8 9 |2 3 6 0 1 8	.

Table 2.3 Two-point crossover at 30% and at 80%

	Chromosome 1	4 7 |5 9 2 3 6 |0 1 8	.
	Chromosome 2	3 5 |8 6 2 9 5 |1 5 4	.
become			
	Chromosome 1A	4 7 |8 6 2 9 5 |0 1 8	
	Chromosome 2A	3 5 |5 9 2 3 6 |1 5 4	.

Both one-point and two-point crossover occur in nature. Schemas which occur at either end of a chromosome will, in single-point crossover, never be re-combined; no matter where crossover occurs, the schema will be broken up. Two-point crossover is arguably one answer, but it will not solve every problem, even though Dumitresceau *et al.* [2000] claim that two-point crossover is the best for minimising disruptive effects.

2.8.3 The Practicalities of Handling Mutation and Crossover

- Our chromosomes are a series of concatenated genes each one of which consists of a number of elements. Setting an arbitrary crossover of 60%, say, may well result in a chromosome being split mid-gene. Similarly, carrying the '1' of the '10' into the next column (see the example in Table 2.1 above) may well result in the mutation of a neighbouring gene. Our computations show that, numerically, neither splitting a gene nor mutating a neighbour seem to matter very much, although in (6) of Section 8.9 we point out that a crossover inside a gene can cause additional (and unintended) 'mutations'.

- Mutation and crossover actually support one another, for at any point there is a subset of operator fitnesses which will lead to best results. After initialisation, the population is hugely diverse, but scattered within it are pieces of good solutions. Crossover is arguably the best way to put these good pieces together, but once this has been done the

---

[7] It does not actually matter from which side counting is done, as long as it is consistent

chromosomes start to converge, and from then on it has been our experience that mutation leads the search for better solutions. If the runs are for a large number (> 50) of generations, however, mutation can become excessive.

- It is possible to be too clever. In an article of the same name, Rudolph [2001] showed that "Self-adaptive mutations may lead to premature convergence".

2.9 Fitness and Fitness Functions

The Glossary defines *fitness* and gives four alternative definitions of *fitness function* (ffn), the first three of which agree that an ffn provides a measure of how individuals (chromosomes) have behaved in the problem domain. Fitnesses based on the direct output of evaluation equations ('evaluation-is-fitness') will favour the superfit individual, so it is better to introduce a linear norm which will retain an individual's performance relative to its peers. In general terms,

$$F(x_i) = \frac{f(x_i)}{\sum_{i=1}^{population} f(x_i)} \quad , \qquad .$$

where the individual fitness, $F(x_i)$ is computed as the individual's raw performance, $f(x_i)$, relative to the whole population, and x_i is the phenotypic value of individual i**;** ffns can be non-linear and even discontinuous [Sanchez *et al.,* 1997].

"Parent selection dynamics are based on an application-dependent metric known as *fitness*. A fitness is a figure of merit computed by an ffn using any domain knowledge which applies, and imposes constraints on what is required, or what is acceptable behaviour. The greater a chromosome's fitness, the higher chance it has of being selected for reproduction" [Buckles and Petry, 1992]. An ffn (sometimes called an *objective function*) is generally chosen for reasons other than mathematics; in our case, it was chosen for political reasons. "Anyone who has spent time applying GAs knows how good they are at debugging ffn's. Any flaw in an ffn that *can* be exploited by the GA *will* be" [Drechsler, 2003]. In Section 5.2 (indeed, throughout the thesis) we say that we wish to minimise the risk of war, so that is what our fitness functions must do. Moreover, they must be expressed in terms of either sides' defence expenditures – the only information we have. At first sight the risk of war might appear least if both sides minimised their defence expenditures, but this is not true. One of the most noticeable characteristics of the charts of percentage defence expenditure drawn from the IMF data for all three conflicts (Greece/Turkey – India/Pakistan – Middle

East, see Figures 6.5, 6.10 and 6.15, respectively) is that when percentage defence expenditure curves intersect, touch or even get close to one another, a conflict often results. (Conflicts occur at other times as well, or there would be no need for this present study). In other words, the further apart the defexp% curves are, the less is the likelihood of war.

Taking our usual x and y to represent defence expenditures, then it is argued that the risk of war is least when $| x - y |$ is *maximised.* Looked at this way, we bestow a higher fitness or 'worth' on those chromosomes where the differences between x and y are larger, and by doing this we ensure that such chromosomes have a greater chance of reproduction in the GA. As the GA progresses, therefore, the risk of conflict is systematically reduced.

A totally different approach is taken in Chapters 5, 6 and 7, where we seek to *minimise* the value of the ffn. In particular, we use the inverse Pythagorean expressions

$$fitness = \frac{10000}{\sqrt{1+x^2+y^2}} \text{ (Chapters 5/6)} \quad \text{or} \quad fitness = \frac{10000000}{\sqrt{1+P^2+Q^2}} \text{ (Chapter 7)}, \qquad (2.3)$$

which minimise x and y, or P and Q, simultaneously. (P and Q stand for the American and Soviet *counter-values*, respectively). The numerators are scaling factors. Note (5) to Section 5.4.2 covers the subject in more detail.

In Section 4.6 we point out that although it is normally convenient to use an RNG for gene initialisation, nevertheless our GAs make use where possible of real-world data. The introduction of Cao *et al.*'s norm is one such example, as is Section 6.3.2, where it will be suggested that since we know all the real (i.e. IMF) values of x_t, y_t, x_m, and y_m (collectively called x_{real} and y_{real}) we can use the prediction GA's evaluation equations, i.e. Eqs(6.4), to obtain x_{t+1} and y_{t+1}, etc (collectively called x_{calc} and y_{calc}) as a *straightforward arithmetical exercise* not involving GAs at all; x_{real} and y_{real} can then be compared with x_{calc} and y_{calc}.

We later took these ideas one stage further by using the ffn

$$fitness = \sqrt{\left|x_{calc} - x_{real}\right|^2 + \left|y_{calc} - y_{real}\right|^2}\ , \qquad (2.4)$$

thereby taking all the available IMF data into account. We will show in Chapter 6 that, although curves computed by the GA using Eq(2.4) mimic the IMF data much more closely than those computed using Eq(2.3), nevertheless there is little difference between the *rate-factors* used for either. In the context of arms race prediction, it is important that *appropriate* rate-factors are transferred from the modelling GA to the prediction GA.

2.10 Binary or Real-valued Encoding?

As in any search-and-learning method, the way in which candidate solutions are encoded is the central factor in the success or otherwise of a GA. As has already been explained, GAs have traditionally employed chromosomes consisting of fixed-length, fixed-order bit strings, but this is principally for historical reasons. GA theory can apply to non-binary codings, such as integer or decimal (variously called *real-valued*, *floating-point* or *continuous parameter* coding). Real-valued or integer strings have three very real practical advantages over binary ones [Wright, 1991; Michalewicz, 1992] :

- The efficiency of the GA is increased since there is no need to convert chromosomes to phenotypes before each function evaluation.
- If the parameters are continuous (i.e. not easily quantised), then the use of real, floating-point numbers allows representation to machine precision without discretisation loss.
- Less memory is required as floating point (or integer) internal computer representations can be used directly

Holland himself later [1992] compared two encodings with roughly the same information-carrying capacity, one with a small number of elements and long strings (bit strings of length 100), and the other with a large number of elements and short strings (decimal strings of length 30). He argued that the former allows for a higher degree of implicit parallelism than the latter (i.e. 2^{100} versus 10^{30}), since an instance of the former contains more schemas than an instance of the latter. Such schema-counting analyses have subsequently been ques-tioned. Indeed, Vose and Liepens [1990] believe them to be wrong.

Aizawa and Wah [1993] claim that the presence of *noise* alleviates tendencies towards premature (and degraded) convergence. Convergence is covered in Section 2.13.

Many GA practitioners [Eshelman and Shaffer, 1993; Montana and Davis, 1989; Adewuya, 1996; and Michalewicz, 1992] have used real-valued encodings and to date there are no hard and fast rules as to which, binary or real-valued, is the better. For many years it has seemed highly probable that fundamental GA theory and processes were not upset by the use of real-valued strings, but until Vose [1999] there was no formal proof of that.

Michalewicz [1992] has said "All experiments indicate that floating-point representation is faster, more consistent from run to run, and provides higher precision, especially with large domains where binary coding would require prohibitively long representations".

2.11 The Representation Used for the GAs Designed for this Application

Only two general points about representation will be made here. More details about the GAs we employ are given in Chapter 4.

2.11.1 Integer-Coded GAs

For the work on Richardson's arms race equations described in this thesis, *integer encoding* was deliberately chosen. This is not a new idea; Lucasius and Kateman [1992] have said that GA chromosomes for subset selection and combinatorial optimisation problems are best suited by integer representation.

2.11.2 Direct Mapping

It so happens that most of the figures resulting from the Richardson arms race equations lie in the thousands. It therefore seemed apt to map the phenotype direct onto the genotype (i.e. avoiding the need for any embryogeny), and use a genotype consisting of strings of mainly four-digit decimal genes, seemingly integers, the only 'coding' necessary being occasional multiplication or division by factors of ten, in order to compute the evaluation equations. Many of these apparent integers will be floating-point numbers and some will be integers, but this does not matter provided that no decimal point ever appears in the genotype. The error introduced by ignoring decimal places is minimal.[8] This is a long way from John Holland's original idea of using binary encoding, but as Vose [1999] has now shown, *the theory of a GA is valid no matter what encoding is used.* If a term is needed, it would be fair to describe all the GAs used in the thesis as 'integer-coded' (to base 10).

2.12 Epistasis

In GA literature, the degree of dependent parameter interaction is called *epistasis*, a biological term for gene interaction. Epistasis is about genes acting in combination to produce (or inhibit) solutions. Michalewicz [1992] says "Epistasis measures the extent to which the contribution to fitness of one gene depends on the values (alleles) of other genes. A high degree of epistasis means that building blocks may not form, and the problem becomes *deceptive*. Bentley [1999] puts it differently: "Epistasis is defined by the genotype (and embryogeny) representation, and not by the ffn. A genetic representation with high epistasis may have many genes whose phenotypic effect relies to a large extent on the alleles of other genes. The converse is true; a representation with low epistasis has few or no genes whose phenotypic effect relies on the alleles of other genes". Experiments investigating whether the level of epistasis should be high or low have so far been inconclusive [Schoenhauer, 1996]. Bentley [1999] considered a fictional representation that used, say, ten genes to represent the entire form of designs, and the phenotypic effect of every gene was completely dependent on all other genes through some process of embryogeny. With maxi-mum epistasis, the ten genes were effectively one. Any attempt to improve one gene would have resulted in changes to all the rest of the design (pleiotropy), making evolution very difficult, if not impossible [Bentley, 1999].

Alternatively, consider some representation with no epistasis. No embryogeny is required, since every gene maps directly onto a specified area (and only that area) of the phenotype. Hence evolution of large-scale characteristics would be immensely difficult. Nature uses 'middling' amounts [Bentley, 1999]. In GAs less epistasis is preferable to more, and this is one reason for introducing redundancy into chromosomes, see Chapter 8. In particular, we found that reducing epistasis makes GAs significantly more robust (i.e. the results are less likely to differ with each starting seed).

An intriguing method of measuring epistasis has been developed by de Jong *et al.* [1997] with a mathematical backing by Naudts *et al.* [1998]. A GA is run complete (i.e. with both mutation and crossover) and a graph of fitness against generations run is plotted. It is then run again keeping crossover but cutting out mutation. It is then run for the third time, keeping mutation but cutting out crossover. All three graphs are now plotted on the same axes. Put crudely, *the closer the graphs, the greater the epistasis*, and vice versa. On a basis of experience, it is expected that the (standard) GA will dominate on low epistasis problems (and it does, as demonstrated in Chapter 8). If epistasis becomes very large, the mutation-only version does better. We will use the de Jong/ Naudts technique later.

2.13 Convergence

In the GA research community the term *convergence* is encountered under various guises. Listed below, for the sake of clarification, are six different versions of it which are used in the thesis. (A discussion about the mathematics of convergence is outside the scope of the thesis.) Except where stated, the taxonomy is our own.

---

[8] The maximum error will be 0.5 in a thousand or 0.05%

(1) *Oscillatory convergence* Successive iterations (generations) oscillate diminishingly either side of a limit.

(2) *Asymptotic convergence* Successive iterations (generations) nudge closer and closer towards a limit.

(3) *Degenerative convergence* The genes of a chromosome tend to zero. This is brought about by allowing too few chromosomes to reproduce, and it has a biological parallel in excessive in-breeding.

(4) *Premature convergence* [Levenick, 1999] This implies that convergence is desirable but should somehow be delayed. In fact, this phrase is often applied to a GA which has converged to something other than the desired value.

(5) *Homogeneous convergence* This implies homogeneity across the whole population to the point where all individuals are identical, i.e. *clones.* This induces a loss of genetic diversity which is never desirable, for it incapacitates crossover and inhibits exploration except by mutation.

(6) *Diverse convergence* [Levenick, 1999]. This implies " ... the maintenance of a diverse population after convergence is reached" This would be an ideal, for it would combine exploration *and* exploitation, Holland's [1975] basic dilemma. Alternatively, it *could* just be a result of frequency-dependent selection.

In Section 7.14 and Table 7.8, we encounter premature convergence in GA-18 (a sequel to GA-2); different seeds applied to non-discretised evaluation equations give markedly different results (violent oscillations followed by rapid convergence). However, those solutions which have high fitness converge to very similar values. We can slow convergence by

- introducing redundant genes (see Chapter 8), and by
- including the evaluated results as genes in the chromosome, i.e. by discretising the evaluation equations.

The advantage of slowing down convergence is that robustness is greatly improved, i.e. the results depend only marginally on the starting conditions.

CHAPTER 3

Background 2: Richardson's Theory of Arms Races

3.1 Richardson's Assumptions

A very brief history of arms races and of Richardson's work were given in Section 1.2. On the basis that "... where your treasure is, there will your heart be ...", Richardson believed that a nation's annual budget gives a very accurate portrayal of its leaders' hopes, aspirations, fears, grievances, prejudices, envy, compassion, humanity, ethics and social *mores* generally. In his model, a nation's strength is determined principally by its current annual expenditure on arms, although it partly depends also on a term denoting "threats minus co-operation".

This chapter explores Richardson's arms race model for two and three nations. Although originally written for x and y (and z) to represent various defence costs, it can equally represent the actual number of weapons. It is shown that discretised arms race equations are of much more use than straightforward ordinary differential equations if only because real information is catalogued by specific years rather than by rates of change.

Canard explosions modelled by ordinary differential equations have been considered in depth by Peng *et al.* [1991]. We connect this work to Richardson's model and this leads to an investigation of how to predict instabilities, and therefore anticipate conflicts, between nations.

3.2 The 2-nation Model

Richardson stressed that he was searching not only for truth, but truth in easily understood form. It was natural, therefore, that he should turn to *Ockham's razor* ("Entities are not to be multiplied without necessity") and start simply. Let there be two nations, X and Y, whose defence expenditure per year is US\$ x billion and US\$ y billion, respectively. Suppose the Defence Minister of Nation X (or Y) makes the following statement: "The intentions of our country are entirely peaceful. We have given ample evidence of this by the treaties which we have recently concluded with our neighbours.

Yet, when we consider the state of unrest in the world at large and the menaces by which we are surrounded, we should be failing in our duty as a government if we did not take adequate steps to increase the defences of our beloved land".

This rhetoric reduces to $\frac{dx}{dt} = cy,$ where $\frac{dx}{dt}$ is the rate of change of the defence expenditure of Nation X, and y represents the defence expenditure of Nation Y; in a 2-nation setting, Y is the only 'menace' surrounding X.

The Defence Minister's opposite number in Nation Y could make a similar statement, result-ing in $\frac{dy}{dt} = ax;$ a and c are called *defence coefficients* and are positive. .

It has been fortunate that the sheer cost of armaments exercises some restraint on each side. This diminishes the upward trend of $\frac{dx}{dt}$ and $\frac{dy}{dt}$ by amounts proportional to the nations' own defence expenditures. Hence an improved expression would be

$$\frac{dx}{dt} = cy - fx \quad \text{and} \quad \frac{dy}{dt} = ax - by, \qquad .$$

where f and b are two positive *expense and fatigue* coefficients; in use, their sign gives them a negative impact.

Finally, Richardson introduced two 'constants'. Nation X still feels it needs some military strength, even if Y has none at all. This is for two reasons:

- First, it takes years to build up, equip and train standing navies, armies and air forces. Without any of these, a nation really could be vulnerable to predators, especially if it possesses attractive natural resources, like oilfields (e.g. Kuwait), a natural harbour (Singapore), or rich agricultural land (Poland, which Hitler coveted in 1939).
- Second, some nations (like India and Pakistan) loathe each other, for religious, racial, or historical reasons (such as wanting to avenge previous defeats or perceived humiliations).

Richardson called the 'constants', g and $h,$ the *grievances*. Hence the above equations become

$$\frac{dx}{dt} = cy - fx + g \quad \text{and} \quad \frac{dy}{dt} = ax - by + h \,. \tag{3.1}$$

We observe the following:

1 *Every parameter in these equations is time-dependent.* In general, x and y will change faster with time than a, b, c, f, g or h; the latter vary slowly with time. Put formally, all the parameters in Eqs(3.1) are functions of time and a, for instance, should be written $a(t)$. To avoid overburdening the text, however, we will just use a.

- Mutual disarmament *without* satisfaction is not permanent, for even if $cy - fx$ is zero, the effect of g still rankles on X. (Mutual disarmament *with* satisfaction can be permanent; it has existed, for example, on the frontier between the United States and Canada since 1817).

We next consider what happens if the curves for $\frac{dx}{dt}$ and $\frac{dy}{dt}$ are both equal to zero – in which case the 'curves' become straight lines.

Let x_0, y_0 be the point at which the two lines meet. We call (x_0, y_0) the *point of balance.* After some algebraic manipulation, we obtain

$$x_0 = \frac{ch + gb}{fb - ac} \quad \text{and} \quad y_0 = \frac{fh + ga}{fb - ac} \tag{3.2}$$

provided that $ac \neq fb$; if $ac = fb$, then the lines for $\frac{dx}{dt}$ and $\frac{dy}{dt}$ would be parallel, and no point of balance would exist except at infinity, but this condition is very unlikely. In addition, a, b, c and f are positive, and g, h are likely to be so. In our scenarios the numerators of Eqs(3.2) are unlikely to be zero, so the indeterminate 0/0 situation does not arise.

Define new co-ordinates $\hat{X}$ and $\hat{Y}$, such that $\hat{X} = x - x_0$ and $\hat{Y} = y - y_0$. Since

$$\frac{dx_0}{dt} = 0 = cy_0 - fx_0 + g \quad \text{and} \quad \frac{dy_0}{dt} = 0 = ax_0 - by_0 + h \,, \quad .$$

on using Eq(3.1), it follows that .

.

$$\frac{d\hat{X}}{dt} = -f\hat{X} + c\hat{Y} \quad \text{and} \quad \frac{d\hat{Y}}{dt} = -b\hat{Y} + a\hat{X} \; . \tag{3.3}$$

The substitutions $\hat{X} = Ae^{pt}$ and $\hat{Y} = Be^{pt}$ form a solution to Eqs(3.3), provided p is one of the roots, say q or r, of the quadratic

$$p^2 + (f + b)p + fb - ac = 0 \; . \tag{3.4}$$

On letting

$$\hat{X} = A_1 e^{qt} + A_2 e^{rt} \quad \text{and} \quad \hat{Y} = B_1 e^{qt} + B_2 e^{rt} \tag{3.5}$$

and substituting Eqs(3.5) into Eqs(3.3), we obtain

$$\frac{A_1}{B_1} = \frac{c}{q+f} = \frac{q+b}{a} \quad \text{and} \quad \frac{A_2}{B_2} = \frac{c}{r+f} = \frac{r+b}{a} \,. \tag{3.6}$$

This shows that the constants A_1 , A_2 , B_1 and B_2 are not independent. This in turn supports the earlier statement (Section 1.2) that Richardson's model is essentially a non-linear dynamic system, the behaviour of which can be difficult to predict.

So far we have considered *rates* of change against time. But how does y vary against x? Ignoring the grievances, setting $\frac{dy}{dt}$ and $\frac{dx}{dt}$ to zero and differentiating with respect to x, we obtain

$$\frac{dy}{dx} = \frac{a}{b} \quad \text{and} \quad \frac{dy}{dx} = \frac{f}{c} \,. \tag{3.7}$$

Richardson proved via a geometrical analysis using equally-scaled axes [Richardson, 1960b, pp 24-27] that $\frac{dx}{dt}$ is positive above the curve (see Figure 5.1)

$$ax - by + h = 0 \tag{3.8}$$

and that $\frac{dy}{dt}$ is positive to the right of the curve

$$cy - fx + g = 0 \,. \tag{3.9}$$

If the slope $\frac{dy}{dx}$ of Eq(3.8) *exceeds* that of Eq(3.9), there will be a region in which both $\frac{dx}{dt}$ and $\frac{dy}{dt}$ are positive lying on that side of the point of balance (x_0, y_0) in which $x > x_0$ and $y > y_0$. The resulting situation diverges away from the point of balance, and is therefore unstable. In other words, since $\frac{dy}{dx}$ can also be expressed in terms of Eqs(3.7), instab-ility occurs when $ac > bf$.

A similar argument for the slope of Eq(3.9) being greater than that of Eq(3.8) shows that, when $ac < bf$, the situation converges towards the point of balance and is therefore stable.

We note that the same results can be found by standard algebraic means.

Finally, we note that:

1 If $ac = bf$, then the system would seem to be in mechanically neutral equilibrium. In practice, however, the transition from stability to instability can take place *explosively*. See Figures 5.9 and 5.10, for the difference that a 0.1 % increase in average defexp% can make.

2 If $a = c$ and $b = f$, then Eqs(3.1) reduce to

$$\frac{d(x+y)}{dt} = (c-b)(x+y) + g + h \quad , \qquad .$$

namely a system with a single dependent variable.

3.3 The 3-nation Model

Generalising the 2-nation equations, Richardson [1960b, p 146] expressed the n-nation situation by

$$\frac{dx_i}{dt} = g_i + \sum_{j=1}^{n} \kappa_{i,j} x_j \qquad i, j \in \{ 1, 2, 3, \dots n\}, \qquad (3.10)$$

where x_i is a measure of 'threats minus co-operation' for the ith nation and g_i and $\kappa_{i,j}$ are time-varying quantities.

On setting $x_1 = x$, $x_2 = y$ and $x_3 = z$, the 3-nation version of Eq(3.10) yields

$$\frac{dx}{dt} = -\kappa_{1,1} x + \kappa_{1,2} y + \kappa_{1,3} z + g_1 \qquad .$$

$$\frac{dy}{dt} = \kappa_{2,1} x - \kappa_{2,2} y + \kappa_{2,3} z + g_2 \qquad .$$

$$\frac{dz}{dt} = \kappa_{3,1} x + \kappa_{3,2} y - \kappa_{3,3} z + g_3 , \qquad (3.11)$$

.

where $\kappa_{1,1}$, $\kappa_{2,2}$ and $\kappa_{3,3}$ are *rate-factors* representing the *fatigue and expense coefficients*, and the remaining six $\kappa_{i,j}$ rate-factors represent the *defence coefficients*, see Section 3.2.

Let $\hat{X} = x - x_0$, $\hat{Y} = y - y_0$ and $\hat{Z} = z - z_0$. Then, as before, it follows that

$$\frac{d\hat{X}}{dt} = -\kappa_{1,1}\hat{X} + \kappa_{1,2}\hat{Y} + \kappa_{1,3}\hat{Z} \qquad .$$

$$\frac{d\hat{Y}}{dt} = \kappa_{2,1}\hat{X} - \kappa_{2,2}\hat{Y} + \kappa_{2,3}\hat{Z} \qquad .$$

$$\frac{d\hat{Z}}{dt} = \kappa_{3,1}\hat{X} + \kappa_{3,2}\hat{Y} - \kappa_{3,3}\hat{Z}. \qquad (3.12)$$

.

Eqs(3.11) were further developed and extended by Mayer-Kress [1989], who developed a *discretised* form, namely

$$x_{t+1} = x_t + \left(k_{1,1}(x_s - x_t) + k_{2,3}(y_t + z_t)\right)(x_m - x_t) \qquad .$$

$$y_{t+1} = y_t + \left(k_{2,2}(y_s - y_t) + k_{1,3}(x_t - z_t)\right)(y_m - y_t) \qquad .$$

$$z_{t+1} = z_t + \left(k_{3,3}(z_s - z_t) + k_{1,2}(x_t - y_t)\right)(z_m - z_t). \qquad (3.13)$$

At this juncture, several points are noted.

1 The discretised Eqs(3.13) are a lot more useful than Eqs(3.11), if only because real-world data is expressed in figures for individual years. For example, nation X's defence expenditure for 1987/1988/1989 will have three discrete entries in the various statistical yearbooks, corresponding to x_{t-2}, x_{t-1} and x_t, and these can be extrapolated to predict x_{t+1}, the defence expenditure for 1990.

2 As written, Eqs(3.13) assume that *nations* Y *and* Z *are allied together against nation* X.

3 The terms $(y_t + z_t)$, $(x_t - z_t)$, and $(x_t - y_t)$ denote the external threat from adversaries for the nations X, Y and Z, respectively. Note the plus and minus signs in the 'threat' parentheses; in the second equation (nation Y), for instance, since Y and Z are allies, any defence expenditure incurred by Z is to Y's advantage. (The subject comes up again in Section 4.11 where the rate-factors associated with threat play a leading role in assessing the stability of the fractal basins).

- x_t, y_t and z_t are the *actual* expenditures of the three nations on arms for the current year. x_{t+1}, y_{t+1}, z_{t+1} are the corresponding defence expenditures for next year.

4 x_s , y_s and z_s are the intrinsic arms expenditures (how much each country spends on defence irrespective of competitive spending by its neighbours). This is difficult to estimate, so the British figure of 78% standing costs [UK Defence Estimates, 1998] has been substituted; thus, for instance, $k_{1,1}\,(x_s - x_t) = -\,0.22\ \ k_{1,1}\ x_t$.

- x_m , y_m, and z_m , respectively, represent the expenditures authorised on arms in the budgets of nations X, Y and Z. Under arms race conditions $x_m \neq x_t$, $y_m \neq y_t$, and $z_m \neq z_t$.

Our 2-nation variant of Eqs(3.13) is given by

$$\begin{aligned} x_{t+1} &= x_t + \big(k_{1,1}\,(x_s - x_t) + k_{1,2}\ y_t\big)(\,x_m - x_t\,) \qquad . \\ y_{t+1} &= y_t + \big(k_{2,2}\,(y_s - y_t) + k_{2,1}\ x_t\big)(\,y_m - y_t) \,. \qquad (3.14) \end{aligned}$$

We know from experience amassed over many years that x_s is typically 80% of x_t . Assuming that the terms $k_{1,1}(x_s - x_t)$ and $k_{2,2}(y_s - y_t)$ are negligible vis-à-vis $k_{1,2}\,y_t\,(x_m - x_t)$ and $k_{2,1}\,x_t\,(\,y_m - y_t)$, respectively – which we confirm from the computed results and verify in Chapter 6 – then

$$\begin{aligned} \frac{dx}{dt} &\approx (x_{t+1} - x_t) \approx k_{1,2}\,y_t\,(x_m - x_t) \approx (k_{1,2}\,x_m)\ y_t - (k_{1,2}\ y_t)\,x_t \qquad . \\ \frac{dy}{dt} &\approx (y_{t+1} - y_t) \approx k_{2,1}\,x_t\,(y_m - y_t) \approx (k_{2,1}\,y_m)\ x_t - (k_{2,1}\ x_t)\ y_t \qquad (3.15) \end{aligned}$$

which are of the general form

$$\frac{dx}{dt} = cy - fx \quad \text{and} \quad \frac{dy}{dt} = ax - by \,, \qquad .$$

cf. Eqs(3.1). Under the said assumptions, we have obtained a relationship between Richardson’s coefficients a and c and Mayer-Kress’s rate-factors. Hence

$$a \approx k_{2,1}\,y_m \quad \text{and} \quad c \approx k_{1,2}\,x_m \qquad (3.16)$$

which we use extensively in the GAs of Chapter 6 to estimate $a - c$ for the purposes of detecting a change in the direction of curvature of the limit cycle. The results of

Chapter 6 seem amply to justify these approximations.

Consider the 3-nation case, i.e. nations X, Y and Z. In such circumstances, nations usually 'take sides' so we are likely to have X versus Y and Z together, for example. In Eqs(3.13), $k_{2,3}$ is associated with the budgeted defence expenditure x_m of nation X, $k_{1,3}$ with the y_m of nation Y, and $k_{1,2}$ with the z_m of nation Z. Hence, the 3-nation approximation expression for $a - c$ is

$$k_{1,3}\, y_m - (k_{2,3}\, x_m - k_{1,2}\, z_m) , \tag{3.17}$$

which we use in Chapter 6. .

Forrest and Mayer-Kress [1996] originally called the $k_{i,,j}$ *rate-constants*, controlling the *defence intensity*, i.e. how quickly a nation responds to changes in external threats. In fact, 'rate-constant' is not a good name for they are not constant (indeed, they appear as genes in our GAs) and they do vary significantly, if slowly. In the thesis we will call them *rate-factors.*

3.4 Using Richardson's Model.

Richardson's equations may be conceptually correct, but they are sometimes difficult to apply in practice, in that they often call for parameters which are either not available or are meaningless in context; for example, requiring the assignment of a numerical value to *goodwill* or *hostility*. The skill in using the said equations lies, therefore, in identifying Richardson's intentions. Sometimes, this is easy; for example, if y is the actual *number* of US nuclear missiles − not defence expenditure as hitherto − and x is the *number* of Russian missiles, then Eqs(3.1) apply, and the analysis can proceed from there, plotting contours which represent the minimum number of nuclear weapons with which America and Russia can inflict unacceptable levels of damage on each other *after* each one has survived a pre-emptive strike by the other. (Nuclear deterrence is the subject of Chapter 7). At other times, Richardson can be difficult to gauge, especially when it is necessary to put numerical values on terms like threats and co-operation.

In order to achieve a pragmatic solution in the case of the Middle East conflict, for example, Eqs(3.13) could be modified to yield

$$x_{t+1} = x_t + \left(-0.22\ x_t\ k_{1,1}/1000 + (y_t + z_t)\, k_{2,3}/1000\right)(GDP_X/50) \quad .$$

$$y_{t+1} = y_t + \left(-0.22\ y_t\ k_{2,2}/1000 + (x_t - z_t)\, k_{1,3}/1000\right)(GDP_Y/50)$$

$$z_{t+1} = z_t + \left(-0.22\ z_t\ k_{3,3}/1000 + (x_t - y_t)\, k_{1,2}/1000\right)(GDP_Z/50) \quad . \qquad (3.18)$$

This is effected by replacing $k_{1,1}(x_s - x_t)$ by $-0.22\ x_t\, k_{1,1}$, $k_{2,2}(y_s - y_t)$ by $-0.22\ y_t\, k_{2,2}$, and $k_{3,3}(z_s - z_t)$ by $-0.22\ z_t\ k_{3,3}$. Additionally, it might be found empirically, for instance, that $x_m - x_t$ could be approximated by taking one-fiftieth of nation X's GDP, and the same obtained for nations Y and Z. The number 1000 is a scaling factor.

Note. Eqs(3.18) merely demonstrate a technique for reducing the number of variables in the evaluation equations of modelling GAs. This technique is used in all four modelling GAs in Chapters 5 and 6 (Eqs(5.4), Eqs(6.2), Eqs(6.6), and Eqs(6.11)). The actual figures for Eqs(6.11) are quite different from those used for demonstration in Eqs(3.18) above.

In Eqs(3.6) it was shown that the constants of the general solution given in Eqs(3.5), namely A_1, A_2, B_1 and B_2, are not independent. This lack of independence is particularly noticeable in the 3-nation prediction GA, GA-4 (Section 6.5.3). We tried making a small (10%) change to each of the rate-factors in turn and this resulted in quite different outputs (not illustrated). Such changes can lead to

- a changed order of 'fitness', leading for instance to different parents being selected for reproduction
- a change of domination (see Section 4.9), a potentially destabilising situation.

In this sense Eqs(3.18) are *entangled* in the sense used by quantum physicists. For this reason, we go to a lot of trouble in the modelling GAs of Chapters 5 and 6 to ensure that the rate-factors are correct *before* they are used in a prediction GA.

3.5 Canard Explosions

It was suggested by my supervisor that instabilities in the model of a certain chemical process − the Edblom-Orbán-Epstein ('EOE') reaction [Peng *et al.,* 1991] − bear a striking resem-blance to those modelled by the 2-nation Richardson arms race equations, and that ideas could usefully be borrowed from the Peng *et al.* analysis of *canard explosions* and employed here to predict instabilities in the Richardson arms race equations. The EOE reaction inv-olves very rapid changes from stable to unstable states and may result in a canard explosion.

A canard explosion is the sudden growth in the amplitude and period of a *limit cycle* of a system of non-linear ordinary differential equations, such as

$$\frac{dx}{dt} = F(x, y, c) \quad \text{and} \quad \frac{dy}{dt} = G(x, y, c)\,, \tag{3.19}$$

in a very narrow range of the parameter c ($c \geq 0$).

The phrase canard explosion was first used in connection with a mixture of iodate, sulphite and ferrocyanide ions in a stirred tank reactor. This mixture is used industrially in the manu-facture of plastics. It appears that the inflow concentration is critical; a minute change in the proportions can cause an otherwise dull-looking sludge to explode violently without prior warning. Such volatile behaviour is clearly unwelcome in an industrial plant, so ample res-ources were allocated to ensure its prevention. The EOE reaction was originally analysed and modelled by Peng *et al.* [1991] who used the van der Pol equation first developed by Benoit *et al.* [1981] and later investigated by Brøns and Bar-Eli [1994].

The *limit cycle* is defined by

$$\frac{d\left(\frac{\left(\frac{dY}{dt}\right)}{\left(\frac{dX}{dt}\right)}\right)}{dt} = 0 \quad . \tag{3.20}$$

Peng *et al.* [1991] found that

1 In practice, explosions *always* occurred just after the limit cycle reached a point with zero curvature (i.e. it touched the *inflection line* tangentially) or when the limit cycle changed the direction of its curvature, for at that point the steady state changed. This is called the *inflection canard point*, and the easiest way to calculate it is to analyse the curvature of the limit cycle. Furthermore, if the curvature of the limit cycle *approached* zero (it does not have to get there), an explosion is imminent. This flattening of the limit cycle's curvature is now used *to predict* (and hence prevent) canard explosions.

2 If the model is correct, the EOE reaction consists of two stable steady states lying either side of an unstable one.

In summary, Peng *et al.* [1991] showed that explosions *always* occur just after the direction of curvature of the limit cycle had changed. Furthermore, such changes were easy to forecast.

To what extent is the above pertinent to the Richardson arms race equations? At first sight there was no obvious connection until it became apparent that the 'patchiness' of blue in the fractal basin in Figure 5.14 was not due to noise, as was first thought, but to definite regions of instability, some of which were quite local. Stable states lying either side of an unstable one are exactly what the EOE equations are trying to model.

3.6 Applying Canard Explosion Theory to Richardson's Equations

Consider the path of the limit cycle, namely Eq(3.20). By using Eq(3.20) and Eqs(3.1), it can be shown that if $ac = bf$ and $g = h = 0$, then every *trajectory* is a straight line.

On adopting polar co-ordinates (with the origin at the point of balance), we have

$$\hat{X} = x - x_0 = r\cos\theta \quad \text{and} \quad \hat{Y} = y - y_0 = r\sin\theta. \tag{3.21}$$

On taking partial derivatives, we have

$$\frac{dx}{dt} = \frac{d\hat{X}}{dt} = \frac{\partial\hat{X}}{\partial r}\frac{dr}{dt} + \frac{\partial\hat{X}}{\partial\theta}\frac{d\theta}{dt} = \cos\theta\frac{dr}{dt} - r\sin\theta\frac{d\theta}{dt} \tag{3.22}$$

$$\frac{dy}{dt} = \frac{d\hat{Y}}{dt} = \frac{\partial\hat{Y}}{\partial r}\frac{dr}{dt} + \frac{\partial\hat{Y}}{\partial\theta}\frac{d\theta}{dt} = \sin\theta\frac{dr}{dt} + r\cos\theta\frac{d\theta}{dt}. \tag{3.23}$$

We next multiply Eq(3.22) by $\cos\theta$ and Eq(3.23) by $\sin\theta$, and then add. On using Eqs(3.1) we obtain

$$(\cos^2\theta + \sin^2\theta)\frac{dr}{dt} = \cos\theta\big(c(y_0 + r\sin\theta) - f(x_0 + r\cos\theta) + g\big) + \sin\theta\big(a(x_0 + r\cos\theta) - b(y_0 + r\sin\theta) + h\big).$$

whence

$$\frac{dr}{dt} = cr\sin\theta\cos\theta - fr\cos^2\theta + ar\cos\theta\sin\theta - br\sin^2\theta$$
$$= r(-f\cos^2\theta + (a+c)\sin\theta\cos\theta - b\sin^2\theta).$$

Hence

$$\frac{1}{r}\frac{dr}{dt} = \frac{d(\log r)}{dt} = -f\cos^2\theta + (a+c)\cos\theta\sin\theta - b\sin^2\theta. \tag{3.24}$$

Next we multiply Eq(3.22) by $\sin\theta$ and Eq(3.23) by $\cos\theta$, and then subtract. As before, it can be shown that

$$\frac{d\theta}{dt} = a\cos^2\theta + (f - b)\cos\theta\sin\theta - c\sin^2\theta. \tag{3.25}$$

From Eq(3.25) we see that

$$\frac{d(2\theta)}{dt} = (a-c) + (a+c)\cos 2\theta + (f-b)\sin 2\theta. \tag{3.26}$$

Now let a constant angle E be defined by

$$\cos 2E = \frac{a+c}{Q} \quad \text{and} \quad \sin 2E = \frac{f-b}{Q},$$

where $Q = \sqrt{(a+c)^2 + (f-b)^2}$, and let $\varphi = 2(\theta - E)$.

It follows from Eq(3.26), and the definitions of E and φ, that

$$\frac{d\varphi}{dt} = a - c + Q\cos\varphi. \tag{3.27}$$

In all our real-life experimental results of Chapter 6, it was found that $\varphi \approx 90°$. Therefore, in the light of Eq(3.27), it is safe to say that in this application the direction of curvature of the limit cycle has the sign of $a - c$. Symbolically

$$\frac{d\varphi}{dt} \text{ has the sign of } a - c. \tag{3.28}$$

If $a > c$ the limit cycle revolves counter-clockwise and if $a < c$ it revolves clockwise. Between the two, obviously, there will be a change in the direction of curvature of the limit cycle.

It is important to appreciate that Richardson's [1960b] concern with the path of the limit cycle arose as part of a systematic and thorough analysis of his arms race equations. As far as is known, he was not specifically looking for sign changes in the direction of curvature of the limit cycle, nor identifying means of predicting points of instability in the way required in the thesis. Neither should it be assumed that, if a tends to c (i.e. if $\frac{d\varphi}{dt} \rightarrow 0$), the Richardson equations will become unstable at the very point where the direction of curvature of the limit cycle changes sign. Indeed, in the EOE reaction, instability occurs at some point *after* the Hopf bifurcation which is so critical. In chaos theory [Ott, 1994], similar sign changes occur at a Hopf bifurcation when, at the moment of transition from a laminar to a chaotic state, the complex-conjugate eigenvalues of the limit cycle change sign from minus to plus, see Section 5.5.1.

3.7 Predicting Instabilities in Richardson's Arms Race Model

Earlier, under certain assumptions, it was shown in Eqs(3.16) that a reasonable approximation to Richardson's coefficients a and c can be obtained (in the 2-nation case), *viz.*

$$a \approx k_{2,1}\, y_m \quad \text{and} \quad c \approx k_{1,2}\; x_m \, . \qquad\qquad .$$

We now ask whether changes of sign in the direction of curvature of the limit cycle can be used to *predict* the approach of instability. In view of Eq(3.28) and the approximate values of a and c, the answer is affirmative**;** predictions are studied in Section 5.5 and throughout Chapter 6.

In context, the significance of the difference $a - c$ lies not in its value but in the number of changes of its sign at each iteration step (see Section 5.5.1). Figure 5.6 shows the results of running the prediction GA, GA-6, by plotting the *number of negative signs in* $a - c$ (maximum 15) at increasing values of *average defexp%.* We found 'sharp drops' (heralding instabilities) occurring at 6.5%, 7.9% and at 10.1%.

Each instability is prefaced by a sudden plunge (a 'sharp drop') from a high number of minus signs to a small number of minus signs. Analysing this and other figures, we observed that if a overtakes c very rapidly (normally denoted by a change of ten or more signs from minus to plus at one iteration step) then instability is imminent. A lesser number of sign changes, say seven, does not have this effect.

The evidence presented in our application would suggest, therefore, that an unstable point in the Richardson equations is always prefaced by a change in the direction of curvature of the limit cycle and that this is manifested by a sudden and marked change in the sign of $a - c$, from minus to plus, across the population. At this point, the experimental evidence for saying this may be slender, but the theoretical argument is quite strong. See Chapter 6 which validates the model. We conclude that instabilities in the Richardson arms race equations *can* be predicted and, therefore, that the timing of the outbreak of open hostilities between nations indulging in an arms race can be forecast. All this assumes – and it is a big assumption – that the prediction GA is being fed with timely and accurate information from military and economic intelligence sources.

3.8 Conclusions

It is not always easy to apply Richardson's equations to real-world situations. To be of practical use the equations need to be replaced with (or transformed into) something more precise and expressed in terms of the information available. It is not sufficient to prove theoretically from Eqs(3.1) that instability occurs when $ac > bf$, and from Eqs(3.16) that the sign of $a - c$ indicates the direction of curvature of the limit cycle, which is crucial in predicting when instability is likely to occur. To justify our approach, we must be able to relate the Richardson coefficients a, b, c and f to the actual discretised evaluation equations using, for instance, Eqs(3.15). In both the 2-nation and 3-nation scenarios, we have estab-lished a reasonable approximation for a and c under certain assumptions.

CHAPTER 4

Genetic Algorithms and the Model used in the Thesis

4.1 Scope

This chapter covers everything which is common to the canonical GAs used in the thesis, and explains the mechanics of their operation in some detail. Choice of *evaluation equations* and *fitness functions* is particular to individual applications and will be covered in the relevant chapters. *Fractal basins* are explained in Section 4.10.

4.2 Early Influences

Over the years we had met a number of problems (particularly in numerical optimisation) which defied easy solution, largely because the search space was too big to be manageable. Thus GAs came as a breath of fresh air, opening new horizons. Our early steps in GAs were guided by Lawrence Davis's *Handbook of Genetic Algorithms* [Davis,1991] which takes its readers systematically through GA fundamentals. As a result, some fifteen trial GAs were written, each attempting to progress from its predecessor in difficulty and complexity. Nothing depended on these original 'learning-curve' GAs, so experiments could be tried and risks could be taken. Several valuable lessons were first learnt in this way, including the following, which are listed in no particular order:

(1) Full audit trails are essential in GAs. It must be possible to follow the progress of every chromosome at every stage and sub-stage at every generation. Even when not used to trace anomalies or 'bugs', audit trails can be very informative about convergence.

(2) Too much turbulence in a chromosome (e.g. from multiple crossovers) is never good.

(3) Seeding chromosomes with real-world data does not necessarily give good results.

(4) GAs must never be *forced*, but they can be *coaxed.*

(5) Canonical GAs do not like having their goalposts moved. It is unwise, for instance, to change fitness functions dynamically without good reason.

(6) Any GA trying to model a real-world scenario must be based on some live data from that scenario.

(7) Varying mutation and crossover dynamically may sound attractive in theory, but it does not always work. If sizeable numbers of generations (in our case, say, > 50) are

involved, mutation may get unreasonably large and of itself help to promote instability. Better results can often be obtained in this application by using fixed but convenient values such as 0.1% for mutation and 60% for crossover.

(8) The GA's *representation* (how the genotype maps onto the phenotype) is crucial. In the representation used here, genes may be real or integer and of varying length. Once concatenated, however, a chromosome must *look like* one big integer.

(9) GAs are stochastic processes; meaningful answers cannot be taken from a single run.

(10) The value of a practical GA is judged by its *robustness.* It is generally worth going to some trouble (e.g. by discretising the evaluation equations, or by delaying convergence) to ensure that results are not overly influenced by the starting condition. See (7) of Section 9.4.1.

(11) "It is always easier to add new stages to an already effective system than it is to modify earlier stages in the sequence" [Simon, 1969].

(12) It is unwise to initiate the contents of genes with zeros; this can cause *degenerative convergence,* i.e. the genes may tend to zero. See Section 2.13.

4.3 Mechanics

Most canonical GAs use only two operators, mutation and one-point crossover. Both operators, however, can be varied at runtime. Our GAs use two-dimensional matrices (typically four of 100×16 and three of 100×3). In the former, the chromosomes are physically embedded between identifying records and results, see Table 4.1. Since the answers (elements of the phenotype) in the case of arms races happen to lie typically in the thousands, a decision was made to have the genes of the genotype map *directly* onto those of the phenotype, the only exception being the rate-factor genes which sometimes need to be divided (or multiplied) by factors of ten to make them useful in the evaluation equations. There is, therefore, little embryogeny (see Glossary). Parent selection is by the standard 'roulette wheel' method (see Section 2.7) using cost-weighted random pairing. This is effected by bubble-sorting Matrix 2 to yield Matrix 1 which shows fitness in a reducing sequence. Once this sequence is established, however, the fitnesses are replaced by a linear norm (in fact, starting with 500 for the fittest and decreasing in steps of 5) – see the last column at the top of Figure 4A.1 – and selection is then based on these normalised fitnesses (called *normfits*). The aim of introducing the linear norm is to prevent superfit chromosomes getting too big a reproductive advantage.

To determine the size of population to be used in our GAs, we tried experiments with populations of 20, 35, 50, 75, 100, 250, 500, 750 and 1000, looking for 'straight line' convergence such as that typically illustrated in Figure 4.3. A population of 50 was just about acceptable, for anything less would not converge satisfactorily. In the best engineering tradition we, therefore, chose twice the acceptable minimum and thenceforth stayed with a population of 100 throughout. This proved entirely satisfactory for our application. This empiricist approach should be contrasted with the rationalist approach [Cantú-Paz, 2000], where the determination of an appropriate population size takes several pages of reasoned argument about schemas and building blocks.

The initialising population is chosen by using the inbuilt RNG to generate a *likely* value for each gene. Here we use domain knowledge to constrain the upper bound of RNG. Some genes (e.g. certain *rate-factors,* $k_{i,j}$, or *intrinsic defence expenditures,* x_s) may use a fixed upper bound (e.g. < $k_{1,2}$ = `random(29)`>) while others (e.g. *budgeted defence expenditure,* x_m) may be variable (e.g.< x_m = `random(defexp)`>). All the genes are initialised in this way, and this is what we mean by a *random start*. When it is desired to use real data (e.g. IMF values for the rate-factors, or actual defence expenditures) the RNG-derived numbers are *overwritten*. This overwriting is deliberate. For our results to be reproducible (and to be comparable using different input settings) the genes must be initialised using the same RNG sequence throughout.

The initial population is set in parallel into each of two (100×16) matrices, Matrices 1 and 2, and these are used throughout. See Table 4.1 below which shows a typical occupancy of the columns of one of our GAs. The chromosomes are shaded. Each chromosome is processed using the two evaluation equations (such as Eqs(5.7)) to give columns 12 and 13, and the fitness is calculated and put in column 14. The contents of Matrix 2 remain in the sequence they were generated, whereas the *entire* contents of Matrix 1 are bubble-sorted [Kantaris, 1994] into a decreasing order of fitness. Although not strictly required, there are two reasons for sorting; firstly, presentation, i.e. printed results are always extracted from Matrix 1, giving the best result last; secondly, to assign the linear norm.

Cost-weighted roulette wheel selection (see Section 2.7) is applied to Matrix 2; to apply it to Matrix 1 would entail selecting parents from a list which had already been sorted, and this makes little sense.

Table 4.1 The columns of Matrices M1, M2, and M3. The evaluation equations have been discretised, so that x_t and y_t are an integral part of the chromosome (and therefore have to be given appropriate starter-values for x_0 and y_0)

Col	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
M1	gen	srl	$k_{1,1}$	$k_{2,2}$	$k_{1,2}$	$k_{2,1}$	x_s	y_s	x_m	y_m	x_t	y_t	x_{t+1}	y_{t+1}	*fitness*	*linear norm*
M2	gen	srl	$k_{1,1}$	$k_{2,2}$	$k_{1,2}$	$k_{2,1}$	x_s	y_s	x_m	y_m	x_t	y_t	x_{t+1}	y_{t+1}	*normfit*	*running total*
M3	gen	srl	$k_{1,1}$	$k_{2,2}$	$k_{1,2}$	$k_{2,1}$	x_s	y_s	x_m	y_m	x_t	y_t	x_{t+1}	y_{t+1}	*normfit*	*random number*

Once the fitnesses are calculated and normalised, a running total is kept and stored in column 15 in Matrix 2. Let the final total be called $w1$. (In the example in Figure 4A.1, $w1$'s value is 4775.) A uniformly distributed random number $w2$ (ranging over $[0, w1]$) is then generated in column 15 of Matrix 3 (by calling `<random(w1)>`). At this point, apart from the list of random numbers in column 15, Matrix 3 is otherwise empty. The running total list in Matrix 2 is then compared in sequence with the random number list in Matrix 3. Looking at the last column (15) in each case, the first row where

$$\textit{Matrix } 2\textit{'s running total} \ \geq\ \textit{Matrix } 3\textit{'s random number } w2 \qquad .$$

becomes a parent for the next generation and the whole of the row from Matrix 2 is copied into Matrix 3. Hence Matrix 3 becomes the 'mating pool'. For instance, in Figure 4A.2, the first random number is 4497. The first 'running' number to exceed it in Figure 4A.1 is 4775 at row serial 109, so the whole of row 109 is copied into Matrix 3, to become the first parent. The next random number is 2159, and the first 'running' number which exceeds it is 2395 in row serial 104; hence the whole of row 104 is transferred into Matrix 3 as the second parent. The process is then repeated until Matrix 3 is full. It should be observed, however that, firstly, not all the rows of Matrix 2 have been transferred to Matrix 3; some (e.g. 100, 101 and 103) have been discarded. Secondly, some rows (e.g.104, 105, and 107) have been transferred more than once. This is *roulette wheel selection* (Section 2.9) and, while the fitter chromosomes are being given a greater chance of becoming parents, none are entirely excluded. This arrangement works well. To be precise, the expected number of offspring is the fitness of the chromosome under consideration divided by the average of the fitnesses of the entire Matrix 2 population.

Pairs of adjacent rows of the chromosomes in Matrix 3 are now 'unbundled' and set into two columns in (the 100×3) Matrix 4. In Figure 4A.3 we show some of the results of unbundling the ninth and tenth rows of Matrix 3 (Serials 107 and 104). Taking pairs of adjacent rows of chromosomes and 'unbundling' them into two columns was deliberate, so that *mutation and crossover could take place uninhibited by gene boundaries.* Every element in the two unbundled chromosomes is scanned by an RNG which decides whether or not that particular element is to be mutated. This is *mutation probability per locus*. Some elements will be mutated, the number depending on the mutation rate. The resulting unbundled chromosomes (containing both mutated and unmutated elements) are transferred into Matrix 4a, see Figure 4A.4. Matrix 4a is then transferred to (the 100×3) Matrix 4b where a single-point crossover ('recombination') takes place. From the crossover point downwards, every element will change columns. In Figure 4A.5 the crossover point is level 19, i.e. 55% of the total chromosome length of 36 elements. This is what is meant by a crossover rate of 0.55. It makes no difference whether mutation precedes or follows crossover [Mitchell, 1996].

The two columns of (the 100×3) Matrix 4b are now *'rebundled'* into rows in (the 100 ×16) Matrix 5, see Figure 4A.2. These rows contain chromosomes and identifying data (generation and serial numbers for diagnostic purposes) only, and form the embryo of the next generation. The generation loop now returns to its start-point, and in Matrices 1 and 2 new serial numbers are assigned, the chromosomes are transferred over into Matrices 1 and 2 from Matrix 5, and x_{t+1} and y_{t+1} are moved over into Matrices 1 and 2 from Matrix 3 to become the new generation's x_t and y_t , respectively. The new chromosomes are then evaluated and their fitness is calculated; as before, Matrix 1 is bubble-sorted (see Figure 4A.6), and the whole process restarts for as many generations as are required. Matrix 5 is illustrated in Figure 4A.2, as is the next stage, the conversion from Matrix 5 of *Generation* 0 to Matrices 1 and 2 of *Generation* 1, in Figure 4A.6.

In the early stages of the study we did try other combinations such as tournament selection (see Section 2.7) and multiple crossover (see Section 2.8.2), *inter alia*. Both of these were rejected in the present work because they behaved too randomly.

There is one other matrix. Part of the best chromosome at each generation (the fitness with its corresponding values of x_t and y_t) is transferred from Matrix 1 into Matrix *result* which is subsequently used both for Option 7 (see Section 4.8), and to plot the graphs at Options 8 and 9. Matrix *result* is not a fundamental part of the GA; it is a 'collecting ring' for administrative convenience only.

4.4 Working Practices

All GAs were used to generate a variety of graphical and textual data including individual chromosomes and summary tables, see Section 4.8. Matrix 5 of the prediction GAs (see

Section 5.5.1) was modified to calculate the direction of curvature of the limit cycle which is crucial in predicting instabilities (see Section 5.5.1).

Whenever a GA was modified, it was our practice initially to reduce the population to ten and run Generations 0 and 1 only. The evaluation of every element of every gene could then be followed stage by stage.

Figures 4A.1 to 4A.6 in Appendix 4A show how GA-1 progresses from Generation 0 to Generation 1. Solely for illustration, it has a population of only 10, an (excessive) mutation rate of 20%, a crossover rate of 55%, and uses rate-factors based on information from IMF statistics. The seed is 8, and the average percentage defence expenditure is 6.5%.

Figures 4B.1 to 4B.3 in Appendix 4B to this chapter show how three algorithms, GA-0 with ten genes, GA-1 with fifteen genes, and GA-9 with twenty genes, progress across forty generations, from Generation 0 to Generation 39. These GAs have a population of 100 and a seed of 9.

It should be remembered that GAs are stochastic processes and that meaningful results, therefore, cannot be taken from a single run but have to be averaged over a series of runs. Runs, charts and other illustrations given in this chapter and Chapter 5 are for demonstration only [1].

4.5 Block Structures

The block structure of the GAs used in the thesis is as follows:

```
begin
   $t = 0$
   Initialise a population $P(t)$ (typical size 100 chromosomes) in Matrices 1 and 2, and
   assign initial values to $x_t$ and $y_t$
  begin
           Using a set of evaluation equations (e.g. Eqs(5.7)) compute $x_{t+1}$ and $y_{t+1}$ for
                   each row of Matrices 1 and 2, and assign a fitness to each one of them
           Sort Matrix 1 into ascending/descending order of fitness
      if (termination condition) end
           Apply a linear norm to the fitness column of Matrix 1
```

---

[1] Having said that, the output of a really robust GA will (by definition) be more or less the same whatever the starting conditions

```
        Using the roulette wheel method on Matrix 2, select a new population $P(t+1)$ from $P(t)$ and put it into Matrix 3
          Unbundle adjacent pairs of the rows (chromosomes) from Matrix 3, and put them into the columns of Matrix 4
          Mutate the $P(t+1)$ chromosomes of Matrix 4, two at a time, and put them in Matrix 4a
          Crossover  (Recombine) the $P(t+1)$ chromosomes of Matrix 4a and put them in Matrix  4b
          Rebundle the columns of Matrix 4b into the rows of Matrix 5.  Where required, calculate $a - c$, and count the number of minus signs
          Transfer the $P(t+1)$ chromosomes from Matrix 5  into Matrices 1 and 2,  respectively
          Transfer the $P(t+1)$ values of $x_{t+1}$ and $y_{t+1}$ from Matrix 3 into Matrices 1 and 2 overwriting $x_t$ and $y_t$, respectively
  end
          $t = t + 1$
end
```

Two points are now made:

- The above fits neatly into three modules; population, evaluation/fitness, and reproduction, the latter including mutation and recombination (crossover) sub-modules.
- The GA makes use of seven two-dimensional matrices, normally four of 100×16 and three of 100×3, for the applications used in the thesis These matrices help to maintain a strict audit trail (see (1) of Section 4.2) by transferring the datasets from one matrix to the next as they are selected for parenthood, mutated, and recombined.

4.6 The Influence of Real-World Data

Chapters 5, 6 and 7 of the thesis each start with a very small amount of data gleaned from either intelligence sources or figures published by the IMF (see Appendices 6B.1, 6B.2 and 6B.3). The principal use of this data is to establish a model (in fact, a modelling GA) with some basis in fact, so that curves can be computed which are "*... a kinetic model faithful to the input data and the physical principles involved*" [Cao *et al*., 2000] and so, more or less, mimic the behaviour of the data presented in our various data sources. Once this has been achieved (and it is no trivial task, since it involves minimising the differences between the

GA-calculated values and the real-world values of x_t and y_t at each generation), it can be assumed that:

- the evaluation equations reflect the problem and exploit the available data;
- the rate-factors, $k_{i,j}$, have values which are credible and safe to use in the prediction GA (even though they may later be refined quite drastically by the action of that GA).

To predict instabilities between nations we now transfer the rate-factors from the modelling GA into the prediction GA and call Option 0 (see Section 4.8). This lists the values and *signs* of $a - c$ at a specified average defexp% (see Figure 4A.7). The data generated by all outputs of Option 0 over the required range of average defexp%'s is the basis for the prediction plot (see Figure 5.6). Sharp drops in the prediction plot can now be correlated with real-world data about actual conflicts (see Chapter 5).

4.7 The Design of Individual GAs

Making every gene an integer looked neat on the monitor screen but rendered the arithmetic of the evaluation equations difficult. Having all genes as real numbers [Michalewicz, 1992] seemed attractive but introduced problems once they were concatenated, for decimal points and fractional numbers do not concatenate well and gave rise to a series of domain fault and floating-point errors during execution runs. The proposed solution is a compromise; declare the genes as floating-point numbers, but ensure that anything put into a chromosome in any matrix is rounded so as at least *to look like* an integer! It later became apparent that this was no new discovery, for Lucasius and Kateman [1992] had said that GA chromosomes for subset selection and combinatorial optimisation problems are best suited by *integer representation*. The other lesson learnt was that (except under controlled conditions) zeros are *never* acceptable as initialising genes, whether they are deliberate (e.g. setting x_0, y_0, or z_0 to 0.00) or accidental (e.g. on initialisation, if and when the RNG hits on a zero). To avoid the problem, in the first case, x_0, y_0, and z_0, etc. are deliberately set to some arbitrary figure ($\neq 0$) while, in the second case, the RNG was made to take a second bite.

It was originally thought that using the four or six rate-factors derived from IMF data would give better and more stable results than allowing the GA to generate and refine them. It turned out that this was not always the case.

The source code expression <`seed(n)`> means that the first *n* numbers generated by the RNG are discarded before it starts to initialise the genes, thereby achieving a different start-point. For example, <`seed(9)`> means that the first nine outputs of, say <`random(42)`>, are discarded and the tenth output (whatever that may be) is the first one used (in fact, to initialise $k_{1,1}$). As already stated, when the start-point makes little or no difference to the results – a highly desirable situation – the algorithm is termed *robust.*

4.8 Deliverables: Textual and Graphical Outputs

Most of our GAs yield at least five 'deliverable' outputs. These are:

- [*Option* 0] (Prediction GAs only). A textual display of fifteen rows of Matrix 5, showing the values *and signs* of a - c for a specified average defexp%. The output of Option 0 is the input data for prediction plots.
- [*Option* 6] A textual display of the ten fittest chromosomes (out of the population of 100) of any two (specified) generations of Matrix 1. These were originally intended for diagnostic purposes but served greatly to enhance confidence in the system. Once stability had been achieved, the values of the genes could be read.
- [*Option* 7] A textual display of the fittest values of x_t, y_t, z_t and fitnesses. One entry (out of the population of 100) for each generation. This data is input for Options 8 / 9.
- [*Option* 8] Using the data from Option 7, a graphical display of the fittest values of x_t, y_t, z_t (and, optionally, fitness) over the generations, with the 'curves' [2] in different colours. *Each Option 8 graph uses only one defexp% value.* This means that Option 8 gives an instant view of the 'worth' of any solution. It is, however, essentially an artifice, *cf.* the concept of schemas or building blocks that is useful despite the fact that GAs do not actually manipulate them at all (see Section 2.4). Similarly, Option 8 graphs are plotted using the *same* defexp% value over the full range of generations, a situation that is most unlikely to happen. Nevertheless, Option 8 graphs do indicate at the specified defexp% whether or not the system is ever likely to become unstable, whether the evaluation equations (such as Eqs(5.7)) will converge asymptotically and, if so, to what and when. This is useful information.
- [*Option* 9]. A *dominance diagram* (see Section 4.9 below) coupled with an Option 8 display graphically portraying the extreme right-hand (i.e. the latest) column of the diagram. These are illustrated for a three-nation conflict in Figure 4.1. In the dominance diagram, the values of x_t, y_t and z_t are plotted against successive generations. (Note how early convergence is achieved).

---

[2] 'Curves' is a misnomer, for they were drawn as straight lines between points, and no attempt was made to 'smoothe' them in the curve-fitting sense

Parameters alterable at runtime were:

- <`seed`>, the 'start-point' from which the RNG initialises the GA (see Section 4.7).
- <`mutation rate`>, which could be increased dynamically. It usually starts at 0.1%.
- (Single-point) <`crossover rate`> which could be reduced dynamically. Typically 60% of the total length (number of elements) of the chromosome (see Section 4.3).
- <`type`> The output required (e.g. *Option* 6) including diagnostic tables.

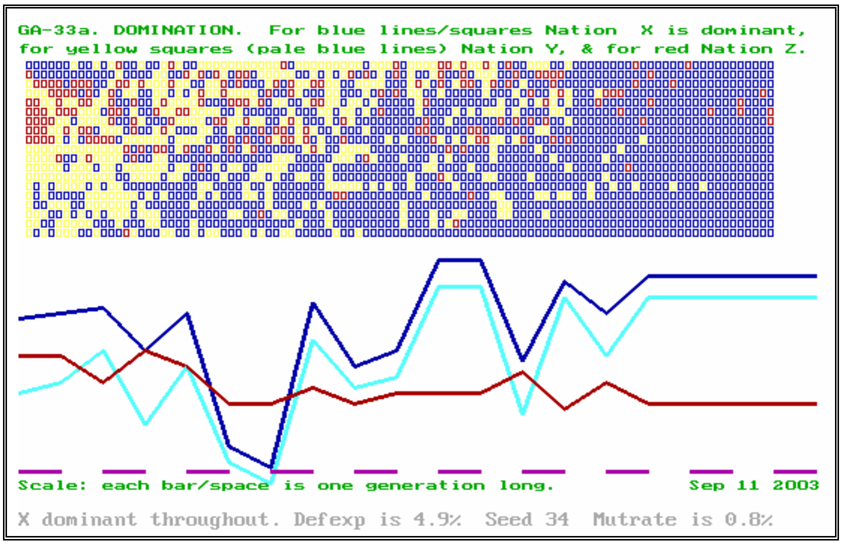


Figure 4.1 GA-12. Values of the defence expenditures x_t, y_t, z_t against generations, together with a 'dominance diagram'. Convergence indicates the parameters necessary for a stable outcome. (For pictorial clarity, this and all other similar figures draw straight lines between the points; in fact, the figures should properly be histograms for the parameters do not change within each generation.)

- <`dominance`> (Whether Nation X was to remain dominant, or domination was to vary dynamically.) Richardson and Mayer-Kress had both assumed (see, for instance, Eqs(3.11) and (3.13)) that Nation X was always dominant. This is clearly not reflected in real life. All our GAs, therefore, had an option allowing domination to change dynamically. In fact, for the three-nation system, after each computation of the evaluation equations, the values of x_t, y_t and z_t were examined to see which was the largest. The evaluation equations were then re-coded to enable one of them to portray the dominant nation. Once domination changes, the re-coded evaluation equations stay as they are until that domination changes again; this could take several generations.

Other alterable parameters at runtime were:

- Size of population (normally set to 100).
- Number of generations required (normally set to 20 or 40, sometimes to 100).
- Whether the rate-factors $k_{i,j}$ were to be pre-set, or to be developed by the GA.
- The linear norm and the decrement (usually set to 500 and 5, respectively).

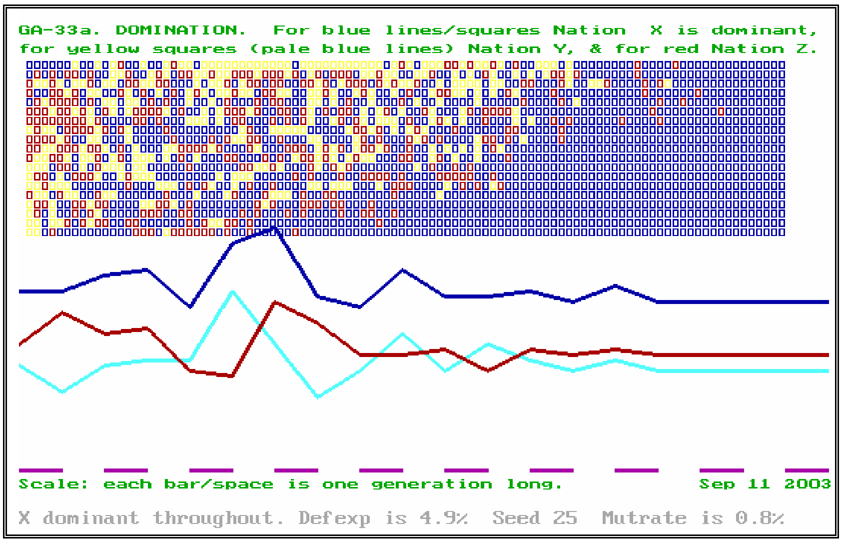


Figure 4.2 GA-12. Seed 25, with Nation X dominant throughout. IMF rate-factors

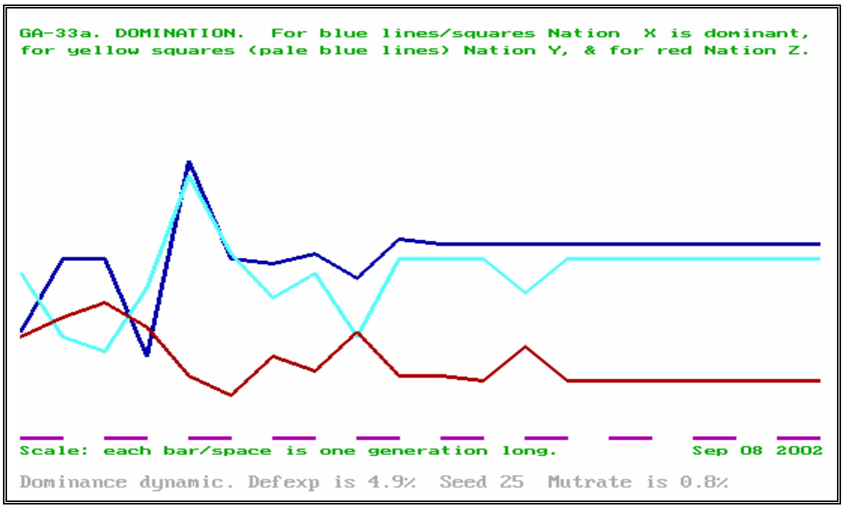


Figure 4.3 GA-12. Seed 25. Dom dynamic. The GA sets its own rate-factors. Faster convergence

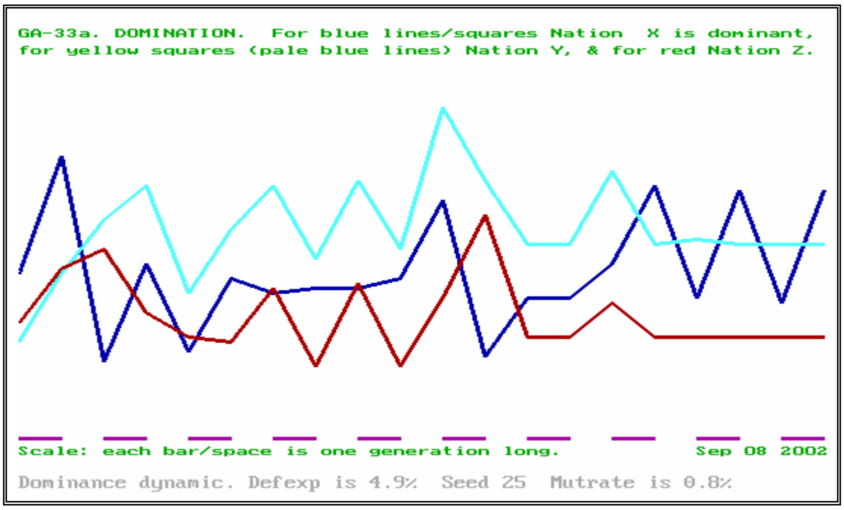


Figure 4.4 GA-12. Seed 25. Dom dynamic. IMF rate-factors. Unstable, chaotic

The way in which some graphs (such as Figure 4.4) had ‘settled’, zigzagging between two levels (‘attractors’), led to the suspicion that the system had sometimes become chaotic. (Figure 4.4 is in marked contrast to the work on nuclear deterrence (Chapter 7) which had resulted in exceptionally stable systems.) It would therefore be rewarding to see if there were any fractal basin boundaries.

4.9 Dominance Diagrams

The idea was to introduce into each GA (typically with a population of 100 for 20 generations) an option which would plot for *every one* of the 2000 sets of equations which one of Nations X, Y or Z was dominant. It should therefore be easy to see where domination changed, for *in these changes lie the seeds of chaos and instability*. The on-screen representation consists of a 20×100 matrix of contiguous small squares, drawn in different colours. Typically, X-dominant is indicated by blue, Y-dominant by yellow and Z-dominant by red. The curves below the coloured matrix of the dominance diagram portray the situation at the matrix's extreme right-hand column. The matrix is called a '*dominance diagram*' and Figures 4.1 and 4.2 illustrate the concept.

Running tests using different seeds and different values for defexp%, Richardson's original equations and a set of IMF rate-factors resulted in a number of curves which converged, many which did not converge, and a number of zigzags (e.g. Figure 4.4). We found that, whenever the value of defexp% was less than 4.9%, all curves converged, meaning that below this level the exponential nature of the defence expenditure graphs all but disappeared,. and there is no danger of an arms race. This is not very useful in practice, as we do not choose (or have any say in) arms race statistics. However, Option 8 will indicate when and where x_t , y_t and z_t are almost parallel, and if they are, then a stable situation exists. It is then possible to read off (from the tables at Option 6) the values of all the parameters (genes). Assuming that they are not trivial, these indicate one set of conditions for stability.

The dominance diagrams of Figures 4.1 and 4.2 illustrate that there are few smooth transitions; the pictures are diffuse to the point where it seems that the system could be *noisy* rather than chaotic. Nevertheless, the zigzags are so marked that the possibility of chaos must not be discounted.

Figure 4.2 used GA-12, a seed of 25 with Nation X dominant throughout, and IMF rate-factors. Figure 4.3 shows what happens if GA-12 has the same seed, but has dynamic domination and is allowed to choose its own rate-factors randomly, subject only to upper bounds on the RNG. Convergence is much quicker. This reinforces the view (see (4) of Section 4.2) that GAs do *not* like being force-fed [Davis, 1991; Haataja 1999; Bentley, 1999] and behave best when nature is allowed to take its course. Figure 4.4 again shows

GA-12 with seed 25, dynamic domination and IMF rate-factors, but its appearance is very different to what has gone before.

4.10 Fractal Basins

It is evident from the shape of the curves in Figure 4.4 that in Richardson's three-nation arms race, we may have a potentially chaotic system [Ott, 1994; Cohen and Stewart, 1994]. We decided, therefore, to take a radically new approach and develop algorithms to generate *fractal basins* with a view to seeing whether they contained any boundaries which might indicate the presence of one or more chaotic systems. We did this by reducing a whole dominance diagram (such as the 20×100 matrix in Figure 4.1) to a single small square which represented one particular set of circumstances, and then printing hundreds of these small squares side by side, each with slightly different circumstances. The resulting small squares were coloured as before (X-dominant is indicated by blue, Y-dominant by yellow, and Z-dominant by red).

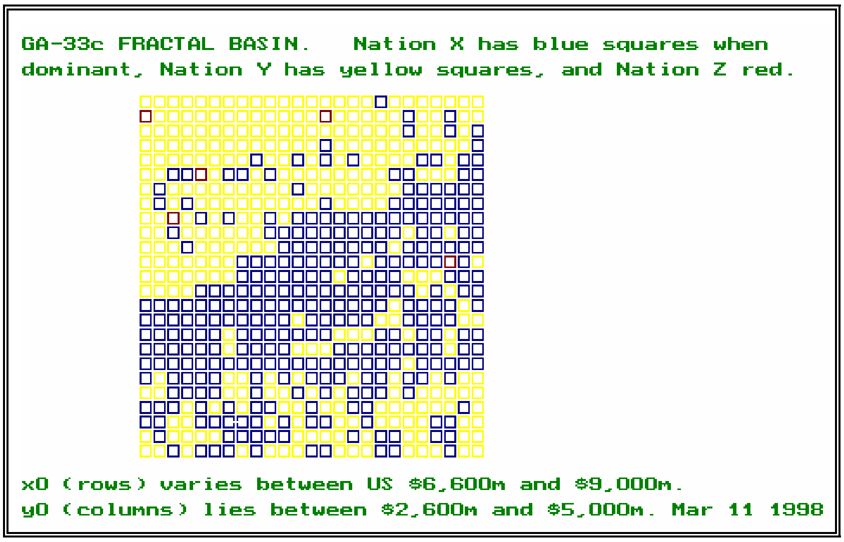
The dominance diagram was reduced to a single small square in a 25×25 matrix by using the technique of *bottom line consensus* (see Glossary).

One requirement was to see how stability varied with initial conditions (i.e. x_0 and y_0). Another was to maximise the number of parameters which could be played off against one another, e.g. each of the rate-factors of Nation X against those of Nation Y, Nation X's various rate-factors against Nation Y's defexp% , or any other pair. In short, we were looking to locate boundaries where dominance changes (i.e. the likely points of instability) and, having done so, to identify the parameters (or conditions) which created these boundaries. To this end, a new GA was developed, GA-10, which puts the generation for-loop of GA-12 *inside* two other for-loops. One loop varies x_0 from \$6.6Bn to \$9.0Bn in 25 steps, and the other varies y_0 from \$2.6Bn to \$5.0Bn. Hence the fractal basin, or plot, is a 25 × 25 square (see Figure 4.5). The third defexp, z_0 , on the other hand, was kept constant at \$3.9Bn.[3]

For each pair of values of x_0 and y_0 there corresponds a dominance diagram, so each fractal basin contains the makings of 625 dominance diagrams. An output of GA-10 is shown in the fractal basin in Figure 4.5. This is for a seed of 25 with dynamic dominance and indicates from the blocks of colour and the presence of 'walls' that there exists a potentially chaotic situation [Cohen and Stewart, 1994].

---

[3] These values for *x, y* and *z* are approximations derived from the 1993 IMF figures for defexp given in Appendix 6A.3



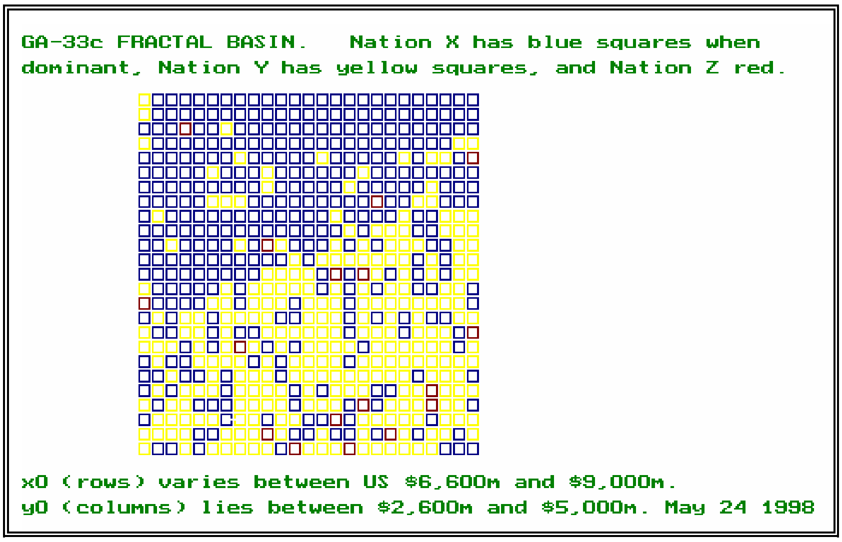
```
GA-33c FRACTAL BASIN.   Nation X has blue squares when
dominant, Nation Y has yellow squares, and Nation Z red.

x0 (rows) varies between US $6,600m and $9,000m.
y0 (columns) lies between $2,600m and $5,000m. Mar 11 1998
```

Figure 4.5 GA-10. Seed 25 (cf. Figure 4.4). IMF rate-factors. Note the ‘walls’

GA-10 yields only one deliverable — *the fractal basin*. Generating fractal basins is laborious as it involves a lot of computation. A fractal basin of 25 × 25, for instance, necessitates a population of 100 being processed (i.e. generated, evaluated, fitness-tested, roulette wheel-selected for parenthood, mutated and re-combined) and run for 20 or 40 generations 625 times! The advantage of the diagrammatic approach was that it became possible to pinpoint interesting areas worthy of further study. There is little point in investigating conditions in a ‘sea’ of red or blue or yellow as it is evident that the Richardson equations are stable in any such area. In contrast, any point in the stability diagram where the colour changes indicates a potential instability.

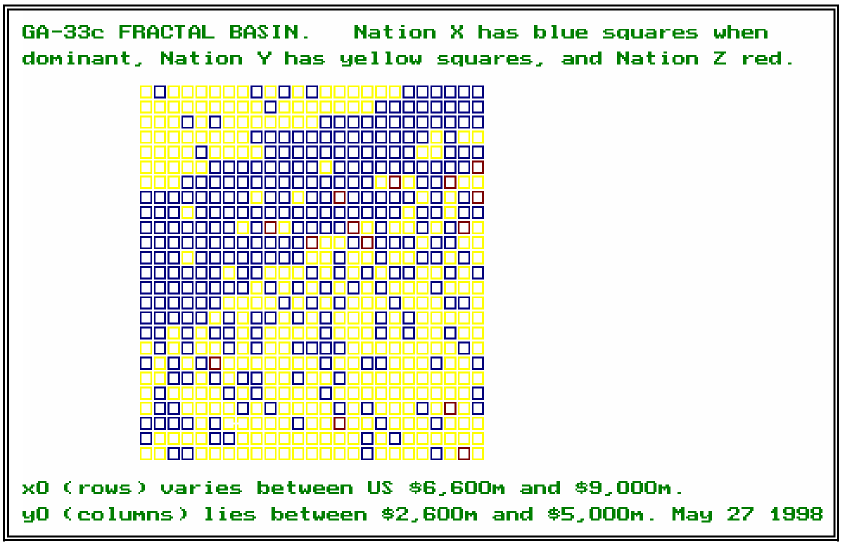
To facilitate comparison, fractal basins illustrated in this chapter all have a 4.9% defexp%.

Figure 4.6 is identical with Figure 4.5, except that the GA sets its own rate-factors.
Figure 4.7 is similar to Figure 4.6 except that $k_{2,3}$ is set to a specific value (8).



```
GA-33c FRACTAL BASIN.   Nation X has blue squares when
dominant, Nation Y has yellow squares, and Nation Z red.

x0 (rows) varies between US $6,600m and $9,000m.
y0 (columns) lies between $2,600m and $5,000m. May 24 1998
```

Figure 4.6 GA-10. Seed 25. The GA sets its own rate-factors. No noise margin



GA-33c FRACTAL BASIN.   Nation X has blue squares when dominant, Nation Y has yellow squares, and Nation Z red.

x0 (rows) varies between US $6,600m and $9,000m.

y0 (columns) lies between $2,600m and $5,000m. May 27 1998

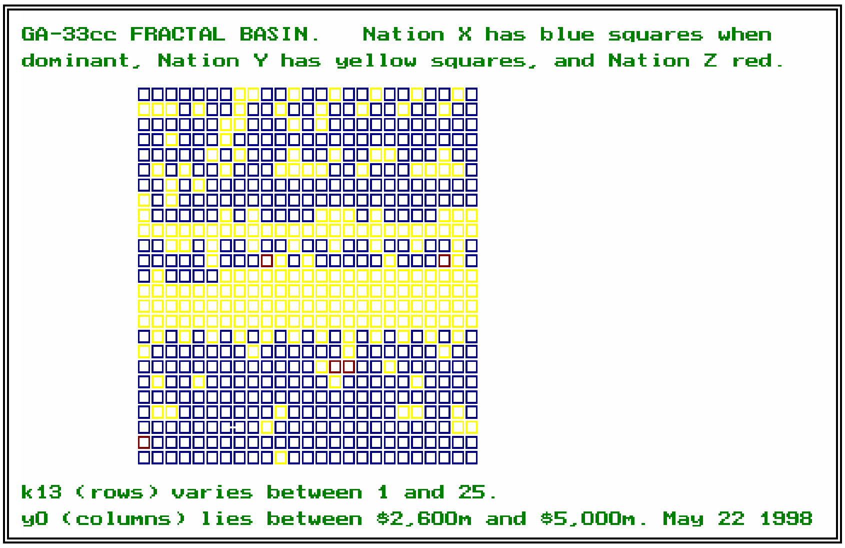
Figure 4.7 GA-10. Seed 25. The GA sets its own rate-factors, except for $k_{2,3} = 8$

It could be argued that these fractal basins show evidence of *noise* as well as of *chaos*. However, according to Chambers [1995], the introduction of noise widens the catchment area of the search space. Indeed, he claims, a noisy environment usually increases the probability of obtaining a better solution, or at least one with higher resolution. Even if this is not entirely true, the presence of noise in a GA has been found by Aizawa and Wah [1993] to alleviate the tendency for premature convergence (see Section 2.13). Later in the thesis, noise is introduced deliberately to see what effect it would have on our prediction work (see Section 6.7).

4.11 Balance versus Stability

In an arms race both sides try to maintain a balance of power, so that the ratio between their levels of armaments remains more or less a constant, an equilibrium. However, both levels can increase without constraint. Systems which are stable for one kind of displacement and unstable for another are not unknown in conventional mechanics, e.g. a ball on a saddle-shaped surface could oscillate – but it is more likely to fall off altogether. Where stability and instability co-exist, the unstable state is likely to be dominant. Hence nations are said to be unstable if there is a drift towards war. Equilibrium does not necessarily lead to long-term stability, for it could exist in an isolated small pocket in one of the fractal basins, and be ephemeral.

GA-10 proved to be very adaptable in that it only took a few moments to write another variant thereof, GA-11, which differs from GA-10 only in that the iterating variables are now $k_{1,3}$ (a representative set of figures from 1 to 25) and y_0 (from $2.6Bn to $5.0Bn, as in GA-10). The value of rate-factor $k_{1,2}$ is fixed (0.14 and 0.16 in Figures 4.8 and 4.9, respectively). The idea of concentrating in these fractal basins on factors which influence threat was suggested by Forrest and Mayer-Kress [1991]. One such factor would be the various defexp%. Another would be the initial conditions x_0 and y_0. Yet another would be the rate-factors $k_{1,2}$, $k_{2,3}$ and $k_{1,3}$. ($k_{1,2}$ is associated with $(x_t - y_t)$, $k_{2,3}$ is associated with $(y_t + z_t)$, and $k_{1,3}$ is associated with $(x_t - z_t)$, see the notes to Eqs(3.13).) We now concentrate on these three parameters.

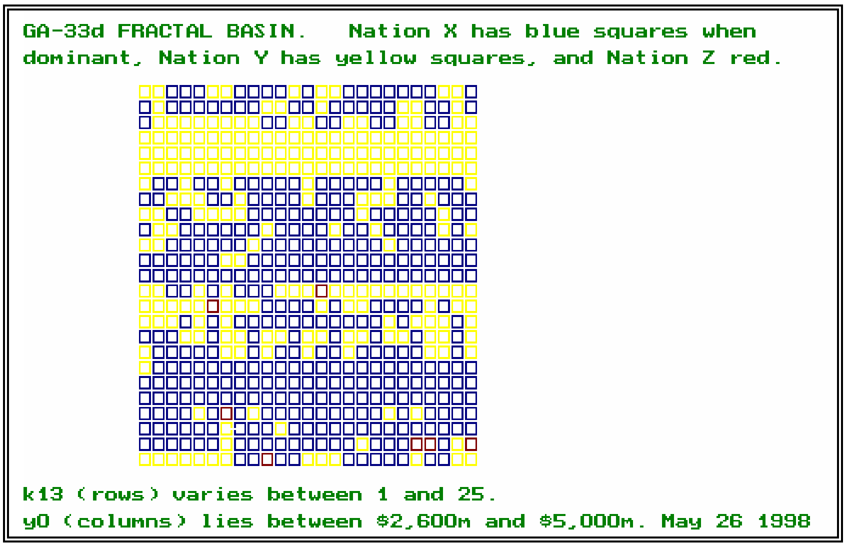


GA-33cc FRACTAL BASIN. Nation X has blue squares when dominant, Nation Y has yellow squares, and Nation Z red.

k13 (rows) varies between 1 and 25.
y0 (columns) lies between $2,600m and $5,000m. May 22 1998

Figure 4.8 GA-11. Seed 25. IMF rate-factors with $k_{1,2} = 0.14$. No noise

Figure 4.9 below gives a totally different picture from that of Figure 4.8, yet the difference between the two lies only in the fixed values of $k_{1,2}$ which is tiny. This means that the value of $k_{1,2}$ is critical to stability. This should not surprise us, for it was shown in Eqs(3.16) that the product $k_{1,2}x_m$ was approximated as c, as in $a - c$, the predictor of instability.



GA-33d FRACTAL BASIN. Nation X has blue squares when dominant, Nation Y has yellow squares, and Nation Z red.

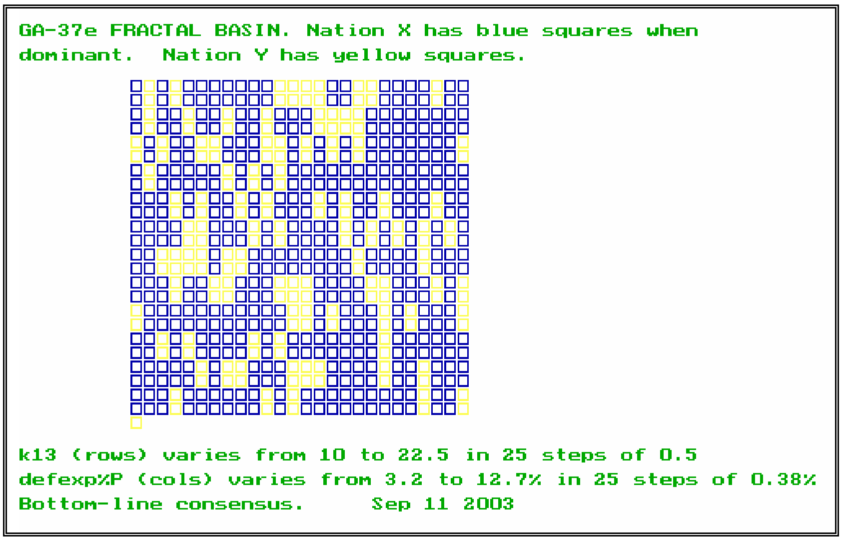
k13 (rows) varies between 1 and 25.
y0 (columns) lies between $2,600m and $5,000m. May 26 1998

Figure 4.9 GA-11. Seed 25. IMF rate-factors with $k_{1,2} = 0.16$. No noise

It can be concluded, from the five fractal basins shown above, that walls do exist. These walls and the zigzag curves of Figure 4.4 are both classic indicators of chaotic conditions

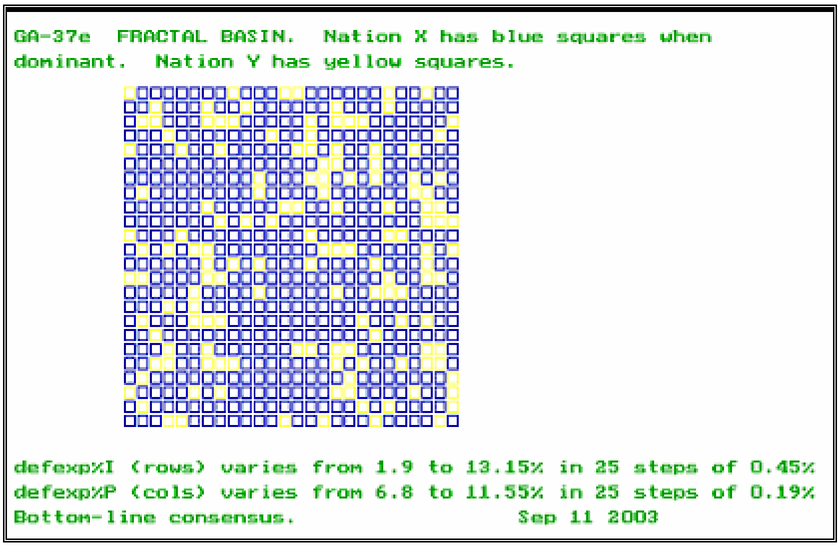
[Ott, 1994] and lead us to believe that there *are* regions of potential chaos in Richardson's three-nation arms race equations.

We next look at two fractal basins for the India/Pakistan conflict, both using GA-7, which are pictured in Figures 4.10 and 4.11. In Figure 4.10 the rows, $k_{1,3}$, vary from 10 to 22.5 in 25 steps. The columns (Pakistan's defexp%) vary from 3.2% to 12.7%. The other rate-factors are generated by <`random(.)`>. One might expect a wall since the seed chosen, 9, is known to give rise to instability at average defexp% of 6.6%, see Figure 5.10. Walls never did occur despite running GA-10 many times varying the pair of parameters in its two outer loops (including all the rate-factors).



```
GA-37e FRACTAL BASIN. Nation X has blue squares when
dominant.  Nation Y has yellow squares.

k13 (rows) varies from 10 to 22.5 in 25 steps of 0.5
defexp%P (cols) varies from 3.2 to 12.7% in 25 steps of 0.38%
Bottom-line consensus.      Sep 11 2003
```

Figure 4.10 GA-7. $k_{1,3}$ against Pakistan's defexp%. Seed 9



```
GA-37e  FRACTAL BASIN.  Nation X has blue squares when
dominant.  Nation Y has yellow squares.

defexp%I (rows) varies from 1.9 to 13.15% in 25 steps of 0.45%
defexp%P (cols) varies from 6.8 to 11.55% in 25 steps of 0.19%
Bottom-line consensus.                   Sep 11 2003
```

Figure 4.11 GA-7. India's defexp% against Pakistan's defexp%. Seed 9

In Figure 4.11 we plot India's defexp% against Pakistan's defexp%. Again, there are no walls. The patchiness was first thought to be due to noise, but the reason is more subtle.

Consider a system of N first-order ODEs (*cf.* Eqs(3.1)) such as

$$\frac{dx_i}{dt} = F_i\left(x_1, x_2, \ldots x_N\right), \quad i = 1, 2, \ldots N \ .$$

Ott [1994] has shown that, in such systems, chaos is only possible when $N \geq 3$. Hence it is reasonable to expect (and we are visibly getting) chaos in a 3-nation conflict, but it can be ruled out in 2-nation conflicts. Nevertheless, *any* system of first-order ODEs contains the seeds of chaos, so it could be argued that we *are* justified in using the language of chaos (e.g. limit cycles) when operating with Richardson's arms race theory, even in a two-nation case. What this means is that there are no general regions of stability or instability in the India/Pakistan conflict. This conflict lurches from one instability to another with periods of respite in between. One cannot, therefore, say with any certainty that there are any set-piece conditions for averting war, except perhaps by reducing the average defexp% to some very low and thus unlikely figure. (The India/Pakistan arms race is covered in Chapter 5.)

4.12 Conclusions

After an extended period of trial and error seeking a combination of stability, fast convergence and robustness – aims which are not always mutually compatible – we finally decided to use a comparatively simple canonical GA with fixed-length integer strings. The application calls for the data to be treated gently, so we chose *weighted random pairing by cost* rather than the more dramatic and rough *tournament selection.* Similarly we avoided multiple crossovers which tend to be destructive of building blocks. Mutation was always sufficient to open up the search space while stopping short of creating instability. Typical answers lie in the hundreds and thousands, figures which could be mapped directly from the genotype onto the phenotype, so no embryogeny was necessary. We do not regard gene boundaries as sacrosanct, so both mutation and crossover could take place anywhere, irrespective of such boundaries. Where possible (and in the three-nation case it is not always possible) we avoided changing fitness functions in mid-stream. In summary, our standard GA does not fit into any of the nineteen categories mentioned in Section 2.7, but we strongly believe that it does fit the application.

Our empirical results suggest that if the GAs are going to converge then they will do so quite quickly, typically after twelve generations from a random start. By some GA standards, twelve generations is small; however, the start-point, the *seed*, is significant.

Runs have been carried out for many different seeds, and they all indicate that in areas of convergence (in fractal basin language, in a 'sea' of blue, yellow or red) it does not matter very much where the start-point is. In other words, the parameter space would seem to contain subspaces of near-perfect fitness. In areas bordering on fractal basin boundaries, however, the start-point appears to be critical.

GAs generating fractal basins (e.g. GA-10 and GA-11) are searching for balance-of-power points. All of them seek near-perfect balance (i.e. minimum fitness). High levels of fitness – a bad thing – occur near the fractal basin boundaries. In these boundary areas the near-perfect balance of power, even when found, appears to be sensitive to small-scale perturbations. (Fitness is a link between genotype and phenotype, i.e. between a genetic representation of the problem and the problem itself [Michalewicz, 1992]).

The balance-of-power points referred to above would appear to lie in hyperplanes through parameter space. According to the s*chema theorem* [Goldberg, 1989] GAs search these hyperplanes seeking schemas with good fitness. It looks, therefore, as if the parameter space in our case does actually contain regular sub-spaces of near-perfect fitness. If this is true, then a GA is ideal for solving high-dimensional non-linear systems such as the 2- or 3-nation Richardson model. In other words, arms races are not 'GA-hard' at all.

Appendix 4A shows a number of print-outs of diagnostic tables for one particular GA, GA-1, at Generation 0. Appendix 4B compares the behaviour of GAs with 10, 15 and 20 genes at Generation 39. Both appendices illustrate the processes explained earlier in Sections 4.3 to 4.8.

Appendix 4A

Complete Set of Diagnostic Tables for GA-1

(showing how the GA develops from Generation 0 to Generation 1).

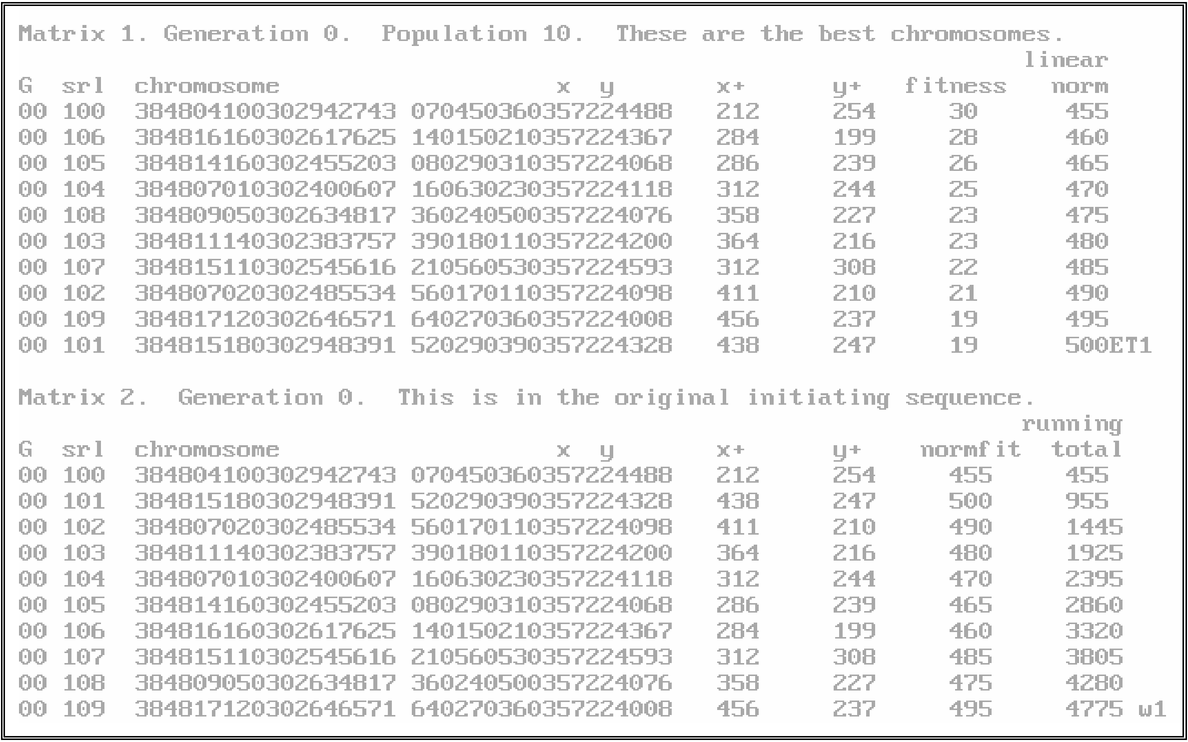
Guidance on Reading the Diagnostics

The population for this series of illustrations is only ten, so the movement of every element of a gene (within a chromosome) can be monitored. Note that all elements of the genes are represented by integers. The tables should be read in the following sequence:

- Matrix 1 in Figure 4A.1 (sorted into decreasing order of fitness, and showing the linear norm). Generation 0.
- Matrix 2 in Figure 4A.1 (unsorted, as generated). Normalised fitnesses (normfit) have replaced the actual fitnesses and a running total is being kept of them. Generation 0.
- Matrix 3 in Figure 4A.2 (as selected for parenthood from the unsorted Matrix 2). Note the last column of random numbers. Rows are transferred from Matrix 2 to Matrix 3 the first time that the running total in Matrix 2 exceeds the random number in Matrix 3. This is roulette wheel selection. Generation 0.
- Matrix 4 in Figure 4A.3. The last two rows of Matrix 3 have been ‘unbundled’ and set side by side in columns ready for mutation and crossover.
- Matrix 4a in Figure 4A.4. The results of 20% mutation on *every element* of Matrix 4. 20% is excessive but this is for demonstration only.
- Matrix 4b in Figure 4A.5. The results of crossover at Level 19. From Level 19, the columns are interchanged.
- Matrix 5 in Figure 4A.2. This shows the ‘rebundled’ chromosomes after the combined effects of selection, mutation and crossover. (Matrix 5 is the embryo of the next generation, Generation 1).
- Matrices 1 and 2 in Figure 4A.6. These correspond to the two matrices in Figure 4A.1 with respect to Generation 1.
- Matrix 0 (see Section 4.8) in Figure 4A.7. The Middle East prediction GA, GA-4 (population 100) showing *a* - *c*. Average defexp% of 8.6% gives 9 minuses.

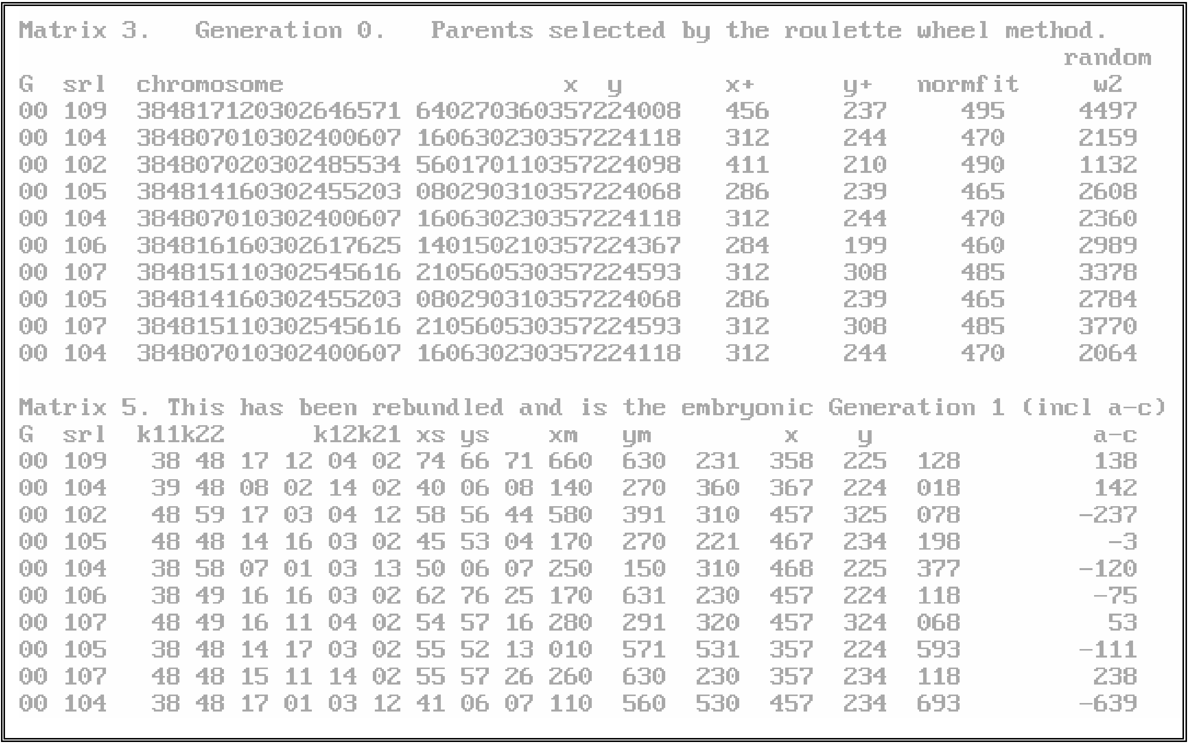
Comment. It will be appreciated that a tiny population combined with an excessive mutation rate and a run of only two generations are unlikely to result in proper solutions. This appendix aims only at demonstrating, primarily, the principles on which our GAs operate and, secondly, the value in the context of the thesis of having a full audit trail.

Figures 4A.1 to 4A.6 show the results of using a population of 10 and a run of 2 generations, with a mutation rate of 20% and a crossover rate of 55%. The figures are screenprints from a C++ source, so subscripts are not possible. The parameters *xs, ys, xm, ym, x, y,* *x*+ and *y*+ should be read as x_s , y_s , x_m , y_m , x_t , y_t , x_{t+1} and y_{t+1}, respectively. G means generation and s*rl* is the serial number.



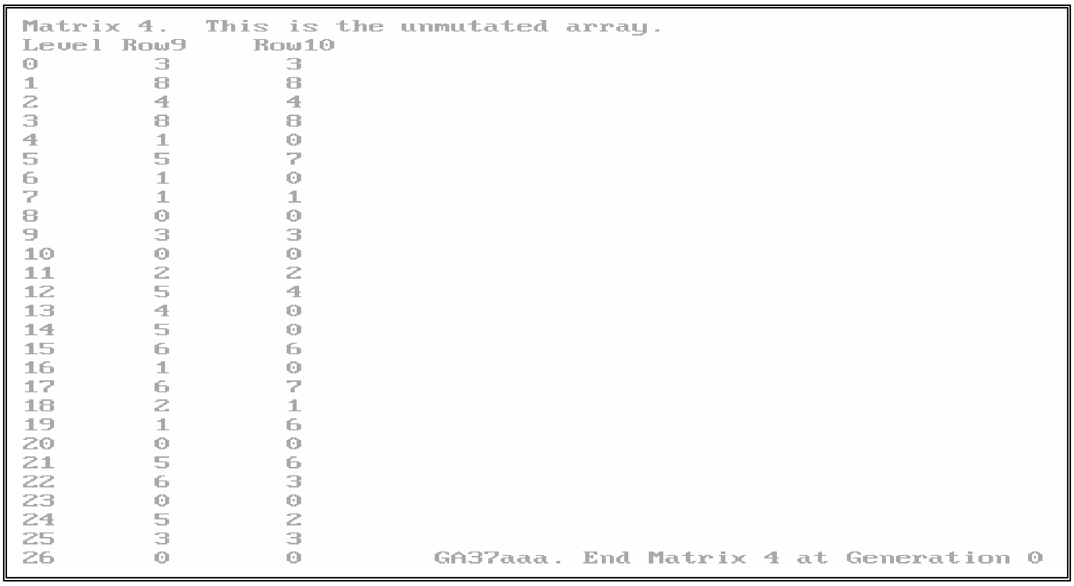
```
Matrix 1. Generation 0.  Population 10.  These are the best chromosomes.
                                                                     linear
G  srl  chromosome                   x  y       x+      y+   fitness   norm
00 100  384804100302942743 070450360357224488   212     254      30     455
00 106  384816160302617625 140150210357224367   284     199      28     460
00 105  384814160302455203 080290310357224068   286     239      26     465
00 104  384807010302400607 160630230357224118   312     244      25     470
00 108  384809050302634817 360240500357224076   358     227      23     475
00 103  384811140302383757 390180110357224200   364     216      23     480
00 107  384815110302545616 210560530357224593   312     308      22     485
00 102  384807020302485534 560170110357224098   411     210      21     490
00 109  384817120302646571 640270360357224008   456     237      19     495
00 101  384815180302948391 520290390357224328   438     247      19     500ET1

Matrix 2.  Generation 0.  This is in the original initiating sequence.
                                                                    running
G  srl  chromosome                   x  y       x+      y+    normfit  total
00 100  384804100302942743 070450360357224488   212     254     455     455
00 101  384815180302948391 520290390357224328   438     247     500     955
00 102  384807020302485534 560170110357224098   411     210     490     1445
00 103  384811140302383757 390180110357224200   364     216     480     1925
00 104  384807010302400607 160630230357224118   312     244     470     2395
00 105  384814160302455203 080290310357224068   286     239     465     2860
00 106  384816160302617625 140150210357224367   284     199     460     3320
00 107  384815110302545616 210560530357224593   312     308     485     3805
00 108  384809050302634817 360240500357224076   358     227     475     4280
00 109  384817120302646571 640270360357224008   456     237     495     4775 w1
```

Figure 4A.1 GA-1. Matrices 1 and 2 at Generation 0. Seed 8. defexp% 6.5. IMF



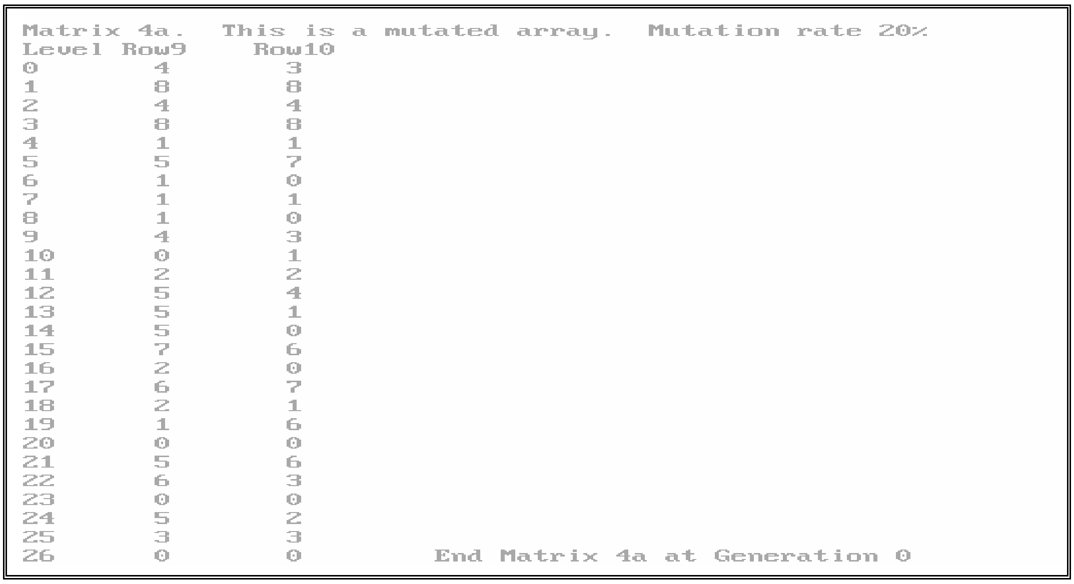
```
Matrix 3.   Generation 0.   Parents selected by the roulette wheel method.
                                                                       random
G  srl  chromosome                   x  y       x+      y+   normfit      w2
00 109  384817120302646571 640270360357224008   456     237     495     4497
00 104  384807010302400607 160630230357224118   312     244     470     2159
00 102  384807020302485534 560170110357224098   411     210     490     1132
00 105  384814160302455203 080290310357224068   286     239     465     2608
00 104  384807010302400607 160630230357224118   312     244     470     2360
00 106  384816160302617625 140150210357224367   284     199     460     2989
00 107  384815110302545616 210560530357224593   312     308     485     3378
00 105  384814160302455203 080290310357224068   286     239     465     2784
00 107  384815110302545616 210560530357224593   312     308     485     3770
00 104  384807010302400607 160630230357224118   312     244     470     2064

Matrix 5. This has been rebundled and is the embryonic Generation 1 (incl a-c)
G  srl  k11k22      k12k21 xs ys     xm  ym         x    y               a-c
00 109   38 48 17 12 04 02 74 66 71 660  630  231  358  225  128          138
00 104   39 48 08 02 14 02 40 06 08 140  270  360  367  224  018          142
00 102   48 59 17 03 04 12 58 56 44 580  391  310  457  325  078         -237
00 105   48 48 14 16 03 02 45 53 04 170  270  221  467  234  198           -3
00 104   38 58 07 01 03 13 50 06 07 250  150  310  468  225  377         -120
00 106   38 49 16 16 03 02 62 76 25 170  631  230  457  224  118          -75
00 107   48 49 16 11 04 02 54 57 16 280  291  320  457  324  068           53
00 105   38 48 14 17 03 02 55 52 13 010  571  531  357  224  593         -111
00 107   48 48 15 11 14 02 55 57 26 260  630  230  357  234  118          238
00 104   38 48 17 01 03 12 41 06 07 110  560  530  457  234  693         -639
```

Figure 4A.2 GA-1. Matrices 3 and 5 at Generation 0. Seed 8. defexp% 6.5. IMF



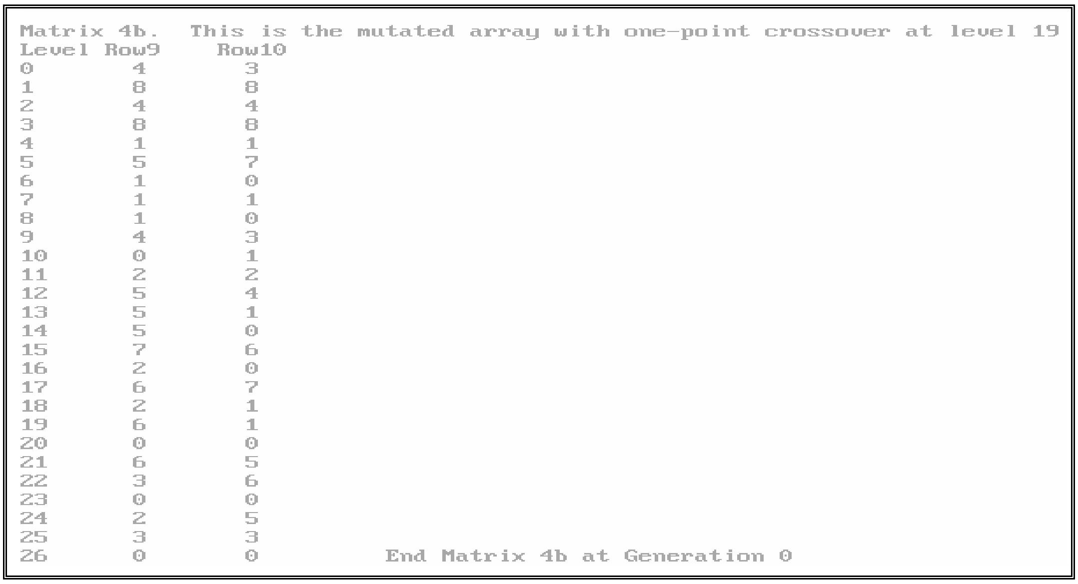
```
Matrix 4.   This is the unmutated array.
Level Row9     Row10
0        3        3
1        8        8
2        4        4
3        8        8
4        1        0
5        5        7
6        1        0
7        1        1
8        0        0
9        3        3
10       0        0
11       2        2
12       5        4
13       4        0
14       5        0
15       6        6
16       1        0
17       6        7
18       2        1
19       1        6
20       0        0
21       5        6
22       6        3
23       0        0
24       5        2
25       3        3
26       0        0         GA37aaa. End Matrix 4 at Generation 0
```

Figure 4A.3 GA-1. Matrix 4 at Generation 0. Seed 8. defexp% 6.5. IMF. Unbundled



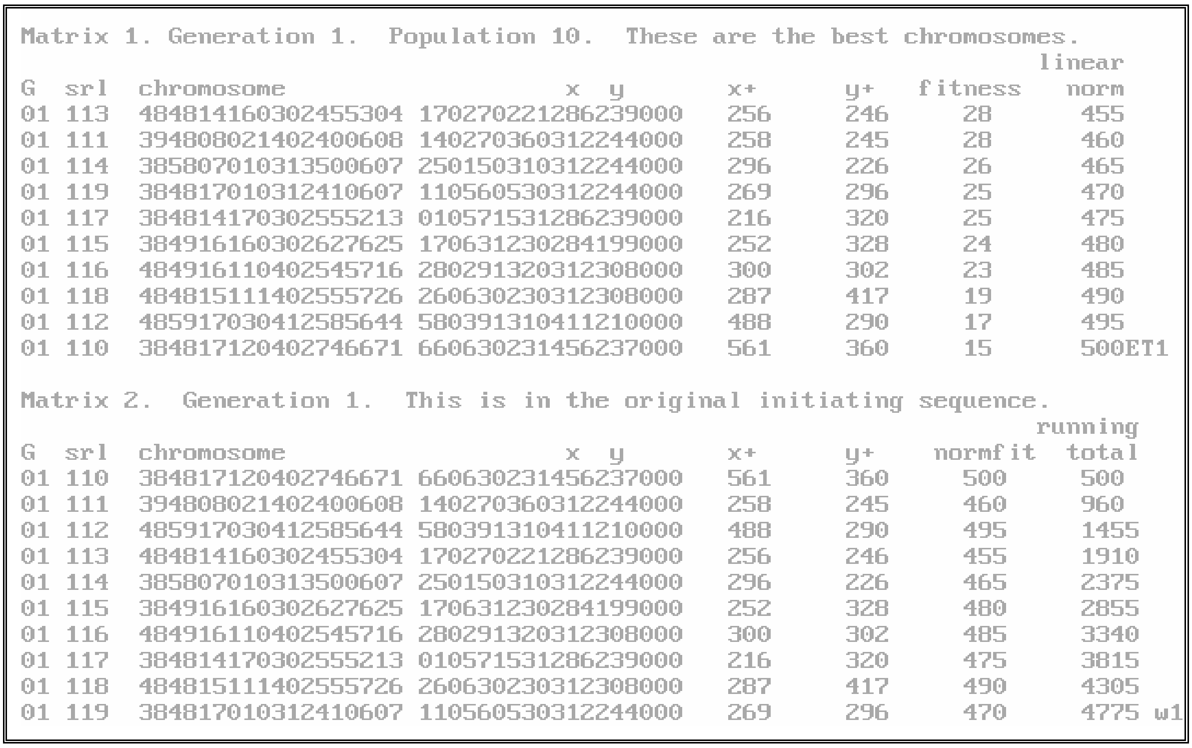
```
Matrix 4a.  This is a mutated array.  Mutation rate 20%
Level Row9     Row10
0        4        3
1        8        8
2        4        4
3        8        8
4        1        1
5        5        7
6        1        0
7        1        1
8        1        0
9        4        3
10       0        1
11       2        2
12       5        4
13       5        1
14       5        0
15       7        6
16       2        0
17       6        7
18       2        1
19       1        6
20       0        0
21       5        6
22       6        3
23       0        0
24       5        2
25       3        3
26       0        0         End Matrix 4a at Generation 0
```

Figure 4A.4 GA-1. Matrix 4 at Generation 0. Seed 8. defexp% 6.5. IMF. Unbundled



```
Matrix 4b.  This is the mutated array with one-point crossover at level 19
Level Row9     Row10
0        4        3
1        8        8
2        4        4
3        8        8
4        1        1
5        5        7
6        1        0
7        1        1
8        1        0
9        4        3
10       0        1
11       2        2
12       5        4
13       5        1
14       5        0
15       7        6
16       2        0
17       6        7
18       2        1
19       6        1
20       0        0
21       6        5
22       3        6
23       0        0
24       2        5
25       3        3
26       0        0         End Matrix 4b at Generation 0
```

Figure 4A.5 GA-1. Matrix 4b at Generation 0. Seed 8. defexp% 6.5. IMF. Unbundled

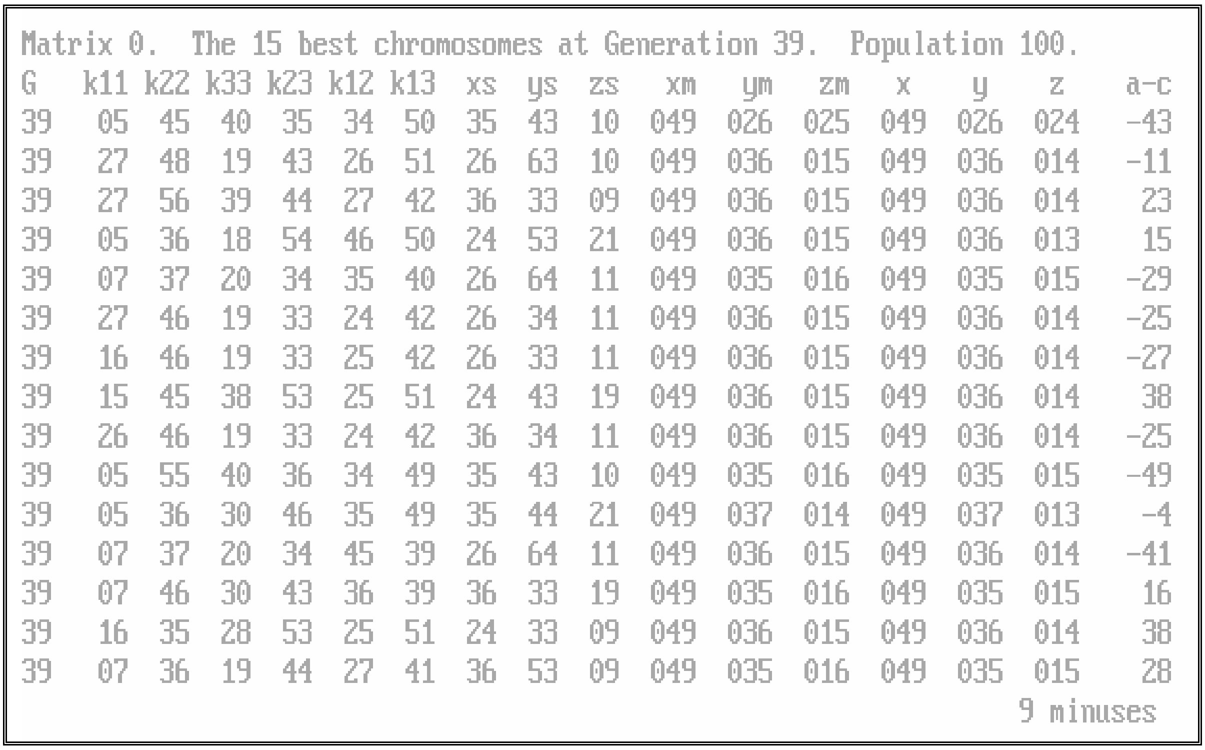


```
Matrix 1. Generation 1.  Population 10.  These are the best chromosomes.
                                                                      linear
G  srl  chromosome                     x  y        x+       y+   fitness   norm
01 113  484814160302455304 170270221286239000   256      246      28       455
01 111  394808021402400608 140270360312244000   258      245      28       460
01 114  385807010313500607 250150310312244000   296      226      26       465
01 119  384817010312410607 110560530312244000   269      296      25       470
01 117  384814170302555213 010571531286239000   216      320      25       475
01 115  384916160302627625 170631230284199000   252      328      24       480
01 116  484916110402545716 280291320312308000   300      302      23       485
01 118  484815111402555726 260630230312308000   287      417      19       490
01 112  485917030412585644 580391310411210000   488      290      17       495
01 110  384817120402746671 660630231456237000   561      360      15       500ET1

Matrix 2.  Generation 1.  This is in the original initiating sequence.
                                                                      running
G  srl  chromosome                     x  y        x+       y+     normfit  total
01 110  384817120402746671 660630231456237000   561      360       500     500
01 111  394808021402400608 140270360312244000   258      245       460     960
01 112  485917030412585644 580391310411210000   488      290       495     1455
01 113  484814160302455304 170270221286239000   256      246       455     1910
01 114  385807010313500607 250150310312244000   296      226       465     2375
01 115  384916160302627625 170631230284199000   252      328       480     2855
01 116  484916110402545716 280291320312308000   300      302       485     3340
01 117  384814170302555213 010571531286239000   216      320       475     3815
01 118  484815111402555726 260630230312308000   287      417       490     4305
01 119  384817010312410607 110560530312244000   269      296       470     4775 w1
```

Figure 4A.6 GA-1. Matrices 1 and 2 at Generation 1. Seed 8. defexp% 6.5. IMF

.

For Figure 4A.7, the population is 100, the run is 40 generations, the mutation rate is 1% and the crossover rate is 65%



```
Matrix 0.  The 15 best chromosomes at Generation 39.  Population 100.
G   k11 k22 k33 k23 k12 k13  xs  ys  zs   xm   ym   zm    x    y    z   a-c
39   05  45  40  35  34  50  35  43  10  049  026  025  049  026  024  -43
39   27  48  19  43  26  51  26  63  10  049  036  015  049  036  014  -11
39   27  56  39  44  27  42  36  33  09  049  036  015  049  036  014   23
39   05  36  18  54  46  50  24  53  21  049  036  015  049  036  013   15
39   07  37  20  34  35  40  26  64  11  049  035  016  049  035  015  -29
39   27  46  19  33  24  42  26  34  11  049  036  015  049  036  014  -25
39   16  46  19  33  25  42  26  33  11  049  036  015  049  036  014  -27
39   15  45  38  53  25  51  24  43  19  049  036  015  049  036  014   38
39   26  46  19  33  24  42  36  34  11  049  036  015  049  036  014  -25
39   05  55  40  36  34  49  35  43  10  049  035  016  049  035  015  -49
39   05  36  30  46  35  49  35  44  21  049  037  014  049  037  013   -4
39   07  37  20  34  45  39  26  64  11  049  036  015  049  036  014  -41
39   07  46  30  43  36  39  36  33  19  049  035  016  049  035  015   16
39   16  35  28  53  25  51  24  33  09  049  036  015  049  036  014   38
39   07  36  19  44  27  41  36  53  09  049  035  016  049  035  015   28
                                                                 9 minuses
```

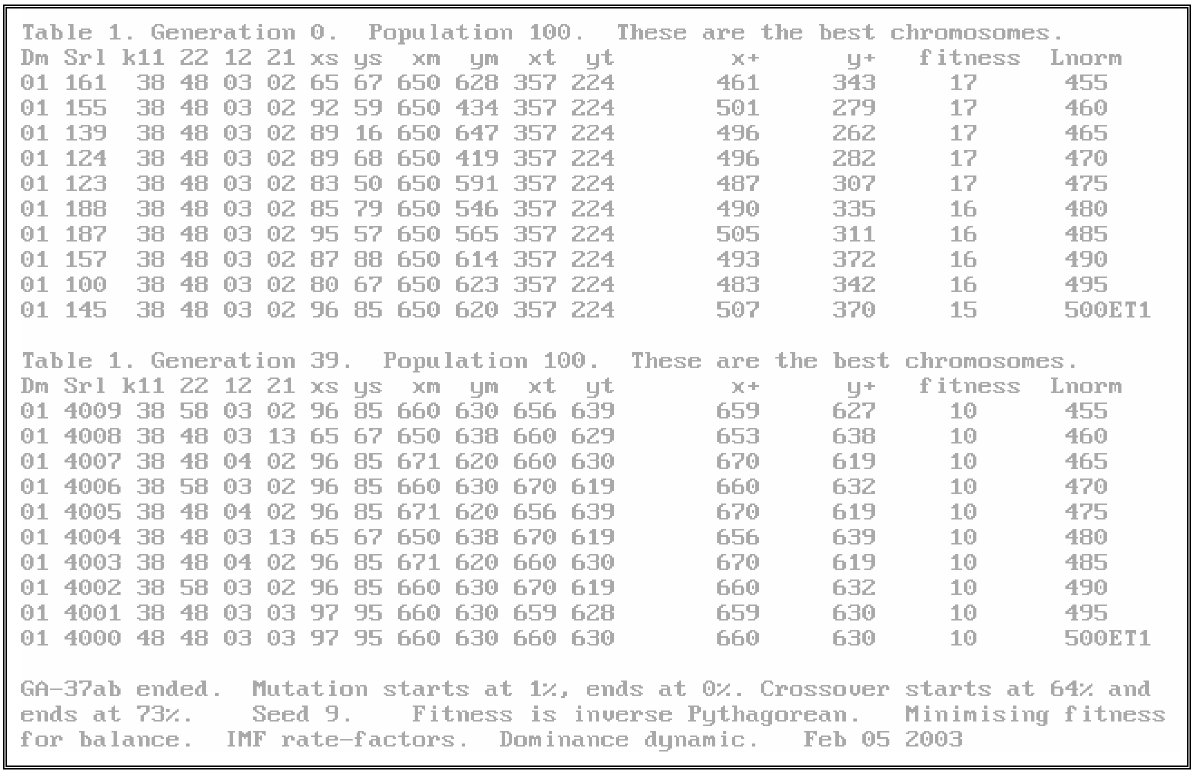
Figure 4A.7 The prediction GA, GA-4, with *a - c*. Average defexp% of 8.6% yields 9 minuses; *cf.* the Middle East *prediction plot* in Figure 6.14

Appendix 4B

Output of Three GAs with 10 (GA-0), 15 (GA-1) , and 20 (GA-9) Genes

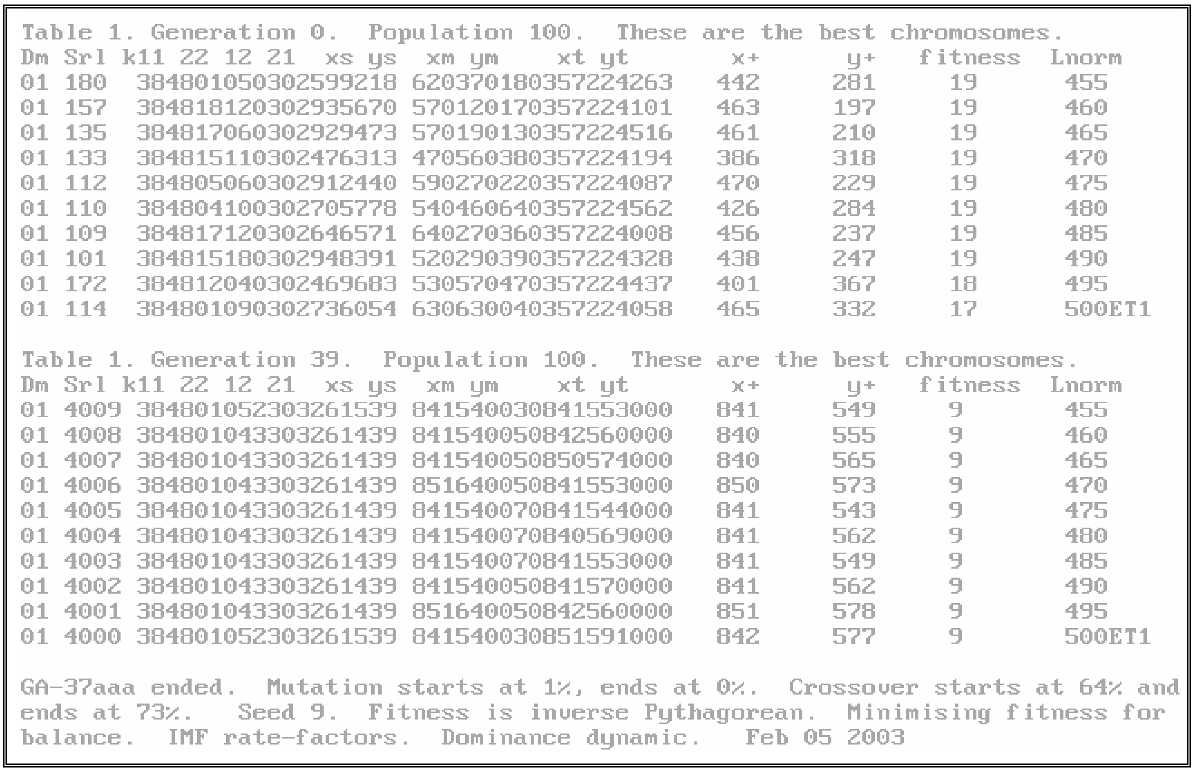
The Three GAs

Each of these has a population of 100 and runs for 40 generations, with varying mutation and crossover rates. As in Appendix 4A, Figures 4B.1, 4B.2 and 4B.3 are screenprints from a C++ source, so subscripts are not possible. The parameters *xs, ys, xm, ym, xt, yt, x+* and *y+* should be read as x_s , y_s , x_m , y_m , x_t , y_t , x_{t+1} and y_{t+1}, respectively. *Dm* means dominant nation and *Srl* is the serial number. The linear norm is here called *Lnorm.*



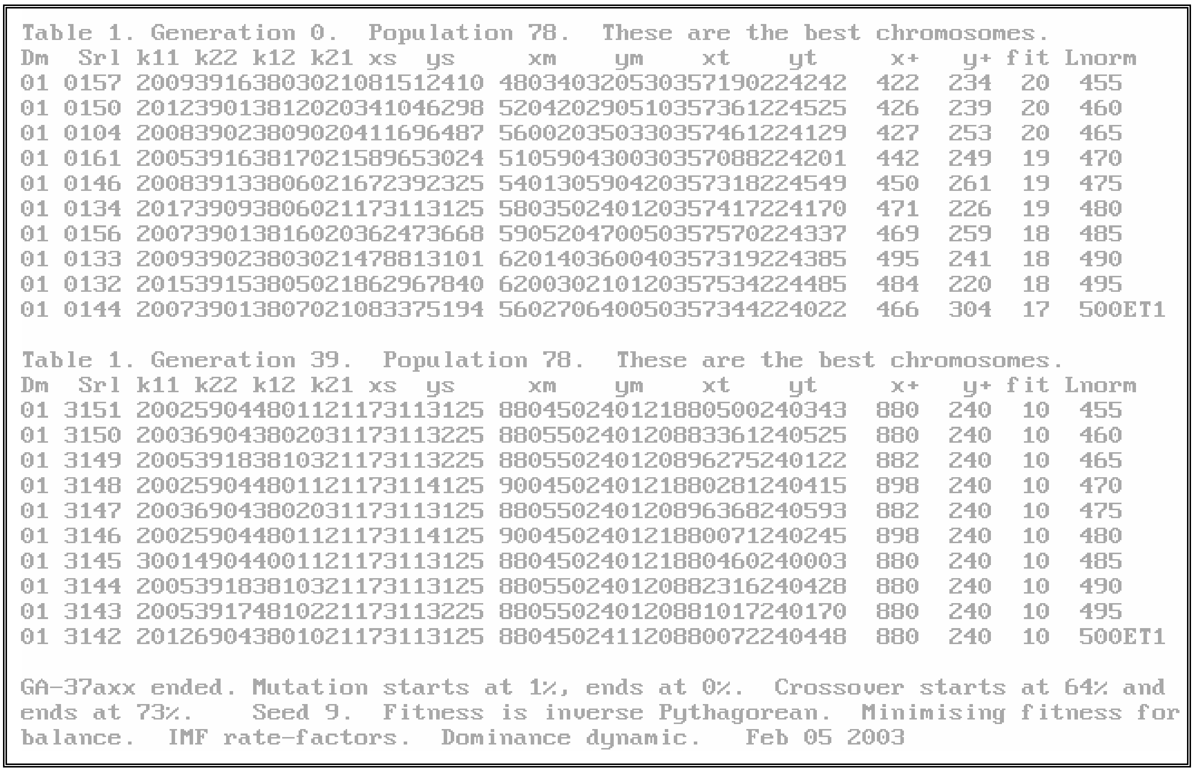
```
Table 1. Generation 0.  Population 100.  These are the best chromosomes.
Dm Srl k11 22 12 21 xs ys  xm  ym  xt  yt        x+      y+   fitness  Lnorm
01 161  38 48 03 02 65 67 650 628 357 224       461     343     17      455
01 155  38 48 03 02 92 59 650 434 357 224       501     279     17      460
01 139  38 48 03 02 89 16 650 647 357 224       496     262     17      465
01 124  38 48 03 02 89 68 650 419 357 224       496     282     17      470
01 123  38 48 03 02 83 50 650 591 357 224       487     307     17      475
01 188  38 48 03 02 85 79 650 546 357 224       490     335     16      480
01 187  38 48 03 02 95 57 650 565 357 224       505     311     16      485
01 157  38 48 03 02 87 88 650 614 357 224       493     372     16      490
01 100  38 48 03 02 80 67 650 623 357 224       483     342     16      495
01 145  38 48 03 02 96 85 650 620 357 224       507     370     15      500ET1

Table 1. Generation 39.  Population 100.  These are the best chromosomes.
Dm Srl k11 22 12 21 xs ys  xm  ym  xt  yt        x+      y+   fitness  Lnorm
01 4009 38 58 03 02 96 85 660 630 656 639       659     627     10      455
01 4008 38 48 03 13 65 67 650 638 660 629       653     638     10      460
01 4007 38 48 04 02 96 85 671 620 660 630       670     619     10      465
01 4006 38 58 03 02 96 85 660 630 670 619       660     632     10      470
01 4005 38 48 04 02 96 85 671 620 656 639       670     619     10      475
01 4004 38 48 03 13 65 67 650 638 670 619       656     639     10      480
01 4003 38 48 04 02 96 85 671 620 660 630       670     619     10      485
01 4002 38 58 03 02 96 85 660 630 670 619       660     632     10      490
01 4001 38 48 03 03 97 95 660 630 659 628       659     630     10      495
01 4000 48 48 03 03 97 95 660 630 660 630       660     630     10      500ET1

GA-37ab ended.  Mutation starts at 1%, ends at 0%. Crossover starts at 64% and
ends at 73%.    Seed 9.    Fitness is inverse Pythagorean.   Minimising fitness
for balance.  IMF rate-factors.  Dominance dynamic.   Feb 05 2003
```

Figure 4B.1 GA-0. 10 genes. (No redundancy). Matrix 1. Initial IMF rate-factors



```
Table 1. Generation 0.  Population 100.  These are the best chromosomes.
Dm Srl k11 22 12 21  xs ys  xm ym    xt yt        x+       y+   fitness  Lnorm
01 180  384801050302599218 620370180357224263   442      281      19       455
01 157  384818120302935670 570120170357224101   463      197      19       460
01 135  384817060302929473 570190130357224516   461      210      19       465
01 133  384815110302476313 470560380357224194   386      318      19       470
01 112  384805060302912440 590270220357224087   470      229      19       475
01 110  384804100302705778 540460640357224562   426      284      19       480
01 109  384817120302646571 640270360357224008   456      237      19       485
01 101  384815180302948391 520290390357224328   438      247      19       490
01 172  384812040302469683 530570470357224437   401      367      18       495
01 114  384801090302736054 630630040357224058   465      332      17       500ET1

Table 1. Generation 39.  Population 100.  These are the best chromosomes.
Dm Srl k11 22 12 21  xs ys  xm ym    xt yt        x+       y+   fitness  Lnorm
01 4009 384801052303261539 841540030841553000   841      549      9        455
01 4008 384801043303261439 841540050842560000   840      555      9        460
01 4007 384801043303261439 841540050850574000   840      565      9        465
01 4006 384801043303261439 851640050841553000   850      573      9        470
01 4005 384801043303261439 841540070841544000   841      543      9        475
01 4004 384801043303261439 841540070840569000   841      562      9        480
01 4003 384801043303261439 841540070841553000   841      549      9        485
01 4002 384801043303261439 841540050841570000   841      562      9        490
01 4001 384801043303261439 851640050842560000   851      578      9        495
01 4000 384801052303261539 841540030851591000   842      577      9        500ET1

GA-37aaa ended.  Mutation starts at 1%, ends at 0%.  Crossover starts at 64% and
ends at 73%.   Seed 9.  Fitness is inverse Pythagorean.  Minimising fitness for
balance.  IMF rate-factors.  Dominance dynamic.   Feb 05 2003
```

Figure 4B.2 GA-1. 15 genes. (33% redundancy). Matrix 1. Initial IMF rate-factors



```
Table 1. Generation 0.  Population 78.  These are the best chromosomes.
Dm  Srl k11 k22 k12 k21 xs  ys     xm    ym    xt    yt      x+   y+ fit Lnorm
01 0157 20093916380302108151 2410 480340320530357190224242  422  234  20  455
01 0150 20123901381202034104 6298 520420290510357361224525  426  239  20  460
01 0104 20083902380902041169 6487 560020350330357461224129  427  253  20  465
01 0161 20053916381702158965 3024 510590430030357088224201  442  249  19  470
01 0146 20083913380602167239 2325 540130590420357318224549  450  261  19  475
01 0134 20173909380602117311 3125 580350240120357417224170  471  226  19  480
01 0156 20073901381602036247 3668 590520470050357570224337  469  259  18  485
01 0133 20093902380302147881 3101 620140360040357319224385  495  241  18  490
01 0132 20153915380502186296 7840 620030210120357534224485  484  220  18  495
01 0144 20073901380702108337 5194 560270640050357344224022  466  304  17  500ET1

Table 1. Generation 39.  Population 78.  These are the best chromosomes.
Dm  Srl k11 k22 k12 k21 xs  ys     xm    ym    xt    yt      x+   y+ fit Lnorm
01 3151 200259044801121173113125 880450240121880500240343  880  240  10  455
01 3150 200369043802031173113225 880550240120883361240525  880  240  10  460
01 3149 200539183810321173113225 880550240120896275240122  882  240  10  465
01 3148 200259044801121173114125 900450240121880281240415  898  240  10  470
01 3147 200369043802031173113125 880550240120896368240593  882  240  10  475
01 3146 200259044801121173114125 900450240121880071240245  898  240  10  480
01 3145 300149044001121173113125 880450240121880460240003  880  240  10  485
01 3144 200539183810321173113125 880550240120882316240428  880  240  10  490
01 3143 200539174810221173113225 880550240120881017240170  880  240  10  495
01 3142 201269043801021173113125 880450241120880072240448  880  240  10  500ET1

GA-37axx ended. Mutation starts at 1%, ends at 0%.  Crossover starts at 64% and
ends at 73%.    Seed 9.  Fitness is inverse Pythagorean.  Minimising fitness for
balance.  IMF rate-factors.  Dominance dynamic.   Feb 05 2003
```

Figure 4B.3 GA-9. 20 genes. (50% redundancy). Matrix 1. Initial IMF rate-factors

CHAPTER 5

Predicting Instabilities in Arms Races: a Rationale

5.1 Introduction

This chapter sets out to explain the rationale behind our prediction of instabilities between two nations. We will do this in general and, throughout the chapter, whenever an illustration is required we will use a snapshot (1993 to 1999) in the continuing arms race between India and Pakistan as a *running example* to demonstrate the general techniques involved.

We make use of two integer-valued GAs which, while related in appearance and operation, are dissimilar in function. The first (the *modelling* GA) generates a model designed to mimic the IMF data *and* to follow the principles of arms races epitomised by the Richardson equations. The second GA (the *prediction* GA) is used to predict instabilities by looking for violent changes in the direction of curvature of the limit cycle of the evaluation equations, a technique pioneered by Peng *et al.* [1991] for forecasting canard explosions.

In explaining the rationale, we illustrate that GAs can find viable solutions to Richardson's arms race equations and, in so doing, predict the timings of instabilities that may lead to open warfare. In the specific case of India and Pakistan, for instance, it was found from our running example: first, that very small changes in the *average* percentage defence expenditure (defexp%) of the two nations could make all the difference between stability and instability; second, that although there are dangerous patches of potential instability in the developing arms race, nevertheless there are large areas of stability as well; and third, that the model appears to meet Richardson's theoretical criteria of stability and instability.

The remainder of this chapter starts by referring back briefly to the Richardson equations and stating what any solution is required to do. It explains the *modus operandi*, how the modell-ing GA is created and how the rate-factors are established and verified. It then turns to the prediction GA and again uses our running example to illustrate how the system works. This is followed by a graphical demonstration of sporadic instability, and an explanation of dominance diagrams and fractal basins. The chapter closes by discussing some alternative non-GA techniques for the Richardson arms race equations (including Monte Carlo and simulated annealing) and by demonstrating the superiority of the GA approach over them in terms of speed, flexibility, ease-of-use and, above all, the ability to predict instabilities.

5.2 Basics

Let us revisit the two-nation equations

$$\frac{dy}{dt} = ax - by + h \quad \text{and} \quad \frac{dx}{dt} = cy - fx + g\,, \tag{5.1}$$

where x and y is the money spent annually on defence by Nations X and Y. Richardson showed (Section 3.2) that stability occurs when $ac < bf$. Our concern, however, is primarily with instability which occurs when

$$ac > bf. \tag{5.2}$$

Figure 5.1 below is the result of plotting x (India) against y (Pakistan) on equally-scaled axes.

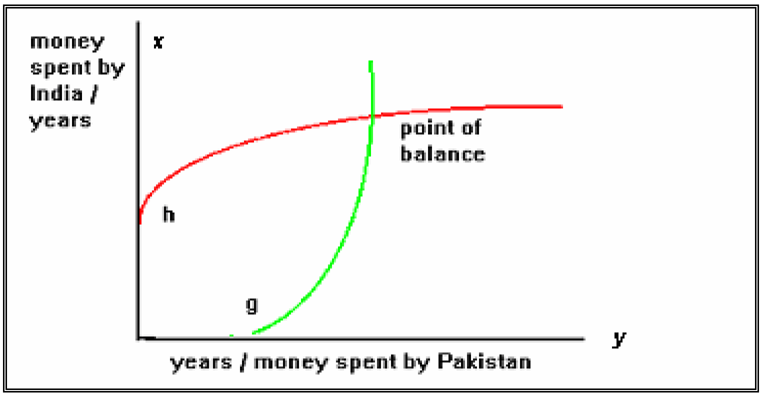


Figure 5.1 Likely contours for the arms race between India and Pakistan

The *point of balance* is where defence expenditure by the two sides is equal and where the threat of war notionally ceases to exist.

A solution is required:

- To establish arms race contours for India and Pakistan.
- To establish specific rates of defence expenditure for both sides over periods of time which, first, mirror real-world data and, second, are potentially capable of extrapolation.
- To seek the fittest solution which in our context means minimising the risk of war.

We next restate Eqs(3.14), namely

$$x_{t+1} = x_t + \big(k_{1,1}(x_s - x_t) + k_{1,2}\, y_t\big)\,(x_m - x_t) \quad .$$

$$y_{t+1} = y_t + (k_{2,2}\,(y_s - y_t) + k_{2,1}\, x_t)\,(y_m - y_t)\,, \tag{5.3}$$

where in a 2-nation scenario:

- x_t and y_t are the actual *defence expenditures* of Nations X and Y in the current year; x_{t+1} and y_{t+1} are those for next year. (The initial values of x_t and y_t are denoted by x_0 and y_0). In the thesis, x_t and y_t are also referred to as *defexp* (see Glossary).

- x_s and y_s are the *intrinsic* expenditures on arms (how much each nation spends on defence irrespective of competitive spending by its neighbours). These are essentially the standing costs of the Armed Forces and are typically 80% of *defexp.*
- x_m and y_m are the expenditures authorised on arms in the budgets of Nations X and Y. In an ideal world, x_t would equal x_m , and y_t would equal y_m . We call x_m and y_m *budgetary defence expenditures.*
- $k_{i,j}$, where $i,j \in \{1,2\}$, are the four *rate-factors*. These have the dimension of $(\text{time})^{-1}$, and may be negative. They are a measure of how fast a nation can react to changing threats.

Everything is time-dependent in these equations so nothing is constant. In our empirical experience, a, b, c and f, and the rate-factors $k_{i,j}$ all vary slowly with time, but nothing like so fast as x and y.

5.3 Modus Operandi: Predicting Instabilities and Conflicts

First, we plot the published IMF defence expenditure figures against time. This is the *IMF plot*. Second, using the evaluation equations and a number of numerical approximations, we employ a *modelling GA* as a vehicle to generate curves that *mimic* those of the IMF plot. This is the *model.* Third, we extract the rate-factors from the evaluation equations that generated the model and use them in a *prediction GA* to predict instabilities which occur when there is a significant number of sign changes in $a - c$. These yield the *prediction plot*. Fourth, we plot the percentage defence expenditure figures of the two or three nations involved against time, and take the average. These figures are obtained from IMF statistics, and yield the *defexp% plot*. Fifth, *we* use the defexp% plot to identify the dates associated with the violent sign changes indicated on the prediction plot. Finally, we verify these dates against appropriate historical references.

Note. A *conflict* is taken to mean an instability likely to lead to an outbreak of war; it does not necessarily mean that open warfare actually resulted. In Richardson's words, "war is more likely than not". As an example, in 1975 India started to divert the waters of the River Jhelem, which, if it were completed, would have made a significant part of Pakistan into a desert. It is not surprising that the Pakistani Government was upset and hurriedly increased their preparations for war, and this is reflected in a change in the direction of the curvature of the limit cycle. It is now known (years later) that in the event the United States stepped in and persuaded India to desist, so no war actually took place. See *Results* (e) of Section 6.4.2.

Similarly, in recent years Greece and Turkey have come very close to open warfare over the Aegean and Cyprus but again, due to United States intervention, this has been averted. The problem is that such instances (where war is averted through, say, covert and therefore unpublished diplomatic action) are likely to lead to false-positives (see Table 5.2).

For our running example, the IMF plot, the model, the prediction plot and the defexp% plot are given in Figures 5.2, 5.5, 5.6 and 5.7, respectively.

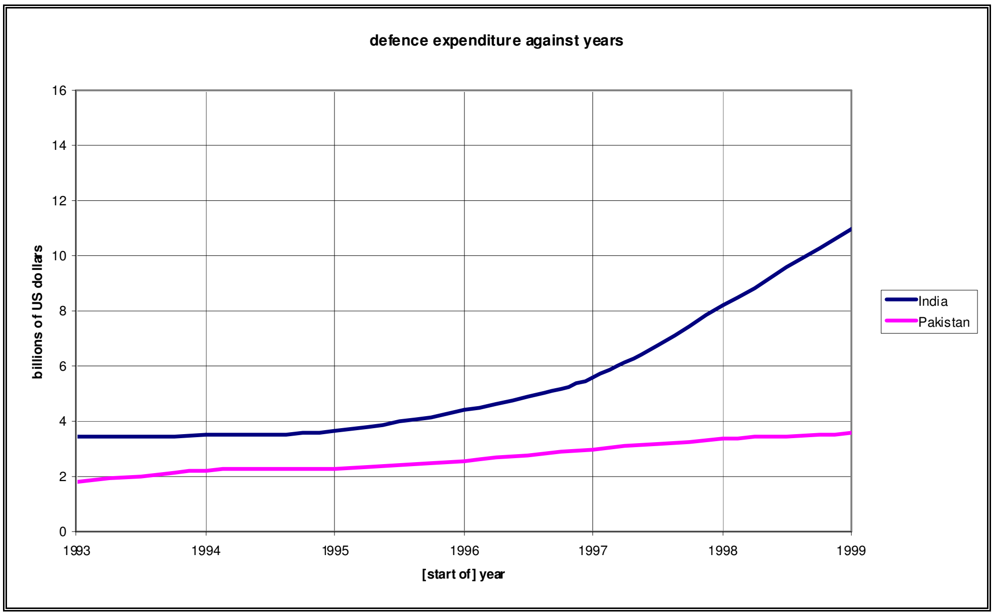


Figure 5. 2 *IMF plot.* Data from Appendix 6A.2

5.4 The Model

Curve-fitting, in the classic sense, is very limiting; it does not reflect the physics of the process under consideration, in that no association is required between the equation of the line joining the points making up a graph, and the equation of whatever process gave rise to those points. For this reason, it is not advisable to extrapolate fitted curves. In contrast, we need " … *a kinetic model faithful to the input data and the physical principles involved"* [Cao *et al.*, 2000]. Such a model will inevitably involve domain knowledge. Our aim is to predict outbreaks of war in the future, so our methods must allow forward extrapolation. To this end we next introduce our modelling GA and illustrate its use with our running example.

5.4.1 The Modelling GA

We employ GA-13 as a *modelling* GA whose sole function is to mimic IMF curves such as those of Figure 5.2. The GA has a population of 100 and its associated matrices follow the pattern given in Table 5.1. The chromosomes (shaded) contain six 2-element genes (columns 2-7) and four 3-element genes (columns 8-11), each element being in the set {0 … 9}, giving a search space of 10^{24}.

Table 5.1 GA-13. Matrices M1, M2 and M3

Col	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
M1	*gen*	*srl*	$k_{1,1}$	$k_{1,2}$	$k_{2,1}$	$k_{2,2}$	x_s	y_s	x_m	y_m	x_t	y_t	x_{t+1}	y_{t+1}	*fitness*	*linear norm*
M2	*gen*	*srl*	$k_{1,1}$	$k_{1,2}$	$k_{2,1}$	$k_{2,2}$	x_s	y_s	x_m	y_m	x_t	y_t	x_{t+1}	y_{t+1}	*normfit*	*running total*
M3	*gen*	*srl*	$k_{1,1}$	$k_{1,2}$	$k_{2,1}$	$k_{2,2}$	x_s	y_s	x_m	y_m	x_t	y_t	x_{t+1}	y_{t+1}	*normfit*	*random number*

At Generation 0, Matrices 1 and 2 are input with the starter-values for the rate-factors, actual values for x_0 and y_0, and input from a constrained RNG for every value of x_s, y_s, x_m and y_m.

The GA then proceeds along the lines described in Chapter 4. At Generation 0 input is fed simultaneously into Matrices 1 and 2, followed by evaluation (using, in this case, Eqs(5.4)), and fitness computation. At this stage Matrix 1 is bubble-sorted into a descending order of fitness and the best result – the ‘bottom line’ – is recorded and plotted. Matrix 2 maintains its original row order, and is subject to roulette wheel selection (*weighted random pairing by cost,* see Section 2.7); the selected output is put into Matrix 3. By this stage we have already lost some of the original (albeit less fit) individuals and fitter ones have taken their place. Matrices 4/4a/4b ‘unbundle’, mutate and recombine the pairs of chromosomes, and Matrix 5 ‘rebundles’ them. We are still in Generation 0, but Matrix 5 already looks different from Matrix 2 both in the individuals it contains and the contents of its chromosomes. At this stage:

- Every chromosome in Matrix 5 is transferred back into Matrices 1 and 2, and Generation 1 begins.
- All the x_{t+1} and y_{t+1} results of Generation 0 are transferred from Matrix 3 back into Matrices 1 and 2 to become the x_t and y_t settings of Generation 1.
- The process is repeated for as many generations as is necessary.

5.4.2 Establishing the Rate-Factors

We next develop a technique to estimate the rate-factors for use in the prediction GA. As before, we use our running example to illustrate it.

We start with the approximate formulae

$$\begin{aligned} x_{t+1} &= x_t + (-0.22\, x_t\, k_{1,1} /10 + y_t\, k_{1,2} / 50)\, GDP_X /1000 \quad . \\ y_{t+1} &= y_t + (-0.32\, y_t\, k_{2,2} /10 + x_t\, k_{2,1} / 50)\, GDP_Y /1000 \quad . \end{aligned} \tag{5.4}$$

As they stand, Eqs(5.4) are unusable. We have two equations with four unknowns, i.e. the rate-factors $k_{i,j}$. However, we do know x, y, GDP_X and GDP_Y at t_0 and t_1 from the IMF data at Appendix 6A.2. Furthermore, we know from experience that $k_{1,2}$ and $k_{2,1}$ happen normally to lie between 2 and 7.

The description that follows is given in algorithmic form. It should be read in conjunction with both the notes shown in parentheses, and with those points that follow the algorithmic presentation.

begin

Initiate a modelling GA, with population 100, a fixed number of generations, using Eqs(5.4) as evaluation equations, and an inverse Pythagorean ffn, see Eq(5.5).

begin

Step 1. Estimate likely values for $k_{1,2}$ and $k_{2,1}$ and, in conjunction with the IMF values for x_0, y_0, x_1, y_1, GDP_X and GDP_Y, use Eqs(5.4) to determine $k_{1,1}$ and $k_{2,2}$. (We now have starter-values for all four rate-factors. As it happens, if we set $k_{1,2} = 6$ and $k_{2,1} = 2$, then $k_{1,1} = 46$ and $k_{2,2} = 2$).

Step 2. Using the four rate-factors in GA-13, compute x_{t+1} and y_{t+1}, see Figure 5.3. (Initially, this approach allows a comparison between the calculated values of x_1 and y_1 and their IMF values.)

Step 3. Run GA-13 over the required period.

Step 4. Examine the output graphically. (At first, the curves will be dysfunctional, except for their start-points x_0 and y_0 which are fixed). Adjust the four rate-factors so as to bring the two model curves closer to those of the IMF plot. (The resulting graph for the running example looks like that in Figure 5.3).

Step 5. Go to *Step* 4, adjusting the rate-factors and iterating the model until the output (model) curves lie as close as possible to those of the IMF plot.

end

(***Comment.*** We elaborate algorithmically further on Steps 4 and 5.)

```
    begin
      Step 6.  Improve the output (model) curves by reducing Cao et al.'s norm, see the Glossary and Figure 5.4.  Systematically adjust the rate-factors by trial and error in order to effect the reduction.
      Step 7.  Repeat Step 6 until the norm is as low as it can be.  (Figure 5.4 is the result of adjusting the rate-factors and attaining the values $k_{1,1} = 17$, $k_{2,2} = 5$, $k_{1,2} = 7$ and $k_{2,1} = 1$; this reduces the norm from 196 (obtained after the final  invocation of Step 5) to 125.  This, in turn, places the calculated values well within 5% of their IMF values.
      Step 8. Run GA-13 and compare it with the IMF plot. (An  output for the running example is  shown in Figure 5.5, and this should be compared with Figure 5.2).
    end
  if (termination condition) end
end
```

The following points should be made:

(1) Eqs(5.4) are a simplified version of Eqs(5.3), i.e. they contain less parameters Additionally, all equations must be scaled so that the curves appear on the screen.

(2) *Step* 4. In general, altering $k_{1,1}$ and $k_{2,2}$ will change the level of x_{t+1} and y_{t+1}, respectively, while adjusting $k_{1,2}$ and $k_{2,1}$ will change the shape of the intervening curve, i.e. between x_0, y_0 and the final x_t, y_t.

(3) *Step* 5. It is easier to make fine adjustments by tabulating x_t and y_t rather than by visual inspection. In practice, it was necessary to iterate GA-13 about twenty times, and the whole procedure took around thirty minutes.

(4) In Chapter 4 it was stated that IMF rate-factors were introduced at Generation 0 only. Whilst true of all prediction GAs, this is *not* true of the modelling GA where, *at every generation*, the IMF rate-factors are made to overwrite those carried forward from the previous generation. In effect, therefore, we are using the same rate-factors for every generation, giving us very tight control over the shape and positions of the resulting curves. We can do this because we know the answers (given in the IMF plot) in advance. We employ this technique only up to the point where the stage is set to reduce Cao *et al.*'s norm.

(5) For most of our work with GA-13 we used the inverse Pythagorean ffn

$$fitness = \frac{10000}{\sqrt{1 + x_{t+1}^2 + y_{t+1}^2}} \quad . \tag{5.5}$$

However, when we reduce the norm we employ the ffn

$$fitness = \sqrt{\left(x_{calc} - x_{real}\right)^2 + \left(y_{calc} - y_{real}\right)^2} \quad , \tag{5.6}$$

where x_{real}, y_{real} stand collectively for all the IMF values and x_{calc}, y_{calc} stand collectively for the corresponding computed values, thereby taking *all* the available IMF data into account. Figure 5.4 shows Cao et al.'s norm in action, while Figure 5.5 shows the output of the modelling GA, GA-13, after reducing the norm.

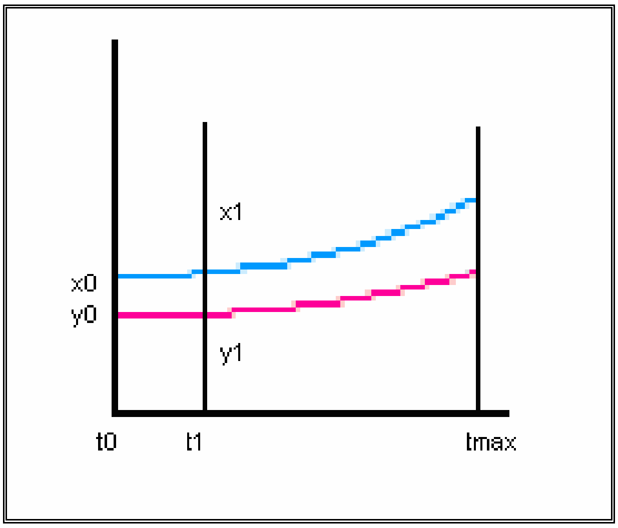
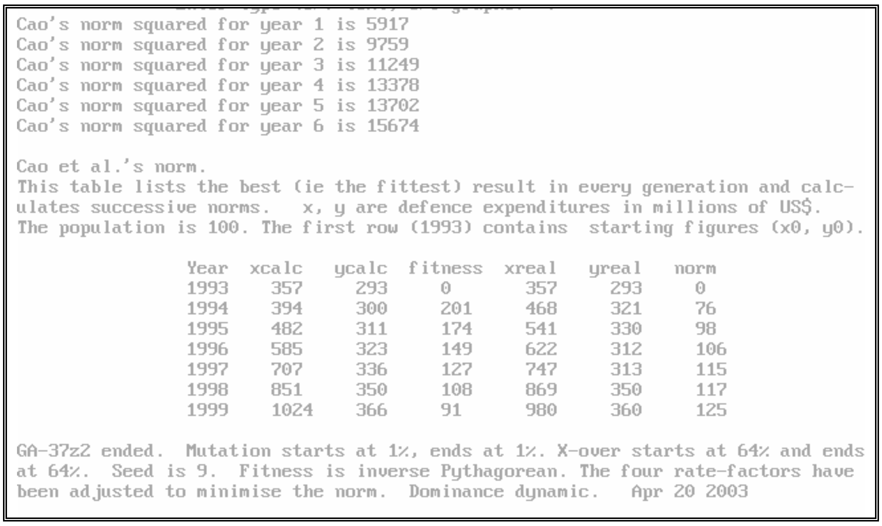


Figure 5.3 Creating the model. Values of x_t and y_t against years



```
Cao's norm squared for year 1 is 5917
Cao's norm squared for year 2 is 9759
Cao's norm squared for year 3 is 11249
Cao's norm squared for year 4 is 13378
Cao's norm squared for year 5 is 13702
Cao's norm squared for year 6 is 15674

Cao et al.'s norm.
This table lists the best (ie the fittest) result in every generation and calc-
ulates successive norms.   x, y are defence expenditures in millions of US$.
The population is 100. The first row (1993) contains  starting figures (x0, y0).

                Year  xcalc   ycalc  fitness  xreal   yreal   norm
                1993   357     293     0       357     293     0
                1994   394     300     201     468     321     76
                1995   482     311     174     541     330     98
                1996   585     323     149     622     312     106
                1997   707     336     127     747     313     115
                1998   851     350     108     869     350     117
                1999   1024    366     91      980     360     125

GA-37z2 ended.  Mutation starts at 1%, ends at 1%. X-over starts at 64% and ends
at 64%.  Seed is 9.  Fitness is inverse Pythagorean. The four rate-factors have
been adjusted to minimise the norm.  Dominance dynamic.   Apr 20 2003
```

Figure 5.4 GA-13, Option 7. This shows (x_{calc} , y_{calc}) alongside the corresponding IMF values (x_{real} , y_{real}) *after* the reduction of Cao *et al.*'s. norm

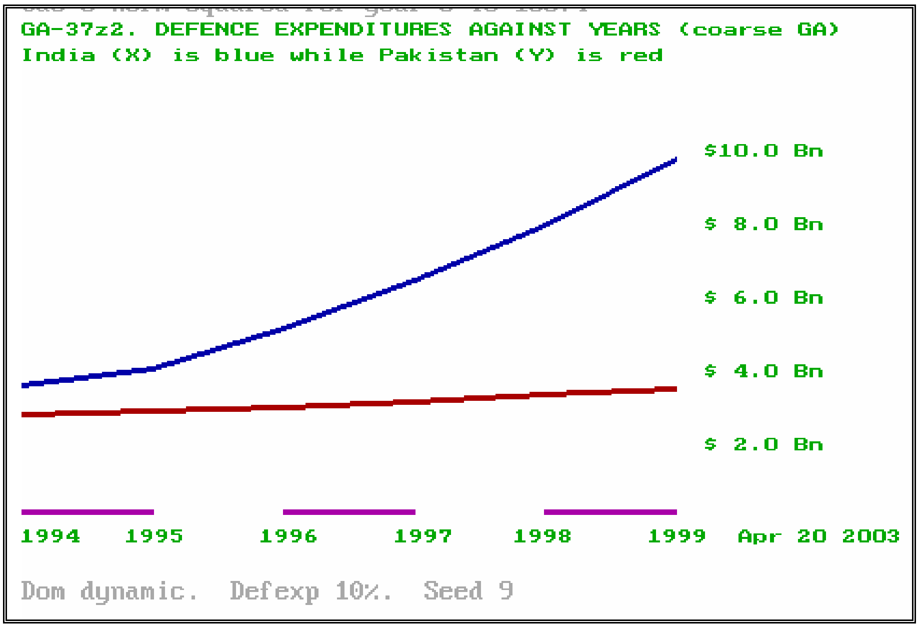


Figure 5. 5 *The model.* GA-13, using Eqs(5.4). The output of the modelling GA mimics Figure 5.2

In the case of India/Pakistan (1993 to 1999), the rate-factors $k_{1,1}$, $k_{2,2}$, $k_{1,2}$ and $k_{2,1}$ finally carried forward to the prediction GA were 17, 5, 7 and 1, respectively.

It is emphasised that getting good approximations to the rate-factors is crucial. In Chapter 6, not only do we employ the Cao *et al.* norm, but we carry out other tests to validate the rate-factors which are to be carried forward to the prediction GA (see Section 6.3.2)..

5.5 Predicting Instabilities in the Arms Race Equations

The modelling GA, GA-13, uses too few genes and too many set values to call it a proper GA, although it establishes workable initial values for the rate-factors $k_{i,j}$ which we can use.

In contrast, GA-6 was designed for prediction purposes and uses the following modification of Eqs(5.3) as evaluation equations:

$$\begin{aligned} x_{t+1} &= x_t + (x_t\ k_{1,1} / 100\ x_s / 100 + y_t\, k_{1,2} / 100)\big((x_m - x_t) / 85\big) \quad . \\ y_{t+1} &= y_t + (y_t\ k_{2,2} / 100\ y_s / 100 + x_t\, k_{2,1} / 100)\big((y_m - y_t) / 85\big) \quad . \end{aligned} \tag{5.7}$$

Eqs(5.7) broadly resemble Eqs(5.3). However, in the chromosomes of GA-6 all the rate-factors $k_{i,j}$ are defined as integers lying between 0 and 99 so we have to divide by 100. (A GA treats all its genes equally, no matter what scaling factors are put on them [Mitchell, 1996].) Moreover, the term $k_{1,1}\ (x_s - x_t)$ in Eqs(5.3) is now expressed as $k_{1,1}\ (x_s /100)\ x_t$. Thus the *intrinsic expenditure* (x_s) introduces two more genes into the GA, and so does the *budgetary defence expenditure* (x_m) which appears in Eqs (5.7) as (x_m - x_t), in place of the earlier $GDP_X/1000$.

Returning to our running example, at Generation 0 we input rate-factors $k_{1,1} = 17$, $k_{2,2} = 5$, $k_{1,2} = 7$, $k_{2,1} = 1$, output from the modelling GA. Forty generations later these had become $k_{1,1} = 27$, $k_{2,2} = 5$, $k_{1,2} = 1$ and $k_{2,1} = 14$ confirming that, like everything in Eqs(5.7), the rate-factors are time-dependent. Eqs(5.7) worked well, and provided convergent solutions; see Figures 5.8, 5.9, 5.11, and 5.12 , and contrast them with Figure 5.10, where an *extra* 0.1% average defexp% has made the system unstable.

In order to predict instabilities in the arms race environment, we first need to be satisfied that our technique would have predicted the conflicts of the past. This we will do next.

5.5.1 The Prediction GA

The GA used, GA-6, is typical of those described in Chapter 4. Ten genes representing the parameters $k_{1,1}$, $k_{2,2}$, $k_{1,2}$, $k_{2,1}$, x_s, y_s, x_m, y_m, x_t and y_t from Eqs(5.7) were set up to appear as one integer (effectively 24 elements long) in a chromosome having a search space of 10^{24}. The population was 100. Each chromosome is a point in the search space of candidate solutions, so the alleles of each of its genes could be used in the evaluation equations Eqs(5.7). The prediction GA was run until the results were stable, and (in our running example) this varied from twenty to forty generations.

In Chapter 3 we drew a parallel between canard explosion theory (where changes in the direction of curvature of the limit cycle can be used to predict instabilities) and the Richardson arms race equations. It was shown in Section 3.7 that the direction of curvature of the limit cycle is given, under certain assumptions, by the sign of $a - c$. If $a > c$ the limit cycle revolves counter-clockwise and, if $a < c$, clockwise.

These changes of sign can now be used to predict instabilities. To this end, it was decided to use the approximations for a and c, derived in Eqs(3.16), and to modify GA-6 to calculate

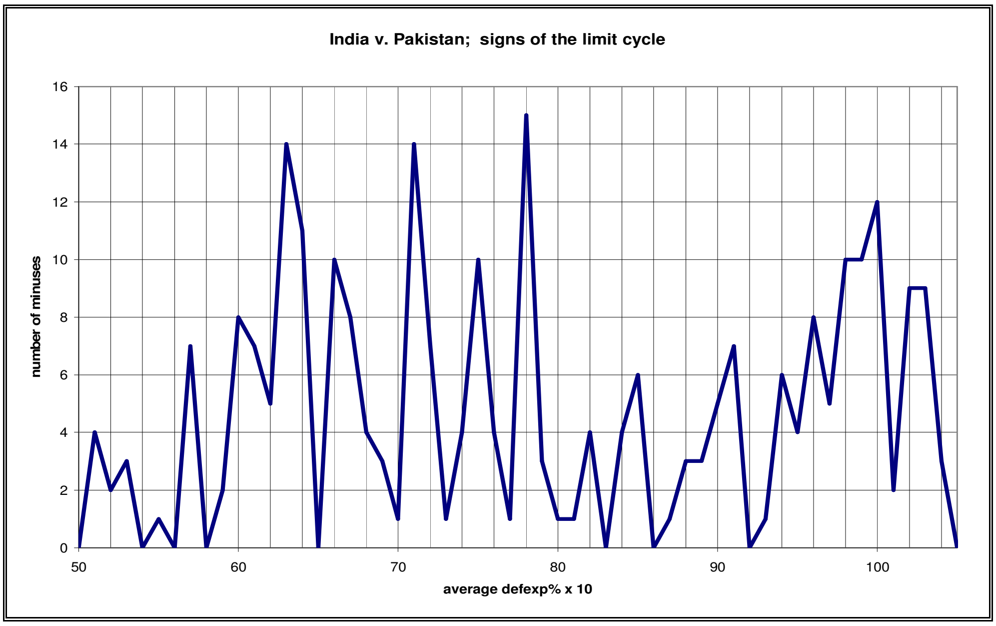
$$a \approx k_{2,1}\, y_m \quad \text{and} \quad c \approx k_{1,2}\, x_m \qquad .$$

and output $a - c$ for the fifteen best chromosomes shown. The resulting GA is called GA-5.

The significance of the difference $a - c$ lies in the number of changes of its sign at each iteration step of average defexp%, mainly in steps of 0.1%. In this context, Figure 5.6 shows the *number of minus signs in* $a - c$ (maximum 15) at increasing values of average *defexp%* (in steps of 0.1%). This is a *prediction plot* (see Section 5.3) It will be shown later by running GA-6 in graphical mode (Option 8) that instabilities occur at 6.5 % (see Figure 5.10) and at 7.9%**.**

Figure 5.6 shows that each of these instabilities is prefaced by a sudden plunge from a high number of minus signs to a small number of minus signs. Analysing this and other charts, if *a* overtakes *c* very rapidly (normally denoted by a change of ten or more signs from *minus to plus* in one iteration) then instability is imminent. (In chaos theory [Ott, 1994], similar sign changes occur at a *Hopf bifurcation* when, at the moment of transition from a laminar to a chaotic state, the complex-conjugate eigenvalues of the limit cycle change sign from minus to plus.) It has been observed empirically that a lesser number of sign changes than ten, say eight, does not have this effect, and that unless the number of minus signs keeps at or above ten there is, therefore, little likelihood of instability.

In our running example, major sign changes occur (Figure 5.6) at an average defexp% of 6.5% , 7.9% and 10.1%. An examination of Figure 5.7 shows that these averages occurred in the spring and autumn respectively of 1997, and in the summer of 1998. The Europa World Yearbook [2001] mentions (pp 1935 and 3068) that in March 1997 an Indian rocket attack on a mosque at Azad in Kashmir killed forty Pakistanis. In September that same year, there were large-scale artillery exchanges in Kashmir and Jammu during which the Indian and Pakistani prime ministers personally used a 'hotline' in a desperate (but successful) attempt to prevent the situation from escalating into a major war**.** Both India and Pakistan tested nuclear weapons in the summer of 1998 and feelings ran high.



indices **1** ***** **2** **3**

Figure 5.6 *Prediction plot*. Predicting when instabilities might occur. On this graph instabilities are predicted at an *average* defexp% of 6.5% (1), 7.9% (2), and 10.1% (3). The asterisk marks a double drop (14→7→1)

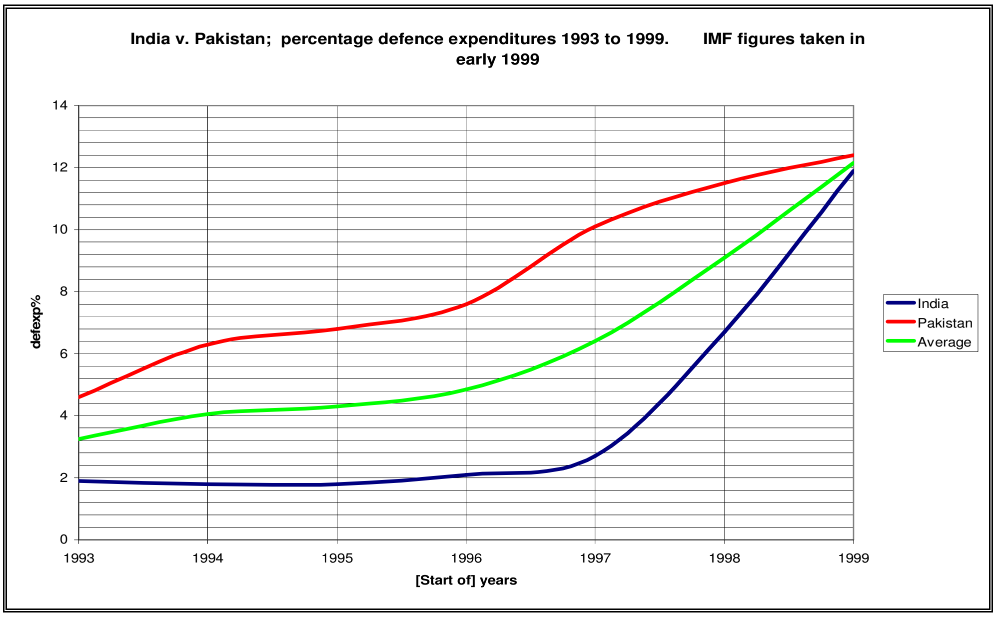


Figure 5.7 *defexp% plot.* Note the *average* defexp% .
Plot originally drawn in early 1999. Note that this is based on data earlier than that of Appendix 6A.2 and is significantly different from the 1955 to 2000 plot (Figure 6.10) which was drawn in late 2001. For an explanation of the difference, see (6) of the Preface to Appendix 6A

Figure 5.6 demonstrates that a prediction plot can (within reason) be extrapolated. It is not so easy, however, to extrapolate Figure 5.7, i.e. to estimate the future movements of the average defexp%; attempts to do so were abandoned. Herein lies the problem of prediction.

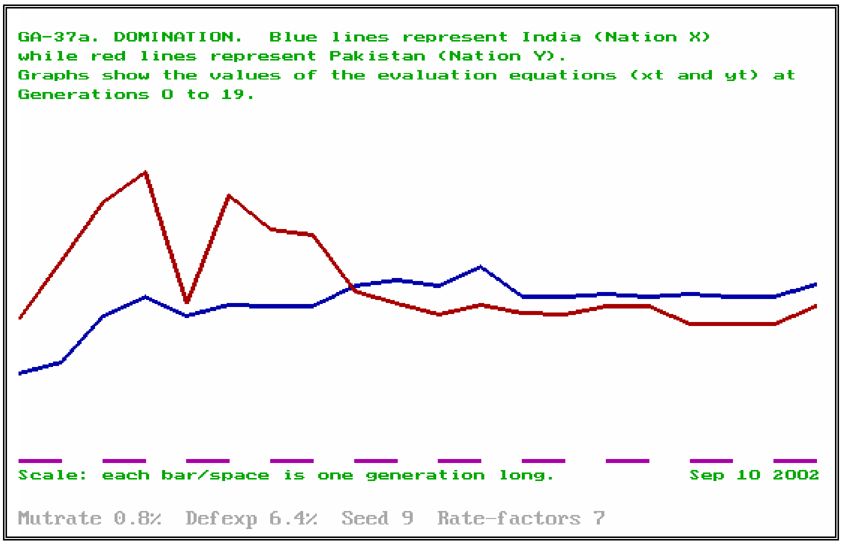
Table 5.2 Truth-table: India/Pakistan instabilities

		occurred?	
		yes	**no**
predicted?	**yes**	3	0
	no	0	

Table 5.2 shows an instability truth table; three conflicts predicted, three occurred historically. We have, therefore, shown that in principle it *is* possible to predict known conflicts by this method. A much stronger validation of this statement is presented in Chapter 6.

5.6 How Minute Changes in Average Defexp% Alter Stability

Using GA-6, we plotted the output of the evaluation equations (Eqs(5.7)), x_t and y_t, against generations for set values of defexp% *increasing by only* 0.1% at each figure. Results are exhibited in Figures 5.8 to 5.12. (The legend *Rate-factors* 7, shown on some of the screenprints, means that the rate-factors are sourced from a constrained RNG. Correspondingly, *Rate-factors* 8 means an IMF source).

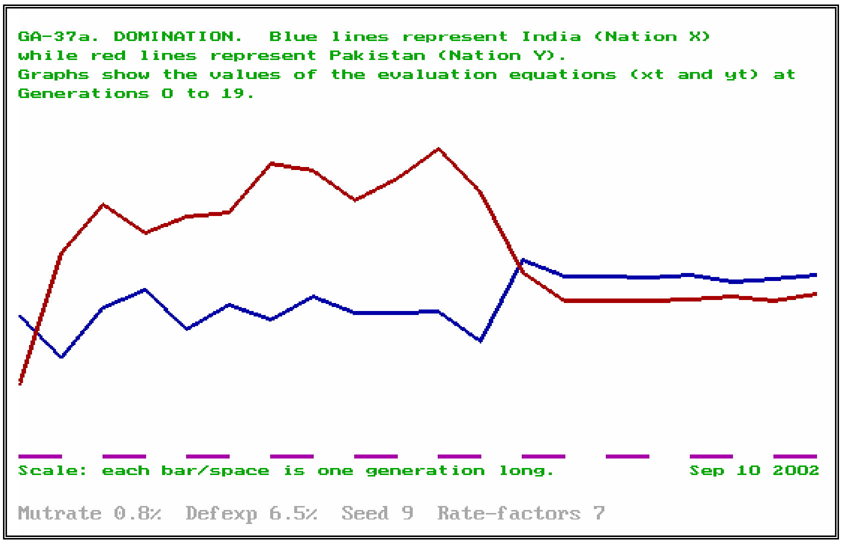


GA-37a. DOMINATION. Blue lines represent India (Nation X)
while red lines represent Pakistan (Nation Y).
Graphs show the values of the evaluation equations (xt and yt) at
Generations 0 to 19.

Scale: each bar/space is one generation long. Sep 10 2002

Mutrate 0.8% Defexp 6.4% Seed 9 Rate-factors 7

Figure 5.8 GA-6. Average defexp% 6.4 %

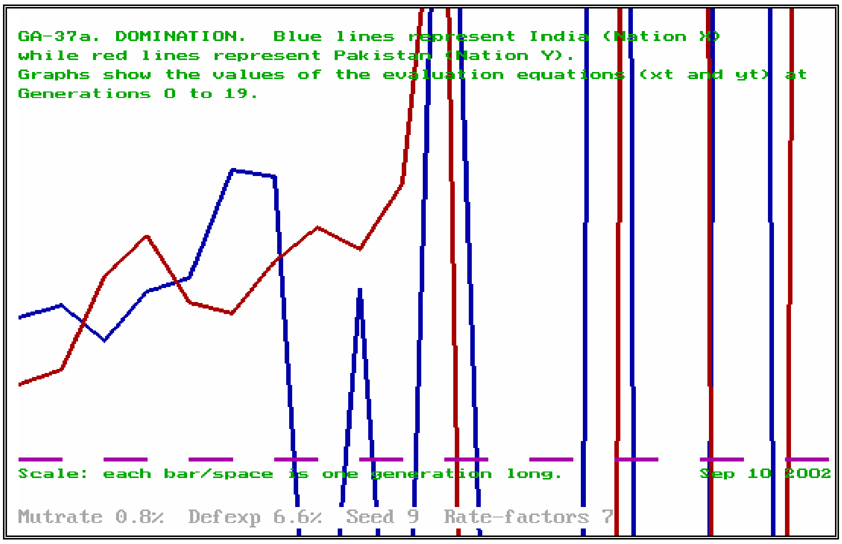


GA-37a. DOMINATION. Blue lines represent India (Nation X)
while red lines represent Pakistan (Nation Y).
Graphs show the values of the evaluation equations (xt and yt) at
Generations 0 to 19.

Scale: each bar/space is one generation long. Sep 10 2002

Mutrate 0.8% Defexp 6.5% Seed 9 Rate-factors 7

Figure 5.9 GA-6. Average defexp% 6.5 %

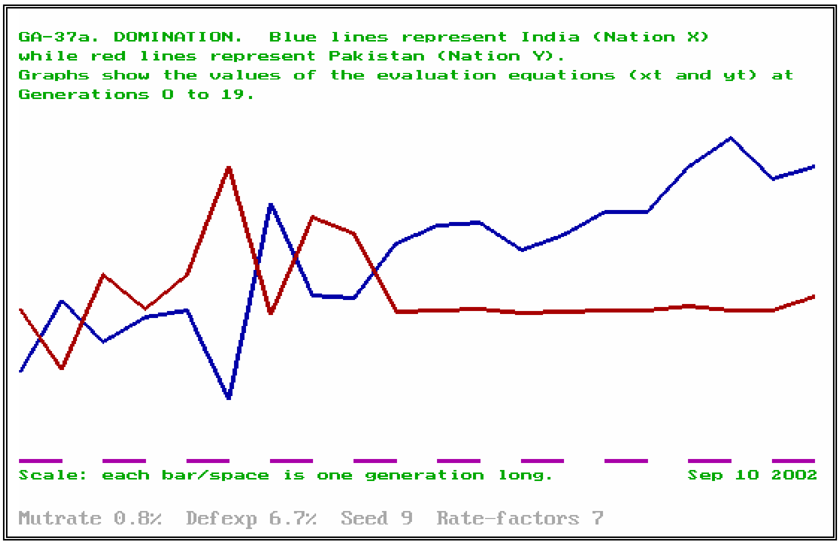


GA-37a. DOMINATION. Blue lines represent India (Nation X) while red lines represent Pakistan (Nation Y). Graphs show the values of the evaluation equations (xt and yt) at Generations 0 to 19.

Scale: each bar/space is one generation long. Sep 10 2002

Mutrate 0.8% Defexp 6.6% Seed 9 Rate-factors 7

Figure 5.10 GA-6. Average defexp% 6.6 %

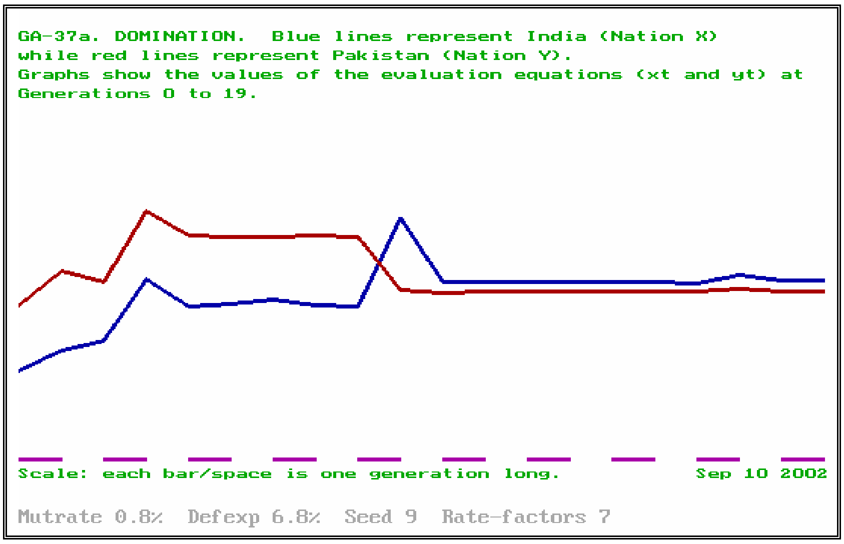


GA-37a. DOMINATION. Blue lines represent India (Nation X) while red lines represent Pakistan (Nation Y). Graphs show the values of the evaluation equations (xt and yt) at Generations 0 to 19.

Scale: each bar/space is one generation long. Sep 10 2002

Mutrate 0.8% Defexp 6.7% Seed 9 Rate-factors 7

Figure 5.11 GA-6. Average defexp% 6.7 %



GA-37a. DOMINATION. Blue lines represent India (Nation X) while red lines represent Pakistan (Nation Y). Graphs show the values of the evaluation equations (xt and yt) at Generations 0 to 19.

Scale: each bar/space is one generation long. Sep 10 2002

Mutrate 0.8% Defexp 6.8% Seed 9 Rate-factors 7

Figure 5.12 GA-6. Average defexp% 6.8 %

Comment. The average defexp% increases by only 0.1% over each of Figures 5.8 to 5.12. Figure 5.8 is stable; Figure 5.9 is excited; Figure 5.10 is unstable; Figure 5.11 is calming down, while Figure 5.12 is quiescent again. This shows how sensitive the Richardson equations can be to minute changes in average defexp%. It also shows how the prediction plot at Figure 5.6 reflects the levels of activity to be expected when unstable points are encountered.

Stability Criteria We next verify that Richardson's stability criteria are borne out in practice. Using the a, c, b, f nomenclature of Eqs(5.1), evaluation Eqs(5.4), the RNG-derived values of $k_{i,j}$ used in GA-6 to generate Figures 5.9 and 5.10 ($k_{1,1} = 40$, $k_{2,2} = 5$, $k_{1,2} = 6$, $k_{2,1} = 2$), and setting the defexp%, g, to 6.5 %, we obtain

$$a = k_{2,1}\, g = 13.0; \qquad c = k_{1,2}\, g = 39.0; \qquad \text{so } ac = 507$$

$$b = 1 - (0.22\, k_{2,2}\, g) = -6.5; \quad f = 1 - (0.32\, k_{1,1}\, g) = -83.5; \quad \text{so } bf = 542$$

Thus $ac < bf$ and the situation is stable, see Figure 5.9.

Correspondingly, for defexp% set to 6.6 % we obtain

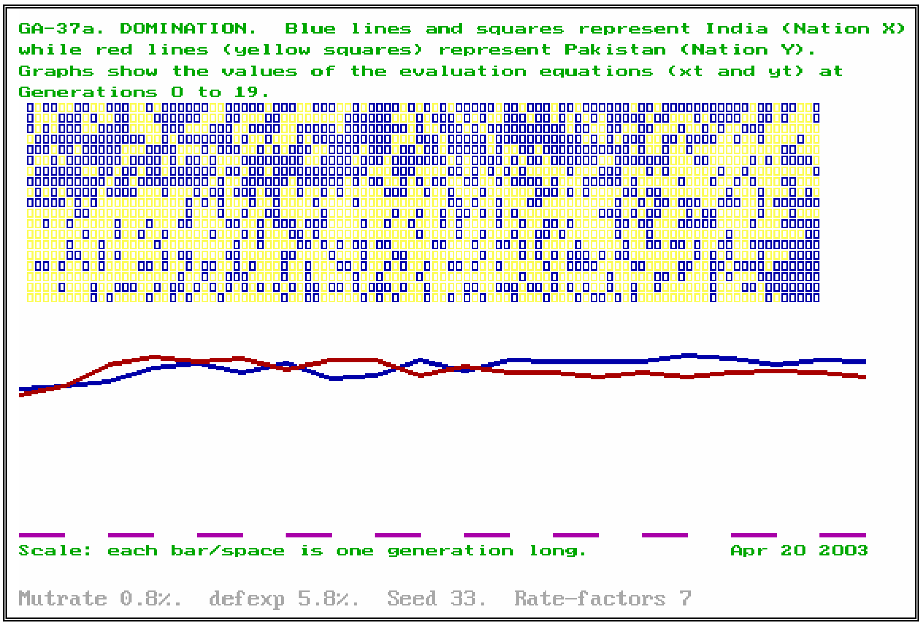
$$a = k_{2,1}\, g = 13.2; \qquad c = k_{1,2}\, g = 39.6; \qquad \text{so } ac = 523$$

$$b = 1 - (0.22\, k_{2,2}\, g) = -6.2; \quad f = 1 - (0.32\, k_{1,1}\, g) = -83.4; \quad \text{so } bf = 517$$

Hence increasing the average defexp% by 0.1% from 6.5% to 6.6% results in $ac > bf$, and the situation becomes unstable as Figure 5.10 demonstrates. *Hence Richardson's stability criterion does work for the India/Pakistan arms race in this instance.*

5.7 Dominance Diagrams and Fractal Basins

It is sometimes of interest to know which side, India or Pakistan, is dominant at any one time. A *dominance diagram* for our running example is shown in Figure 5.13. For each iteration of the population of 100 pairs of evaluation equations run for 20 generations, a small square is drawn. The colour of this square indicates whether India (dark blue) or Pakistan (yellow) is dominant (i.e. whether $x_t > y_t$ or $y_t > x_t$) at the end of each of the 2000 iterations. The two curves, below the coloured matrix, portray the situation at the extreme right-hand column of the matrix. Recall that the dominance diagram is Option 9 of every prediction GA.



GA-37a. DOMINATION. Blue lines and squares represent India (Nation X) while red lines (yellow squares) represent Pakistan (Nation Y). Graphs show the values of the evaluation equations (xt and yt) at Generations 0 to 19.

Scale: each bar/space is one generation long. Apr 20 2003

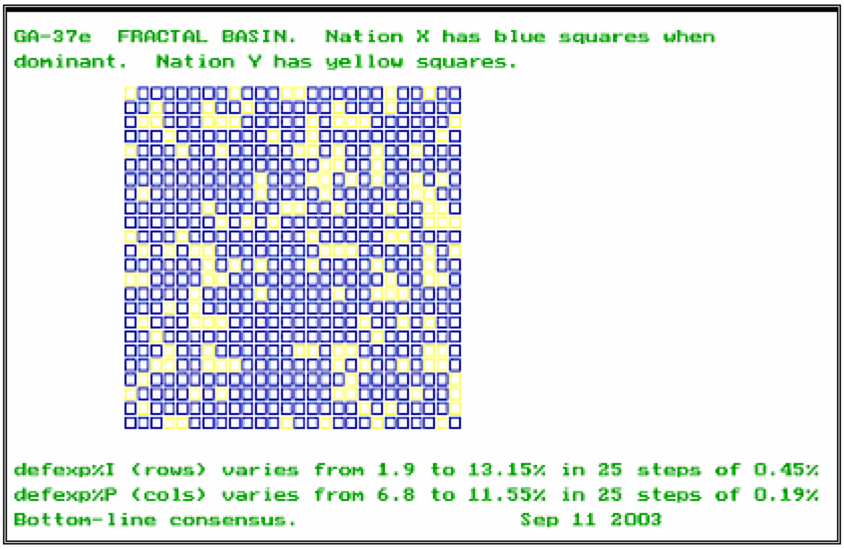
Mutrate 0.8%. defexp 5.8%. Seed 33. Rate-factors 7

Figure 5.13 *A dominance diagram.* GA-6. Seed 33. Average defexp% 5.8

In Section 4.9 we explained how we were concerned about domination changes, for in these changes lie the seeds of chaos and instability, and there is evidence that chaos is present in some of our GAs (viz. GA-4, GA-10 and GA-11, see Section 4.10). We therefore set out to create a series of *fractal basins*, the behaviour of which under chaotic conditions is well catalogued [Ott, 1994].

GA-7, a new algorithm, puts GA-6 (the prediction GA) inside two additional *for-loops*, so that it is possible to examine the behaviour of the GA as any two of its parameters (such as average *defexp%* and x_0, or average *defexp%* and any of the rate-factors $k_{i,j}$, or India's defexp% and Pakistan's defexp%) are varied over the whole of their likely ranges. The output is a 25 ×25 square (effectively an aggregation of 625 separate dominance diagrams); see, for example, Figure 5.14.

Figure 5.14 should be read as a series of rows rather than as a two-dimensional matrix. The rows, representing India's defexp%, vary from 1.9% to 13.15% in 25 steps. The columns, Pakistan's defexp%, vary from 6.8% to 11.55%. The figures are based on Appendix 6A.2.



```
GA-37e  FRACTAL BASIN.  Nation X has blue squares when
dominant.  Nation Y has yellow squares.
```

```
defexp%I (rows) varies from 1.9 to 13.15% in 25 steps of 0.45%
defexp%P (cols) varies from 6.8 to 11.55% in 25 steps of 0.19%
Bottom-line consensus.                Sep 11 2003
```

Figure 5.14 *A fractal basin*. GA-7. India's defexp% against Pakistan's defexp%. Seed 9

We had been led [Forrest and Mayer-Kress, 1991] to expect 'walls', i.e. there would be large well-defined, contiguous and clearly-separated areas of blue and yellow (see Figures 4.5 to 4.9) especially as the seed chosen, 9, is known to give rise to instability at an average defexp% of 6.6%, see Figure 5.10. Walls never did occur in the case of India/Pakistan, for the reason given in Section 4.11, namely that chaos is unlikely in a two-nation system. Despite varying a large number of parameters (including all the rate-factors $k_{i,j}$ in turn) the India/Pakistan fractal basin was always 'patchy', as if there were a lot of noise. It only then became apparent that the 'patchiness' was not (or only partly) due to noise, but to definite regions of instability some of which were quite local. We have, therefore, stable states lying either side of unstable ones, the phenomenon known to chemists as a precursor of a *canard explosion.*

According to this analysis, there are no general areas of stability in the India/Pakistan conflict; the system lurches from one instability to another with periods of respite in between. One cannot therefore say with any certainty that war will be averted if, say, India's defexp% drops below some fixed percentage, or if rate-factor $k_{2,1}$ is more than some stated figure, or if Pakistan's defence budget exceeds some declared sum, a situation best described as *volatile*.

5.8 Alternative Approaches

So far we have solved Mayer-Kress's discretised two-nation variant of the Richardson arms race equations by using GA-6, a conventional GA.

Figure 5.15 shows values of x_t and y_t in Option 8 of GA-6, a normal GA, using a seed of 8.

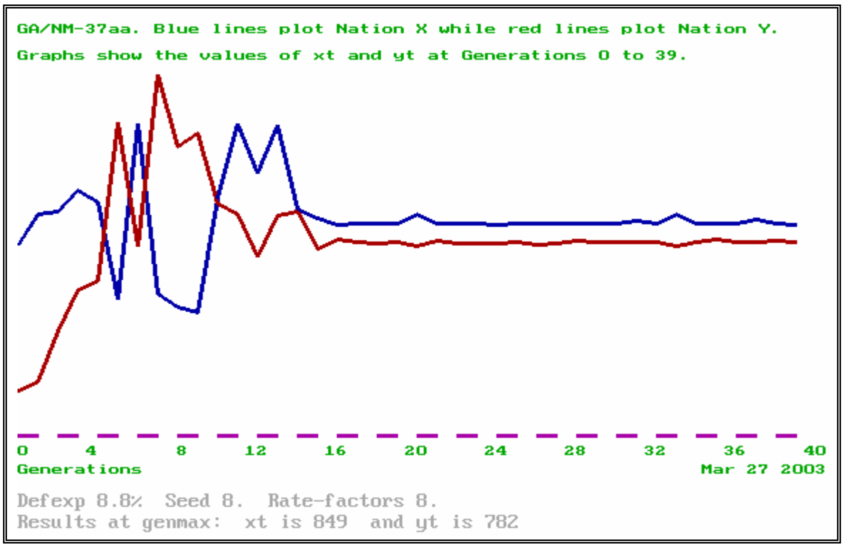


Figure 5.15 GA-6. Values of x_t and $y_{t.}$. Convergence occurs at 849 and 782

We will now look for alternative solutions to Eqs(5.3) that do *not* use a GA. For our first attempt we took GA-6, copied it and renamed the copy GA-5, and then systematically and progressively removed from GA-5 all traces of GA activity (i.e. selection, mutation and crossover). The revised algorithm was called NM-5. We then compared the behaviour of GA-5 with that of NM-5. Results of NM-5 were interesting in that there was some unexpected convergence, but evaluations of $a - c$ failed to generate the sign changes essential for our prediction method. It was not, therefore, possible to create a *prediction plot*. This was unacceptable in the context of the thesis, so the attempt was abandoned.

A second non-GA approach, again in effect stripping GA-5 of its GA characteristics and then using Eqs(5.3) to compute x_{t+1} from x_t, and y_{t+1} from y_t by varying *only the four rate-factors* $k_{i,j}$, encountered the same problems with respect to the sign changes of a - c, and suffered a similar fate. (This second approach, however, was the pre-cursor of the validation method to be used throughout Chapter 6).

5.8.1 A Monte Carlo Treatment

For our third non-GA approach, we converted NM-5 into a true Monte Carlo operation, calling the resulting algorithm MC-5. Monte Carlo methods rely on the deliberate use of random numbers in calculations that have the structure of a *stochastic process* [Kalos and Whitlock, 1986]. In practice, Monte Carlo methods involve the evaluation of multiple integrals. The central idea of Monte Carlo is that an integral may be represented by a sum. For example,

$$\int_{-\infty}^{\infty} g(x)f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} g(x_i) , \qquad .$$

where we draw a series of random variables x_i from a distribution with density function $f(x)$, and evaluate $g(x)$ for each x_i [Kalos and Whitlock, 1986]. The arithmetic mean of all N values of $g(x)$ is an approximation for the integral, and the variance of this approximation decreases as the number of terms increases.

Section 4.5 explained how Eqs(5.7) were run 100 times (one for each population member), evaluating x_{t+1} and y_{t+1} each time, and obtaining slightly different results each time. In GA-5, for each generation the 100 results were sorted by order of fitness, and the best results were filed in the Matrix *result,* see Section 4.3. In our Monte Carlo treatment, however, we do not *sort* the 100 results to find the *fittest*; instead, we simply use their *average.* The *result* matrix, therefore, contains the *average* for N = 100 at each generation. Figure 5.16 shows the output of MC-5 pictorially.

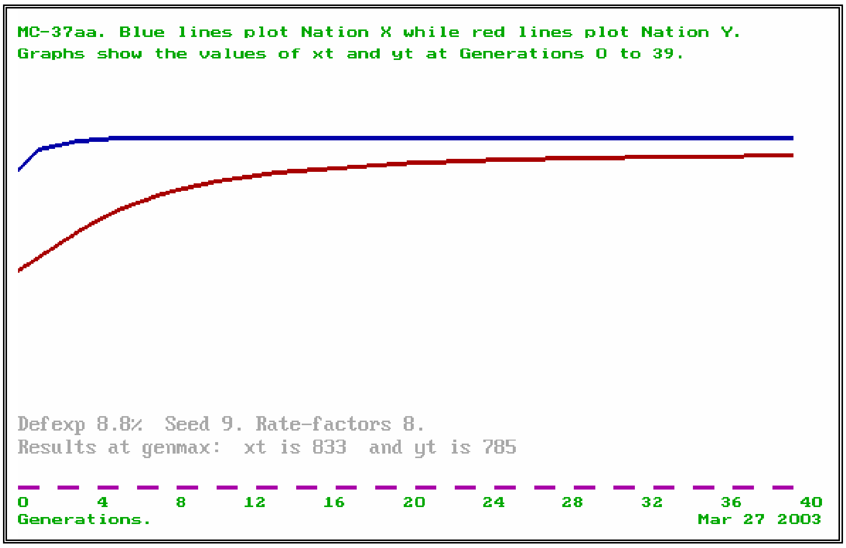


Figure 5.16 MC-5. Note how x_t and y_t converge on 833 and 785, respectively

After some teething problems, the results of several Monte Carlo trials were very encouraging. In every case, after about Generation 15, the *average* x_t and y_t , taken over N = 100, were within 1.8% and 0.4%, respectively, of the corresponding *fittest* results at each generation of GA-6, as a comparative glance at Figures 5.16 and 5.15 demonstrates.

Despite the encouraging appearance of Figure 5.16, the use of Monte Carlo methods to predict instabilities in arms races is inadvisable. Evaluations of $a - c$ failed to generate the sign changes essential for our prediction method. It was stated earlier that mutation is the principal source of these sign changes, without which we cannot use Peng *et al.*'s technique, our vehicle for predicting instabilities.

5.8.2 A Simulated Annealing Approach

Kirkpatrick, Gellat and Vecchi [1983] proposed an algorithm which is based on the analogy between the annealing of solids and the task of solving combinatorial optimisation problems [Pham and Karaboga, 2001]. The algorithm consists of a sequence of iterations. After initiation, each iteration randomly changes the current solution to yield a new solution. The principal mechanism for change (in our case) is *mutation*. Once a new solution is created, the corresponding change in the cost function (in our case the *fitness*) is calculated to determine whether or not the new solution is fitter than the current one. If it is, namely

$$\Delta E = \mathit{ffn}(\mathit{new\ solution}) - \mathit{ffn}(\mathit{current\ solution}) < 0 \,, \qquad .$$

then the new solution replaces the current one (which is discarded). Otherwise, the new solution may possibly be acceptable according to *Metropolis's criterion* which is based on the Boltzmann probability

$$P(E) \propto e^{-E/kT} \qquad .$$

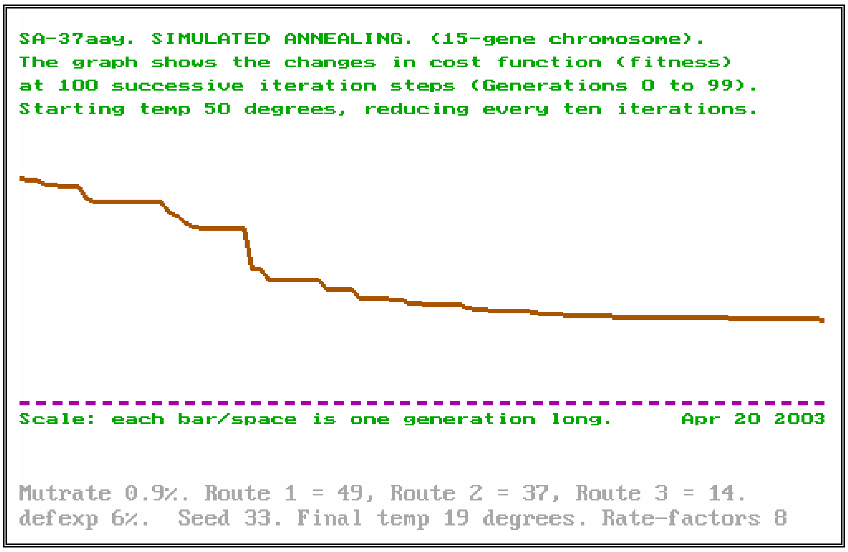
where *P*(*E)* is the probability that the system is in a state of energy *E, T* is the *temperature*, and *k* is Boltzmann's constant. Metropolis says, in our terms, that if $\Delta E \geq 0$, then a random number δ in [0,1] is generated from a uniform distribution and if

$$\delta \leq e^{\Delta E/T} \qquad .$$

then the new solution is allowed to replace the current solution. If not, the current solution stands.

It appears to be normal practice to take ten iterations at each temperature (T_i) before reducing the temperature in geometrical progression by some expression such as $T_{i+1} = cT_i$, where *c* is typically 0.9. Typical temperatures start at 100° (a purely notional figure, for it has no units) and reduce until a plot of *fitness* shows that asymptotic convergence has occurred.

In Section 7.7 we develop an *evolution strategy* [Schwefel, 1995] (ES) to handle nuclear deterrence, and the similarity between an ES and a simulated annealing (SA) algorithm is striking. The only significant difference is that, in an ES, if $\Delta E \geq 0$, then the new solution is discarded immediately so there is no probabilistic 'second chance'. We actually took our existing ES algorithm, ES-1, and modified it to assume an SA role (renaming it SA-5) without too much difficulty. In effect, we initiate a GA with *a population of two* using Eqs(5.3) as the evaluation equations, and then utilise the inverse Pythagorean ffn (i.e. Eq(5.5)) to compare the two. Whereas in the ES the comparison would have been straightforward, the SA has the added complexities of the Metropolis criterion and of falling temperatures, making the SA approach more complex.

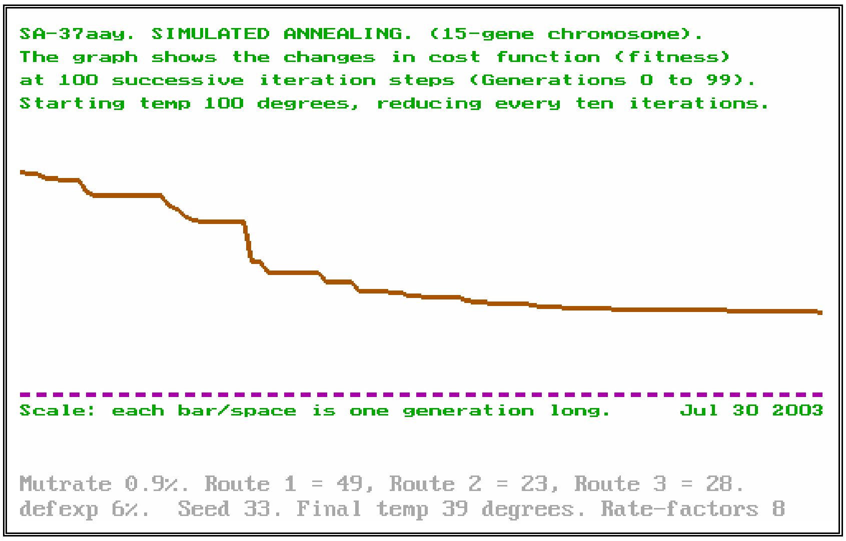


SA-37aay. SIMULATED ANNEALING. (15-gene chromosome).
The graph shows the changes in cost function (fitness)
at 100 successive iteration steps (Generations 0 to 99).
Starting temp 50 degrees, reducing every ten iterations.

Scale: each bar/space is one generation long. Apr 20 2003

Mutrate 0.9%. Route 1 = 49, Route 2 = 37, Route 3 = 14.
defexp 6%. Seed 33. Final temp 19 degrees. Rate-factors 8

Figure 5.17 SA-5. Simulated Annealing 1. Fitness against generations. Temperature starts at 50°, ends at 21°. Mutation rate 0.9%. Fitness ranges from 98 to 8



```
SA-37aay. SIMULATED ANNEALING. (15-gene chromosome).
The graph shows the changes in cost function (fitness)
at 100 successive iteration steps (Generations 0 to 99).
Starting temp 100 degrees, reducing every ten iterations.

Scale: each bar/space is one generation long.      Jul 30 2003

Mutrate 0.9%. Route 1 = 49, Route 2 = 23, Route 3 = 28.
defexp 6%.  Seed 33. Final temp 39 degrees. Rate-factors 8
```

Figure 5.18 SA-5. Simulated Annealing 2. Fitness against generations. Temperature starts at 100°, ends at 39°. Same mutation rate and fitness limits as Figure 5.18, but a different ‘cooling profile’[1]. Fitness ranges from 98

on rate of 0.9%, initial temperatures of 50° and 100°, and a defexp% of 6.0% we ran SA-5 for 100 iterations (generati1 1ons). Every ten iterations the temperature dropped by a factor of 0.9. The results can be seen in Figures 5.17 and 5.18 which plot fitness (in SA parlance *cost function*) against the number of iteration steps at the two different starting temperatures. For the two runs, we kept count of the number of times the three routes were taken, see Table 5.3.

Table 5.3 Simulated Annealing. The number of times (out of 100) that the three routes were used

Condition/Initial Temperature T_0	$T_0 = 50°$	$T_0 = 100°$
$\Delta E < 0$ Route 1	49	49
$\Delta E \geq 0$, but Metropolis’s criterion is met Route 2	37	23
$\Delta E \geq 0$, but Metropolis’s criterion is not met Route 3	14	28

Evaluations of *a* - *c* can only produce a single sign change since we can only consider one chromosome at a time. Thus SA cannot be used for our prediction method.

5.9 Comment on the Non-GA Methods

It is apparent that there *are* methods of solving Eqs(5.3) without resorting to GAs. The first three non-GA approaches, however, failed to produce the sign changes necessary for our prediction method. The same applies to simulated annealing. (Rudolph [1993] describes a method of using SA to emulate an EA, but this involves the use of massively parallel processors; we do not have access to such facilities.)

The strength of our approach, namely applying GAs to Richardson's arms race theory, lies in their ability to predict instabilities. (On the empirical evidence available, this is done reasonably well). The approximation for $a - c$ (namely $k_{2,1}\, y_m - k_{1,2}\, x_m$) developed in Eqs(3.16) is not conditional on our using a GA, but it does help to use a method which, first, widens the search space by deliberately introducing a lot of mutation and, second, effectively evaluates more than one individual (chromosome) at a time.

5.10 Conclusions

In this chapter we set out to explain the rationale underpinning our prediction of instabilities. We demonstrated this by developing some general arguments, and then illustrating their application in a snapshot (1993 to 1999) taken from the continuing arms race between India and Pakistan. We made use of two integer-valued GAs which, while related in appearance and operation, were dissimilar in function. The first (the *modelling GA*) generated a model designed to be faithful to the IMF data *and* to Richardson's equations. The modelling GA vindicated Richardson's theoretically-derived criteria for stability. The second (the *prediction GA*) was used to predict instabilities by looking for violent changes in the signs of a - c which in turn indicate changes in the direction of curvature of the limit cycle of the evaluation equations, a technique pioneered by Peng *et al.* [1991] for forecasting canard explosions.

The prediction GA found that, in the said snapshot between India and Pakistan , there were large areas of stability separated by quite local but violent pockets of instability. It would be premature to say at this stage that known and likely instabilities between India and Pakistan seem *always* to be prefaced by a sudden change in the direction of curvature of the limit cycle, but this will be demonstrated in Chapter 6. This change is very simple and (as has been said) is expressed by the sign of a - c. From an empirical viewpoint, if the sign of a - c of two-thirds of the GA's population on display (15) changes in successive iteration steps from minus to plus, *then it seems that an instability will occur at the next timeframe.* Such changes of sign are evident in the prediction plot in Figure 5.6 and relate directly to known historical events in the spring and autumn of 1997, and in the summer of 1998.

We closed the chapter by seeking alternative methods of solutions to the Mayer-Kress equations by means *other than a GA*. The first and second non-GA methods took the

---

[1] The rate of cooling is more marked in Figure 5.18 which starts at 100° than in Figure 5.17 which starts at 50°. It is tempting to draw comparisons with Newton's law of cooling which says that the *rate* of cooling by a hot object is proportional

modelling GA and removed all trace of GA activity, the third took a Monte Carlo approach, and the fourth used simulated annealing. All four provided solutions of some sort (the Monte Carlo solution was particularly accurate in its assessment of x_t and y_t), but they were all laborious, and none of them could identify a sufficient number of changes in the direction of curvature of the limit cycle. Collectively, therefore, they are of no use for prediction. In marked contrast, GAs have the advantage over non-GA methods in speed, flexibility and ease-of-use but, above all, their *implicit parallelism* allows them to evaluate a whole population's values of $a - c$ simultaneously, and not just one at a time. Hence the GA results can be used for predicting changes in the direction of curvature of the limit cycle and hence the likelihood of instability between nations engaged in an arms race.

---

to the *difference* in temperature between the object and its surroundings

CHAPTER 6

Validating the Richardson Model

6.1 Introduction

Chapter 3 established the connection that a combination of Richardson's arms race equations and Peng *et al.*'s canard explosion theory could predict the timing of the outbreak of hostilities between nations. Chapter 5 explained the general principles on which our prediction techniques are based and took a six-year window in the India/Pakistan conflict as a running example to illustrate the application of those principles. This chapter validates these ideas empirically by taking three recent long-running conflicts and demonstrating over significant periods of time that they work. The three chosen conflicts are:

- The dispute between Greece and Turkey over Cyprus and the Aegean, from 1955 to 2000. This consisted of two invasions, and the threat of a major war which was defused at the last moment.
- The dispute between India and Pakistan over Kashmir, from 1955 to 2000. This consisted of two major wars and the current unrest.
- The Middle East, from 1955 to 2000. This included three major wars, three minor wars and a number of not-insignificant skirmishes.

Appendix 6A lists detailed statistical information from IMF and UN handbooks and other authoritative sources. Its preface discusses the particular problems of extracting data for the arms race application.

The rest of this chapter briefly recapitulates the six steps of prediction (from Chapter 5) and then deals individually at length with the three conflicts listed above. Truth-tables for each conflict reveal that there are a small number of false-positives. These do not necessarily indicate a flaw in the prediction techniques, for it is known that some instabilities are defused by covert diplomatic action. Only one false-negative has been found. The chapter closes with a brief examination of the effects of introducing noise into the system.

6.2 Modus Operandi

Herein we recall the six steps leading to prediction. The first step is to take IMF statistics and plot defence expenditure against years. This is the *IMF plot;* see, for example, Figure 6.1. The second step is to develop a model designed to mimic the IMF plot while adhering to the principles of arms races embodied in the Richardson equations. The modelling GA generates curves which *mimic* those of the IMF plot. We call this the *model;* see, for example, Figure 6.2. Once the model has been improved by reducing Cao's norm, we transfer the rate-factors across to the prediction GA.

The third step is to determine changes in the direction of curvature of the limit cycle. This is manifested by marked changes in the sign of $a - c$, from minus to plus. We run the prediction GA and plot the *number of minuses* (maximum 15 which conveniently fills the screen) at each iteration step against increasing values of average defexp%. This is the *prediction plot;* see, for example, Figure 6.4. A sudden marked change (normally ten or more) from a high number of minuses to a low number of minuses in successive iteration steps indicates that the direction of curvature of the limit cycle has changed and that, from canard theory, an instability is imminent. Such changes (known from their appearance as *sharp drops*) not only occur whenever defexp% curves intersect, as might be expected; they also occur in other places.

The fourth step is to plot graphs of defexp% against time, including their average. This is the *defexp% plot;* see, for example, Figure 6.5. Richardson was always careful to work in terms of *overall* defence expenditures; thus, from our running example, India's could be four times that of Pakistan (reflecting the disparity in both their populations and their GDPs). In contrast, it now seems that conflict is likely when *percentage* defence expenditures touch or intersect. (Perhaps this is because percentage defence expenditures reflect a level of national and individual commitment). The fifth step is to use the defexp% plot to identify dates associated with the sharp drops shown on the prediction plot. The sixth and final step is to verify these dates against appropriate historical references.

Comments

We used both dominance diagrams (Section 4.9) and fractal basins (Section 4.10) to identify those regions where instability would be likely to occur. Furthermore we verified, for each of the three conflicts considered, that $Q\cos\varphi \ll a - c$ (see Eq(3.27)). In particular, in each instance, $88° < \varphi < 89.3°$.

Throughout the chapter:

- Modelling GAs (Figures 6.2, 6.7 and 6.12) are run for 45 generations (years) as we have data from 1955 to 2000. Prediction GAs (Figures 6.4, 6.9 and 6.14) are run for 40 generations, as this was found experimentally to be a workable minimum. Neither modelling GA nor prediction GA curves have been smoothed, but all other curves have.

- Everything is plotted at one-year intervals, with the exception of the validation diagrams (Figures 6.3, 6.8 and 6.13) which are plotted at five-year intervals. Data for the validation diagrams was worked on spreadsheets using Microsoft *Excel* as a scripting language.

6.3 Cyprus, Aegean: Greece versus Turkey, 1955 to 2000

Herein we examine the very long-running dispute between Greece and Turkey over Cyprus and the Aegean culminating in a serious instability in 1997 when the two nations very nearly went to war. We use the two-nation version of Mayer-Kress's [1989] discretised Richardson equations (Nation X is Greece, Nation Y is Turkey), namely

$$x_{t+1} = x_t + \big(k_{1,1}(x_s - x_t) + k_{1,2}\, y_t\big)(x_m - x_t) \quad .$$

$$y_{t+1} = y_t + \big(k_{2,2}(y_s - y_t) + k_{2,1}\, x_t\big)(y_m - y_t) \,. \qquad (6.1)$$

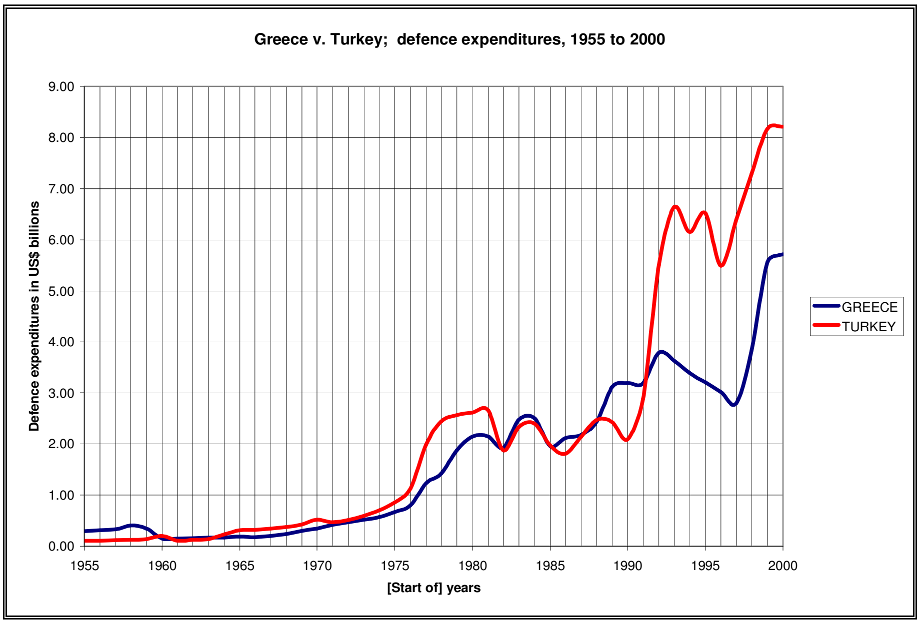


Figure 6.1 *IMF plot.* Data from Appendix 6A.1

6.3.1 The Model

Using a modelling GA, GA-14, initial curves were developed by manipulating the evaluation equations derived from Eqs(6.1) (see Section 5.4.2, steps 1-5), namely

$$x_{\,t+1} = x_t + (-0.14\, x_t\, k_{1,1}/50 + y_t\, k_{1,2}/50)\, G_X / 1740 \quad .$$

$$y_{\,t+1} = y_t + (-0.5\, y_t\, k_{2,2}/45 + x_t\, k_{2,1}/100)\, G_Y / 1988 \qquad (6.2)$$

yielding the *starter-values* $k_{1,1} = 51$, $k_{2,2} = 12$, $k_{1,2} = 32$ and $k_{2,1} = 6$ (hereafter referred to simply as 51, 12, 32, 6). The curves, i.e. Eqs(6.2), have yet to be refined.

6.3.2 Improving and Validating the Model

We *improve* the model by, first, employing an appropriate ffn and, second, by reducing Cao *et al.*'s norm. We then *validate* our techniques by comparing the values of x_t, y_t obtained from the evaluation equations of the *prediction GA* (Eqs(6.4)) against the corresponding IMF values.

(1) In order to achieve any sort of mimicry vis-à-vis Figure 6.1 we require an ffn which incorporates all the IMF data. To this end we employ the ffn

$$fitness = \sqrt{(x_{calc} - x_{real})^2 + (y_{calc} - y_{real})^2} \quad , \tag{6.3}$$

where x_{real}, y_{real} stand collectively for all the IMF values, and x_{calc}, y_{calc} stand collectively for the corresponding computed values.

(2) *Cao et al.'s norm.* GA-15 input x_{real} and y_{real} every year from 1955 to 2000 from IMF sources, and made use of the x and y already calculated by GA-14, see Section 5.4.3. By adjusting the four rate-factors (as explained in Section 5.4.2, steps 6-8) the norm was reduced from 243 to 175. After adjustment, the final values for the rate-factors were 27, 11, 30, 7.

Figure 6.2 shows the output of the modelling GA, GA-14, after reducing Cao *et al.*'s norm.

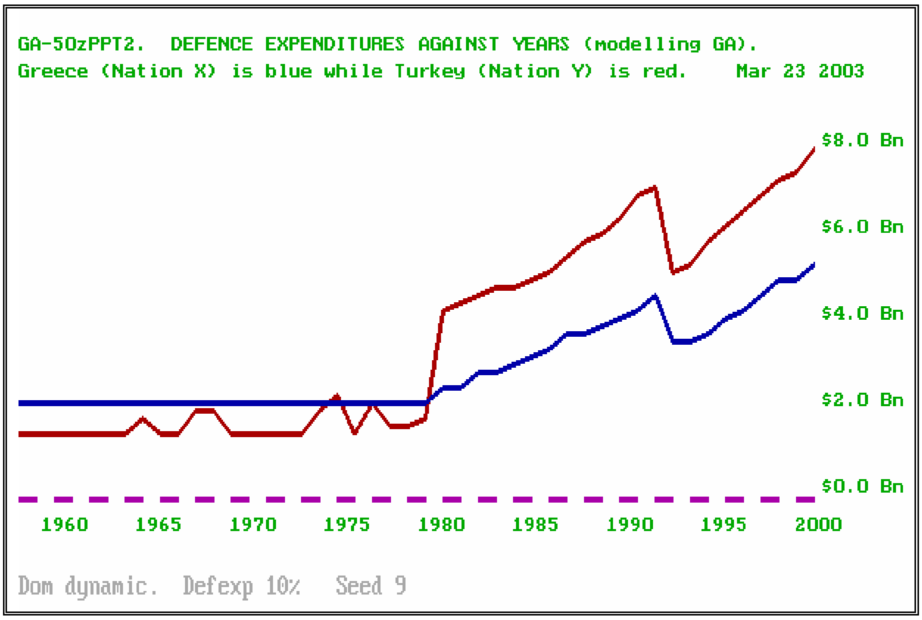


Figure 6.2 *The model.* GA-14, using Eqs(6.2). The curves follow the IMF plot (Figure 6.1) quite tightly

We now proceed to validate the model. Since we know all the IMF values of x_t, y_t, x_m and y_m (collectively designated x_{real} and y_{real}), we could use the evaluation equations of the prediction GA, Eqs(6.4), to obtain x_{t+1} and y_{t+1}, etc (collectively denoted by x_{calc} and y_{calc}) as a *straightforward arithmetical exercise* not involving GAs at all; x_{real} and y_{real} can then be compared with x_{calc} and y_{calc}. To initiate the calculation we use only $x_0(real)$ and $y_0(real)$. The calculated and the real curves are plotted alongside each other in Figure 6.3 and show quite a good match; it would seem that 27, 12, 30, 7 are appropriate values for the four rate-factors.

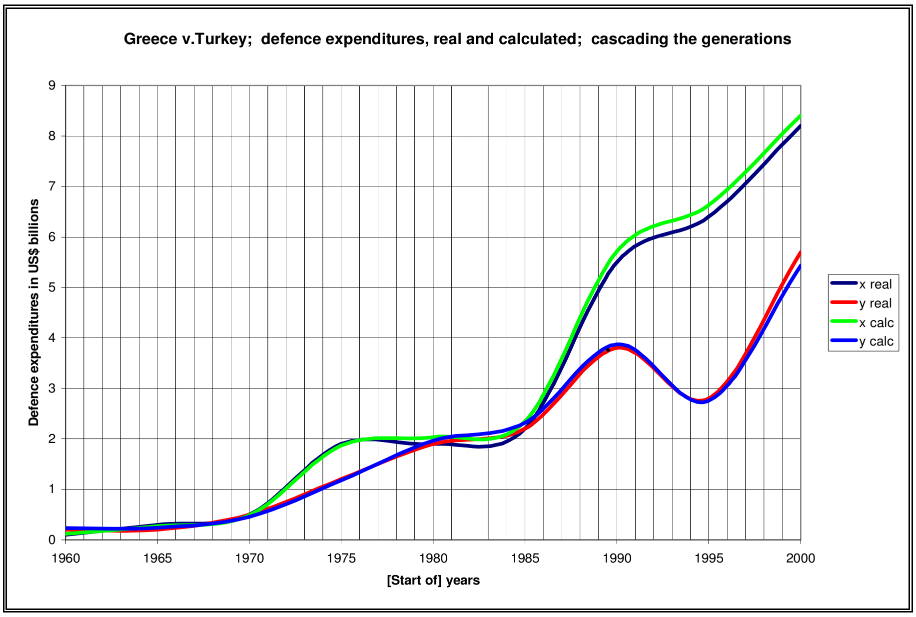


Figure 6.3 Comparing real with predicted values of x_t and y_t; using the prediction GA to verify that the appropriate rate-factors have been carried forward from the modelling GA[1]

6.3.3 The Prediction

Armed with refined starter-values for the rate-factors (27, 12, 30, 7), we developed a prediction GA, GA-16, for Greece/Turkey using the evaluation equations

$$x_{t+1} = x_t + \big((k_{1,1} / 38)\, x_t\, (x_s / 100) + (k_{1,2} / 100)\, y_t\big)\big((x_m - x_t) / 510\big) \quad .$$

$$y_{t+1} = y_t + \big((k_{2,2} / 50)\, y_t\, (y_s / 100) + (k_{2,1} / 55)\, x_t\big)\big((y_m - y_t) / 190\big). \qquad (6.4)$$

Eqs(6.4) are derived from Eqs(6.1), see Section 5.5. These equations are used to calculate the sign changes in the direction of curvature of the limit cycle, see Figure 6.4. By Generation 39 the rate-factors had become 23, 25, 32, 7.

Results. We list the average defexp% at which sharp drops occur in Figure 6.4, together with the corresponding dates extracted from the defexp% plot in Figure 6.5. Alongside are the years pertaining to relevant historical events. Sharp drops in Figure 6.4 occurred for the eight index numbers shown as follows:

---

[1] Since x_{calc} and y_{calc} were set to equal x_{real} and y_{real}, respectively, there is little point in showing the curves at Generation 0 (1955). Figures 6.4, 6.9 and 6.14 are all drawn to start at 1960

① 11→ 0 @ 3.5% which ties with 1997

② 11→ 1 @ 3.9% which ties with 1960 1963 1967 1997

③ 11→ 1 @ 4.4% which ties with 1958 1971 1974 1976 1992 1997

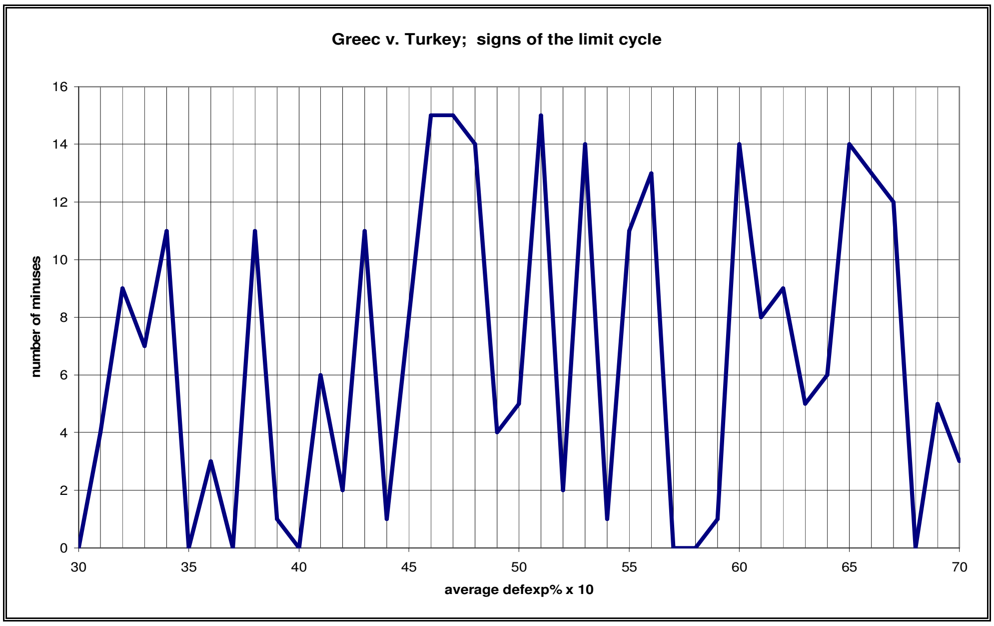
④ 14 → 4 @ 4.9% which ties with 1958 1976 1982 1988 1992

⑤ 15→ 2 @ 5.2% which ties with 1958 1976 1982 1988 1992

⑥ 14→ 1 @ 5.4% which ties with 1958 1976 1982 1988

⑦ 13→ 0 @ 5.7% which ties with 1958 1976 1982

⑧ 12→ 0 @ 6.8% which ties with 1956



index 1 2 3 4 5 6 7 8

Figure 6.4 *Prediction plot.* GA-16. The number of minuses of a - c in successive iteration steps. The index numbers show where sharp drops occur

Figure 6.5 *defexp% plot.* The defexp% of the two sides, and their average

Corresponding historical events were as follows:

a. **1956** Start of the Enosis movement for Union with Greece, led by Archbishop Makarios and General Grivas. At this stage, Enosis was aimed against the British. Makarios was deported in 1957

b. **1958** Makarios's proposal (from exile) that Cyprus should become an independent state seemed to please no-one, there was much inter-communal fighting, and both sides attacked British troops and their families. Grivas's deputy, Kyriakos Matsis, was killed

c. **1960** Turkish Government overthrown in May by a military coup led by General Gemal Gũrsel. Meanwhile Cyprus became independent in August under President Makarios. A general state of unrest existed in the South-Eastern Mediterranean

d. **1963** Serious inter-communal fighting following withdrawal of all Turkish-Cypriot participation in the central Cyprus Government. UNFICYP formed

e. **1967** Military coup in Greece. Colonel Papadopoulos is PM. Concern in Turkey

f. **1971** Grivas returns to Cyprus to turn Enosis against the Makarios Government

g. **1974** Grivas dies and Makarios purges all Enosis sympathisers. Greece sends a force into Cyprus. Makarios is deposed by the National Guard. Turkey also invades, and the island is currently divided

h. **1982** Rioting in Turkey over the powers of President Evren. Greek proposal that Greek and Turkish troops be withdrawn from Cyprus caused inter-communal tensions

i. **1992** Failure of (first round) UN talks to settle the Greek/Turkish differences over Cyprus

j. **1994** Failure of (second round) UN talks advocating a single nationality for all Cypriots

k. **1996** Greece and Turkey on the "verge of battle" with sizeable naval and air forces. The dispute was over the ownership of Imia, an uninhabited little island off the coast of Turkey, legally owned by Greece, but claimed by both

l. **1997** Threatened deployment by Greece of a Soviet S300 air defence missile system to Cyprus. Turkey said they would use force to prevent its deployment. Situation saved by diplomacy

Predictions for **1976** and **1988** are false-positives, i.e. instabilities were predicted but (as far as is known) did not materialise. There was no prediction for 1996, so we have one false-negative.

Table 6.1 Truth-table: Greece/Turkey instabilities

		occurred?	
		yes	**no**
predicted?	**yes**	11	2
	no	1	

6.4 India versus Pakistan, 1955 to 2000

Here we examine the India/Pakistan conflict and use (as before) the two-nation version of Mayer-Kress's discretised Richardson equations, namely

$$\begin{aligned} x_{t+1} &= x_t + \big(k_{1,1}(x_s - x_t) + k_{1,2}\, y_t\big)(x_m - x_t) \quad . \\ y_{t+1} &= y_t + \big(k_{2,2}(y_s - y_t) + k_{2,1}\, x_t\big)(y_m - y_t)\, . \end{aligned} \tag{6.5}$$

Nation X is India, Nation Y is Pakistan.

There have been several major wars between India and Pakistan since partition in 1947 as the two countries squabble over Kashmir. Appendix 6A.2 gives the data.

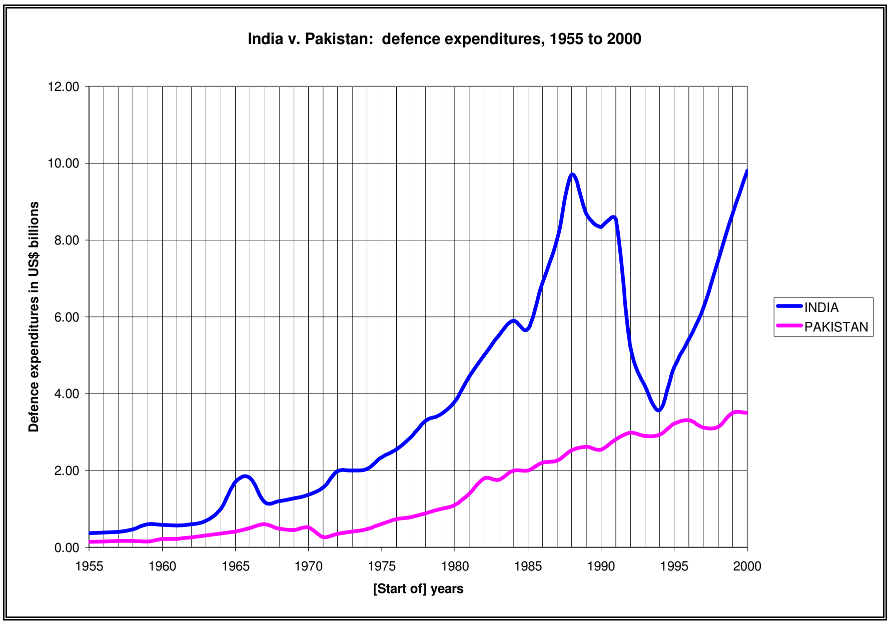


Figure 6.6 *IMF plot.* Data from Appendix 6A.2

6.4.1 The Model

Using a modelling GA, GA-15, derived from GA-6, initial curves were obtained by manipulating the evaluation equations Eqs(6.5) (see Section 5.4.2, steps 1-5), namely

$$\begin{aligned} x_{t+1} &= x_t + (-0.158\, x_t\ k_{1,1}/50 + y_t\, k_{1,2}/65)\, G_X / 1710 \quad . \\ y_{t+1} &= y_t + (-0.49\, y_t\ k_{2,2}/38 + x_t\, k_{2,1}/65)\, G_Y / 7001 \end{aligned} \tag{6.6}$$

yielding the starter-values $k_{1,1} = 59$, $k_{2,2} = 12$, $k_{1,2} = 32$, $k_{2,1} = 9$ (to be expressed simply as 59, 12, 32, 9).

As in Section 6.3.1, we use the ffn given by Eq(6.3); Cao et al.'s norm was then reduced from 998 to 623, at which point the rate-factors had become 96, 44, 23, 2. The output of the modelling GA, GA-15, after reducing the norm, is shown in Figure 6.7.

As in the case of Greece/Turkey, we use the evaluation equations of the prediction GA, Eqs(6.7), to obtain x_{t+1} and y_{t+1}, etc (collectively denoted by x_{calc} and y_{calc}) as a straightforward arithmetical exercise; x_{real} and y_{real} can then be compared with x_{calc} and y_{calc}.

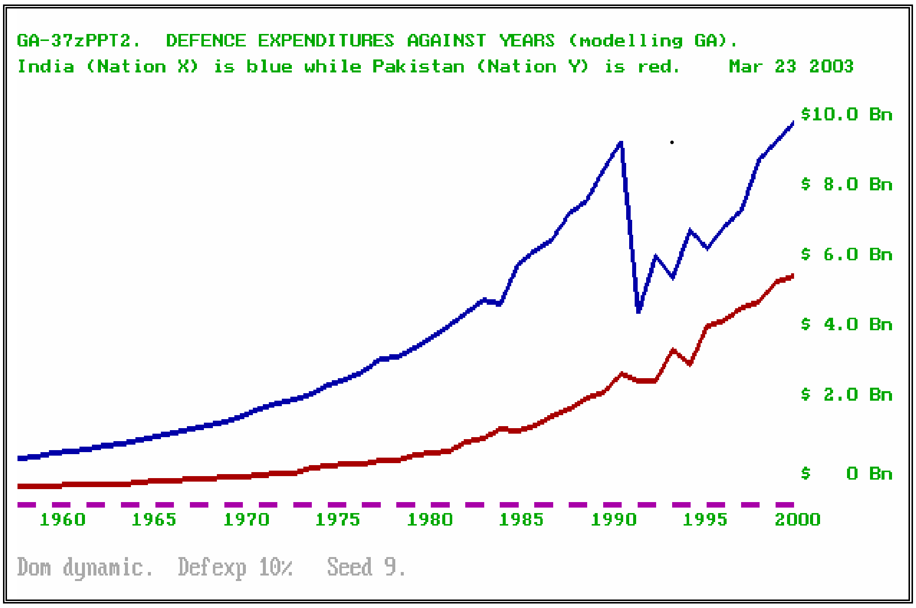


Figure 6.7 *The model.* GA-15, using Eqs(6.6). The curves follow the IMF plot (Figure 6.6) reasonably well

To initiate the calculation we use only x_0(*real*) and y_0(*real*). The calculated and the real curves are plotted alongside each other in Figure 6.8 and show quite a good match. Bearing in mind that the calculated values are cascaded, i.e. the only common points between calculated and real values are x_0 and y_0, it is remarkable that the values in year 2000 are so close together. This is an indication that 96, 44, 23, 2 are appropriate values for the four rate-factors.

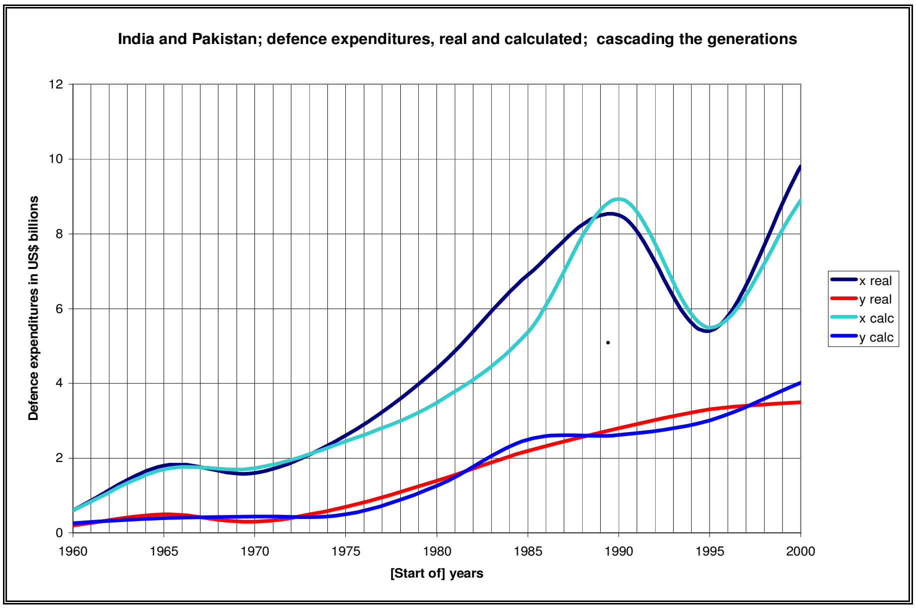


Figure 6.8 Comparing real with predicted values of x_t and y_t; using the prediction GA to verify that the appropriate rate-factors have been carried forward from the modelling GA

6.4.2 The Prediction

Armed with refined starter-values for the rate-factors (96, 44, 23, 2), we developed a prediction GA, GA-26, for India/Pakistan using the evaluation equations

$$x_{t+1} = x_t + \big((k_{1,1}/100)\, x_t\, (x_s/100) + (k_{1,2}/100)\, y_t\big)\big((x_m - x_t)/267\big) \; .$$
$$y_{t+1} = y_t + \big((k_{2,2}/100)\, y_t\, (y_s/100) + (k_{2,1}/100)\, x_t\big)\big((y_m - y_t)/267\big) \; . \qquad (6.7)$$

Eqs(6.7) are derived from Eqs(6.5), see Section 5.5. They are used to calculate the sign changes in the direction of curvature of the limit cycle, see Figure 6.9. By Generation 39, the rate-factors had become 20, 39, 38, 2.

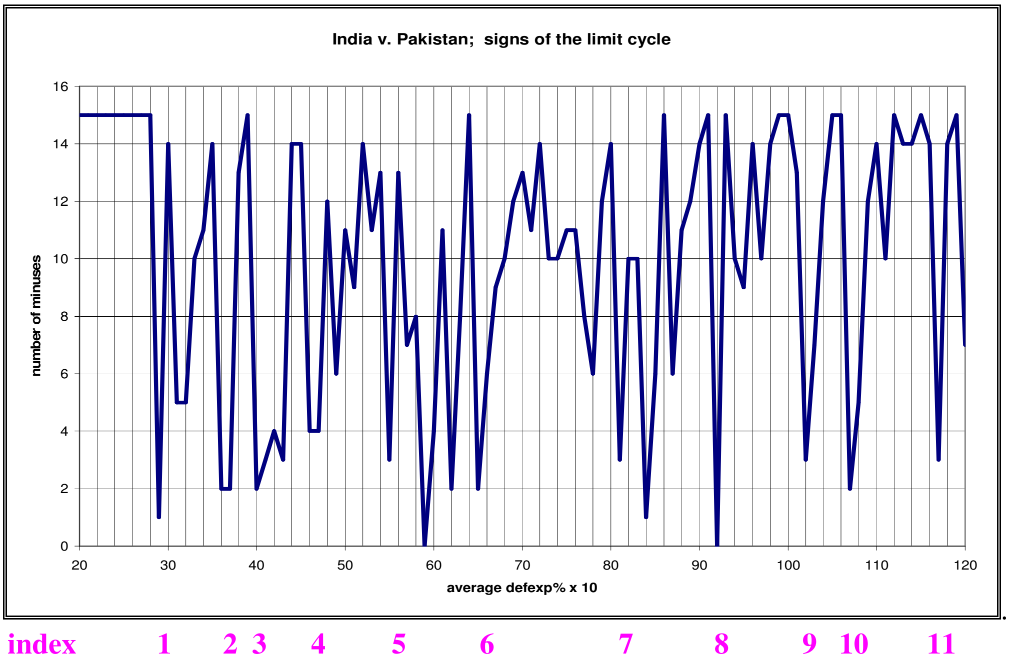


Figure 6.9 *Prediction plot*. GA-26. The number of minuses of *a* - *c* in successive iteration steps. The index numbers show where sharp drops occur

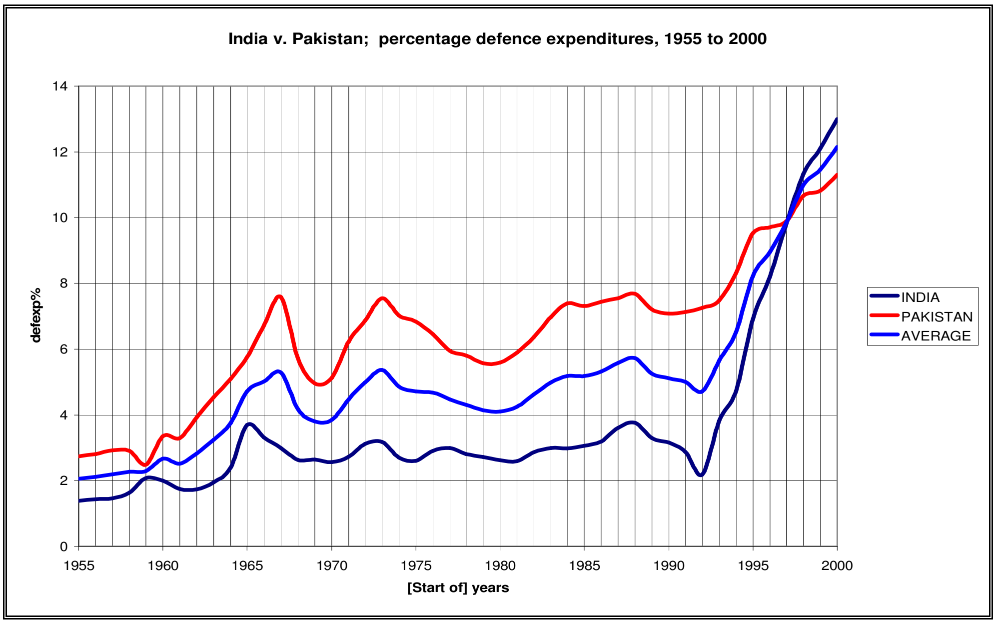


Figure 6.10 d*efexp% plot*. The defexp% of the two sides, and their average

Note from Figure 6.10 that the two defexp% curves intersect in mid-1998, shortly before the hostilities erupted into the smouldering warfare that continues to this day.

Results. Below we list the average defexp% at which sharp drops occur in Figure 6.9 together with the corresponding dates extracted from Figure 6.10. Alongside are the years pertaining to relevant historical events**.** Sharp drops occurred for the eleven index numbers shown as follows:

① 15→ 1 @ 2.8% which ties with 1960

② 14→ 2 @ 3.6% which ties with 1965 1969

③ 15→ 2 @ 4.0% which ties with 1965 1967 1971 .

④ 14→ 4 @ 4.6% which ties with 1965 1967 1971 1975 1983 1992

⑤ 13→ 3 @ 5.5% which ties with 1987 1992

⑥ 15→ 2 @ 6.5% which ties with 1994

⑦ 14→ 3 @ 8.1% which ties with 1995

⑧ 15→ 0 @ 9.2% which ties with 1996

⑨ 13→ 3 @ 10.2% which ties with 1998

⑩ 15→ 2 @ 10.7% which ties with 1999

⑪ 14→ 3 @ 11.6% which ties with 2000

Corresponding historical events were as follows:

a.	**1960**	River Indus Treaty
b.	**1965**	Full scale war
c.	**1969**	Unrest on both sides, martial law declared in Pakistan
d.	**1971**	The ‘separation war’
e.	**1975**	India diverted the waters of the River Jhelen. Tension defused by diplomatic action
f.	**1983**	Calls for the creation of a separate Sikh state. A massacre in Amritsar. Numerous deaths in Sindh Province
g.	**1987**	The Pathans riot in Karachi; India blamed
h.	**1992**	An Indian flag hoisted in Srinagar (the Kashmiri capital) caused rioting. A right-wing Hindu party stormed the joint mosque/temple complex at Ayodha, destroying the mosque
i.	**1994**	Islamic condemnation of India for terrorism
j.	**1995**	India foments riots over Shari’a enforcement
k.	**1996**	No war
l.	**1997**	No war, but a very critical report from the UN Human Development Centre
m.	**1998**	Both India and Pakistan hold nuclear tests
n.	**1999**	Open hostilities start
o.	**2000**	One inflammatory speech causes widespread unrest and nearly starts a full-scale war

The prediction for **1996** was a false-positive, i.e. an instability was predicted but (as far as is known) did not materialise. On the other hand, Gilbert [2000] does not record any military activity on the sub-continent in the spring of 1997, but he does say that at the beginning of May 1997 the UN Human Development Programme called worldwide attention to a report from the UN Human Development Centre at Mahbub-ul-Haq in Pakistan to the effect that the per capita income of South Asia had for the first time fallen behind that of sub-Saharan Africa, making it "the poorest, most illiterate and malnourished region on earth". The report firmly put the blame on arms expenditure, first by India, with the world's highest arms import bill and 142nd in terms of per capita income, and second by Pakistan, with the tenth largest arms import bill and 119th in the income table. In the event, there was no India/Pakistan war in the summer of 1997. A sharp drop does not *have* to result in an outbreak of war (see, for instance, 1975 or 1994), but it is more likely than not.

Table 6.2 Truth-table **:** India/Pakistan instabilities

		occurred?	
		yes	**no**
predicted?	**yes**	14	1
	no	0	

6.5 The Middle East, 1955 to 2000

We use Mayer-Kress's discretised three-nation variant of Richardson's equations, namely

$$\begin{aligned} x_{t+1} &= x_t + \big(k_{1,1}(x_s - x_t) + k_{2,3}(y_t + z_t)\big)(x_m - x_t) && . \\ y_{t+1} &= y_t + \big(k_{2,2}(y_s - y_t) + k_{1,3}(x_t - z_t)\big)(y_m - y_t) && . \\ z_{t+1} &= z_t + \big(k_{3,3}(z_s - z_t) + k_{1,2}(x_t - y_t)\big)(z_m - z_t) \ , && (6.8) \end{aligned}$$

where

- x_t , y_t and z_t are the expenditures of Israel (Nation X), Egypt (Nation Y) and Syria (Nation Z) on arms for the current year.
- x_s, y_s and z_s are the intrinsic arms expenditures (how much each country spends on defence irrespective of competitive spending by its neighbours). These figures are not published by the IMF; here we use the UK's $0.78x_t$ as an approximation for x_s.
- The terms $(y_t + z_t)$, $(x_t - z_t)$ and $(x_t - y_t)$ are the model's method of denoting the external threat from adversaries for the nations X, Y and Z, respectively.
- x_m , y_m and z_m represent, respectively, the expenditure authorised by the budgets of Nations X, Y and Z to be spent on arms. During an arms race x_m , y_m and z_m are not necessarily equal to x_t, y_t and z_t, respectively.

6.5.1 *Fitness Functions* : *Three-nation Scenario*

When ffn's were discussed in Section 2.9, it was said that different criteria applied to the three-nation case. After calculating the three evaluation equations for x_t, y_t and z_t we need to know which one of Nations X, Y or Z is dominant at the time, i.e. which one of x_t, y_t or z_t is numerically the biggest. Let F be the risk of war. If we drop the suffix t, then

$$F = x - (y + z) \quad \text{or} \quad F = y - (x - z) \quad \text{or} \quad F = z - (x - y), \tag{6.9}$$

depending on whether Nation X, Y or Z is temporarily dominant. (In absolute terms, all three expressions are identical).

As before, we improve the model by reducing Cao *et al.*'s norm; the ffn we use is given by

$$fitness = \sqrt{(x_{calc} - x_{real})^2 + (y_{calc} - y_{real})^2 + (z_{calc} - z_{real})^2} \quad . \tag{6.10}$$

The IMF plot. Appendix 6A.3 presents the data. As before, all the IMF figures are converted into billions of US dollars (using the conversion rate applicable at the time) as a precaution against distortion by the effects of inflation [SIPRI, 1982]. The IMF plot is given in Figure 6.11.

(We have not mentioned Lebanon and Jordan. At one stage we did include their figures, but they made little difference to those of the three big players and so were discarded).

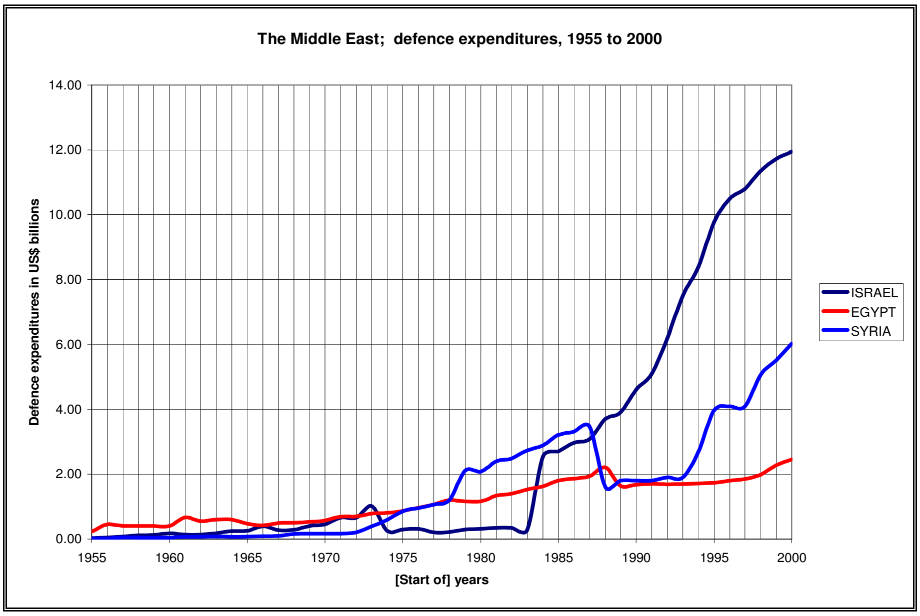


Figure 6.11 *IMF plot.* Data from Appendix 6A.3

6.5.2 *The Model*

Using a modelling GA, GA-17, initial curves were developed by manipulating the evaluation equations Eqs(6.8) (see Section 5.4.2, steps 1-5), namely

$$x_{t+1} = x_t + \left(-0.22\, x_t\, k_{1,1}/22 + (y_t + z_t)\, k_{2,3}/11\right) G_X / 3500 \quad .$$

$$y_{t+1} = y_t + \left(-0.22\, y_t\, k_{2,2}/42 + (x_t - z_t)\, k_{1,3}/20\right) G_Y / 3800 \quad .$$

$$z_{t+1} = z_t + \left(-0.22\, z_t\, k_{3,3}/19 + (x_t - y_t)\, k_{1,2}/19\right) G_Z / 3050 \qquad (6.11)$$

yielding the starter-values $k_{1,1} = 720$, $k_{2,2} = 220$, $k_{3,3} = 29$, $k_{2,3} = 50$, $k_{1,3} = 12$ and $k_{1,2} = 11$ (expressed simply as 720, 220, 29, 50, 12, 11). By adjusting the values of the rate-factors to 714, 226, 31, 50, 12, 11 Cao *et al.*'s norm was reduced from 225 to 138. The output of GA-17 after the reduction of the norm is shown in Figure 6.12.

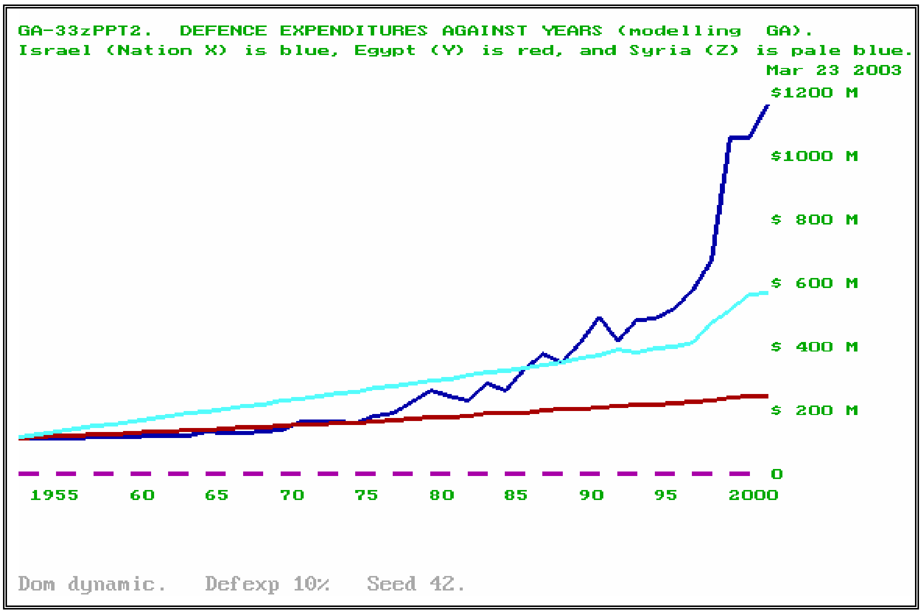


Figure 6.12 *The model.* GA-17, using Eqs(6.11). The curves mimic the IMF plot (Figure 6.11) well

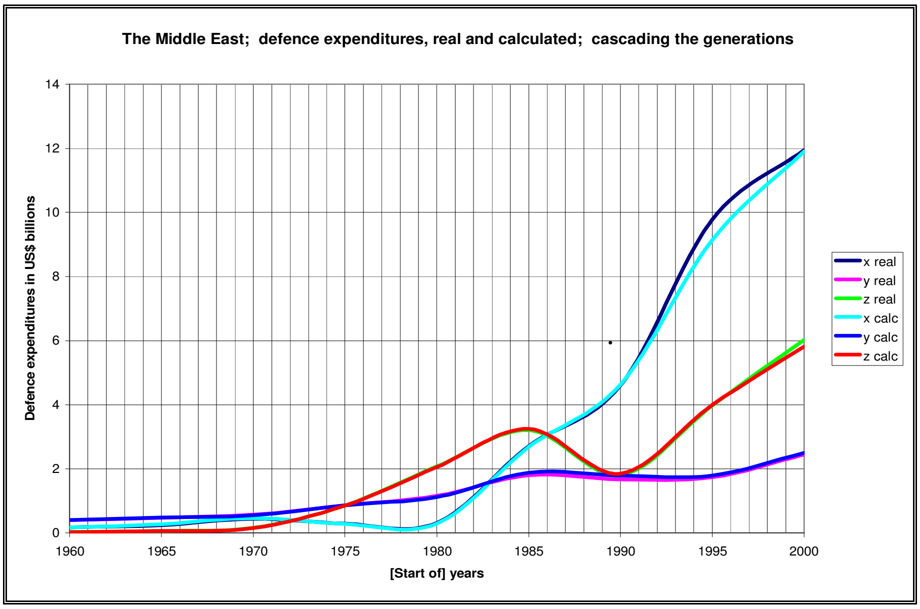


Figure 6.13 Comparing real with predicted values of x_t , y_t and z_t**;** using the prediction GA to verify that the appropriate rate-factors have been carried forward from the modelling GA

As in the case of Greece/Turkey and India/Pakistan, we use the evaluation equations of the prediction GA, Eqs(6.12), to obtain x_{t+1}, y_{t+1} and z_{t+1}, etc (collectively denoted by x_{calc}, y_{calc}

and z_{calc}) as an arithmetical exercise; x_{real} , y_{real} and z_{real} can then be compared with x_{calc} , y_{calc} and z_{calc} . To initiate the calculation we use only $x_0(real)$, y_0 $(real)$ and $z_0(real)$. The calculated and the real curves are plotted alongside each other in Figure 6.13 and show a very good match. This is an indication that 714, 226, 31, 50, 12, 11 are appropriate values for the six rate-factors.

6.5.3 The Prediction

Armed with refined starter-values for the rate-factors, (714, 226, 31, 50, 12, 11), we developed a prediction GA, GA-4, for the Middle East using the evaluation equations

$$x_{t+1} = x_t + \left(k_{1,1}(x_s - x_t)/1000 + k_{2,3}\,(y_t + z_t)/1000\right)(x_m - x_t) \quad .$$

$$y_{t+1} = y_t + \left(k_{2,2}(y_s - y_t)/1000 + k_{1,3}\,(x_t - z_t)/1000\right)(y_m - y_t) \quad .$$

$$. \quad z_{t+1} = z_t + \left(k_{3,3}(z_s - z_t)/1000 + k_{1,2}\,(x_t - y_t)/1000\right)(z_m - z_t)\,. \qquad (6.12)$$

Eqs(6.12) are derived from Eqs(6.8), see Section 5.5. These equations will be used to calculate the sign changes in the direction of the curvature of the limit cycle for Figure 6.14, but before we do so we need to decide how to calculate the equivalent of a - c in the three-nation case. It was shown in Section 3.8 that, for two nations X and Y, a - c could be approximated by $k_{2,1}\,y_m - k_{1,2}\,x_m$. For three adversaries (Nations X, Y and Z) engaged in a two-sided conflict, it was shown in Eq(3.17) that we could approximate a - c by

$$k_{1,3}\,y_m - (k_{2,3}\,x_m - k_{1,2}\,z_m) \; . \qquad (6.13)$$

We thus use Eq(6.13) in plotting Figure 6.14.

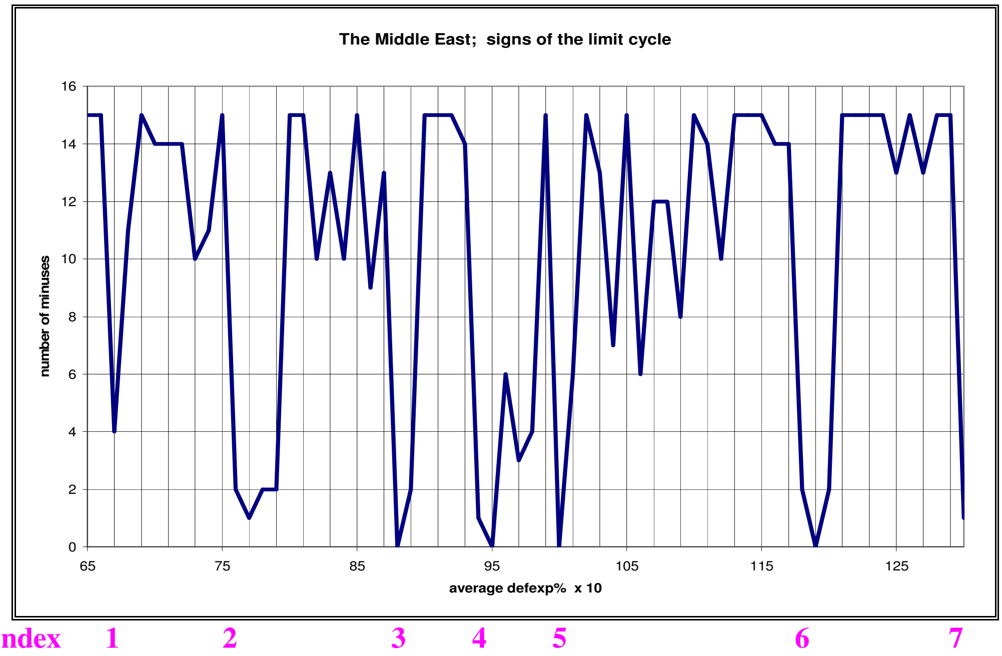


Figure 6.14 *Prediction plot.* GA-4. The number of minuses of a - c in successive iteration steps. The index numbers show where sharp drops occur

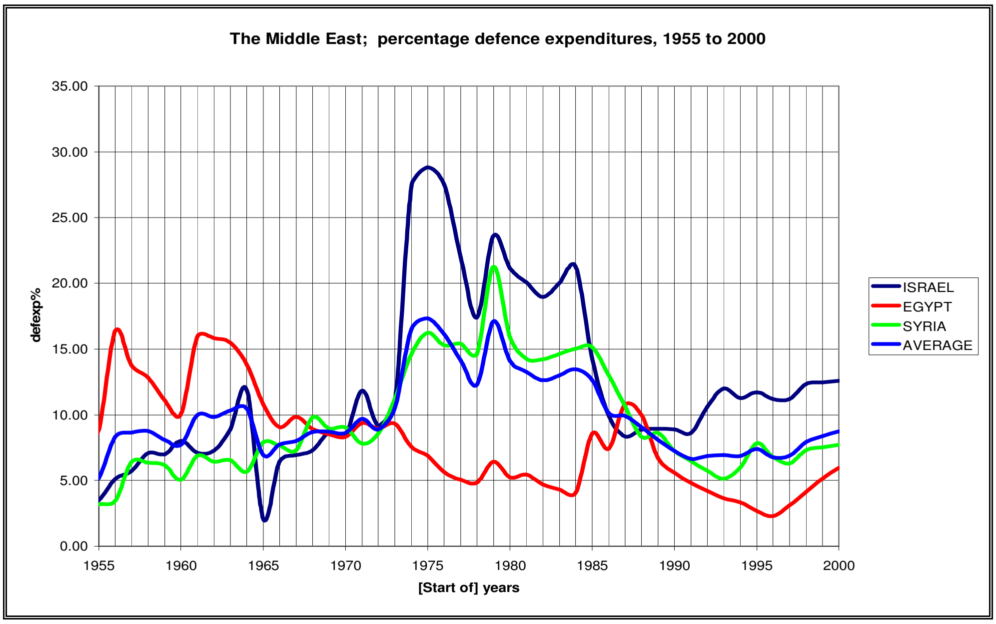


Figure 6.15 d*efexp% plot*. The defexp% curves of the three sides, and their average

Results. Below we list the average defexp% at which sharp drops occur in Figure 6.14, together with the corresponding dates extracted from Figure 6.15. Alongside are the years pertaining to relevant historical events. Sharp drops occurred at an average defexp% for the seven index numbers shown as follows:

① 15→ 4 @ 6.7% which ties with 1956 1991
② 15→ 2 @ 7.6% which ties with 1967 1965 1967 1991 1999
③ 13→ 0 @ 8.8% which ties with 1962 1968 1971 1973 1987
④ 14→ 1 @ 9.4% which ties with 1962 1967 1973 1987
⑤ 15→ 0 @ 10.0% which ties with 1962 1965 1973 1987
⑥ 14→ 2 @ 11.8% which ties with 1973 1987
⑦ 15→ 1 @ 13.0% which ties with 1973 1978 1982 1987

Corresponding historical events were as follows:

a. **1956** Suez
b. **1967** The Six-day War
c. **1968** Start of the year-long War of Attrition
d. **1971** War on the Golan Heights
e. **1973** The Yom Kippur war
f. **1974** Repeated Israeli reprisal attacks on Lebanon
g. **1978** Another Israeli reprisal attack on Lebanon
h. **1982** Israel invaded Lebanon; Sabra and Chatila massacres

i. **1987** Heavy fighting in Beirut between Amal and Druze militias. 2500 killed by Christian militias during siege of Palestinian refugee camps in Tyre and Sidon

j. **1991** The Gulf War

k. **1999** Start of current hostilities

Predictions for 1962 and 1965 are false-positives, i.e. instabilities were predicted but did not materialise.

Table 6.3 Truth-table: Middle East instabilities

		occurred?	
		yes	**no**
predicted?	**yes**	10	2
	no	0	

Figures 6.14 and 6.15 can be said to vindicate our approach and, by predicting the outset of most of the hostilities in the Middle East during the period 1955 to 2000, appear to have validated the model. At least in this case, given statistical data from nations indulging in an arms race, it seems that it *is* possible to predict with reasonable accuracy when smouldering hostilities will erupt into open warfare.

6.6 Predicting the Future

We did make – but later abandoned – an attempt to predict the future by extending the trial range of the prediction plot and looking for more sharp drops. We then put this information together with an extrapolated *average* defexp% curve in an effort for genuine prediction. In fact, the difficult part of this operation was forecasting future average defexp%'s. However, we may reasonably say that there will potentially be conflicts when the following conditions are reached:

Greece/Turkey . When the average defexp% reaches 10.1%, 10.4% and 10.6%

India/Pakistan. When the average defexp% reaches 12.9%, 13.1% and 14.0%

Note. An extrapolation of Figure 6.10 (see the large-scale version at Appendix R1) shows that the average defexp% would appear to have reached 12.9% in late 2001. Open hostilities between India and Pakistan did indeed start again in November 2001.

The Middle East. When the average defexp% reaches 13.0%, 13.8%, 14.8% and 15.1%.

6.7 The Effects of Noise

As a further check on the validity of our model we deliberately added *noise* to x_s, y_s and to x_m, y_m, x_t, y_t, (IMF figures) at Generation 0, in the three prediction GAs. This was achieved by collecting all the Generation 0 data together in Matrices 1 and 2 in the normal way (see Chapter 4), scanning the data, and using a separate RNG [2] to select certain genes (with a probability p_n%) to have noise added. Genes so selected had their alleles doubled (how much they are physically altered, within reason, seems not to be critical). We kept count of how many of the 600 or 900 genes[3] were altered on each run. The process just described is very similar to that to be used for mutation later in the GAs , except that

- We are *not* mutating the elements of a chromosome for GA purposes, but we are merely adding noise at a rate of p_n% to the initialising genes. Mutation comes later.
- We introduce noise to *whole genes*, cf. Shannon's 'pulse trains' [1949]. In contrast, the mutation process examines *individual elements* and selects them (or not) for mutation.

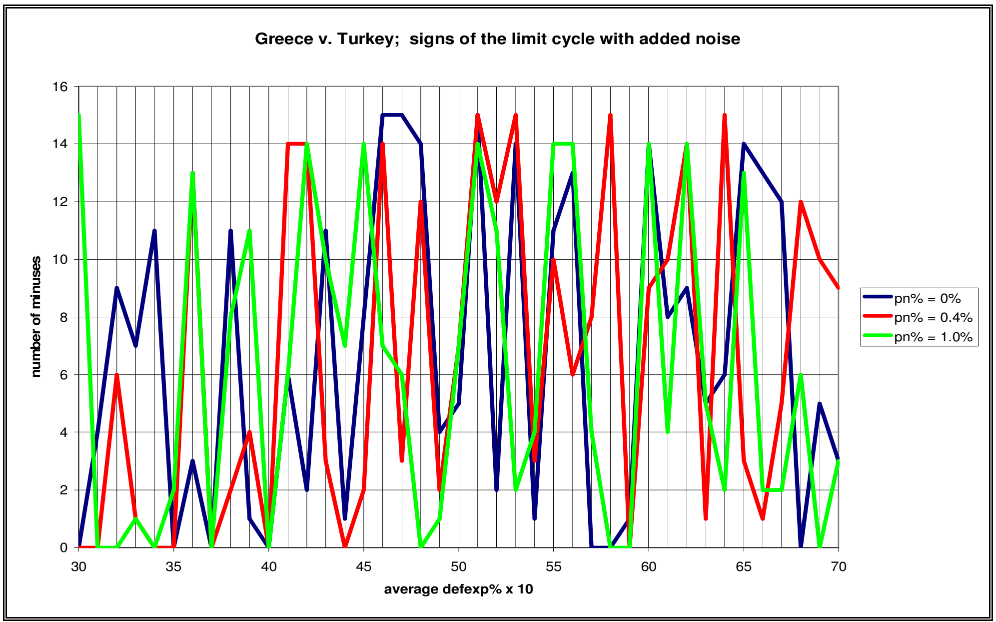


Figure 6.16 Signs of the (Greece/Turkey) limit cycle with added noise[4]. Compare with Figure 6.4

We ran GA-16 and plotted the progress of the direction of the curvature of the limit cycle, previously Figure 6.4, using different rates of p_n%. Edited results are given in Figure 6.16. Up to a noise level of 0.3%, there was only one *alteration* in the alleles of the genes of Matrices 1 and 2, which meant that there was virtually no change to the original Figure 6.4 for which $p_n\% = 0$. At $p_n\% = 0.4\%$, however, there was a marked change (three alter-

---

[2] The simple RNG used to introduce noise was of the "linear congruential" type, and was a separate user-declared function. This was done to avoid fouling the sequence of the embedded 'system' RNG used so extensively in the various GAs

[3] 6 genes (2-nation) or 9 genes (3-nation) x 100 population

[4] On the particular printer used, should lines coincide then red dominates black and green dominates both black and red

ations), although the overall shape and the sharp drops of Figure 6.4 were clearly recognisable. There was then no further change until p_n% reached 1.0%, when there were five alterations, and another change at $p_n\% = 1.6\%$, whereupon the original shape was effectively lost, although the drops at an average defexp% of 4.4%, 4.9%, 5.4% and 5.7% (corresponding to indices ③,④,⑥,⑦, respectively, of Figure 6.4) were still recognisable.

In the **India/Pakistan** case, the 'quiet' scenario (Figure 6.9) is shown in black in Figure 6.17. Here we run GA-26. Noise up to 0.13% (which involved three alterations to the genes of Matrices 1 and 2) made no difference at all to the quiet scenario. Noise between 0.14% and 0.32% (four alterations) is shown in red while noise from 0.33% upwards (five alterations) is shown in green.

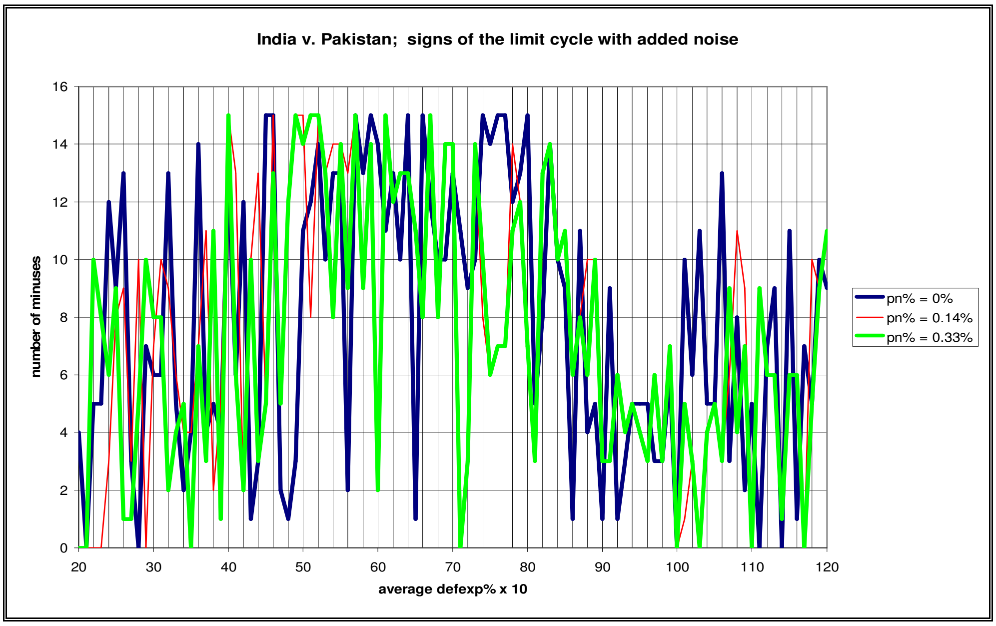


Figure 6.17 Signs of the (India/Pakistan) limit cycle with added noise. Compare with Figure 6.9.

A similar picture arises in the **Middle East**. Here we run GA-4. By adding 0.4% and 0.8% noise to the basic scheme of Figure 6.14 we arrive at Figure 6.18. Up to 0.3% noise (one alteration), there was no change to the 'quiet' scenario of Figure 6.14. Between 0.4% and 0.7% (two alterations) there was a change portrayed in red in Figure 6.18. Above 0.8% (four alterations) there was another jump, which is pictured in green.

In summary, the effects of increasing noise did not take place gradually but in quantum jumps depending on the number of alterations made while the data of Generation 0 in Matrices 1 and 2 was being scanned. The India/Pakistan prediction is much more susceptible to noise than the other two, in that the first perceptible change occurred with only p_n % = 0.14% added noise. Both Greece/Turkey and the Middle East first changed with added noise at $p_n\% = 0.4\%$.

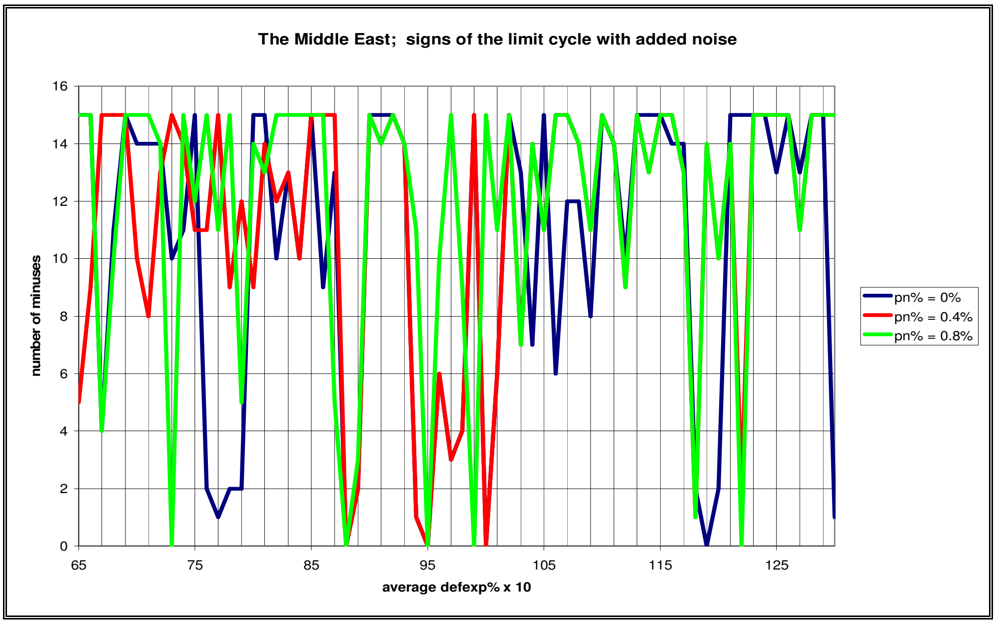


Figure 6.18 Signs of the (Middle East) limit cycle with added noise. Compare with Figure 6.14

6.8 Conclusions

The almost continuous conflict in the Middle East from 1955 to 2000, the India/Pakistan wars of 1965, 1971 and 1998, and the armed dispute between Greece and Turkey over Cyprus and the Aegean in 1974 and 1997, all had very different scenarios. In all three conflicts, however, the use of Richardson's equations in modelling and prediction GAs has meant that the timing of the various outbreaks of hostilities could have been predicted with considerable accuracy.

In compiling Appendix 6A we made the initial mistake of using Gilbert's magisterial *History of the 20th Century 1952 – 2000* as the sole authority. This left us with some unexplained sharp drops, such as the average defexp% of 4.6% in Figure 6.9 (India/Pakistan), which from Figure 6.10 corresponds to both 1965 (the full scale war) and to 1975. Reference to the *Europa World Yearbook*, which is more economics-based than Gilbert, shows that in 1975 India threatened to divert the waters of the River Jhelen (and, indeed, did so for quite a time) effectively making sizeable parts of Pakistan semi-arid. It is not surprising that the Government in Islamabad was alarmed! (At the risk of gross over-simplification, it does seem that most wars arise over the ownership and use of land and natural resources such as water and oil).

So much for predicting wars with hindsight, which is what we have largely done. Can we use these methods for *genuine* prediction, e.g. are there going to be wars between Greece and Turkey, India and Pakistan, or the Middle East in, say, 2003/4? We believe that we can, provided and only provided that accurate, up-to-date data can be obtained so that meaningful extrapolation is possible. The cover sheet to Appendix 6A lists some of the problems of obtaining really up-to-date figures, and Section 6.7 has indicated the penalties of using noisy information. Nevertheless in Section 6.6 we hazarded a guess as to specific conditions under which conflicts can be expected.

In spite of the above, this chapter has shown that statistical figures taken from reference works published only annually result in reasonable predictions. Results would be better if accurate figures (e.g. from governmental military and economic intelligence sources) could be fed in on, say, a weekly basis but this requires resources well beyond the capability of a university researcher.

Section 3.7 states *"The evidence presented ... would suggest, therefore, that an unstable point in the Richardson equations is always prefaced by a change in the direction of curvature of the limit cycle, and that this is manifested by a sudden and marked change in the sign of $a - c$, from minus to plus, across the population. The experimental evidence for saying this may be slender, but the theoretical argument is quite strong".* While far from proving anything, this chapter has shown that the experimental evidence is not so slender after all.

In conclusion we can say with confidence that in our context we have developed a GA-based technique which can be used to anticipate impending conflicts.

Appendix 6A lists statistical data from IMF, UN and other official sources. Its preface discusses the particular problems of extracting data for the arms race application.

Appendix 6A

STATISTICAL DATA

Appendix 6A.1: Greece and Turkey 1955 to 2000

Appendix 6A.2: India and Pakistan 1955 to 2000.

Appendix 6A.3: The Middle East (Israel, Egypt and Syria) 1955 to 2000.

Principal Sources

[IMF Statistics; UN Statistical Yearbooks; The Statesman's Year Books; Handbooks of the Stockholm International Peace Research Institute (SIPRI) [5]; Europa World Yearbooks].

Preface

(1) Many techniques have been devised over the years to prepare a wide variety of datasets for use as the foundation of both predictive and inferential models. Collectively these techniques are now known as *data mining* [Pyle, 1999]. A clear statement of the problem is half the battle of mining data. It is (and points directly to) the solution needed. Defence statistics are not difficult to handle so the problem space does not need much exploration. We do not, for instance, need *pairwise ranking* or other econometric techniques to tell us how to calculate percentage defence expenditures. Similarly, the solution space permits a precise, real-world, implementable solution. As in all real-world problems, we expect the target to move as the study progresses. We have already said (Section 3.2) that nothing in Richardson's equations is constant, so we should not expect the answers to be.

(2) A clear distinction has to be drawn between the datasets used for training or testing and the execution dataset, even though they are all drawn ultimately from the same 'pool' of information. It so happens that Chapter 5, dealing among other things with the India/Pakistan 'snapshot', was written in 1999. Today, however, we could use that information as a training dataset to determine whether by its use our model was able to predict the near-war of September 2000, despite the fact that Appendix 6A.2 lists all (or, more precisely, a *representative sample* of) the India/Pakistan data seamlessly from 1955 to 2001 as a single 'pool' of information.

(3) It is recognised that in publishing defence statistics, governments are often 'economical with the truth' in the interests of their own security. Nevertheless, it would not be prudent for them to falsify defence costs too much because the international bankers who make loans might consider that a nation was spending too little on defence in the

---

[5] SIPRI figures are quoted but are not actually used in any calculations. SIPRI defence statistics appear to be generally higher than those from other sources, possibly due to their evident political agenda

face of an evident threat, and therefore was not a good investment. In any case, when viewed year on year, such tables do indicate trends.

(4) All the figures in these appendices have been converted to billions of US dollars, using the conversion rates from local currency applicable *during the month when the figures were published.* As the US dollar has held steady over the past few years, this currency conversion is designed to take account of local inflation which has sometimes, as in the case of Turkey, been extreme. However, even if the dollar does gently increase in value, it is a firm baseline for other currencies. Hence like can be compared with like.

(5) The 'ten golden rules' for exploring data [Pyle, 1999] have been followed as closely as possible. In particular, the problem drives the model, not vice versa. No formal checks were made on data variability, deviation or confidence as we assumed (with some justification)[6] that the IMF and UN Statistics Departments would have done this already.

(6) Whereas all the authorities are unanimous about defence statistics before about 1998, getting more up-to-date information is much more difficult:

- The IMF updates its latest figures each year (e.g. the 2001 Annual IMF Report, the 2000 Annual IMF Report, and the 1999 Annual IMF Report can give different figures for the same fiscal parameter). For example, the data used to plot Figure 5.7[7] was taken from the 1999 IMF Report, whereas that for Figure 6.10 came from the 2001 Report. Over the period 1993 to 1999, the two sets of figures are materially different.
- The IMF and the United Nations Reports do not always agree with each other.
- Some of the more recent figures are not even consistent internally; for example, figures given for population and for GDP per head, for defence expenditure, and for defexp% do not always tally arithmetically. Where there are gaps or obvious inconsistencies, the data probably come from different sources. The only uncertainties in the model lie in the most recent years, e.g. in 1999, 2000, 2001 and 2002.

Prediction of conflicts for the immediate future (e.g. for 2003 or 2004) based on the published information open to us will necessarily have low confidence. We believe, however, that we can justifiably claim *proof of concept.*

---

[6] Footnotes to the IMF statistics make reference to Normalised Root Mean Square Errors, Correlation Coefficients, Mean Relative Errors, Mean Absolute Errors and Mean Square Errors, leading us to infer that they had applied a number of corrections to their raw data

[7]

1999 IMF data:	1993	1994	1995	1996	1997	1998	1999		1993	1994	1995	1996	1997	1998	1999
India GDP	219	195	204	213	202	202	190	Pakistan GDP	38.79	35.25	33.69	33.97	31.56	29.37	29.03
India defexp	4.2	3.57	3.68	4.41	5.59	13.51	22.61	Pakistan defexp	1.79	2.24	2.32	2.59	3.19	3.37	3.6
India defexp%	1.9	1.8	1.8	2.1	2.7	6.7	11.9	Pakistan defexp%	4.6	6.3	6.8	7.6	10.1	11.5	12.4

GREECE versus TURKEY, 1955 to 2000

						Appendix 6A		**1**
STATISTICAL INFORMATION			**GREECE**				**Sheet**	**1**
Year	GREECE	GREECE	GREECE	GREECE	GREECE	GREECE	GREECE	Year
	GDP	conversion	GDP	Defexp	Defexp	%defexp	%defexp	
	Bn-drach	drach = $1	$Bn	Bn-drach	$Bn		*SIPRI*	
1955	56.3	30.00	1.88	8.49	0.28	6.19		1955
56	64	30.00	2.13	9.33	0.31	5.73		56
57	71	30.00	2.37	8.60	0.29	5.77		57
58	79	30.00	2.63	8.16	0.27	5.60		58
59	82	30.00	2.73	6.00	0.20	5.52		59
1960	88	30.00	2.93	4.86	0.16	4.95		1960
61	100	30.00	3.33	4.95	0.17	4.97		61
62	105	30.00	3.50	5.22	0.17	4.78		62
63	116	30.00	3.87	5.55	0.19	4.20		63
64	130	30.00	4.33	5.46	0.18	3.59		64
65	167	30.00	5.57	6.00	0.20	3.69		65
66	195	30.00	6.50	7.20	0.24	4.30		66
67	208	30.00	6.93	8.95	0.30	4.59		67
68	226	30.00	7.53	10.37	0.35	4.83		68
69	258	30.00	8.60	12.46	0.42	4.75		69
1970	299	30.00	9.97	14.21	0.47	4.75	*4.80*	1970
71	330	30.00	11.00	15.48	0.52	4.69	*4.70*	71
72	378	30.00	12.60	17.21	0.57	4.55	*4.60*	72
73	484	29.70	16.30	19.87	0.67	4.10	*4.10*	73
74	564	30.00	18.80	24.13	0.80	4.28	*4.30*	74
75	672	35.60	18.88	43.92	1.23	6.54	*6.50*	75
76	825	37.03	22.28	52.67	1.42	6.38	*6.40*	76
77	964	35.50	27.15	66.58	1.88	6.91	*6.90*	77
78	1161	36.01	32.24	77.60	2.15	6.68	*6.70*	78
79	1429	38.28	37.33	82.30	2.15	5.76	*5.80*	79
1980	1711	46.54	36.76	90.00	1.93	5.26	*5.70*	1980
81	2050	57.53	35.63	142.80	2.48	6.97	*7.00*	81
82	2574	70.57	36.47	176.20	2.50	6.85	*6.80*	82
83	3079	98.67	31.21	193.34	1.96	6.28	*6.30*	83
84	3805	128.48	29.62	271.90	2.12	7.15	*7.10*	84
85	4617	147.76	31.25	321.98	2.18	6.97	*7.00*	85
86	5515	138.76	39.74	338.00	2.44	6.13	*6.20*	86
87	6259	125.93	49.70	393.00	3.12	6.28	*6.30*	87
88	7526	148.10	50.82	472.00	3.19	6.27	*6.40*	88
89	8777	157.79	55.62	503.00	3.19	5.73	*6.80*	89
1990	10455	157.63	66.33	597.00	3.79	5.71		1990
91	12898	175.28	73.59		3.63	4.93		91
92	19832	214.58	92.42		3.39	3.67		92
93	21135	249.22	84.80		3.21	3.79		93
94	23755	240.10	98.94		3.02	3.05		94
95	27235	237.04	114.90		2.80	2.44		95
96	29835	247.02	120.78		3.84	3.18		96
97	33021	282.61	116.84		5.55	4.75		97
98	35910	282.57	127.08		5.72	4.50		98
1999		328.44						1999
2000		337.90						2000

GREECE versus TURKEY, 1955 to 2000

							Appendix 6A	1
STATISTICAL INFORMATION				**TURKEY**			**Sheet**	**2**
Year	TURKEY	TURKEY	TURKEY	TURKEY	TURKEY	TURKEY	TURKEY	**GRE/TUR**
	GDP	conversion	GDP	Defexp	Defexp	%defexp	%defexp	**average**
	Bn-lira	Lira = $1	$Bn	Bn-lira	$Bn		*SIPRI*	**defexp%**
1955	12	2.80	4.29	0.31	0.11	7.58		6.89
56	15	2.80	5.36	0.33	0.12	6.34		6.04
57	27	2.80	9.64	0.45	0.16	3.63		4.70
58	35	4.90	7.14	0.95	0.19	2.74		4.17
59	42	9.00	4.67	1.12	0.12	2.29		3.91
1960	49	9.04	5.42	1.25	0.14	2.35		3.65
61	54	9.04	5.97	1.28	0.14	2.30		3.64
62	58	9.04	6.42	2.11	0.23	3.64		4.21
63	62	9.04	6.86	2.80	0.31	4.52		4.36
64	71	9.08	7.82	2.91	0.32	4.10		3.85
65	80	9.08	8.81	3.09	0.34	3.86		3.78
66	93	9.08	10.24	3.48	0.38	3.74		4.02
67	103	9.08	11.34	3.87	0.43	3.76		4.17
68	114	9.08	12.56	4.72	0.52	4.14		4.48
69	125	9.08	13.77	4.27	0.47	3.42		4.08
1970	145	12.90	12.73	7.69	0.51	4.01	*4.30*	4.38
71	187	14.20	13.17	8.65	0.60	4.53	*4.70*	4.61
72	232	14.20	16.34	9.96	0.70	4.29	*4.40*	**4.42**
73	295	14.20	20.77	12.19	0.86	4.13	*4.30*	**4.12**
74	410	14.00	29.29	15.83	1.13	3.86	*4.10*	**4.07**
75	519	15.20	34.14	30.20	1.99	5.82	*6.00*	**6.18**
76	664	16.70	39.76	40.69	2.44	6.13	*6.40*	**6.26**
77	863	19.40	44.48	49.79	2.57	5.77	*6.00*	**6.34**
78	1275	25.30	50.40	66.24	2.62	5.20	*5.50*	**5.94**
79	2156	35.40	60.90	94.03	2.66	4.36	*4.90*	**5.06**
1980	4328	90.10	48.04	169.47	1.88	3.92	*4.30*	**4.59**
81	6414	133.60	48.01	313.00	2.34	4.88	*4.90*	**5.92**
82	8620	186.80	46.15	448.00	2.40	5.20	*5.20*	**6.02**
83	11532	282.80	40.78	557.00	1.97	4.83	*4.80*	**5.55**
84	18212	444.70	40.95	803.00	1.81	4.41	*4.40*	**5.78**
85	27552	576.90	47.76	1235.00	2.14	4.48	*4.50*	**5.73**
86	39228	757.80	51.77	1868.00	2.47	4.76	*4.80*	**5.45**
87	58299	1021.00	57.10	2477.00	2.43	4.25	*4.20*	**5.26**
88	100826	1815.00	55.55	3789.00	2.09	3.76	*3.80*	**5.01**
89	167770	2314.00	72.50	6683.00	2.89	3.98	*3.90*	**4.86**
1990	282800	2390.00	118.33	13136.00	5.50	4.64	*3.90*	**5.18**
91			129.73		6.64	5.12		**5.03**
92	1129		141.02		6.15	4.36		**4.01**
93	2062		152.70		6.52	4.27		**4.03**
94	4026		161.95		5.49	3.39		**3.22**
95	7891	5965.00	171.30		6.39	3.73		**3.08**
96	14772	10777.00	182.70		7.29	3.99		**3.58**
97	28836	20560.00	194.50		8.17	4.20		**4.48**
98	51625	31446.00	186.10		8.21	4.41		**4.46**
1999		54140.00						
2000		55923.00						

INDIA versus PAKISTAN, 1955 to 2000

							Appendix 6A	**2**
STATISTICAL INFORMATION			**INDIA**				**Sheet**	**1**
Year	INDIA	INDIA	INDIA	INDIA	INDIA	INDIA	INDIA	Year
	GDP	conversion	GDP	Defexp	Defexp	%defexp	%defexp	
	Bn-rupees	I rup = $1	$Bn	Bn-rupees	$Bn		*SIPRI*	
1955								1955
56	131.00	4.80	27.29	1.92	0.40	1.46		56
57	133.30	4.77	27.95	2.17	0.46	1.63		57
58	136.00	4.78	28.45	2.83	0.59	2.08		58
59	140.00	4.78	29.29	2.79	0.58	1.99		59
1960	153.00	4.77	32.08	2.67	0.56	1.75		1960
61	161.00	4.77	33.75	2.81	0.59	1.74		61
62	168.00	4.77	35.22	3.28	0.69	1.95		62
63	198.00	4.78	41.42	4.74	0.99	2.39		63
64	220.90	4.79	46.12	8.16	1.70	3.69		64
65	261.5	4.77	54.82	8.65	1.81	3.31	3.60	65
66	295.7	7.58	39.03	8.83	1.17	2.99		66
67	346.1	7.55	45.86	9.09	1.20	2.63		67
68	366.7	7.63	48.07	9.68	1.27	2.64		68
69	403.9	7.56	53.43	10.33	1.37	2.56		69
1970	431.6	7.58	56.97	11.75	1.55	2.72	*3.00*	1970
71	462.6	7.28	63.55	14.40	1.98	3.11	*3.40*	71
72	510.1	8.08	63.13	16.20	2.00	3.18	*3.50*	72
73	620.1	8.20	75.59	16.74	2.04	2.70	*3.00*	73
74	732.4	8.15	89.87	19.04	2.34	2.60	*3.00*	74
75	787.6	8.94	88.13	22.82	2.55	2.90	*3.30*	75
76	848.9	8.88	95.59	25.40	2.86	2.99	*3.40*	76
77	960.7	8.21	117.03	27.04	3.29	2.81	*3.20*	77
78	1042.0	8.19	127.26	28.22	3.45	2.71	*3.10*	78
79	1144.0	7.91	144.68	29.99	3.79	2.62	*3.05*	79
1980	1360.0	7.93	171.50	35.18	4.44	2.59	*3.00*	1980
81	1579.0	9.10	173.54	45.37	4.99	2.87	*3.00*	81
82	1781.0	9.63	184.87	53.19	5.52	2.99	*3.10*	82
83	2075.0	10.49	197.75	61.90	5.90	2.98	*3.10*	83
84	2313.0	12.45	185.77	70.80	5.69	3.06	*3.20*	84
85	2619.0	12.17	215.27	83.65	6.88	3.19	*3.30*	85
86	2920.0	13.12	222.53	105.29	8.02	3.61	*3.70*	86
87	3326.0	12.88	258.29	124.90	9.70	3.76	*3.90*	87
88	3949.0	14.95	264.16	129.80	8.68	3.29	*5.80*	88
89	4506.0	17.04	264.51	142.00	8.34	3.15	*7.10*	89
1990	5355.0	18.07	296.30	154.38	8.54	2.88	*5.10*	1990
91	6168.0	25.83	238.76	134.34	5.20	2.18		91
92	5737.0	26.20	218.97	110.04	4.20	3.82		92
93	8769.0	31.38	279.45	112.03	3.57	4.73		93
94	10378.0	31.38	330.72	146.86	4.68	6.92		94
95	12179.0	35.18	346.19	190.32	5.41	8.21		95
96	14098.0	35.93	392.37	223.48	6.22	9.81		96
97	15635.0	39.28	398.04	293.42	7.47	11.32		97
98		42.48			8.69	12.10		98
1999		43.49			9.80	13.00		1999
2000		43.64				13.70		2000

INDIA versus PAKISTAN, 1955 to 2000

							Appendix 6A	**2**
STATISTICAL INFORMATION			**PAKISTAN**				**Sheet**	**2**
Year	PAKISTAN	PAKISTAN	PAKISTAN	PAKISTAN	PAKISTAN	PAKISTAN	PAKISTAN	**IND/PAK**
	GDP	conversion	GDP	Defexp	Defexp	%defexp	%defexp	**average**
	Bn-rupees	P rup = $1	$Bn	Bn-rupees	$Bn		*SIPRI*	**defexp%**
1955								
56	26.30	4.80	5.48	0.77	0.16	2.92		**2.19**
57	26.90	4.77	5.64	0.78	0.16	2.90		**2.27**
58	28.00	4.78	5.86	0.70	0.15	2.49		**2.28**
59	29.70	4.79	6.20	0.99	0.21	3.35		**2.67**
1960	30.34	4.77	6.36	1.00	0.21	3.29		**2.52**
61	31.60	4.77	6.62	1.24	0.26	3.92		**2.83**
62	32.40	4.78	6.78	1.47	0.31	4.54		**3.24**
63	33.75	4.79	7.05	1.72	0.36	5.10		**3.74**
64	34.43	4.80	7.17	1.98	0.41	5.75		**4.72**
65	35.25	4.78	7.37	2.37	0.50	6.72		**5.02**
66	37.66	4.81	7.83	2.86	0.59	7.58		**5.28**
67	40.12	4.77	8.41	2.29	0.48	5.72		**4.17**
68	44.16	4.81	9.18	2.19	0.45	4.95		**3.80**
69	47.75	4.79	9.97	2.45	0.51	5.13		**3.84**
1970	47.75	11.03	4.33	2.98	0.27	6.23	*6.10*	**4.48**
71	50.49	9.90	5.10	3.46	0.35	6.86	*6.60*	**4.99**
72	54.06	9.90	5.46	4.08	0.41	7.55	*6.80*	**5.36**
73	66.87	9.90	6.75	4.70	0.47	7.02	*6.10*	**4.86**
74	86.85	9.90	8.77	5.93	0.60	6.83	*6.00*	**4.72**
75	111.80	9.90	11.29	7.21	0.73	6.45	*5.90*	**4.67**
76	130.36	9.90	13.17	7.75	0.78	5.95	*5.50*	**4.47**
77	149.75	9.90	15.13	8.70	0.88	5.81	*5.40*	**4.31**
78	176.33	9.90	17.81	9.82	0.99	5.57	*5.30*	**4.14**
79	194.92	9.90	19.69	10.89	1.10	5.59	*5.10*	**4.10**
1980	234.18	9.90	23.65	13.79	1.39	5.89	*5.70*	**4.24**
81	278.20	9.90	28.10	17.73	1.79	6.37	*5.90*	**4.62**
82	324.16	12.84	25.25	22.64	1.76	6.98	*6.60*	**4.98**
83	364.39	13.50	26.99	26.92	1.99	7.39	*6.90*	**5.18**
84	419.80	15.36	27.33	30.69	2.00	7.31	*6.80*	**5.19**
85	472.16	15.98	29.55	35.11	2.20	7.44	*6.80*	**5.32**
86	514.53	17.25	29.83	38.86	2.25	7.55	*7.10*	**5.58**
87	572.48	17.45	32.81	44.00	2.52	7.68	*7.10*	**5.72**
88	675.39	18.65	36.21	48.60	2.61	7.20	*6.70*	**5.24**
89	769.75	21.42	35.94	54.48	2.54	7.08	*6.70*	**5.11**
1990	862.45	21.90	39.38	61.46	2.81	7.13	*7.10*	**5.00**
91	1020.00	24.80	41.13	73.90	2.98	7.25		**4.71**
92	1190.32	26.32	45.22	76.33	2.90	7.48		**5.65**
93	1341.00	30.05	44.63	88.03	2.93	8.31		**6.52**
94	1573.00	30.72	51.20	98.62	3.21	9.53		**8.22**
95	1882.00	34.17	55.09	112.74	3.30	9.71		**8.96**
96	2165.00	40.02	54.10	124.86	3.12	9.89		**9.85**
97	2404.00	43.94	54.71	137.53	3.13	10.66		**10.99**
98	2759.00	45.89	60.12	160.62	3.50	10.82		**11.46**
1999			63.29	222.14	3.50	11.30		**12.15**
2000						11.67		**12.69**

THE MIDDLE EAST, 1954 to 1960

						Appendix 6A		**3**
STATISTICAL INFORMATION			**MIDDLE EAST**				**Sheet**	**1**
	Year	**1954**	**1955**	**1956**	**1957**	**1958**	**1959**	**1960**
Israel GDP	Shekel bn	1.17	1.53	1.76	2.44	2.77	3.10	3.82
conversion	sh to $1	1.80	1.80	1.80	1.80	1.80	1.80	1.80
GDP	$bn	0.65	0.85	0.98	1.36	1.54	1.72	2.12
Defexp	Shekel bn	0.04	0.05	0.09	0.14	0.20	0.22	0.31
Defexp	$bn	0.02	0.03	0.05	0.08	0.11	0.12	0.17
defexp%	%	3.42	3.52	5.12	5.73	7.10	6.99	8.03
Egypt GDP	£E million	881	918	985	1018	1092	1262	1394
conversion	£E to $1	0.35	0.35	0.35	0.35	0.35	0.35	0.35
GDP	$bn	2.52	2.63	2.79	2.89	3.10	3.58	3.96
Defexp	£E bn	66	81.5	160.9	140	140	140	140
Defexp	$bn	0.19	0.23	0.46	0.40	0.40	0.40	0.40
defexp%	%	7.50	8.88	16.38	13.76	12.83	11.11	10.04
Syria GDP	£S million	1881	2078	2352	2514	2198	2266	2765
conversion	£S to $1	3.50	3.56	3.58	3.58	3.58	3.58	3.58
GDP	$bn	0.54	0.58	0.66	0.70	0.61	0.63	0.77
Defexp	£S bn	51	66	81.5	160.9	140	140	140
Defexp	$bn	0.015	0.019	0.023	0.045	0.039	0.039	0.039
defexp%	%	2.71	3.18	3.47	6.40	6.37	6.18	5.06
AVERAGE defexp%		4.54	5.19	8.32	8.63	8.77	8.09	7.71
		1955	1956	1957	1958	1959	1960	1961
Is defexp	$bn	0.02	0.03	0.05	0.08	0.11	0.12	0.17
Eg defexp	$bn	0.19	0.23	0.46	0.40	0.40	0.40	0.40
Sy defexp	$bn	0.015	0.019	0.023	0.045	0.039	0.039	0.039
		1955	1956	1957	1958	1959	1960	1961
Is defexp%	%	3.42	3.52	5.12	5.73	7.10	6.99	8.03
Eg defexp%	%	7.50	8.88	16.38	13.76	12.83	11.11	10.04
Sy defexp%	%	2.71	3.18	3.47	6.40	6.37	6.18	5.06
Av defexp%	%	4.54	5.19	8.32	8.63	8.77	8.09	7.71
Signs of *a - c*								
GA-33app for ME								
av defexp%	65	66	67	68	69	70	71	72
minuses	15	15	4	11	15	14	14	14

THE MIDDLE EAST, 1961 to 1967

							Appendix 6A	**3**
STATISTICAL INFORMATION			**MIDDLE EAST**				**Sheet**	**2**
	Year	**1961**	**1962**	**1963**	**1964**	**1965**	**1966**	**1967**
Israel GDP	Shekel bn	4.40	5.60	6.10	6.30	9.20	10.50	10.20
conversion	sh to $1	2.30	3.00	3.00	3.00	3.00	3.00	3.50
GDP	$bn	1.91	1.87	2.03	2.10	3.07	3.50	2.91
Defexp	Shekel bn	0.32	0.41	0.55	0.75	0.76	1.17	0.96
Defexp	$bn	0.14	0.14	0.18	0.25	0.25	0.39	0.27
defexp%	%	7.16	7.32	8.93	11.90	8.26	11.14	9.43
Egypt GDP	£E million	1469	1513	1685	2051	2403	2481	2533
conversion	£E to $1	0.35	0.43	0.43	0.43	0.43	0.43	0.43
GDP	$bn	4.17	3.48	3.88	4.35	4.38	4.70	5.04
Defexp	£E bn	234	239	261	261	233.45	185.1	215.7
Defexp	$bn	0.66	0.55	0.60	0.60	0.47	0.43	0.50
defexp%	%	15.94	15.82	15.50	13.82	10.80	9.07	9.85
Syria GDP	£S million	3396	3712	3980	4596	4614	4698	5437
conversion	£S to $1	3.58	3.58	3.58	3.82	3.82	3.82	3.82
GDP	$bn	0.95	1.04	1.11	1.20	0.97	1.09	1.32
Defexp	£S bn	234	239	261	261	310	365	587
Defexp	$bn	0.065	0.067	0.073	0.068	0.077	0.083	0.096
defexp%	%	6.89	6.44	6.56	5.68	7.91	7.64	7.30
AVERAGE defexp%		10.00	9.86	10.33	10.47	8.99	9.28	8.86
		1962	1963	1964	1965	1966	1967	1968
Is defexp	$bn	0.14	0.14	0.18	0.25	0.25	0.39	0.27
Eg defexp	$bn	0.66	0.55	0.60	0.60	0.47	0.43	0.50
Sy defexp	$bn	0.065	0.067	0.073	0.068	0.077	0.083	0.096
		1962	1963	1964	1965	1966	1967	1968
Is defexp%	%	7.16	7.32	8.93	11.90	8.26	11.14	9.43
Eg defexp%	%	15.94	15.82	15.50	13.82	10.80	9.07	9.85
Sy defexp%	%	6.89	6.44	6.56	5.68	7.91	7.64	7.30
Av defexp%	%	10.00	9.86	10.33	10.47	8.99	9.28	8.86
Signs of *a - c*								
GA-33app for ME								
av defexp%		73	74	75	76	77	78	79
minuses		10	11	15	2	1	2	2

THE MIDDLE EAST, 1968 to 1974

						Appendix 6A		**3**
STATISTICAL INFORMATION			**MIDDLE EAST**				**Sheet**	**3**
	Year	**1968**	**1969**	**1970**	**1971**	**1972**	**1973**	**1974**
Israel GDP	Shekel bn	1.06	1.61	1.87	2.46	3.20	4.18	5.60
conversion	sh to $1	3.50	3.50	3.50	4.20	4.20	4.20	6.00
GDP	$bn	3.85	4.57	5.37	5.57	7.17	9.25	0.93
Defexp	Shekel bn	0.85	0.98	1.34	1.91	1.94	1.98	1.53
Defexp	$bn	0.28	0.40	0.45	0.66	0.66	1.02	0.26
defexp%	%	7.30	8.64	8.45	11.83	9.19	10.97	27.36
Egypt GDP	£E million	2696	2971	3146	3337	3417	3663	4190
conversion	£E to $1	0.43	0.43	0.43	0.43	0.43	0.39	0.39
GDP	$bn	5.69	6.20	6.83	7.37	7.86	8.43	10.72
Defexp	£E bn	220.5	229.7	486.3	772.5	1289	2431	3175
Defexp	$bn	0.51	0.53	0.57	0.69	0.70	0.79	0.81
defexp%	%	8.92	8.52	8.33	9.37	8.96	9.32	7.56
Syria GDP	£S million	5514	5947	6433	7562	8891	9404	14870
conversion	£S to $1	3.82	3.82	3.82	3.82	3.82	3.80	3.70
GDP	$bn	1.57	1.75	1.78	2.10	2.43	3.47	4.02
Defexp	£S bn	600	624	676	793	1485	1682	2174
Defexp	$bn	0.154	0.157	0.161	0.164	0.208	0.389	0.588
defexp%	%	9.81	8.98	9.04	7.82	8.57	11.21	14.62
AVERAGE defexp%		8.67	8.71	8.61	9.67	8.91	10.50	16.51
		1969	1970	1971	1972	1973	1974	1975
Is defexp	$bn	0.28	0.40	0.45	0.66	0.66	1.02	0.26
Eg defexp	$bn	0.51	0.53	0.57	0.69	0.70	0.79	0.81
Sy defexp	$bn	0.154	0.157	0.161	0.164	0.208	0.389	0.588
		1969	1970	1971	1972	1973	1974	1975
Is defexp%	%	7.30	8.64	8.45	11.83	9.19	10.97	27.36
Eg defexp%	%	8.92	8.52	8.33	9.37	8.96	9.32	7.56
Sy defexp%	%	9.81	8.98	9.04	7.82	8.57	11.21	14.62
Av defexp%	%	8.67	8.71	8.61	9.67	8.91	10.50	16.51
Signs of *a - c*								
GA-33app for ME								
av defexp%		80	81	82	83	84	85	86
minuses		15	15	10	13	10	15	9

THE MIDDLE EAST, 1975 to 1981

						Appendix 6A		**3**
STATISTICAL INFORMATION			**MIDDLE EAST**				**Sheet**	**4**
	Year	**1975**	**1976**	**1977**	**1978**	**1979**	**1980**	**1981**
Israel GDP	Shekel bn	7.19	9.89	14.39	23.75	44.49	111.61	265.15
conversion	sh to $1	7.10	8.70	15.40	19.00	35.30	75.40	156.00
GDP	$bn	1.01	1.14	0.93	1.25	1.26	1.48	1.70
Defexp	Shekel bn	2.07	2.72	3.16	4.14	10.50	23.60	53.20
Defexp	$bn	0.29	0.31	0.21	0.22	0.30	0.31	0.34
defexp%	%	28.83	27.52	21.97	17.41	23.60	21.14	20.06
Egypt GDP	£E million	4886	6726	8210	9788	12610	15470	17150
conversion	£E to $1	0.39	0.39	0.39	0.39	0.70	0.70	0.70
GDP	$bn	12.50	17.20	21.00	25.03	18.01	22.1	24.5
Defexp	£E bn	3345	3778	4160	4740	8287	8145	9378
Defexp	$bn	0.86	0.97	1.06	1.21	1.16	1.16	1.34
defexp%	%	6.88	5.64	5.05	4.83	6.44	5.25	5.47
Syria GDP	£S million	20597	24725	27013	32389	38974	51270	65777
conversion	£S to $1	3.93	3.93	3.93	3.93	3.93	3.93	3.93
GDP	$bn	5.25	6.30	6.88	8.25	9.93	13.06	16.76
Defexp	£S bn	3345	3778	4160	4740	8287	8145	9378
Defexp	$bn	0.85	0.96	1.06	1.21	2.11	2.08	2.39
defexp%	%	16.24	15.28	15.40	14.63	21.26	15.89	14.26
AVERAGE defexp%		17.32	16.15	14.14	12.29	17.10	14.09	13.26
		1976	1977	1978	1979	1980	1981	1982
Is defexp	$bn	0.29	0.31	0.21	0.22	0.30	0.31	0.34
Eg defexp	$bn	0.86	0.97	1.06	1.21	1.16	1.16	1.34
Sy defexp	$bn	0.85	0.96	1.06	1.21	2.11	2.08	2.39
		1976	1977	1978	1979	1980	1981	1982
Is defexp%	%	28.83	27.52	21.97	17.41	23.60	21.14	20.06
Eg defexp%	%	6.88	5.64	5.05	4.83	6.44	5.25	5.47
Sy defexp%	%	16.24	15.28	15.40	14.63	21.26	15.89	14.26
Av defexp%	%	17.32	16.15	14.14	12.29	17.10	14.09	13.26
Signs of *a* - *c*								
GA-33app for ME								
av defexp%		87	88	89	90	91	92	93
minuses		13	0	2	15	15	15	14

THE MIDDLE EAST, 1982 to 1988

							Appendix 6A	**3**
STATISTICAL INFORMATION			**MIDDLE EAST**				**Sheet**	**5**
	Year	**1982**	**1983**	**1984**	**1985**	**1986**	**1987**	**1988**
Israel GDP	Shekel bn	596.31	1542	7636	28437	44191	56572	70181
conversion	sh to $1	333.70	1078	638.7	1499	1480	1530	1685
GDP	$bn	1.79	1.43	11.96	18.97	29.86	36.98	41.65
Defexp	Shekel bn	113	309	1626	4055	4396	4720	
Defexp	$bn	0.34	0.29	2.55	2.71	2.97	3.08	3.70
defexp%	%	18.95	20.04	21.29	14.26	9.95	8.34	8.88
Egypt GDP	£E million	20881	24834	27886	32516	36039	45249	54553
conversion	£E to $1	0.70	0.70	0.70	0.70	0.70	0.70	0.70
GDP	$bn	29.83	35.48	39.84	21.0	25.0	18.0	22.1
Defexp	£E bn	9778	10729	11309	12601	13000	13600	15500
Defexp	$bn	1.40	1.53	1.62	1.80	1.86	1.94	2.21
defexp%	%	4.69	4.31	4.07	8.57	7.43	10.77	10.00
Syria GDP	£S million	68788	73291	75342	83225	99933	127712	186047
conversion	£S to $1	3.93	3.93	3.93	3.93	3.93	3.93	11.23
GDP	$bn	17.53	18.67	19.20	21.20	25.46	32.54	16.57
Defexp	£S bn	9778	10729	11309	12601	13000	13600	15500
Defexp	$bn	2.49	2.73	2.88	3.21	3.31	3.46	1.38
defexp%	%	14.21	14.64	15.01	15.14	13.01	10.65	8.33
AVERAGE defexp%		12.62	13.00	13.46	12.66	10.13	9.92	9.07
		1983	1984	1985	1986	1987	1988	1989
Is defexp	$bn	0.34	0.29	2.55	2.71	2.97	3.08	3.70
Eg defexp	$bn	1.40	1.53	1.62	1.80	1.86	1.94	2.21
Sy defexp	$bn	2.49	2.73	2.88	3.21	3.31	3.46	1.38
		1983	1984	1985	1986	1987	1988	1989
Is defexp%	%	18.95	20.04	21.29	14.26	9.95	8.34	8.88
Eg defexp%	%	4.69	4.31	4.07	8.57	7.43	10.77	10.00
Sy defexp%	%	14.21	14.64	15.01	15.14	13.01	10.65	8.33
Av defexp%	%	12.62	13.00	13.46	12.66	10.13	9.92	9.07
Signs of *a - c*								
GA-33app for ME								
av defexp%		94	95	96	97	98	99	100
minuses		1	0	6	3	4	15	0

THE MIDDLE EAST, 1989 to 1995

						Appendix 6A		**3**
STATISTICAL INFORMATION			**MIDDLE EAST**				**Sheet**	**6**
	Year	**1989**	**1990**	**1991**	**1992**	**1993**	**1994**	**1995**
Israel GDP	Shekel bn	85471	105831	134855	161738	186576	224838	261586
conversion	sh to $1	1963	2048	2280	2764	2986	3018	3135
GDP	$bn	43.54	51.68	59.15	58.52	62.48	74.50	83.44
Defexp	Shekel bn							
Defexp	$bn	3.90	4.60	5.10	6.20	7.50	8.40	9.78
defexp%	%	8.96	8.90	8.62	10.60	12.00	11.28	11.72
Egypt GDP	£E million	65577	96100	105000	139100	157000	175000	205000
conversion	£E to $1	1.10	2.00	3.30	3.38	3.37	3.39	3.39
GDP	$bn	24.5	29.8	35.5	39.8	46.5	51.5	64.6
Defexp	£E bn	18000						5892
Defexp	$bn	1.64	1.67	1.71	1.68	1.70	1.72	1.74
defexp%	%	6.69	5.60	4.82	4.22	3.66	3.34	2.69
Syria GDP	£S million	208892	278638	311564	371630	413755	506101	570975
conversion	£S to $1	11.23	11.23	11.23	11.23	11.23	11.23	11.23
GDP	$bn	18.61	24.82	27.76	33.11	36.86	45.09	50.87
Defexp	£S bn	18000						
Defexp	$bn	1.60	1.80	1.80	1.90	1.90	2.70	3.98
defexp%	%	8.62	7.25	6.49	5.74	5.15	5.99	7.82
AVERAGE defexp%		8.09	7.25	6.64	6.85	6.94	6.87	7.41
		1990	1991	1992	1993	1994	1995	1996
Is defexp	$bn	3.90	4.60	5.10	6.20	7.50	8.40	9.78
Eg defexp	$bn	1.64	1.67	2.71	1.68	1.70	1.72	1.74
Sy defexp	$bn	1.60	1.80	1.80	1.90	1.90	2.70	3.98
		1990	1991	1992	1993	1994	1995	1996
Is defexp%	%	8.96	8.90	8.62	10.60	12.00	11.28	11.72
Eg defexp%	%	6.69	5.60	4.82	4.22	3.66	3.34	2.69
Sy defexp%	%	8.62	7.25	6.49	5.74	5.15	5.99	7.82
Av defexp%	%	8.09	7.25	6.64	6.85	6.94	6.87	7.41
Signs of *a - c*								
GA-33app for ME								
av defexp%		101	102	103	104	105	106	107
minuses		6	15	13	7	15	6	12

THE MIDDLE EAST, 1996 to 2000

						Appendix	**3**
STATISTICAL INFORMATION						**Sheet**	**7**
	Year	**1996**	**1997**	**1998**	**1999**	**2000**	
Israel GDP	Shekel bn	304682	339992	382795	417446	463055	
conversion	sh to $1	3250	3530	4160	4435	4872	
GDP	$bn	93.75	96.32	92.02	94.13	95.04	
Defexp	Shekel bn						
Defexp	$bn	10.50	10.80	11.35	11.72	11.95	
defexp%	%	11.20	11.21	12.33	12.45	12.57	
Egypt GDP	£E million	228130	256250	280220	305000	322400	
conversion	£E to $1	3.39	3.39	3.40	3.40	3.40	
GDP	$bn	77.9	59.6	48.1	44.1	41.2	
Defexp	£E bn						
Defexp	$bn	1.80	1.85	1.98	2.28	2.45	
defexp%	%	2.31	3.10	4.12	5.16	5.95	
Syria GDP	£S million	676441	728294	775786	822790	879376	
conversion	£S to $1	11.23	11.23	11.23	11.23	11.23	
GDP	$bn	60.26	64.88	69.11	73.30	78.34	
Defexp	£S bn						
Defexp	$bn	4.09	4.09	5.06	5.51	6.03	
defexp%	%	6.79	6.30	7.32	7.52	7.70	
AVERAGE defexp%		6.77	6.87	7.93	8.38	8.74	
		1997	1998	1999	2000	2001	
Is defexp	$bn	10.50	10.80	11.35	11.72	11.95	
Eg defexp	$bn	1.80	1.85	1.98	2.28	2.45	
Sy defexp	$bn	4.09	4.09	5.06	5.51	5.66	
		1997	1998	1999	2000	2001	
Is defexp%	%	11.20	11.21	12.33	12.45	12.57	
Eg defexp%	%	2.31	3.10	4.12	5.16	5.95	
Sy defexp%	%	6.79	6.30	7.32	7.52	7.70	
Av defexp%	%	6.77	6.87	7.93	8.38	8.74	
Signs of *a - c*							
GA-33app for ME							
av defexp%		108	109	110	111	112	
minuses		12	8	15	14	10	

CHAPTER 7

The Stability of Nuclear Deterrence

7.1 Introduction

This chapter applies the Richardson arms race equations to nuclear deterrence using GAs. It sets the scene with the Strategic Arms Limitation Talks of 1972 and the very limited means of calculation available then. From there, it applies GAs to the problem. The GAs used are quite different from those used in Chapters 5 and 6 to forecast instabilities between nations. Here, given the number of nuclear missiles believed to be held by (what was then) the Soviet Union, the GA is required to determine how many missiles the United States has to field if nuclear deterrence is to be achieved. This is not a trivial calculation for the effectiveness of nuclear weapons (their *destructive capability*) depends not only on their size and the numbers deployed, but also on the probability of their timely and accurate delivery – and there are many types of weapons (some of which, the *MIRV*s, house more than one warhead) and at least three types of delivery platform. Furthermore, the Americans always assumed that the Russians would strike first, so the number of missiles available to the Americans to ensure deterrence had to be that which *survived* an initial Soviet attack. The GA's evaluation equations are based on the physics of aerodynamic shock (nuclear blast), and the fitness function aims to minimise the total number of nuclear weapons available to both sides. It is shown that, with their ability to search huge parameter spaces efficiently, nuclear deterrence GAs are highly adaptable and permit the investigation of many current problems of the '*what if ...* ' variety, e.g. "What happens if our opponents' missile delivery probabilities have been under-estimated and our own over-estimated?" or "Would we be overwhelmed if our opponents mounted a surprise attack?"

Our initial GAs were far from robust; a scatter of possible answers was generated, but a Pareto-sort reduced their number dramatically. To enhance performance, redundancy and a modest elitism were introduced. After trial and error, the GAs became much more robust.

The remainder of this chapter covers SALT and explains in depth how we have approached the problem (both by our check on the Saaty/Dalkey model by iterative methods in 1972, and by GAs very recently). The results of a number of trials, by varying the delivery probabilities of both sides and by discretising the evaluation equations, are presented.

The chapter concludes by comparing the results obtained by some traditional numerical methods, such as Gauss-Seidel, Jacobi and Monte Carlo, with those obtained via GAs; it then explains why GAs are used in preference to these other methods, and discusses the use of GAs in this type of application.

Appendices cover the basics of aerodynamic shock and the derivation of delivery probabilities.

7.2 The Strategic Arms Limitation Talks

In May 1972, as a culmination of SALT, President Nixon and Mr Brezhnev signed an Agreement aimed at limiting the number of strategic nuclear weapons held by the United States and the Soviet Union. One of a number of calculations done at the time in the UK to verify SALT made use of the Saaty/Dalkey model for strategic nuclear deterrence. (The Americans, in contrast, are said to have worked their calculations in terms of the classic zero-sum Game Theory of Von Neumann and Morgenstern (1947).)

The Saaty/Dalkey model is based on (and is a variant of) the Richardson arms race equations. The model needs a huge search space and so it was not very practical until the advent of evolutionary techniques such as GAs.

As we said in Chapter 3, the problem with the Richardson arms race equations is that they are conceptually accurate but difficult to compute in practice. To achieve the latter, they have to be transformed.

Let the annual defence expenditures of Nation X and Nation Y be US\$ x billion and US\$ y billion, respectively. Then, from Richardson's theory,

$$\frac{dy}{dt} = ax - by + h \qquad \text{and} \qquad \frac{dx}{dt} = cy - fx + g \ , \tag{7.1}$$

where a and c are called *defence coefficients*, b and f are called *fatigue and expense coefficients*, while h and g are called *grievances* when they are positive, and *goodwill* when they are negative. It should be recalled that all the coefficients are functions of time.

Richardson [1960(b)] had made three assumptions:

- That each country would attempt to increase its armaments over its opponents.
- That economic factors impose constraints that tend to diminish the rate of increase by an amount proportional to the size of the existing friendly forces, and that
- A nation will build arms motivated by ambition and hostility even if other nations pose no threat to it.

7.3 The Saaty/Dalkey Model

Recognising that the x and y in Eqs(7.1) could be used to represent any means of portraying an arms race (not only money), Saaty and Dalkey [Saaty and Alexander, 1981] were subsequently able to modify Richardson's theory, so that it could simulate the threatened use of missiles between two antagonists, the United States and the Soviet Union, to deter each other from initiating war. They did this by re-defining and limiting the problem, and then substituting a transform expressed in terms of the available parameters (see Section 7.5).

If y is the number of US missiles and x the number of Soviet missiles, then the rates of growth of y and x are given by

$$\frac{dy}{dt} = ax - by + \hat{k}_1 \quad \text{and} \quad \frac{dx}{dt} = cy - fx + \hat{k}_2 \ , \tag{7.2}$$

where $\hat{k}_1 \equiv \hat{k}_1(t),\ \hat{k}_2 \equiv \hat{k}_2(t)$ stand for the numbers of missiles which will ensure that even if neither side has plans to acquire further missiles, there still exist $\hat{k}_1$ and $\hat{k}_2$ missiles, respectively, to ensure the destruction of the other side's cities.

Saaty and Dalkey, however, made five further assumptions in addition to the three already listed by Richardson**:**

- Each side will attempt to destroy the missile system of the other side before starting to attack the other's cities.
- Each ICBM has unlimited range and, similarly, submarines armed with SLBMs will be so located that they can attack any target for which they are likely to be tasked.
- ABMs are limited in range so that ABMs assigned to defend the capital city cannot be used to defend an ICBM field and vice versa. The SALT Agreement stipulates that each side may only field 100 ABMs.
- Both the United States and the Soviet Union believe that the SALT Agreement is to their advantage and that therefore a stable situation exists at most times.
- Initially, each missile has only one warhead .

The value of each missile, its *destructive capability*, is assessed by its owner in terms of the damage it can inflict on the opponent's missile sites or cities if war has to be declared. The value depends, not only on the weapon's size, but also on the probability of its delivery.

Eqs(7.2) can be represented by two curved contours, each plotted against generations. The regions *outside* the contours (i.e. above the red and to the right of the green) give the minimum number of missiles with which the United States and the Soviet Union, respectively, can inflict so-called 'unacceptable damage' on each other. If the curves are plotted on equally-scaled axes, the region outside *both* curves is the 'region of stability'. By stockpiling missiles up to the numerical levels indicated by the region of stability, both the United States and the Soviet Union have the ability to deter each other from initiating war.

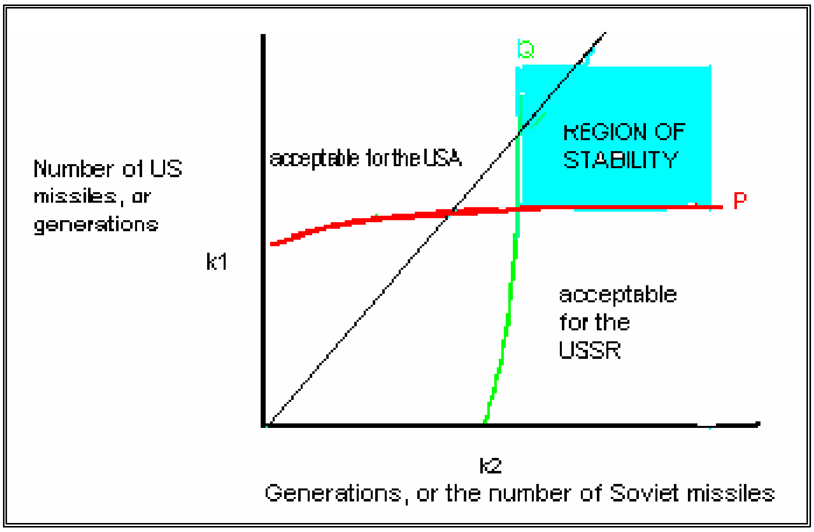


Figure 7.1 Region of Stability**;** curved contours against generations on equally-scaled axes. The P contour is drawn in red, the Q contour in green

This is illustrated in Figure 7.1, where the 45° line represents missile parity. Neither side, however, regards parity as an issue. Instead, each wants to be able to destroy a certain proportion of the other's cities, the proportion depending on what each considers the other's criterion of *unacceptable damage* to be. The US contour represents the number of missiles which will guarantee levels of unacceptable damage on the Soviet Union *after* they have survived an initial Soviet attack. This is called the 'counter-value'.

Let the American counter-value be P and the Soviet counter-value be Q. It should be observed that we match *destructive capability*; actual numbers of missiles on either side can be quite different.

Given a contour for P, the United States can now estimate the number of missiles which will survive a Soviet attack. Armed with a number of missiles greater than that indicated by the contour for P (i.e. above the curve) the United States can ensure the virtual destruction of the Soviet Union. Similarly the latter can do likewise to America, provided her holding of missiles is to the right of the Q contour.

Any solution is, therefore, required:

- To establish contours for the counter-values P and Q using some of the physics of nuclear explosions (Appendix 7A), a set of delivery probabilities (Appendix 7B), and known and estimated facts about the weapons holdings of either side (Table 7.1).
- Having determined a specific value for P, to establish the corresponding value for Q.
- To define the number of America's missiles which survive solely in terms of the Soviet Union's destructive capability and vice versa.
- To seek the 'fittest' solution, which in this context means minimising the total number of nuclear missiles available to both sides.

We next restrict the problem and try to solve it practically.

7.4 Initialising Real-World Data

Table 7.1 lists actual holdings of strategic nuclear weapons in the early 1970s for the United States and the Soviet Union. (The figures for the Soviet Union were the best that could be derived from intelligence sources; it will be appreciated that at the time all these figures would have been accorded very high security classification). M stands for the number of American missiles and W for the number of warheads each missile can carry; m and w stand for those of the Soviet Union.

Table 7.1 Holdings of nuclear weapons. The figures date from 1972

	UNITED STATES			SOVIET UNION		
Type	Missiles or aircraft	Nuclear warheads	Probabilities of delivery	Missiles or aircraft	Nuclear warheads	Probabilities of delivery
SLBM	656 (M_1)	1808 (M_1W_1)	0.72 (μ_1)	740 (m_1)	740 ($m_1 w_1$)	0.76 (v_1)
ICBM	1054 (M_2)	1454 (M_2W_2)	0.60 (μ_2)	1618(m_2)	1618($m_2 w_2$)	0.75 (v_2)
Aircraft-Delivered	520 (M_3) *Note* 4	3134 (M_3W_3)	0.33 (μ_3)	140 (m_3) *Note* 4	140 ($m_3 w_3$)	0.40 (v_3)
		Notes 1 *and* 2	*Note* 3		*Notes* 1 *and* 2	*Note* 3
Totals	2230 (M)	6396 (MW)		2498 (m)	2498 (mw)	

Note 1. Those working in the nuclear deterrence field conventionally use T or t to denote the number of warheads per weapon/aircraft. In the thesis, we use W or w to avoid confusion with time

Note 2. The US had the ability to put several warheads into each missile or aircraft, but this was not shared in 1972 by the Soviet Union. (Hence, in 1972, $w_1 = w_2 = w_3 = w = 1$, although for later years we must give them different values as the Soviets did ultimately achieve a MIRVing capability.) Several types/marks of missile are included in each total, so the overall MW/M ratio is not nec-essarily an integer

Note 3. These probabilities were not casual 'guesstimates'. The Americans had put much separate work into their derivation

Note 4. In the context of Richardson's equations and of this chapter, a manned aircraft carrying nuclear warheads counts as a missile. The only exception to this is in Section 7.16

7.5 A General Approach to the Problem

In essence, nuclear deterrence is represented by a set of equations. If a small change to these equations results in a divergent system then it is argued that the SALT Agreement would become unstable. Conversely, if the equations converge (see Section 2.13) it can be concluded that stable deterrence has been achieved.

We next assume that each of the 100 ABMs permitted to each side by the SALT Agreement can be guaranteed to *counter*[1] a missile.

The number of Soviet *missiles* is m of which the Americans can destroy the first 100 with their ABMs. Hence the number of Soviet *warheads* destroyed will (probably) be $(m - 100)w$. The fraction of Soviet warheads assigned to destroy each American missile *on the ground* is therefore $(m - 100)w/M$. Correspondingly, the fraction of American warheads assigned to destroy each Soviet missile on the ground is $(M - 100)W/m$.

Let μ be the probability that a Soviet missile is destroyed on the ground by an American warhead, and let v be the corresponding probability for the destruction of an American missile. It is shown in Appendix 7A that the probability of destruction of a Soviet missile by an American warhead is $\mu / W^{1/3}$ and the corresponding probability for the destruction of an American missile by a Soviet warhead is $v / w^{1/3}$. The probability that an American missile on the ground *survives* an attack by a Soviet warhead is $1 - v / w^{1/3}$. (The probability that it survives two such attacks is $(1 - v / w^{1/3})^2$.) It follows, therefore, that the probability that *one* American missile on the ground will survive a series of attacks by $(m - 100)w / M$ warheads is

$$\left(1 - \frac{v}{w^{1/3}}\right)^{\frac{(m-100)w}{M}} \tag{7.3}$$

and that the expected total number of American missiles which will survive a Soviet onslaught, *which by definition is the American counter-value P*, is given by

$$P = M\left(1 - \frac{v}{w^{1/3}}\right)^{\frac{(m-100)w}{M}} . \tag{7.4}$$

Likewise, the Soviet counter-value, Q, is given by

$$Q = m\left(1 - \frac{\mu}{W^{1/3}}\right)^{\frac{(M-100)W}{m}} . \tag{7.5}$$

---

[1] 'Counter' means destroy the solid-state chips, etc in the missile by very steep gradients of electromagnetic pulse (EMP) generated by the ABM. Blast needs air, so it is not effective at the sort of altitudes that ABMs intercept incoming missiles

Furthermore, account has to be taken in the model of two or more different missile systems which may have varying numbers of multiple warheads.

If v_1, v_2 and v_3 are three different probabilities that American missiles will be destroyed by Soviet warheads delivered by three different systems, the American missiles' chances of survival from separate attacks are $1-v_1$, $1-v_2$ and $1-v_3$, respectively. The chances of *destruction* by a combined attack consisting of three warheads delivered in turn by each of the three delivery systems will be $1 - (1-v_1)(1-v_2)(1-v_3)$, and their chance of *survival* will be

$$(1-v_1)(1-v_2)(1-v_3)\,.$$

If, for $i = 1, 2, 3$, $m_i w_i$ and $v_i w_i$ correspond to $M_i W_i$ and $\mu_i W_i$, respectively, then

$$P = M\left(\left(1-\frac{v_1}{w_1^{1/3}}\right)^{(m_1-50)w_1}\left(1-\frac{v_2}{w_2^{1/3}}\right)^{(m_2-50)w_2}\left(1-\frac{v_3}{w_3^{1/3}}\right)^{m_3 w_3}\right)^{\frac{1}{M}} \tag{7.6}$$

and

$$Q = m\left(\left(1-\frac{\mu_1}{W_1^{1/3}}\right)^{(M_1-50)W_1}\left(1-\frac{\mu_2}{W_2^{1/3}}\right)^{(M_2-50)W_2}\left(1-\frac{\mu_3}{W_3^{1/3}}\right)^{M_3 W_3}\right)^{\frac{1}{m}} \quad . \tag{7.7}$$

Eq(7.6) can be written in the form

$$P = M j_1^{n_1/M}\, j_2^{n_2/M}\, j_3^{n_3/M} = \beta M, \tag{7.8}$$

where $j_i = 1 - v_i/w_i^{1/3}$, $n_i = \big(m_i - 50\ (0)\big)w_i$, and $\beta = j_1^{n_1/M}\, j_2^{n_2/M}\, j_3^{n_3/M}$.

(*Note.* In writing β in this way, we should not forget the literal meaning of β as part of $P = \beta M$. If, for instance, $\beta = 0.75$, we are saying that a quarter of the American missiles will be destroyed by a Soviet first strike.)

We observe in passing that, since $P < M$, it must be the case that $0 < \beta < 1$. It can be verified that for all β

$$\frac{dP}{dM} = j_1^{n_1/M}\, j_2^{n_2/M}\, j_3^{n_3/M}\left(1+\frac{\left|n_1 \ln j_1 + n_2 \ln j_2 + n_3 \ln j_3\right|}{M}\right) = \beta\big(1+\left|\ln\beta\right|\big) < 1 \ . \tag{7.9}$$

We next consider the first-order iterative method given by Eq(7.12). For convergence to take place (see Theorem 3.6 in [Epperson, 2002]), it is sufficient that the modulus of the first derivative of f in the neighbourhood of the fixed point is less than one. Since

$$\frac{df}{dM_i} = \frac{n}{M_i}\ln k,$$

on using Eq(7.8), it follows that convergence takes place if

$$M > \left| (m_1 - 50) w_1 \ln\left(1 - \frac{v_1}{w_1^{1/3}}\right) + (m_2 - 50) w_2 \ln\left(1 - \frac{v_2}{w_2^{1/3}}\right) + m_3 w_3 \ln\left(1 - \frac{v_3}{w_3^{1/3}}\right) \right| . \qquad (7.10)$$

In such a case, *the SALT Agreement is stable* for the United States (see the introductory paragraph for this Section). Correspondingly, for the Soviet Union we have

$$m > \left| (M_1 - 50) W_1 \ln\left(1 - \frac{\mu_1}{W_1^{1/3}}\right) + (M_2 - 50) W_2 \ln\left(1 - \frac{\mu_2}{W_2^{1/3}}\right) + M_3 W_3 \ln\left(1 - \frac{\mu_3}{W_3^{1/3}}\right) \right| . \qquad (7.11)$$

The above two inequalities, Eq(7.10) and Eq(7.11), feature prominently in the subsequent development of GAs applied to the nuclear deterrence problem.

7.6 Approaching the Nuclear Deterrence Problem in 1972

We were tasked in 1972 by the UK Ministry of Defence [Hackworth, 1973] to verify the outcome of SALT. By the early 1970s much had been written about deterrence, not all of it consistent or pragmatic.

RE Jones[2] had defined deterrence as a calculated attempt to induce an adversary to do something or refrain from doing something by threatening retaliation for non-compliance. Let G be the gain desired by the United States and p_1 be the probability of its attainment *without the overt use of force*. Then the expected gain for the US would be p_1G. If war had, however, to be declared first, the gain would instead be some lesser value, which we will call $\overline{\overline{p_1 G}}$. Moreover, let D be the American capability for destruction and p_2 be the probability of its being used against the Soviet Union to prevent, in turn, gains against America. In the event of war, the USA's expected level of destruction in the Soviet Union would be p_2D. Let p_3, g, d, p_4 and $\overline{\overline{p_3 g}}$ be the corresponding values for the Soviet Union. Jones showed that the *gain-deterrence ratio* for the United States, R, was given by

$$R = \frac{p_1 G (1 - p_4) + \overline{\overline{p_1 G}}\, p_4}{p_4 d} .$$

(whence $R \approx \dfrac{p_1 G}{p_4 d}$ for sufficiently large d) .

while for the Soviet Union, the deterrence ratio, r, was given by .

$$r = \frac{p_3 g (1 - p_2) + \overline{\overline{p_3 g}}\, p_2}{p_2 D} .$$

---

[2] It has not been possible to locate current references for the reports of either Jones or Ellsberg (see over); both were originally classified internal reports for establishments of the US Department of Defense, and both would now be over 30 years old

(whence $r \approx \dfrac{p_3 g}{p_2 D}$ for sufficiently large D).

Although this easily led to some neat qualitative arguments about what happened and who deterred whom when $R \ll 1$ and $r \ll 1$, for instance, the theory was extraordinarily difficult to quantify, largely because certain terms were vague; for example, what precisely counted as a gain, or would a border skirmish count as a war?

Taking a totally different approach, Daniel Ellsberg [2] treated deterrence as a decision-making system with matrices of time-dependent utilities; he argued that it should be Washington's goal to ensure that at all times Moscow believed that the advantages gained by striking, V_{strike}, were inferior to those gained by waiting, V_{wait}. The worth of the deterrent depended on the width of the 'margin'

$$V_{wait} - V_{strike}.$$

Again, this was not easy to quantify with real-world parameters.

Compared with these analyses, Richardson's arms race equations appeared very straightforward, although some transformations were needed first.

Our preferred solution at the time was to adapt Richardson's equations and to rearrange Eq(7.8) into an iterative form, namely

$$M_{i+1} = f\left(M_i\right) = k^{-\frac{n}{M_i}} P \quad , \tag{7.12}$$

where $k^{-n} = e^{-(n_1 \ln(j_1) + n_2 \ln(j_2) + n_3 \ln(j_3))}$.

This converges to a solution *provided that* the initial approximation is numerically close to a fixed point. This meant that for our initial value, M_0, we had to make extensive use of domain knowledge because a casual guess would not suffice[3]. According to our 1972 calculations, using Eq(7.12), the total number of American missiles had to be greater (see Eq(7.10)) than 2354 if stability (nuclear deterrence) was to be achieved. (In fact, they had 2230 at the time, although it is understood that this figure was later increased to about 3000 for reasons which are summarised in the *Note* to Section 7.15).

---

[3] From notes taken in 1972 we recall that Eq(7.12) will not converge if $M_0 < 2280$. This is within 3.1% of the answer

7.7 Applying an Evolution Strategy

As a first step in solving Eq(7.6) and Eq(7.7), we applied an ES and developed an algorithm, ES-1. Using an RNG, a chromosome was generated and one copy thereof was put into the first row of a two-row chromosome array, and another one into the second row. Both chromosomes use Eq(7.10) and Eq(7.11) as evaluation equations, the probabilities of Table 7.1, and the fitness function given by Eq(7.13). The fitness became an integral part of each chromosome. At Generation 0, obviously, the two rows would be identical. The second chromosome was then mutated and again compared with the first. The two chromosomes would now have different fitnesses. If the fitness of the second chromosome was not as good as that of the first, then the mutation process on the second chromosome would be repeated until its fitness was *better* than that of the first chromosome which it would then replace. The overall process is then repeated until ES-1 converges, see (2) below. This really is 'survival of the fittest', the advantage being that no time is wasted on unfit or infeasible solutions.

ES-1 had mutation as its only operator, as explained above. To facilitate comparison with the GAs in this chapter, ES-1 was developed so that the probabilities of delivery could be altered, and the full range of counter-value equations (Section 7.8.1) could be used. ES-1 was run with different start conditions (seeds ranging from 0 to 79). As a result, several lessons were learnt:

(1) Each run gave only one answer. Applying domain knowledge[4] eliminated some answers.

(2) Convergence often took place after a few generations to be followed by increasing divergence as mutation took its toll and became untenable. This was solved by setting the ES-1 program code to run for 50 generations but stopping it if a certain number of consecutive iterations had converged on the same value. The program typically stopped after 15 generations.

(3) It was observed that the first mutation generally gave a poorer fitness than the original.

(4) In some runs the original was, and remained, the best.

Eighty runs of ES-1 gave eighty answers but only thirteen of these satisfied the domain conditions. A Pareto sort reduced these to two, see Table 7.2 below.

---

[4] In this case, $P > Q > 0.65P$, and *fitness* > 2000

.

Table 7.2 The best answers from ES-1

algorithm	*seed*	*P*	*Q*	*fitness*
ES-1	11	3307	2024	2579
ES-1	73	2259	2237	3145

It is a lot easier to develop an ES than a GA, but the ES suffers from all the disadvantages traditionally associated with hill-climbing methods. To overcome these disadvantages, we abandoned the ES approach for nuclear deterrence, and turned instead to a GA-based approach. We did, however, return to ES-1

- to derive comparative figures for the *nightmare* scenario, see Section 7.12.2, and
- to help develop an algorithm for simulated annealing, see Section 5.8.2.

7. 8 Applying a GA

A canonical integer-valued GA, GA-19, was developed to see, initially, if the figures given in Table 7.1 appeared to be realistic from the viewpoint of both sides. The GA uses Eq(7.10) and Eq(7.11) as its evaluation equations, and its fitness function is given by Eq(7.13). The general configuration of its chromosomes is presented in Table 7.3. GA-19 fits the general description given in Chapter 4; in particular, several population sizes were tried and anything less than 50 gave inappropriate results. 100 was chosen as being safe but not excessive or wasteful.

7.8.1 Equation Sets

Four alternative pairs of equation sets were used, all giving contours for the counter-values P and Q. The probabilities μ and v are taken from Table 7.1.

- ***P*1 and *Q*1**. General expressions for P and Q (three delivery systems). The evaluation equations were Eq(7.6) and Eq(7.7). Delivery probabilities were taken from Table 7.1, but the numbers of warheads and missiles were generated randomly.

- ***P*2 and *Q*2**. As $P1$ and $Q1$, with actual numbers of warheads and missiles (Table 7.1).

- ***P*3 and *Q*3**. In this case the evaluation equations were Eq(7.10) and Eq(7.11). Delivery probabilities were taken from Table 7.1, but numbers of warheads and missiles were generated randomly.

- ***P*4 and *Q*4**. As $P3$ and Q3, with actual numbers of warheads and missiles (Table 7.1).

It should be observed that

- all these equations sets are reflexive**,** i.e. the number of America's missiles which survive a nuclear first strike is defined solely in terms of the destructive capability of the Soviet Union and vice versa, and

- anything derived from Eq(7.10) or Eq(7.11) is in fact M or m rather than P or Q. In order not to cause confusion, we will continue to refer to P (and, correspondingly, to Q) as the number of missiles required by either side to achieve deterrence.

7.8.2 Fitness Function

It is known that in the early 1970s both Americans and Russians wanted to minimise their combined holdings of nuclear weapons. This was not only due to the sheer cost of building and maintaining them but also to the growing realisation, even then, that

- the environment would suffer should the number of nuclear weapons get out of hand, and

- chaos would result if such weapons were to fall into irresponsible hands.

Our model must, therefore, follow suit and aim to minimise the *overall number* of nuclear weapons. To this end, the fitness function is given by

$$fitness = \frac{10000000}{\sqrt{1+P^2+Q^2}} \quad , \tag{7.13}$$

which we maximise. (The numerator is just a scaling factor).

7.8.3 Options

The evaluation equations can be computed by assigning to the variables M_1 , M_2 , M_3 , m_1 , m_2 and m_3 the values shown in Table 7.1. Alternatively, values assigned to the same variables can be generated randomly (within reasonable, constrained limits) and evolve during the execution of the GA under consideration.

- The initialisation of the chromosomes by the RNG took cognisance of known figures, i.e. the biggest number likely to appear in practice among the M's and m's in Table 7.1 is 1618, so the RNG was set to an upper bound of 1650.

- Rates of mutation and crossover could be (and often were) varied dynamically.

- Every run started with a seed in the range $0 \leq seed \leq 79$ which set the point at which the RNG started to feed the program emulating the GA. It was observed that the order of the concatenated parameters (in GA language, the *loci* of the genes in the chromosomes) did make a difference to the results obtained. This will be discussed later.

Table 7.3 shows the sequence of the concatenated parameters in GA-19 and GA-2. (GA-2 was developed later to overcome the shortcomings of GA-19, see Section 7.9.)

Table 7.3 Layout of Matrices 1 and 2 of GA-19 and GA-2

GA/col	0 *Note* 1	1 *Note* 1	2	3	4	5	6	7	8	9	10	11 *Note* 1	12
GA-19	*gen*	*srl*	M_1	M_2	M_3	m_1	m_2	m_3	*P*	*Q*	*fitness*	*norm-fit*	*running total*
GA- 2	*gen*	*srl*	M_1	m_1	M_2	m_2	M_3	m_3	*P*	*Q*	*fitness*	*norm-fit*	*running total*

Note 1. *gen* is generation. *srl* is serial. A *normfit* is a linearly normalised fitness, see Section 4.3

Note 2. Columns 2 - 7 inclusive form the 6-gene, 24-element (6×4) chromosomes

Table 7.4 Progress of the chromosome (shaded) of GA-2. This is shown in Figure 7.2; Seed 42

Generation	M_1	m_1	M_2	m_2	M_3	m_3	*P*	*Q*	*fitness*
0 (initiation)	0588	1220	1399	1340	1821	2411	4695	3555	1698
1	2221	0365	2916	0295	0137	1558	6884	2857	1341
2	0583	1224	1394	1343	0357	0037	5772	3277	1506
3	0583	1224	1394	1344	0287	3377	2375	5218	1744
4	0583	1224	1394	1344	0287	3377	3486	2417	2358
5	0583	1224	1394	1343	0357	0037	3486	2413	2357
6 (convergence)	0583	1224	1394	1343	0357	0037	3486	2413	2358
7 (convergence)	0583	1224	1394	1343	0357	0037	3486	2413	2358

7.9 Results from GA-19: Early Trials

We first used *Equation Set P*4 and *Q*4 (Section 7.8.1). This entails using *all* the data of Table 7.1 (including the numbers of weapons) and Eq(7.10) and Eq(7.11). The figures obtained are $P = 3229$ for the United States and $Q = 2573$ for the Soviet Union. This is straightforward arithmetic, *and no GA action is involved.*

We then compared these values of *P* and *Q* with those obtained from GA-19, again using Table 7.1, but generating the number of weapons randomly. Since both crossover and mutation changed dynamically, the answers never stabilised, e.g. successive generations could go through 3229 and 2573, and then wander off somewhere else. Fitness was over-dependent on actual figures, and this distorted the selection process. The choice of seed was crucial. Moreover, varying the location of genes within the chromosome could make a difference to the results obtained; it is considered that this was not so much due to epistasis as to the fact that conditions in the left-hand end of the chromosome string were more turbulent than in the right-hand end since the crossover took place on the left.

GA-19 was trialled in a number of variants, altering many parameter settings. Some were successful, others not, 'success' being reckoned by non-degenerative convergence (Section 2.13), the highest attainable fitness factor (in practice, greater than 2000), and figures *reasonably close to those desired*. Richardson's equations are bound to yield several different but mathematically legitimate answers, not all of which will be of practical use. In the real world, P is likely to lie fairly close to Q under 'normal' delivery conditions, so solutions where they are wildly apart can safely be ignored on using domain knowledge.

In order to overcome the shortcomings of GA-19, a new algorithm, GA-2, was developed.

The direct evaluation of *Equation Set* $P4$ and $Q4$ had yielded $P = 3229$ and $Q = 2573$. It was tempting to attempt a *hybrid* scheme; for example, using *actual* values of M_1, M_2 and M_3 to initialise the Matrices 1 and 2 of GA-19, and randomly generated ones to initialise m_1, m_2 and m_3. In practice this did not work at all well.

It was decided, therefore, to let the random number generator initialise the whole chromosome and then let the synergy of parent selection, mutation and crossover take their course. The improved results of this latter approach became apparent in the behaviour, outputs and robustness of the *nuclear deterrence suite* of GAs which we are about to introduce.

7.10 Suite of GAs for Nuclear Deterrence

A nuclear deterrence suite of GAs, containing the eight algorithms shown in Table 7.5, was developed. GA-2 was straightforward; GA-20 incorporated elitism; GA-22 added some redundant genes; and GA -24 had both. The same obtains for GA-18, GA-21, GA-23 and GA-25, respectively, all of which incorporated a Pareto-sort.

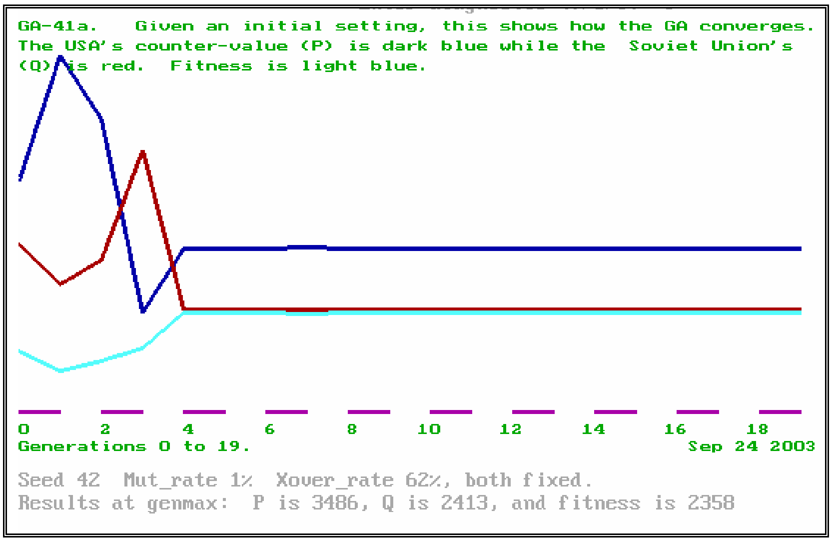
Table 7.5 Capabilities of the GA suite for nuclear deterrence

Name	*Genes*	*Vary probabilities of delivery*	*Redundant chromosomes*	*Elitist*	*Pareto-sort*
GA-2	6	yes	No	No	No
GA-18	6	yes	No	No	Yes
GA-20	6	yes	No	yes (1%)	No
GA-21	6	yes	No	yes (1%)	Yes
GA-22	9	yes	yes (33%)	No	No
GA-23	9	yes	yes (33%)	No	Yes
GA-24	9	yes	yes (33%)	yes (1%)	No
GA-25	9	yes	yes (33;%)	yes (1%)	Yes

Note 1. For the configuration of all six-gene chromosomes, see Table 7.3 (lower row); for those with nine genes, see Table 7.11

Note 2. All GAs of the nuclear deterrence suite used a population of a hundred chromosomes over twenty generations. Mutation and crossover were fixed at 1% and 62%, respectively

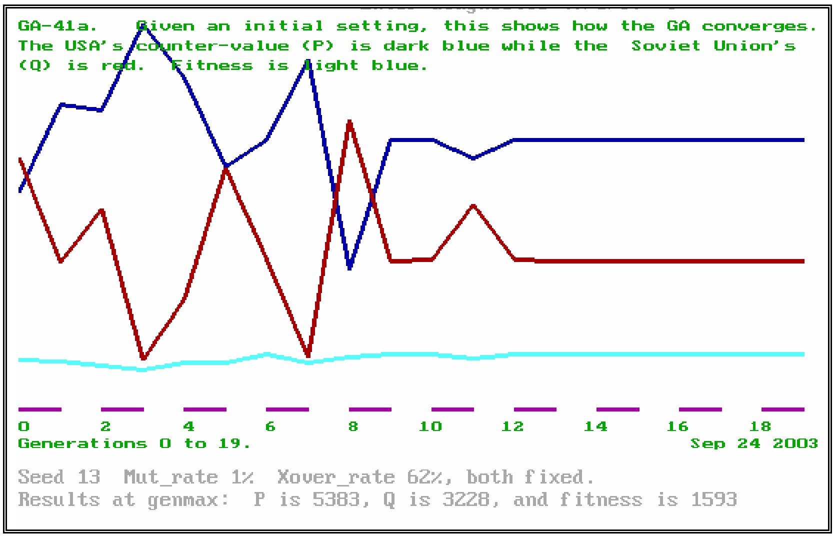
GA-2 was more robust than its predecessor, GA-19, and yielded P = 3486, Q = 2413 and *fitness* = 2358, as Table 7.4 shows. The values of P (blue) and Q (red) of Table 7.4 are shown graphically in Figure 7.2. GA-2 yielded different values for P and Q, depending on the starting conditions. See, for example, Figure 7.3, where, for a seed of 13, P = 5583, Q = 3228 and *fitness* = 1593.



GA-41a. Given an initial setting, this shows how the GA converges. The USA's counter-value (P) is dark blue while the Soviet Union's (Q) is red. Fitness is light blue.

Seed 42 Mut_rate 1% Xover_rate 62%, both fixed.
Results at genmax: P is 3486, Q is 2413, and fitness is 2358

Figure 7. 2 GA-2 with a seed of 42



GA-41a. Given an initial setting, this shows how the GA converges. The USA's counter-value (P) is dark blue while the Soviet Union's (Q) is red. Fitness is light blue.

Seed 13 Mut_rate 1% Xover_rate 62%, both fixed.
Results at genmax: P is 5383, Q is 3228, and fitness is 1593

Figure 7.3 GA-2 with a seed of 13

So far, the GA-approach has not proved very useful. We next consider varying the delivery probabilities, μ and v, in order to cater for '*what if* ...' scenarios.

7.11 Varying the Probabilities of Delivery

Appendix 7B outlines how the delivery probabilities, μ and v, of Table 7.1 were derived. Each μ and v is a product of associated probabilities. *Vary_μ* and *vary_v* are *multiplying* *factors* applied to μ and v, respectively, in Tables 7.6a - e, using as base GA-2, a common seed of 42, a random limit of 6845, and values of P and Q obtained from Figure 7.2. These tables present various scenarios.

Table 7. 6a As v is reduced, the Soviet Union can do less damage to America, so P decreases and Q remains unchanged

Vary_μ	*Vary_v*	*P*	*Q*	*Fitness*
1	1	3486	2413	2358
1	0.99	3411	2413	2329
1	0.9	2822	2413	2305

Table 7.6b As μ is reduced, America can do less damage to the Soviet Union, so Q decreases and P increases

Vary_μ	*Vary_v*	*P*	*Q*	*Fitness*
1	1	3486	2413	2358
0.99	1	4227	2289	2259
0.9	1	4227	2071	2124

Table 7.6c As v *i*s increased, the Soviet Union has a bigger chance of damaging America, so P has to rise sharply, but Q decreases slightly

Vary_μ	*Vary_v*	*P*	*Q*	*Fitness*
1	1	3486	2413	2358
1	1.01	4317	2326	2019
1	1.1	5238	2276	1732

Table 7.6d As μ is increased, America will cause more destruction in the Soviet Union, so Q rises (but nothing like as sharply as P in Table 7.6c) and P increases

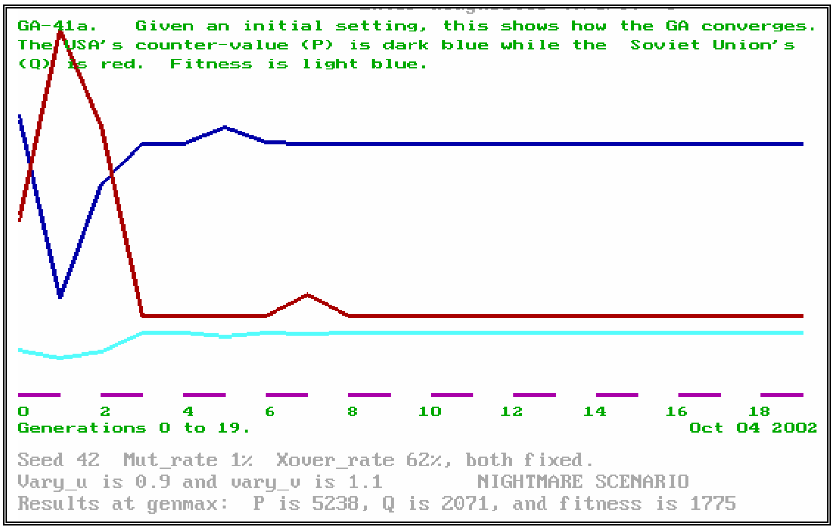
Vary_μ	*Vary_v*	*P*	*Q*	*Fitness*
1	1	3486	2413	2358
1.01	1	3498	2647	2284
1.1	1	4259	2739	1974

Table 7.6e Various other likely scenarios

Vary_μ	*Vary_v*	*P*	*Q*	*Fitness*
1	1	3486	2413	2358
1.1	1.1	5274	2739	1682
0.9	0.9	3720	1996	2368
1.1	0.9	2822	3033	2413
0.9	1.1	5238	2071	1775

The bottom line of Table 7.6e would be a *nightmare scenario* for the Americans. Their own delivery probabilities have been reduced by 10% while those of their opponents have been increased by a similar amount. For SALT to remain stable, i.e. for nuclear deterrence to be maintained, it would seem that America would have to find 5238 missiles (a 50% increase over their 'standard' 3486) while the Soviet Union is already holding sufficient. (Fortunately for the Americans, a Pareto-sort (Table 7.9) later shows that this is not the catastrophe that it now seems.) Figure 7.4 shows a graph of nightmare conditions.

It is worth reflecting that the only thing *known* by each side is the physical holding of its own weapons. Everything else is guesswork.



GA-41a. Given an initial setting, this shows how the GA converges. The USA's counter-value (P) is dark blue while the Soviet Union's (Q) is red. Fitness is light blue.

Seed 42 Mut_rate 1% Xover_rate 62%, both fixed.
Vary_u is 0.9 and vary_v is 1.1 NIGHTMARE SCENARIO
Results at genmax: P is 5238, Q is 2071, and fitness is 1775

Figure 7.4 GA-2 with a seed of 42. Nightmare scenario

7.12 Validating and Improving the Results

So far, we have developed two algorithms to model nuclear deterrence, GA-19, and its greatly improved successor, GA-2. Nevertheless, every seed still gives a different result, i.e. the system is not at all robust. This could be due to the possibility that the data is not consistent (in which case no search algorithm will perform satisfactorily). We need, therefore, to check for continuity in the data. Assuming that the data is consistent, one way to get over the problem of scattered answers is to apply a Pareto-sort (see Section 7.12.2).

7.12.1 Continuity

It would be nice to be assured that there is some continuity in the figures of Table 7.1, i.e. that a small change to M_1, M_2, M_3, m_1, m_2 and m_3 produces approximately the same change in the results of the evaluation equations. If this does not happen, then " … no search strategy will do particularly well, and the Richardson approach could well be flawed" [5].

Accordingly, the above parameters were both reduced and increased by various percentages, and applied to *Equation Set* $P4$ and $Q4$ of GA-2, see Section 7.8.1, up to ± 25%.

The resulting values of P and Q are given in Table 7.7. This indicates that continuity exists in the data of Table 7.1.

---

[5] Comment by an anonymous referee for *GECCO* 2003

Table 7.7 Continuity

% change in $M_1 \ldots m_3$	*P*	% change in *P*	*Q*	% change in *Q*
-25	2387	-26.1	1909	-25.8
-15	2724	-15.6	2174	-15.5
-10	2892	-10.4	2308	-10.3
-5	3061	-5.2	2441	-5.1
0(baseline)	**3229**	**none**	**2573**	**none**
5	3398	+5.2	2706	+5.2
10	3566	+10.4	2838	+10.3
15	3735	+15.6	2969	+15.4
25	4072	+26.1	3236	+25.8

7.12.2 A Pareto-Sort

We next employ a Pareto-sort [Pareto, 1896] in order to obtain the best results from GA-18 over the seeds under consideration, in addition to varying the probabilities of delivery. To this end, GA-18 invokes GA-2 repeatedly, altering the start-point ('seed') at each invocation. Results (a tabulation of seed, *P, Q*, and fitness) were only recorded if they exceeded some pre-set fitness (normally around 2200). Once completed, the table underwent a Pareto-sort over *P* and *Q*. See, for example, Table 7.8, a typical instance.

Table 7.8 Output of GA-18. Normal. Seed 0 through 79. Fitness > 2222

seed	*P*	*Q*	*fitness*	*dominates*	*dominated*
5	3002	2515	2553	54, 59, 60, 66	X
23	3418	2129	2483	42, 54, 60, 66	X
29	3077	2310	2599	42, 54, 59, 60	X
35	2728	2086	2889	5, 23, 29, 42, 54, 59, 60, 66	
42	3486	2413	2358	54, 60, 66	X
54	3486	2658	2281		X
59	3077	2683	2449	42	X
60	3486	2607	2297	54, 66	X
61	3677	1030	2618		
66	3512	2607	2286		X
76	2727	2129	2831	23, 29, 59, 60, 66	

Of eighty invocations of GA-2, the above eleven resulted in fitnesses > 2222. Of these, eight were dominated and can be discarded — they have been shaded. Of the remaining three, seed 35 is clearly the best, since:

- It has a high fitness, so the total number of nuclear missiles has been minimised.
- It has a reasonable balance between *P* and *Q*. Using domain knowledge, it is very unlikely that *Q* will be less than 0.6 *P*. This fact is used hereafter.

Let us now re-visit Table 7.8, but this time the delivery probabilities are those of the American 'nightmare scenario', see Section 7.11 above. Table 7.9 gives some of the results; of the eighteen invocations of GA-2 whose fitness was > 1774, only three were not domin-ated by others. The inclusion of two additional entries (seeds 23 and 48) is to show

how the entry at seed 48 (5238 / 2071, see the last line of Table 7.6e) is dominated by that of seed 23 which in turn is dominated by the entry of seed 41. (The domination of P = 5328 (see Table 7.6e) by P = 3421 must have been a great relief to the Americans.)

Table 7.9 Output of GA-18. Nightmare. Seed 0 through 79. Fitness > 1774. *Vary_μ* 0.9; *Vary_v* 1.1

seed	*P*	*Q*	*fitness*	*dominates*	*dominated*
19	3038	3001	2341		
23	3421	1823	2579	48 (and others)	X
41	3421	1816	2581	23 (and others)	
48	5238	2071	1775		X
61	4532	1621	2077		

The result giving the highest fitness is P = 3421 and Q = 1816. The Americans, therefore, were wise in 1972 to settle for a figure in excess of the 2728 (Table 7.8) missiles they thought they needed at the time; at least they had in a sense a contingency to cater for worst-case scenarios, see Section 7.15.

Running ES-1 (Section 7.7) with a seed of 42 under the 'nightmare' scenario gave P = 4156, Q = 1995, and a fitness of 3145, figures higher than those of Table 7.9.

In our context, whoever attacks with surprise can expect a hundred more of their missiles to reach their target (instead of being countered by defending ABMs, see Section 7.3). This alters the picture materially; compare Figure 7.2 with Figure 7.4.

Figure 7.5 was obtained by modifying *Equation Set P3* and *Q*3 in GA-18 (in effect Eq(7.10) and Eq(7.11)) to remove the *countering* effect of the 100 (American) ABMs. After a Pareto-sort, the output (on assuming a fitness > 1774) consisted of three lines (Table 7.10).

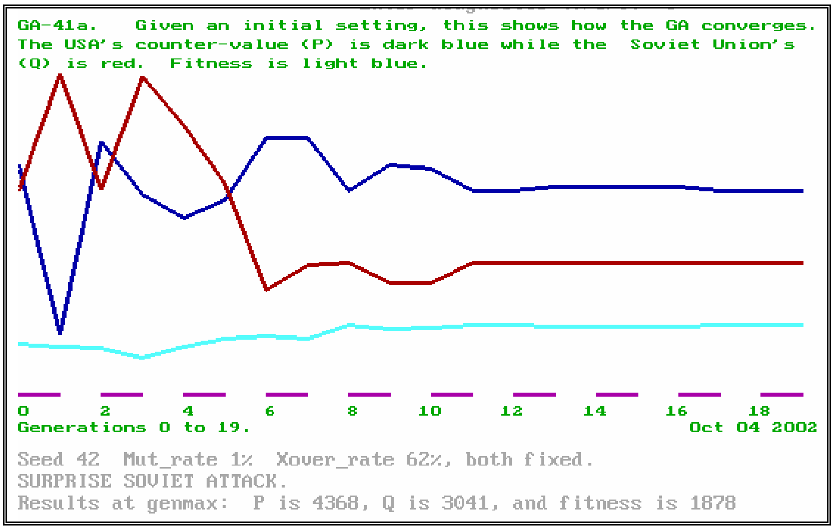
Table 7.10 Output of GA-18. A surprise Soviet attack. Seed 0 through 79. Fitness > 1774

seed	*P*	*Q*	*fitness*	*dominates*	*dominated*
47	3124	2129	2645		
56	4251	1834	2159		
65	2866	2180	2777		

Note. The output of seed 42 (illustrated in Figure 7.5) does not feature in the above table, having been dominated somewhere in the Pareto-sort process

Taking the line of Table 7.10 with the best fitness, the United States needs 2866 weapons to withstand a surprise Soviet attack successfully. This is the second reason (the 3421 weapons needed by the nightmare scenario (Table 7.9) being the first) for the United States to keep their nuclear armoury above the 2728 weapons needed under the 'normal' conditions of Table 7.8.

Successive American Presidents have declared that the US would never initiate a first strike, so no attempt has been made in the thesis to examine the opposite situation, i.e. the United States mounting a surprise attack on the Soviet Union.



```
GA-41a.   Given an initial setting, this shows how the GA converges.
The USA's counter-value (P) is dark blue while the  Soviet Union's
(Q) is red.  Fitness is light blue.
```

```
Seed 42  Mut_rate 1%  Xover_rate 62%, both fixed.
SURPRISE SOVIET ATTACK.
Results at genmax:  P is 4368, Q is 3041, and fitness is 1878
```

Figure 7.5 GA-2. A surprise attack by the Soviet Union

It is worth noting that there are a significant number of Pareto dominations in Tables 7.8, 7.9 and 7.10. Shaw *et al.* [1999] say (not in context of any specific application) that, where this occurs, it could indicate that the problem is multi-objective. However, our fitness function was chosen to minimise the overall number of nuclear weapons, so our problem has a single objective. Had we set, for example, a fitness function for the United States different from that for the Soviet Union, then a multi-objective picture could emerge.

7.13 How to Improve the Performance of GAs for this Application Domain

The results obtained in previous sections indicate that GA-2 is not the best algorithm for this application. The fact that nearly every start-point (every seed) gives a different answer indicates that the GA is not robust. In order to improve the situation, three ideas come to mind; to inject a few redundant genes, to introduce a modest elitism, and to *discretise* the evaluation equations.

7.13.1 Redundancy in Chromosomes

Several authors, including ourselves, have found that a modest redundancy in chromosomes improves the performance and robustness of GAs. Why this should be so is the subject of Chapter 8, but in broad terms it is a manifestation of the *exploration versus exploitation* dilemma [Holland, 1992]. In order to improve GA-2, another algorithm, GA-22, was developed with a 33% redundancy, i.e. there were nine genes instead of six. The structure of its matrices is shown in Table 7.11; the heavily shaded cells of the second row are introns.

Table 7. 11 Configuration of the columns of the matrices for GA-22

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
gen	*srl*	M_1		m_1	M_2		m_2	M_3		m_3	*P*	*Q*	*fit-ness*	*norm-fit*	*running total*

Note. As before, *gen* is generation, *srl* is serial, and *normfit* is a linearly normalised fitness

All the genes were filled from the same (constrained) random number generator. The introns at columns 3, 6 and 9 were initially filled with a constant[6] and not constrained in any other way; these constants did not participate in any GA operation. We observe that we are now handling 9-gene, 36-element (9 × 4) chromosomes.

7.13.2 Elitism

Up to now, every new generation of a nuclear deterrence GA replaced the old one. Elitism allows the best chromosome(s) of one generation to be carried forward into the next – provided, of course, that its/their fitness is better than that of the worst chromosome(s) of the new generation (which it replaces). Thus, one can claim the best of both worlds, i.e. pitting experience against youth. The mechanics of this are simple. A separate record is kept of the details of the best chromosome of (the bubble-sorted) Matrix 1. When Matrix 1 comes around again, fitnesses are compared and, if appropriate, the best chromosome of the last generation ousts the worst chromosome of the new. The best chromosome also has to find its way into the *unsorted list*, see the end of Section 4.3. In conjunction with redundancy, elitism appears to speed up convergence rapidly; to see this, compare, for example, Figure 7.8 with Figure 7.2.

7.13.3 Improved Results

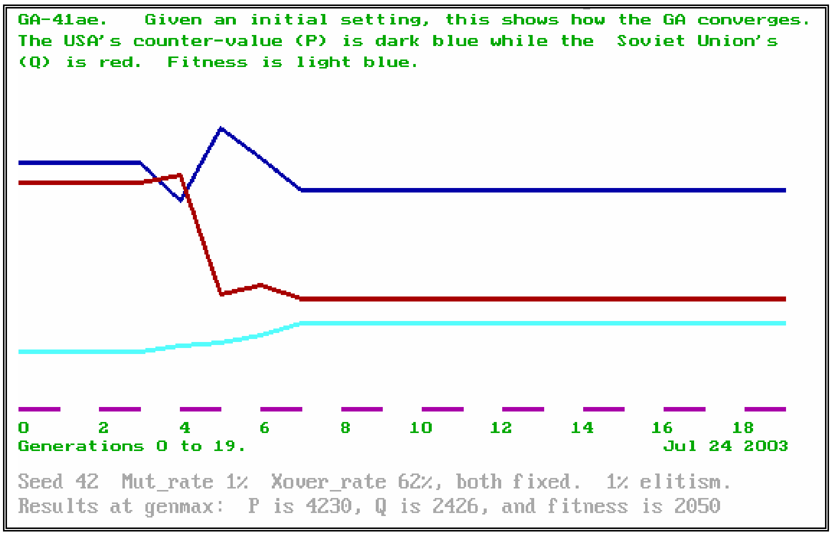
Table 7.12 compares just one result, that at (an arbitrarily chosen) seed 42, with the known results (P = 3229, Q = 2573, see Section 7.9). Except for the last line, the results are reasonably close, and the GA-2 and GA-22 figures are the closest to the known answers.

Table 7.12 Comparing the different GAs at a seed of 42

algorithm	*seed*	*P*	*Q*	*fitness*	*Figures*
'known'		3229	2573		
GA-2 (straight)	42	3486	2413	2358	7.2
GA-20 (elitist)	42	4230	2426	2050	7.6
GA-22 (redundant)	42	3448	2612	2311	7.7
GA-24 (both)	42	4227	2426	2051	7.8

---

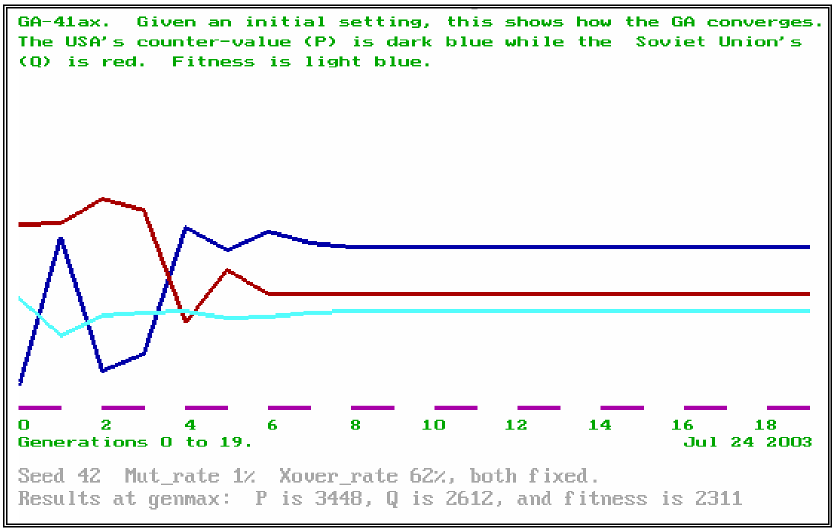
[6] This was to aid comparison between GA-2 and GA-22. Had these introns been filled from the RNG, the sequences of random numbers filling the 'live' genes would have been upset, and comparisons would not have been possible



GA-41ae. Given an initial setting, this shows how the GA converges. The USA's counter-value (P) is dark blue while the Soviet Union's (Q) is red. Fitness is light blue.

Seed 42 Mut_rate 1% Xover_rate 62%, both fixed. 1% elitism.
Results at genmax: P is 4230, Q is 2426, and fitness is 2050

Figure 7.6 GA-20 with 1% elitism



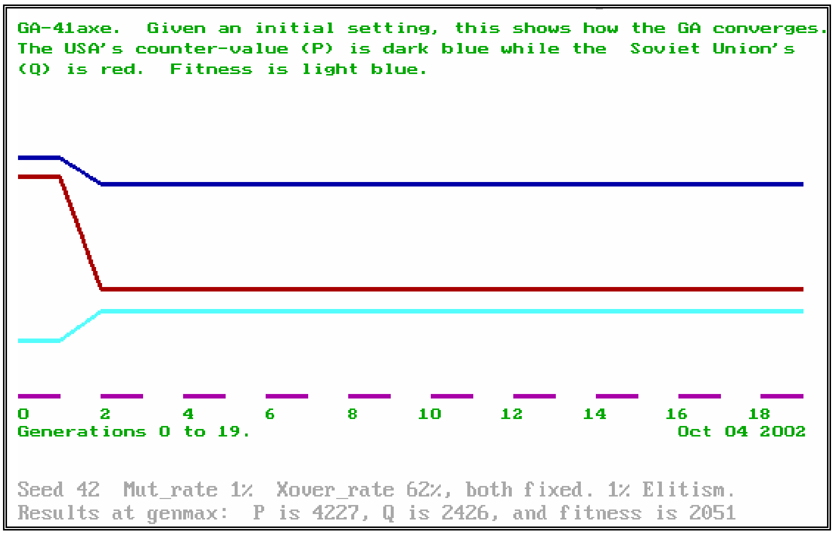
GA-41ax. Given an initial setting, this shows how the GA converges. The USA's counter-value (P) is dark blue while the Soviet Union's (Q) is red. Fitness is light blue.

Seed 42 Mut_rate 1% Xover_rate 62%, both fixed.
Results at genmax: P is 3448, Q is 2612, and fitness is 2311

Figure 7.7 GA-22 with 33% redundancy

It is stressed that the experimental trials pictured above and below were all conducted with one seed (42) so that the various outputs could be visually compared. To obtain workable numerical results, GA-18 (with Pareto-sort) was appropriately modified to yield GA-21, GA-23 and GA-25. Each of these GAs was run eighty times, with seeds 0, 1...79, under *normal* delivery probabilities, and the following results were obtained:

GA-18 (straight)	11 had fitness > 2222 of which 3 were undominated with 2 equal
GA-21 (elitist)	11 had fitness > 2222 of which 2 were undominated and equal
GA-23 (redundant)	8 had fitness > 2222 of which 4 were undominated and equal
GA-25 (elitist, redundant)	12 had fitness > 2222 of which 5 were undominated and equal.



```
GA-41axe.  Given an initial setting, this shows how the GA converges.
The USA's counter-value (P) is dark blue while the  Soviet Union's
(Q) is red.  Fitness is light blue.
```

```
Seed 42  Mut_rate 1%  Xover_rate 62%, both fixed. 1% Elitism.
Results at genmax:  P is 4227, Q is 2426, and fitness is 2051
```

Figure 7.8 GA-24 with 1% elitism and 33% redundancy

Each of the above GAs was run eighty times, with seeds 0, 1...79 as before, under *nightmare* conditions (*Vary_*$\mu = 0.9$, *vary_v* = 1.1), and the following results were obtained:

GA-18 (straight)	6 had fitness > 1774 of which 2 were undominated
GA-21 (elitist)	7 had fitness > 1774 of which 2 were undominated
GA-23 (redundant)	7 had fitness > 1774 of which 2 were undominated
GA-25 (elitist, redundant)	9 had fitness > 1774 of which 1 was undominated.

The best results of the above, after applying domain knowledge constraints, are summarised in Tables 7.13 and 7.14.

Table 7.13 The best results from four GAs. *Normal* delivery probabilities

Source	*Type of GA*	*P*	*Q*	*Fitness*
GA-18	straight	2728	2086	2889
GA-21	elitist	2725	2086	2891
GA-23	redundant	2725	2090	2891
GA-25	elitist and redundant	2725	2090	2891

Table 7.14 The best results from four GAs. *Nightmare. Vary_*$\mu = 0.9$, *Vary_v* =1.1

Source	*Type of GA*	*P*	*Q*	*Fitness*
GA-18	straight	3038	3001	2341
GA-21	elitist	3420	3371	2082
GA-23	redundant	3047	2618	2489
GA-25	elitist and redundant	3038	2614	2495

To conclude:

- It is known that in the 1970s the Americans took the nightmare scenario quite seriously, which probably accounts for their enthusiasm for the 'known' solution of $P = 3229$ (see

Section 7.9). Missile holdings as low as 2728 (Table 7.8) might not, arguably, have made them vulnerable, but it certainly made them uneasy!

- There is very little difference in the results emanating from Tables 7.13 and 7.14. The reason for preferring one GA over another will therefore depend on other factors, such as enhanced stability or robustness.

We will now look at another option. The evaluation equations used in Chapters 5 and 6 were already discretised. So far, those used in this chapter are not.

7.14 Discretising the Evaluation Equations: Delaying Convergence

In order to put P and Q inside the chromosome, it is necessary to discretise the evaluation equations, i.e. to obtain

$$P_{t+1} = P_t + \dots \quad \text{and} \quad Q_{t+1} = Q_t + \dots \,. \tag{7.14}$$

For discretisation we require to know

$$\frac{dP}{dt} = \frac{dP}{dM}\frac{dM}{dt} \,. \qquad .$$

Eq(7.9) yields dP/dM. Moreover, from our experimental results, P and Q appear to take on constant values as they near the Region of Stability. We assume, therefore, that as the Region of Stability is approached then dM/dt and dm/dt take on constant values, $O(1)$, which we denote by γ and $\hat{\gamma}$, respectively.

In the process of re-calculating β for the discretised case we will, in addition to γ and $\hat{\gamma}$, take the opportunity to introduce six rate-factors, k_i. As before, these rate-factors have the dimension of $(\text{time})^{-1}$. Thus, for the United States

$$\beta = \left(\left(1 - \frac{v_1}{w_1^{1/3}}\right)^{\frac{k_1(m_1-50)w_1}{M}} \left(1 - \frac{v_2}{w_2^{1/3}}\right)^{\frac{k_2(m_2-50)w_2}{M}} \left(1 - \frac{v_3}{w_3^{1/3}}\right)^{\frac{k_3 m_3 w_3}{M}} \right) \gamma \,, \tag{7.15}$$

where $M = M_1 + M_2 + M_3 = 2230$ from Table 7.1.

Similarly, for the Soviet Union

$$\hat{\beta} = \left(\left(1 - \frac{\mu_1}{W_1^{1/3}}\right)^{\frac{k_4(M_1-50)W_1}{m}} \left(1 - \frac{\mu_2}{W_2^{1/3}}\right)^{\frac{k_5(M_2-50)W_2}{m}} \left(1 - \frac{\mu_3}{W_3^{1/3}}\right)^{\frac{k_6 M_3 W_3}{m}} \right) \hat{\gamma} \,, \tag{7.16}$$

where $m = m_1 + m_2 + m_3 = 2498$ from Table 7.1. It should not be assumed that $\beta \equiv \hat{\beta}$, nor that $\gamma \equiv \hat{\gamma}$.

It is easily seen from the above that

$$P_{i+1} = P_i + \beta\left(1+\left|\ \ln\beta\ \right|\right)\gamma \qquad .$$

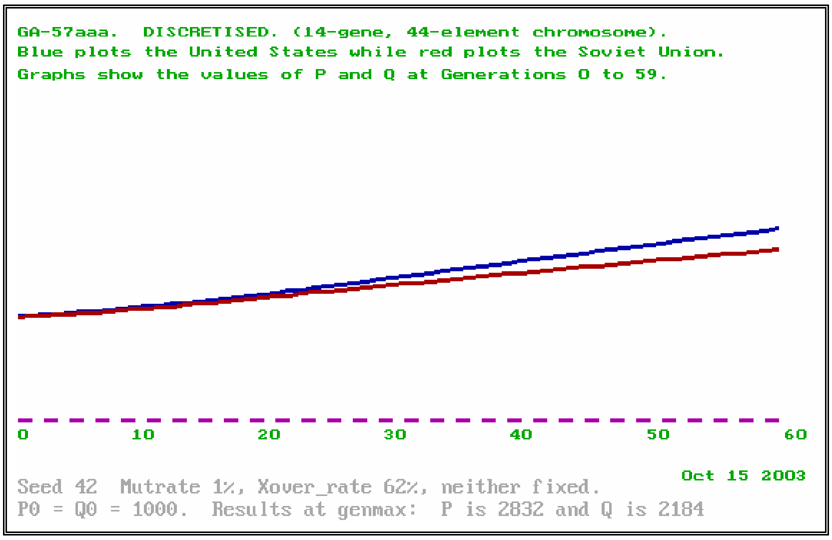
$$Q_{i+1} = Q_i + \hat{\beta}\left(1+\left|\ \ln\hat{\beta}\ \right|\right)\hat{\gamma} \ . \qquad (7.17)$$

Eqs(7.17) conform to Eqs(7.14) and the configuration of the chromosome used in GA-3 is shown in Table 7.15. P and Q are now part of the chromosome, and must be initialised with some starter-values P_0 and Q_0.

Table 7.15 GA-3 (15 genes, 48-elements). The columns of Matrices 1, 2 and 3. The evaluation equations have been discretised, so that P_i and Q_i (columns 15 and 16) are now an integral part of the (shaded) chromosome. The bottom row shows the elements in each gene. Gene 8 is deliberately void

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
gen	*srl*	k_1	k_2	k_3	k_4	k_5	k_6		M_1	m_1	M_2	m_2	M_3	m_3	P_t	Q_t	P_{t+1}	Q_{t+1}
		2	2	2	2	2	2	4	4	4	4	4	4	4	4	4		

GA-3 was developed using Eqs(7.17) with empirical values for γ and $\hat{\gamma}$. Its output has been scaled to produce Figure 7.9 (this time, over 60 generations). Figure 7.9 shows that the discretised process leads to a much smoother output. Results are very robust (i.e. the seed makes little difference to the results). The numerical results, P = 2832 and Q = 2184, are only 4% away from the 'best' results given in Table 7.13 (2728 and 2086, respectively).



GA-57aaa. DISCRETISED. (14-gene, 44-element chromosome).
Blue plots the United States while red plots the Soviet Union.
Graphs show the values of P and Q at Generations 0 to 59.

Seed 42 Mutrate 1%, Xover_rate 62%, neither fixed.
P0 = Q0 = 1000. Results at genmax: P is 2832 and Q is 2184

Figure 7.9 GA-3. This is for the same seed (42) as Figure 7.2. γ and $\hat{\gamma}$ have empirical values 0.99984 and 0.99953, respectively, and appear to be both sensitive and heavily coupled

The significant difference between Eq(7.10) and Eq(7.11) vis-à-vis Eqs(7.17) is that via the latter P_i and Q_i are *included in the genes of the chromosome* (and so, in a sense, help to operate on themselves). This technique was suggested by Forrest and Mayer-Kress [1991] who indicated that it might be beneficial to include the initial values P_0 and Q_0 (and their successors) as genes in the chromosome, putting them under the control of the GA and making them subject to adaptation. In contrast, when Eq(7.10) and Eq(7.11) are used as evaluation equations, *P and Q cannot be part of the chromosome* and hence successive results could only influence each other by the genes they had left behind. It has been demonstrated experimentally that this latter approach often led to very rapid (and homogeneous) convergence, sometimes to the point of cloning. The problem with rapid convergence is that different seeds will give very different results, although it was soon appreciated that those with the highest fitness will converge to very similar solutions. The advantage of retarding convergence is that robustness is greatly improved, i.e. the results depend only marginally on the starting conditions.

(Figure 8.19, a sequel to Figure 7.9, shows what happens when GA-3 is modified so that all crossover takes place in one intron, thereby minimising disruption to the building blocks).

7.15 Summary of Results

In order to help the reader form a global view from the plethora of techniques employed, we present a synopsis of the results:

- In Table 7.1 we gave actual holdings as $M = 2230$ (US) and $m = 2498$ (Soviet Union).
- In 1972 we calculated M to be 2354 (Section 7.6).
- By calculating *Equation Set P*4 and *Q*4 directly, we obtained $P = 3229$ and $Q = 2573$ (Section 7.9).
- With a seed of 42, GA-2 yielded $P = 3486$ and $Q = 2413$ (Figure 7.2).
- The initial nightmare scenario gave $P = 5238$ and $Q = 2071$ (Table 7.6e).
- After a Pareto-sort, the best GA-18 yielded $P = 2728$ and $Q = 2086$ (Table 7.8).
- After a Pareto-sort, the nightmare GA-18 gave $P = 3421$ and $Q = 1816$ (Table 7.9).
- Under surprise attack conditions, GA-18 gave $P = 2866$ and $Q = 2180$ (Table 7.10).
- Under normal conditions, the optimised result was $P = 2728$ and $Q = 2086$ (Table 7.13)

Note. It is understood that, shortly after 1972, the Americans increased their actual holdings (viz. Table 7.1) from 2230 to about 3000 nuclear missiles, presumably to give themselves a contingency against abnormal conditions. (It is ironic that as a direct result of SALT, effectively the world's first major non-proliferation treaty, the number of nuclear weapons actually *increased*!).

We next give reasons why we used GAs to determine nuclear deterrence.

- If we are to pose '*what if* ...' questions, we need algorithms that are both adaptable and flexible.
- We are looking for fast convergence with some self-correcting facility and reflexive action, namely that the number of missiles required by the Americans needs to be expressed solely in terms of the Soviet capability to destroy them, and *vice versa.*
- It was evident both from the literature of the early 1970s and our own experience that there are a lot of 'hills' in the nuclear deterrence problem and that therefore it would be prudent to avoid hill-climbing methods. The need for such avoidance is less true today with commercial software packages like *Mathematica*™ or *MATLAB*™.

In the next section, we demonstrate that nuclear deterrence is still a non-trivial problem in purely numerical terms and that, with one exception, the resulting algorithms are extremely inflexible.

7.16 Six Alternative Numerical Methods for Nuclear Deterrence

We look again at Section 7.5 but only from the American viewpoint and take a slightly different approach, namely destruction rather than survival. We simplify the ensuing calculations by taking the total number of Soviet weapons, m, and a weighted average delivery probability v.

The total number of Soviet *missiles* is m of which we have assumed (Section 7.5) that the Americans can destroy the first 100 with their ABMs. The fraction of Soviet *warheads* assigned to destroy each American missile *on the ground* is therefore $(m-100)w/M$, where M is the total number of American missiles before any attack.

Let v be the probability that an American missile is destroyed on the ground by a Soviet warhead. Although it is shown in Appendix 7A that the probability of destruction of an American missile by one of w Soviet warheads is $v/w^{1/3}$, we will assume here the actual case that in 1972 the Soviets had no multi-warhead capability, so $w = 1$. It follows, therefore, that the probability that any one US missile is destroyed on the ground is $v^{(m-100)/M}$. Thus the expected number of American missiles *destroyed* is $M\,v^{(m-100)/M}$.

Let P be the number of American missiles expected to *survive* a Soviet first strike (the *counter-value*). Using Eq(7.6), we have

$$P = M\left((1-v_1)^{m_1-50}(1-v_2)^{m_2-50}(1-v_3)^{m_3}\right)^{\frac{1}{M}} \quad = \beta M\ , \text{ where } 0 < \beta < 1\ .$$

From Table 7.1, $m = m_1 + m_2 + m_3 = 740 + 1618 + 140 = 2498$; also $v_1 = 0.76$, $v_2 = 0.75$, and $v_3 = 0.4$, giving a weighted average $v = 0.73$. The above equation then simplifies to yield

$$P = (1 - v)^{(m-100)/M} M \ = \beta M, \qquad .$$

whence .

$$M = (m - 100)\ln(1 - v) / \ln \beta \tag{7.18}$$

$$\ln \beta = (m - 100)\ln(1 - v) / M\ . \tag{7.19}$$

We are now faced with one equation in two unknowns (since we can only guess at β). Five ways of solving this problem are described below.

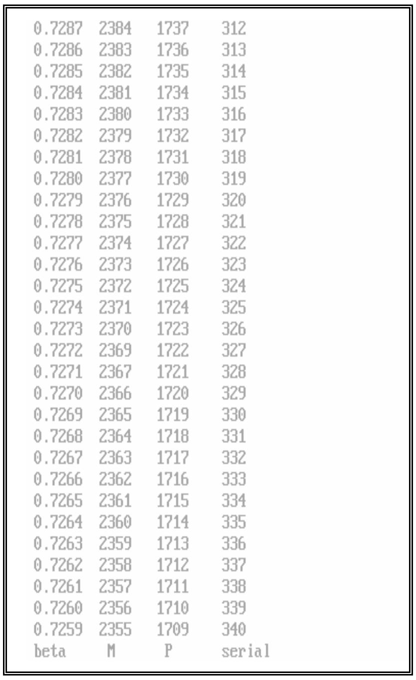
7.16.1 The Brute Force Method

One way of solving one equation with two unknowns (especially with the high-speed facilities of MATLAB) is to try all feasible values of β and *discover when M gives a reasonable answer,* 'reasonable' being when $M_{calculated} \approx M_{real}$, the latter being about 2358.

We use MATLAB to compute Eq(7.18), viz

$$M = (2498 - 100)\ln(1 - 0.73) / \ln \beta. \qquad .$$

We took values of β from 0.76 to 0.72 in steps of 0.0001 and output β, M and $P = \beta M$ in tabular format. These are separate trials, since the result of one trial is not used as data for the next. Results show that *after* 337 *trials* $M_{calculated}$ is 2358 when β is 0.726 and this ties in with the results from the Gauss-Seidel and the Jacobi methods detailed below



```
0.7287  2384   1737    312
0.7286  2383   1736    313
0.7285  2382   1735    314
0.7284  2381   1734    315
0.7283  2380   1733    316
0.7282  2379   1732    317
0.7281  2378   1731    318
0.7280  2377   1730    319
0.7279  2376   1729    320
0.7278  2375   1728    321
0.7277  2374   1727    322
0.7276  2373   1726    323
0.7275  2372   1725    324
0.7274  2371   1724    325
0.7273  2370   1723    326
0.7272  2369   1722    327
0.7271  2367   1721    328
0.7270  2366   1720    329
0.7269  2365   1719    330
0.7268  2364   1718    331
0.7267  2363   1717    332
0.7266  2362   1716    333
0.7265  2361   1715    334
0.7264  2360   1714    335
0.7263  2359   1713    336
0.7262  2358   1712    337
0.7261  2357   1711    338
0.7260  2356   1710    339
0.7259  2355   1709    340
beta     M      P      serial
```

Figure 7.10 The Brute Force Method. Note the entry at serial 337

The corresponding value for P is 1712, quite a low figure, see Figure 7.10. It became evident during this computation that P, the counter-value, is of little real significance – it is the number of actual missiles, M, which counts. However, the figures confirm that the Americans could expect to lose just over a quarter of their missiles as casualties of a Soviet first strike.

MATLAB™ was preferred for this application over *Mathematica*™ [Jacob, 2001] because it is designed to handle data in matrix form.

7.16.2 Leap-frogging Iteration **1** (*a Jacobi Variant*)

Consider the general equation

$$g\,(x,y) \;\equiv y - g_1\,(x,y) \;\equiv x - g_2\,(x,y) = 0. \qquad .$$

We next iterate by leap-frogging between x and y, viz.

$$y_{n+1} \;=\; g_1\,(x_n\,,\, y_n), \qquad .$$

$$x_{n+1} \;=\; g_2\,(x_n\,,\, y_n), \qquad .$$

$$y_{n+2} \;=\; g_1\,(x_{n+1}\,,\, y_{n+1}), \qquad .$$

$$x_{n+2} \;=\; g_2\,(x_{n+1}\,,\, y_{n+1})\,, \text{ etc.} \qquad .$$

where $n = 0, 1, 2, \ldots$ [Buckingham, 1962].

Let f be (2498 – 100) ln(1 - 0.73). Then, if we substitute β for y and M for x in Eq(7.18) and Eq(7.19), respectively, we obtain

$$\beta_1 \;= \exp(f\,/\,M_0), \;\; M_1 \;=\; f\,/\,\ln(\beta_0) \qquad .$$

$$\beta_2 \;= \exp\,(f\,/\,M_1), \;\; M_2 = f\,/\,\ln(\beta_1) \qquad .$$

$$\beta_3 = \exp(f / M_2),\ M_3 = f / \ln(\beta_2)\text{, etc.}$$

We initialised M_0 with 3229 (the result of ‘knowing’ the results for both sides, see Section 7.9), and β_0 with 0.8. M dropped successively to 3222, 3215 and 3208 but it was not until iterations 299/300 that the curve flattened asymptotically at 2360, with β at 0.735.

7.16.3 Leap-frogging Iteration 2 (*a Gauss-Seidel Variant*)

We now iterate by leap-frogging between x and y in a different fashion, viz.

$$y_{n+1} = g_1(x_n, y_n),$$

$$x_{n+1} = g_2(x_n, y_{n+1}),$$

$$y_{n+2} = g_1(x_{n+1}, y_{n+1}),.$$

$$x_{n+2} = g_2(x_{n+1}, y_{n+2}),\quad \text{etc.}$$

where $n = 0, 1, 2, \ldots$ [Buckingham, 1962]. Then, having set an initial value M_0, if we substitute β for y and M for x in Eq(7.18) and Eq(7.19), respectively, we obtain

$$\beta_1 = \exp(f / M_0),\ M_1 = f / \ln \beta_1$$

$$\beta_2 = \exp(f / M_1),\ M_2 = f / \ln \beta_2$$

$$\beta_3 = \exp(f / M_2),\ M_3 = f / \ln \beta_3\text{, etc.}$$

In this case, because M_0 and M_1 are the same, it was necessary to perturb β very slightly in order to start convergence. Starting, as before, with $M_0 = 3229$, $M_1 = 3222$, $M_{149} = 2361$ and $M_{150} = 2360$. Asymptotic convergence had occurred by the 150th iteration. At this point β is 0.735, meaning that the Americans could expect to lose just over a quarter of their missiles as a result of a Soviet first strike. The figure obtained in 1972 was 2354, see Section 7.6.

7.16.4 Leap-frogging Iteration 3 (*a Second Gauss-Seidel Variant*)

We next iterate by leap-frogging between x and y but in yet another sequence, taking two attempts at each x and y, viz.

$y_{n+1} = g_1(x_n, y_n)$, $y_{n+2} = g_1(x_n, y_{n+1})$, $x_{n+1} = g_2(x_n, y_{n+2})$, $x_{n+2} = g_2(x_{n+1}, y_{n+2})$, $y_{n+3} = g_1(x_{n+2}, y_{n+2})$, $y_{n+4} = g_1(x_{n+2}, y_{n+3})$, $x_{n+3} = g_2(x_{n+2}, y_{n+4})$, $x_{n+4} = g_2(x_{n+3}, y_{n+4})$, etc. where $n = 0, 1, 2 \ldots$ [Buckingham, 1962].

Then, having set an initial value M_0, and using the same machinery as before, we obtain

$$\beta_1 = \exp(f/M_0)\,,\ \beta_2 = \exp(f/M_0 + \xi_M)\,,\quad M_1 = f/\ln(\beta_2),\quad M_2 = f/\ln(\beta_2 - \xi_\beta),\qquad .$$

$$\beta_3 = \exp(f/M_2),\ \beta_4 = \exp(f/M_2 + \xi_M)\,,\quad M_3 = f/\ln(\beta_4),\ M_4 = f/\ln(\beta_4 - \xi_\beta)\,,\qquad .$$

$$\beta_5 = \exp\,(f/M_4),\ \beta_6 = \exp\,(f/M_4 + \xi_M),\quad M_5 = f/\ln(\beta_6)\,,\ M_6 = f/\ln(\beta_6 - \xi_\beta)\,,\quad \text{etc.}$$

It was necessary to perturb M and β very slightly (by $\xi_M = 3.5$, $\xi_\beta = 0.0015$) in order to ensure convergence. (The perturbations ξ_M and ξ_β are of comparable order.) Starting, as before, with $M_0 = 3229$, $M_1 = 3210$ while $M_{112} = 2362$ and $M_{113} = 2360$. Hence asymptotic convergence had occurred; indeed, at 113 iterations it was significantly quicker than the earlier Gauss-Seidel at 150 iterations. At this point β was 0.734.

7.16.5 A Monte Carlo Method

In Section 5.8.1 we used a Monte Carlo method to see if we could predict instabilities in the India/Pakistan situation, and while reasonable numerical results were obtained, the lack of sign changes mitigated against it. The idea of the Monte Carlo method is that an integral may be represented by a sum , namely

$$\int_{-\infty}^{\infty} g(x) f(x) dx \approx \frac{1}{N} \sum_{i=1}^{N} g(x_i), \tag{7.20}$$

so to create a Monte Carlo solution [Kalos and Whitlock, 1986] we draw a series of random variables x_n from $f(x)$, and evaluate $g(x)$ for each x_n. The arithmetic mean of all N values of g is an approximation to the integral, and the variance of this approximation decreases as the number of terms increases.

We next set out to develop a new Monte Carlo algorithm, which we call MC-6.

In the India/Pakistan case, Eqs(5.3) had been discretised so at each generation x_{t+1} and y_{t+1} became x_t and y_t, respectively. Even with no selection, mutation or crossover, this transfer was sufficient to ensure that each chromosome was marginally different from its neighbour and its predecessor, so that the population had the variation required by Eq(7.20) to make a Monte Carlo method viable.

With nuclear deterrence, however, Eqs(7.10) and (7.11) cannot be discretised, so as one generation succeeded another, *nothing changed*, i.e., the diagnostic tables of genes at Generation 39 (cf. Figure 4B.2) looked exactly as they had when they were initiated at Generation 0. To break this impasse, we deliberately introduced mutation into MC-6, not so much to *mutate* the genes as to *perturb* the chromosomes, and thus make each chromosome marginally different from its neighbour and predecessor. In fact, the optimal perturbation rate per locus was found to be 4.5% (which would have been high for a GA mutation rate). The results can be seen in Figure 7.11 from which the performance of MC-6 can be compared with that of GA-2 in Figure 7.2. In terms of numerical results, the comparison is poor (even allowing for the fact that one is seed 42 and the other is seed 35, and that it took forty generations to achieve results from MC-6, as distinct from twenty from GA-2). However, a much more interesting aspect arose.

It will be recalled that results from GA-2 were not at all robust (such that a Pareto-sort became necessary, see Section 7.12.2). In contrast, results from MC-6 were surprisingly consistent as Table 7.16 shows. The solution P = 2706 and Q = 2125 from Figure 7.11 can be compared with the 'best' result in the top (GA-18) row of Table 7.13, which in turn relates to the 'winning' seed, 35, of the Pareto-sort in Table 7.8 (P = 2728, Q = 2086). The implication in this application is that Monte Carlo results have little need of a Pareto-sort.

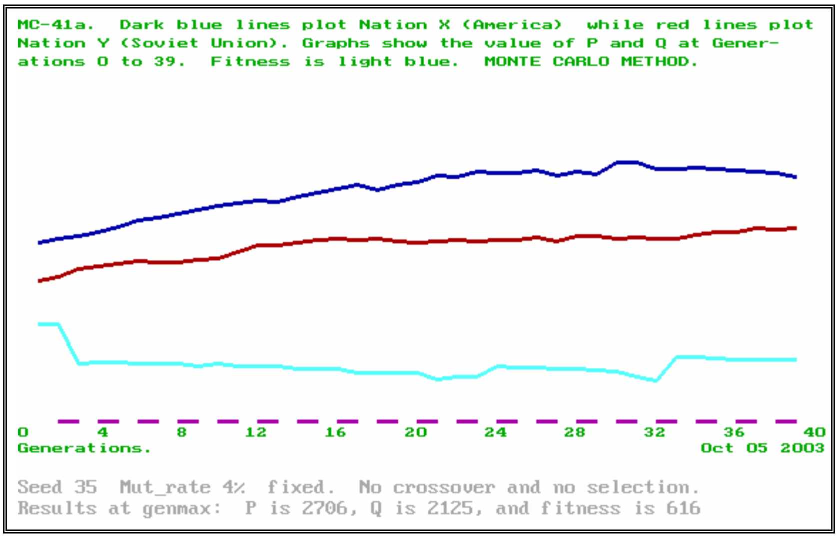


Figure 7.11 MC-6. A Monte Carlo approach. Seed 35. Here P is 2706 and Q is 2125, compared with Table 7.8 where, for GA-2 (cf. GA-18) at the same seed, P was 2728 and Q was 2086

Table 7.16 A comparison between the outputs of MC-6 and GA-2 at different seeds

	MC-6	*MC-6*	*GA-2*	*GA-2*
seed	*40 generations* *P*	*40 generations* *Q*	*20 generations* *P*	*20 generations* *Q*
0	2894	2015	4259	2342
9	2825	2045	3659	5043
16	2875	2074	5204	1817
33	2928	2003	4853	2151
35 (*Tables 7.8 and 7.13*)	2706	2125	2728	2086
42 (*Figure* 7.2)	2891	2091	3486	2413
54	2794	2035	3486	2658
76	2824	2132	2670	5954

% difference between biggest and smallest	7.5%	6.5%	82%	227%

7.16.6 Simulated Annealing

It will be remembered that in Section 7.7 we developed an evolutionary algorithm, ES-1, for the nuclear deterrence problem [Schwefel, 1995]. At first sight this appeared to be an attractive option, for an ES really is 'survival of the fittest', the advantage being that no time is wasted on unfit or infeasible solutions. We soon found, however, that ES-1 suffers from all the disadvantages traditionally associated with hill-climbing methods. Furthermore, it could only produce one answer at a time; there was none of the *implicit parallelism* (Section 2.4) to be found in GAs. To overcome these disadvantages, we abandoned the ES approach in favour of a GA-based approach. We did, however, return to ES-1 to help develop a *simulated annealing* algorithm, SA-5, for the India/Pakistan problem, see Section 5.8.2. While successful in some ways, SA-5 failed to produce the sign changes necessary for our prediction technique, and so was abandoned, see Section 5.9. Simulated annealing is, in essence, an ES with one refinement; Metropolis's criterion, based on Boltzmann probabilities, which gives rejected chromosomes a probabilistic 'second chance' as Route 2 in Table 5.3 shows. This still does not alter the fact that simulated annealing only gives one answer per generation, and there is none of the implicit parallelism to be found in GAs or, in a sense, in Monte Carlo, for both these techniques implicitly process far more information than is apparent in their answers. For this reason, we have not considered it worth applying simulated annealing to the nuclear deterrence problem.

7.17 Comments on the Non-GA Methods

The figures calculated today confirm those obtained with much difficulty in 1972, namely that M is about 2360 and that β is about 0.735. However, even with all of today's facilities for handling numerical methods, including MATHEMATICA™ and MATLAB™, finding a solution by numerical means for the American counter-value, P, is still a non-trivial problem.

We consider the results in a reverse order of effectiveness. The 'Brute Force' method took 337 separate trials to achieve acceptable convergence, despite the fact that the initial values were very near the fixed point; where this is not the case, the number of trials could rise to a thousand or more. The Jacobi leap-frogging method took some 300 iterations to achieve convergence. Likewise, the first Gauss-Seidel method took 150 iterations, while the second

Gauss-Seidel method took only 113 iterations to converge. The Monte Carlo method produced answers which were both accurate and robust, but it took 40 generations to work.

All these methods are computation-intensive. In contrast, the canonical GAs we have been using in this chapter converge typically after only twelve generations. Additionally, GAs are flexible, enabling us to change parameters and work out a host of *'what if'* problem instances. All in all, only Monte Carlo proved as successful as the various GAs.

7.18 Applying GAs to Nuclear Deterrence

In general terms, it seems that a GA *can* efficiently find an acceptable solution to the nuclear deterrence problem in the context of the following observations:

- The data has *continuity*, see Section 7.12.1
- The evaluation equations are reflexive, so that a better chromosome improves both sides rather than being partisan. The number of nuclear weapons needed by the United States depends on the Soviet Union's capability to destroy a sizeable proportion of them, and vice versa – and the means of doing both appears in the same chromosome simultaneously.
- The fitness function *minimises* the fitness of both sides simultaneously.
- A GA must never be *forced*, but it can be *coaxed.* A good way to coax a GA is to constrain the RNG to a likely limit for each parameter[7]. On the other hand, increasing the search space from 10^{24} to 10^{36} by adding three introns made the GA more robust. The increased search space, however, also means that many more of the answers will be duplicated, and that a Pareto-sort will result in more dominations than previously. It thus seems that there may be 'useful' and 'not-so-useful' search spaces. Perhaps at this juncture it may be appropriate to invoke a *'terrain-based GA'* [Scott-Gordon *et al.*, 1999] capable of pinpointing with fair accuracy those areas of any search space where solutions are most likely to be found.

- The GA is designed in such a way that an audit trail is possible. Even with the most complex of outputs it has always been possible, for instance, to see what was happening to each individual chromosome. A standard test with any new development has been to reduce the population of chromosomes to ten, and the generations to two. Hence the behaviour of two generations of individual chromosomes could be examined in detail on

---

[7] For example, if a particular gene is most unlikely to exceed, say, 72, then the call to the RNG can be set to <`random` (72)>

the screen. A useful side-effect of this was that genuine convergence could be distinguished from degeneration.

- The probabilities of delivery must come from a realistic scenario.
- Initially, both mutation and crossover were varied dynamically, but this did not always work well. Stable results were often obtained by using fixed values such as 0.1% and 60%, respectively.
- Several types of crossover were tried, including uniform crossover [Syswerda, 1989]. On balance, single-point crossover was found to be the most effective.
- The GA approach affords us the ease of examining the effects of varying the delivery probabilities, of being subjected to a surprise attack, of including (or ignoring) warheads delivered by piloted aircraft, of investigating various ‘worst-case’ scenarios, and of other such scenarios.
- When dealing with varying initialising seeds, on using domain knowledge, namely

$$P > Q > 0.65P \quad \text{and} \quad fitness > 2000 \qquad .$$

combined with a Pareto-sort, we were able to identify the undominated answers.

7.19 Conclusions

In this chapter we set out to find out if a GA-based approach could be applied to the problems of nuclear deterrence. In 1972 it was thought that America could have withstood an onslaught of 2354 Soviet missiles. Using Richardson’s equations and all the data about numbers of weapons and delivery probabilities available to us in Table 7.1, the counter-values for America and the Soviet Union can be calculated directly by using *Equation Set P*4 and Q4 (see Section 7.8.1) as P = 3229 and Q = 2573. Neither side needs so many missiles *under normal conditions* but, under abnormal conditions, they may need a lot more.

An evolutionary algorithm, ES-1, was developed, but this suffered from all the problems associated with hill-climbing methods. As a result, a suite of inter-related GAs was then developed in succession. These GAs made no assumptions about missile numbers but we allowed them to evolve. The first, GA-19, did not give good results but demonstrated that, depending on the starting seeds, many answers were possible but those with the highest fitness always came up with approximately the same answers. The second, GA-2, incorporated a number of refinements and yielded P = 3486 and Q = 2413. The third, GA-20, injected a 1% elitism. The fourth, GA-22, deliberately introduced 33% redundancy into its chromosomes. The fifth (GA-24) combined elitism and redundancy. The next four GAs, GA-18, GA-21, GA-23 and GA-25, invoked GA-2, GA-20, GA-22 and GA-24 over a range of starting conditions and then applied a Pareto-sort so that dominated results could be eliminated. The results are given in Table 7.13. Finally, another GA, GA-3, used the discretised evaluation equations, i.e. Eqs(7.17), and put the results of those equations into the genes of subsequent generations.

The reason for all these variants was that we were trying to improve, not so much the accuracy of the numerical solutions, but the *robustness* of our GAs, see (7) of Section 9.3.1.

The number of missiles needed to ensure deterrence (P = 2728, Q = 2086, see Table 7.13) has been determined by developing a GA and subjecting its results to a Pareto-sort. It is much easier to use this GA than a conventional numerical method (as we had to in 1972 when our result for P was 2354), as the GA has greater *functionality*. It makes little difference whether the GA is 'straight', elitist, redundant or both elitist and redundant, but these variations do impact the GA's stability and robustness, as do the discretised evaluation equations, namely Eqs(7.17).

Appendix 7A

Nuclear Weapons Effects

Blast

The effects of nuclear weapons depend on the heights at which they explode. Strategic nuclear weapons would normally be detonated at heights guaranteed to maximise aero-dynamic shock or 'blast'[8]. The mathematics of blast are complex, and there is no simple relationship connecting pressure with the *size* of the weapon (the 'yield', normally expressed in kilotons (KT) of conventional explosive) and distance from the target.

It is possible to scale blast effects for different yields. For a given static overpressure for two different weapons, 1 and 2,

$$\frac{Distance_1}{Distance_2} = \left(\frac{Yield_1}{Yield_2}\right)^{1/3} . \qquad (7A.1)$$

There is an established chart of pressure effects at a yield of 1KT, thus allowing Eq(7A.1) to be applied to any other yield, Y, viz.

---

[8] Detonating these big weapons at heights designed to maximise heat would cause widespread destruction to natural resources such as forests and arable farmland (which it turns into volcanic-type rock), while using heights known to maximise nuclear radiation could cause the target to be a 'no-go' area to humans for about a decade. As a weapon of war, it would be normal to maximise blast, in that this does least long-term damage

$$Distance_Y = Distance_{1KT} \times Y^{1/3}. \qquad (7A.2)$$

The damage done by *multiple* warheads can also be scaled. If a single warhead of yield Y is replaced on the same missile by w equal-yield MIRV warheads, then on using Eq(7A.1) we obtain

$$Distance_w = Distance_Y \times \left(\frac{\left(\frac{Y}{w} \right)}{Y} \right)^{1/3} = Distance_Y \times \left(\frac{1}{w} \right)^{1/3}. \qquad (7A.3)$$

If the probability of destruction of a Soviet missile is μ, then the probability of its destruction by one of W MIRV American warheads is $\mu / W^{1/3}$. Similarly, the probability of dest-ruction of an American missile by one of w MIRV Soviet warheads is $v / w^{1/3}$.

The Probabilities of Delivery

Principles

Suppose one side were to decide that it was vital to their defence that a particular target in their opponent's territory be destroyed. Intelligence reports would give the location and size of the target and to what extent it was 'hardened' to withstand a nuclear hit. From this information it would be easy to calculate the size of weapon to be used and the Height-of-Burst and ground zero to optimise damage and minimise fallout. Assume for the present that one missile is capable of doing the job, and that it carries a warhead of suitable yield. What can go wrong?

- The missile may not be serviceable at the precise moment it is required, either because of a mechanical or electrical fault.
- It may already been destroyed by the opponent's first strike.
- The launching may be aborted.
- The missile may be intercepted or 'countered' while in flight.
- There may be serious guidance or navigational errors.
- Nuclear fission or fusion may not take place on target, i.e. the missile may land at the right place, but 'blind'.
- Even if everything functions correctly the accuracy of the delivery system may be incompatible with the warhead's 'radius of damage', i.e. the warhead explodes but not in the right place to damage the target to the extent required.

A table of probabilities is given below. This gives the chances of things **not** going wrong.

Table 7B.1 The component probabilities of delivery

Type	*ICBM*	*SLBM*	*Aircraft delivered*
Mechanical fault	0.9	0.9	0.8
Victim of first strike	0.8	0.95	0.8
Abortive launch	0.95	0.95	0.95
Intercepted or 'countered'	0.9 (Note 1)	0.9 (Note 1)	0.6
Guidance error	0.99	0.99	0.9
Blind	0.99	0.99	0.95
Combined probability	**0.6** (0.75, *Note* 2)	**0.72** (0.76, *Note* 2)	**0.33** (0.41, *Note* 2)

Note 1. Destruction by an ABM is not included in these figures. It is assumed that each ABM is given a 100% chance of destroying its target missile

Note 2. It is unlikely that the United States would be allowed to strike first. Figures in parentheses give typical probabilities thought to have been open to the Soviet Union

It should now be apparent why an ‘overkill’ is always needed. Each side needs an assurance (say, 95%) that a target will be destroyed. Even if an ICBM and an SLBM were tasked to destroy the same target, the probability of destruction would still only be

$$1-(1-0.6)(1-0.72) \; = \; 0.888 \;\; \text{or} \;\; 89\% \qquad .$$

which is still not a sufficient guarantee that the target will be destroyed. It follows that triple- and sometimes even quadruple-tasking is the norm.

7.16.4 Leap-frogging Iteration 3 (*a Second Gauss-Seidel Variant*)

We next iterate by leap-frogging between x and y but in yet another sequence, taking two attempts at each x and y, viz.

$y_{n+1} = g_1(x_n, y_n)$, $y_{n+2} = g_1(x_n, y_{n+1})$, $x_{n+1} = g_2(x_n, y_{n+2})$, $x_{n+2} = g_2(x_{n+1}, y_{n+2})$, $y_{n+3} = g_1(x_{n+2}, y_{n+2})$, $y_{n+4} = g_1(x_{n+2}, y_{n+3})$, $x_{n+3} = g_2(x_{n+2}, y_{n+4})$, $x_{n+4} = g_2(x_{n+3}, y_{n+4})$, etc. where $n = 0, 1, 2 \ldots$ [Buckingham, 1962].

Then, having set an initial value M_0, and using the same machinery as before, we obtain

$$\beta_1 = \exp(f/M_0), \ \beta_2 = \exp(f/M_1), \ M_1 = f/\ln(\beta_0 - \xi), \quad M_2 = f/\ln(\beta_1 - \xi), \qquad .$$

$$\beta_3 = \exp(f/M_2), \quad \beta_4 = \exp(f/M_3), \ M_3 = f/\ln(\beta_2 - \xi), \quad M_4 = f/\ln(\beta_3 - \xi), \qquad .$$

$$\beta_5 = \exp(f/M_4), \ \beta_6 = \exp(f/M_5), \quad M_5 = f/\ln(\beta_4 - \xi,), \ M_6 = f/\ln(\beta_5 - \xi), \quad \text{etc.}$$

It was again necessary to perturb β very slightly (by $\xi = 0.0002$) in order to start convergence. This was a pity, for the desire to avoid perturbation had spawned this trial. Starting, as before, with $M_0 = 3229$, $M_1 = 3214$ while $M_{65} = 2368$ and $M_{66} = 2365$. Hence asymptotic convergence had occurred; indeed, at 66 iterations it was significantly quicker than the earlier Gauss-Seidel at 150 iterations. At this point β was 0.738.

where $n = 0, 1, 2, \ldots$ [Buckingham, 1962]. Then, having set an initial value M_0, if we substitute β for y and M for x in Eq(7.18) and Eq(7.19), respectively, we obtain

$$\beta_1 = \exp\,(f\,/\,M_0),\;\; M_1 = f\,/\ln\beta_1 \qquad ,$$

$$\beta_2 = \exp\,(f\,/\,M_1),\;\; M_2 = f\,/\ln\beta_2 \qquad .$$

$$\beta_3 = \exp\,(f\,/\,M_2),\;\; M_3 = f\,/\ln\beta_3\,,\;\text{etc.} \qquad .$$

In this case, because M_0 and M_1 are the same, it was necessary to perturb β very slightly in order to start convergence. Starting, as before, with $M_0 = 3229$, $M_1 = 3222$, $M_{149} = 2361$ and $M_{150} = 2360$. Asymptotic convergence had occurred by the 150th iteration. At this point β is 0.735, meaning that the Americans could expect to lose just over a quarter of their missiles as a result of a Soviet first strike. The figure obtained in 1972 was 2354, see Section 7.6.

7.16.4 Leap-frogging Iteration 3 (*a Second Gauss-Seidel Variant*)

We next iterate by leap-frogging between x and y but in yet another sequence, taking two attempts at each x and y, viz.

$$y_{n+1} = g_1(x_n, y_n), \quad y_{n+2} = g_1(x_n, y_{n+1}), \quad x_{n+1} = g_2(x_n, y_{n+2}), \quad x_{n+2} = g_2(x_{n+1}, y_{n+2}),$$
$$y_{n+3} = g_1(x_{n+2}, y_{n+2}), \; y_{n+4} = g_1(x_{n+2}, y_{n+3}), \; x_{n+3} = g_2(x_{n+2}, y_{n+4}), \; x_{n+4} = g_2(x_{n+3}, y_{n+4}),$$

etc. where $n = 0, 1, 2 \ldots$ [Buckingham, 1962].

Then, having set an initial value M_0, and using the same machinery as before, we obtain

$$\beta_1 = \exp(f\,/\,M_0)\,,\; \beta_2 = \exp(f\,/\,M_0 + \xi_M)\,,\;\; M_1 = f\,/\ln(\beta_2),\;\; M_2 = f\,/\ln(\beta_2 - \xi_\beta), \qquad .$$

$$\beta_3 = \exp(f\,/\,M_2),\;\; \beta_4 = \exp(f\,/\,M_2 + \xi_M)\,,\;\; M_3 = f\,/\ln(\beta_4)\,,\;\; M_4 = f\,/\ln(\beta_4 - \xi_\beta)\,, \qquad .$$

$$\beta_5 = \exp\,(f\,/\,M_4),\; \beta_6 = \exp\,(f\,/\,M_4 + \xi_M),\;\; M_5 = f\,/\ln\,(\beta_6)\,,\;\; M_6 = f\,/\ln(\beta_6 - \xi_\beta)\,, \quad \text{etc.}$$

It was necessary to perturb M and β very slightly (by $\xi_M = 3$, $\xi_\beta = 0.0025$) in order to ensure convergence. This was a pity, for the desire to avoid perturbation had spawned this trial. Starting, as before, with $M_0 = 3229$, $M_1 = 3210$ while $M_{52} = 2367$ and $M_{53} = 2353$. Hence asymptotic convergence had occurred; indeed, at 53 iterations it was significantly quicker than the earlier Gauss-Seidel at 150 iterations. At this point β was 0.734.

7.16.5 A Monte Carlo Method

In Section 5.8.1 we used a Monte Carlo method to see if we could predict instabilities in the India/Pakistan situation, and while reasonable numerical results were obtained, the lack of sign changes mitigated against it. The idea of the Monte Carlo method is that an integral may be represented by a sum , namely

where $n = 0, 1, 2, ...$ [Buckingham, 1962]. Then, having set an initial value M_0, if we substitute β for y and M for x in Eq(7.18) and Eq(7.19), respectively, we obtain

$$\beta_1 = \exp(f/M_0),\ M_1 = f/\ln\beta_1 \qquad ,$$

$$\beta_2 = \exp(f/M_1),\ M_2 = f/\ln\beta_2 \qquad .$$

$$\beta_3 = \exp(f/M_2),\ M_3 = f/\ln\beta_3 \text{ , etc.} \qquad .$$

In this case, because M_0 and M_1 are the same, it was necessary to perturb β very slightly in order to start convergence. Starting, as before, with $M_0 = 3229$, $M_1 = 3222$, $M_{149} = 2361$ and $M_{150} = 2360$. Asymptotic convergence had occurred by the 150th iteration. At this point β is 0.735, meaning that the Americans could expect to lose just over a quarter of their missiles as a result of a Soviet first strike. The figure obtained in 1972 was 2354, see Section 7.6.

7.16.4 Leap-frogging Iteration 3 (*a Second Gauss-Seidel Variant*) ORIGINAL

We next iterate by leap-frogging between x and y but in yet another sequence, taking two attempts at each x and y, viz.

$y_{n+1} = g_1(x_n, y_n)$, $y_{n+2} = g_1(x_n, y_{n+1})$, $x_{n+1} = g_2(x_n, y_{n+2})$, $x_{n+2} = g_2(x_{n+1}, y_{n+2})$, $y_{n+3} = g_1(x_{n+2}, y_{n+2})$, $y_{n+4} = g_1(x_{n+2}, y_{n+3})$, $x_{n+3} = g_2(x_{n+2}, y_{n+4})$, $x_{n+4} = g_2(x_{n+3}, y_{n+4})$, etc. where $n = 0, 1, 2 ...$ [Buckingham, 1962].

Then, having set an initial value M_0, and using the same machinery as before, we obtain

$$\beta_1 = \exp(f/M_0)\text{ , } M_1 = f/\ln(\beta_0 - \xi)\text{ , } \beta_2 = \exp(f/M_1)\text{ , } M_2 = f/\ln(\beta_1 - \xi)\text{ ,} \qquad .$$

$$\beta_3 = \exp(f/M_2)\text{ , } M_3 = f/\ln(\beta_2 - \xi)\text{ , } \beta_4 = \exp(f/M_3)\text{ , } M_4 = f/\ln(\beta_3 - \xi)\text{ ,} \qquad .$$

$$\beta_5 = \exp(f/M_4)\text{ , } M_5 = f/\ln(\beta_4 - \xi,)\text{ , } \beta_6 = \exp(f/M_5)\text{ , } M_6 = f/\ln(\beta_5 - \xi)\text{ , etc.}$$

It was again necessary to perturb β very slightly (by $\xi = 0.0008$) in order to start convergence. This was a pity, for the desire to avoid perturbation had spawned this trial. Starting, as before, with $M_0 = 3229$, $M_1 = 3214$ while $M_{65} = 2368$ and $M_{66} = 2365$. Hence asymptotic convergence had occurred; indeed, at 66 iterations it was significantly quicker than the earlier Gauss-Seidel at 150 iterations. At this point β was 0.738.

7.16.5 A Monte Carlo Method

In Section 5.8.1 we used a Monte Carlo method to see if we could predict instabilities in the India/Pakistan situation, and while reasonable numerical results were obtained, the lack of sign changes mitigated against it. The idea of the Monte Carlo method is that an integral may be represented by a sum , namely

where $n = 0, 1, 2, \ldots$ [Buckingham, 1962]. Then, having set an initial value M_0, if we substitute β for y and M for x in Eq(7.18) and Eq(7.19), respectively, we obtain

$$\beta_1 = \exp\,(f/M_0),\ \ M_1 = f/\ln\beta_1 \qquad ,$$

$$\beta_2 = \exp\,(f/M_1),\ \ M_2 = f/\ln\beta_2 \qquad .$$

$$\beta_3 = \exp\,(f/M_2),\ \ M_3 = f/\ln\beta_3\ ,\ \text{etc.} \qquad .$$

In this case, because M_0 and M_1 are the same, it was necessary to perturb β very slightly in order to start convergence. Starting, as before, with $M_0 = 3229$, $M_1 = 3222$, $M_{149} = 2361$ and $M_{150} = 2360$. Asymptotic convergence had occurred by the 150th iteration. At this point β is 0.735, meaning that the Americans could expect to lose just over a quarter of their missiles as a result of a Soviet first strike. The figure obtained in 1972 was 2354, see Section 7.6.

7.16.4 Leap-frogging Iteration 3 (*a Second Gauss-Seidel Variant*)

We next iterate by leap-frogging between x and y but in yet another sequence, taking two attempts at each x and y, viz.

$y_{n+1} = g_1(x_n, y_n)$, $y_{n+2} = g_1(x_n, y_{n+1})$, $x_{n+1} = g_2(x_n, y_{n+2})$, $x_{n+2} = g_2(x_{n+1}, y_{n+2})$,
$y_{n+3} = g_1(x_{n+2}, y_{n+2})$, $y_{n+4} = g_1(x_{n+2}, y_{n+3})$, $x_{n+3} = g_2(x_{n+2}, y_{n+4})$, $x_{n+4} = g_2(x_{n+3}, y_{n+4})$,
etc. where $n = 0, 1, 2 \ldots$ [Buckingham, 1962].

Then, having set an initial value $M_{0,}$ and using the same machinery as before, we obtain

$$\beta_1 = \exp\,(f/M_0)\,,\ \beta_2 = \exp\,(f/M_1)\,,\ M_1 = f/\ln\,(\beta_0 - \xi\,),\quad M_2 = f/\ln\,(\beta_1 - \xi\,)\,, \qquad .$$

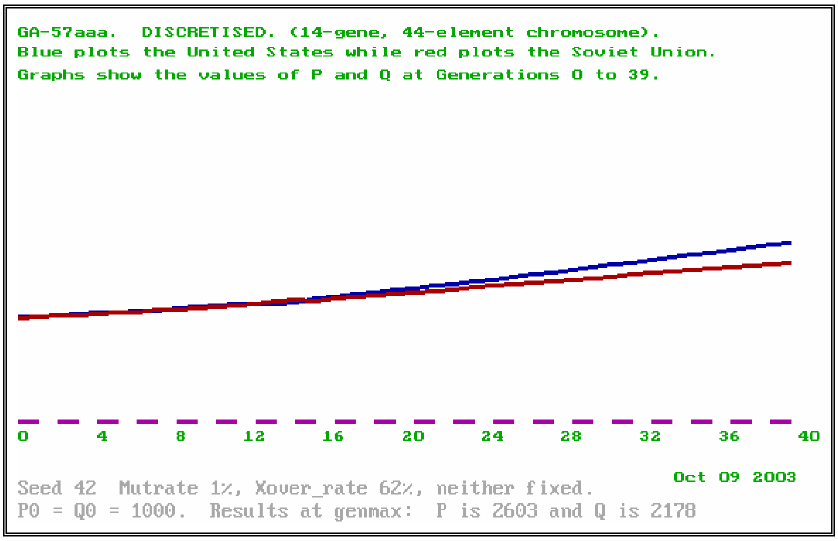
$$\beta_3 = \exp\,(f/M_2),\quad \beta_4 = \exp\,(f/M_3)\,,\ M_3 = f/\ln\,(\beta_2 - \xi\,),\ \ M_4 = f/\ln\,(\beta_3 - \xi\,)\,, \qquad .$$

$$\beta_5 = \exp\,(f/M_4)\,,\ \beta_6 = \exp\,(f/M_5),\quad M_5 = f/\ln\,(\beta_4 - \xi,)\,,\ M_6 = f/\ln\,(\beta_5 - \xi\,)\,,\quad \text{etc.}$$

It was again necessary to perturb β very slightly (by $\xi = 0.0002$) in order to start convergence. This was a pity, for the desire to avoid perturbation had spawned this trial. Starting, as before, with $M_0 = 3229$, $M_1 = 3214$ while $M_{65} = 2368$ and $M_{66} = 2365$. Hence asymptotic convergence had occurred; indeed, at 66 iterations it was significantly quicker than the earlier Gauss-Seidel at 150 iterations. At this point β was 0.738.

7.16.5 *A Monte Carlo Method*

In Section 5.8.1 we used a Monte Carlo method to see if we could predict instabilities in the India/Pakistan situation, and while reasonable numerical results were obtained, the lack of sign changes mitigated against it. The idea of the Monte Carlo method is that an integral may be represented by a sum , namely



GA-57aaa. DISCRETISED. (14-gene, 44-element chromosome).
Blue plots the United States while red plots the Soviet Union.
Graphs show the values of P and Q at Generations 0 to 39.

Seed 42 Mutrate 1%, Xover_rate 62%, neither fixed.
P0 = Q0 = 1000. Results at genmax: P is 2603 and Q is 2178

CHAPTER 8

The Effects of Redundancy in the Chromosomes of an Arms Race GA

8.1 Introduction

Several authors, including the present one, have found that a modest redundancy in chromosomes improves the performance and robustness of GAs. At the Genetic and Evolutionary Computation Conference in Orlando, Florida, July 1999, four papers involving redundancy [Levenick, 1999; Julstrom, 1999; Raich, 1999; Hackworth, 1999] were presented. All four authors had found the same thing in different fields, and although each of them had put forward different reasons as to why this should be so, all agreed that it was a manifestation of the exploration/exploitation dilemma [Holland, 1992].

A sizeable literature on redundancy in the chromosomes of a GA is now readily available[1]. Our interest only goes as far as determining whether or not redundancy will enhance the performance of our arms race model and, if so, how to exploit its beneficial effects.

Much has been written [Cohen and Stewart, 1994] that the biological chromosome is hugely redundant. Apes and humans have, it seems, millions of genes which have no apparent role. These genes (called *introns*)[2] are widely regarded by geneticists as 'junk', although they are freely interspersed with other genes (*exons*) which are vital to our everyday existence and continued well-being. Nevertheless, *all* our genes are mutated, selected for reproduction (or not) and crossed-over, no matter whether they have a function or not. It was decided to apply this principle to our GAs, and to test whether or not redundancy in an electronic chromosome resulted in changed behaviour or a different output.

A particular real-valued GA involving an arms race between two nations, India and Pakistan (see Chapter 5), needs ten genes (five per nation). It so happened chronologically that we had previously been working on the 3-nation (Israel/Egypt/Syria) problem and so had the source code for an integer-coded GA using 15-gene chromosomes (called GA-4) readily available. In the interests of speed, the 15-gene chromosome was used for the 10-gene India/Pakistan problem (under the name GA-26) effectively making the chromosome 33% redundant. The redundant genes, the introns, were not left empty, but were filled randomly

---

[1] See, *inter alia*, the bibliographies maintained at the University of Sussex (http://www.cogs.susx.ac.uk/lab/adapt/nnbib.html) and at the Santa Fé Institute (http://www.santafe.edu/sfi/publications/95wplist.html)

[2] *Introns* are defined biologically as segments of transcribed (functional) genes which are removed during RNA maturation. Technically, there is much more redundant DNA other than introns. For simplicity, however, the word *intron* is used in the thesis (as in most EA literature) to refer generically to *all* redundant DNA

with integers (1 to 999), selected (or not) for reproduction, crossed-over and mutated in the normal way.

The results from GA-26 were robust, but it did seem appropriate to develop a new GA with no redundancy in the chromosome, and this was called GA-0. It was later found (by interchanging genes in two more GAs, GA-27 and GA-28, see Section 8.6) that GA-0 was epistatic. Finally, another GA was developed with twenty-gene chromosomes of which (as before) only ten genes were used, so there was 50% redundancy. It was called GA-9.

In essence, this chapter compares the behaviour and performance of three GAs, all created *as identical as possible,* and all looking at the same problem from different chromosomal viewpoints. The first GA, GA-0, uses ten-gene chromosomes, thus having no redundancy; the second, now called GA-1, has fifteen gene chromosomes and hence 33% redundancy, and the third, GA-9, has twenty gene chromosomes and consequently 50% redundancy.

In summary, the search space of GA-0 is 10^{24}, that of GA-1 is 10^{36}, and that of GA-9 is 10^{48}.

8.2 The Canonical GA

The ten/fifteen/twenty genes of the three GAs are set up as *genotypes*, i.e. populations of encoded chromosomes, and initially filled from a constrained RNG. Genes from each string are then used appropriately in the two *evaluation equations* taken from the Mayer-Kress [1989] discretisation of the Richardson arms race Eqs(5.3), namely

$$x_{t+1} = x_t + \left(k_{1,1}(x_s - x_t) + k_{1,2}\, y_t\right)(x_m - x_t) \quad .$$

$$y_{t+1} = y_t + \left(k_{2,2}(y_s - y_t) + k_{2,1}\, x_t\right)(y_m - y_t)\,. \qquad (8.1)$$

All three algorithms used Eqs(5.7) as evaluation equations, namely

$$x_{t+1} = x_t + \left((k_{1,1}/100)\,(x_s/100)\,x_t + (k_{1,2}/100)\,y_t\right)(x_m - x_t)/85 \quad .$$

$$y_{t+1} = y_t + \left((k_{2,2}/100)\,(y_s/100)\,y_t + (k_{2,1}/100)\,x_t\right)(y_m - y_t)/85. \qquad (8.2)$$

The best solution in each generation is chosen in accordance with the *fitness function* given by Eq(5.5), whose role is to highlight those chromosomes which best fit the requirements of the situation under consideration. Roulette wheel selection (see Section 4.3) is then used to choose individuals on the basis of their fitness, giving the ‘better’ individuals (i.e. those with greater fitness) a chance to reproduce more frequently than the poorer ones. The selected chromosomes are then mutated, crossed-over, and reassembled, and then taken into use as the new generation replacing the old. If the figures converge at all, they do so quite quickly, typically in twenty generations in this type of application.

8.2.1 Fundamental Issues

Before proceeding any further, three fundamental points must be made:

- The genotype is filled directly with integers, which are the contents of the phenotype on a one-to-one mapping basis, the only embryogeny being the occasional requirement to divide certain genes by factors of ten in order to compute the evaluation equations. Actual arms race phenotype figures typically lie in the hundreds or thousands and so lend themselves naturally to direct mapping. Direct mapping may be unusual but, as Bentley [1999] argues, phenotype representation does not *have* to be different from genotype encoding. Nevertheless, finding a good coding may be the single most important factor in the success or failure of a GA. If we use a bad coding the GA may perform worse than other probabilistic search methods [Miettinen *et al*., 1999].
- During mutation, every element (locus) of every gene is sampled individually to see if it should be mutated or not, so mutation is quoted in terms of a *mutation probability per locus*.
- The meaning of the word *redundancy*. An *intron* can mean two things:
    - An inactive gene which is lying around waiting to be *expressed* or *activated*, the principle being to introduce additional genetic material when needed in order to adapt to environmental changes. One classic study of non-stationary optimisation [Goldberg and Smith, 1987] created and stored two representations of each gene, but only one was expressed at any one time. The structured GA (sGA) [Dasgupta, 1995] has as its central feature several levels of genes in a hierarchical 'pecking order' together with a gene activation mechanism. The Implicit Redundant Representation GA (IRR GA) [Raich and Ghaboussi, 1999] uses redundancy to represent a variable number of location-independent design parameters. Put in more general terms, several distinct chromosomes represent each candidate solution to the target problem [Julstrom, 1999].

- A gene which is an integral part of its chromosome but is not used in any evaluation equation or in any other calculation. It is just there. It is not left empty, but is initially filled randomly with reasonable integers (e.g. 1 to 99 or 1 to 999, see (12) of Section 4.2), selected (or not) for reproduction, crossed over and mutated in the normal way. It may have to be constrained, for 'wild' genes can grow excessively large and may disrupt their neighbours for reasons which are discussed in Section 8.3.

In this chapter we employ the second type of intron. At first sight, the inclusion of redundant genes (or any other superfluous material) might seem to hinder genetic search. We will demonstrate experimentally that in fact the insertion of 'useless' introns brings a number of advantages (see Section 8.9).

8.2.2 The Three GAs

The GAs under experimental investigation, GA-0, GA-1 and GA-9, were made as identical as possible and then compared. (They had individually been prepared at separate times for disparate purposes, so initially they were different in many ways). They were not the only GAs which could have been chosen for the experimentation; this is one case among many others. The three GAs ultimately had a population of 100, and were run for 40 generations. For the ensuing presentation, the GAs employed a seed of 7,[3] and an average defexp% of 6.5%. Except where otherwise stated, mutation *increased* [4] by 0.0005 at each iteration (generation). The single-point crossover started at 0.65 and *decreased* by 0.002. The evaluation equations were Eqs(8.2), with scaling factors chosen so that the shape of the graphical output of the GA bore a fair resemblance to the IMF plot, see Chapter 5.

The configuration of the genes in the three GAs is as shown in Tables 8.1, 8.2 and 8.3 below. The introns are shaded. The parameters are those found in Eqs(8.2), all of which were represented as 2- or 3-digit integers. The reason for choosing this particular configuration for GA-1 (Table 8.2) is that the shaded introns were originally genes associated with the third nation which were, literally, made redundant in the case of the 2-nation scenario. There was no special reason for choosing the 'alternates' configuration of GA-9 (Table 8.3). In neither GA-1 nor GA-9 did the actual point (locus) of crossover play any part in the placing of the introns; this will be considered later (Section 8.9).

---

[3] Many other seeds were of course used, and they achieved substantially the same results. For demonstration purposes it was thought better to concentrate on just one seed so that everything was consistent

[4] Received wisdom would reduce the mutation rate and increase the crossover rate at each iteration (generation). It was found that doing the reverse made the GA more robust

Table 8.1 Configuration of genes in the chromosome of GA-0. No redundancy[5]

Column	2	3	4	5	6	7	8	9	10	11
GA-0	$k_{1,1}$	$k_{2,2}$	$k_{1,2}$	$k_{2,1}$	x_s	y_s	x_m	y_m	x_t	y_t

Table 8.2 Configuration of the 15 genes in the chromosome of GA-4 (designed for a 3-nation problem) and GA-1. The redundancy here is 33% (10 genes used out of a possible 15)

Column	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
GA-4	$k_{1,1}$	$k_{2,2}$	$k_{3,3}$	$k_{2,3}$	$k_{1,2}$	$k_{1,3}$	x_s	y_s	z_s	x_m	y_m	z_m	x_t	y_t	z_t
GA-1	$k_{1,1}$	$\mathrm{k}_{2,2}$	▨	▨	$k_{1,2}$	$k_{2,1}$	x_s	y_s	▨	x_m	y_m	▨	x_t	y_t	▨

Table 8.3 Configuration of genes in the chromosome of GA-9.
The redundancy here is 50% (10 genes used out of a possible 20)

Column	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
GA-9	$k_{1,1}$	▨	$k_{2,2}$	▨	$k_{1,2}$	▨	$k_{2,1}$	▨	x_s	▨	y_s	▨	x_m	▨	y_m	▨	x_t	▨	y_t	▨

8.2.3 Ensuring Similarity

By making all the settings identical where possible, the three GAs were made as similar as they could be. This posed certain problems:

(1) *Random Numbers.* GAs make extensive use of RNG. The output of these RNG is designed, despite their randomness, always to be in the same sequence (so that, in general, results are reproducible). A program implementing a GA with bigger size chromosomes (i.e. ones with more genes or more elements, or both) calls for random numbers a lot more often than one with small size chromosomes, and not only during the GA's initialisation; during mutation every element of every gene is sampled individually to see whether it should be mutated or not. Hence there can be no precise mapping between GAs with chromosomes of different sizes.

(2) *Scaling.* The three GAs, GA-0, GA-1 and GA-9, had originally used different scaling factors, so a mean had to be found if comparisons were to be made. Additionally, the *x, y* inputs to the fitness function, given by Eq(5.5), must be scaled so that they lie between ± 100, otherwise the results are not valid [Schaffer *et al.,* 1993]. At best one can compromise and hope that juggling the figures to match these two requirements does not distort the performance of (and hence invalidate the comparison between) these GAs.

---

[5] Each population member in a GA consists of a chromosome embedded in a lot of other administrative and numerical information, making up quite a long string. For a typical example see Table 4.1. In our GAs, columns 0 and 1 are always used for generation and serial number, respectively (for identification purposes). Hence the first column of the chromosome is always 2, and the last one is 11 (10-gene), 16 (15-gene) or 21 (20-gene)

(3) *Mutation.* Every element (in every gene) in the chromosomes of GA-0, GA-1 and GA-9 is sampled individually to see if it should be mutated or not. Although it is customary to refer to a *mutation rate*, it would be more correct to refer to a *mutation probability per element* (or *locus*). When we started, a count was put on the RNG of the three GAs, and it was noted that GA-9 with twenty genes called for an RNG upwards of 120,000 times per run, while GA-0 with ten genes required less than half that number. In the context of mutation, therefore, the effect of introducing introns was merely to scale the mutation probability up or down, and this is neither productive nor illuminating. This was overcome by altering the mutation rate appropriately. For GA-0 (ten genes) it was set to 1.4%, for GA-1 (fifteen genes) it was set to 1.05%, and for GA-9 (twenty genes) it was set to 0.7%. Hence the same overall mutation is applied to each chromosome irrespective of its length, and this ensures a more equitable comparison.

(4) *Crossover.* Crossover, although essential, is fundamentally disruptive. In Section 4.3 it was explained that our GAs are designed so that mutation and crossover can take place *uninhibited by cell boundaries.* This means, for example, that crossover can occur in the middle of a gene. Far from being undesirable, this should (and is intended to) reduce the disruptive effect of crossover (at the risk of introducing a small level of 'quantum' noise from the 'bin-ends'). Levenick [1991] showed that introns reduce schema loss due to crossover. Indeed, he demonstrated that crossover provides a mechanism for evolution in general as intron regions are not (directly) subject to selection pressure and thus may vary widely and so more of the parameter space is searched. It also reduces the effects of epistasis, and so makes the algorithms more robust. We shall see if this is borne out in our experiments.

8.2.4 Outputs

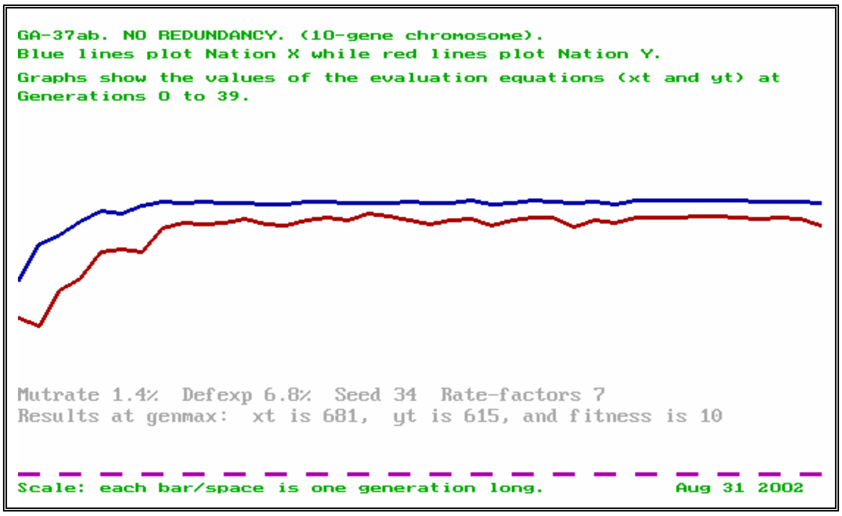
Graphical outputs from the three GAs, plotting defence expenditures x_t (blue) and y_t (red) against forty generations are shown at Figures 8.1, 8.2 and 8.3. From these graphs it would appear that increasing redundancy makes convergence take place later; is this good? To answer this, we need to consider two types of convergence:

(1) *Homogeneous convergence*, see (5) of Section 2.13. A little of this is desirable, but too much induces a loss of genetic diversity which incapacitates crossover and inhibits exploration.

(2) *Diverse convergence*; see (6) of Section 2.13. " … the maintenance of a diverse population after convergence is reached" [Levenick, 1991]. This would be an ideal, for it would combine exploration *and* exploitation, Holland's basic dilemma [1992].

We believe that the chromosomes of GA-0, GA-1 and GA-9 *are* maintaining some level of genetic diversity – albeit not as much as we would have liked – and that this is not only demonstrated by a comparison between Figures 8.1, 8.2 and 8.3, respectively, but also by the fact that GA-1 and GA-9, when run, are noticeably more robust than GA-0.

(*Note.* Figures 8.1, 8.2 and 8.3 are all Option 8 graphs using a defexp% of 6.8% throughout (see Section 4.8). Population is 100, the seed is 34, and the rate-factors are the same for the three GAs. The final values of x_t, y_t and *fitness* at genmax (Generation 39) are shown in grey print. The reason for the different mutation rates is explained in (3) of Section 8.2.3).



GA-37ab. NO REDUNDANCY. (10-gene chromosome).
Blue lines plot Nation X while red lines plot Nation Y.
Graphs show the values of the evaluation equations (xt and yt) at Generations 0 to 39.

Mutrate 1.4% Defexp 6.8% Seed 34 Rate-factors 7
Results at genmax: xt is 681, yt is 615, and fitness is 10

Scale: each bar/space is one generation long. Aug 31 2002

Figure 8.1 The performance of GA-0 (x_t and y_t against generations).
No redundancy. Mutation rate 1.4% Convergence starts at Generation 10

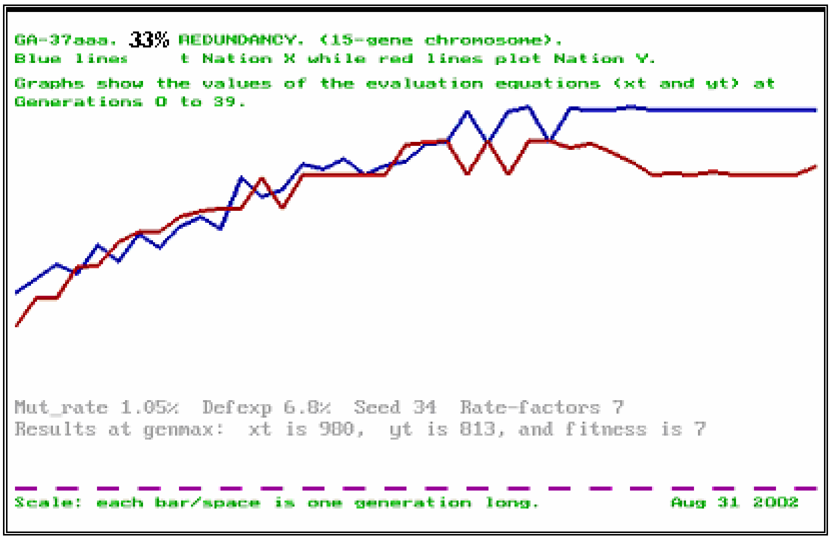


Figure 8.2 The performance of GA-1 (x_t and y_t against generations).
33% redundancy. Mutation rate 1.05%. Convergence starts at Generation 32

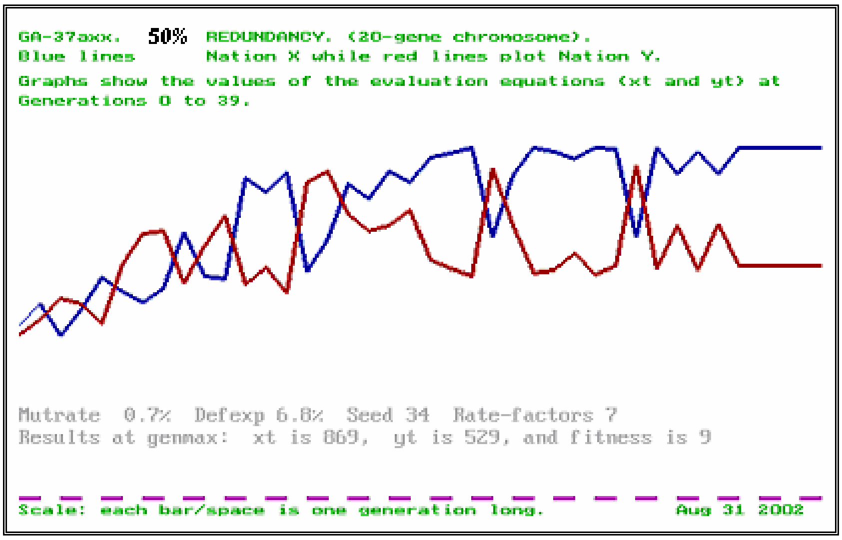


Figure 8.3 The performance of GA-9 (x_t and y_t against generations).
50% redundancy. Mutation rate 0.7%. Convergence is uncertain until Generation 36

8.3 An Architectural Difficulty

Great care was taken over the architecture of the original GAs. Section 4.3 explained that one of their features was that they took pairs of rows of genes and 'unbundled' them into two columns so that mutation and crossover could take place *uninhibited by cell boundaries*. This was a deliberate ploy which worked well until it proved necessary to write a new GA, GA-9, to cater for 50% redundancy. It was suddenly found that many runs did not reach completion, being interrupted by floating-point errors. Examination of the stage-by-stage diagnostic tables revealed that whereas the 'active' genes – the exons – were behaving impeccably (they are, after all, constrained for each generation by the evaluation equations), some of the redundant genes had grown uncontrollably (by the thirtieth generation the values of two of them had exceeded 10^5). Furthermore, these 'rogue' cells appeared to be infecting their neighbours. The reason was that some of the redundant genes had grown under mutation to the point where they occupied more than their allotted number of elements in the gene and so 'overflowed' into the space belonging to the two neighbouring genes. In the 'unbundled' state (where mutation takes place) there was nothing to stop an overflow happening. When it did happen, the 'rebundled' genes were corrupted. The answer to this problem was to limit all the redundant genes at every generation to the number of elements catered for in the unbundling and rebundling processes. This situation had considerable nuisance value at the time, and was a classic instance of a deliberate and desired design feature turning into a near-disaster.

8.4 What the Introns do

Our introns are not used for any purpose, unlike those referred to in Section 8.8, where there are mechanisms for translating introns into exons and vice versa dynamically. Since the introns are not used, they are not directly subject to selection pressure, neither can they help to find new solutions, nor enhance the *useful* search space. However, they are not totally useless; they do affect the mutation probability, and they also affect how the cross-over operator tears up good building blocks.

8.5 Predicting Instability

Figures 8.4, 8.5 and 8.6 plot the sign of $a - c$ which is an approximate measure of the direction of curvature of the limit cycle. (As in Chapters 5 and 6, ten or more sign changes indicate an impending instability).

In particular, the three figures predict instabilities in GA-0, GA-1 and GA-9, respectively. As explained in Section 3.8, instabilities between warring parties are always prefaced by a sharp drop in the number of minus signs. Sharp drops can clearly be seen in all three figures at average defexp% of about 6.6% and 8%.

From this we conclude that the *changes in the direction of curvature of the limit cycle,* so vital in predicting instability, have not been altered by redundancy (in that sharp drops still occur in the same places). It is difficult to quantify the generality of the conclusion, i.e. that changes in the direction of curvature of the limit cycle are unaffected by redundancy in the chromosomes, but it seems valid in the context of our application.

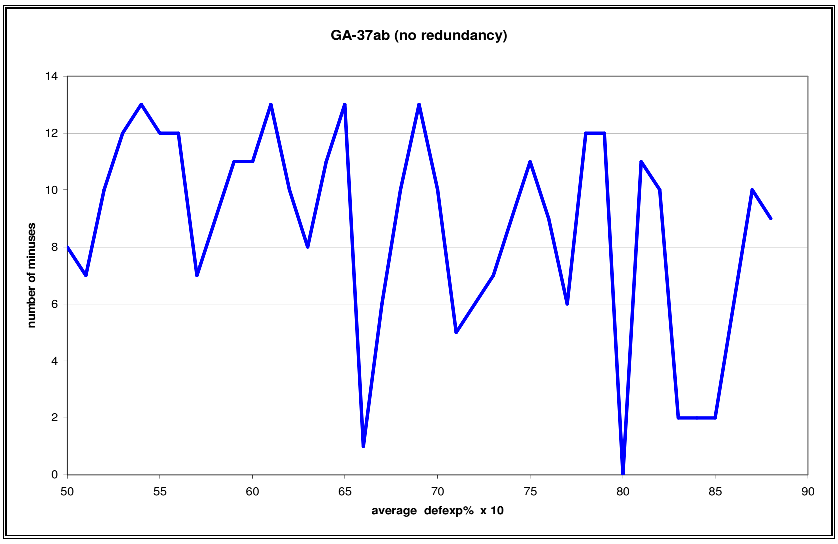


Figure 8.4 Prediction plot for GA-0 (no redundancy)

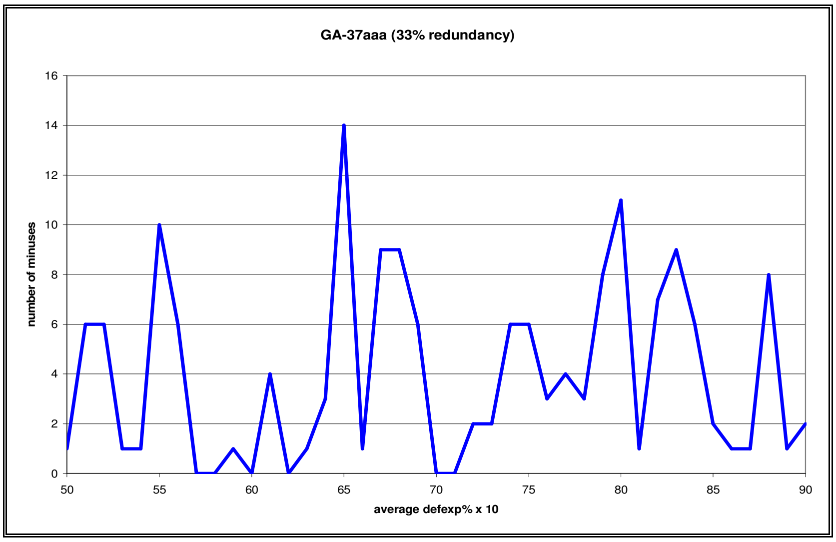


Figure 8.5 Prediction plot for GA-1 (33% redundancy)

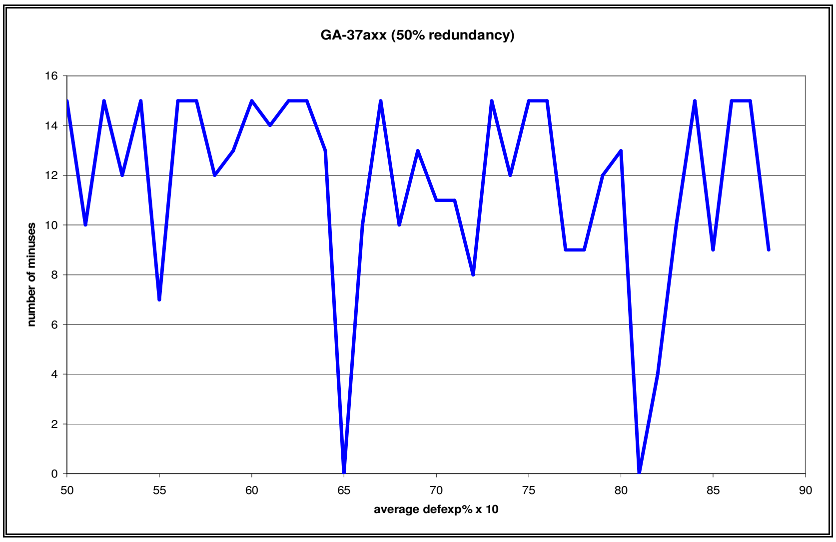


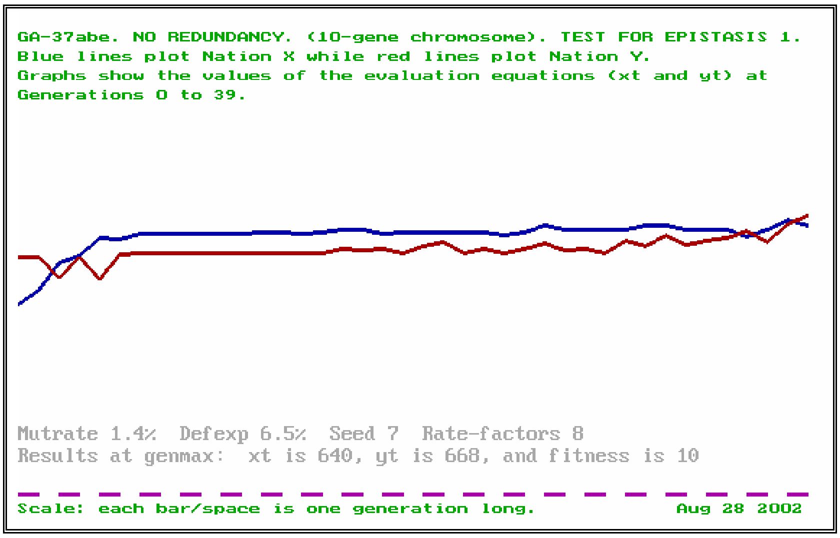
Figure 8.6 Prediction plot for GA-9 (50% redundancy)

8.6 Epistasis

In Section 2.12, we said that the degree of dependent parameter interaction is called *epistasis*. Epistasis is about genes acting in combination to produce (or inhibit) solutions. In GAs, less epistasis is preferable to more, and this is one reason for introducing redundancy into chromosomes, for redundancy, within reason, is known to reduce epistasis. To see the effects of epistasis, we interchanged the positions of several genes in GA-0, resulting in two other GAs, GA-27 and GA-28, as shown in Table 8.4. In GA-27 all the x terms were moved to the left of two rate-factors, while the y terms were contiguously placed to the right. This representation leaves each evaluation equation essentially intact, even after crossover. Hence its output (Figure 8.7) is not dissimilar from that of Figure 8.1. In contrast, in GA-28 like stays with like even after a crossover, so x_t and y_t are likely to end up in the same individual. Hence the output is likely to be very different from the other two GAs and so it is, as can be seen in Figure 8.8.

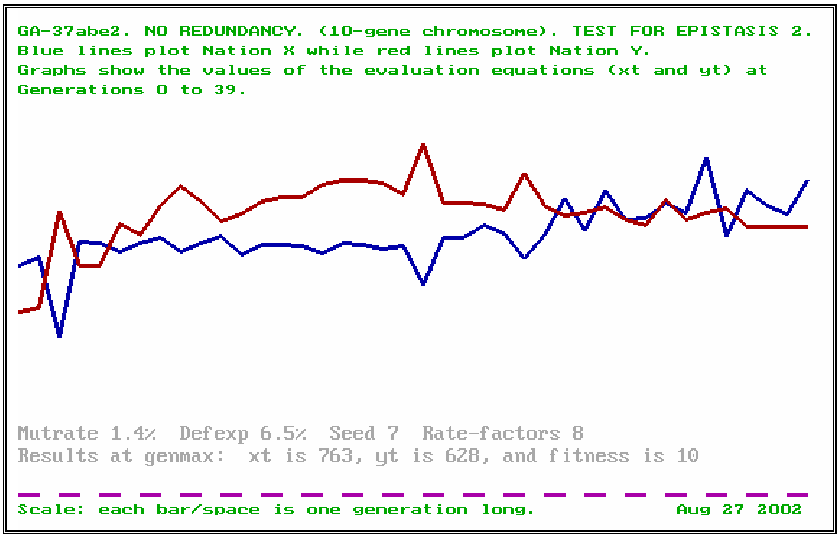
Table 8.4 Configuration of genes in the chromosomes of the non-redundant GA-0, GA-27 and GA-28 which test for epistasis. Note the interchange of genes between the three

genes	2	3	4	5	6	7	8	9	10	11
GA-0	$k_{1,1}$	$k_{2,2}$	$k_{1,2}$	$k_{2,1}$	x_s	y_s	x_m	y_m	x_t	y_t
GA-27	x_t	x_s	x_m	$k_{1,1}$	$k_{1,2}$	y_t	y_s	y_m	$k_{2,1}$	$k_{2,2}$
GA-28	x_t	y_t	x_s	y_s	x_m	y_m	$k_{1,1}$	$k_{2,1}$	$k_{1,2}$	$k_{2,2}$



```
GA-37abe. NO REDUNDANCY. (10-gene chromosome). TEST FOR EPISTASIS 1.
Blue lines plot Nation X while red lines plot Nation Y.
Graphs show the values of the evaluation equations (xt and yt) at
Generations 0 to 39.

Mutrate 1.4%  Defexp 6.5%  Seed 7  Rate-factors 8
Results at genmax:  xt is 640, yt is 668, and fitness is 10

Scale: each bar/space is one generation long.            Aug 28 2002
```

Figure 8.7 The performance of GA-27 (x_t and y_t against generations). No redundancy. Mutation rate 1.4% Convergence starts at Generation 6



```
GA-37abe2. NO REDUNDANCY. (10-gene chromosome). TEST FOR EPISTASIS 2.
Blue lines plot Nation X while red lines plot Nation Y.
Graphs show the values of the evaluation equations (xt and yt) at
Generations 0 to 39.

Mutrate 1.4%  Defexp 6.5%  Seed 7  Rate-factors 8
Results at genmax:  xt is 763, yt is 628, and fitness is 10

Scale: each bar/space is one generation long.           Aug 27 2002
```

Figure 8.8 The performance of GA-28 (x_t and y_t against generations).
No redundancy. Mutation rate 1.4% No convergence

8.7 Testing for Epistasis

Naudts *et al.* [1998], taking a slightly different viewpoint, state that epistasis is the lack of independence amongst genes with respect to the fitness function. In other words, it is a measure of how important two loci are in determining a chromosome's fitness. In our experiments with GA-0, differing arrangements of genes within the same (non-redundant) chromosome gave different answers and this indicates the influence of epistasis. According to de Jong *et al.* [1997], the conventional wisdom is that recombination (crossover) should have a relative advantage over mutation when epistasis is small, while mutation should have the edge when epistasis is high. It was decided to test the level of epistasis by developing three new GAs based on the architecture of a problem generator [Kauffman, 1989]. These GAs were designated GA-29, GA-30 and GA-31 and correspond to GA-0, GA-1 and GA-9. The genes are as before, but the ffn used is the inverted 'Fitness Function F6' designed by Schaffer *et al.* [1993] for the abstract examination of GAs.

F6 decodes the chromosome (or pertinent parts thereof) and generates two numbers lying between ±100 which it plugs into the ffn

$$0.5-\frac{\sin^{2}\sqrt{x^{2}+y^{2}}-0.5}{\left(1.0+0.001\left(x^{2}+y^{2}\right)\right)^{2}}\ , \tag{8.3}$$

whose output lies between 0.0 and 1.0.

Following the examples from de Jong *et al.*'s [1997] problem generators, curves for fitness were drawn for all three GAs against the number of generations; one using only crossover, *GA-c* (green), one using only mutation, *GA-m* (red), and one using both, *GA-b* (blue). The degree of epistasis can then be assessed by pictorial comparison with de Jong *et al.*'s [1997] curves. Put crudely, the wider apart the three curves are, the lower the epistasis is.

The population for all three GAs was 100 chromosomes, the crossover rate a typical 0.6, and the initial mutation rates were 1.4%, 1.05% and 0.07%, respectively, to ensure an equitable comparison, see (3) of Section 8.2.3. Each run covered 100 generations.

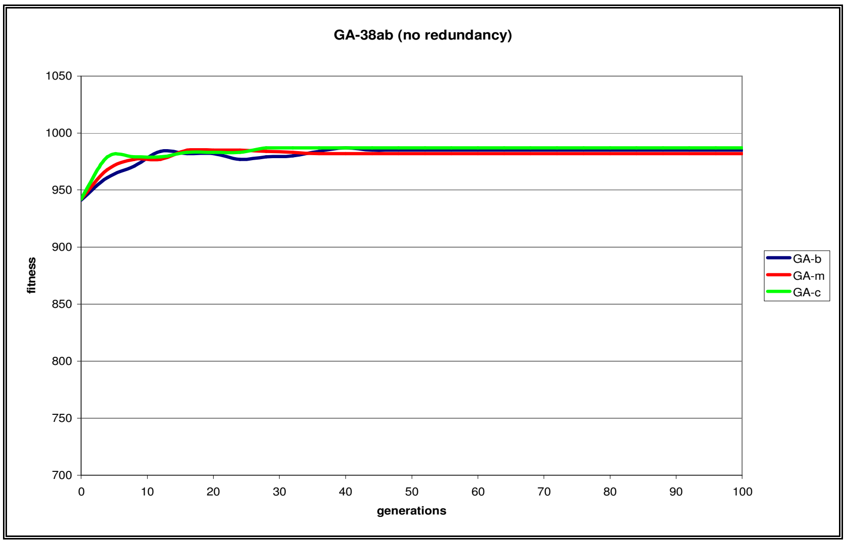


Figure 8.9 GA-29 (no redundancy). Seed 7. Comparative fitnesses for GA-b, GA-m and GA-c. defexp% 3.3, initial mutation rate 1.4% minus 0.0001% per generation for 100 generations, initial crossover rate 0.6 % plus 0.0006% per generation. Note the offset zero; this is for pictorial clarity

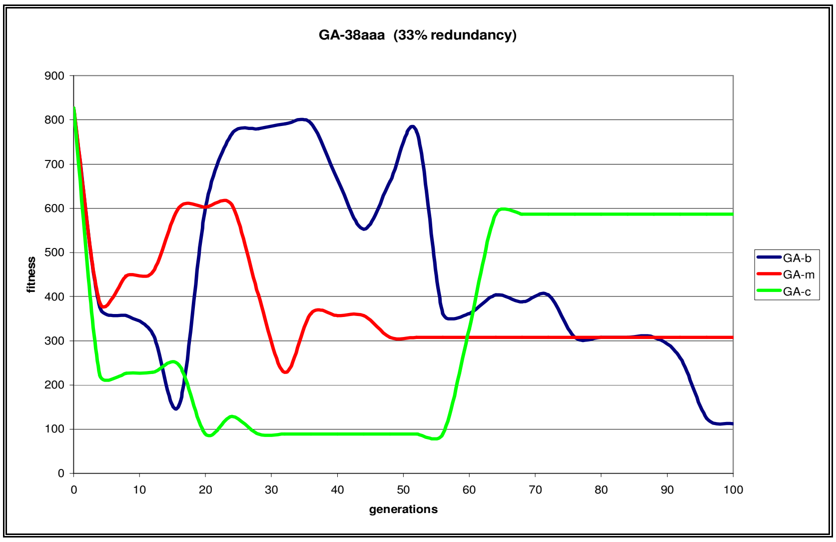


Figure 8.10 GA-30 (33% redundancy). Comparative fitnesses for GA-b, GA-m and GA-c

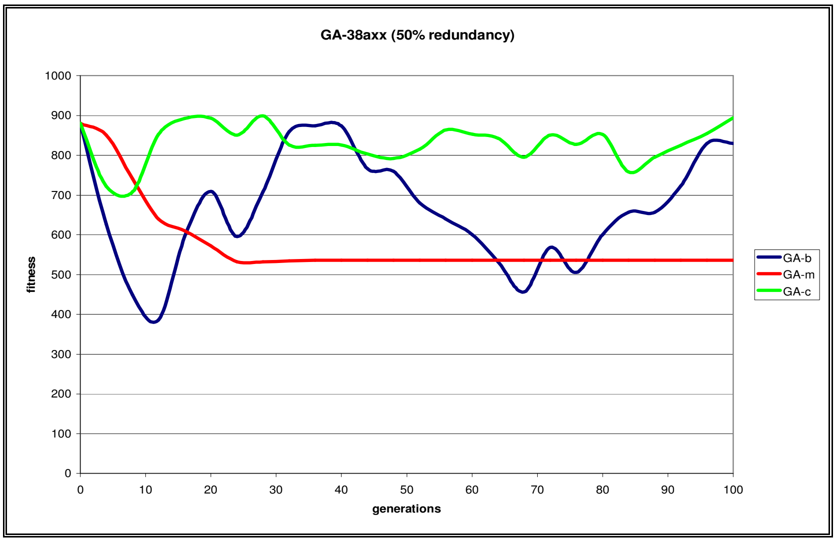


Figure 8.11 GA-31 (50% redundancy). Comparative fitnesses for GA-b, GA-m and GA-c

de Jong *et al.* [1997] lead us to believe that (the standard) *GA* will dominate on low epistasis problems, and it does. As epistasis increases *GA-m* does better. As de Jong *et al.* [1997] say, crossover and mutation do not operate in a vacuum. They are components which interact in complex ways with all the other components of an evolutionary system, and the effects of their synergy can be seen in the charts above. The performance of GAs depends on a proper balance in the next generation between exploration, via genetically different offspring, and exploitation via genetically identical offspring. Too high a default mutation rate will result in little or no cloning and this could be catastrophic for these simple, non-elitist GAs.

Figures 8.9, 8.10 and 8.11 show clearly that imposing redundancy separates the *GA-b, GA-m* and *GA-c* curves. By de Jong *et al.*'s [1997] criteria, therefore, the two redundant GAs are less epistatic than the non-redundant one. Additional experiments were run to test the sensitivity of these results to various GA parameter settings such as the rate-factors, mutation, and crossover. In most cases, mutation had the bigger influence. As de Jong *et al.* [1997] also say, " … having problem generators allows us to report such results over a randomly generated set of problems rather than a few hand-chosen examples. By increasing the number of problems we increase the predictive power of the results for the problem class as a whole".

"Three swallows do not make a summer", neither do a few hundred experiments by one individual prove anything. There is little doubt, however, that there are grounds for believing that redundancy leads to a reduction in the levels of epistasis. This of itself brings attendant benefits to our GAs.

8.8 Related Work to Redundancy

There have been several studies on GAs where the fitness ('objective') function changes over time [Cobb, 1990; Grefenstette, 1992], the principle being to introduce additional genetic material in the population as and when needed to adapt to environmental changes. In particular, Goldberg and Smith [1987] introduced redundancy by creating and storing two representations of each gene, but only one was *expressed* at any one time, according to some dominance mechanism which also changed the fitness function dynamically. Changing the fitness function in this way, however, is difficult to engineer and the technique has never found general application.

8.8.1 Structured GAs

Dasgupta and McGregor [1994] created a s*tructured Genetic Algorithm* (sGA) aimed at tracking an optimum in time-varying environments. The model incorporated redundancy and used a gene activation mechanism allowing multiple changes to occur simultaneously. Such changes could be both frequent and irregular. In a standard GA, phenotypic convergence normally leads to genotypic homogeneity and the GA loses its ability to search for a new optimum. Hence standard GAs are not suitable for non-stationary function optimisation. In other words, standard GAs do not like having their goal-posts moved! The reason for this is that the simple chromosome does not have sufficient genetic diversity to allow the search to continue as the environment changes. Dasgupta and McGregor [1994] make a point of saying that to correct this would necessitate multiple correlated mutations but this would introduce its own problems, and is likely to lead to degenerate solutions.

In contrast, an sGA changes genes dynamically by playing on several levels of genes in a hierarchical 'pecking' order. Apart from this an sGA is entirely conventional.

Redundant or *unexpressed* genes serve a dual purpose in sGAs:

- They provide non-destructive diversity during the search process. Since only the expressed genes undergo selection pressure, Dasgupta [1997] believes that "....the unexpressed genes are neutral although they experience silent genetic changes".
- They work as a distributed memory of variation within the population, allowing the model to be efficient in non-stationary and rapidly–changing environments.

In other words, an sGA may introduce extra genes into its chromosomes, but the unexpressed genes have to be controlled just as carefully as the expressed ones. The lesson for us is, first, *not* to move the goalposts and, second, to keep control of the search space.

8.8.2 Swappers

Levenick [1999] modelled the performance of E.coli bacteria fed variously on glucose, lactose and fructose. It seems that E.coli, while primitive, is able to adapt from living on one to either of the other two rather fast. Seemingly, it does this by changing its genes. Once it begins to starve, it activates certain introns (making them into *exons*). Levenick called these dual-role genes *swappers*.

Levenick [1999] argues that introns offer solutions to the plasticity/stability and exploration/exploitation dilemmas. The first of these is the ability to adapt to new situations without losing the capability of handling previously mastered ones. The second is the dichotomy between using knowledge (i.e. information already acquired) to exploit the current situation, and exploring the environment in order to discover better strategies for future use. If a system spends too much energy profiting in the short term it may fail to discover some superior strategy and so fail in the long term. Conversely, if it uses too much energy exploring, it may not survive to use the knowledge that it gains. The lesson for us is to keep a balance between the two.

8.8.*3 Implicit Redundant Representation GAs* (IRR GAs)

Raich and Ghaboussi [1999] were looking for an evolutionary method for synthesising alternatives for the design of roof trusses and other structural frames. They wanted to encode a variable number of design parameters, making them both independent of their location within the chromosome and allowing self-organisation of their linkage. The IRR GA provides a mechanism which allows essential and redundant sections of a string to interact dynamically, and it does this by using a [chromosome] string length longer than that required to encode only the parameter values. The specific location of each parameter (a gene *instance*) is not designated explicitly. Instead, each instance is allowed to drift along the length of the string, a record being kept of its whereabouts by a pre-selected g*ene locator* which also specifies the number of ‘useful’ bits which encode the parameter values. When decoded, anything that is not part of an instance (such as bits separating the instances) is redundant. New instances can be created dynamically by mutation or crossover, for they do not need to be specified by the designer. It is of particular interest to us that at no time did Raich and Ghaboussi [1999] record that their undefined, unconstrained and redundant strings had caused them any problems.[6]

8.8.4 Similarities and Differences

Similarities with the present work are that sGAs, Swappers and IRR GAs do contain a number of redundant (unexpressed) genes, mainly defined but sometimes undefined, which are processed in the same way as every other gene. In fact, the unexpressed genes of sGAs and Swappers are viable alternatives to the expressed genes, and their behaviour is controlled just as carefully. In marked contrast, the redundant genes of the GAs discussed in the thesis were originally intended to encourage robustness and diversity; they were not intended to be controlled or restrained at all. That they had sometimes to be restrained indicates that they can be a mixed blessing.

The *differences* are that at some stage the redundant genes in sGAs, Swappers and IRR GAs will be activated (or brought into effective play in some other way) and therefore play some active part in (or influence) search, or the computation of the evaluation equations. Our redundant genes, in contrast, do no such thing.

One anonymous referee unkindly coined the epithet *totally useless redundancy* (TUR). Why, therefore, pursue it?

8.9 Does Totally Useless Redundancy Confer Benefits?

(1) On theoretical grounds, it is very unclear how the variation (by mutation) of unused genes can be in any way helpful in searching the problem space. The genetic variation which happens in this way plays no part in the computation of the evaluation equations at any time, even indirectly, unlike explorations by sGAs, Swappers and IRR GAs where genes can be switched on and off during a run. It appears that the only real effect of adding redundant genes is fed back via crossover which may be less disruptive when chromosomes contain redundancy. This will be explained in (5).

(2) TUR undoubtedly slows convergence (as has been demonstrated experimentally in Figures 8.1 to 8.3) but the non-converged portions of the chromosome are likely to be the redundant parts which, while not experiencing any direct selection pressure, are of no benefit in finding new solutions.

---

[6] At GECCO in July 2003, Anne Raich confirmed verbally that there had been no problem with their unconstrained strings

(3) TUR will reduce epistasis when averaged over *all* the chromosomes (including the redundant parts) but the epistasis between functional parts of the genotype will not be affected because introns in themselves cannot be epistatic. Having said that, epistasis normally has a negative effect − like friction in mechanical systems − but (also like friction) there are methods and techniques for reducing its effects but not eliminating them. We have to learn to live with epistasis. To improve the output of our arms race GAs, we have to look somewhere else. Epistatic interactions are discussed in detail by Kauffman [1993], for example.

(4) TUR can facilitate genetic diversity, but this diversity is not useful simply because it exists in introns.

The consensus view is that the use of introns, which are never expressed, simply scales down the mutation probability which, in turn, prolongs the search. Our trials, however, had already taken account of this. Mutation rates for the three GAs were appropriately scaled to give different chromosome lengths, the same overall mutation, see (3) of Section 8.2.3, *and still* the search by the 15- and 20-gene chromosome GAs was prolonged (see Figures 8.1, 8.2, and 8.3). A possible reason for this is given below, and it has nothing to do with mutation.

(5) The introns have had a smoothing effect on crossover. If introns are used, there is less likelihood of disrupting good building blocks because the chance of slicing an intron (rather than an exon) are higher, and intact values of the exons may be merged together. In other words, *the chances of slicing an active gene and destroying a good solution will be reduced if introns are present.* We know from our diagnostic tables that the majority of crossovers occur *inside* genes, not at their boundaries. In the absence of introns, a crossover *inside* a gene introduces additional ‘mutations’. Ordinary mutation is carefully controlled, but the ‘mutations’ caused by crossover are essentially random and are likely to create two offspring which are significantly worse than their parents.

In running many of the GAs listed on page 13 hundreds of times, it had long been noted that while changing the mutation rate dynamically (given in Section 8.2.2. as a typical 0.0005 per generation for the set of trials demonstrated) could have far-reaching effects on the output results, doing the same to crossover (0.002 per generation, or any other rate) made little difference to the output results. This confirms the work of Dumitrescu *et al.*

[2000] who had also found that varying the crossover rate was much less critical to the output results than varying the mutation rate.

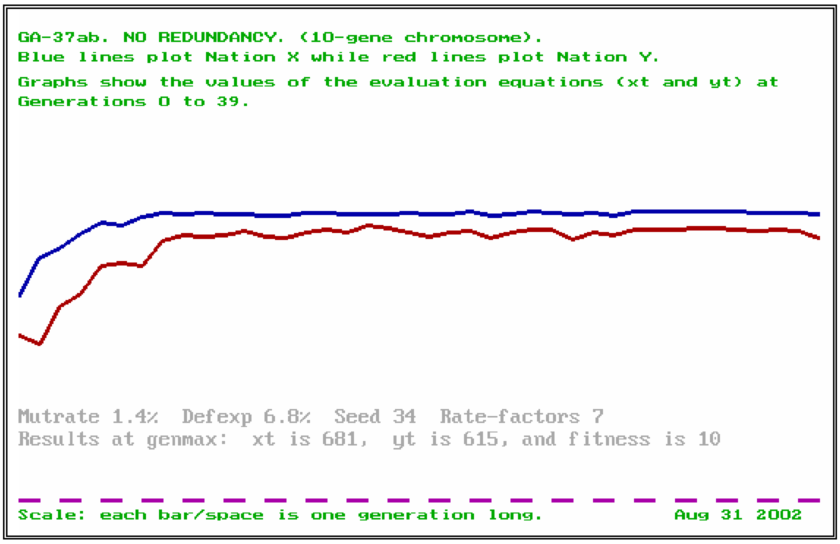
It was decided, therefore, to develop just one more GA, GA-32, and use a *fixed* crossover. GA-32 was copied from the non-redundant, 10-gene GA, GA-0, with the same basic 24 elements in its chromosome, occupying the same element slots (5 to 28), and therefore was *identical* with it, except that in GA-32 we added just one 3-element intron, occupying slots (loci) 20-22, see Table 8.5.

When GA-32 runs with a *fixed* 63% crossover, crossover will always occur[7] at or near element 21, *in the middle of the intron.* We thus avoid splitting an exon at the expense of splitting an intron.

Table 8.5 Configuration of the genes of GA-0, and of GA-32 (including one lone intron)

genes	2	3	4	5	6	7	8	9	10	11	12
elements	5-6	7-8	9-10	11-12	13-14	15-16	17-19	20-22	23-25	26-28	29-31
GA-0	$k_{1,1}$	$k_{2,2}$	$k_{1,2}$	$k_{2,1}$	x_s	y_s	x_m	y_m	x_t	y_t	*not used*
GA-32	$k_{1,1}$	$k_{2,2}$	$k_{1,2}$	$k_{2,1}$	x_s	y_s	x_m	***intron***	y_m	x_t	y_t

Figure 8.12 is the output of GA-0 while Figure 8.13 is that of GA-32. Both GAs used a seed of 34, a defexp% of 6.8% and a mutation rate of 1.4%. For Figures 8.14 and 8.15 we used a seed of 34 and a defexp% of 6.8%, but a mutation rate of 4%. For Figures 8.16 and 8.17 we also used a mutation rate of 4% but a seed of 41 and a defexp% of 7.6%.

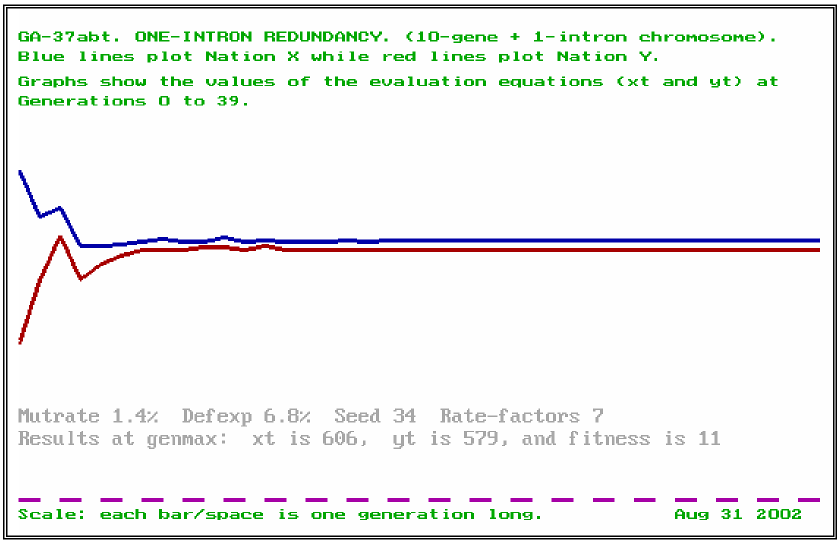


```
GA-37ab. NO REDUNDANCY. (10-gene chromosome).
Blue lines plot Nation X while red lines plot Nation Y.
Graphs show the values of the evaluation equations (xt and yt) at
Generations 0 to 39.

Mutrate 1.4%  Defexp 6.8%  Seed 34  Rate-factors 7
Results at genmax:  xt is 681,  yt is 615, and fitness is 10

Scale: each bar/space is one generation long.           Aug 31 2002
```

Figure 8.12 Output of GA-0. Seed 34, mutation rate 1.4%, and a defexp% of 6.8%

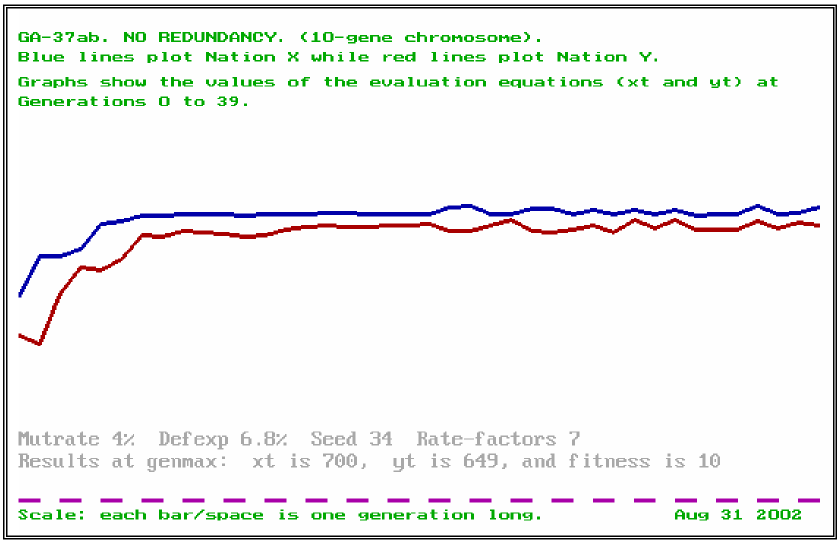
---

[7] In terms of element slots (24 + 3) x 0.63 ≈ 17.0. Add 4 to cater for columns 1 and 2 and, on allowing for rounding, we can expect crossover at any point in elements 20, 21 or 22



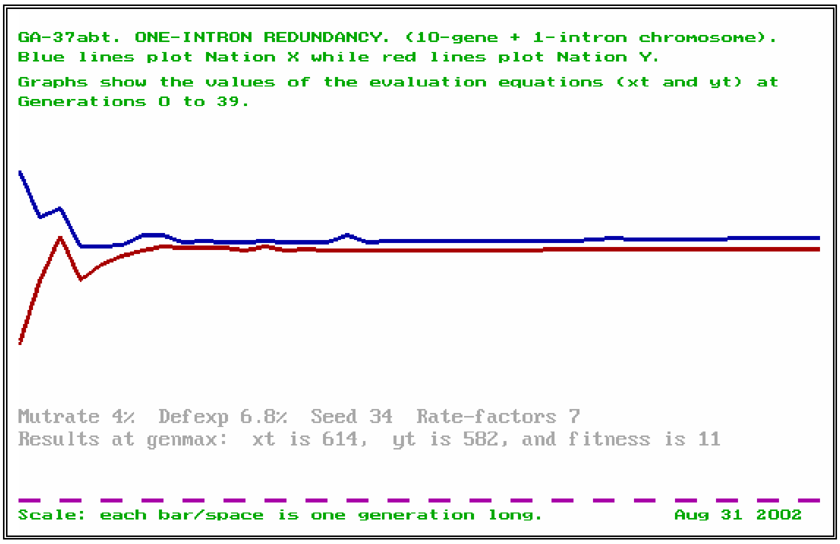
```
GA-37abt. ONE-INTRON REDUNDANCY. (10-gene + 1-intron chromosome).
Blue lines plot Nation X while red lines plot Nation Y.
Graphs show the values of the evaluation equations (xt and yt) at
Generations 0 to 39.

Mutrate 1.4%  Defexp 6.8%  Seed 34  Rate-factors 7
Results at genmax:  xt is 606,  yt is 579, and fitness is 11

Scale: each bar/space is one generation long.          Aug 31 2002
```

Figure 8.13 Output of GA-32. Seed 34, mutation rate 1.4%, and a defexp% of 6.8%



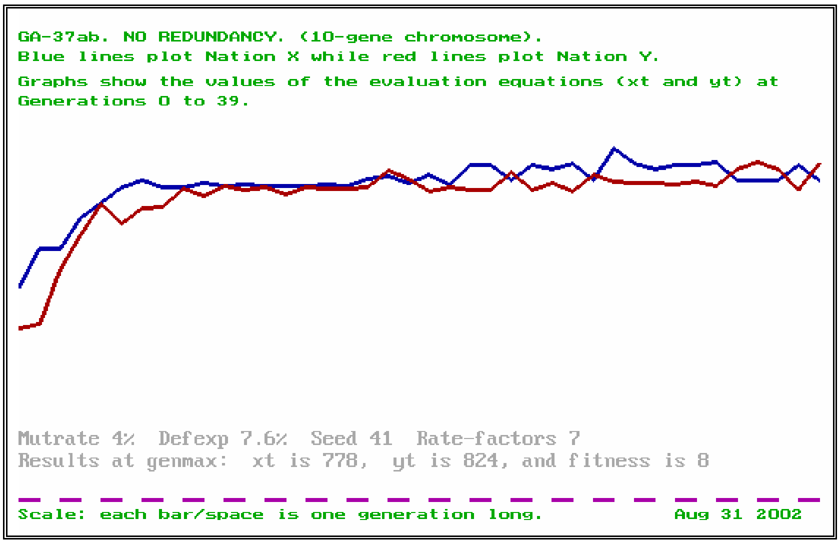
```
GA-37ab. NO REDUNDANCY. (10-gene chromosome).
Blue lines plot Nation X while red lines plot Nation Y.
Graphs show the values of the evaluation equations (xt and yt) at
Generations 0 to 39.

Mutrate 4%  Defexp 6.8%  Seed 34  Rate-factors 7
Results at genmax:  xt is 700,  yt is 649, and fitness is 10

Scale: each bar/space is one generation long.          Aug 31 2002
```

Figure 8.14 Output of GA-0. Seed 34, mutation rate 4%, and a defexp% of 6.8%



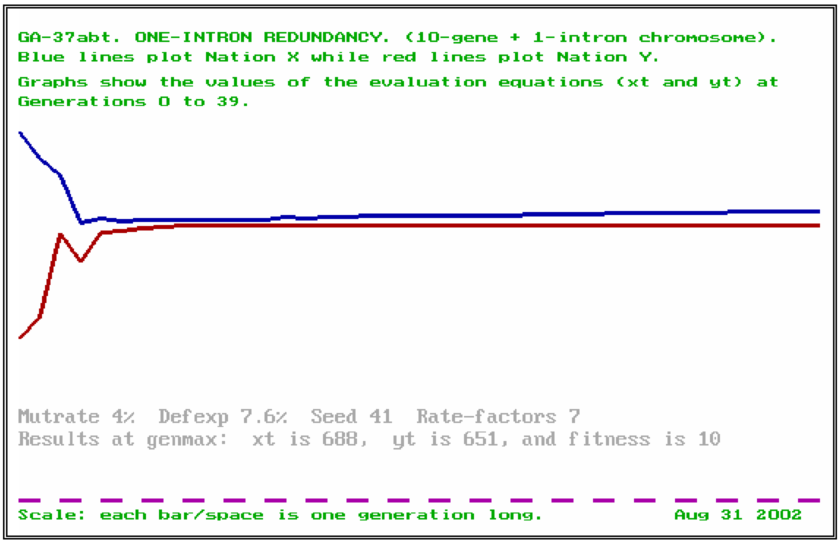
```
GA-37abt. ONE-INTRON REDUNDANCY. (10-gene + 1-intron chromosome).
Blue lines plot Nation X while red lines plot Nation Y.
Graphs show the values of the evaluation equations (xt and yt) at
Generations 0 to 39.

Mutrate 4%  Defexp 6.8%  Seed 34  Rate-factors 7
Results at genmax:  xt is 614,  yt is 582, and fitness is 11

Scale: each bar/space is one generation long.          Aug 31 2002
```

Figure 8.15 Output of GA-32. Seed 34, mutation rate 4%, and a defexp% of 6.8%



```
GA-37ab. NO REDUNDANCY. (10-gene chromosome).
Blue lines plot Nation X while red lines plot Nation Y.
Graphs show the values of the evaluation equations (xt and yt) at
Generations 0 to 39.

Mutrate 4%  Defexp 7.6%  Seed 41  Rate-factors 7
Results at genmax:  xt is 778,  yt is 824, and fitness is 8

Scale: each bar/space is one generation long.           Aug 31 2002
```

Figure 8.16 Output of GA-0. Seed 41, mutation rate 4%, and a defexp% of 7.6 %



```
GA-37abt. ONE-INTRON REDUNDANCY. (10-gene + 1-intron chromosome).
Blue lines plot Nation X while red lines plot Nation Y.
Graphs show the values of the evaluation equations (xt and yt) at
Generations 0 to 39.

Mutrate 4%  Defexp 7.6%  Seed 41  Rate-factors 7
Results at genmax:  xt is 688,  yt is 651, and fitness is 10

Scale: each bar/space is one generation long.           Aug 31 2002
```

Figure 8.17 Output of GA-32. Seed 41, mutation rate 4%, and a defexp% of 7.6 %

Only three sets of outputs are shown here, although many more with different input parameters and seeds were tried with analogous results.

Even from the small sample shown, it is evident that introns are having a smoothing effect on the behaviour of our GAs. Indeed, it looks as if convergence is being ‘encouraged’ by the removal of turbulence. In order to investigate why this should be, we converted GA-32 into yet another GA, GA-33 (using Schaffer *et al.*’s ffn) in order to make use of de Jong *et al.*’s [1997] curves to assess the epistasis level of GA-33, as we did in Section 8.7.

The result is shown in Figure 8.18 which, in order to facilitate comparison, is drawn to the same (offset zero) scale as GA-29 in Figure 8.9. These two figures merit direct comparison. The three curves of Figure 8.18 are further apart than those of Figure 8.9 and so, under de

Jong *et al.*'s [1997] criteria, the chromosomes of GA-33 are *less epistatic* than those of GA-0. Similar effects have been found with other seeds, such as 8, 9 and 10.

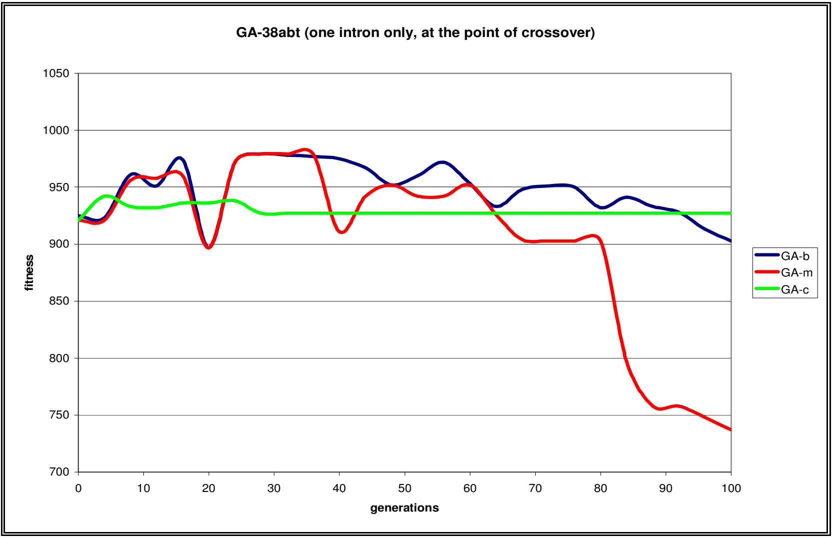
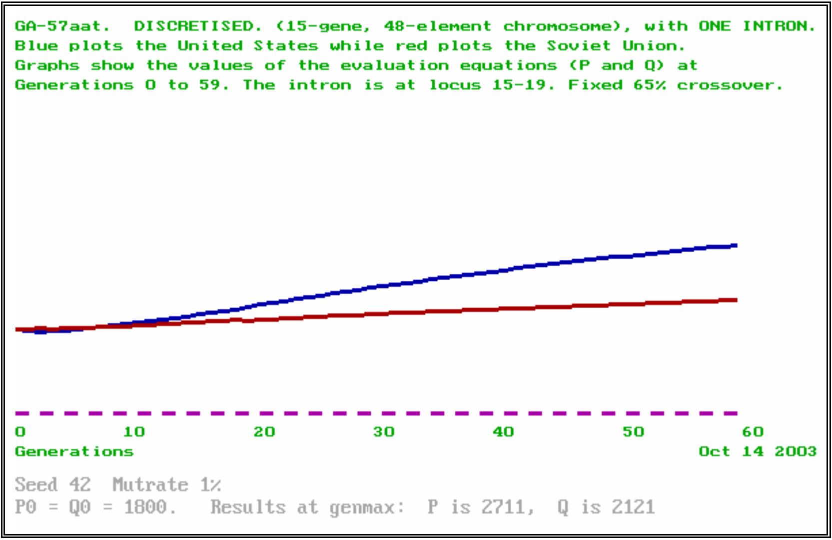


Figure 8.18 GA-33 (one intron only). Seed 7. Comparative fitnesses for GA-b, GA-m and GA-c. defexp% 3.3, initial mutation rate 1.4% minus 0.0001% per generation, fixed crossover rate 60%. Apart from the fixed crossover rate, the parameter settings are *identical* with those of GA-29 in Figure 8.9. Note the offset zero, for pictorial clarity. Figures 8.9 and 8.18 are to the same scale

Figure 8.19 shows a third example of this technique. One intron was placed in GA-3 (a nuclear deterrence GA) at the fixed point of crossover (elements 14-16). We call the resulting GA, GA-34. Figure 8.19 (P = 2711, Q = 2121) can be compared directly with Figure 7.9 (P = 2832, Q = 2184), and Q is under 2% away from the best results (P = 2728, Q = 2086, see Table 7.13). The smoothing effect is quite marked but, more to the point, when compared with the other GAs employed in Chapter 7, GA-34 is noticeably more robust.



```
GA-57aat.  DISCRETISED. (15-gene, 48-element chromosome), with ONE INTRON.
Blue plots the United States while red plots the Soviet Union.
Graphs show the values of the evaluation equations (P and Q) at
Generations 0 to 59. The intron is at locus 15-19. Fixed 65% crossover.
```

```
Seed 42  Mutrate 1%
P0 = Q0 = 1800.   Results at genmax:  P is 2711,  Q is 2121
```

Figure 8.19 Output of GA-34 (nuclear) corresponding to Figure 7.9. Seed 42. 60 generations. One intron only. γ and $\hat{\gamma}$ have empirical values 0.9851 and 0.9952, respectively, see Eqs(7.17), and, as before, are heavily coupled

We are a long way from actually *proving* anything but it does seem at first sight that the introduction of just one intron, judiciously placed at the point of crossover, has a beneficial effect on the GA's behaviour, reducing its epistasis level, smoothing its output, and making it more robust.

8.10 Does GA-32 Evolve?

It has been suggested [Ofria *et al.,* 2002] that a GA can only be said to *evolve* if it is robust to the point where we get the same values whatever the number of generations. It would seem that GA-32 comes very close to meeting this criterion. If run with IMF rate-factors (38, 48, 3, 2, respectively), a mutation rate of 4% , a defexp% of 6.8%, then for a representative selection of seeds, in particular 0, 3, 5, 9, 13, 24, 34, 41, 55 and 76 we get identical results for x_t (= 614), y_t (= 582) and *fitness* (= 11), respectively, see Figure 8.15 for one example. Furthermore, the results are only marginally different if the mutation rate takes on values of anything between 3% and 5%, and the same is true of defexp% which can vary between 6.3% and 7.5%. Compared with the output results of other GAs in the thesis, GA-32 is extraordinarily robust and yet its only significant difference is the additional intron. Again, we are a long way from proving anything, but it does seem as if the additional intron helps the GA to evolve. Ofria *et al.* [2002] also said that GAs can be viewed as a tool to evolve specialised problem-solving code and, furthermore, that isolating those aspects of the GA's design that are directly responsible for evolvability is of fundamental importance if dedicated evolvable instruction-sets are to be designed. To this end, Ofria *et al.* have devised a means of quantifying *robustness*, namely the fraction f_v of single mutations of a chromosome that are neutral or beneficial, i.e.

$$f_v = \frac{N_v}{(D-1)\ell} \quad ,$$

where N_v is the number of neutral or beneficial mutants, D is the size of the instruction set and ℓ is the chromosome length. According to Ofria *et al.*'s reckoning, GA robustness yields a low figure for f_v , typically lying between 0.02 and 0.14. They used these figures to show that where fitness continues to increase during the evolutionary process (which we would regard as normal) the robustness (on average) stays constant, suggesting that the "adaptive process has led the population to a *comfortable* level that avoids evolutionary dead-ends". This echoes Levenick's concept of *diverse convergence* [1999], see Section 2.13. Our only argument with Ofria *et al.*'s approach is that it seems to give too big a weight to disruption caused by mutation. Our empirical experience, in contrast, is that the selection process causes much more disruption than the comparatively smooth mutation process which we use. Nevertheless, it does look, on balance, as if GA-32 can be said to *evolve.*

8.11 Conclusions

It has been observed experimentally that a redundant GA behaves rather better than a non-redundant one. Only one class of GA application has been examined, so this was far from being an exhaustive study, even though the use of problem generators could possibly have improved the generality of the observation. Nevertheless, we set out to look at redundancy in chromosomes, specifically in order to see if the use of introns could enhance the performance of our arms race model and, if so, how.

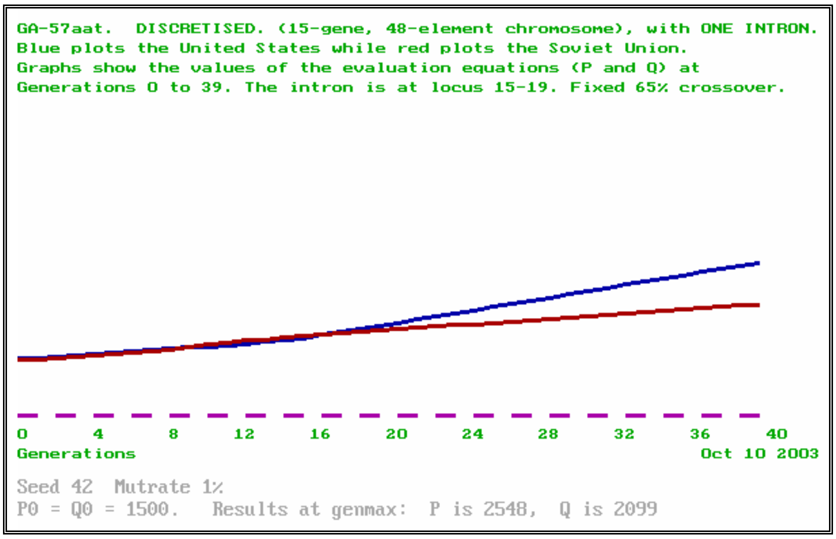
In one sense, our introns were unique; other researchers used introns which could be *activated* or *expressed* as and when required (see Section 8.8) and hence were effectively alternatives for some of the exons in use. In contrast, our introns remained dormant (*unexpressed*) throughout the computation of the GA under consideration, and this severely limited their usefulness in terms of their effects on the parameter space which they could search. Even so, it was demonstrated that increasing redundancy slowed convergence, and it was hoped that as a result the chromosomes of the GA under consideration would remain genetically active for longer, perhaps even attaining the long-hoped-for state of *diverse convergence* (see (2) of Section 8.2.4). This was not to be.

It was stated in (4) of Section 8.2.3 that Levenick [1991] had shown that introns reduce schema loss due to crossover. Introns also reduce the effects of epistasis, and so make the GAs more robust.

Herein our solution is to employ a fixed crossover rate, insert just one intron, and ensure that (the single) crossover takes place at the intron's locus. An intron in the right place at the right time can prevent the destruction of good building blocks by allowing *intact* active genes to merge together after crossover. In Figures 8.12 to 8.17, and 8.19, we have demonstrated experimentally that the insertion of this one intron:

- smoothes the output results of the evaluation equations, i.e. makes the curves less turbulent. This indicates that the disruptive effects of crossover have been reduced;
- reduces epistasis, and this in turn makes the GAs more robust. Increased robustness makes any stochastic process more useful and more cost-effective.

We believe that the above is no mean achievement, and is a major contribution to the effectiveness of our arms race model.



GA-57aat. DISCRETISED. (15-gene, 48-element chromosome), with ONE INTRON.
Blue plots the United States while red plots the Soviet Union.
Graphs show the values of the evaluation equations (P and Q) at
Generations 0 to 39. The intron is at locus 15-19. Fixed 65% crossover.

Seed 42 Mutrate 1%
P0 = Q0 = 1500. Results at genmax: P is 2548, Q is 2099

CHAPTER 9

Conclusions

9.1 General

The thesis has entailed the development of sixty-eight algorithms covering different aspects of the problem over nearly six years and, since GAs are stochastic processes, probably upwards of two hundred thousand runs. Cohesion is therefore a major requirement for these conclusions and this made the choice of section headings for this chapter difficult. We settled on:

- The viability of Richardson's arms race equations
- Using GAs; lessons learnt
- Redundancy in chromosomes; not so useless after all
- Are our results credible?
- Opportunities for further research
- The use of GAs in political applications
- Epilogue

9.2 The Viability of Richardson's Arms Race Equations

The value of Richardson's work lies not so much in the particular formulation of his theory, but in that he was the first to show that the causes of war could be subjected to mathematical thought and treatment, and he did this despite serious limitations in both his data and the computational techniques open to him in the late 1940s.

It is not always easy to apply Richardson's arms race equations to real-world situations. To be of practical use the equations need to be replaced with (or transformed into) something more precise and expressed in terms of the information available. It is all very well to be able to prove theoretically from Eqs(3.1) that instability occurs when $ac > bf$, and from Eq(3.28) that the sign of the difference $a - c$ indicates the direction of the curvature of the limit cycle which is crucial in predicting when instability is likely to occur. However, to do this we must be able to relate the Richardson coefficients a, b, c and f to the actual discretised evaluation equations used in the GA, and, in particular, to the rate-factors, $k_{i,j}$. In our case we derive a first-order approximation given by Eqs(3.16). Nevertheless, finding a relationship in general terms is part of the skill of using Richardson's equations. To be valid the evaluation equations must always reflect the spirit and intent of Richardson's theory. This is why initially, in Chapters 5 and 6, we did not use conventional curve-fitting techniques on the IMF defence expenditure figures, but instead we used Richardson-like equations (cf. Eqs(5.7)) whose output could be adjusted to *mimic* the IMF patterns. That way, we achieved what has been described [Cao *et al.,* 2000] as *a kinetic model faithful to the input data and the physical principles involved.* As Cao *et al.* [2000] argue, such models can legitimately be extrapolated. Much of the thesis is concerned with validating Richardson's model, and this entails 'predicting' events which have already happened. Our ultimate aim, however, is to use the model to predict conflicts happening in the future, so the ability to extrapolate is paramount.

9.3 Using GAs: Lessons Learnt

9.3.1 General

(1) With two major exceptions, the GAs developed for use in the thesis were fairly standard. Both crossover and mutation were varied dynamically. In our early work, received wisdom dictated that crossover started high but reduced, while mutation started low but increased (albeit neither by much). Later, however, we sometimes found by experiment that increasing both made the GAs more robust. Nevertheless, it remains true generally that crossover is a good way to put the early (diverse) pieces of good solutions together, but once convergence starts then the search for better solutions is led by mutation.

(2) The two exceptions were that:

- o We used a direct mapping between phenotype and genotype, i.e. there was no embryogeny at all. That this could be done was due to the arms race phenotype figures lying typically in the hundreds and thousands, numerals that could easily be encoded directly.
- o Some of the genes were integers, some were (rounded) real numbers, and of both groups some had two digits and others three (see Table 0.1 in the Glossary). When concatenated in a chromosome, however, it all *looked like* one big integer! We call this *integer-encoding*. The downside of this is that at crossover, some chromosomes get split mid-gene. Information Theory [Shannon, 1948] says that loose 'bin-ends' introduce noise and, in our experience, to some extent this is true. Crossover *inside* an exon introduces additional random 'mutations' and is likely to create two offspring significantly different from (and possibly not even

resembling) their parents. To overcome this, we settled on a fixed crossover rate and introduced just one intron (Section 8.9) at that fixed point. The only gene to be butchered, therefore, is an intron. This is far less disruptive on building blocks than slicing an exon.

(3) GAs must never be *forced*, but they can be *coaxed*. A good way to coax is to constrain the initialising random number generator to a likely value for each parameter. In other words, the search space should always be both controlled and feasible.

(4) Different populations were tried, ranging from 20 to 1000. For most of the experimental work, we settled on 100 which was sufficient for this application.

(5) Different forms of crossover were tried, including *uniform crossover* [Syswerda, 1989] but, on balance, single crossover appeared to be best for the arms race application.

(6) It proved very difficult, despite resetting the random number generator at the start of each loop, to ensure that our multiple-iteration (looped) GAs (such as GA-18 and others used for a Pareto-sort, see Table 7.5) gave results which were *identical* with those of single-iteration GAs.

(7) In (10) of Section 4.2 we made an early comment that "*The value of a practical GA is judged by its robustness. It is generally worth going to some trouble to ensure that results are not overly influenced by the starting conditions"*. This led us in turn to consider epistasis and its effects, delaying convergence, introducing redundant genes, and introducing *elitism.* Furthermore, in order to enhance robustness, we discretised the evaluation equations when (as in Chapter 7) this was not already done, so that the computed results for P and Q became an integral part of the chromosome. We did all of these, separately and in various combinations, with varying levels of success. In particular, the use of a chromosome with some redundancy not only appeared to make the locations of genes in the chromosome much less critical, but it also seemed to have a smoothing effect on the results. Even better was a GA using chromosomes with just one intron placed judiciously at the (fixed) point of crossover. This proved to render the GA very robust and, indeed, *cost-effective* inasmuch that it was no longer necessary to have quite so many runs to obtain a solution. With hindsight and in the light of our experience we should, in the italicised quotation above, have said "*It is always worth going to a lot of trouble to ensure that results are not overly influenced by the starting conditions"*.

(8) We have often been asked “Why use a GA at all?” In the thesis, specifically in Sections 5.10, 6.8 and 7.18, we give some detailed answers. In more general terms, developing a GA from scratch can be tedious, but once a GA has been developed it is both adaptable and flexible. It examines a big search space and so is not misled by local maxima or minima or (reasonable levels of) added noise. It is also self-correcting.[1] The last three sentences, however, make it look as if a GA was merely another optimisation technique. In contrast, Vose [1999] focuses on the *Simple GA* as an evolutionary system based on a broad class of *Random Heuristic Search* (RHS) algorithms and demonstrates that a GA is a special case of an RHS. There may well be better methods of solving specific problems, but on balance we believe that for the arms race application the GA approach is as adaptable, as universal, and as user-friendly as any other. Furthermore, GAs introduce mutation, and act implicitly on a whole population of individuals at once, an achievement which more conventional numerical methods cannot, in general, match.

9.3.2 Specific

(1) It is evident from the results that if the GAs embodied in our models of the India/Pakistan, Greece/Turkey and Middle East conflicts and of nuclear deterrence are going to converge, then they will do so quite quickly, typically after twelve to fifteen generations. By the standards of some GAs, twelve generations is tiny. Equally, the start-point, the *seed*, is crucial. Runs have been carried out for many different seeds, and they all indicate that in areas of convergence (in fractal basin language, in a ‘sea’ of blue, yellow or red) it does not matter very much where the GA started. In other words, the parameter space would seem to contain sub-spaces of near-perfect fitness. In areas bordering on fractal basin boundaries, however, the starting conditions appear to be critical.

(2) GAs in the ‘fractal basin’ series (e.g. GA-7 and GA-10) are searching for balance-of-power points. All of them seek perfect balance (i.e. zero fitness). High levels of fitness − a bad thing − occur near the fractal basin boundaries in unstable regions of the space. In these boundary areas the perfect balance of power, even when found, is (at least as far as our experiments can reveal) sensitive to small-scale perturbations. Fitness is a link between genotype and phenotype, i.e. between a genetic representation of the problem and the problem itself [Michalewicz, 1992].

---

[1] We once mistakenly entered some rate-factors, $k_{1,1}$ and $k_{2,2}$, the wrong way round; forty-five generations later the mistake had corrected itself

(3) The balance-of-power points would appear to lie in hyperplanes through parameter space. According to the s*chema theorem* [Goldberg, 1989] GAs search for hyperplanes with good fitness. From our results, it looks as if the parameter space does actually contain sub-spaces of perfect fitness. If this is true, then a GA is ideal for solving high-dimensional non-linear systems such as the 2- or 3-nation Richardson model.

9.4 Are Our Results Credible?

Chapter 5 set out to investigate a short window (1993 – 1999) in the arms race between India and Pakistan by making use of a GA to search the large space needed by Richardson's arms race theory. It used existing IMF data to generate *rate-factors* and scaling factors such that it became possible to formulate two evaluation equations usable in a conventional GA, and to test that the results of the GA did vindicate Richardson's theoretically-derived criteria for stability. For India/Pakistan it found that, far from the expected 'seas' of stability and instability separated by clearly marked 'walls' found in other applications (such as the Middle East scenario) there were large areas of stability separated by quite local but violent pockets of instability. It would be bold to say at this stage that known and likely instabilities between India and Pakistan are *always* prefaced by a sudden change in the direction of curvature of the limit cycle, but this does seem to be happening in this case. This change, the result of applying Peng *et al.*'s theory of canard explosions to Richardson's arms race theory, is expressed by the sign of $a - c$. If the sign of $a - c$ of two-thirds of the GA's population changes from minus to plus in successive iteration steps of average defexp%, *then it seems that an instability will occur at the next timeframe.* Such changes of sign are evident in Figure 5.6 (at average percentage defence expenditures of 6.5% and 7.9%) which relate directly to known historical events in the spring and autumn of 1997 (when in March there was a deliberate rocket attack by India on a mosque in Pakistan, followed in September by large-scale artillery exchanges between the two countries which very nearly led to a full-scale war (see Section 5.5.1)).

Chapter 5 concluded by using the data assembled for that chapter in 1999 as a "training set" to see if anything could be predicted for 2000. A change in the direction of curvature of the limit cycle at an average percentage defence expenditure of 10.1% was forecast for September 2000. (In the event, the situation that September deteriorated to the point where a single inflammatory speech by a Kashmiri politician very nearly triggered a war.)

The almost continuous conflict in the Middle East from 1955 to 2000, the India/Pakistan wars of 1965, 1971 and 1998, and the armed dispute between Greece and Turkey over Cyprus in 1974 and over the Aegean in 1997, all have very different scenarios. In all three conflicts, however, Chapter 6 demonstrates that the use of Richardson's equations in a GA has meant that the timing of the various outbreaks of hostilities *could have been predicted with some accuracy.*

So much for predicting wars with hindsight, which is what we have largely done. Can we use these methods for *genuine* prediction, e.g. are there going to be wars in the Middle East, or India/Pakistan in, say, 2003 or 2004? We believe that we can, provided and only provided that accurate, up-to-date data can be obtained so that meaningful extrapolation is possible. In the preface to Appendix 6A we present some of the problems encountered in obtaining really up-to-date figures, and in Section 6.7 we indicate the penalties of using noisy information. In the meantime, we will content ourselves with going halfway by stating with some confidence (Section 6.6) that there will be conflicts when average percentage defence expenditures reach 10.1%, 10.4% and 10.6% between Greece and Turkey, 12.9%, 13.1% and 14.0% between India and Pakistan, and 13.0%, 13.8%, 14.8% and 15.1% between Israel, Egypt and Syria. In making this forecast we are, of course, opting out of predicting future long-term movements in average percentage defence expenditures. Nevertheless, an extrapolation of the average defexp% for India and Pakistan for late 2001 (for which IMF and UN figures have only just been published) did suggest that the average defexp% might reach 12.9% in very late 2001. This date tallies with an attack on the Indian Parliament by Kashmiri suicide bombers in December 2001 which, in turn, led to sizeable incursions by both sides into the other's territories. One such incursion, by India, was successfully ambushed by the Pakistanis and a pitched battle resulted, with much loss of life.

Chapter 6 demonstrated that statistical figures taken from reference works published only once a year result in reasonable predictions. Results might be more spectacular if (as has just been said) accurate figures (e.g. from governmental military and economic intelligence sources) could be fed in on, say, a weekly basis but this requires resources well beyond our capabilities.

To quote Section 3.7, *"The evidence presented ... would suggest, therefore, that an unstable point in Richardson's equations is always prefaced by a change in the direction of curvature of the limit cycle, and that this is manifested by a sudden and marked change in* *the sign of $a - c$, from minus to plus, across the population. The experimental evidence for saying this may be slender, but the theoretical argument is quite strong".* While far from proving anything, Chapter 6 has shown that the experimental evidence is not so slender after all, sufficient in fact for us to be able to claim *proof of concept.*

Chapter 7 set out to find if an EA could find an acceptable solution to the problem of nuclear deterrence and hence increase our chances of survival. In 1972 it was thought that the Americans could have withstood an onslaught of 2354 Soviet nuclear warheads. We derived this figure, with difficulty, using an iterative numerical method (see Eq(7.12)). In contrast, using Richardson's equations and all the data about numbers of weapons and delivery probabilities available to us, the counter-values for the United States and the Soviet Union, respectively, can be calculated directly (*without* using GAs, see Section 7.9) as $P = 3229$ and $Q = 2573$.

An ES was developed, but this suffered from all the problems associated with hill-climbing methods. A suite of inter-related GAs was then developed in succession. These made no assumptions about weapons numbers but let those numbers evolve. The first (GA-19) did not give good results but demonstrated that, depending on the starting conditions, many answers were possible but those of the highest fitness always came up with approximately the same values for P and Q. The second (GA-2) incorporated a number of refinements and produced $P = 3486$, $Q = 2413$. The third (GA-20) injected a 1% elitism. The fourth (GA-22) deliberately introduced 33% redundancy into its chromosomes. The fifth (GA-24) combined elitism and redundancy. The remainder of the suite (GA-18, GA-21, GA-23 and GA-25) repeated the four original algorithms (GA-2, GA-20, GA-22 and GA-24) over a wide range of starting conditions and then applied a Pareto-sort, so that dominated results could be eliminated. The collective results are given in Table 7.13.

It seems that although the 1972 answer ($P = 3254$) was of the same order as those given in Table 7.12, a better answer ($P = 2728$, $Q = 2086$) has been found (much more simply) by subjecting the results of GA-25 to a Pareto-sort. In terms of 'best' numerical results, it makes little difference whether the GA is 'straight', elitist, redundant or both elitist and redundant, but these variations do have some effect on the GA's stability and robustness. Elitism, in turn, does not give dramatic effects but, again, helps to smoothe the output of the GA and make changes less turbulent. Discretising the evaluation equations has a pronounced smoothing effect, and results in a robust and reasonably accurate GA ($P = 2732$, $Q = 2189$) – see Figure 7.9.

Finally, in 2002 we re-visited our 1972 solution, first using MATLAB™ , and then variants of Jacobi and Gauss-Seidel. The numerical processing was much faster, but the basic equation has not changed. In effect, there is one equation with two unknowns. This stymies all conventional numerical methods, but it does not feeze a GA at all, largely because the GA looks at the whole search space rather than at a particular part of it.

9.5 Redundancy in Arms Race Chromosomes: Not so Useless after all

It has been observed experimentally that (in terms of robustness, stability and reduced epistasis) a redundant GA behaves rather better than a non-redundant one, so we set out deliberately to look at redundancy in arms race chromosomes. We tried originally to show that, in general:

(1) Increasing redundancy slowed convergence, but did so in such a way that the chromosomes remained genetically active. Indeed, introducing introns was one way to attain the long-hoped-for state of *diverse convergence* (see (2) in Section 8.2.4).

(2) Increasing redundancy reduced epistasis and this of itself made the GA more robust and easier to handle.

(3) Far from hindering genetic search, introns proved beneficial.

Experimental results for (2) and (3) above show that an intron in the right place at the right time can prevent the destruction of good building blocks by allowing *intact* active genes to merge together after crossover. In effect, we are arguing that the disruptive effect of crossover can be significantly reduced by ensuring that the only gene to be butchered is an intron. This last statement is of general application and can, in this instance, be most easily achieved by using a fixed crossover rate and *one* appropriately-located intron. We demonstrated this in GA-32 and GA-34. The smoothing effect of this combination can be seen in Figures 8.12 to 8.15 and in Figure 8.17.

It was stated in (4) of Section 8.2.3 that introns reduce schema loss due to crossover. Introns also reduce the effects of epistasis, see Figure 8.18, and so render GAs more robust. In particular, one judiciously placed intron has a beneficial effect on the GA's behaviour, reducing its epistasis level, smoothing its output, and making it more robust.

It seems that redundancy in chromosomes, while not living up to our original expectations, makes a significant contribution to our arms race application.

9.6 Opportunities for Further Research

We believe that we have shown that outbreaks of hostilities *can* be predicted by using a combination of GAs, Richardson's arms race theory and Peng *et al.*'s canard explosion theory. With the resources we can currently muster, however, we have taken this about as far as we can go. Genuine prediction (into the future from now onwards) needs a lot more accurate and timely information than is available to us, and this can only come from governmental military and economic intelligence services. Two areas merit further investigation:

- Examining more fully the implications and the effects of introducing just one intron at the point of crossover. From what we have done already, it appears that this has a smoothing effect on the output of the GA, that it reduces epistasis, and that it makes the algorithm more robust. Any improvement in robustness is of major practical significance; not so many runs have to be made, so the cost-effectiveness of the GA is improved.
- Examining more closely the similarities between GAs and Monte Carlo methods.

9.7 The Use of GAs in Political Applications

Some would argue that it is not valid to extrapolate from mechanistic cause and effect (for, example, the certainty that certain concentrations of ions will explode) to the vagaries of human interaction. Nations go to war, not because of their percentage defence expenditures but because their leaders believe that they should, and that they will be backed in that decision by their people. Equally, those same leaders like to believe that they have free will and are not pre-destined or pre-programmed automata. The truth may lie somewhere in between. It has been shown in the thesis that the situation between India and Pakistan contains large areas of stability and only pockets of instability. No matter how belligerent their leaders, no matter how much sabre-rattling is done, *it will not physically be possible to start a war when all other parameters are stable.* The same is true of other examples presented. The danger comes when the parameters are unstable, for at that point almost anything – such as one injudicious speech – could trigger a conflict. It follows that it *is* worth predicting the pockets of instability provided it is appreciated (and hoped) that a trigger may never materialise and the potentially unstable situation may quietly revert to stability.

At the risk of gross over-simplification, it does seem that most wars are about the ownership and use of land and other natural resources such as water and oil. We suspect that overt militarism generally plays a smaller part in modern warfare than is popularly supposed, and that a nation's desire to go to war is often uncomfortably democratic.

9.8 Epilogue: Déja-vu?

We close with four quotations:

- "*The ultimate goal of mathematics is to eliminate all need for intelligent thought*" [Graham, Knuth and Patashnik, 1989]. Outputs from the thesis are there to guide the world's policy-makers, not to absolve them of a need to think!
- *"The quality of a mathematical conclusion is determined by a lot more than just the accuracy of the calculations*" [Cohen and Stewart, 1994]. If this is true of the mathematical niceties which have been discussed in the thesis, it is even more true of social and/or political problems over which even wise men differ.
- "*The scientist describes what is; the engineer creates what never was*" [Colwell, 2002, paraphrasing von Karman]. The thesis has been written by a lifelong engineer.
- We leave the very last word to Richardson himself. He said: " *At the time of writing, July 1951, India and Pakistan are accusing each other of aggressive intentions, while the United States and the Soviet Union have been doing so for years. Personally, I think that much of what is blamed as aggressive intention* (*g* and *h*) *is really only defensiveness* (*a* or *c*)*"*.

Glossary and Abbreviations

It is appreciated that certain words, contractions, phrases and acronyms found in the thesis may not be familiar to all its readers. The explanations which follow are minimalist, and apply as used in the thesis; they are not necessarily exclusive, nor of general application. There are, for instance, definitions of ***robust*** *which differ from that given below.*

GENERAL

Defence Expenditures

- ***Defence Expenditure*** (*normally denoted by* x_t, y_t, *etc*)

  The amount of money *actually spent* by a nation in a financial year on defence, including its armed forces and their pay, pensions, land, accommodation and training facilities, ships, aircraft, munitions, tanks, vehicles, weapons, and all military stores. It may be more or less than the budgeted amount.

- ***Budgeted Defence Expenditure*** (*normally denoted by* x_m, y_m , *etc*)

  The amount of money *budgeted* by a nation each year for its defence, normally expressed in millions or billions of US dollars.

- ***Percentage Defence Expenditure*** (*defexp%*)

  That fraction of a nation's GDP *budgeted* for spending on arms. It is conventionally expressed as a percentage.

- ***Intrinsic Defence Expenditure*** (*normally denoted by* x_s, y_s, *etc*)

  The amount of money each nation spends on defence, irrespective of competitive spending by its neighbours. These are essentially the 'standing costs' of a nation's armed forces, and can be as high as 80% of the total Budgeted Defence Expenditure.

Gross Domestic Product **(GDP)** (*normally denoted by* G_X , G_Y , etc)

The overall 'worth' of a nation expressed in monetary terms.

International Monetary Fund **(IMF)**

The international body, based in New York, whose role is to maintain a fiscal balance throughout the world. Part of their function is to regulate the flow of money between one country and another, and to arrange loans usually to third-world nations. The IMF publishes very authoritative *IMF Statistics* annually.

In order to demonstrate some fundamental concepts frequently encountered in the GA technical literature pertaining to the thesis, we present below a table that shows the relationship between (and configuration of) the more commonly-used parameters.

Table 0.1 Configuration and Nomenclature of Chromosomes – GA-0[1]

across – column nos *down – row contents*	2	3	4	5	6	7	8	9	10	11
gene	$k_{1,1}$	$k_{2,2}$	$k_{1,2}$	$k_{2,1}$	x_s	y_s	x_m	y_m	x_t	y_t
number of elements	2	2	2	2	2	2	3	3	3	3
type of number	*int*	*int*	*int*	*int*	*int*	*int*	*real*	*real*	*real*	*real*
typical contents	17	09	25	16	36	29	561	703	458	753

The whole concatenated 10-gene, 24-element chromosome shown above is expressed and handled by the GA seamlessly as

170925163629561703458753

which, despite the inclusion of four real numbers, *looks and handles like* one big integer. Each element has an equal chance of being mutated, and crossover can take place anywhere, irrespective of gene boundaries.

Allele

The *allele* of a gene is its actual value [Bentley, 1999], e.g. the allele of gene x_m in the table above is 561.

Bottom-line consensus

In constructing a *fractal basin*, it is necessary to represent the coloured 20×100 matrix of a dominance diagram by one coloured square only. Which colour is it to be? We count the squares on the bottom-line (i.e. the latest generation) and take a 'majority vote'. The colour of the resulting square is that usually associated with the dominant nation.

Canonical

A canonical GA is a standard one and is generally accepted as being of the Holland-type, i.e. operating on a population of chromosomes with mutation and crossover operators, and subjecting the population to selection processes whereby fitter chromosomes reproduce at the expense of the others. 'Canonical' does not imply any particular representation; in particular it should *not* be exclusively associated with 'binary' representation.

---

[1] In order to avoid involving readers in the taxonomy and numbering of our algorithms, all algorithms are numbered consecutively. A cross-referenced catalogue of GAs is shown on page 14. The example shown here (GA-0) is Algorithm GA-37ab

Cao's Norm

We define the norm of a rectangular matrix *A*, where $A \in \Re^{mn}$, as

$$\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j}^{2}} \quad .$$

This is also known as the Schur or Fröbenius norm [Cao *et al.*, 2000]. The norm is used extensively in Chapter 6 to improve the output of modelling GAs.

Cardinality

The number of elements in a set.

Chromosome

An ordered set of genes [Chambers, 1995]. The basic collective building blocks of evolution. In our context, a chromosome is an individual population member of a GA. The size of a chromosome is the total number of elements in its genes.

Continuity

Our data must be consistent, i.e. a small percentage change to the basic data must produce approximately the same percentage change in the result of the GA. If this does not happen, then no search strategy will succeed [Bentley, 1999].

Convergence

The gradual realisation of some desired end-point. See Section 2.16.

Crossover

Changing elements of a chromosome for the same elements of its partner chromosome or mate thereby exchanging genetic material between two parents. Crossover puts high-fitness 'building blocks' together on the same string in order to create strings of increasingly higher fitness. The 'crossover point', typically fixed by the user at 65% of the number of elements in a chromosome, can vary dynamically between 50% and 80%. It does not matter statistically if the crossover point is measured from the right or the left of the chromosome, provided that it is done consistently [Davis,1991; Vose,1999].

Curvature

The amount or rate of deviation of a curve from a straight line. Specifically, the curvature *K* of a phase plane trajectory is defined as the rate of turn of the tangent with respect to the arc length *s* along the trajectory [Peng *et al.,* 1991 p 279]. See *Inflection Line*.

Dominance

It is sometimes of interest to know which nation in a dispute is dominant at any one time. The 'dominance diagram' does this. For *every one* of the 2000 sets of equations (100 population × 20 generations) the colour of a little square indicates which of Nations X or Y is dominant, for example. It is therefore easy to see where domination changes. See, for example, Figure 4.2.

Element

A single-digit number contributing towards the representation of a gene. For instance, the gene 467 has three elements 4, 6 and 7.

Elitism

Elitism allows the best chromosome(s) of one generation to be carried forward into the next provided that its/their fitness is better than that of the worst member(s) of the new generation.

Embryogeny

The process that maps the search space (containing genotypes) onto the solution space (containing phenotypes) [Bentley, 1999].

Epistasis

In GA literature, the degree of dependent parameter interaction is called *epistasis*, a biological term for gene interaction. Epistasis is about genes acting in combination to produce (or inhibit) solutions. In other words, epistasis measures the extent to which the contribution to fitness of one gene depends on the values (alleles) of the other genes [Bentley, 1999]. See Sections 2.12 and 8.6.

Fitness

A measure of the worth of an individual chromosome in a particular domain. In our case, fitness is conveniently scaled to be a positive integer.

Fitness Function (ffn) Four alternative definitions:

(1) "An ffn takes a single solution from an evaluation equation as a parameter and returns a number, the *fitness*, indicating how good the solution is, in terms of what is desired. By itself, the number returned by the ffn means nothing; only when we compare the values returned by all possible solutions can we select the better solutions" [Chambers, 1999].

(2) "An ffn provides a measure of how individuals have behaved in the problem domain. In the case of a minimisation problem, the fittest individuals will have the lowest numerical value" [Chipperfield, 1997].

(3) "Parent selection dynamics are based on an application-dependent metric known as fitness. A fitness is a figure of merit computed by an ffn using any domain knowledge which applies. The greater a chromosome's fitness, the higher chance it has of being selected for reproduction" [Buckles and Petry, 1992].

(4) "An ffn computes the *distance* between the real and the simulated curves of the phenomenon studied. An ffn is normally chosen for domain reasons rather than for mathematical reasons, i.e. it is a means of gauging how well the output(s) of the evaluation equation(s) of a particular chromosome fulfil(s) the intentions of the domain". [Sanchez *et al.*, 1997].[2]

Gene

A set of elements; a coded parameter representing some aspect of a solution.

Genetic Algorithm **(GA)** Three alternative definitions:

(1) "A GA is an adaptive algorithm for solving problems using computational models of natural evolutionary systems" [Davis, 1991].

(2) "A GA is a problem-solving method which uses genetic rules of reproduction, gene crossover and mutation as its model of problem-solving. Applying these rules to pseudo-organisms can pass beneficial and survival-enhancing traits to new generations. As a real organism's characteristics are stored in its DNA, genetic algorithms store the characteristics of artificial organisms in an electronic genotype which consists of a string of bits" [Chambers, 1995].

(3) "A GA is an iterative, global search procedure whose goal is the optimisation of the fitness function. The algorithm works in parallel on a population of candidate solutions (chromosomes) from the search space" [Dumitrescu, 2000].

Genotype

The abstract collection of genes possessed by an individual. The actual structure containing the genes is called the *chromosome.* Each gene has a value (*allele*) and a position (*locus*) in

---

[2] Note that this is the only definition of ffn which assigns a dimension to fitness

the genotype [Dasgupta and Michalewicz, 1997]. Genotypes consist of coded versions of the parameters which make up a phenotype [Bentley, 1999].

Hopf Bifurcation

An intermittent transition to a chaotic attractor in which a function changes from having a 'laminar' or stable periodic existence into an unstable existence unrelated to its previous state. Shortly after a Hopf bifurcation, the function's limit cycle will change the direction of its curvature [Ott, 1994] and, in our case, instability will result.

Inflection and Inflection Line

An *inflection* is a change of curvature from convex to concave at a particular point on a curve. Consider the smooth (continuously differentiable) parameterised system:

$$\frac{dx}{dt} = F(x, y, c) \quad \text{and} \quad \frac{dy}{dt} = G(x, y, c), \quad \text{where } c \geq 0\,. \qquad .$$

Consider c fixed. At all points (x,y) in the phase plane, the curvature $K_t(x,y)$ of the trajectory which passes through it can be computed. The *inflection line* associated with the above equations is the locus of points where $K_t = 0$, and typically consists of one *or more* smooth curves dividing the phase plane into regions. In regions where $K_t > 0$ we say that the trajectories are *concave*, while in regions where $K_t < 0$ they are *convex*. .

On setting $c = 0$ and eliminating time, we obtain the rational function $\frac{dy}{dx} = \frac{G(x,y)}{F(x,y)}$.

For curves in the (x, y)-plane of the form $y = y(x)$, the curvature of arc length s can be computed as

$$K = \frac{d\vartheta}{ds} = \frac{\left(1 + y'(x)^2\right)^{3/2}}{y''(x)}$$

using the positive root of the numerator.

Integer-coded GA

A GA whose genes are populated with integers from the set {0, 1, ... , 9}.

Intron

In biological terms, an *intron* is a non-functional sequence of DNA. In GA terms, an intron is an unexpressed genotypic region meaning **either** an inactive gene which is lying around waiting to be *expressed* or *activated* [Levenick, 1991] **or** a gene which is an integral part of its chromosome but is not used in any evaluation equation.

Linear Norm

Given an increasing (decreasing) sequence of fitness values we assign a stepwise function to those values in order to smoothe out large variations between pairs of values. For example, the actual fitness sequence 251 / 325 / 555 / 555 / 555 / 1115 would be mapped in our case into the sequence 475 / 480 / 485 / 490 / 495 / 500. These are called *normalised fitnesses* or *normfits*. (Without normalisation 1115, over twice as fit as its nearest rival, would dominate selection and be chosen as a parent for nearly all offspring).

Locus

The *locus* of a gene is its juxtaposition to its neighbours in the chromosome. In Table 0.1 for instance, the locus of x_m lies between y_s and y_m. If the physical position of a gene among its neighbours makes material difference to the solution, then the chromosome is said to be *epistatic*. *Locus* is also sometimes used to indicate the precise position of an element in a chromosome, e.g. " … crossover is to take place at locus 21".

Mutation

Mutation is achieved by changing certain elements of the chromosome randomly. In the thesis we use an inbuilt random number generator of known Gaussian distribution. During mutation, every element (*locus*) of every gene is sampled individually to see if it should be mutated or not, so mutation is termed a *mutation probability per locus*. Mutation may be fixed, or vary dynamically. A typical mutation rate is 1%, but it can be set up in any number of ways, e.g.

$$mutation\ rate = \frac{1}{number\ of\ genes} \quad \text{or} \quad mutation\ rate = \frac{1}{number\ of\ elements} \quad .$$

but both of these yield too high a rate for our purposes.

Parameter Space / Search Space

A parameter or search space (in an EA context) is one filled with all possible solutions, and a point in that space defines a solution. Each chromosome represents an individual, and is a 'candidate solution'.

Pareto-Sort

Pareto [1896] devised a means of ordering a scatter of points by arguing that some 'dominate' (and thus effectively duplicate) others. A solution is Pareto-optimal if it is not dominated by other solutions.

A vector **x** is partially less than **y**, symbolically $\mathbf{x} <_p \mathbf{y}$, when

$$(\mathbf{x} <_p \mathbf{y}) \Leftrightarrow (\forall_i)\ (\mathbf{x}_i \leq \mathbf{y}_i) \wedge (\exists_i)\ (\mathbf{x}_i < \mathbf{y}_i) \quad .$$

Any **y** so dominated can be discarded.

Phenotype

"A phenotype is a collection of live parameters of real-world problems expressed in whatever manner is conventional and appropriate" [Bentley, 1999].

"In GAs the phenotype means the defining characteristics or qualities of the entire genotype" [Dasgupta and Michalewicz, 1997].

Random Start

All our GAs are initialised using an RNG. The RNG is set to generate likely and reasonable upper bounds for the genes in terms of the arms race domain. This is what is meant by a *random start*. If it is desired to use real-world data, then such data is made to overwrite the randomly initiated values.

Reflexive

An expression is said to be reflexive if the *effects* on one party are generated solely by *causes* created by the other party, e.g. the evaluation equation which lists the number of American missiles expected to survive a Soviet first strike is defined solely in terms of Soviet destructive capability, see Section 7.8.1 and Eq(7.10).

Region of Stability

If P and Q plot the American and Soviet counter-values, respectively, then 'occupation' of the area outside *both* contours will discourage either side from making war, see Figure 7.1.

Robust

We call a GA *robust* if the results are not unduly influenced by the starting conditions, i.e. in our case by the seed used to initialise the random number generator.

Schema

A binary *schema* is a string comprising zeroes, ones and asterisks; the last stand for 'don't-cares'. For example, schema 1 * * * * 1 is a 6-bit string which begins and ends with a 1.

Stochastic Process

A sequence of states whose evolution is determined by random events. In a computer these are generated by random numbers.

Trajectory

The path of any body moving under the action of given forces; a curve or surface passing through a given set of points, or intersecting each of a given set of curves or surfaces according to a given law.

Undominated

A result of a Pareto-sort is *undominated* if it has not been *dominated* by other results.

MILITARY AND NUCLEAR

ABM

Anti-ballistic missile, a defensive weapon that will *counter* incoming missiles in flight by generating a very steep-sided shock wave (an e*lectromagnetic pulse*) designed to destroy the semiconductor circuits of incoming missiles.

FGA	Fighter/Ground Attack [aircraft]
ICBM	Inter-Continental Ballistic Missile
SLBM	Submarine-Launched Ballistic Missile

Counter-value

The number of nuclear missiles held by Side A which will guarantee levels of unacceptable damage to Side B *after* they have survived an initial attack by Side B.

Destructive Capability

The value of a weapon in terms of the damage it can inflict on an enemy. This depends, not only on the weapon's size, but also on the probability of its accurate and timely delivery.

MIRV **(*Multiple Independent Re-entry Vehicle*)** See *Warhead.*

Warhead (sometimes ***Nuclear Warhead***)

A nuclear explosive device. One or more may be built into a missile or carried by an aircraft. Some missiles are *Multiple Independent Re-entry Vehicles* (MIRV), meaning that one missile may, on launching, house several warheads which, after separating in mid-air, are then capable of navigating themselves towards, and attacking, targets which are geographically separated.

Yield

The size of a nuclear weapon. It is normally expressed in Kilotons (KT) or Megatons (MT) of conventional explosives.

References

Ackley, DH 1987 *A Connectionist Machine for Genetic Hill Climbing.* Kluwer, Boston, MA

Adewuya, AA 1996 *New Methods in Genetic Search with Real-valued Chromosomes.* Master's Dissertation. Massachusetts Institute of Technology, Cambridge, MA

Aizawa, AW and Wah, BW 1993 Dynamic control of genetic algorithms in a noisy environment. *Proceedings of the 5th International Conference on Genetic Algorithms,* Urbana-Champaign, IL, pp 44-55

Aldawoodi, N and Perez, R 2003 Formula prediction using genetic algorithms. *Proceedings of the 2003 Genetic and Evolutionary Computation Conference,* Chicago, IL , **3**, pp 1-6

Andreou, AS and Zambanakis, GA 2001 A neural network measurement of relative military security in Greece and Cyprus. *Defence and Peace Economics*, **12** (4), pp 303-324

Axelrod, R 1997 *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration.* Princeton University Press, New Jersey

Bäck, T and Schwefel, H-P 1993 An overview of evolutionary algorithms for parameter optimisation. *Evolutionary Computation,* **1**, pp 1-23

Baker, JE 1985 Adaptive selection methods for genetic algorithms. *Proceedings of an International Conference on Genetic Algorithms and their Applications.*, July 1985, San Mateo CA, pp 101-111

Bedau, MA and Packard, NH 1992 Measurement of evolutionary activity, teleology, and life. In *Artificial Life 10.* Langton, CG (Editor). Addison-Wesley, Redwood, CA, pp 431-462

Benoit, E Callot, JL Diener, F and Diener, M 1981 Some aspects of the van der Pol equation. *Collectonea Mathematica,* **32**, pp 37-119

Bentley, PJ and Wakefield, JP 1997 Finding acceptable solutions in the Pareto-optimal range using multi-objective genetic algorithms. In *Soft Computing in Engineering Design and Manufacturing.* Chawdhry, PK Roy, R and Pant, RK (Editors). Springer-Verlag, London, pp 231-240

Bentley, PJ 1999 *Evolutionary Design by Computers.* Morgan Kaufmann, San Francisco, CA

Berman, D 2000 George Berkeley. In *The Great Philosophers* (*Essays on Socrates, Plato, Descartes, Berkeley, Hume, Russell, Heidegger, Wittgenstein, Popper and Turing*). Monk, R and Raphael, F (Editors). Phoenix Press, London

Bertona, A and Dorigo, M 1993 Implicit parallelism in genetic algorithms *Artificial Intelligence,* **61**(2), pp 307-314

*Bhattacharyya, S Pictet, OV and Zumbach, G 2002. Knowledge-based genetic discovery in foreign exchange markets. *IEEE Transactions on Evolutionary Computation***,** **6**, pp 169-181

Biethahn, J and Volker-Nissen, P, Eds. 1996 *Evolutionary Algorithms in Management Applications.* Springer-Verlag, Heidelberg

Brøns, M and Bar-Eli, K 1994 Asymptotic analysis of canards in the EOE equations and the role of the inflection line. *Proceedings of the Royal Society of London, Series A,* **445**, pp 305-322

Buckingham, RA 1962 *Numerical Methods.* Pitman Press, Bath

Buckles, BP and Petry, FE 1992 *Genetic Algorithms*. IEEE Computer Society Press, New York

Cantú-Paz, E 2000 *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer, Boston, MA

Cao, H Kang, L Chen, Y and Yu, J 2000 Evolutionary modelling of systems of ordinary differential equations with genetic programming. *Genetic Programming and Evolvable Machines,* **1**, pp 309-338

Chambers, LD 1995 *Practical Handbook of Genetic Algorithms,* **1**. CRC Press, Boca Raton, FLA

Chambers, LD 1999 *Practical Handbook of Genetic Algorithms,* **3**. CRC Press, Boca Raton, FLA

Chipperfield, A 1997 Introduction to genetic algorithms. In *Genetic Algorithms in Engineering Systems*. Zazala, AMS and Fleming, PJ (Editors). Peter Peregrinus, London, pp 5-45

Cobb, HG 1991 An investigation into the use of hyper-mutation as an adaptive operator in genetic algorithms having continuous, time-dependent, non-stationary environments. NRL Memorandum Report 6790 AIC-90-001. Naval Research Laboratory, Washington, DC

Cohen, J and Stewart, I 1994 *The Collapse of Chaos*. Penguin Books, London

Coley, DA 1999 *An Introduction to Genetic Algorithms for Scientists and Engineers.* World Scientific Publishing Co, Singapore

Colwell, R 2002 Engineering, science and quantum mechanics. *Computer,* **35**(3), pp 8-10

Dasgupta, D and McGregor, DR 1994 A more biologically-motivated genetic algorithm**:** the model and some results. *Cybernetics and Systems,* **25**, pp 447-469

Dasgupta, D and Michalewicz, Z 1997 Evolutionary algorithms **:** an overview. In *Evolutionary Algorithms in Engineering.* Dasgupta, D and Michalewicz, Z (Editors). Springer-Verlag, New York, pp 3-28

Davidor,Y and Schwefel, H-P 1992 An introduction to adaptive optimisation algorithms based on principles of natural evolution. In *Dynamic, Genetic and Chaotic Programming.* Souçek, B and the IRIS Group (Editors). John Wiley and Sons, New York, pp 183-202

Davis, L , Ed. 1991 *Handbook of Genetic Algorithms.* Van Nostrand Rheinhold, New York

Davis, L and Steenstrup, M 1987 *Genetic Algorithms and Simulated Annealing*. Morgan Kaufmann, Los Altos, CA

Dawkins, R 1983 Universal Darwinism. In *Evolution from Molecules to Men.* Bendall, D (Editor). Cambridge University Press, UK, pp 247-256

Deboeck, G 1994 *Trading on the Edge: Neural, Genetic and Fuzzy Systems for Chaotic Financial Markets*. John Wiley and Sons, London

de Duve, C 1984 A guided tour of the living cell. *Scientific American Library,* **2**, pp 257-265

de Jong, KA Potter, MA and Spears WM 1997 Using problem generators to explore the effects of epistasis. *Proceedings of the 7th International Conference on Genetic Algorithms,* East Lansing, IL, pp 338-345

de Jong, KA 1999 Evolutionary computation **:** recent developments and open issues. In *Evoluttionary Algorithms in Engineering and Computer Science.* Miettinen, K Mäkelä, MM Neittaanmäki, P and Périaux, J (Editors). John Wiley and Sons, Chichester, pp 43-54

Dittrich, P Skusa, A Kantschik, W and Banzhaf, W 1999 Dynamical properties of the fitness landscape of a GP-controlled random morphology robot. *Proceedings of the 1999 Genetic and Evolutionary Computing Conference,* Orlando, FLA, pp 1002-1008

Drechsler, R and Drechsler, N, Eds. 2003 *Evolutionary Algorithms for Embedded System Design*. Kluwer Academic Publishers, Boston, MA

Dumitrescu, D Lazzerini, B Jain, LC and Dumitrescu, A 2000 *Evolutionary Computation.* CRC Press International Series, Boca Raton, FLA

Epperson, JF 2002 *An Introduction to Numerical Methods and Analysis*. John Wiley and Sons Inc., New York

Eshelman, LJ and Shaffer, DJ 1993 Real-coded genetic algorithms and interval-schemata. In *Foundations of GAs,* **2**. Whitley, DL (Editor). Morgan Kaufmann, San Mateo, CA, pp 187-202

Europa World Yearbooks 1999/2000/2001/2002 *Europa World Yearbook*. Maher, J (Editor) Europa Publications, London

Falkenauer, E 1996 Tapping the full power of genetic algorithms through suitable representation and local optimisation. In *Evolutionary Algorithms in Management Applications.* Biethahn, J and Volker-Nissen, P (Editors). Springer-Verlag, Heidelberg, pp 167-182

Fogel, DB 1995 *Evolutionary Computation : Toward a New Philosophy of Machine Intelligence.* IEEE Computer Society Press, Piscataway, NJ

Fogel, DB 1999 Some recent important foundational results in evolutionary computation. In *Evolutionary Algorithms in Engineering and Computer Science.* Miettinen, K Mäkelä, MM Neittaanmäki, P and Périaux, J (Editors). John Wiley and Sons, Chichester, pp 55-72

Fogel, DB and Fogel, LJ 1996 Using evolutionary programming to schedule tasks on a suite of heterogeneous computers. *Computing and Operational Research,* **23**, pp 527-534

Fogel, DB and Ghozeil, A 1997 Schema processing under proportional selection in the presence of random effects. *IEEE Transactions on Evolutionary Computation,* **1**, pp 290-293

Fogel, LJ 1997 Evolutionary programming in perspective: a top-down view. In *Evolutionary Algorithms in Engineering Applications - an Overview.* Dasgupta, D and Michalewicz, Z (Editors). Springer-Verlag, New York, pp 135-146

Forrest, S and Mayer-Kress, G 1991 GAs, nonlinear dynamical systems and models of international security. In *Handbook of Genetic Algorithms.* Davis, L (Editor). Van Nostrand Rheinhold, New York, pp 166-187

Gariaev, PP Kaempf, U Marcer, PJ and Tertishny, GG 2000 *The DNA Wave Bio-computer.* Institute of Control Sciences, Russian Academy of Sciences, Moscow

Gilbert, Sir Martin 1999 *Challenge to Civilisation: A History of the 20th Century*, **3**. Harper Collins, London

Goldberg, DE Deb, K and Korb, B 1991. Don't worry, be messy! *Proceedings of the 4th International Conference on Genetic Algorithms,* San Diego, CA, pp 24-30

Goldberg, DE 1989 *Genetic Algorithms in Search, Optimisation, and Machine Learning.* Addison-Wesley, Reading, CT

Goldberg, DE Korb, B and Deb, K 1989 Messy genetic algorithms **:** motivation, analysis and first results. *Complex Systems,* **3**, pp 493-530

Goldberg, DE and Smith, DR 1987 Non-stationary function optimisation with dominances and diploidy. *Proceedings of the 2nd International Conference on Genetic Algorithms,* Cambridge, MA, pp 58-69

Graham, RL Knuth DE and Patashnik, O 1989 *Concrete Mathematics*, Addison Wesley, Reading CT

Grefenstette, J Ramsey, C and Schultz, A 1990 Learning sequential decision rules using simulation models and competition. *Machine Learning,* **5**, pp 355-381

Grefenstette, J 1992 Genetic algorithms for changing environments. *Proceedings, Parallel Problem Solving from Nature*, Amsterdam, pp 137-149

Haataja, J 1999 Using genetic algorithms for optimisation : technology transfer in action. In *Evolutionary Algorithms in Engineering and Computer Science.* Miettinen, K Mäkelä, MM Neittaanmäki, P and Périaux, J (Editors). John Wiley and Sons, Chichester, pp 3-22

Hackworth, TW 1973 *The Relevance of Mathematical Logic to the Strategic Arms Limitation Talks.* National Defence College, Latimer, UK

Hackworth, TW 1999 (a) Genetic algorithms **:** some effects of redundancy in chromosomes. *Proceedings of the 1999 Genetic and Evolutionary Computing Conference,* Orlando, FLA, pp 99-106

Hackworth, TW 1999 (b) India and Pakistan, a classic "Richardson" arms race**:** a genetic algorithm approach. *Proceedings of the 1999 Genetic and Evolutionary Computing Conference,* Orlando, FLA, pp 1543-1550

Hackworth, TW 2001 Combining Richardson's equations with canard explosion theory in a genetic algorithm to predict instabilities in the India / Pakistan arms race. *International Journal of Computer Mathematics,* **77**, pp 1-24

Hackworth, TW 2003 Applying genetic algorithms to Richardson's arms race equations to determine the stability of nuclear deterrence. *Proceedings of the 2003 Genetic and Evolutionary Computation Conference,* Chicago, IL , **3**, pp 135-141

Hammersley, JM and Handscomb, DC 1964 *Monte Carlo Methods.* Methuen and Co Ltd, London

Han, K-H and Kim, J-H 2003 Quantum-inspired evolutionary algorithms for a class of combinatorial optimisation. *IEEE Transactions on Evolutionary Computation*, **6** (6), pp 580-593

Harik, GR Lobo, FG and Goldberg, DE 1999 The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation,* **3**, pp 287-297

Haupt, RL and Haupt, S-E 1998 *Practical Genetic Algorithms.* John Wiley and Sons, Chichester

Heidegger, M 1926 *Being and Time*. Macquarrie and Robinson (Translators). Blackwell, Oxford

Holland, JH 1975 *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI

Holland, JH 1992 Genetic algorithms. *Scientific American,* **536**, pp 66 - 72

Hunter, B Ed. 1992 to 1997 (separate annual volumes) *Statesman's Year Book.* Macmillan Reference Press, London

Huxley, Sir Julian 1942 *Evolution : A Modern Synthesis*. Allen and Unwin, London

International Monetary Fund 1997 to 2001 (separate annual volumes) *IMF Statistics*. International Monetary Fund, New York

Jacob, C 2001 *Illustrating Evolutionary Computation with Mathematica*. Morgan Kaufmann, San Francisco, CA

Jones, T 1995 *Evolutionary Algorithms : Fitness Landscapes and Search*. PhD Thesis, University of New Mexico, Albuquerque, NM

Julstrom, BA 1999 Redundant genetic encodings may not be harmful. *Proceedings of the 1999 Genetic and Evolutionary Computing Conference,* Orlando, FLA, p 791

Kalos, MH and Whitlock, PA 1986 *Monte Carlo Methods* **1**: *Basics*. John Wiley and Sons, New York

Kantaris, N 1993 *Programming in C* (2nd edition). Bernard Babani Publishing, London

Kauffman, SA 1989 Adaptation on rugged fitness landscapes. In *Lectures in the Science of Complexity.* Stein, DL (Editor). Addison-Wesley, New York, pp 527-618

Kauffman, SA 1993 *The Origins of Order.* Morgan Kaufmann, San Francisco, CA

Köckler, N 1994 *Numerical Methods and Scientific Computing*. Clarendon Press, Oxford

Kohonen, T 1989 *Self-organisation and Associative Memory*. (3rd edition). Springer-Verlag, New York

Koza, J 1994 *Genetic Programming,* **2**. MIT Press, Cambridge, MA

Krzanowski, RM 1997 *Evaluation of Spatial Evolutionary Algorithms for Spatial Modelling.* PhD Thesis, Birkbeck College, University of London

Kumar, R and Rockett, P 1987 Assessing the convergence of rank-based multi-objective GAs. *Proceedings of the 2nd International Conference on Genetic Algorithms,* Cambridge, MA, pp 123-127

Lennox, SC and Chadwick, M 1977 *Mathematics for Engineers and Applied Scientists* (2nd edition). Heinemann, London

Levenick, JR 1991 Inserting introns improves GA success rate: taking a cue from biology. *Proceedings of the 4th International Conference on Genetic Algorithms,* San Diego, CA, pp 356-363

Levenick, JR 1999 Swappers : introns promote flexibility, diversity and invention. *Proceedings of the 1999 Genetic and Evolutionary Computation Conference*, Orlando, FLA, pp 361-368

Lucas, S 1997 Towards the evolution of scalable neural architectures. In *Genetic Algorithms in Engineering Systems.* Zazala, AMS and Fleming, PJ (Editors). Peter Peregrinus, London, pp 99-117

Lucasius, CB and Kateman, G 1992 GAs for subset selection and combinatorial optimisation. *Proceedings of the 3rd International Conference on Genetic Algorithms,* Arlington, VA, pp 170-176

Luce, AA 1934, reprinted 1947 *Berkeley and Malebranche: a study in the origins of Berkeley's thought.* Oxford University Press

Lukacs, E 1975 *Stochastic Convergence.* Academic Press Inc, New York

Marcer, PJ and Schempp, W 2000 A mathematically-specified template for DNA and the genetic code in terms of the physically-realisable processes of quantum holography. *Lehrstuhl für Mathematic,* **1**, pp 45-62

Mayer-Kress, G 1989 A non-linear dynamical approach to international security. Technical Report LA-UR-98-1355. Los Alamos National Laboratory, NM

Menczer, F Degeratu, M and Street, WN 2001 Efficient and scalable Pareto optimisation by evolutionary local selection algorithms. *Evolutionary Computation*, **8**, pp 223-247

Miettinen, K Neittaanmäki, P Mäkelä, MM and Périaux, J, Eds. 1999 *Evolutionary Algorithms in Engineering and Computer Science.* John Wiley and Sons, Chichester

Michalewicz, Z 1992 *Genetic Algorithms + Data Structures = Evolutionary Programs.* Springer-Verlag, Berlin

Mitchell, M 1996 *An Introduction to Genetic Algorithms.* MIT Press, Cambridge, MA

Moore, FW 2002 A methodology for missile countermeasures optimisation under uncertainty. *Evolutionary Computation*, **10** (2), pp 129-150

Mühlenbein, H 1994 The breeder genetic algorithm : a provisional optimal search algorithm and its application. *Proceedings of the IEEE Colloquium on Practical Genetic Algorithms,* San Diego, CA, pp 273-290

Murphy, IS 1984 *Basic Mathematical Analysis*. Arklay Publishers, Stirling, Scotland

Naudts, B Suys, D and Vershoren, A 1998 Epistasis as a basic concept in formal landscape analysis. *Proceedings of the 7th International Conference on Genetic Algorithms,* East Lansing, IL, pp 65-72

Nishio , K Murakami, M Mizutani, E and Honda, N 1997 Fuzzy fitness assignment in an interactive genetic algorithm for a cartoon face search. In *Genetic Algorithms and Fuzzy Logic Systems : Soft Computing Perspectives.* Sanchez, E Shibata, T and Zadeh, LA (Editors). World Scientific Publishing, Singapore, pp 175-192

Ofria, C Adami, C and Collier, TC 2002 Design of an evolvable computer language. *IEEE Transactions on Evolutionary Computation,* **6**, pp 420-424

Ott, E 1994 *Chaos in Dynamical Systems.* Cambridge University Press, Cambridge, UK

Pareto, V 1896 *Cours d'Economie Politique*. F Rouge, Lausanne, Switzerland

Patel, A 2000 Quantum algorithms and the genetic code. *Proceedings of the Winter Institute on Foundations of Quantum Theory and Quantum Optics,* Calcutta, pp 125-137

Peng, B Gáspár, V and Showalter, K 1991 False bifurcations in chemical systems. *Philosophical Transactions of the Royal Society of London, Series A,* **337**, pp 275-289

Peterson, I 1998 *The Jungles of Randomness: Mathematics at the Edge of Certainty*. Penguin Books, London

Pham, DT and Karaboga, D 2001 *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks.* Springer, London

Porter, R , Ed. 1994 *Hutchinson's Dictionary of Scientific Biography*. Helicon Press, London

Pudlák, P 2001 Complexity theory and genetics: The computational power of crossing over. *Information and Computation*, **171**, pp 201-223

Pyle, D 1999 *Data Preparation for Data Mining*. Morgan Kaufmann, San Francisco, CA

Radcliffe, N 1991 Formal analysis and random respectful recombination. *Proceedings of the 4th International Conference on Genetic Algorithms,* San Diego, CA, pp 143-150

Raich, A and Ghaboussi, J 1999 Frame design synthesis using implicit redundant genetic algorithms. *Proceedings of the 1999 Genetic and Evolutionary Computation Conference,* Orlando, FLA, pp 1691-1698

Richardson, JT Palmer, MR Liepins, G and Hilliard, M 1995 Some guidelines for genetic algorithms with penalty functions. *Proceedings of the 3rd International Conference on Genetic Algorithms,* Arlington, VA, pp 191-197

Richardson, LF 1960 (a) *The Statistics of Deadly Quarrels*. Boxwood Press, Pittsburgh, PA

Richardson, LF 1960 (b) *Arms and Insecurity*. Boxwood Press, Pittsburgh, PA

Rudolph, G 1993 Massively parallel simulated annealing and its relation to evolutionary algorithms. *Evolutionary Computation,* **1**(4), pp 381-407

Rudolph, G 2001 Self-adaptive mutations may lead to premature convergence. *IEEE Transactions of Evolutionary Computation,* **5**, pp 410-414

Saaty, TL and Alexander, J 1981 *Thinking with Models: Mathematical Models for the Physical, Biological and Social Sciences.* Pergamon Press, Salem, OR

Sanchez, E Shibata, T and Zadeh, LA , Eds. 1997 *Genetic Algorithms and Fuzzy Logic Systems* **:** *Soft Computing Perspectives.* World Scientific Publishing, Singapore

Schaffer, JD Caruana, RA Eshelman, LJ and Das, R 1993 A study of control parameters affecting the on-line performance of GAs for function optimisation. *Proceedings of the3rd International Conference on Genetic Algorithms,* Arlington, VA, pp 51-60

Schoenauer, M 1996 Shape representations and evolution schemes. *Proceedings of the 5th International Conference on Evolutionary Programming,* San Diego, CA, pp 121-129

Schrödinger, E 1944. *What is Life?* (1996 edition). Cambridge University Press, Cambridge, UK

Schwefel, H-P 1995 *Evolution and Optimum-Seeking*. John Wiley and Sons, Chichester

Scott-Gordon, V Pirie, R Wachter, A and Sharp, S 1999 Terrain-based genetic algorithms (TBGA): modelling parameter space as terrain. *Proceedings of the 1999 Genetic and Evolutionary Computation Conference,* Orlando, FLA, pp 229-235

Shannon, CE 1949 *The Mathematical Theory of Communication.* The University of Illinois Press, Urbana, IL

Shaw, KJ Northcliffe, AL Thompson, M Fonseca , CM Love, J and Fleming, PJ 1999 Assessing the performance of multi-objective genetic algorithms for the optimisation of a batch process scheduling problem. *Proceedings of the Congress on Evolutionary Computation 1999,* Washington DC, pp 37-45

Simon, HA 1969 *The Sciences of the Artificial*. MIT Press, Cambridge, MA

Simon, R 1997 Robust encodings in genetic algorithms. In *Evolutionary Algorithms in Engineering Applications.* Dasgupta, D and Michalewicz, Z (Editors). Springer-Verlag, New York, pp 29-44

Skeina, S 1994 The *Algorithm Design Manual*. Telos-Springer-Verlag, New York

Smith, RE and Goldberg, DE 1992 Diploidy and dominance in artificial genetic search. *Complex Systems,* **6**, pp 251-285

Stockholm International Peace Research Institute 1970 to 1982 (thirteen separate volumes) *SIPRI Annual Handbook*. Oxford University Press, UK

Syswerda, G 1989 Uniform crossover in genetic algorithms. *Proceedings of the 3rd International Conference on Genetic Algorithms*, Arlington, VA, pp 2-9

Turner, B , Ed. 1992 to 2003 inclusive (twelve separate volumes). *The Statesman's Year Book.* Macmillan Press, London

Turner, M 1999 *Performance-based Self-adaptive Evolutionary Behaviour*. PhD Thesis, George Washington University, Mount Vernon, DC

UK Ministry of Defence 1999 *UK Defence Estimates*. HM Stationery Office, London

United Nations Statistical Office 1997 to 2000 (four separate volumes) *United Nations Statistical Yearbook.* United Nations, New York

Vose, MD 1999 *The Simple Genetic Algorithm: Foundations and Theory.* MIT Press, Cambridge, MA

Vose, MD and Liepens, GE 1990 Genetic algorithm applications to set covering and travelling salesman problems. In *OR / AI Integration of Problem-solving Strategies.* Brown, PH (Editor). Springer-Verlag, New York, pp 132-145

Whitley, D 1988 GENITOR: a different genetic algorithm. *Proceedings of the Rocky Mountain Conference on Artificial Intelligence,* Denver, CO , pp 185-197

Whitley, D Mathias, K and Fitzhorn, P 1991 Delta coding **:** an iterative search strategy for genetic algorithms. *Proceedings of the 4th International Conference on Genetic Algorithms,* San Diego, CA, pp 77-84

Wiles, J and Hallinan, J 2001 Evolutionary computation and cognitive science: modelling evolution and evolving models. *IEEE Transactions on Evolutionary Computation,* **5**, pp 85-91

Wolpert, D and Macready, W 1995 No free lunch theorems for search. Santa Fé Technical Institute Technical Report SFI –TR-95-02-010. Santa Fé, NM

Wright, AH 1991 Genetic algorithms for real parameter optimisation. In *Foundations of Genetic Algortihms,* Rawlins, JE (Editor). Morgan Kauffman, San Francisco, CA, pp 205-218

Yao, X 1999 *Evolutionary Computation: Theory and Application*. World Scientific Publishing Co., Singapore

Yu, T and Bentley, PJ 1998 Methods to evolve legal phenotypes. *Proceedings of the* 5th *International Conference on Parallel Problem Solving from Nature,* Amsterdam, pp 280-291

Zadeh, L 2002 *Soft Computing*. http:// www.soft-computing.de/def.html

Appendix R1

FULL-SIZE CHARTS

Charts for percentage defence expenditures are shown here full size for

- Greece and Turkey 1955 to 2000,
- India and Pakistan 1993 to 1999
- India and Pakistan 1955 to 2000, and
- The Middle East 1955 to 2000

These previously appeared in the text as Figures 6.5, 5.7, 6.10 and 6.15.