BRUNEL UNIVERSITY

Power Transmission Planning Using Heuristic Optimisation Techniques: Deterministic Crowding Genetic Algorithms and Ant Colony Search Methods

A thesis submitted for the degree of Doctor of Philosophy

by

Hind Munzer Chebbo

Department of Electronic and Computer Engineering

Brunel University

January 2000

Power Transmission Planning Using Heuristic Optimisation Techniques: Deterministic Crowding Genetic Algorithms and Ant Colony Search Methods

Hind Munzer Chebbo

Department of Electronic and Computer Engineering,

Brunel University, Uxbridge, Middlesex, UB8 3PH, U.K.

Ph.D Thesis, January 2000

Abstract

The goal of transmission planning in electric power systems is a robust network which is economical, reliable, and in harmony with its environment taking into account the inherent uncertainties. For reasons of practicality, transmission planners have normally taken an incremental approach and tended to evaluate a relatively small number of expansion alternatives over a relatively short time horizon.

In this thesis, two new planning methodologies namely the Deterministic Crowding Genetic Algorithm and the Ant Colony System are applied to solve the long term transmission planning problem. Both optimisation techniques consider a 'green field' approach, and are not constrained by the existing network design. They both identify the optimal transmission network over an extended time horizon based only on the expected pattern of electricity demand and generation sources.

Two computer codes have been developed. An initial comparative investigation of the application of Ant Colony Optimisation and a Genetic Algorithm to an artificial test problem has been undertaken. It was found that both approaches were comparable for the artificial test problem.

These two methods have then been applied to a range of problem classes derived from a 23-node 49-route transmission network design that represents a simplified version of the England and Wales transmission network. The various problem classes are classified according to the objective function of the transmission-planning problem. It was found that the Deterministic Crowding Genetic Algorithm and Ant Colony Search methods are applicable to the transmission-planning problem. However, the Deterministic Crowding Genetic Algorithm model is more efficient computationally than the Ant Colony System for more realistic problem classes.

Acknowledgements

First and foremost I wish to express my profound thanks and gratitude to my supervisor Professor Malcolm Irving for his supervision, guidance and support throughout the project. I would like to express my gratitude to the assistance and help given to me by Professor Young Hua Song my second supervisor, and members of the BIPS group, especially Dr Jeremy Daniel, the Computer System Manager for the group, for his technical support. Thanks are also due to the EPRSC and the National Grid Company plc, for their financial support, and especially to Dr Lewis Dale, Dr Min Zhu, Dr Chris Aldridge, Dr Peter Pals, and Miss Ursula Bryan for their valuable suggestions, encouragement, and guidance. Finally I would like to express my thanks to my parents for their belief in me and for teaching me the value of hard work, my husband for his love, patience, understanding and encouragement through the years, and my two wonderful children Mostafa and Zakie for keeping me sane while this work was ongoing.

DECLARATION

The work described in this thesis has not been previously submitted for a degree in this or any other university, and unless otherwise referenced it is the author's own work.

STATEMENT OF COPYRIGHT

Contents

Figures

Tables

Chapter 1

Introduction

1.1 Power System Planning

Most power system planners use the existing system as a starting point to expand their network. The power system expansion planning problem is concerned with obtaining expansion plans which dictate what new generation and transmission facilities to add, when to add, and where. The objective is to select the most economical and reliable expansion plans in order to meet future power demand at minimum cost and maximum reliability over a long period of time, usually 5 to 40 years, subject to a multitude of technical, economical, environmental, legal and political constraints. The practical planning problem is an extremely difficult and complex design problem. Costs and constraints can not easily be quantified and are difficult to predict.

At present, most power systems expansion-planning methods are based on a sequence of simulation studies such as load flow, stability, reliability, and economic studies. Clearly this approach requires a large number of simulation runs and a combination of experience and judgement by the system planner, to determine a satisfactory (not necessary optimum) plan. Since the cost of power system equipment is high and the operational and reliability requirements are severe, most of the current effort is concentrated on more accurate mathematical models and efficient computational analysis methods.

Because of the complexity of the problem, there is no single method that can solve the whole expansion problem. Therefore the problem is normally decomposed into two major long-term expansion problems, namely, optimal generation expansion planning and optimal transmission expansion planning [Kern, Young and Kwang, 1988].

1.2 Transmission Planning Problem

The goal of the transmission planning problem is the design of an electricity transmission network, which is as economical as possible while providing a reliable energy supply. The mathematical formulation leads to a complicated, integer-valued, non-convex, non-linear mathematical programming problem. The complexity of the problem arises mainly from the large number of problem variables, combined with the multitude of technical and economical constraints.

The planning problem is typically broken down into two stages:

- long-term transmission planning,
- mid-term transmission planning.

In the long-term stage, the objective is to meet the total demands at the lowest investment cost, so as to establish the guidelines for the future network structure, while leaving a number of details be decided in the mid-term planning, e.g., those concerning transient stability limits, voltage violations, reactive power flow and short circuit capacity.

The total cost of the transmission system may be up to half the capital cost of generating plant. By ensuring economical design it is possible to produce significant savings in the overall cost of the power system.

Good transmission system design should deliver the following:

- maximum security of supply commensurate with the cost of providing the service,
- provision for future expansion,
- ease of maintenance,
- safety in operation,
- minimum operating costs.

There are several planning algorithms available for the solution of the long-term problem, each based on a different interpretation of the system model and choice of the design objective. These include heuristic algorithms and mathematical optimisation techniques. In this thesis, two novel heuristic methodologies namely, the Deterministic Crowding Genetic Algorithm (DCGA) and Ant Colony Search (ACS), are proposed for the solution of the long-term planning problem. These are based on an original problem formulation termed the 'green field' approach.

1.3 Transmission Planning Methodologies

Transmission planning methods can be classified into two main categories:

- 'traditional' or 'incremental' approach,
- 'green-field' or 'non-incremental' approach.

1.3.1 Traditional Approach

For reasons of practicability, transmission planners have normally taken an incremental approach and tended to evaluate a relatively small number of expansion alternatives over a relatively short time horizon. The general form of the network expansion problem can be stated as follows: Given

(i) load-generation patterns for the target year,

(ii) existing network configuration,

(iii) possible new routes (lengths and way-leaves),

(iv) available line types and the corresponding cost,

estimate:

the optimum network which feeds the loads with the energy required while respecting the transmission security standards.

The appropriate solution tools for such problems include the standard mathematical programming techniques. The general form of these techniques is

$$\text{Minimise } F(x_1, x_2, x_3, \dots, x_n)$$

$$\text{Subject to } G_i(x_1, x_2., x_3, \dots, x_n), \quad i = 1, 2, \dots, m.$$

According to the form of F, G and x, the techniques are classified into linear, integer, zero-one integer, mixed integer linear, non-linear, etc.

1.3.2 Green-field Approach

Numeric techniques now offer the possibility of explicitly evaluating a much larger number of alternatives including the automatic selection of route alternatives, voltage levels and line construction types. The objective function can be designed to derive the cheapest cost network that enables the generated energy to be transmitted to satisfy the demand while respecting the transmission security standards. This approach is also based on the load and demand patterns for the target year, all possible routes, and the available transmission line types and their corresponding costs. However, it is not restricted by an existing network. The application of a 'green field' approach was suggested by the National Grid Company, who also provided the basic data used in this research.

The two proposed optimisation techniques, namely the Deterministic Crowding Genetic Algorithms and the Ant Colony System, will be applied to this problem formulation.

1.4 Conventional Optimisation Techniques

Before an optimisation method is applied, the transmission-planning problem must be adequately modelled, taking into account the various transmission system constraints.

The number of options to be analysed increases exponentially with the size of the network problem. There are a large number of local optimal solutions (a highly multimodal landscape), which makes the chance of a solution method becoming trapped in one of them very high.

Several classical optimisation techniques have been used previously:

- linear programming [Chanda and Bhattacharjee, 1994]
- non-linear programming [Gilles, 1986], [Youssef and Hackam, 1989],
- mixed integer programming [Gilles, 1986], [Santos, Franca and Said, 1989] ,
- Benders Decomposition [Pereira, Pinto, Cunha, Oliveira, 1985], [Romero and Monticelli, 1994] and an associated hierarchical approach [Romero and Monticelli, 1994], [Latorre-Bayona and Perez-Arriaga, 1994],
- and others [Pereira, Pinto, Cunha, and Oliveira, 1985], [Monticelli, Santos, Pereira, Cunha, Parker and Praca, 1982], [Rudnick, Palma, Cura and Silva, 1996].

Although these methods are successful in transmission planning (for small and medium size problems), they present some drawbacks:

- Due to the non-convexity existing in the network expansion planning, the success of the search still largely depends on the starting points. Therefore, the optimisation process sometimes stops at non-optimal solutions.
- The non-linearity of the problem increases the iterations of the optimisation algorithm and sometimes causes divergence.

As an alternative to conventional optimisation methods, different heuristic search algorithms have been utilised, based on sensitivity analysis [Pereira, Pinto, Cunha, Oliveira, 1985], [Monticelli, santos, Pereira, Cunha, Parker and Praca, 1982], [Latorre-Bayona and Perez-Arriaga, 1994].

1.5 Heuristic Search Algorithms

Most of the conventional optimisation algorithms used in practice are however unable to generate optimal solutions for larger, complex networks [Gallego, Monticelli, and Romero, 1998]. As an alternative, various heuristic search algorithms have been addressed. These include Genetic Algorithms (GA), Ant Colony Search methods (ACS), Simulated Annealing (SA), Tabu Search (TS) and/or hybrids of the aforementioned algorithms. To a certain extent these approaches are based on processes found in nature. GAs are based on natural genetic and evolutionary processes. ACS methods take inspiration from the behaviour of real ant colonies. SA mimics the physical process of solidification and formation of perfect crystals. Finally, TS generalises concepts from the field of Artificial Intelligence (AI).

In this thesis we will concentrate on two proposed methodologies for the long term planning problem namely the Deterministic Crowding GA model and the Ant Colony Search model.

1.5.1 Genetic Algorithms

Genetic Algorithms [Holland, 1975], [Davis, 1991], [Goldberg, 1989] belong to a class of evolutionary computation techniques based on models of biological evolution. These methods have proved useful in domains that are not well understood, or for search spaces which are too large to be efficiently searched by standard methods. The GA paradigm uses selection and recombination in various formulations to sample the (usually coded) search space. In the GA, solutions to the problem are coded to mimic the genetic make up of biological organisms. Each individual in the population represents a possible solution to the problem. A fitness value, derived from the problem objective function is assigned to each member of the population. Individuals that represent better solutions are awarded higher fitness values, thus enabling them to survive more generations.

There are many ways of defining and interpreting GAs, but for the purpose of the thesis we will primarily consider them as computational optimisation techniques. As such, the GAs themselves have to compete with alternative heuristic methods such as SA, TS, ACS etc, and also with mathematical programming techniques such as non-linear programming.

Different GA models depend on factors such as:

- selection method and mechanism,
- parent replacement method,
- crossover and mutation method,
- problem to be solved, whether a single or multiple objective formulation is required.

The GA model to be used is chosen after a careful analysis of the problem to be solved. In this thesis, a particular GA model known as the Deterministic Crowing Genetic Algorithm (DCGA) has been applied to the transmission planning problem. Previous experience with power system problems has shown this technique to be more efficient and robust than the canonical GA.

1.5.1.1 Deterministic Crowing Genetic Algorithm

Crowding is inspired by the ecological phenomena, where similar members of a natural population compete for the same resources [Spears, De Jong and Baeck, 1993], providing a model where only a fraction of the population reproduces and dies in each generation, each newly created population member replacing an existing member, preferably the most similar. Through the analysis of the model described in reference [Spears, De Jong and Baeck, 1993] and other crowding methods, the deterministic crowding genetic algorithm (DCGA) [Mahfoud, 1992] has been designed and exhibits extensive capabilities in solving a wide range of problems such as:

- multi-modal function optimisation,
- multi-objective function optimisation,
- simulation of complex and adaptive systems,
- learning and classification.

The technique can be regarded as a generalised niching method that helps in overcoming the time and memory limitation problems faced by many practical GAs, that use population sizes which cannot maintain the required diversity as the GA run progresses. It is also capable of forming and maintaining single or multiple solutions to a problem, and basically achieves this by providing selection pressure within but not across regions of the search space, leaving the search across the regions to the crossover operator. The combination of its crossover and selection-replacement mechanisms is mainly responsible for its success.

1.5.2 Ant Colony Search

Ant Colony Search (ACS) methods are artificial systems that take inspiration from the behaviour of real ant colonies and which are used to solve function or stochastic combinatorial optimisation problems. These are population based, co-operative search algorithms. The first ACS system was the Ant System (AS), proposed by Dorigo in his Ph.D. thesis (1992). Currently, most work has been done in the direction of applying

ACS to combinatorial optimisation. Ant Colony Search algorithms, to some extent, mimic the behaviour of real ants. As is well known, real ants are capable of finding the shortest path between and from food sources to the nest without using visual cues. They are also capable of adapting to changes in the environment, for example, finding a new shortest path once the old one is no longer feasible due to a new obstacle. It was found that such capabilities are due to what is called a pheromone trail, which ants use to communicate information among individuals regarding paths. Ants deposit a certain amount of recruit pheromone when they move, and each ant prefers to follow a path rich in pheromone rather than a poorer one.

In ACS, the colony consists of many homogeneous artificial ants communicating among them by recruit pheromone. The ants change their behaviour according to the situation. Firstly, they walk randomly to search for a food source, operating in a discrete-time environment. They will not be completely blind, a decision is made based on the intensity of trail perceived and the visibility. Each ant will have also some memory about its location and the next possible move. According to the objective function, their performance will be weighted as a fitness value, which directs influence to the level of trail quantity deposited in the path selected by ants. Each ant's decision to choose the next node to move to depends on two parameters: the visibility of the node and the trail intensity previously laid by other ants.

The main characteristics of this model are:

- positive feedback,
- distributed computation,
- and the use of a constructive greedy heuristic.

Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence, and the greedy heuristic helps find acceptable solutions in the early stages of the search.

1.6 Advantages of Heuristic Search Algorithms

Over the past three decades, there has been a considerable interest in heuristic search techniques for complex optimisation problems [Reeves, 1995], [Aarts and Lenstra, 1997]. The main reasons for considering the two proposed heuristics algorithms (DCGA and ACS) are summarised as follows:

- The transmission planning problem is a complex large scale mixed integer non-linear optimisation problem, which has given enormous challenges to the presently available computational techniques. The flexibility, computational simplicity, robustness and recent success of these heuristic methods in various complex problem domains, in particular power optimisation problems, make such methods attractive.
- The heuristic search algorithms use payoff (fitness or objective function) information directly for the search direction, not derivatives or other auxiliary knowledge. Therefore, they can deal with non-smooth, non-continuous and non-differentiable functions that are characteristic of many real-life optimisations problems. This property also relieves these heuristic search methods from the need for approximate assumptions for many practical optimisation problems (assumptions which are quite often required by traditional optimisation methods).
- These heuristic methods use probabilistic transition rules in search for the global optimum, as opposed to the deterministic rules used by mathematical techniques. An important advantage of these heuristic methods is the possibility that more than one

local optimum will be explored, and there is a chance that these methods may discover a global optimal solution. This is in contrast to mathematical programming techniques, which are generally equipped to seek only a local optimum.

1.7 Thesis Layout

Chapter 2 introduces the modelling of the transmission-planning problem and its goals. A literature review of the various optimisation techniques is then presented.

In chapter 3, two heuristic methodologies namely DCGA and ACS are considered for the solution of the transmission-planning problem. A summary of the theoretical foundations, modelling framework and previous areas of application is presented.

Chapter 4 illustrates the structure of the two software programs which have been developed. A comparative investigation of the application of Ant Colony Search (ACS) and a Deterministic Crowding Genetic Algorithm to an artificial test problem is undertaken.

Chapter 5 reports on the application of the proposed methods to a real system, a simplified version of the England and Wales transmission network with 23 nodes and 49 wayleaves. We present an analysis of the performance of both proposed methods for the various categories of the transmission-planning problem considered. These include minimisation of:

- losses only,
- investment only,
- investment and losses,
- investment, losses and security requirements.

In chapter 6, we extend the transmission modelling design by including voltage transformation and maintenance costs.

Chapter 7 presents two sensitivity analyses based on discount rate and decision variables and interpretation of the optimum results.

In chapter 8, the main conclusions of the research are presented, and some suggestions for future research are given.

Finally, a list of references and bibliography are provided at the end of the thesis.

1.8 Original Contributions of the Thesis

1. This work has reviewed the theoretical development of the two proposed heuristic methodologies: the Deterministic Crowding Genetic Algorithm and the Ant Colony System. The thesis also includes an overview of their various applications with emphasis on power system fields.
2. The Deterministic Crowding Genetic Algorithm and the Ant Colony System have been applied to solve the long-term transmission planning problem. Both optimisation techniques consider a 'green field' or 'non-incremental' approach, and are not constrained by an existing network design.
3. Two detailed algorithm-modelling frameworks, DCGA and ACS, for the long-term transmission planning are developed. A rigorous analysis for both methods based on different categories of the objective function of the transmission planning problem is undertaken. These categories represent a range of problems derived from a 23-node 49-route transmission network design problem that represents a simplified version of the England and Wales grid. The performance of the algorithms, their effectiveness and suitability for the solution of planning problem, have been investigated.

4. Two GA representations namely, binary and integer, are implemented and their effects on the performance of DCGA are investigated. A limited number of experiments with the integer representation, have been carried out to assess the relative merits of this representation for the present problem. It was concluded that the binary approach was superior in the cases considered. Therefore, further modelling and analysis of the transmission system design is undertaken with the binary representation and only the corresponding simulation results are reported in this thesis.
5. A modified definition of uniform crossover, first proposed in this thesis, is implemented and has been shown to result in substantial improvements in GA performance.
6. An initial comparative investigation of the application of Ant Colony Optimisation and a Genetic Algorithm to an artificial test problem has been undertaken. It was found that both approaches had comparable performances on the artificial test problem.
7. The DCGA and ACS methods have then been applied to a realistic network model with 23 nodes and 49 possible wayleaves that represents a simplified version of the England and Wales grid. It was found that both algorithms are applicable to the transmission planning problem but the Deterministic Crowding Genetic Algorithm model is more efficient computationally than the Ant Colony System.
8. Further modelling of the transmission system design to incorporate transformation and maintenance costs is performed using the DCGA model.

9. Two sensitivity analyses based on discount rate and decision variables are carried out independently. The objective being to investigate the effect of the discount rate on the best solution provided for problem class D and to further investigate the search space of the transmission-planning problem.
10. Finally, conclusions are drawn and some proposals for future work are presented.

Chapter 2

Transmission Network Planning Problem

2.1 Introduction

It is recognized that the allocation of transmission costs in a competitive environment will require more careful evaluation of alternative transmission plans. As a result, the need for methods that are able to synthesize optimal transmission plans has become more important.

Good transmission system design should deliver the following:

- maximum security of supply commensurate with the cost of providing the service,
- provision for future expansion,
- ease of maintenance,
- safety in operation,
- minimum operating costs.

Transmission planning is usually performed using an incremental approach that acknowledges the existence of an initial network.

Network expansion planning can be classified as:

- static, or
- dynamic.

Static expansion involves one-stage transitions. The network model is analyzed for (usually) one year in the planning horizon and consequently expansion alternatives are evaluated. By dynamic planning, it is usually meant the year-by-year expansion that starts from the initial year through to the horizon year. In almost all cases the dynamic (timed-phased) mode of planning [Youssef and Hackam, 1989] has been either ignored or handled by a sequence of static plans, which might not lead to the overall optimal solution. Transmission planners have worked mainly with static models as opposed to the dynamic model which only a few researchers have considered. Dynamic planning is usually undertaken with the help of an interactive tool [Youssef and Hackam, 1989] and is not discussed in the present research.

2.2 Problem Description

The goal of the transmission-planning problem is the design of an electricity transmission network, which is as economical as possible while providing a reliable energy supply. The mathematical formulation leads to a complicated, integer-valued, non-convex, non-linear mathematical programming problem. The complexity of the problem arises mainly from the large number of problem variables, combined with the multitude of technical and economical constraints.

The planning problem is typically broken down into two stages:

- long-term transmission planning,
- mid-term transmission planning.

In the long-term stage, the objective is to meet the total demands at the lowest investment cost, so as to establish the guidelines for the future network structure, while leaving a number of details be decided in the mid-term planning, e.g., those concerning transient stability limits, voltage violations, reactive power flow and short circuit capacity. This research is exclusively devoted to the solution of the long term-transmission planning.

The representation of the transmission planning problem in mathematical form is achieved by defining the cost function (problem objective function) and constraints.

2.2.1 Problem Objective Function

The objective of transmission planning is to minimize investment cost and annuitised energy loss cost while satisfying system constraints.

The objective function has the following main components: investment cost, operation cost (including loss cost), and in some methodologies penalties associated with the violation of the constraints. Investment cost is a function of the decision variables (variables representing the addition of new transmission equipment); operation costs and penalties are functions of the continuous operation variables (power flows).

2.2.2 Problem Constraints

The constraints include the following:

- performance equations: these equations are the main constraints as they govern the power flow. They may be in exact nonlinear form (AC load flow equations) or in approximate linear form (DC load flow equations) [El-sobki, El-Metwally and Farrag, 1986]. More approximate linear forms may be used where impedance-less lines are assumed (the power conservation at each bus is the only constraint).
- Quality constraints: these include line loading and/or magnitude of bus-voltage constraints.
- Reliability level constraints: these encompass security against outage and/or maintenance criteria. Reliability is usually realized in a separate stage after finding the least cost configuration satisfying quality constraints.

2.2.3 Transmission Planning Methodologies

The transmission planning methods can be classified into two main categories:

- 'traditional' or 'incremental' approach,
- 'green-field' or 'non-incremental' approach.

2.2.3.1 Traditional Approach

For reasons of practicability, transmission planners have normally taken an incremental approach [Garver, 1970], [El-sobki, El-Metwally and Farrag, 1986], [Rudnick, Palma, Cura. and Silva, 1996] and tended to evaluate a relatively small number of expansion alternatives over a relatively short time horizon. The general form of the network expansion problem can be stated as follows:

Given:

(i) load and generation patterns for the target year,

(ii) an existing network configuration,

(iii) all possible new routes (lengths and way-leaves),

(iv) the available transmission line types and their corresponding costs.

Determine:

an optimum network configuration which supplies the loads with the energy required at the lowest possible cost, while meeting specified transmission security standards.

The appropriate solution tools for such problems include the standard mathematical programming techniques. The general form of the problem to be solved by these techniques is

$$\text{Minimise } F(x_1, x_2, x_3, \ldots, x_n)$$

$$\text{Subject to } G_i(x_1, x_2, x_3, \ldots, x_n), \quad i = 1, 2, \ldots, m.$$

According to the form of F, G and x, the techniques are classified into linear, integer, zero-one integer, mixed integer-linear, non-linear, etc. Table 2.1 illustrates the various mathematical techniques applied to the transmission-planning problem.

The majority (if not all) of transmission planning techniques belong to this category.

Table 2.1 Mathematical techniques applied to the transmission-planning problem

Optimisation Techniques	Variables		Constraints		Objective Function				Global Optimisation
	continuous (real-value)	discrete (integer)	linear	non-linear	linear	non-linear	convex	Non-convex	
Linear Programming (LP)	√	X	√	X	√	X	√	X	√
Quadratic Programming (QP)	√	X	√	X	X	√	√	X	√
nonlinear programming (NLP)	√	X	√	√	√	√	√ most methods	√ some methods	√ for convex
Integer Programming (IP)	X	√	√	X	√	X	X	√	√*
Mixed Integer Programming (MIP)	√	√	√	X	√	X	X	√	√*
Genetic Algorithms (GA)	√	√	√	√	√	√	√	√	√
Ant Colony Search (ACS)	√	√	√	√	√	√	√	√	√

* may require excessive computation to obtain exact global optimum.

2.1.1.2 Green Field Approach

An alternative approach, termed as a 'green field', or 'non-incremental' approach, is to design an optimal network based on available routes, but without regard to the existing network. This problem can be stated as:

Given:

(i) load-generation patterns for the target year,

(ii) all possible routes (particularly their location and lengths),

(iii) the available transmission line types and their corresponding costs.

Determine:

an optimum network configuration which supplies the loads with the energy required at the lowest possible cost, while meeting specified transmission security standards.

This non-incremental approach is impractical for monthly and annual planning decisions, which must be compatible with the existing network configuration. Nevertheless, there is some motivation for developing the green field approach as a complementary facility. In particular, for deregulated utilities there may be a need for an optimal network design against which the existing configuration may be benchmarked. It is useful for planners (who are routinely applying the practical incremental approach) to have a long-term future target available for reference. For example, if an optimal green-field network design is available, engineers can gain insight into which of their plans are heading 'towards' the optimal design. Conversely, some short-term plans may appear to be diverging from the future optimum, and could therefore be regarded as less effective in the longer term.

2.3 Review of Optimisation Techniques

Practical application of conventional (mathematical) optimisation techniques to large scale nonlinear mixed integer problems, such as transmission network expansion, are usually not possible due to their complexity. As an alternative, different heuristic search algorithms rooted in natural and physical processes have been addressed. Optimisation techniques are classified into two main categories (see figure 2.1):

- Conventional (mathematical) optimisation techniques,

- Heuristic optimisation techniques.

Good solutions can also be obtained by hybridizing both categories.

There are several planning algorithms available for the solution of the long-term problem, each based on a different interpretation of the system model and a choice of the design objective.

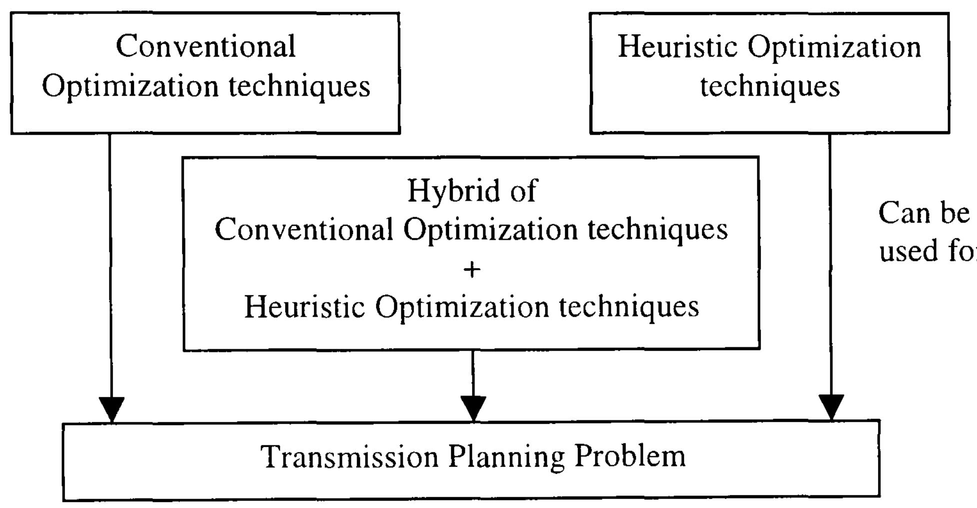


Figure 2.1 Transmission planning optimization techniques

2.3.1 Review of Heuristic Techniques

Over the past three decades, there has been a growing interest in heuristic search methods for complex optimisation problems [Reeves, 1995], [Aarts; Lenstra , 1997]. In particular, heuristic search algorithms rooted in natural and physical processes have been applied successfully to various combinatorial optimisation problems. These include Tabu Search (TS), Simulated Annealing (SA), Evolutionary Algorithms (EA), and Ant Colony Search (ACS) among others.

2.3.1.1 Tabu Search

With its roots going back to the late 1960's and early 1970's, the Tabu Search was proposed in its present form by Glover [Glover, Laguna, 1997], [Glover, 1989 and 1990]. As a heuristic search strategy for efficiently solving combinatorial optimisation problems, TS has now become an established optimisation approach that is rapidly spreading into many new fields, and has achieved impressive practical successes in extensive application areas [Song, 1999]. For example, TS has been successfully applied to obtain optimal or sub-optimal solution to problems such as the Travelling Salesmen Problem (TSP), time table scheduling and network layout design. TS is a restricted neighborhood search technique, and is an iterative algorithm. The fundamental idea of TS is the use of flexible memory of search history, which thus guides the search process to surmount local optimal solutions.

2.3.1.2 Evolutionary Algorithms

Evolutionary Algorithms (EA) are computer-based problem-solving systems based on principles of evolutionary theory. A variety of EA have been developed and they all share a common conceptual base of simulating the evolution of individual structures via processes of selection, mutation, and recombination. The processes depend on the perceived performance of the individual structures as defined by an environment. The interest in these algorithms has risen fast, for they provide robust and powerful adaptive search mechanism. Furthermore these approaches are well suited to deal with all kinds of problems that usually represent nightmares for researchers and developers: integer variables, non-convex-functions, non-differential functions, domains not connected, badly-behaved functions, multiple optima, multiple objectives, and fuzzy data.

The most popular EAs developed so far are the following: Genetic Algorithms, Evolutionary Programming, Genetic Programming and Classifier Systems.

These methods have been applied to various applications in power systems and other related fields. The vast majority of the applications use GA. However the interest in the use of other EA techniques (especially CS and EP) is rising fast.

[Lai, Ma, Wong, Yokoyama, Zhao and Sasaki, 1996] proposed an application of evolutionary programming (EP) to solve the long-term transmission-planning problem. The objective function considered is to minimize the investment cost and the load curtailment subject to the following constraints: satisfaction of load demand (DC power flow is used as equality constraint), the active power flow limits in the transmission lines, the generation restrictions, the voltage angle difference limits of the transmission lines and the maximum number of new lines. The constraints are treated as quadratic penalty terms that are added to the objective function. EP has been applied to a six-bus test system (maximum load is 240 MW, maximum generation is 600 MW). The authors claim that EP is capable of solving the transmission-planning problem.

Rudnic, Palma, Cura and Silva [1996] formulate a new methodology based on GA to determine an economically adapted transmission system in a deregulated open access environment. The objective function is to evaluate the transmission investment and losses combined with the variable cost of generation subject to an optimal generation investment indicative plan, generation operational costs, estimated load growth and distribution, and predefined transmission paths or rights of way. (n-1) security is also included within the simulation. This method is applied successfully to a reduced model of the original Chilean system (8 buses and 10 possible line paths). The computational time is 8 hours on HP 715 workstation.

Gallego, Monticelli, and Romero [1998] present an extended genetic algorithm for solving the optimal transmission network expansion-planning problem. Two main improvements have been introduced to the conventional GA: a) initial population obtained by a modified Garver's algorithm [Garver, 1970]; b) mutation approach inspired by simulated annealing. The extended GA has been successfully applied to three different systems. Two small systems with known solutions (to tune the main parameters of the GA) and a large system (the North-Northeastern system with 89 buses and 183 possible routes) for which no solution is known. The total demand is 29754 MW. The decimal representation has been adopted and each member represents the line additions in the corresponding initial routes. The objective function considered has two components: investment costs and penalties associated with loss of load. The transmission network used is the transportation model (impedance-less network; Kirchhoff Voltage Law, KVL, not represented) which is less accurate than the DC power flow model although solutions can be computed faster. The authors claim that the methodology proposed in this paper is more efficient than any other method proposed before, such as mathematical optimisation based on Benders decomposition and simulated annealing methods [Romero, Gallego and Monticelli, 1996], [Gallego, Alves and Monticelli, 1997].

2.3.1.3 Simulated Annealing

Simulated Annealing is an optimisation technique based on a well-known process found in nature, the metal cooling process. It was first proposed by Kirkpatrick in the mid 1970s. He demonstrated that the method is successfully applied to combinatorial optimisation problem. He has introduced several applications [Kirkpatrick, Gelatt and

Vecchi, 1983]. These include physical design of computers, wiring and the traveling salesmen problem.

SA tries to avoid local optima by allowing a temporary limited deterioration of the current solution. In that regard, it radically differs from conventional optimisation techniques that always proceed by deterministic exchanges, which may lead to local optima. In the SA approach, state transitions leading to increases in the objective function can be accepted with a certain probability.

SA has been also applied successfully to a wide range of applications in electrical engineering including power systems [Song, 1999]. These include unit commitment, generator maintenance scheduling, VAR resource planning, network planning and distribution system planning, etc.

Generally speaking, SA can be effectively applied to combinatorial optimisation problems where it is difficult to develop an efficient mathematical programming algorithm for solving them.

The method proposed by Romero, Gallego and Monticelli, [996], has been applied to a large, difficult test case (89 buses and 189 possible routes) for which no optimum is known. The objective function is to minimize investment cost with zero loss of load subject to the generation limits, transmission limits, and the satisfaction of load demand requirement through the use of a DC load flow to represent the network.

It is been proven that the SA produced good results, but the computer time was in the range of 18 to 19 hours. The potential drawback of this approach, as has been reported in other power system applications, is the computational effort required solving large-scale problems.

2.3.1.4 Expert Systems

An Expert System is a computer program that behaves like an expert in some, usually narrow, domain and is capable of solving problems that require that knowledge. Expert systems derive their power mainly from the problem domain knowledge they possess and from their use of 'rules of thumb' in going about their problem solving activities. They can be used as computer-based consultants to humans in the performance of complex tasks. Early applications of expert system were in medical diagnosis and therapy, and computer system configurations and trouble shooting. Expert systems have been also suggested for various power systems applications [Fouad and Venkataraman, 1991]. Typical applications include reactive power and voltage control, unit commitment, load forecast, power system trouble analysis, contingency screening, etc. Expert Systems is rarely applied to transmission planning problem [Francisco D., Galiana, Donald, Mcgillis, and Miguel, 1992], [Teive, Silva and Fonseca, 1998]. Teive, Silva and Fonseca [1998] apply an expert system to a realistic system (500 bus). The solutions were rapidly obtained.

2.3.1.5 Hybrids

Due to the complexities of real-world problems and the pros and cons of various search techniques, it is apparent that hybridization is a way forward to develop more powerful algorithms. Hybridization allows search techniques that display particular properties to be produced. The development of hybrids and their theoretical underpinning is a new area of research that is just starting to be explored. Some of the emerged hybrid techniques [Song, 1999] integrate: SA and TS, SA and GA, GA and local search, fuzzy logic and GA.

Gallego, Monticelli, and Romero [1998] propose a hybrid algorithm based on TS algorithms which incorporate efficient features of SA and GA algorithms. Simulation results confirm the claims made in the literature regarding the superiority of this approach in dealing with complex, large scale combinatorial problems.

2.3.1.6 Ant Colony Optimisation

Ant Colony Search (ACS) methods are artificial systems that take inspiration from the behavior of real ant colonies and which are used to solve function or stochastic combinatorial optimisation problems. These are population based, co-operative search algorithms. The first ACS system was the Ant System (AS), proposed by Dorigo in his Ph.D. thesis (1992). Currently, most work has been done in the direction of applying ACS to combinatorial optimisation. This was first proposed for tackling the well known TSP problem, but has been also successfully applied to problems such as the Quadratic Assignment Problem (QAP), Job-shop Scheduling Problem (JSP), vehicle routing and graph coloring. ACS has been also applied successfully to solve the problem of economic dispatch.

Ant Colony Search algorithms [Dorigo, Maniezzo, and Colorni, 1995], to some extent, mimic the behavior of real ants. In ACS, the colony consists of many homogeneous artificial ants communicating among themselves by recruit pheromone. The ants change their behavior according to the situation. Firstly, they walk randomly to search for a food source, operating in a discrete-time environment. They will not be completely blind, a decision is made on the intensity of trail perceived and the visibility. Each ant will have also some memory about its location and the next possible move. According to the objective function, their performance will be weighted as a fitness value, which directs influence to the level of trail quantity deposited in the path selected by ants. Each ant's decision to choose the next node to move to depends on two parameters: the visibility of the node and the trail intensity previously laid by other ants.

The main characteristics of this model are:

- positive feedback,
- distributed computation,
- and the use of a constructive greedy heuristic.

Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence, and the greedy heuristic helps find acceptable solutions in the early stages of the search.

2.3.2 Review of Transmission Planning Techniques

There are several planning algorithms available for the solution of the long-term problem; each based on a different interpretation of the system model and the choice of the design objective. These methods are classified into two main approaches:

- heuristic methods,
- mathematical optimisation models.

Most commercial programs combine both approaches [Santos, Franca and Said, 1989].

Several classical optimisation techniques have been used in an attempt to solve the transmission-planning problem. These include linear programming [Chanda.and Bhattacharjee, 1994], non-linear programming [Gilles, 1986], [Youssef and Hackam, 1989], mixed integer programming [Gilles 1986], [Santos, Franca and Said, 1989], Benders decomposition [Pereira, Pinto, Cunha, Oliveira, 1985], [Romero and Monticelli, 1994] and others [Rudnick et al., 1996], [Monticelli, Santos, Pereira, Cunha, Parker and

Praca, 1982], [Latorre-Bayona and Perez-Arriaga, 1994]. Heuristic models have also been utilized, based on sensitivity analysis [Pereira, Pinto, Cunha, Oliveira, 1985], [Monticelli, Santos, Pereira, Cunha, Parker and Praca, 1982], [Latorre-Bayona and Perez-Arriaga, 1994].

Practice has shown that conventional optimisation methods are unable to produce optimal solutions for large networks [Romero Gallego and Monticelli, 1996], [Gallego, Alves and Monticelli, 1997]. As an alternative to conventional optimisation methods, various heuristic search algorithms rooted in natural and physical processes have been applied: simulated annealing, genetic algorithms, and evolutionary programming among others. Expert systems have also been applied to the transmission-planning problem.

2.3.2.1 Linear Programming

The main disadvantage of this method is that all the objective functions and constraints must be linearised, which often leads to sub optimal solutions of the original non-linear problem. Several methods have been introduced to handle transmission expansion planning. Linear (or linearised) cost functions were adopted in most of these models [Garver, 1970], [Kaltenbach, Peschon and Gehrig, 1970], [Puntel, Reppen, Ringlee, Platts, Ryan and Sullivan, 1973], [Sawey and Zinn, 1977]. This assumption often led to inaccurate results producing near but not optimal solutions to the original problem.

Levi and Calovic [1993] suggest a linear-programming based decomposition method. Only investment cost optimisation is addressed in this paper.

In [Levi and Calovic, 1993], overall transmission network expansion planning is separated into two independent problems of investments and operations. Within the framework of this separation, a new methodology for optimal investment is proposed. It is specified as a minimum cost linear programming model and is solved by further division into two sub-problems. The first considers the maximum use of the existing network facilities for future system needs and the choice of the best candidates for network reinforcements to be applied in an interactive expansion planning process. The second sub-problem considers the prospective expanding network and decides on the economically best reinforcements, enabling the optimum-based expansion planning policy. The objective function considered is to minimize the investment cost while satisfying the following constraints: load balance equation, minimum and maximum generation limits, and transmission limits of existing network elements. The (n-1) security is also taken into account in the model. This methodology is applied to the Garver six-node system and the eastern part of Yugoslavia (61 nodes and 72 branches). Although this methodology produced good results compared to the previous work [Levi and Calovic, 1991], the computational time is still considerable due to the need to solve a highly constrained linear programming problem.

2.3.2.2 Non-Linear Programming

In this paper [Youssef and Hackam, 1989], the authors introduced a model that deals with both static and dynamic modes of transmission planning. An accurate and continuous nonlinear cost function (with a dependence on time and line ratings) for the system is formulated. It includes both the fixed and the variable cost for all planned lines, in addition to the cost of the power losses. The objective function is minimized subject to demand satisfaction, overloading, and security constraints on bus voltage magnitudes and angles. The AC load flow was considered for the modeling of the network. This method was applied successfully to the Garver six-bus system [Garver, 1970]. This test system is often used as a test bed for a new methodology but it is considered as a small system. The main limitation of this method is the replication of the same line type in the corresponding wayleave.

2.3.2.3 Mixed Integer Programming

Mixed Integer Programming (MIP) [Gilles, 1986], [Santos, Franca and Said, 1989] is based on at least two different algorithms: the first of which determines the integer variables and the second (usually a linear program) solves the remaining continuous problem and provides for a new run of the first algorithm. The integer variables are usually determined by some enumeration algorithm [Romero and Monticelli, 1994] such as branch and bound, which searches for feasible states. Though theoretically guaranteed to find the optimal solution, the MIP program usually cannot be allowed to complete the full search because of computation time limitations, hence this techniques is limited to small systems.

2.3.2.4 Branch and Bound

The Branch and Bound method [Romero and Monticelli, 1994], [Lee, Hicks and Hnyilicza, 1974] is a powerful enumeration strategy that helps to reduce the number of combinations of integer variables considered in a mixed integer non linear programming problem. The advantage of the Branch and Bound technique is that it can provide a sequence of solutions with estimates of their sub-optimality. The Branch and Bound method suffers from the curse of dimensionality problem.

2.3.2.5 Benders Decomposition

The most successful approach to solve the transmission planning is based on Benders Decomposition (BD) [Benders, 1962], [Pereira, 1985], [Granvill and Alli, 1988].

This method decomposes the original problem into two sub-problems: the investment sub-problem (that proposes a trial expansion plan) and the operational sub-problem (that analyzes the performance of each trial plan).

The operational sub-problem (linear programming problem) expresses the operational limit violations in terms of discrete investment variables (Benders cuts). These linear constraints are added to the investment sub-problem and new iterations are repeated. The main difficulty of this approach is due to the high computational time when solving medium-large discrete investment sub-problems to optimality. Since the Benders method generally requires many iterations to reach convergence, the main effort is due to the repeated solution of the investment sub-problem. Another difficulty pertains to the non-convexity of the problem [Granvill and Alli, 1988], which can lead to the exclusion of feasible solutions when solved by the decomposition method. Hierarchical decomposition methods try to overcome this shortcoming. Benders decomposition can also be extended to solve non-linear problems [Geoffrion, 1972].

[Tsamasphyrou, Renaud and Carpentier, 1999] proposed an efficient method based on Benders Decomposition. This method was applied on a reduced model of the French network (385 lines, 194 nodes). The authors claim that the decomposition scheme used has substantially reduced the computational time.

2.3.2.6 Hierarchical Decomposition

Hierarchical decomposition methods are generally based on Benders decomposition. These methods try to overcome the following shortcoming: the non-convexity that can leads to the exclusion of feasible solutions when solved by decomposition. The Hierarchical decomposition methods achieves that by initially solving the operation sub-problem using a simple network flow program and after some Bender iterations switches to a more adequate formulation (the linearized power flow) [Romero and Monticelli, 1994]. Nevertheless, repeated solutions of integer problems are still required.

Successful approaches using hierarchical Benders decomposition incur a high computational cost, mainly due to the need to solve a large integer investment sub-problem for every Benders iteration.

[Oliveira, Costa, Binato, 1995] proposed a combined optimisation/heuristic method using a customized decomposition approach in an attempt to solve real world problems, taking advantage of the operational sub-problem structure and reducing the computational effort required to solve the non-structured investment sub-problem. This approach allows the use of heuristics during the iterative process. A general branch and bound algorithm was used so as to obtain a feasible integer solution for each investment sub-problem. The objective function considered is to minimize investment cost and load shedding subject to generation and transmission network constraints. The proposed method was applied to the reduced Southern Brazilian network (79 bus and 155 circuits), and for this case the computing time was about 3 minutes. The results presented in this case study indicate that this approach can solve difficult expansion problems with a large number of alternatives with moderate computational time.

2.4 Conclusions

This chapter presented a detailed description of the transmission-planning problem, namely, its formulation and methodologies. Various optimisation techniques, which can be used to optimise the network design, have also been reviewed. These have included conventional and heuristic techniques.

Chapter 3

Genetic Algorithms and Other Heuristics

3.1 Introduction

There has been considerable interest in Genetic Algorithms, and other heuristic search algorithms (such as Ant Colony Search) rooted in natural and physical processes, among researchers in various application fields. The simplicity, flexibility and robustness of such algorithms has opened up new areas of application, and has also encouraged a re-appraisal of some traditional problems that were either very difficult or even intractable for traditional optimisation techniques.

This chapter presents a brief review about GAs, their backgrounds and the basic operation steps involved. A particular variant of GA, known as the Deterministic Crowding Genetic Algorithm, is explored. Some of the successful applications of GA in various areas including power systems are reported. This chapter also considers a novel optimisation technique, Ant Colony Search (ACS), as an alternative technique to solve the transmission-planning problem. An overview of ACS, background and search algorithm steps and areas of application, is presented. Some of the advantages and disadvantages of both optimisation techniques, for the purposes of computational optimisation, are also discussed. Finally, a comparison is made between the two methods.

3.2 Genetic Algorithms

3.2.1 Background

Genetic Algorithms [Holland, 1975], [Davis, 1991], [Goldberg, 1989] belong to a class of evolutionary computation techniques based on models of biological evolution. The basic theory of GA can be found in [Holland, 1975], [Goldberg, 1989] with extensions to the theory in various GA conferences [ICGA, 1985, 1987, 19891991, 1993, 1995, etc.], [IEE/IEEE GALISIA, 1995, 1997] and other evolutionary computation related conference publications and journals. GAs have proved to be effective for integer and non-convex problems. They have proved useful in domains that are not well understood, or for search spaces which are too large to be efficiently searched by standard methods. Since their introduction GAs have proved to be a great success, with many researchers and practitioners adopting them to solve a wide variety of problems.

The distinguishing feature of a GA from other function optimisers is that the search algorithm proceeds *not* by incremental changes to a single structure but by *maintaining a population of structures* from which new structures are created. These structures individually and in combination with other members of the population contain information about the various sub-structures making up the optimal solution.

The basic power of a GA arises from the concept of implicit parallelism [Holland, 1975], [Davis, 1991], the simultaneous allocation of trials to many regions of the search space. This theory suggests that through the repeated process of selection, crossover and mutation, the schemata (building blocks) of competing hyper-planes decrease or increase their presence in the population according to the relative fitness of those strings.

3.2.2 Standard Genetic Algorithm

A typical Genetic Algorithm can be described as follows:

1. Select (at random) an initial population of strings $\{s^0_1, \ldots s^0_k, \ldots s^0_p\}$, where P is the number of population members.

2. Evaluate the cost $C^i_k = C(s^i_k)$ for all k.
3. Perform random crossovers, introducing new population members.
4. Perform random mutations.
5. Select a subset of population members to be deleted; the selection should depend on the cost of each population member, so that high-cost strings are more likely to be deleted.
6. Re-iterate from 2. until no improvement in the best member of the population has occurred for some iterations or until a set number of iterations.

Crossover is the main genetic operator that allows information to be exchanged between individuals in the population. The use of crossover is the defining characteristic of a GA and has a powerful influence on the speed of convergence of the population. Crossover is not performed on every pair of individuals, its frequency being controlled by a crossover probability. There are various alternative techniques such as single-point, multi-point and uniform crossover. Uniform crossover is now generally regarded as the best option.

The main reason for using the mutation operator is to prevent the permanent loss of any particular bit values (genes), as without mutation there is no possibility of re-introducing a bit value that is missing from the population. It is usually applied to each new structure individually. A given mutation consists of randomly altering each gene with a small probability. The mutation rate must be kept low since a high value tends to make the algorithm behave like a random search strategy.

An important component of the GA algorithm is the choice of the subset of population members to be deleted (or conversely the selection of the individuals to survive into the next generation). This process is called 'selection' and variety of techniques is possible. All selection techniques allocate some proportion of the fitness (function of cost) of an individual relative to the population average fitness. These selection methods include roulette wheel [Goldberg, 1989], [Davis, 1991], and tournament selection [Goldberg], [Deb] among others [baker]. In tournament selection some (usually small) subset of the population is chosen (usually at random) and the higher cost members (less fit) are deleted. In roulette wheel selection, the fitness determines the likelihood of an individual being selected for survival. By analogy with a physical roulette wheel, a random experiment is performed with the probability of selecting any individual being proportional to its fitness, which is inversely related to its cost. The mapping from cost value to fitness value, known as fitness scaling, is designed to give a *reasonable* likelihood of survival for any string even in circumstances where cost function values can exhibit extremely high, or extremely low numerical variation. An advantage of tournament selection is that fitness scaling is not needed. Often 'elitism' is introduced in the selection mechanism, whereby the best individual or the best few individuals are automatically retained, so that each generation does at least include the individual which is 'best so far'.

3.2.3 Genetic Algorithm models

Although the main operators that influence the GA performance are only three, i.e. selection, crossover and mutation, their interaction is highly complex and slight variations in their implementation result in a variety of models. The different models depend on factors such as:

- selection method and mechanism,

- parent replacement method,
- crossover and mutation method,
- problem to be solved, whether a single or multiple objective formulation is required.

The GA model to be applied is chosen after a careful analysis of the problem to be solved. A particular model known as the Deterministic Crowding Genetic Algorithms (DCGA), which represent a further enhancement over the standard GA, is adopted for the solution of the optimum transmission design and is considered in this chapter.

3.2.4 Deterministic Crowding Genetic Algorithms

An area of further enhancement to the standard GA has been the introduction of niche methods [Goldberg, 1989], [Mahfoud,1992], [De Jong, 1975]. These methods reduce competition among population members when there is a sufficiently large difference (or distance) between them, allowing sub-populations centring on good solutions to co-exist. Niche methods fall into two broad categories: 'crowding' and 'sharing'. Crowding methods restrict the replacement of individuals by discouraging competition among widely differing individuals, while sharing methods de-rate an individual's effective fitness when similar individuals co-exist. The potential disadvantage of most niche-based methods is the computational burden of comparing each individual to many other individuals, in order to evaluate the similarity measure.

Niche GA methods are inspired by a corresponding ecological phenomenon, where similar members of a natural population compete for the same resources. A niche GA attempts to maintain a population of diverse individuals in the course of the simulated evolution. Genetic Algorithms that incorporate these ideas are thus better able to locate multiple local-optimal solutions within a single population. The particular niche method described here is based on the crowding concept.

In a crowding GA the new population is created by allowing child strings to replace the parents that are most similar to themselves. De Jong proposed a crowding factor model where only a fraction of the population reproduces and dies in each generation, with each newly created population member replacing an existing member, preferably the most similar. Following analysis and modification of the De Jong's (and other) other crowding methods, Mahfoud proposed the Deterministic Crowding Genetic Algorithm (DCGA). Mahfoud's model is computationally efficient, since each offspring is only compared with its two parents, and competes only with the more similar parent. The DCGA provides selection pressure *within* but not *across* regions of the search space, leaving the search across regions to the crossover operator.

The DCGA randomly pairs all population members in each generation to yield P/2 pairs of parents, for a population of size P, each pair undergoes crossover, possibly followed by mutation, to produce two offspring. Each of the two offspring competes with one of the two parents, chosen according to a similarity measure. The fitter among them forms the population of the next generation. For example, given a pair of parents and their two offspring, two sets of parent-offspring tournaments are possible:

set1

- parent 1 against child 1
- parent 2 against child 2

set 2

- parent 1 against child 2
- parent 2 against child 1

The set of tournaments which forces competition among more similar individuals is held, where similarity is defined as the average distance between the parent-child combinations in the set.

The DCGA can be summarised as follows:

1. Select (at random) an initial population of strings $\{s^0_1,\ldots s^0_k, \ldots s^0_P\}$, where P is the number of population members.
2. Evaluate the cost $C^i_k = C(s^i_k)$ for all k.
3. While $(I \leq P)$
4. Choose two parents, par_1 and par_2 at random, without replacement.
5. Perform crossover and mutation to produce offspring ch_1 and ch_2.
6. Evaluate fitness (f) of parents and offspring.
7. If (distance (par_1, ch_1) + distance (par_2, ch_2)) $\leq$ (distance (par_1, ch_2) + distance (par_2, ch_1)) then

   IF ($f(ch_1) > f(par_1)$) replace par_1 by ch_1

   IF ($f(ch_2) > f(par_2)$)replace par_2 by ch_2

   Else

   IF ($f(ch_2) > f(par_1)$)replace par_1 by ch_2

   IF ($f(ch_1) > f(par_2)$) replace par_2 by ch_1

Endif

8 Next i

9 Re-iterate from 2. Until no improvement in the best member of the population has occurred for some iterations or until a set number of iterations.

3.2.5 Characteristics of the Deterministic Genetic Algorithm

The main proprieties that distinguish DCGA from other GA models are:

- selection and population replacement are combined into a single step,
- random crossover is performed on the whole population,
- there is no need for any scaling mechanism,
- embedded elitism (only inferior parents are replaced by children),
- there are fewer control parameters to be set.

The major components of the DCGA model that can be explicitly varied are the parent replacement-selection strategy, crossover method (mutation is optional in DCGA), population size and the number of generations required for the evolution.

Uniform crossover (now generally considered as the best option) is usually applied in DCGA model.

3.2.5.1 Uniform Crossover

Taking as an example two parents par_0 and par_1 with string length 12 and using binary symbols:

par_0 = 1 0 1 1 0 0 0 1 0 1 1 0

par_1 = 0 0 1 0 1 0 1 1 1 0 1 0

Define a (random) 'mask string'

mask = 1 0 1 1 0 0 0 1 0 1 1 0

Then, according to uniform crossover, a child can be defined as containing the corresponding symbol from parent 0 for every bit position in the mask with value 0, and containing the corresponding symbol from parent 1 for every bit position in the mask with value 1, giving:

Ch1 = 0 0 1 0 0 0 0 1 0 0 1 0

A second child can also be obtained by using the bit-wise inverse of the same mask:

Ch2 = 1 0 1 1 1 0 1 1 1 1 1 0

The conventional definition of crossover probability [Spears W.M, De Jong K.A, 1991] is the probability of a mask bit being given a value of 1. With this definition, a probability of 0.5 means that a child has an equal likelihood of inheriting bit values from either parent, a probability of 1 means that child 1 is identical to parent 1, etc.

A modified definition of uniform crossover, first proposed in this thesis, is adopted in the DCGA model and will be considered here. This can be defined by following simple rules:

1. Randomly choose the first bit of the mask string as either 0 or 1, with equal probability.
2. Randomly choose each consecutive bit of mask string with a given probability of being the same as the preceding bit.

Using this definition, a crossover probability of 0.5 (which corresponds to uniform crossover) gives statistical properties which are identical to those of the previous definition, but other crossover probability values bias the 'expected run length' of consecutive 1s or 0s in the mask.

3.2.6 Applications of Genetic Algorithms

The robustness of Genetic Algorithms has enabled them to be applied in a wide range of problem solving areas. GAs are capable of finding global optima for mathematical problems having a multiplicity of local optima and hard non-convexities. Highly complex problems such as the travelling salesman problem, turbine design and pipe line scheduling have been solved successfully. [Goldberg, Millemen and Tidd], References [ICGA conferences, 1983-1999], [IEE/IEEE GALISIA conferences, 1995- 1997] contain a wide range of applications of GA and other evolutionary techniques. Broadly classified, the GA has found widespread application in areas including:

- computer aided design in all engineering branches,
- pattern recognition and image processing,
- artificial intelligence, machine learning and robotics,
- power telecommunication network optimisation,
- biotechnology and medical systems,
- chemical process optimisation,
- production planning and scheduling,
- neural network optimisation,
- non-linear optimisation.

Recently GAs have been successfully applied to various areas of power system. Some of these applications include:

- unit commitment [Dasgupta and McGregor], [Shebble and Maifeld], [Orero and Irving, 1997], etc.,
- economic despatch [Walters., and Shebble.], [Chen., and Chang.], etc.,

- distribution system planning [Nara et al.], [Yeh et al.], [Miranda et al.], [Wen et al.],
- harmonic analysis in distribution network [Boone and Chiang], [Lee et al.],
- reactive power optimisation and voltage scheduling [Iba], [Lee and Park],
- load flow solution [Yin and Germay],
- network partitioning [Taylor et al.], [Ding et al.], [Orero and Irving],
- load forecasting [Maifeld and Sheble],
- power stability and frequency control [Finch and Besmi], [Lansbery et al.],
- hydro co-ordination [Hulselman et al.],
- power transmission planning [Gallego et al.], [Rudnick et al.].

3.3 Ant Colony Search

3.3.1 Background

Ant Colony Search (ACS) studies artificial systems that take inspiration from the behaviour of real ant colonies and which are used to solve functional or stochastic combinatorial optimisation problems.

It is a population-based approach that uses exploitation of positive feedback as well as greedy search. The main characteristics of this model are positive feedback, distributed computation, and the use of a constructive greedy heuristic. Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence, and the greedy heuristic helps find acceptable solutions in the early stages of the search.

The first ACS system was the Ant System (AS), proposed by Dorigo in his Ph.D. thesis (1992). It was first proposed for tackling the well known TSP problem [Dorigo et al., 1996, 1997], but has been also successfully applied to problems such as Quadratic

Assignment Problem (QAP) [Gamberdella et al., 1997], [Maniezzo et al., 1994], Job-shop Scheduling Problem (JSP) [Dorigo et al., 1996], vehicle routing and graph colouring [Bullnheimer, 1997].

More recently Dorigo and Gamardella have been working on various extended versions of the Ant System paradigm. Ant-Q is an hybridisation of AS with Q-learning, a well known reinforcement learning algorithm. Ant Colony System (ACS) [Dorigo et al., 1997] is a further extension of Ant-Q. Both have been applied to the symmetric and asymmetric travelling salesman problem [Dorigo et al., 1997], [Dorigo and Gamberdella, 1995, 1996].

Schoonderwoerd et al. [1997] have developed an ant colony algorithm called ABC, for routing and load balancing in circuit switched telecommunications networks, and Di Caro and Dorigo [1997] for routing in packet switched telecommunications networks.

Stutzle and Hoos [1997] have been working on various extensions on Ant System. Bilchev has recently developed an ACO method for the optimisation of continuous functions [Bilchev and Parmee, 1996].

3.3.2 Ant Colony Search Algorithm

ACS [Dorigo et al., 1996], [Dorigo, 1992] is a population based, co-operative search algorithm inspired by the behaviour of real ants. A colony of ants is able to succeed (for instance to find the shortest path between the nest and the food source) whereas a single ant would probably fail, especially as ants are almost blind. It was found that ants leave a trail of pheromone when they move. This pheromone trail can be observed by other ants and motivates them to follow the path, i.e. a randomly moving ant will follow the pheromone trail with high probability. The trail is then reinforced and more ants follow that trail.

One of the basic ideas of ACS is to use the counterpart of the pheromone trail used by real ants as a medium for co-operation and communication among a colony of (artificial) ants. The artificial ants are simple agents that have some basic capabilities.

The artificial ants of Ant System behave in a similar way. They differ from their natural counterparts in two aspects. They are not blind, i.e. they have information regarding their environment and they use this information to be greedy in addition to being adaptive. Second, they have a memory, which is necessary to ensure that only feasible solutions are generated [Bullnheimer, 1997].

To apply the ACS algorithm to a problem requires defining the following:

1. an appropriate graph representation of the problem for search by a number of agents;
2. the autocatalytic (i.e. positive) feedback process;
3. the heuristic that allows a constructive definition of the solutions (greedy force);
4. and the constraint satisfaction method.

A typical ACS algorithm can be stated as follows:

1. Initialise A (t): Select (at random) an initial population of the colony.
2. Evaluate A (t): Evaluate the fitness of all ants based on the problem objective function.
3. Deposit-trail: pheromone trail quantity is deposited into the particular nodes (of problem graph) selected by the ants according to equation 3.1.
4. Send-ants A(t): Each ant chooses the next node to move to taking into account two parameters: the visibility of the node and the pheromone intensity previously laid by

other ants. The decision for an ant k, to move to a node, is made on the basis of the probability p_{ij}^{k} defined by equation 3.3.

Note: nodes are labelling by their location (i, j) in a two dimensional space.

For one iteration of the ACS algorithm, m moves are carried out respectively by the m ants in the interval $(t, t+1)$, then each ant will complete a path (*cycle*) every n (number of nodes in a path) iterations of the algorithm. At this point $(t = t+n)$ the trail intensity is updated according to equation 3.1.

5. Re-iterate from 2. Until no improvement in the best member of the population has occurred for some iterations (cycles) or until a set number of iterations is satisfied.

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \qquad (3.1)$$

where

$\tau_{ij}(t)$ is the intensity of trail on node (i,j) at time t,

$\Delta\tau_{ij}$ is the quantity per unit length of pheromone laid on node (i,j) by the ants between step t and $(t + n)$ and is defined in equation 3.2,

ρ is the persistence of the trail, thus $(1-\rho)$ simulates the evaporation,

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \qquad (3.2)$$

where

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/F_k & \text{if } node(i,j) \in \text{path used by the kth ant} \\ 0 & \text{otherwise} \end{cases}$$

F_k is the cost value of the k^{th} ant (low values are better than high values),

Q is a constant quantity per unit length of pheromone laid by the ant,

m is the number of ants.

$$P_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{k\varepsilon\, allowed_k}[\tau_{ik}(t)]^{\alpha}.[\eta_{ik}]^{\beta}} & if\ j\varepsilon\ allowed_k \\ 0 \ \ otherwise \end{cases} \quad (3.3)$$

where

η_{ij} is the visibility (relative local cost) of the node,

α, β are the heuristically defined parameters,

$allowed_k$ is the list of possible moves of the *kth* ant.

The standard ACS algorithm can be refined by introducing some sort of elitism. This can be accomplished by retaining automatically the best ant or the best few ants so that each ant colony does at least include the ant which is 'best so far'. The concept of keeping the best ant(s) takes advantage of experience and knowledge of elitism in GA. As a result more emphasis is put on the best path which helps directing the search towards the optimum solution and improving the performance of the standard ACS algorithm.

The most important part in ACS algorithms is the treatment of the trail intensities. In practice, the long term effect of the trail intensities is to reduce the size of the effective search space by concentrating the search on a relatively small subset of the initial space. Different choices about how to compute the incremental trail and when to update the trail cause different instantiations of the ACS algorithms [Colorni et al., 1996].

3.3.3 Ant Colony Search Applications

ACS algorithms [Dorigo et al., 1996] are new emerging techniques that have been proposed as powerful tools to solve some order based problems such as Travelling Salesmen Problem (TSP) and Quadratic Assignment Problem (QAP). Currently most work has been done in the direction of applying ACS to combinatorial optimisation problems. They have been applied to the Asymmetric Travelling Salesmen Problem (ATSP), Job-shop Scheduling Problem (JSP) and vehicle routing and graph colouring. Other ACS applications include:

- Adaptive Routing in Communication Networks,
- Load Balancing in Telecommunications Networks,
- Economic Dispatch Problem [Chou and Song, 1997].

3.4 Advantages of GA and ACS

Some of the advantages of GA and ACS, for the purposes of computational optimisation, are highlighted in the following sections.

3.4.1 Global Optimisation

Many practical optimisation problems contain multiple local optima. An important advantage of GA (or ACS) is the possibility that more than local optimum will be explored and there is a chance that GA (or ACS) *may* discover a global optimal solution. This property is shared with many heuristic search techniques (such as SA, TS, etc.) where as mathematical programming techniques are generally equipped to seek only a local optimum.

3.4.2 Generality of Objective Functions

The only restriction on the type of objective function that can be accommodated in a GA (or ACS) is simply that it must be a computable scalar function of any feasible candidate solution. This is in contrast to most mathematical programming based techniques, which may impose quite severe restrictions (such as linearity, differentiability, continuity, convexity, etc.) on the type of objective function which can be accepted.

3.4.3 Relative Ease of Programming

GA (ACS) are relatively straightforward to program, in contrast with mathematical programming techniques which generally require sophisticated linear algebra routines, calculation of partial derivatives, robust line searching methods, etc.

3.4.4 Algorithm Flexibility

There are many possible variants to GA (ACS) some of which will be more efficient for particular problems than others. Further modelling of a problem can be readily included in the objective function with relative ease.

3.4.5 Numerical Robustness

The simplicity of GAs (ACS) leads to a further advantage. They do not suffer from the numerical robustness problems, which can occur in some mathematical optimisation techniques (e.g. matrices becoming ill-conditioned or singular, iteration steps becoming too short prematurely, etc.).

3.5 Disadvantage of GAs and ACS

Despite the considerable advantages of GAs and ACS there are some disadvantages that must be considered:

3.5.1 Potential Premature Convergence

In GAs, crossover and selection tend to cause unstable reproduction of schemas associated with above-average fitness, which can result in rapid loss of diversity in the population and premature convergence to a sub-optimal solution. For ACS, as already mentioned, the parameters α and β control the relative importance of pheromone trail (representing global information) versus visibility (representing local information). An appropriate trade-off between those two measures should be adopted to avoid premature convergence to a sub-optimal solution.

3.5.2 Long Computing Times

The number of candidates considered in a GA (ACS) population and the number of generations (cycles) which must be simulated before convergence is obtained, can combine to produce excessive run times for problems in which the fitness calculation is non-trivial.

3.5.3 Parameter Tuning

Many GA (ACS) variants include a variety of parameters that can be adjusted by the user (e.g., for GA, mutation and crossover probabilities, population size, selection mechanisms, etc.; for ACS, the parameters that control exploration versus exploitation, ant colony size, selection mechanism, etc.). In some ways the ability to 'tune' these parameters can be seen as an advantage, because the analyst can adjust the GA (ACS) to suite his particular class of problems. However, since this tuning is generally based on extensive numerical experiments and can require considerable effort, methods that require less tuning are often preferred.

3.5.4 Modelling Constrained Problems

GAs (ACS) are probably better suited to unconstrained optimisation problems than to constrained problems. In some cases a special coding technique can be defined so that any candidate string is guaranteed to satisfy all (or some of) the constraints. This is a very advantageous approach, when it is available. In other cases penalty factors can be applied to convert a constrained problem into an unconstrained problem. The simplistic approach, whereby any candidate is checked against the constraint set and is rejected outright if found to be infeasible, can be very inefficient if a large proportion of the candidates that are generated prove to be infeasible.

3.5.5 Non-deterministic Solutions

Most mathematical programming techniques provide additional information (such as gradient values, approximate Hessian matrix values, duality- or complimentary-gap values, etc.) which can provide some reassurance that a good solution has been obtained. The results obtained by GAs (ACS), or by other similar probabilistic search methods, do not incorporate this type of subsidiary information. This can be an important drawback for significant technical and commercial applications where a high level of assurance is required. In some respects it can be regarded as an advantage that a better solution might be obtained by GAs (ACS) simply by allowing further evolution, greater population sizes etc., but the difficulty is that for non-trivial problems some uncertainty always remains as to the quality of the final solution produced.

3.6 A Comparison between Genetic Algorithms and Ant Colony Search Algorithms

There are many similarities between ACS and GA techniques. Both techniques require coding the problem parameters in order to apply them. They both search from a population of points and use payoff (objective function) information in the pursuit of the optimum. Both approaches also use probabilistic transition rules, not deterministic rules to solve the problem in hand. In addition both approaches require appropriate parameter 'tuning' in order to converge.

However, both techniques differ in two main aspects. In ACS there is a communication between ants (synergy) through the pheromone trail, where ants can benefit from the experience of other ants. This is in contrast to GAs where every member is evaluated independently and without exchange of information among population members. In addition, the decision made by the ants takes into account global information (i.e. ants lay an amount of pheromone trail which is proportional to how good the solution produced was) as well as local information (the search is not directed by any measure of the final results achieved). Table 3.1 reports on the main similarities and differences between GAs and ACS.

Table 3.1 Comparison between GAs and ACS approaches

GAs	**ACS**
Need for problem representation (coding of the parameter set)	Need for problem representation (coding of the parameter set)
Search from a population of points	Search from a population of points
Iterative process	Iterative process
Need for parameter tuning	Need for parameter tuning
Use of payoff information	Use of payoff information
Use of probabilistic transition rules (through the use of crossover and mutation probabilities)	Use of probabilistic transition rules (through the decision of ants to move to another node)
-	Use of local information provided by the visibility concept
-	Communication among agents (synergy) through pheromone trail

3.7 Solution of Constrained Problems

Simple constraints, such as upper and lower limits on variables, can easily be represented implicitly within the problem coding. In some cases, it may also be possible to represent more complex constraints by an appropriate choice of coding. However, for general constraints, it is usual to represent constraints via penalty functions added to the cost function. An overall cost function, consisting of the original cost function plus additional penalty costs for any violated constraints, is then optimised. In general, penalty functions should be chosen to satisfy the following requirements:

- The penalty function should be progressive, so that a more severe violation of a constraint attracts a higher penalty cost- allowing the GA (ACS) to be guided towards feasibility.
- The penalty factors for each violated constraint should be summed to form the overall penalty cost, so that a well-behaved penalty cost surface is produced.
- The value of penalty costs should be higher than actual costs, so that a solution at the optimum of the overall cost function (including any penalty costs) would not include any non-zero penalties (i.e., violated constraints). If the penalty cost values are too small relative to the actual costs, it would be possible for the GA (ACS) to trade-off some constraint violations against cheaper actual costs, possibly arriving at an overall optimum solution which includes some violated constraints.
- On the other hand, the value of penalty costs should not be too high in relation to the actual costs, so that the overall cost surface (including penalty costs) is reasonably well-conditioned. In this respect it is frequently possible to determine a realistic penalty cost value in economic or physical terms. For example, in some problem formulations the maximum available level of a commodity may be represented as a constraint, whereas in reality some additional supply may actually be available (albeit with a higher cost), e.g. from a spot market, or by using contingency resources, etc. In such cases a suitable penalty cost value for the constraint is already available.

It is well known in conventional mathematical programming that the use of penalty costs can lead to an ill-conditioned overall problem. This arises from the need to satisfy both of the final two points mentioned above. By applying penalty values that are large enough to guarantee feasible solutions it is difficult to avoid creating severely distorted overall cost function surfaces, which are then problematic for a mathematical programming method. This drawback of the penalty factor approach is much less significant for GAs (ACS), and similar methods, since they do not impose any particular requirements on the mathematical properties of overall cost function surface. The fact that an overall cost function (including penalty costs) may not be differentiable or smooth, and may have disadvantageous curvature properties (e.g. having singular or ill-conditioned Hessian matrices at certain points) could not necessary impede the progress of GA (ACS).

3.8 Conclusions

This chapter reviewed two novel optimisation techniques, namely GA and ACS, which are proposed to solve the transmission-planning problem. A particular variant of GA, known as the Deterministic Crowding Genetic Algorithm, is explored. A variation of the uniform crossover operator is introduced. Some of the successful applications of both techniques in various areas including power systems are reported. This chapter also addressed some of the advantages and disadvantages of both techniques, for the purposes of computational optimisation. Finally, a comparison is made between the two methods.

Chapter 4

Algorithms Design and Testing

4.1 Introduction

In this chapter, the structure of the two proposed algorithms namely, the Ant Colony Search (ACS) and the Deterministic Crowding Genetic Algorithm, is illustrated. Then, the application of both optimisations algorithms to an artificial test problem (shortest path problem) is considered in an attempt to investigate their performance. Next, a comparative investigation of both algorithms is carried out. Then, the experimental results are analysed to examine the effectiveness of both algorithms as optimisation techniques. Finally, we conclude that both algorithms exhibit comparable performance which encourage us to apply them to the transmission-planning problem.

4.2 Artificial Test Problem

Given a rectangular grid of size (*nroute* x *ntype+1*), where *nroute* and *ntype* represent respectively the vertical and horizontal axis (figure 4.1) and assuming the grid elements are of equal length (e,g. unit length); the problem is to find the shortest path along the vertical axis. The solution to this artificial test problem is obviously known a priori; it is any straight path parallel to the vertical axis and its length is equal to *nroute-1*.

We have chosen this particular test because it resembles, to a certain extent, our transmission-planning problem. In this artificial problem *nroute* and *ntype* have no particular significance, except that they label the vertical and horizontal axes of the problem. (In the transmission planning problem, considered later, *nroute* will represent the number of routes and *ntype* the possible line types.) Moreover, we aim to investigate the effectiveness of ACS and GA algorithms in the search for the known optimum solution.

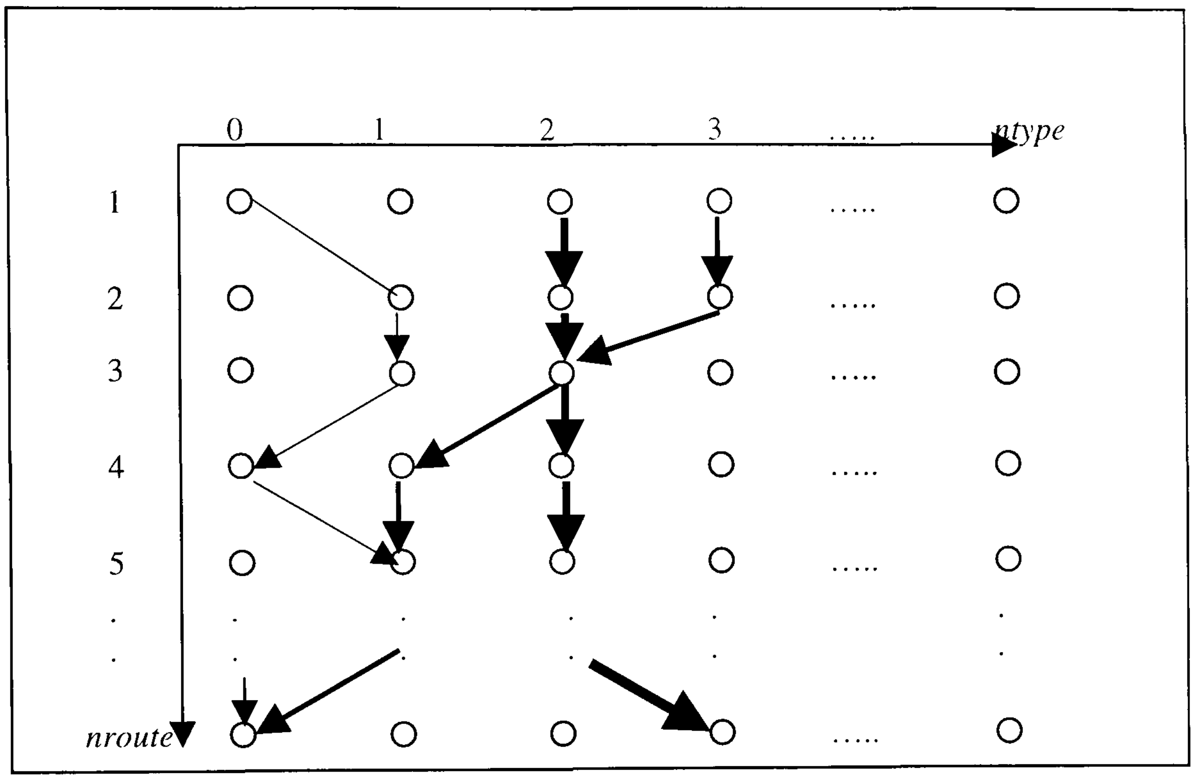


Figure 4.1 Problem Mapping: shortest path problem to (*nroute* x *ntype+1*) grid

Different paths (possible solutions)

4.3 Structure of Both Programs: Ant Colony Search and Deterministic Crowding Genetic Algorithms

The ACS & DCGA programs are implemented on a Pentium 233 MHz processor, under the Windows NT operating system using Fortran 77. Both programs are modular and can be applied to various problems with slight variation of the code to accommodate the new objective function.

4.3.1 Ant Colony Search Program

Given *ntype* types and *nroute* possible routes, *m* artificial ants much like those used in the TSP application [Dorigo, Maniezzo, and Colorni , 1995] are used to search for good solutions.

Every ant k is assigned a random solution in which every route i is randomly allocated a type j in the range {0…14}. Then, a fitness value, representing the objective function and penalties for constraint violations, is assigned to each ant. This fitness will influence the level of pheromone deposited (trail intensity). Each ant's decision to choose the next node to move to depends on two parameters: the visibility of the node and the trail intensity previously laid by other ants.

The selection of an assignment, equivalent to a move for an ant k, is made on the basis of the probability p_{ij}^{k} defined by equation (4.1). Therefore, the next node is determined by selecting a particular type j according to a specific selection scheme (e.g. roulette wheel selection) for a route i. Then a move value is associated with this assignment, which represents the change in the objective function value. Move values generally provide a fundamental basis for evaluating the quality of moves.

$$P_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{k\varepsilon\, allowed_k} [\tau_{ik}(t)]^{\alpha}.[\eta_{ik}]^{\beta}} & if\ j\varepsilon\ allowed_k \\ 0\ \ otherwise \end{cases} \tag{4.1}$$

where

η_{ij} is the visibility of the node

$\tau_{ij}\,(t)$ is the intensity of trail on node (i,j) at time t

α, β are the heuristically defined parameters to allow tuning of the weighting between visibility and trail

$allowed_k$ is the list of possible moves, in this case {0…14}

The visibility of the node is given by the following formula:

$$\eta_{ij} = 1/(\,cbase + \Delta c_i\,) \tag{4.2}$$

where

$$cbase = cMin \,/\, nroute$$

$$\Delta c_j = c_j - cMin$$

$$cMin = \min_{j=0,\, ntype} C_j$$

c_j are the values of node *j*.

For one iteration of the ACS algorithm, *m* moves are carried out respectively by the *m* ants in the interval *(t, t+1)*, then each ant will complete a path (*cycle*) every *nroute* iterations of the algorithm. At this point *(t = t+nroute)* the trail intensity is updated according to equation (4.3).

$$\tau_{ij}(t + nroute) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \qquad (4.3)$$

where

ρ is the persistence of the trail, thus (1-ρ) simulates the evaporation

$\Delta\tau_{ij}$ is the quantity per unit length of pheromone laid on node *(i,j)* by the ants between time *t* and *(t+nroute)*

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

and

$$\Delta\tau_{ij}^{k} = \begin{cases} Q / F_k & \text{if } node(i,j) \in \text{path used by the kth ant} \\ 0 & \text{otherwise} \end{cases}$$

F_k is the fitness (or value) of the k^{th} ant,

Q is a constant quantity per unit length of pheromone laid by the ant.

The aforementioned process is repeated for a certain number of cycles or until a satisfactory solution is found. This algorithm represents the standard ACS.

Finally, a further refinement over the standard ACS algorithm has been included. This was accomplished by keeping the best ant so far during simulation. The concept of keeping the best ant takes advantage of experience and knowledge of elitism in GA. As a result more emphasis is put on the best path (as will be explained later) which helps directing the search towards the optimum solution and improving the performance of the standard ACS algorithm.

To summarise, figure 4.2 illustrates the modified ACS algorithm as applied to the problem.

4.3.2 Deterministic Crowding Genetic Algorithm Program

The DCGA randomly pairs all population members in each generation to yield P/2 pairs of parents, for a population of size P. Each pair undergoes crossover, possibly followed by mutation, to produce two offspring. Each of the two offspring competes with one of the two parents, chosen according to a similarity measure. The fitter among them forms the population of the next generation. Figure 4.3 illustrates the DCGA algorithm.

The method of choosing the two parents without replacement is important, since it inherently gives the DCGA the property of elitism. Every population member (including the best-so-far) must enter a tournament as a parent, and can only be eliminated from the population if it is to be replaced by a fitter child.

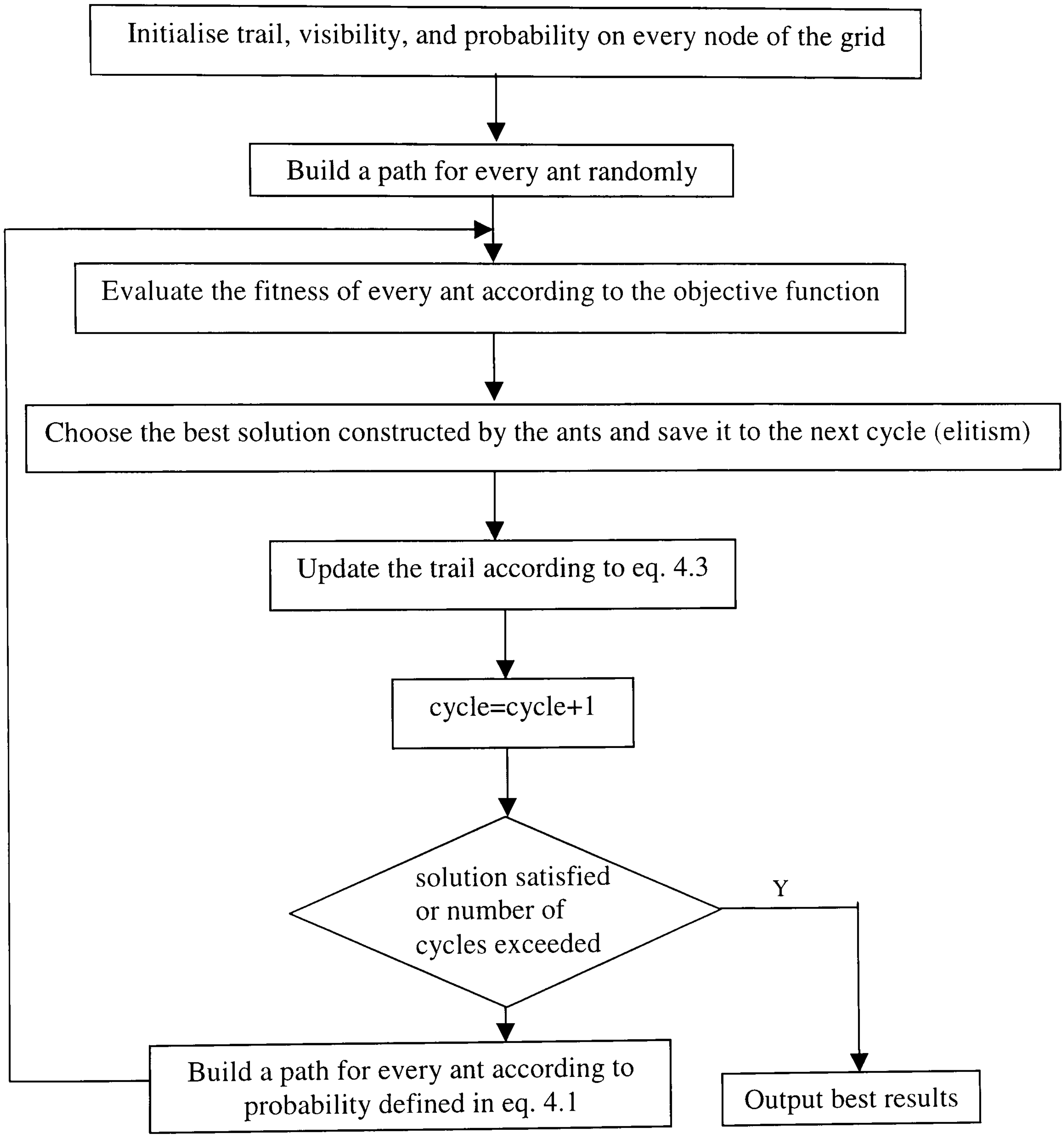


Figure 4.2 Modified ACS algorithm

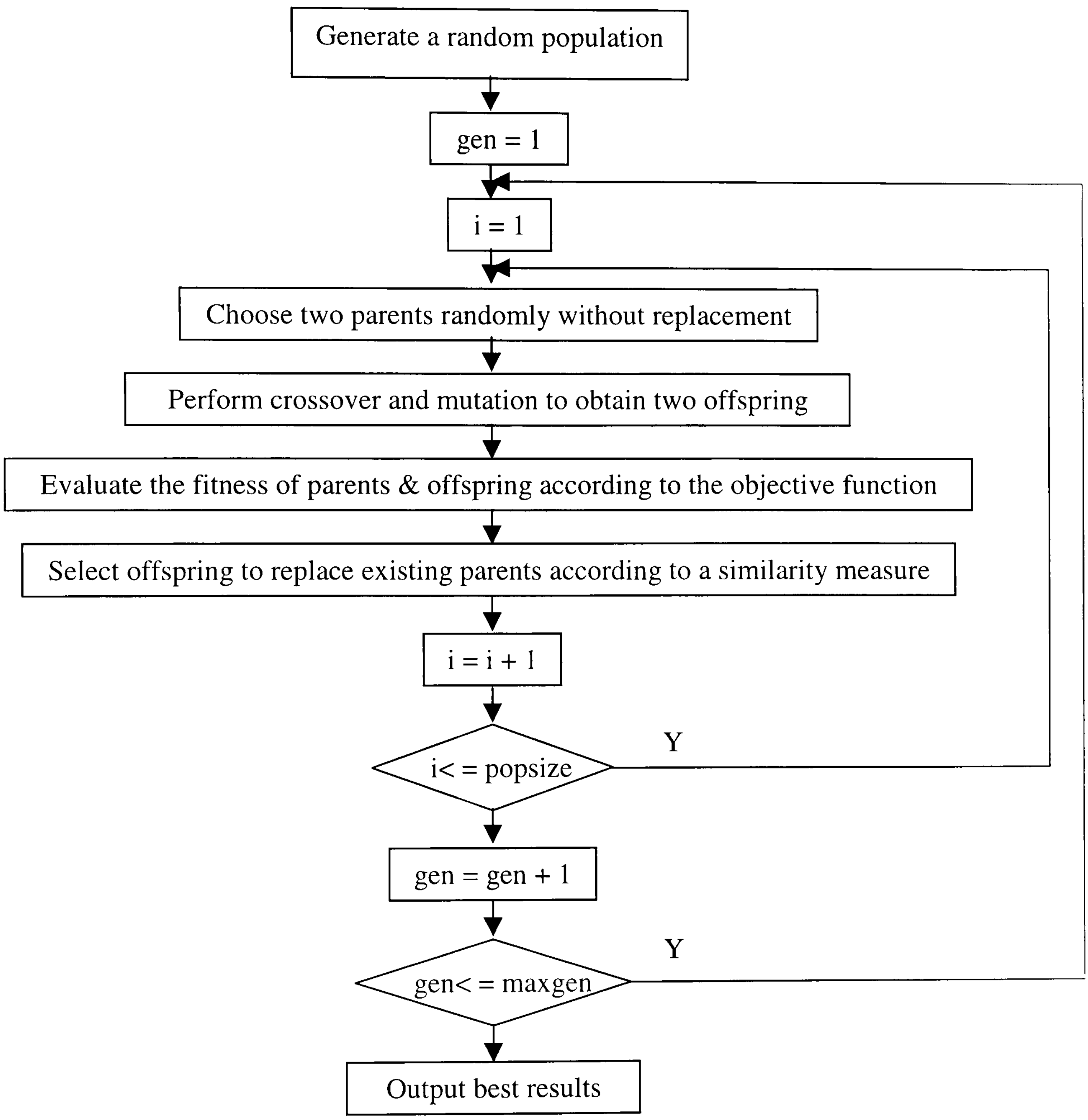


Figure 4.3 DCGA structure

4.4 Computational Results

4.4.1 Shortest Path Problem with Ant Colony search

Several tests have been carried out on artificial test problems. The parameters considered are those that affect directly or indirectly the computation of the probability (eq. 4.1). The aim is to investigate the effect of the different parameters on the performance of the ACS algorithm.

Table 4.1 illustrates some of the simulation results and the corresponding parameter settings of the standard ACS (without elitism).

A range of artificial test problems has been considered (from *ntype*=6 and *nroute*=5, to *ntype*=14 and *nroute*=25). The number of ants used is given as *m* in the table. Since an optimal solution of the artificial problem is any (complete) vertical line found in the problem graph, the optimal cost is (*nroute*-1) in every case. Where an optimal solution has been found the *min* value is shown in bold. The parameters α, β, ρ and Q are varied to assess the effect on the performance of the algorithm.

Table 4.1 ACS parameter settings and simulation results (without elitism) for various grid sizes of artificial problems

ntype	*nroute*	*m*	*iseed*	*Ctetrail*	α	β	ρ	*Q*	*ncycles*	*nants*	*min*
6	5	6	123456789	0.7	1	3	0.7	100	30(maxc)	6	4.4142 4.8284
6	5	6	987654321	0.7	1	3	0.7	100	30	6	4.4142
6	5	6	987654321	0.7	2	3	0.7	100	30	6	4.4142
6	5	8	987654321	0.7	1	3	0.7	100	2	1	**4.0000**
6	5	8	987654321	0.7	1	3	0.4	100	2	1	**4.0000**
6	5	8	987654321	0.7	1	3	0.2	100	2	1	**4.0000**
6	5	8	987654321	0.7	1	3	0	100	2	1	**4.0000**
6	6	12	987654321	0.5	1	3	0.7	100	2	1	**5.0000**
6	14	20	987654321	0.5	1	3	0.7	100	11	2	**13.0000**
14	14	25	987654321	0.5	1	3	0.7	100	100	25	13.8284 14.2426
14	14	30	987654321	0.5	1	3	0.7	100	11	2	**13.0000**
14	20	42	987654321	0.5	1	3	0.7	100	100	32	21.0711
14	20	45	987654321	0.5	1	3	0.8	100	60	45	19.4142 **(o)** 19.8284 **(l)**
14	20	45	987654321	0.5	1	3	0.7	100	60	45	19.4142
14	20	45	123456789	0.5	1	3	0.8	100	13	1	**19.0000**
14	20	45	987654321	0.5	1	3	0.8	1000	100	45	20.2426
14	20	45	123456789	0.5	1	3	0.8	1000	12	1	**19.0000**
14	20	45	987654321	0.5	1	3	0.8	10000	60	45	20.2426
14	20	40	987654321	0.5	1	3	0.7	100	100	40	20.6569 **(o)** 21.0711 **(l)**
14	20	40	123456789	0.5	1	3	0.7	100	100	40	19.4142 **(o)** 20.2426 **(l)**
14	20	40	123456789	0.5	0.5	3	0.7	100	23	1	**19.0000**
14	20	45	123456789	0.5	0.5	3	0.7	100	35	1	**19.0000**
14	25	55	987654321	0.5	1	3	0.7	100	70	55	25.0711 **(o)** 26.6568 **(l)**
14	25	60	987654321	0.5	1	3	0.7	100	70	60	25.0711 **(o)** 26.6568 **(l)**
14	25	60	987654321	0.5	1	3	0.8	100	70	60	25.0711 **(o)** 26.6568 **(l)**
14	25	55	987654321	0.5	0.5	3	0.8	100	50	2	**24.0000**

Where

min is the best obtained during the last cycle,
(o) represents the best so far during the simulation process (previous cycles),
(l) represents the best obtained during the last cycle,
iseed is the random seed (used for the random number generation software),
ctetrail is the initial value of the intensity of trail,
nants are the number of ants that reached *min*,
ncycle is the number of cycles needed to reach the best (min=l),
bold numbers represent the optimum for the corresponding grids considered.

Figure 4.4 show the convergence graph of the standard ACS algorithm applied to a 20 x14 grid. The parameter settings for that run are as follows:

ntype=14 *nroute*=20 *m*=40 *iseed* =123456789 *ctetrail*=0.5 α=0.5 β=3 ρ=0.7 *Q*=100.

It is noticeable that the best length fluctuates because the best so far was not transferred to the next cycle. That is, at every cycle the trail laid on the edges of the best length is reinforced with a quantity that is a function of the best length (not the best length so far). As a consequence, the performance of the ACS decreases and the computational time increases. This finding was the motivation to refine the standard ACS by introducing a form of elitism.

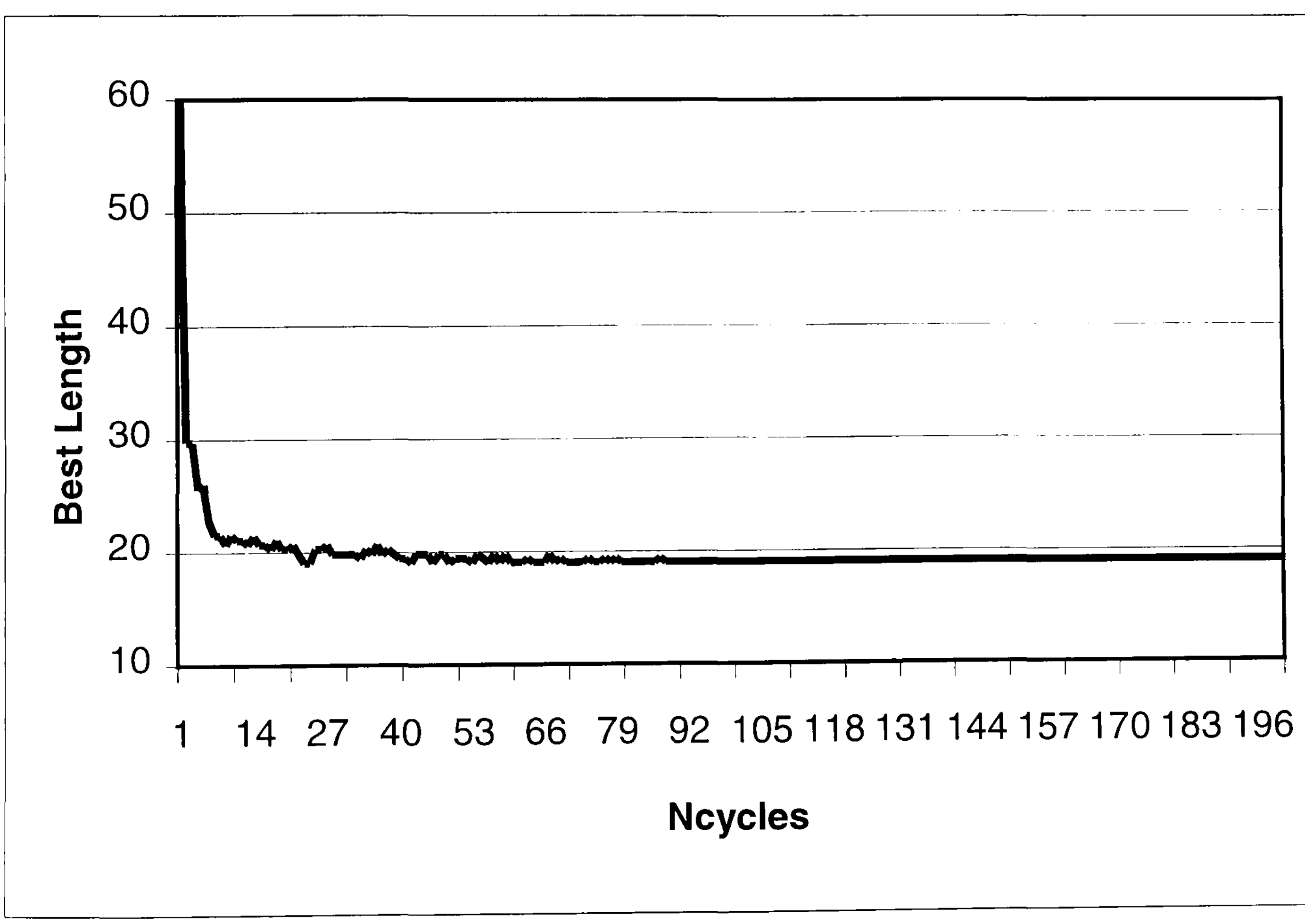


Figure 4.4 Evolution of the best path for a (20x14)grid without elitism

4.4.1.1 Elitism in Ant Colony Search

The quality of the solutions produced by the standard ACS can be improved by using the so-called elitist ants [Dorigo, Maniezzo, and Colorni, 1995]. The idea of the elitism in the context of ACS is to give extra emphasis to the best path found so far after every iteration. When the trail levels are updated, this path is treated as if a certain number of ants, namely the elitist ants, had chosen that path. Therefore, at every cycle the trail laid on the edges belonging to the best so far is reinforced more than in the standard version of the ACS. A quantity of value $elite.Q/L_{best}$ is added to the trail of each edge of the best path, where *elite* is the number of elitist ants and L_{best} is the length of the best path. The idea is that the trail of the best path so reinforced will direct the search of all other ants in probability towards a solution composed by some edges of the best path.

Figure 4.5 depicts the convergence graph of the modified ACS algorithm (with elitism) applied to the same 20x14 grid problem. The parameter settings for that run are as follows:

elite=1 *ntype*=14 *nroute*=20 *m*=40 *iseed* =123456789 *ctetrail*=0.5 α=0.5 β=3 ρ=0.7 Q=100.

By introducing elitism, the performance of ACS has improved and the computational time has been reduced (ACS converged in 48 cycles). It is important to note how in the early cycles the ACS identifies good paths, which are subsequently refined towards the end of the run.

Several tests have been conducted to investigate the effect of the *elite* ants on the performance of the algorithm. The simulation results and parameter settings are illustrated in table 4.2. The stopping criterion was to halt after a certain number of iterations and the computational time is only recorded for those tests with successful results.

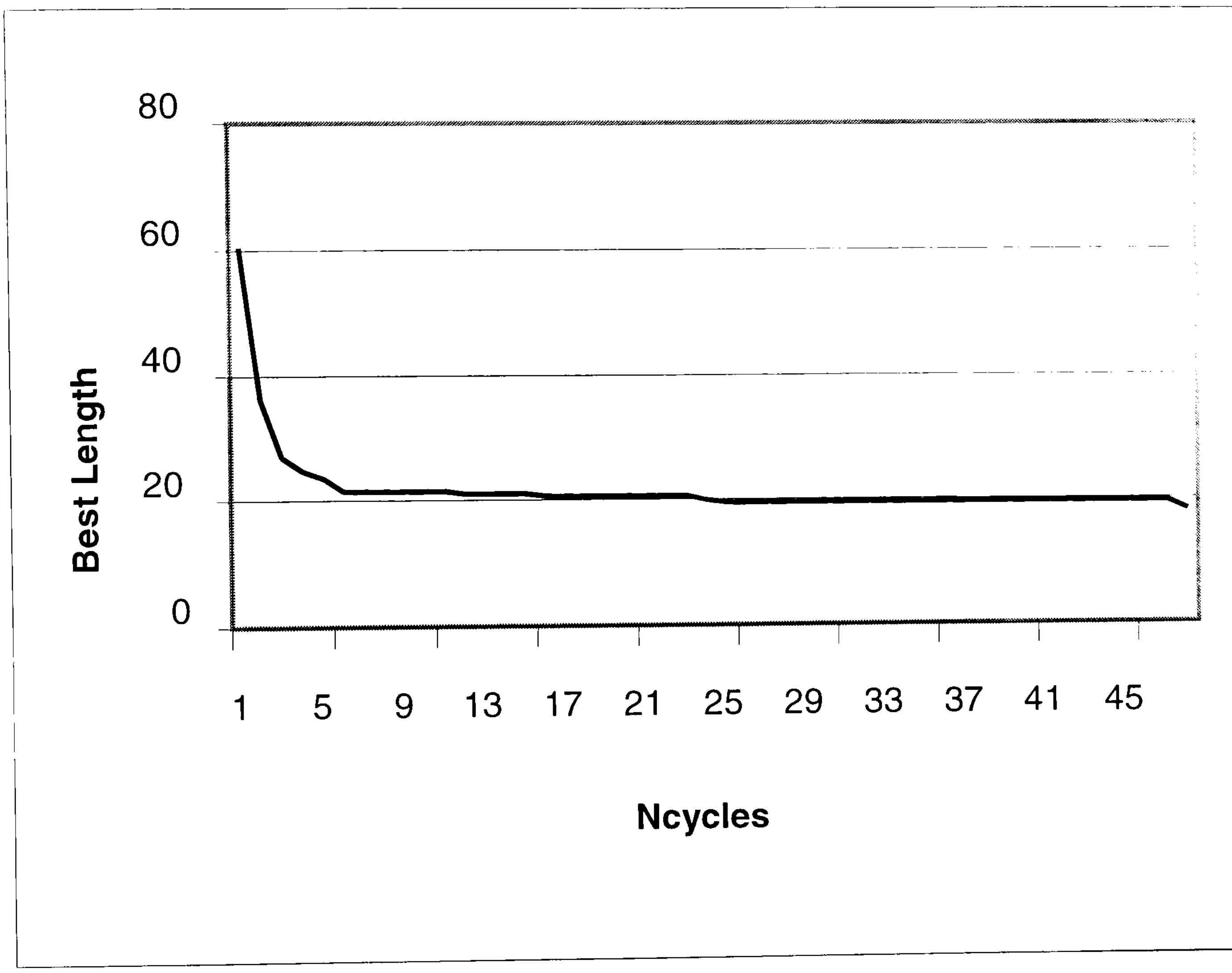


Figure 4.5 Evolution of the best path for a (20x14)grid (with elitism)

From table 4.2, it can be seen that there is an optimal range for the number of *elite* ants ($elite \in \{1...2\}$in which the ACS converged in few seconds); above it, the ants force the exploration around suboptimal paths in the early stages of the search, so that a decrease of performance results.

Table 4.2 Effect of the parameter (*elite*) on the performance of ACS

***ntype*=14 *nroute*=20 *m*=6 *iseed*=123456789 *ctetrail*=0.5 *ctevisib*=1 α=0.5 β=3 ρ=0.8 *Q*=100 *maxcycle*=300**			
elite	***ncycles***	**min**	**time (sec)**
1	38	**19.0000**	2.25324
2	40	**19.0000**	2.20317
3	----	20.2426	----
4	----	20.2426	----
5	----	20.2426	----

Bold numbers represent the optimum for a grid of (20x14)

4.4.1.2 Synergistic Effects

In ACS, ants deposit pheromone when they move and follow (in probability) the pheromone previously deposited by other ants. This allows an indirect form of communication, or *synergy*, which is responsible of locating short paths through this self-reinforcing process.

Several tests have been carried out to assess both the impact of the number of ants and the importance of communication through trail on the efficiency of ACS. The stopping criterion is to end the search after a certain number of cycles. The parameter settings and simulation results are summarised in table 4.3.

Figure 4.6 illustrates the effect of the number (m) of ants on the performance of the algorithm. The ordinate shows the time required to reach the optimum solution of the shortest path problem. The algorithm has been able to identify the correct solution with the optimum number of ants appearing to be 6. It is also noticeable that the algorithm is unreliable and fast for m in the interval {4…16}. However, it is reliable and slow for m in the interval {25…64}.

Figure 4.7 compares a situation in which ants do not communicate (α=0) with a situation in which they communicate (α=0.5). It is important to note the synergistic effect in using many ants and using the trail communication system; that is, a run with m ants is more effective with communication among ants (b) than with no communication (a). In other words, by ignoring the pheromone trail (α=0), the ants search independently relying only on the local information (visibility) which lead to the convergence to a local optimum. On the other hand, communication among ants through pheromone trail (global information) combined with the local information contributed to the convergence of the algorithm.

Table 4.3 Effect of the parameter (m) on the performance of ACS

***ntype*=14 *nroute*=20 *elite*=1 *iseed*=123456789 *ctetrail*=0.5 *ctevisib*=1 ρ=0.8 Q=100 α=0.5 β=3 *maxcycle*=300**			
m	ncycles	min	Time (sec)
4	300	21.8995	---
5	300	19.4142	---
6	38	**19.0000**	2.2532
7	300	19.8284	---
8	45	**19.0000**	3.4850
16	300	19.4142	----
25	119	**19.0000**	29.8029
30	59	**19.0000**	20.5395
32	43	**19.0000**	16.0831
40	42	**19.0000**	20.0188
45	23	**19.0000**	13.3692
50	33	**19.0000**	20.2291
64	35	**19.0000**	27.9001

bold numbers represent the optimum solution for (20x14) grid

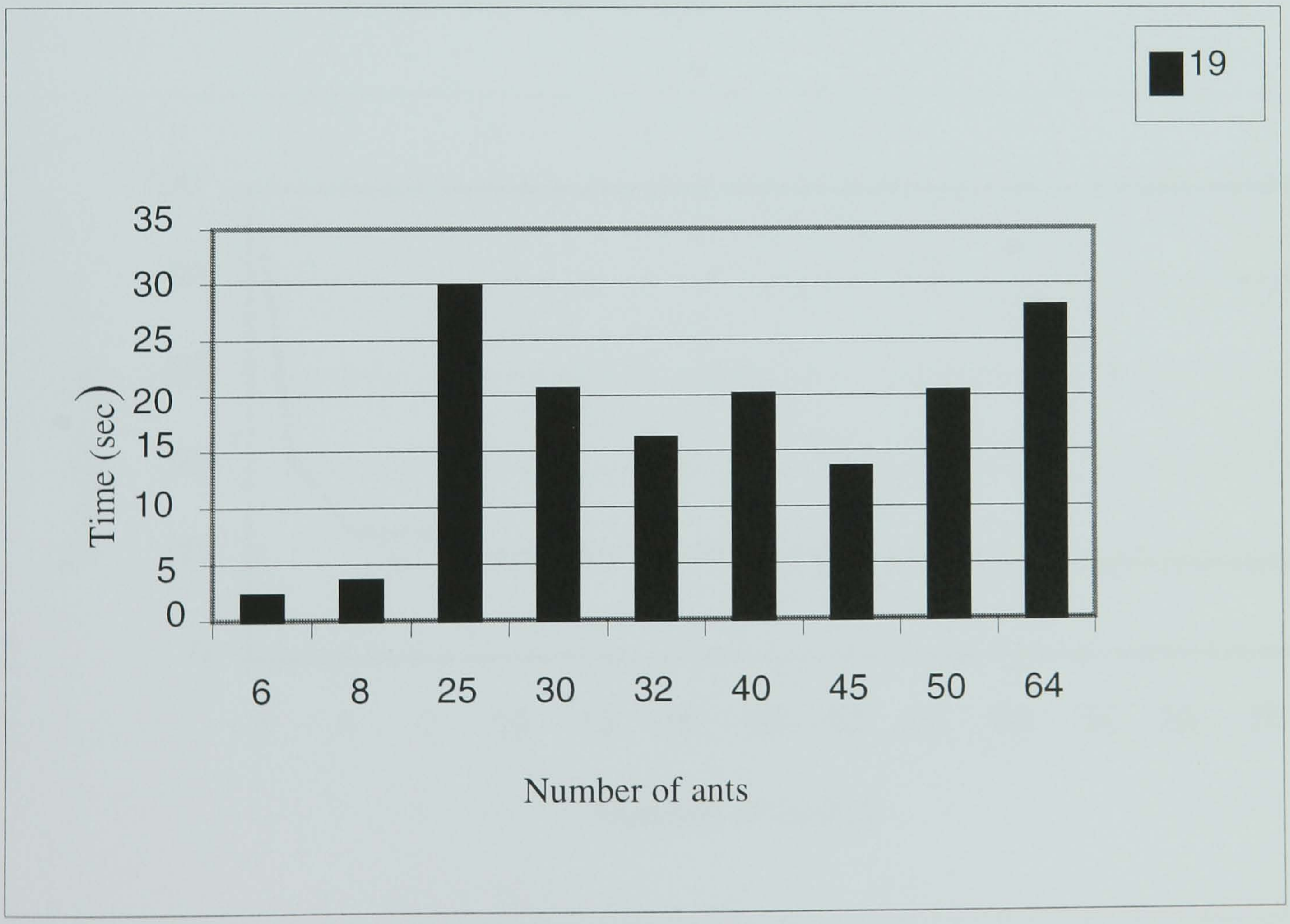
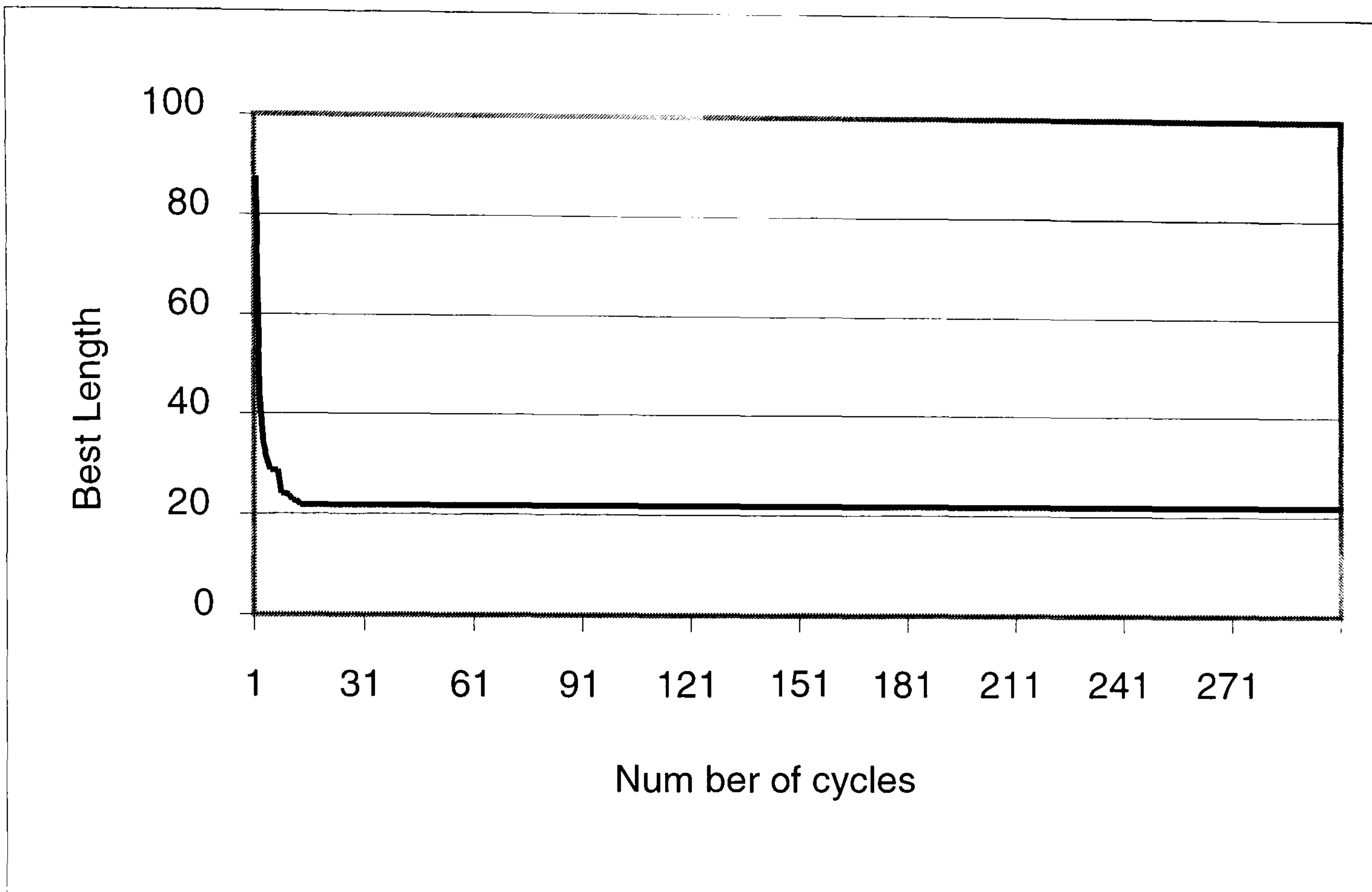
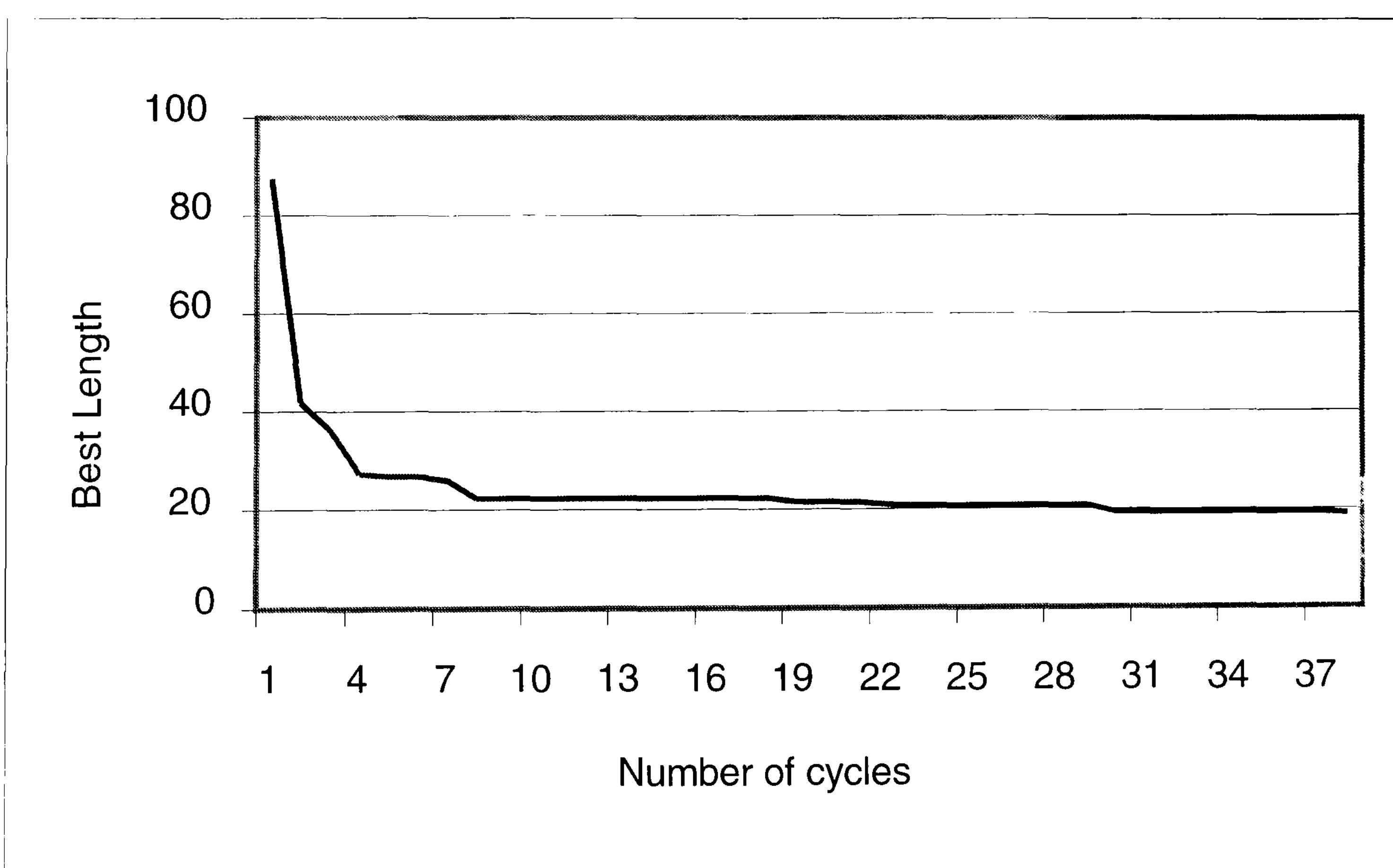


Figure 4.6 Time (required to reach the optimum) as a function of the number of ants (m) for (20x14) grid



a) α=0; no communication among ants



b) α=0.5; there is communication among ants

Figure 4.7 Synergy: communication among ants (α>0) improves performance. Compare b) against a)
m=6 β=3 $elite$=1

4.4.1.3 Parameter Settings (α & β)

The most important part of ACS is the treatment of the trail intensities. As already mentioned, the transition probability (reflecting the decision of an ant to choose a particular node) is a trade off between two desirability measures: the visibility and the trail intensity. The visibility implies that a node with a minimum path should be chosen with a high probability thus implementing a greedy constructive heuristic. The trail intensity suggests that if on a node there has been a lot of traffic then it is highly desirable, thus implementing the autocatalytic process. The relative importance of trail (representing global information) versus visibility (representing local information) is controlled by the parameters α and β.

Several tests have been conducted to investigate the behaviour of the ACS. Therefore, various combination of ACS parameters, namely, α and β , have been considered taking into account the best settings of the number of ants and elite ants in previous experiments. The possible stagnation behaviour is also investigated; i.e., the situation in which all ants followed the same path. This indicates that the system has ceased exploring new possibilities and therefore no better solution will arise.

Simulation results are summarised in table 4.4 and classified into the three following categories as illustrated in figure 4.8:

- Bad solutions and stagnation: for high values of α the algorithm enters stagnation behaviour very quickly without finding very good solutions.
- Bad solutions and no stagnation: if enough importance was not given to the trail (i.e., α was set to a low value) and visibility is low, then the algorithm did not find very good solutions.

- Good solutions: Good solutions are found for a range of parameter combinations.

Figure 4.8 suggests that good solutions are not limited by clear boundaries. Moreover, the ACS algorithm does not find the optimum for high values of $\alpha\,(\alpha \geq 1)$.

On the other hand, the ACS algorithm frequently converged for low values of α ($\alpha \leq 0.5$) and high values of $\beta\,(\beta \geq 4)$ with no clear boundaries to the convergence area (see figure 4.8).

The results obtained are consistent with our understanding of the algorithm. A high value for α means that the trail is very important. Therefore, ants tend to choose edges chosen by other ants in the past, and that has led to the poor performance. On the other hand, low values of α make the algorithm very similar to a stochastic multi-greedy algorithm guided with a very good visibility (valuable amount of local information). That is why the ACS has converged even for particularly high values of β compared to α for this particular test problem.

4.4.2 Shortest Path Problem with Deterministic Crowding Genetic Algorithm

For comparison purposes, DCGA model has been applied to the same artificial test problem (shortest path problem). The chromosome used in this case is in binary and includes *nroute* genes. Each gene is a binary number of sufficient bits to enumerate the *ntype* possible types. Several tests have been conducted to investigate the effect of the GA parameters, namely, the population size, crossover and mutation probabilities on the performance of the DCGA algorithm. The stopping criterion is to end the search after a certain number of generations. The parameter settings and simulation results are summarised in tables 4.5 & 4.6.

Figure 4.9 shows the time required to reach the optimum (best length) as a function of the population size. Results suggest that the optimum (in this case 19) could be obtained with different settings of the population size but with different computational times. The optimum computational time appear to be 7 seconds for a population size of 150.

In addition, figure 4.10 depicts the behaviour of DCGA with different crossover and mutation probabilities. The corresponding results are classified into the three following classes:

- Good solutions: DCGA finds the best solution (in this case 19).
- Close solution: DCGA gets close to the best solution.
- Bad solution: DCGA is getting far from the best solution.

It is noticeable that the optimum can be obtained with a limited set of parameter settings. Moreover, the DCGA requires fine-tuning of the parameters due to the narrow range of convergence.

Table 4.4 Effect of the parameters (α & β) on the performance of ACS

ntype=14 nroute=20 m=6 elite=2 iseed=123456789 ctetrail=0.5 ctevisib=1 ρ=0.8 Q=100 maxcycle=300				
α	β	**ncycles**	**nants**	**min**
0.2	1	---	1	34.0822
0.2	2	---	1	23.1356
0.2	3	---	1	19.8284
0.2	4	130	1	**19.0000**
0.2	5	76	1	**19.0000**
0.2	6	108	1	**19.0000**
0.2	7	26	1	**19.0000**
0.2	8	18	1	**19.0000**
0.2	9	---	3	19.4142
0.3	1	---	1	21.0711
0.3	2	---	1	21.0711
0.3	3	---	1	19.4142
0.3	4	---	3	19.4142
0.3	5	---	1	19.4142
0.3	6	---	3	19.4142
0.3	7	---	5	19.4142
0.3	8	---	4	19.4142
0.4	1	---	1	23.9574
0.4	2	---	1	19.8284
0.4	3	---	3	19.4142
0.4	4	68	1	**19.0000**
0.4	5	---	5	19.4142
0.5	1	---	1	21.4853
0.5	2	---	2	20.2426
0.5	3	40	1	**19.0000**
0.5	4	---	1	19.8284
0.5	5	---	6	19.8284
1	1	---	6	30.4789
1	2	---	6	26.5339
1	3	---	6	21.4853
1	4	---	6	22.7213
1	5	---	6	21.0711
2	1	---	6	50.6180
2	2	---	6	48.1339
2	3	---	6	34.6013
2	4	---	6	27.2646
2	5	---	6	24.8836

Bold numbers represent the optimum for (20x14) grid

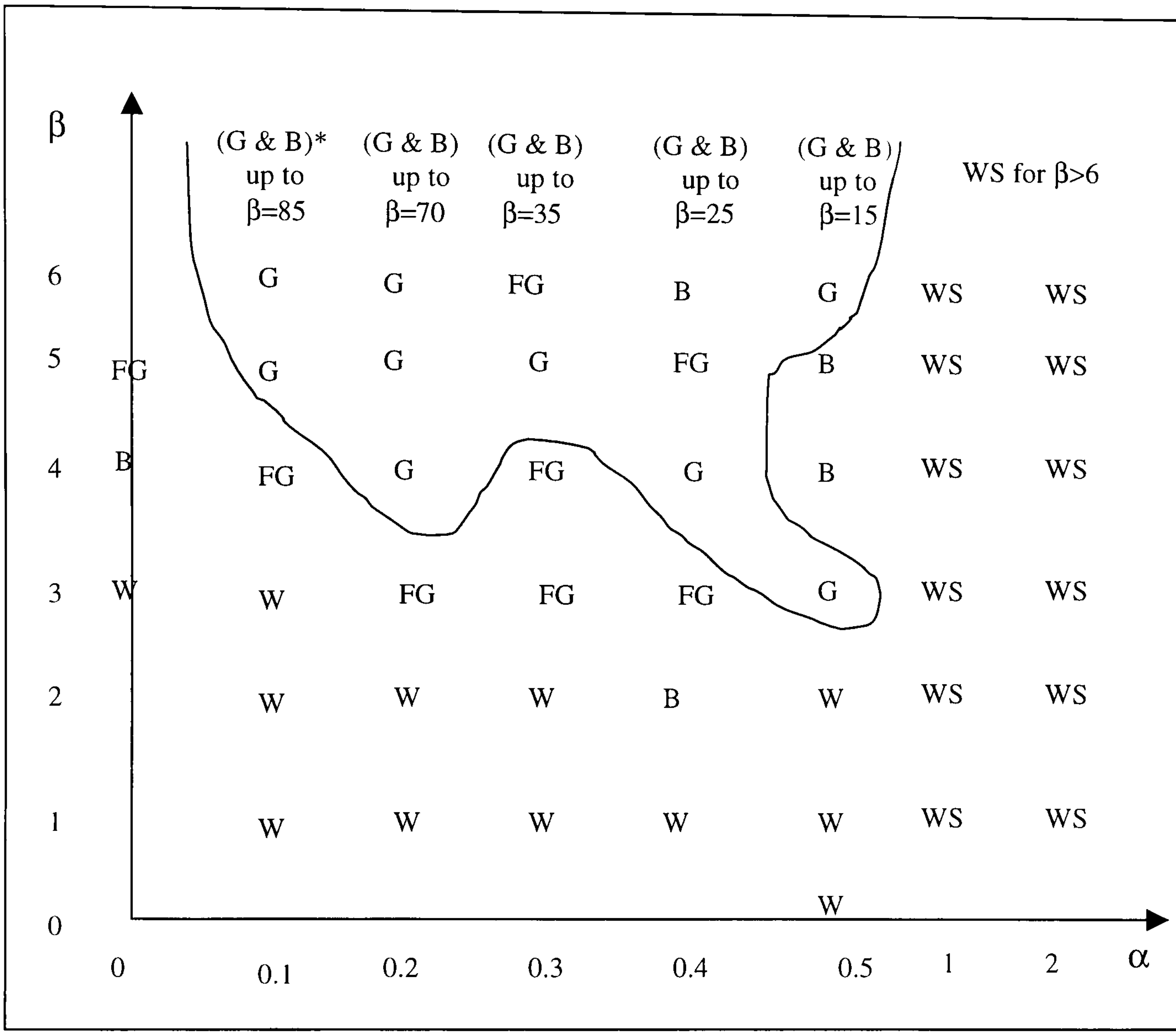


Figure 4.8 ACS behaviour for various combinations of α & β for (20x14) grid

m=6 *elite*=2

G	ACS finds the best solution without entering the stagnation behaviour.
FG	ACS is very close to the solution.
B	ACS does not find good solutions but does not enter the stagnation behaviour.
W	ACS is getting worse solutions,
WS	ACS is getting worse solutions and enters the stagnation behaviour.
*	Experiments were continued up to β=85 resulting mainly in good solutions (G) with some bad solutions (B).

Table 4.5 Effect of the population size (*popsize*) on the performance of GA

***ntype*=14 *nroute*=20 *iseed*=987654321 *pswitch*=0.04 *pmut*=0.004 *maxgen*=300**			
popsize	**gen**	**min**	**time (sec)**
100	128	21.4787	5.44783
125	152	20.2426	6.72968
150	247	**19.0000**	**6.72960**
200	225	19.4142	10.20460
225	232	19.4142	11.92715
250	276	**19.0000**	**12.11742**
280	227	**19.0000**	**11.60669**
300	219	**19.0000**	**11.57665**
325	282	19.4142	17.21475
350	158	19.4142	17.89573
400	252	**19.0000**	**17.96583**
425	220	**19.0000**	**17.06454**
500	215	**19.0000**	**19.78845**

Bold numbers represent the optimum for a grid of (20x14)

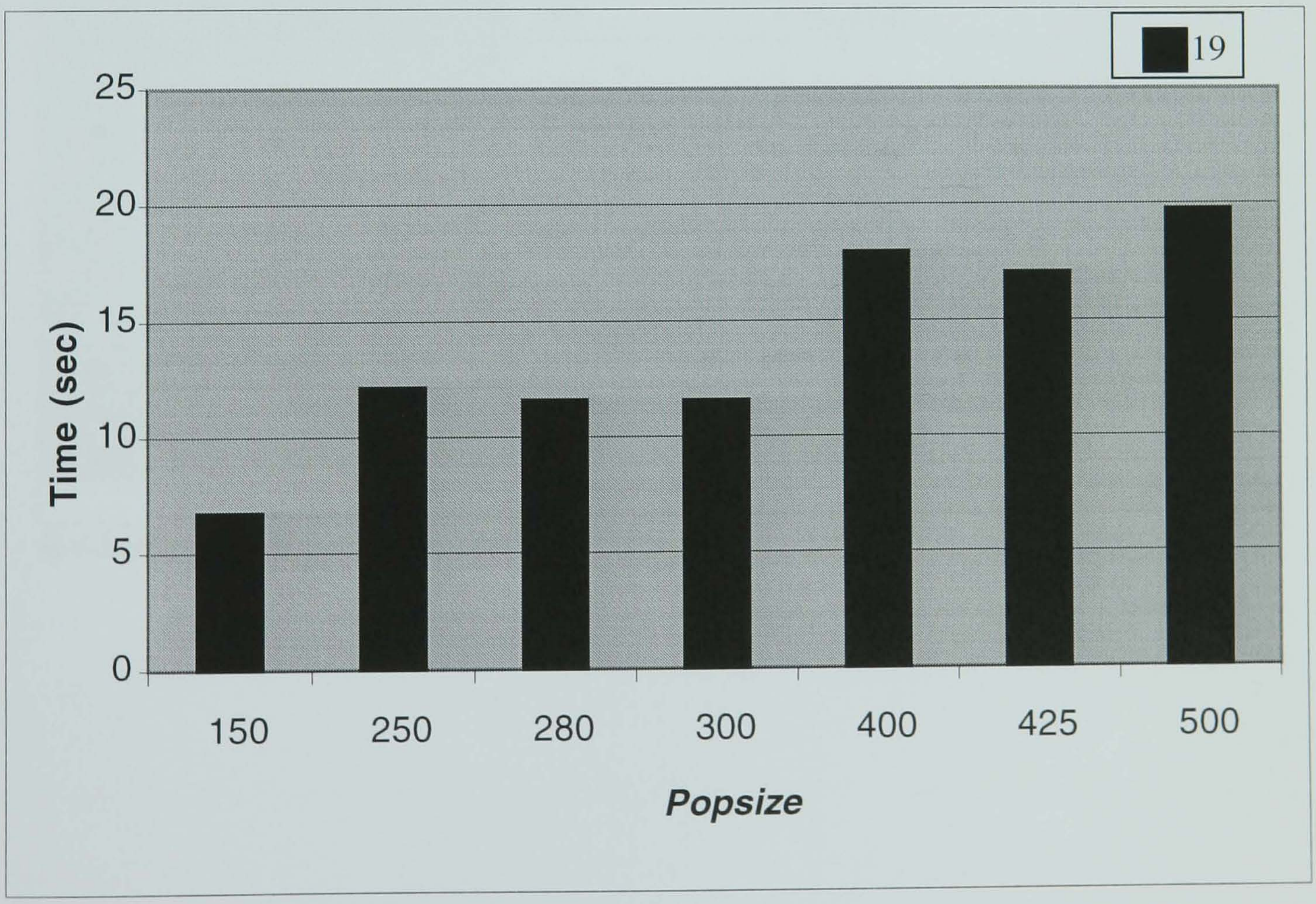


Figure 4.9 Time required to reach the optimum as a function of the population size for (20x14) grid

Table 4.6 Effect of crossover and mutation probabilities (*pcross* & *pmut*) on GA performance

***ntype*=14 *nroute*=20 *iseed*=987654321 *popsize*=150 *maxgen*=1000**			
pcross	*pmut*	**gen**	**min**
0.03	0	169	20.2426
0	0.004	973	22.7213
0.01	0.003	161	21.0711
0.02	0.002	151	19.4142
0.02	0.003	209	**19.000**
0.02	0.004	227	19.8284
0.02	0.005	196	19.4142
0.03	0.002	183	20.2426
0.03	0.003	171	19.4142
0.03	0.004	205	**19.0000**
0.03	0.005	206	19.4142
0.04	0.002	146	20.2426
0.04	0.003	174	19.8284
0.04	0.004	247	**19.0000**
0.04	0.005	259	19.4142
0.05	0.002	223	20.2426
0.05	0.003	284	**19.0000**
0.05	0.004	355	**19.0000**
0.05	0.005	260	20.6569
0.06	0.002	218	19.8284
0.06	0.003	187	19.8284
0.06	0.004	367	19.4142
0.06	0.005	360	19.4142

Bold numbers represent the optimum for a grid of (20x14)

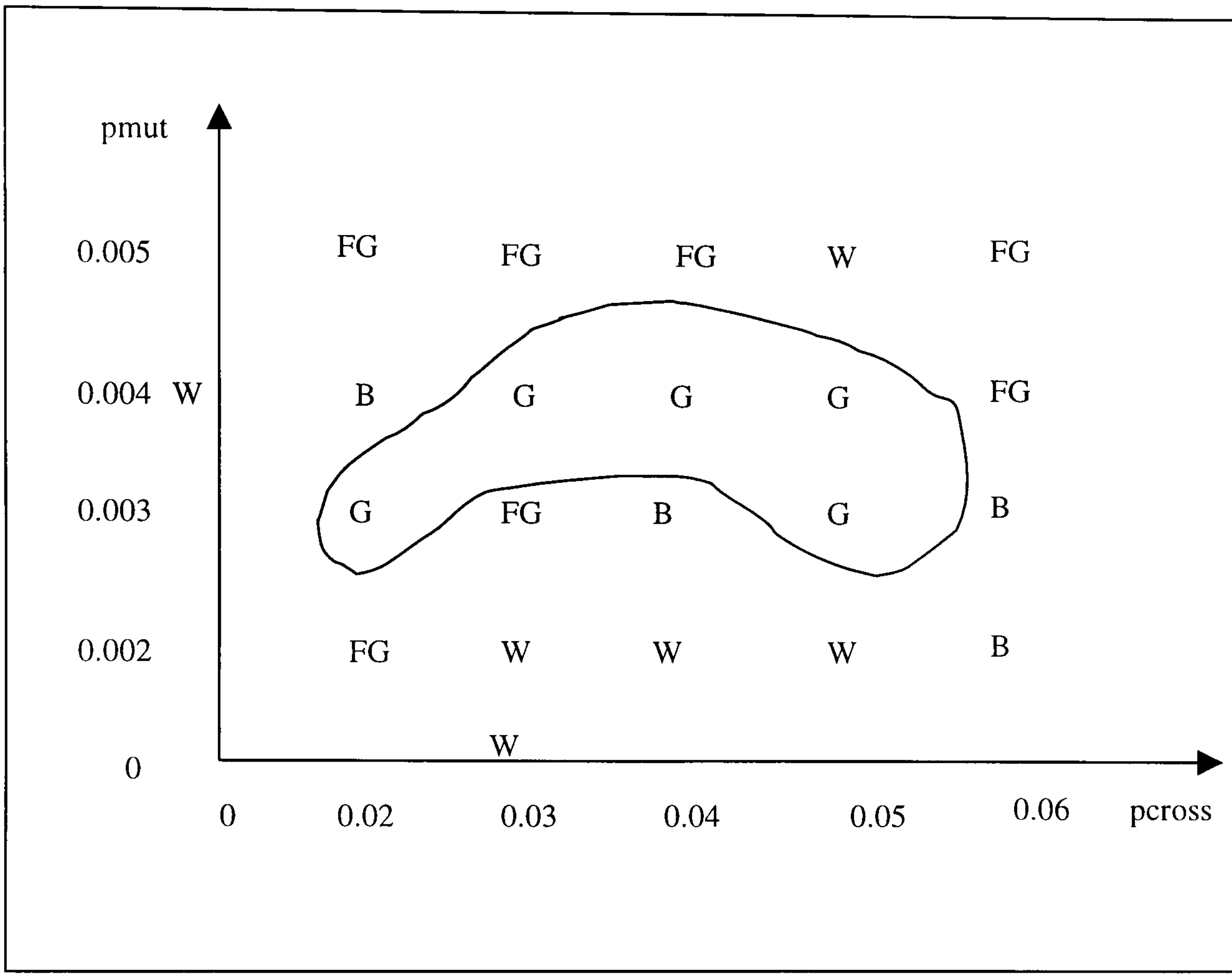


Figure 4.10 GA behaviour for different combinations of crossover and mutation probabilities for (20x14) grid.

G GA finds the best solution.
FG GA Gets close to the best solution.
B GA is getting far from the best solution
W GA is getting further from the best solution

4.5 Conclusions

The structure of the two proposed algorithms namely, the Ant Colony Search (ACS) and the Deterministic Crowding Genetic Algorithm (DCGA), have been illustrated in this chapter. Then, the application of both optimisations algorithms to the shortest path problem (with a known optimum) has been performed in an attempt to investigate their performance. Finally, a comparative investigation of both algorithms was carried out. Simulation results have shown that both algorithms (DCGA & ACS) are robust and effective in finding the known optimum for the artificial test problem (shortest path problem). Moreover, the ACS algorithm converged in relatively less computational time (few seconds less).

Experimental results have also demonstrated the importance of synergy in ACS. That is, the search carried out by a given number of ants is more effective with co-operation than without (ants acting independently).

Moreover, the effect of certain parameters of ACS namely, α and β, has been investigated. As already pointed out, α and β are parameters that control the relative importance of trail (representing global information) versus visibility (representing local information). It was found that the optimum can be obtained with different combination of α and β . However, the ACS algorithm provides a wider range regarding the visibility parameter (β). That is, ACS frequently converged with $\beta \in \{4...90\}$ and for low values of α ($\{0.1...0.5\}$). The reason may be the high visibility in conjunction with low values of α, for this particular test problem, is effective in guiding the search towards convergence. On the other hand, GA does not have the concept of visibility and therefore has less local search capability, resulting in a narrower range of convergence. Consequently, ACS appears to be more reliable than the DCGA model regarding the tuning of the parameter settings.

Therefore, it is interesting to apply the ACS & DCGA algorithms to the power transmission-planning problem. However, we can not generalise about the optimum parameter settings nor the behaviour of the ACS & GA from this particular problem. The reason is that the transmission-planning problem is a difficult non-linear, non-convex, discrete-variable constrained optimisation problem.

Chapter 5

Application of Genetic Algorithms and Ant Colony Search to Power Transmission Planning

5.1 Introduction

The goal of transmission planning in electric power systems is a robust network, which is as economical as possible, reliable and in harmony with its environment taking into account the inherent uncertainties. The mathematical formulation of the problem leads to a complicated, integer-valued, non-convex, non-linear mathematical programming problem.

In this chapter, we present two heuristic approaches for the transmission planning problem, namely the Deterministic Crowding Genetic Algorithm and the Ant Colony Search.

Several tests have been carried out on the NGC system described later. These tests have been subdivided into four categories according to the objective function. The aim is to assess the GA & ACS models as planning tools to optimise the configuration of the system.

Because the computational time increases with the complexity of the problem, and the ACS appears to be less efficient than the GA, further modelling of the transmission planning problem will subsequently be carried out using the GA approach.

Moreover both GA representations namely, the binary and integer, are implemented in the DCGA program. However, the binary representation has been adopted .

5.2 Network Modelling and Contingency Analysis

To allow more rapid evaluation of the present network model, the DC load flow [Knight, 1972] has been adopted. The DC load flow is also used in conjunction with the Householder modified matrix formula for outage studies [Cheng, 1983].

5.2.1 DC Load Flow Model

The DC load flow provides a linear active power flow model that is sufficiently accurate for the present application. The DC model assumes that:

- The system is lossless and each line is represented by its series reactance.
- Each bus has rated system voltage of 1.0 per unit.
- Angular differences between voltages at adjacent buses *m* and *n* are small.

On the basis of these assumption, the per unit power and current are synonymous and therefore can be computed as follows:

$$P_{mn} = (\theta_m - \theta_n) / X_{mn} \qquad (5.1)$$

Where

X_{mn} is the series reactance for a line between buses *m* and *n* (per unit),

θ_i is the voltage phase angle at bus *i*.

5.2.2 Branch Outage Simulation

When considering line additions to, or removals from, an existing network, it is not always necessary to build a new admittance matrix; especially if the only requirement is to establish the impact of the changes on the remaining network (e.g. overload, islanding, satisfaction of load, etc). Therefore, further modified DC models are then constructed for every outage case to be considered.

Let $[B_o]$ denote the coefficient matrix for the base-case DC load flow. In general, the outage of a line can be reflected in $[B_o]$ by modifying two elements in row k and two elements in row m. The new matrix (with the outage) is therefore as follows:

$$[B_1] = [B_o] + b[M]^t[M]$$

where

b is a line or nominal transformer series admittance,

$[M]$ is a row vector which is null except for $M_k = a$, and $M_m = -1$,

a is the off-nominal turns ratio referred to the bus corresponding to row m, for a transformer

or

$= +1$ for a line.

Depending on the types of the connected buses, only one row, k or m, might be present in $[B_1]$, in which case M_k or M_m is zero, as appropriate.

Therefore, the new bus angle vector can be computed as follows [Cheng, 1983]:

$$[\theta_1] = [\theta] - C'[X'][M][\theta] \quad (5.2)$$

where

$$C' = (1/b + [M][X'])^{-1}$$

$$[X'] = [B_1][M]^t$$

$[\theta]$ = Bus angle vector of the base-case DC load flow.

Once the voltage phase angles at the buses are evaluated the new power flows in the remaining network can be computed according to eq. 5.1.

5.2.3 Load Demand Constraints

In the representation of the network, a route with no line (type 0) is simulated by a very high impedance. To avoid the possibility of using such a route to satisfy the load demand requirement, a test criteria is introduced. This involves the observation of the angle difference of every route. If the latter is greater than a predefined limit, a penalty is applied to the line violating that limit. This penalty is chosen to be proportional to the corresponding line flow.

5.2.4 Network Security Requirement

In order to meet security requirements, it is necessary that no line shall be overloaded when one or more circuits are removed from the planned network. In this work, the so-called 'n-line' security analysis is adopted, where the optimum network plan is designed to withstand any single line outage without overloading. Furthermore the possibility of 'islanding', whereby the network operates as two or more disconnected parts, is also precluded (via high penalties) both in the intact network and in any of the outage case networks. Figure 5.1 illustrates the implementation of (n-line) security.

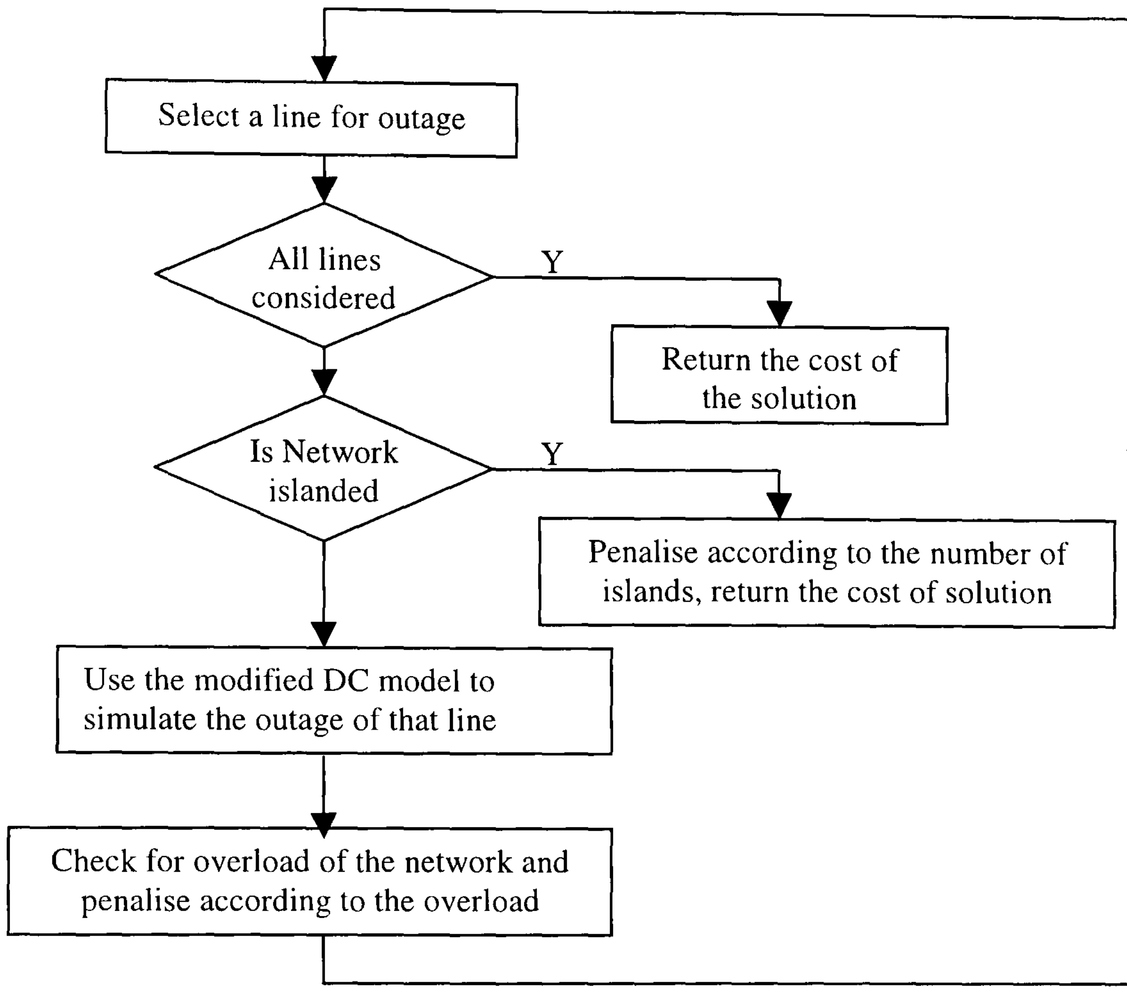


Figure 5.1 (n-line) security modelling

5.3 Evaluation of the Annuitised Loss Cost

The evaluation of the loss cost encompasses the following elements:

- the capacity cost related to the maximum MW level of losses and reflecting the cost of building extra generation capacity,
- and the energy cost related to the total MWhr consumed and reflecting fuel cost, etc.

In order to compute the annuitised loss cost, the capacity and energy loss costs have to be evaluated for a single year, and are then assumed to remain constant during the planning period (40 years in our case study) and into the future. Next, the annual costs have to be converted to equivalent present values so that they are compared on an equal basis with the capital investment cost (which is assumed to be made "now" and is one-off).

Assuming the cost of the first year is x_1, then the present values of costs in the future years form a geometric series:

$$x_1$$

$$x_2 = x_1 \,/\, (1 + disc\_rate)$$

$$x_3 = x_1 \,/\, (1 + disc\_rate)^2$$

$$x_4 = x_1 \,/\, (1 + disc\_rate)^3$$

.

$$x_{noyr} = x_1 \,/\, (1 + disc\_rate)^{noyr-1}$$

where

disc_rate is the discount rate

A discount rate of 0.075 has been applied in the present study.

Therefore the present value of loss costs for the next 40 years in the future is equivalent to the sum of the above geometric series [Khatib, 1997] as follows:

$$present\ value = x_1 + x_2 + x_3 + ... + x_{noyr} = x_1 \cdot (1 - 1/(1 + disc\_rate)^{noyr-1}) \,/\, disc\_rate$$

To work out the annual energy losses, the load factor and shape of the load curve play a significant role. Because the losses are not constant during the year a loss load factor will be applied to the peak losses. By definition the loss load factor is the ratio of the average real annual losses to the annual peak losses. In the network model adopted, a loss load factor is introduced for individual routes.

Consequently, the annuitised capacity and energy costs are computed respectively as follows:

$$annuitised\ capacity\ \cos t = peak\_losses \cdot lccf$$

$$\textit{annuitised energy} \cos t = peak\_losses \cdot lecf \cdot \sum llf_i$$

where

$$lccf = cap\_\cos t \cdot (1 - 1/(1 + disc\_rate)^{noyr-1}) / disc\_rate \cdot 1000000$$

$$lecf = smp \cdot 8760hrs \cdot (1 - 1/(1 + disc\_rate)^{noyr-1}) / disc\_rate \cdot 1000000$$

llf_i	are the loss load factor for route i,
$lccf$	is the loss capacity cost factor,
$lecf$	is the loss energy cost factor,
$noyr$	is the number of years (40 years in our case study),
cap_cost	is the capacity cost factor per year (30000 £/MW/year),
smp	is the system marginal price (25 £/MWhr).

5.4 Fitness Function for Transmission Planning

The objective of transmission planning is to minimise investment cost together with the annuitised cost of energy losses, while satisfying system constraints. Both approaches (GA & ACS) require mapping the objective cost function into a fitness function. Any violation of constraints implied by a candidate solution is handled using a penalty function approach, in which penalty costs are incorporated into the fitness function so as to reduce the apparent fitness of an infeasible candidate.

The overall objective function considered is therefore as follows:

$$\begin{aligned} F \; = \; & lecf \, . \sum \lg_i \, . \, llf_i \; + \; lccf \, . \sum \lg_i \\ & + \; E \, . \sum eg_i \; + \; \Psi \, . \sum | \, pf_i \, | \\ & + \; \phi \, . \sum \left(| \, pf_i \, | - ratl_i \right) \; + \; \sum l_i \, . \cos tl_i \\ & + \; p_0 \, . (nisl - 1) \; + \; \sum \mu_i \, . \sum \left(| \, pf_k \, | - ratl_k \right) \end{aligned} \tag{5.3}$$

where

F	is the overall fitness value,
	(Σ implies summation over the appropriate elements)
$lecf$	is the loss energy cost factor,
lg_i	are the thermal energy losses for each line at peak load,
llf_i	are the loss load factors,
$lccf$	is the energy loss capacity cost factor,
E	is the environmental impact cost factor,
eg_i	are the environmental impact factors for each line,
Ψ	is a penalty cost factor for unsatisfied loads (based on the power flow that would be required to satisfy the load),
pf_i	are the power flow in the lines,
ϕ	is a penalty cost factor for line overloading,
$ratl_i$	are the power flow ratings of each line,
l_i	are the lengths of each line,
$costl_i$	are the capital costs (per unit length) of each line,
p_0	is a penalty cost factor for network islanding,
$nisl$	is the number of network islands,
μ_i	is a penalty cost for line overloading following an outage.

Any additional factors that are pertinent to the planning problem can easily be added to the fitness function, provided that they can be computed as a function of a candidate string. However the computational time required in evaluating the fitness function will have a direct impact on the overall solution.

The candidate string generated by the GA or ACS is interpreted as a specific power network configuration. Power flows, overloads (if any), and approximate energy losses are then evaluated using the DC model. Further modified DC models are constructed for every outage case to be considered. Then the string candidate is checked against security criteria. Furthermore, the possibility of 'islanding' is also precluded (via high penalties) both in the intact network and any of the outage case networks.

Figure 5.2 summarises the process of analysing the feasibility of the network design generated by the GA or ACS and also evaluating its fitness.

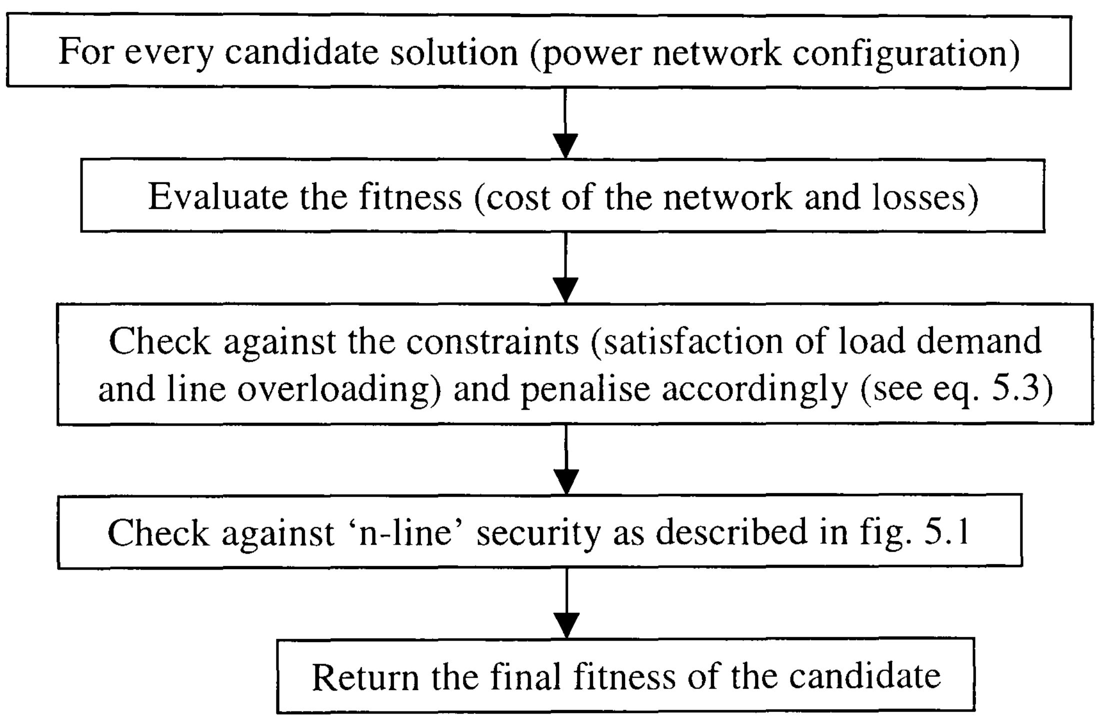


Figure 5.2 Analysis of the feasibility of a candidate solution and evaluation of its fitness

5.5 Cost Optimisation Problems

The problem objective function has been subdivided into four different cost optimisation problems. The various classes of problem are described in the following sections. For two of the problem classes the nature of the optimal solution is known in advance from theoretical considerations. These cases provide an opportunity to test the validity of the solutions proposed by GA and AS approaches. The remaining two problem classes are more realistic, but are also complex and do not have known optima. For these cases, only GA has been considered as this approach was found to be significantly more efficient than ACS (as will be shown later). Moreover, it is not possible to assess the solutions provided by GA theoretically and it is necessary to rely on numerical experiments.

5.5.1 Problem Class A: Cost of Energy Losses Only

For this problem class, the cost of energy losses is to be minimised subject only to the satisfaction of required generation and load levels throughout the network. Other costs and constraints are temporarily neglected. From considerations based on simple electrical network theory, it is apparent that an optimal solution for this case consists of a transmission network design with each available route occupied by the line type having the lowest electrical resistance.

5.5.2 Problem Class B: Investment Cost Only

In this problem class, only the capital cost of transmission lines is considered, subject to satisfying required generation and load levels. Other costs and constraints are temporarily neglected. A theoretical optimal solution for this case consists of a transmission network design based on a minimum length spanning tree (i.e. a radial network with shortest possible total line length) in which the line with lowest capital cost is used throughout.

5.5.3 Problem Class C: Energy Losses and Investment Cost

This problem class represents a realistic planning problem in which all factors, except security against outages, are considered. In particular, the solutions obtained for this class of problems show how energy loss costs are to be traded-off against initial capital costs. There are no known optimal solutions for this class of problems, and so the solutions obtained by GAs are of significant interest.

5.5.4 Problem Class D: Energy Losses, Investment, and Security Analysis Cost

This final problem class represents the full-scale problem with all factors considered. Again no optimal solutions are known in advance. The comparison of the solutions obtained for problem class C with those obtained for class D allow the additional cost of designing a secure network to be assessed.

5.6 Performance Measures

The performance measure used to judge the two proposed algorithms is based on the efficiency of the algorithms (which is assessed by observing the computational time needed to identify the optimum) and the convergence characteristics.

The algorithms can be terminated after a given number of generations (or cycles) based on empirical evidence. After the algorithm is stopped the member with the best fitness at the end of the run is selected as the optimum solution. The optimum setting of algorithm parameters corresponds to the least computational time to achieve the optimum solution.

5.7 Deterministic Crowding Genetic Algorithm for Transmission Planning

Before using any of the GA models for transmission planning, the problem must be represented in a suitable format that allows the application of the GA operators. This is the problem encoding process. The GA works by maximising a single variable, the fitness function. Therefore, the objective function and some of the constraints of the transmission-planning problem must be transformed into some measure of fitness as mentioned earlier.

The GA model to be used for the optimal solution of the transmission network is the Deterministic Crowding Genetic Algorithm (DCGA). Moreover, a modified definition of uniform crossover is adopted.

5.7.1 Problem Encoding

Binary representation has been widely used for GA analysis, partly because of the ease of binary manipulation and the fact that GA theory is based on the binary alphabet [Holland, 1975]. However some recent implementations have adopted other, more natural knowledge representation, with many successful results. When a representation other than bit strings is used, it is often necessary to redefine the genetic operators.

Exhaustive tests have been conducted with the binary representation to analyse the performance of the GA and to determine the range of optimum parameters. In addition, a limited number of experiments with a special symbol representation, namely the integer representation, have been carried out to assess the relative merits of this representation for the present problem. It was concluded that the binary approach was superior in the cases considered. Therefore, further analysis is undertaken with the binary representation and the corresponding simulation results are only reported in this chapter. However, for the sake of completeness, the alternative integer representation for the present problem is presented.

5.7.1.1 Binary Representation

Adopting a binary representation, each member of the population corresponds to a string of 1s and 0s, and represents a given transmission network design.

Every route is assigned a binary number, which represents the type of transmission line selected for that route. In the test model presented later, 15 line types are available, including line type 0 to represent an unused route, and various single-circuit and double-circuit line types. If it is anticipated that two (or more) double circuits may be required on a given route, it is necessary to specify two parallel routes in the problem description.

Associated with each line type is the corresponding capital cost, the electrical resistance, etc. A bit string of length 4 is needed to represent the 14 types. This implies that one undefined string, which does not have any corresponding line type, may occur. If this string is generated during the execution of GA, the GA operators that produce the invalid string are repeated until a valid string is produced. The 4-bit strings for each route are concatenated to form an overall string which represents a particular network design, and which is a member of the GA population.

A typical string representing a network design is as follows:

1001 1100 0010 0001

T_1 T_2 T_i .... T_{nroute}

Where

T_i represent the line types of route i

and

nroute is the number of routes (or wayleaves) in the transmission network.

5.7.1.2 Integer Representation

In this representation, each member of the population corresponds to a string of integers, and represents a given transmission network design.

Every route is represented by an integer number, which represents the type of the line for that route. In the test model presented later, 15 line types are available. These include line type 0 to represent an unused route, and various single-circuit and double-circuit line types.

The integers for each route are concatenated to form an overall string which represents a particular network design, and which is a member of the GA population.

A typical string representing a network design is as follows:

9 14 0 1

T_1 T_2 T_i .... T_{nroute}

Where

T_i represent the line types of route *i*

and

nroute is the number of routes (or wayleaves) in the transmission network.

5.7.2 Experimental Results

The DCGA model has been applied to the range of problems (problem classes A, B, C and D) derived from a 23-node 49-route transmission network design problem, which represents a simplified version of the England and Wales transmission network (table 5.3). The demand/generation profiles represent the predicted peak values for the immediate future (see table 5.1). With 15 line types to choose from, the DCGA model has the ability to choose a line type in the range 0 to 14 for every route, with type 0 representing an unused route and 1 to 14 representing standard transmission line (single-circuit or double-circuit lines) operating at voltage levels 275 kV, 400 kV, 750 kV (see table 5.2). The highest voltage level (750kV) is not used in the U.K at present and thus represents a hypothetical option.

Rigorous tests that consumed several months of CPU time have been performed on the sample problem and considering the four classes of cost function. The aim is to investigate the performance of the DCGA in the search for the optimum network design.

The effect of various combinations of DCGA parameters, namely crossover and mutation probabilities, population size and the seed (*iseed* is an arbitrary large integer used to initialise the random number generation software) responsible for the randomisation process (*pswitch*, *pmut*, *popsize*, and *iseed* respectively), have been considered. The convergence criterion is chosen so that a run is terminated after a fixed number of generations and the best obtained so far is recorded. The number of generations could be different for the various optimisation problems considered. Table 5.4 reports on the organisation of simulation results corresponding to the range of problem classes considered and the various combinations of DCGA parameters adopted.

For set values of *popsize* and *iseed*, the DCGA program is executed for a range of values of *pmut* and *pswitch*. The best results are illustrated in graphs. Every graph is associated with a certain value of *pmut* and shows the variation of the best (outcome of a run) as *pswitch* increases and corresponds to a series of DCGA runs that are generated automatically in the main program.

Adopting the same *iseed* and the optimum setting of *pmut* of previous runs, the DCGA is run again for a range of *popsize* and *pswitch* values. For those tests, every graph is associated with a certain value of *popsize* and shows the variation of the best as *pswitch* increases The optimum setting of *pmut* corresponds to the least computational time to achieve the optimum solution. More tests are also carried out by setting *pmut* and *popsize* to the same values considered in previous runs, and running DCGA program for a range of *iseed* and *pswitch* values. In this case, every graph is associated with a certain value of *iseed* and shows the variation of the best as *pswitch* increases.

Table 5.1 The demand / generation profiles for the network model for a horizon of 40 years

Nodes	Node Name	PD(MW)	PG(MW)
1	CHLX	1400	0
2	COTT	475	4079
3	DAIN	3585	246
4	DEES	523	1631
5	DUSE	318	2484
6	EXET	1254	0
7	HARK	334	1000
8	HEYS	1033	2440
9	HINP	28	1430
10	INDQ	1416	1
11	LOND	15902	10402
12	LOVE	3436	115
13	MELK	1272	1026
14	MERS	2362	2405
15	NEAS	2645	4697
16	NWAL	453	2564
17	PEMB	0	1597
18	SIZE	746	1377
19	SWAL	1975	1234
20	SYOR	4608	10040
21	WALP	745	574
22	WMID	5950	0
23	WMIG	741	1860

Table 5.2 Transmission line types: costs and characteristics

Types	Voltages (kV)	Circuit Types	Rating (MW)	Reactance (%/km)	Resistance (%/km)	Cost (£m/km)
1	750	Double-circuit	18866	0.0026	0.00014	0.88
2	750	Single-circuit	9433	0.0052	0.00027	0.616
3	400	Double-circuit	6840	0.0083	0.00055	0.578
4	400	Double-circuit	5720	0.0091	0.00055	0.517
5	400	Double-circuit	5040	0.0092	0.00065	0.503
6	400	Double-circuit	4360	0.00925	0.0008	0.405
7	400	Double-circuit	4020	0.0095	0.0009	0.394
8	400	Double-circuit	3420	0.0098	0.00105	0.343
9	275	Double-circuit	4700	0.01925	0.00116	0.578
10	275	Double-circuit	3000	0.01957	0.00169	0.405
11	275	Double-circuit	2760	0.0201	0.0019	0.394
12	275	Double-circuit	2360	0.02073	0.00222	0.343
13	275	Double-circuit	1730	0.025	0.0027	0.262
14	275	Double-circuit	1350	0.02	0.00525	0.223

Table 5.3 Transmission network model

Routes	Sending	Receiving	Length(km)	LLF
1	WMID	CHLX	47	0.5
2	LOND	CHLX	106	0.5
3	MELK	CHLX	58	0.5
4	SWAL	CHLX	113	0.5
5	SYOR	COTT	103	2*
6	SYOR	COTT	34	2
7	WMIG	COTT	71	0.5
8	LOND	COTT	222	0.5
9	LOND	COTT	191	0.5
10	WALP	COTT	104	0.5
11	SYOR	DAIN	98	2
12	SYOR	DAIN	110	2
13	SYOR	DAIN	126	2
14	HEYS	DAIN	60	0.5
15	DEES	DAIN	60	0.5
16	WMIG	DAIN	99	0.5
17	MERS	DEES	12	0.5
18	WMID	DEES	117	0.5
19	NWAL	DEES	79	0.5
20	NWAL	DEES	130	0.5
21	LOND	DUSE	101	0.5
22	LOVE	DUSE	179	0.5
23	LOVE	EXET	210	0.5
24	HINP	EXET	65	0.5
25	INDQ	EXET	143	0.5
26	NEAS	HARK	107	2
27	HEYS	HARK	148	2
28	MERS	HEYS	81	0.5
29	INDQ	HINP	198	0.5
30	MELK	HINP	87	0.5
31	SIZE	LOND	161	0.5
32	WMIG	LOND	178	0.5
33	WALP	LOND	122	0.5
34	MELK	LOND	132	0.5
35	LOVE	LOND	57	0.5
36	SWAL	MELK	107	0.5
37	SWAL	MELK	157	0.5
38	SYOR	NEAS	289	2
39	SYOR	NEAS	176	2
40	SWAL	PEMB	144	0.5
41	SWAL	PEMB	146	0.5
42	WALP	SIZE	184	0.5
43	COTT	SYOR	27	2
44	WMIG	WMID	52	0.5
45	WMIG	WMID	84	0.5
46	WMIG	WMID	113	0.5
47	WMID	DEES	117	0.5
48	WMID	CHLX	47	0.5
49	WMIG	COTT	71	0.5

* for some routes the flow at system peak is relatively low (non-conforming lines) and a loss load factor of greater than 1.0 must be applied.

Table 5.4 Classification of simulation results corresponding to the range of problems considered and the various combinations of DCGA parameters adopted

	Class A	**Class B**	**Class C**	**Class D**
	Figures 5.4 (optium)	Figure 5.9 (optimum)	Figure 5.11 (proposed optimum)	Figure 5.14 (proposed optimum)
Pmut and *pswitch*	Tables 5.5-5.7 Figures 5.3	Tables 5.8-5.10 Figures 5.5-5.6 Figure 5.10 (convergence graph)	Tables 5.13-5.15 Figures 5.12-5.13	Tables 5.16-5.18 Figures 5.15-5.16
popsize and *pswitch*	-	Tables 5.11-5.12 Figures5.7-5.8	-	-
iseed and *pswitch*	-	-	-	Tables 5.19-5.20 Figures 5.17-5.18

pmut is the probability of mutation *popsize* is the population size
pswitch is the probability of crossover *iseed* is responsible for initialising the randomisation process

5.7.2.1 Problem Class A

The tests were conducted by setting up some of the GA parameters as shown in table 5.5 and varying other parameters such as the crossover and mutation probabilities (*pswitch* and *pmut*). A summary of the performance of the DCGA is given in tables 5.6 and 5.7 and figures 5.3 and 5.4.

Figure 5.4 shows the power network design obtained by the DCGA model. It is important to note that every route has been used and the least resistive line type has been chosen (line type 1) throughout. This result agreed with the known theoretical optimum for this problem class.

Figure 5.3 suggests that the optimum is obtained with a wide range of the GA parameters considered particularly when $pswitch \in \{0.01...0.5\}$and $pmut \in \{0.002...0.009\}$. The reason is that the problem of loss minimisation appears to be quite easy for GA. It seems straightforward for GA to use all possible routes, and the least resistive line type (which increases the fitness of the candidate) will be chosen through evolution. That is, once good solutions are obtained for particular routes, it is very likely for them to be propagated to the next generations through crossover and then selection and replacement (survival of the fittest). The best of a generation is always copied to the next generation, due to the embedded elitism in the DCGA model, and therefore good solutions (parents) are only replaced by the most similar solutions (children) if they are fitter.

In addition, it can be seen from table 5.7 that the number of generations needed to converge is different for the various combinations of *pmut* and *pswitch* considered, and therefore the computational time is different. The optimum setting is when *pmut* $\in$ $\{0.002 \ldots 0.003\}$ and *pswitch* $\in$ $\{0.02 \ldots 0.07\}$.

Table 5.5 GA parameter settings for class A

GA parameter settings (class A)	
Popsize	500
iseed	978456333
maxgen	1000
ptest	0.5

Popsize is population size, *iseed* is responsible for the randomisation process
maxgen is maximum number of generations of a run,
ptest is probability for crossover applied on first bits of the parents.

Table 5.6 GA Parameter settings (*pmut* & *pswitch*) for Class A

Best (£m) after 1000 generations								
pmut-->	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
pswitch								
0.01	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.02	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.03	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.04	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.05	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.06	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.07	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.08	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.09	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.1	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.2	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.3	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.4	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**
0.5	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**	**238.095**

pmut is the probability of mutation
pswitch is the probability of crossover

Table 5.7 Number of generations needed to converge for Class A

Number of generations needed to converge								
pmut-->	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
pswitch								
0.01	221	248	262	292	357	378	382	457
0.02	191	178	223	255	316	321	368	439
0.03	186	190	239	242	284	318	350	404
0.04	177	194	224	252	271	315	354	393
0.05	187	213	233	259	254	316	350	424
0.06	187	205	235	247	280	355	364	410
0.07	196	212	245	266	322	358	382	435
0.08	205	217	259	276	281	378	384	430
0.09	203	239	254	271	322	379	406	420
0.1	221	235	276	289	355	362	388	473
0.2	266	309	315	387	453	458	525	517
0.3	342	368	412	454	526	559	630	642
0.4	392	430	457	512	619	637	753	764
0.5	426	486	532	565	707	753	795	979
Run-time (min) ≅ 21 for 1000 generations								

pmut is the probability of mutation
pswitch is the probability of crossover

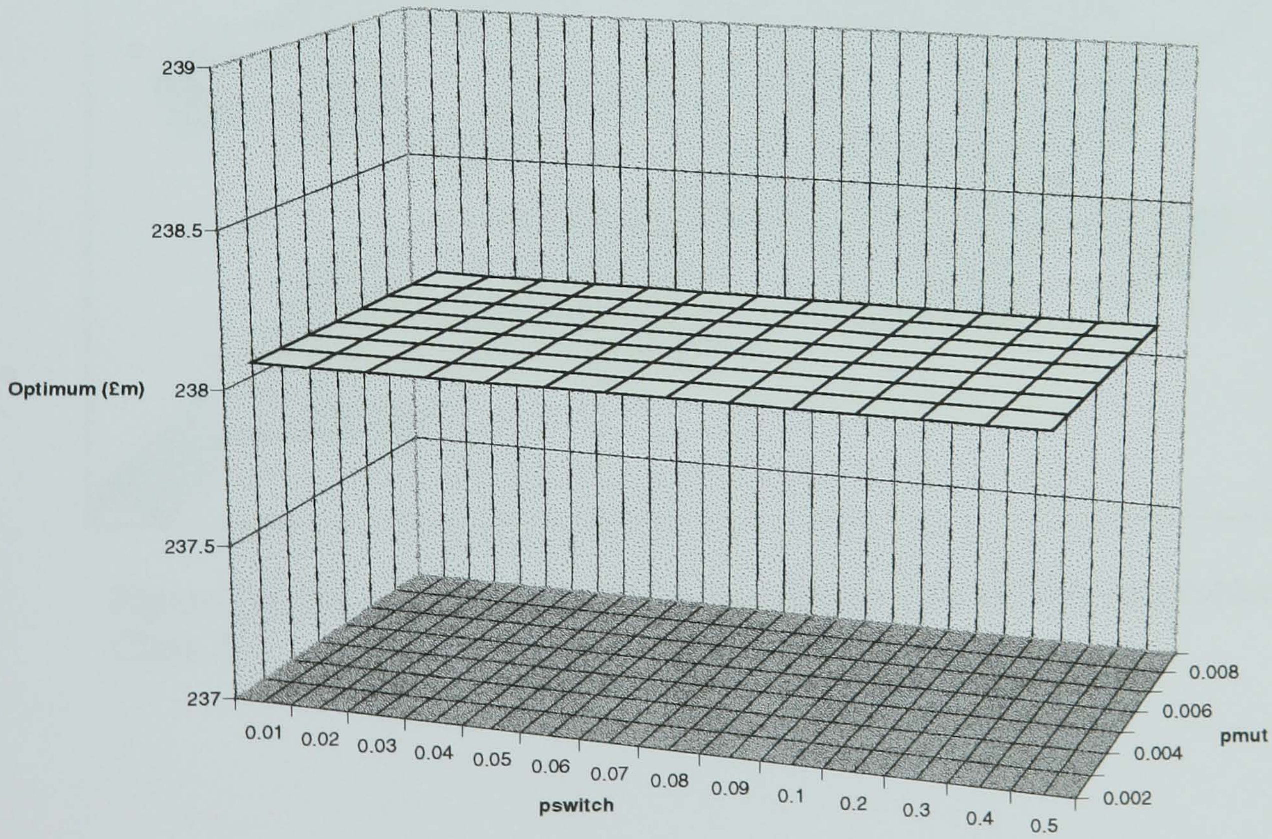


Figure 5.3 Landscape representing the optimum as a function of *pmut* and *pswitch* for problem Class A

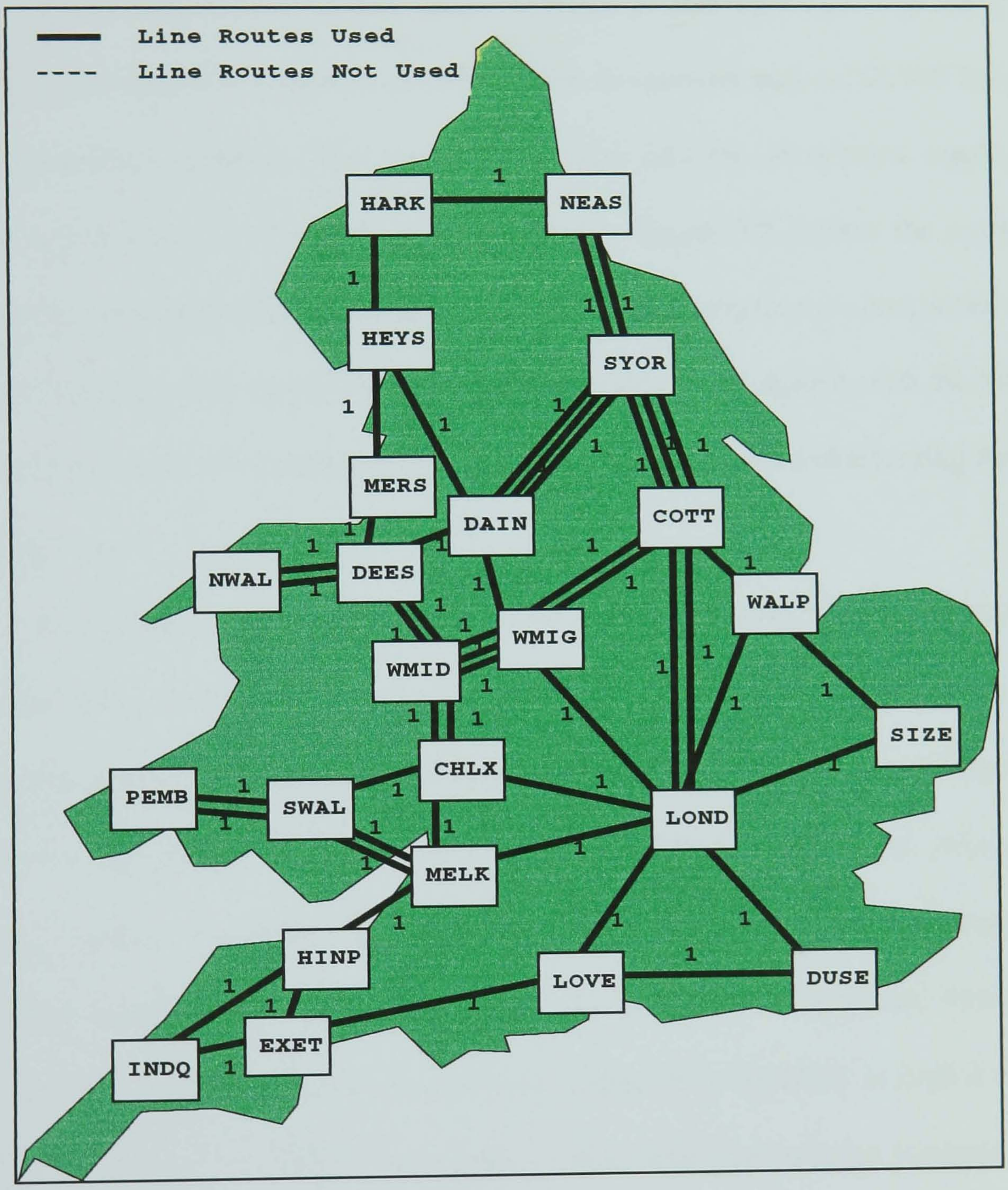


Figure 5.4 The optimum network design obtained by DCGA for problem Class A

5.7.2.2 Problem Class B

In these simulations the combined effect of crossover and mutation probabilities (*pswitch* and *pmut* respectively) is investigated. More analysis is also undertaken to study the effect of the population size (*popsize*) and *pswitch* on the performance of DCGA keeping the other parameters constant. The parameter settings and the simulation results are reported in tables 5.8 to 5.12 and figures 5.5 to 5.10. Figure 5.9 depicts the optimum network design obtained by DCGA whereby a minimal tree structure is constructed with the cheapest line type (14) chosen throughout. Again, this result agreed with the known theoretical optimum for this problem class, which is found to be more challenging for GA than problem class A.

Tables 5.9 shows that the optimum can be obtained with different setting of *pmut* and *pswitch* (*pmut* $\in$ {0.003...0.009} and *pswitch* $\in$ {0.02...0.2}). However, the computational time is different as can be seen in table 5.10. The optimum setting (least computational time) seems to occur when *pmut* $\in$ {0.003...0.004} and *pswitch* $\in$ {0.03...0.1}. Figures 5.5 and 5.6 illustrate the best as a function of *pswitch* and *pmut*. It is noticeable that the GA requires some tuning in order to obtain the optimum. Moreover, the DCGA exhibit a poor performance when the crossover probability is higher than a certain value (*pswitch* > = 0.2). The reason is that high performance structures are discarded faster than selection can produce improvements. However if the crossover probability is low (*pswitch* < 0.02) the search might stagnate, and the performance would degrade.

Figures 5.7 and 5.8 depict the effect of the population size on the performance of the algorithm. It is important to note that GA exhibits poor performance for small population size (the GA did not find the optimum when *popsize* is less than 200) whereas the GA was able to identify the optimum as *popsize* increases. The reason is that the population size affects both the performance and efficiency of a GA. Very small populations do not provide a sufficient sample of the problem domain whereas a large population is more likely to contain more representatives from the search space leading to a more informative search. However a large population requires more evaluations per generation possibly resulting in an unacceptable slow computation.

Figure 5.10 illustrates the convergence graphs (variation of the best as the number of generations increase) of several runs corresponding to a particular value of *pmut* (0.004) and various values of *pswitch* in the range {0.01...0.1} respectively. Figure 5.10 (b) is a magnified version of figure 5.10 (a) showing the variation of the best towards the end of evolution (best in the range £420m to £460m). It is important to note the convergence characteristic of DCGA in the search for the optimum. The best improves fairly quickly in the early stage of evolution, whereas this improvement slows down towards the end of the run, where the competition is held among similar and fit members. In addition this improvement depends on the crossover probability more than on mutation probability (optional for the DCGA model). The convergence graph is slower as *pswitch* increases and sometimes leads to poor performance. Moreover, we can see clearly that the best remains constant for a number of generations before it improves again. This implies that the stopping criteria for a run based on a maximum number of generations should be chosen carefully in order to allow enough exploration. Therefore, if the GA is left running for more generations, it is more likely to get better solutions and the landscape of figure 5.6 and 5.8 would be smoother.

Therefore, for production purposes it is recommended to run the program several times trying different parameter settings to ensure that the solution falls within the convergence area. Moreover, the GA provides other solutions (slightly more expensive) which are of interest to the planning engineer. These solutions (network designs) might be worthy of consideration due to additional factors, which were not included in the computer evaluation of the design cost.

Table 5.8 GA Parameter settings for Class B

GA parameter settings (class B)	
Popsize	500
iseed	978456333
maxgen	2000
ptest	0.5

popsize is population size, *iseed* is responsible for the randomisation process
maxgen is maximum number of generations of a run,
ptest is probability for crossover applied on first bits of the parents.

Table 5.9 GA Parameter settings (*pmut* & *pswitch*) and simulation results for Class B

Best(£m) after 2000 generations (class B)							
pmut-->	0.003	0.004	0.005	0.006	0.007	0.008	0.009
pswitch							
0.01	425.93	425.93	425.93	432.62	422.585	425.93	426.153
0.02	426.153	425.93	426.376	**422.362**	**422.362**	425.93	425.93
0.03	**422.362**	**422.362**	**422.362**	432.174	431.059	**422.362**	423.923
0.04	425.038	426.153	425.93	**422.362**	425.93	432.174	427.045
0.05	427.491	425.93	422.362	**422.362**	**422.362**	**422.362**	**422.362**
0.06	**422.362**	425.93	425.93	**422.362**	426.153	425.93	**422.362**
0.07	427.491	**422.362**	422.808	422.585	**422.362**	422.808	427.045
0.08	425.93	425.93	427.045	425.93	425.93	**422.362**	425.93
0.09	422.808	**422.362**	**422.362**	**422.362**	425.93	**422.362**	429.944
0.1	**422.362**	423.7	**422.362**	**422.362**	**422.362**	427.937	**422.362**
0.2	423.923	436.857	426.153	423.7	**422.362**	434.627	427.853
0.3	427.714	425.93	428.829	452.315	431.728	423.031	425.707
0.4	431.282	425.93	423.031	434.283	454.653	450.461	444.996
0.5	436.411	439.31	443.324	459.904	492.026	512.23	473.368

pmut is the probability of mutation
pswitch is the probability of crossover

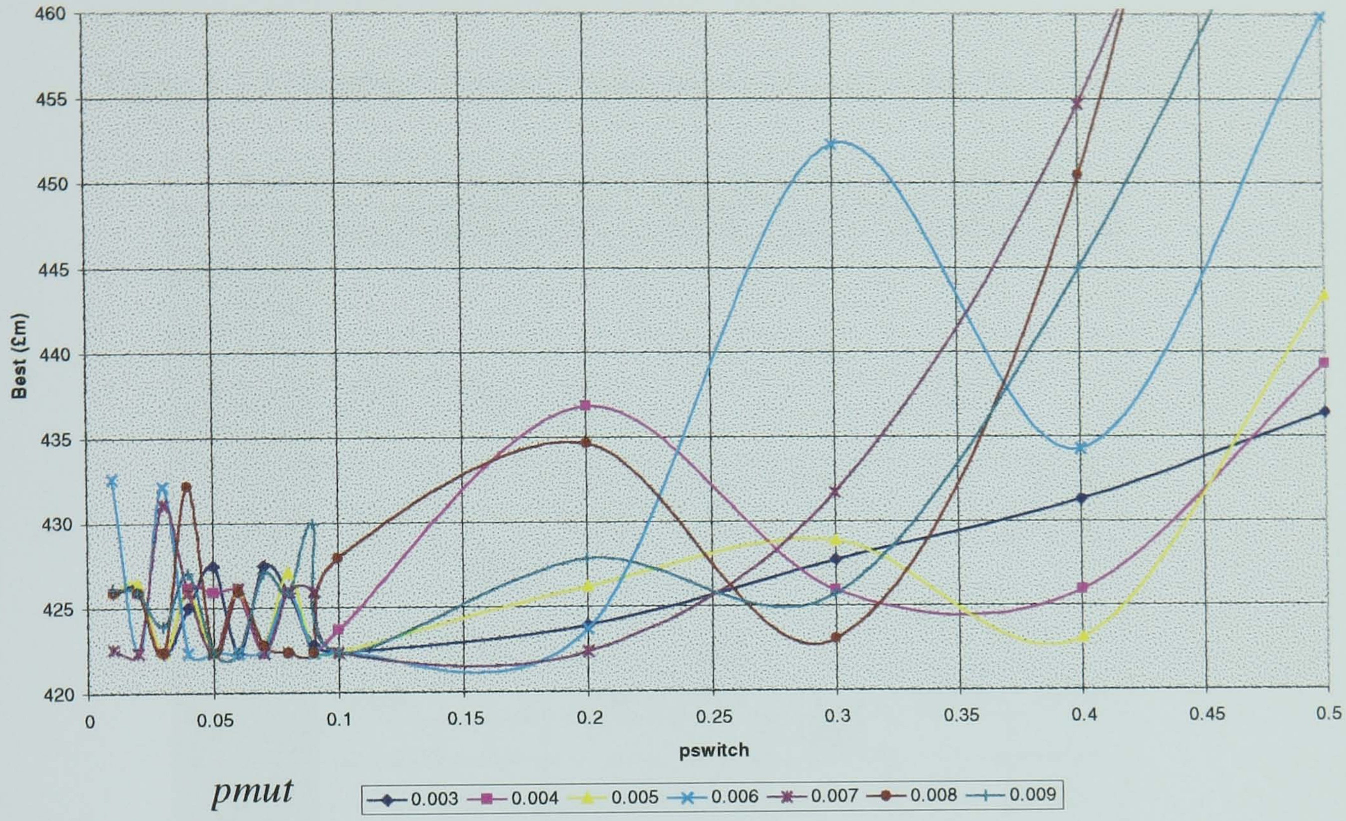


Figure 5.5 Variation of the best as a function of *pmut* and *pswitch* for problem Class B

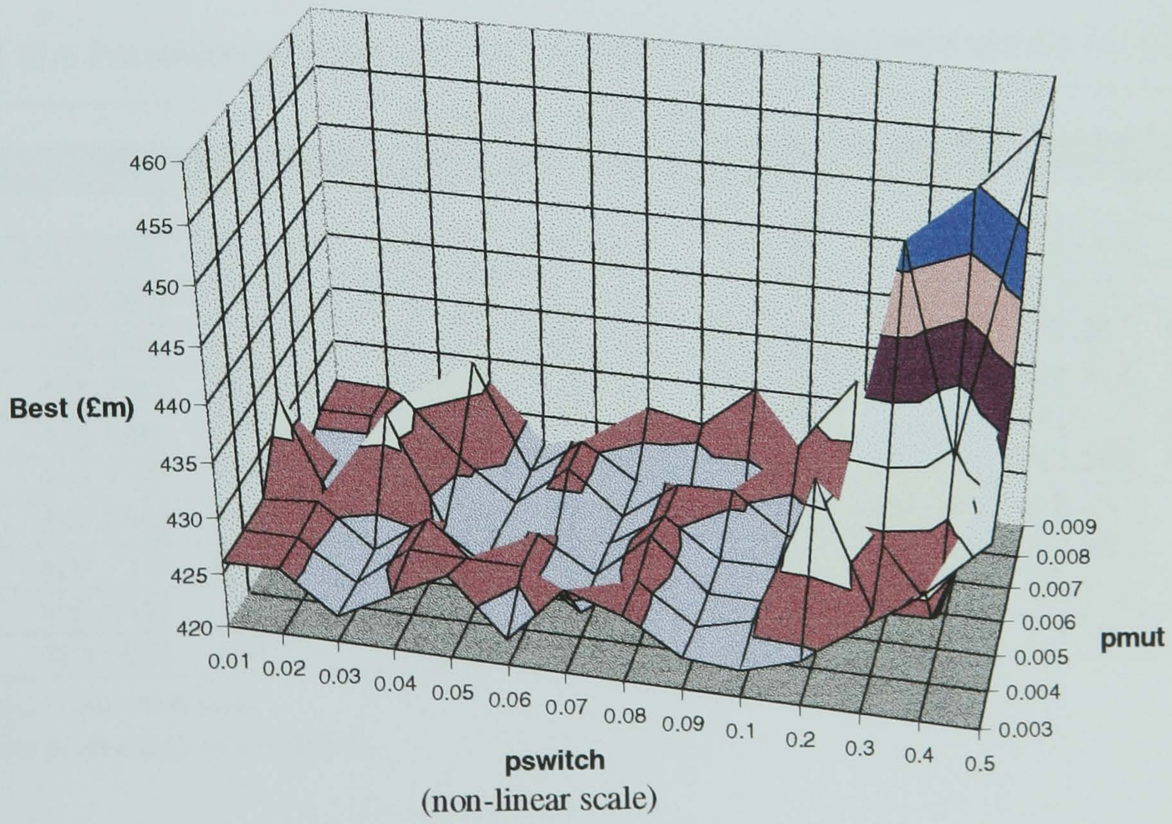


Figure 5.6 Landscape representing the best as a function of *pmut* and *pswitch* for problem Class B

Table 5.10 Number of generations needed to converge for Class B

Number of generations needed to converge (class B)							
pmut-->	0.003	0.004	0.005	0.006	0.007	0.008	0.009
pswitch	-	-	-	-	-	-	-
0.01	-	-	-	-	-	-	-
0.02	-	-	-	1731	938	-	-
0.03	712	585	956	-	-	1409	-
0.04	-	-	-	823	-	-	-
0.05	-	-	1017	1348	1345	1434	1668
0.06	632	-	1016	1280	-	-	1209
0.07	-	939	-	-	1108	-	-
0.08	-	-	-	-	-	1953	-
0.09	-	791	857	1549	-	-	-
0.1	869	-	1267	1680	1916	1623	1972
0.2	-	-	-	-	1633	-	-
0.3	-	-	-	-	-	-	-
0.4	-	-	-	-	-	-	-
0.5	-	-	-	-	-	-	-
Run-time (min) ≅ 40 for 2000 generations							

pmut is the probability of mutation
pswitch is the probability of crossover

Table 5.11 GA Parameter settings (*pswitch & popsize*) and simulation results for Class B

Best (£m) (class B)								
popsize	100	200	300	400	500	600	700	800
pswitch								
0.01	470.7531	429.7211	448.23	428.1601	425.93	**422.362**	425.93	425.93
0.02	453.582	430.1671	422.585	426.153	425.93	425.93	425.93	425.93
0.03	435.296	427.045	430.167	**422.362**	**422.362**	**422.362**	**422.362**	**422.362**
0.04	466.739	442.432	427.491	**422.362**	426.153	**422.362**	**422.362**	**422.362**
0.05	430.836	427.268	428.606	432.174	425.93	425.93	425.93	**422.362**
0.06	433.735	426.376	422.808	425.93	425.93	**422.362**	**422.362**	**422.362**
0.07	457.819	429.275	431.059	**422.362**	**422.362**	**422.362**	**422.362**	425.93
0.08	427.714	**422.362**	422.808	**422.362**	425.93	**422.362**	**422.362**	425.93
0.09	433.512	425.93	**422.362**	429.721	**422.362**	**422.362**	423.923	423.923
0.1	429.052	427.714	422.808	427.937	423.7	426.376	427.491	425.93

popsize is the population size
pswitch is the probability of crossover

Table 5.12 Number of generations needed to converge for Class B

Number of generations needed to converge after 2000 generations								
Popsize->	100	200	300	400	500	600	700	800
pswitch	-	-	-	-	-	-	-	-
0.01	-	-	-	-	-	1218	-	-
0.02	-	-	-	-	-	-	-	-
0.03	-	-	-	862	585	817	656	639
0.04	-	-	-	723	-	775	857	928
0.05	-	-	-	-	-	-	-	936
0.06	-	-	-	-	-	1197	1624	749
0.07	-	-	-	1109	939	1887	1145	-
0.08	-	873	-	986	-	627	1185	-
0.09	-	-	766	-	791	778	-	-
0.1	-	-	-	-	-	-	-	-
Run-time (min)	-	16	24	32	-	49	56	64

pmut is the population size
pswitch is the probability of crossover

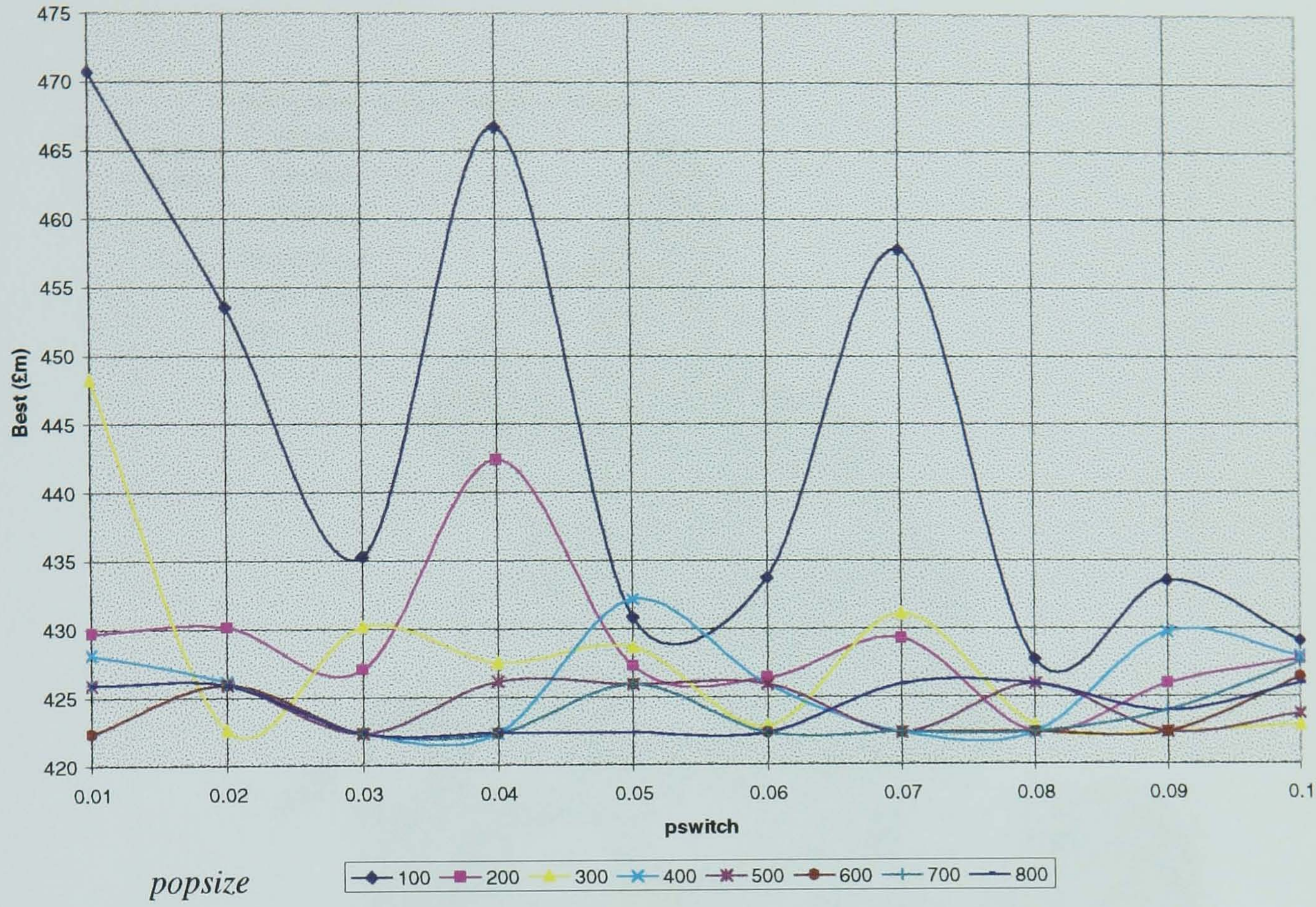


Figure 5.7 Best as a function of *pswitch* and *popsize* for problem Class B

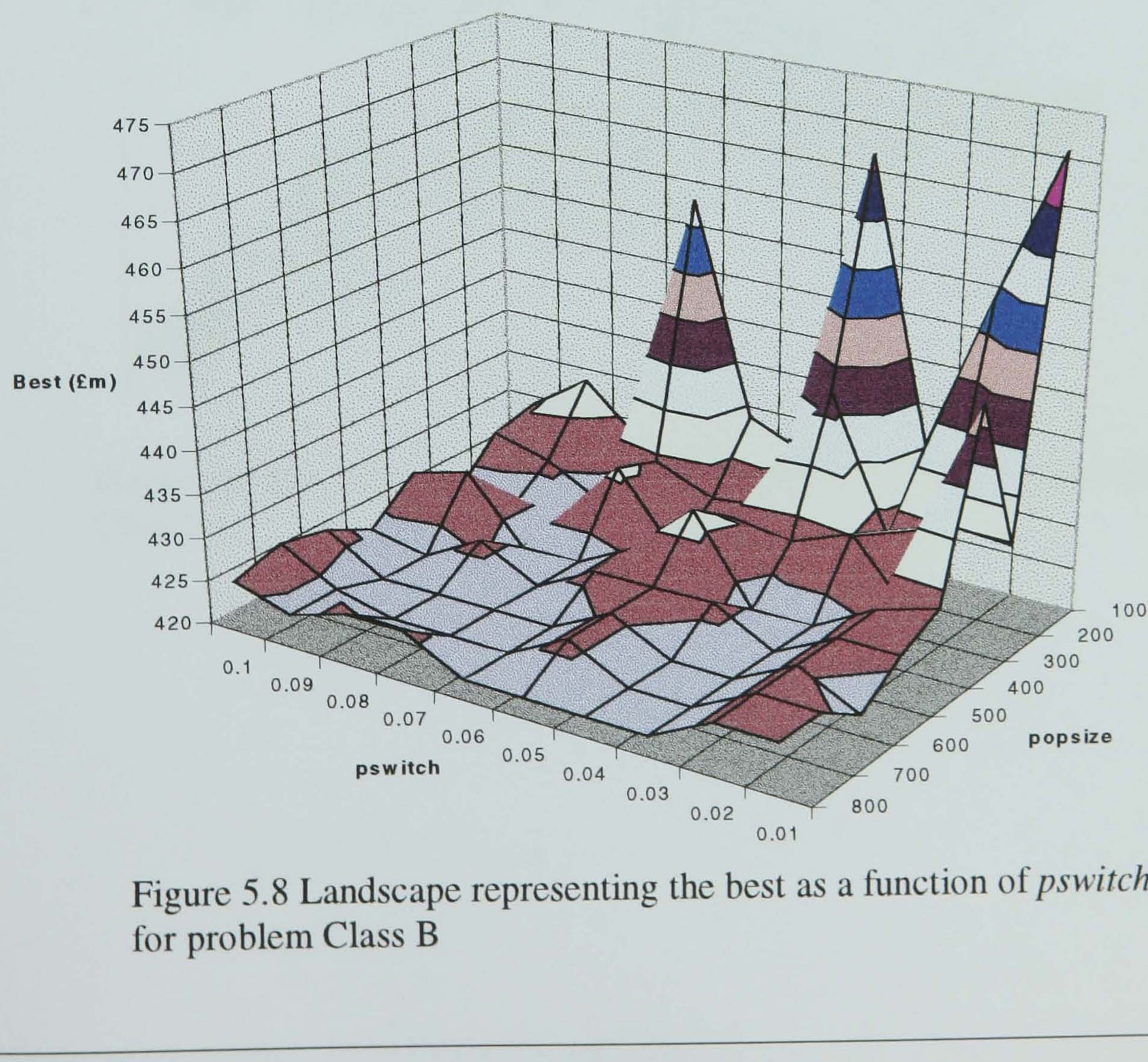


Figure 5.8 Landscape representing the best as a function of *pswitch* and *popsize* for problem Class B

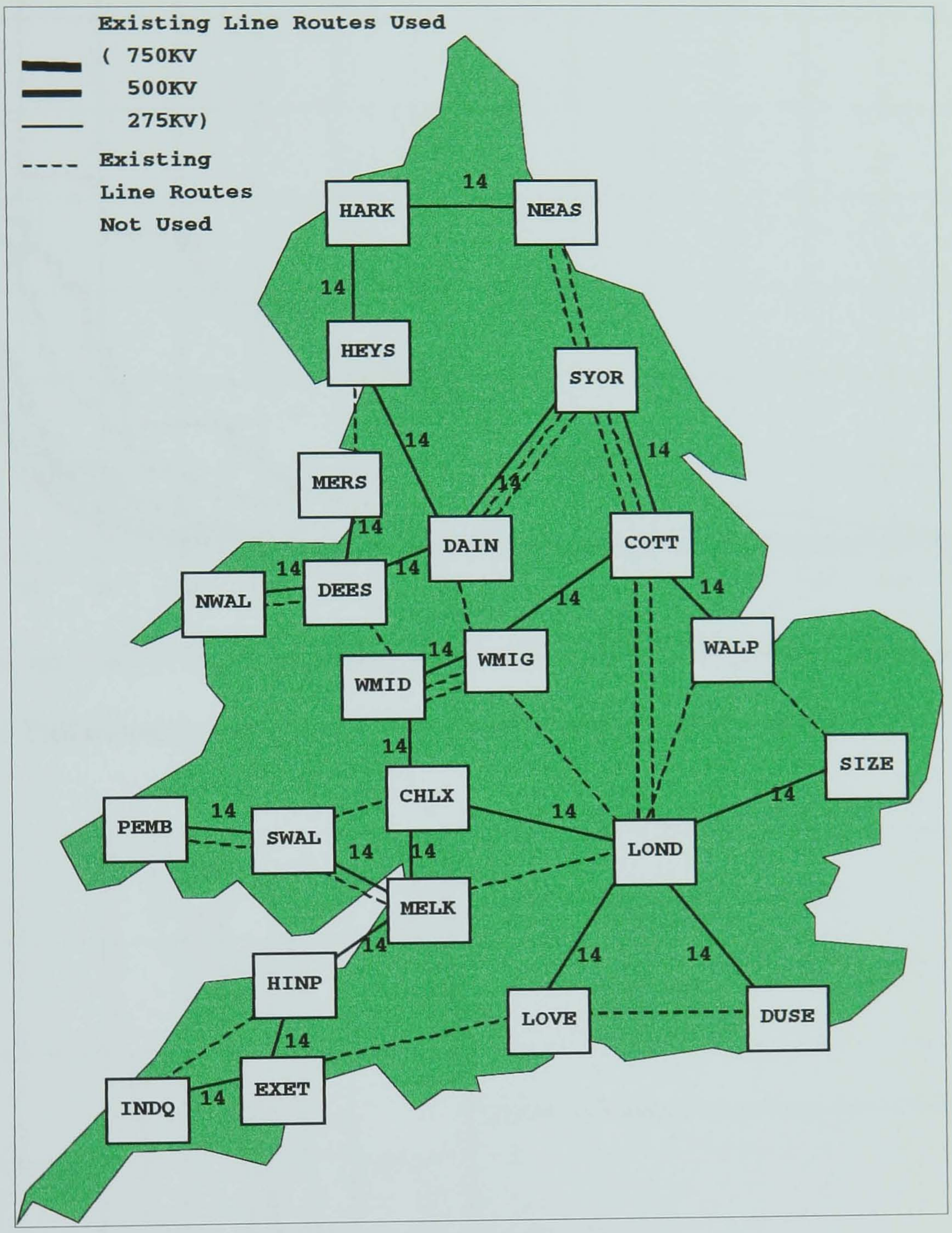
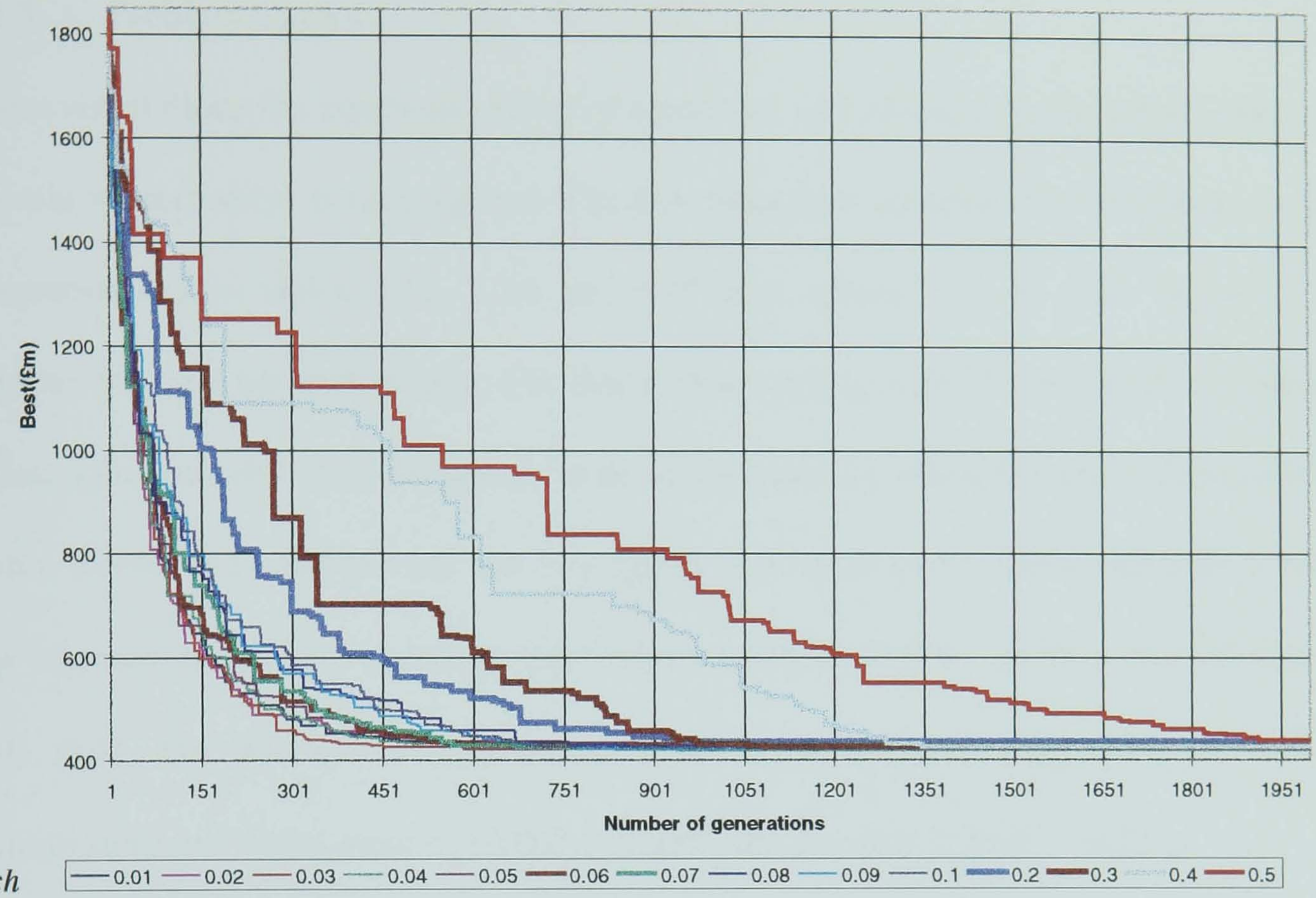


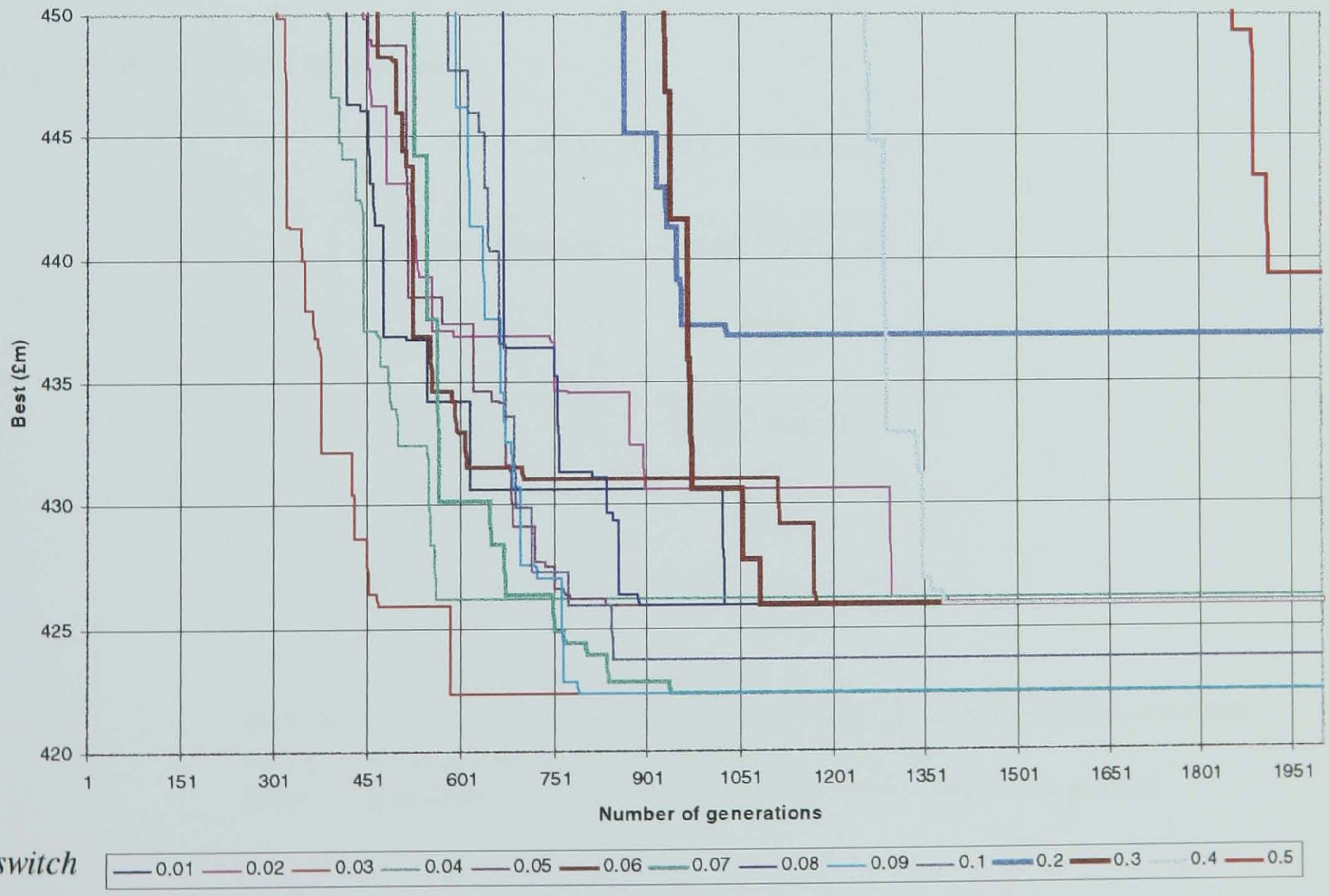
Figure 5.9 The optimum network design obtained by DCGA for problem Class B

pswitch



a) Full convergence graph (best in the range £m400..£m1800)

pswitch



b) A magnified version of the convergence graph (best in the range £420m-£450m)

Figure 5.10 Convergence graph for class B for *pmut*=0.004 and *pswitch* ∈ {0.01…0.5}

5.7.2.3 Problem Class C

In these simulations the combined effect of crossover and mutation probabilities (*pswitch* and *pmut* respectively) is investigated. The GA parameter settings and simulation results are summarised in tables 5.13, 5.14 and 5.15 and figures 5.11 to 5.13. Figure 5.11 illustrates the best network design. For this realistic problem (with no known optimum), the best solutions obtained did appear to be of high quality when inspected manually by experts. Tables 5.14 shows that the best (proposed optimum) is obtained with a wide range of *pmut* and *pswitch* settings, particularly *pmut* $\in$ {0.003...0.009} and *pswitch* $\in$ {0.01...0.4}, but different computational time as can be seen from table 5.15. The optimum setting is when *pmut* $\in$ {0.003...0.004} and *pswitch* $\in$ {0.01...0.07}.

Figure 5.12 and 5.13 depict the effect of *pmut* and *pswitch* on the performance of DCGA. It is noticeable how the DCGA is very reliable, relatively ease to get the best solutions in the range of parameters specified.

Table 5.13 GA parameter settings for class C

GA parameter settings (class C)	
Popsize	500
iseed	978456333
maxgen	2000
ptest	0.5

popsize is population size, *iseed* is responsible for the randomisation process
maxgen is maximum number of generations of a run,
ptest is probability for crossover applied on first bits of the parents.

Table 5.14 GA Parameter settings (*pmut* & *pswitch*) and simulation results for Class C

Best (£m) (class C)							
pmut-->	0.003	0.004	0.005	0.006	0.007	0.008	0.009
pswitch							
0.01	1959.729	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**
0.02	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**
0.03	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**
0.04	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**
0.05	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**
0.06	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**
0.07	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**
0.08	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**
0.09	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**
0.1	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**
0.2	**1942.698**	**1942.698**	**1942.698**	**1942.698**	**1942.698**	1953.031	**1942.698**
0.3	**1942.698**	**1942.698**	**1942.698**	1945.195	**1942.698**	**1942.698**	1944.648
0.4	**1942.698**	**1942.698**	**1942.698**	1963.065	1965.583	1948.691	1988.463
0.5	1959.729	1959.729	1967.02	1980.112	1972.515	2009.477	2030.096

pmut is the probability of mutation,
pswitch is the probability of crossover

Table 5.15. Number of generations needed to converge to the best so far for Class C

Number of generations needed to converge (class C)							
pmut-->	0.003	0.004	0.005	0.006	0.007	0.008	0.009
pswitch							
0.01	-	489	792	713	1190	1075	1093
0.02	315	388	1092	807	757	1232	1426
0.03	504	534	741	560	606	834	971
0.04	504	551	511	597	706	1044	1193
0.05	508	629	597	847	785	999	1276
0.06	559	638	688	846	652	1004	959
0.07	528	494	740	691	1038	1019	1257
0.08	608	593	1200	1051	1156	1412	1297
0.09	622	518	621	690	1312	1099	1581
0.1	749	791	723	812	1384	1096	1434
0.2	1539	931	1252	1431	1624	-	1912
0.3	1360	1595	1400	-	1921	1973	-
0.4	1521	1495	1943	-	-	-	-
0.5	-	-	-	-	-	-	-
Run-time (min) ≅ 41 for 2000 generations							

pmut is the probability of mutation
pswitch is the probability of crossover

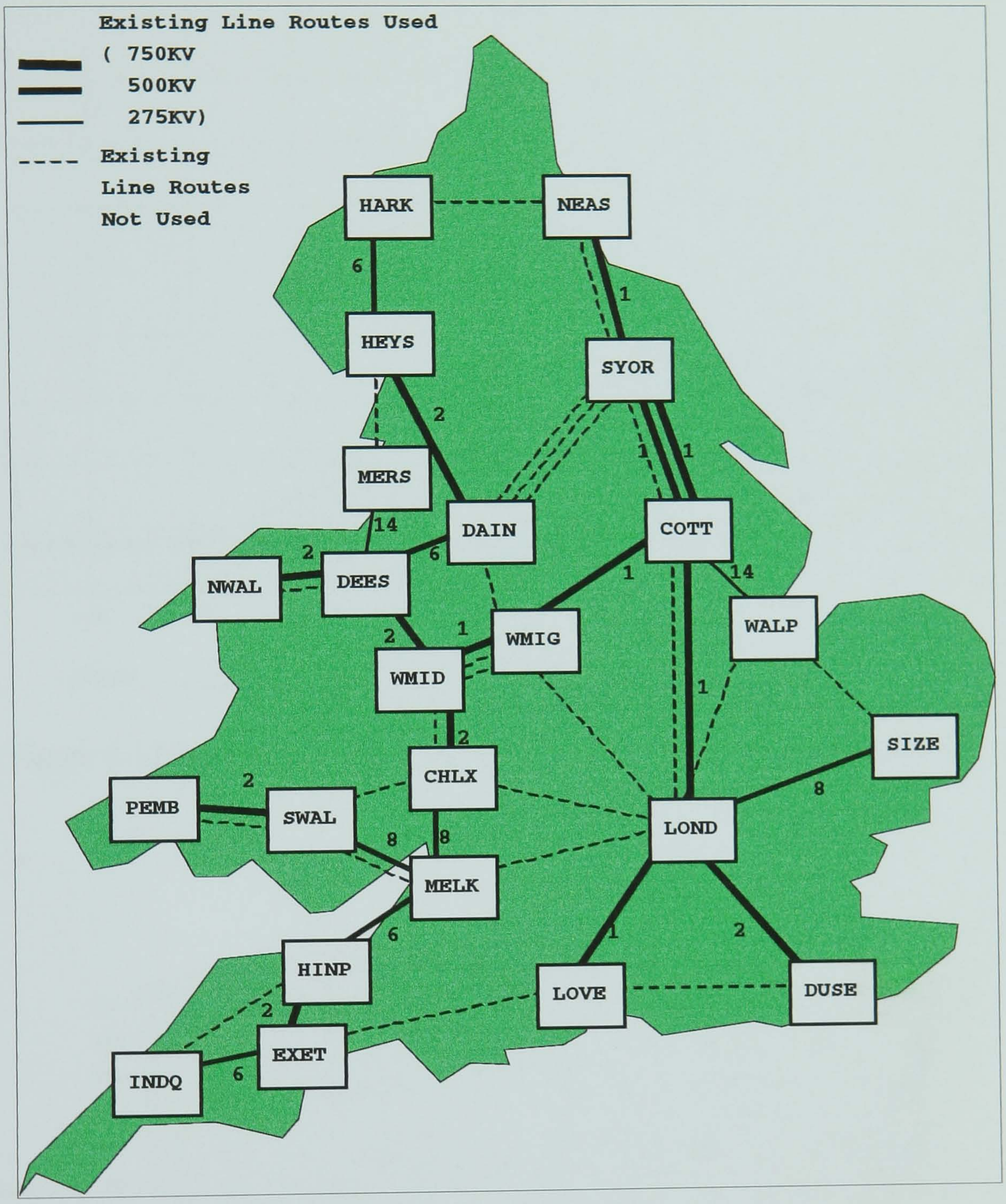


Figure 5.11 The optimum network design obtained by DCGA for problem Class C

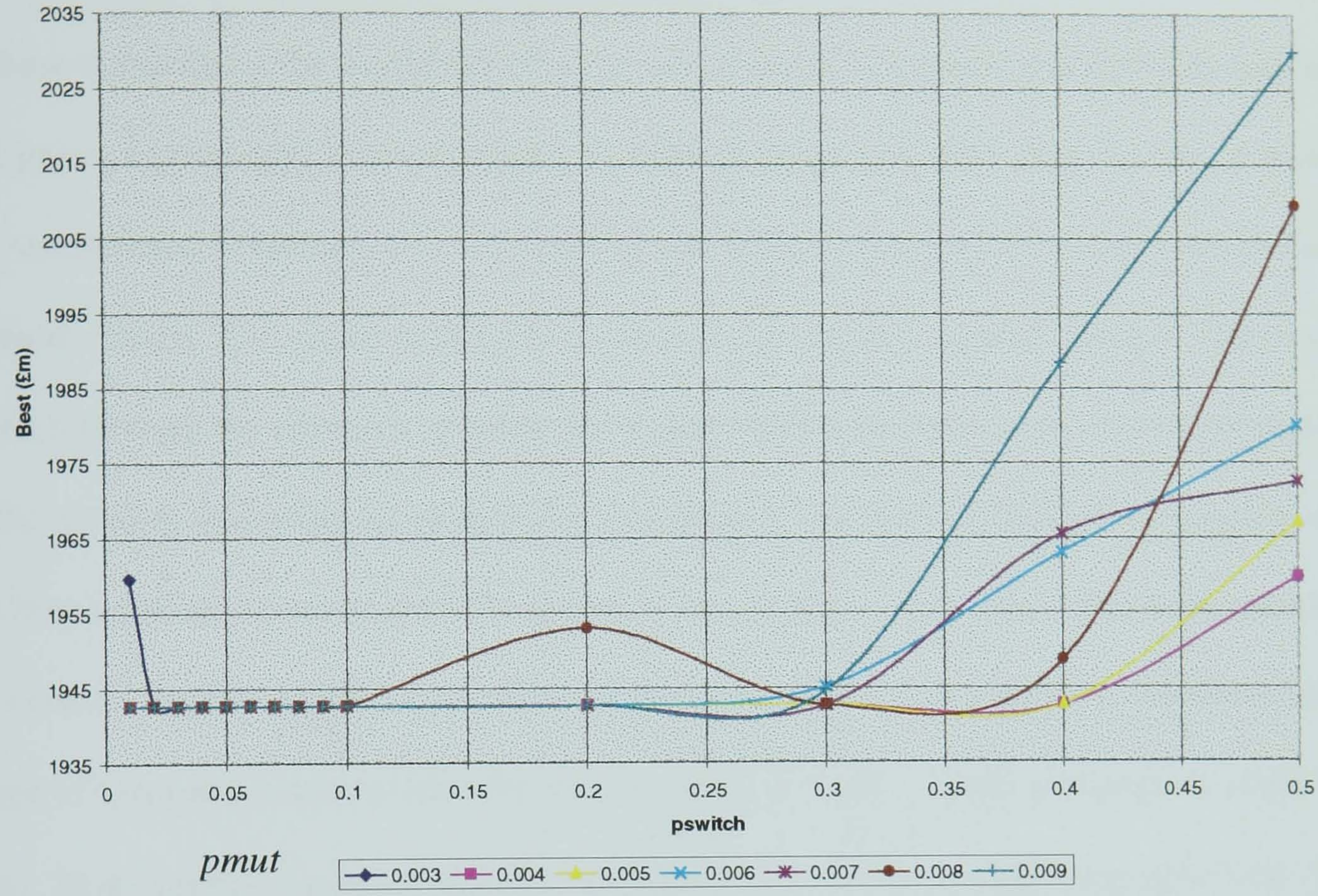


Figure 5.12 Best as a function of *pswitch* and *pmut* for problem class C

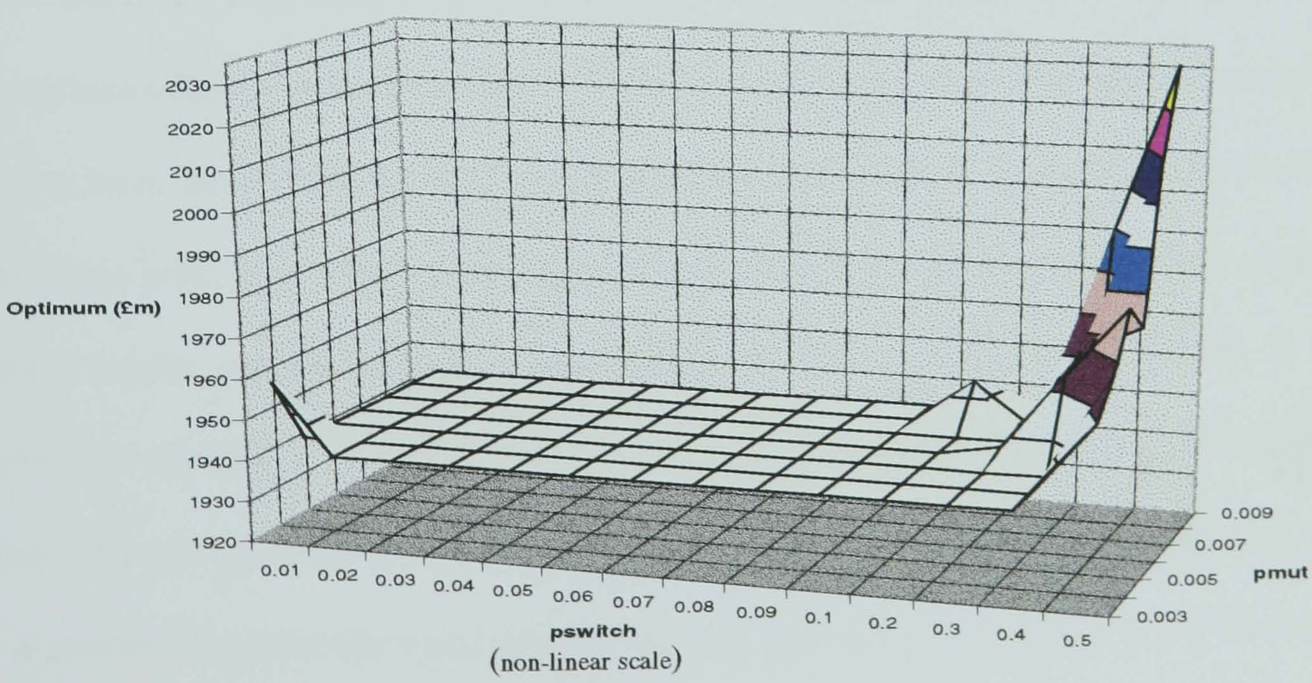


Figure 5.13 Landscape representing the best as a function of *pswitch* and *popsize* for problem Class C

5.7.2.4 Problem Class D

In these simulations the combined effect of crossover and mutation probabilities (*pswitch* and *pmut* respectively) is investigated. The GA parameter settings and simulation results are summarised in tables 5.16 - 5.18 and figures 5.14-5.16. Figure 5.14 illustrates the best network design. It is apparent that the network is 'n-1 secure' against outages. This is can be seen through the presence of loops which ensure the connectivity of load or generation to the network following an outage. For this realistic problem (with no known optimum), the best solutions obtained also did appear to be of high quality when inspected manually by experts. Figures 15-16 suggest that the best is obtained with a narrow and intermittent range of GA parameters particularly when *pswitch* ∈ {0.04...0.08} and *pmut* ∈ {0.004... 009}. However the computational time is different as can be implied from table 5.18. The optimum setting is *pmut* =0.004 and *pswitch* =0.06.

Therefore for this realistic problem, the GA required a fine-tuning of GA parameters in order to obtain the best results. In addition to providing the best, the DCGA provides other solutions (slightly more expensive) which can be of interest to the planner engineer. More tests have been carried out to investigate the effect of *iseed* responsible for the randomisation process on the performance of the DCGA. The GA parameter settings and simulation results are reported in tables 5.19, and 5.20 and figures 5.17-5.18. It is noticeable that the GA is still able to locate the best with different initial population but also with different parameter settings. The results obtained emphasises the effectiveness of the algorithm but also the need to tune the GA parameters in order to get the best results.

Table 5.16 GA parameter settings for class D

GA parameter settings (class D)	
Popsize	500
iseed	978456333
maxgen	2000
ptest	0.5

Popsize is population size, *iseed* is responsible for the randomisation process
Maxgen is maximum number of generations of a run,
ptest is probability for crossover applied on first bits of the parents.

Table 5.17 GA Parameter settings (*pmut* & *pswitch*) for Class D

Best (£m) (classD)							
Pmut→	0.003	0.004	0.005	0.006	0.007	0.008	0.009
pswitch							
0.01	2265.952	2262.158	2265.377	2268.574	2262.34	2262.158	2265.377
0.02	2271.995	2265.377	2264.781	2269.433	2262.158	2269.17	2262.603
0.03	2262.158	2262.158	2262.158	2265.952	2262.158	2262.158	2262.158
0.04	2262.158	2262.158	2262.158	**2261.537**	2262.158	2262.158	2262.158
0.05	2265.377	2262.158	2262.158	2262.158	2262.158	2262.158	**2261.537**
0.06	2262.158	**2261.537**	2271.995	2271.995	**2261.537**	2262.545	2262.158
0.07	2262.158	**2261.537**	**2261.537**	2262.158	2262.158	2262.158	2262.545
0.08	2262.158	2262.158	2262.158	2262.158	**2261.537**	2262.158	2262.158
0.09	2262.158	2262.158	2262.158	2262.158	2262.158	2262.545	2262.603
0.1	2262.158	2262.158	2262.158	2262.158	2262.158	2262.158	2262.158
0.2	2262.158	2266.397	2262.158	2275.459	2267.522	2270.038	2277.477
0.3	2267.522	2274.981	2285.377	2272.066	2274.29	2308.377	2292.257
0.4	2269.549	2278.585	2290.583	2293.089	2293.625	2318.006	2295.12
0.5	2284.051	2299.145	2296.9	2289.413	2334.029	2310.895	2356.677

pmut is the probability of mutation
pswitch is the probability of crossover

Table 5.18 Number of generations needed to converge for Class D

Number of generations needed to converge (class D)							
Pmut→	0.003	0.004	0.005	0.006	0.007	0.008	0.009
pswitch							
0.01	-	-	-	-	-	-	-
0.02	-	-	-	-	-	-	-
0.03	-	-	-	-	-	-	-
0.04	-	-	-	815	-	-	-
0.05	-	-	-	-	-	-	1531
0.06	-	579	-	-	1548	-	-
0.07	-	1024	1061	-	-	-	-
0.08	-	-	-	-	1822	-	-
0.09	-	-	-	-	-	-	-
0.1	-	-	-	-	-	-	-
0.2	-	-	-	-	-	-	-
0.3	-	-	-	-	-	-	-
0.4	-	-	-	-	-	-	-
0.5	-	-	-	-	-	-	-
Run-time (min) ≅ 86 for 2000 generations							

pmut is the probability of mutation
pswitch is the probability of crossover

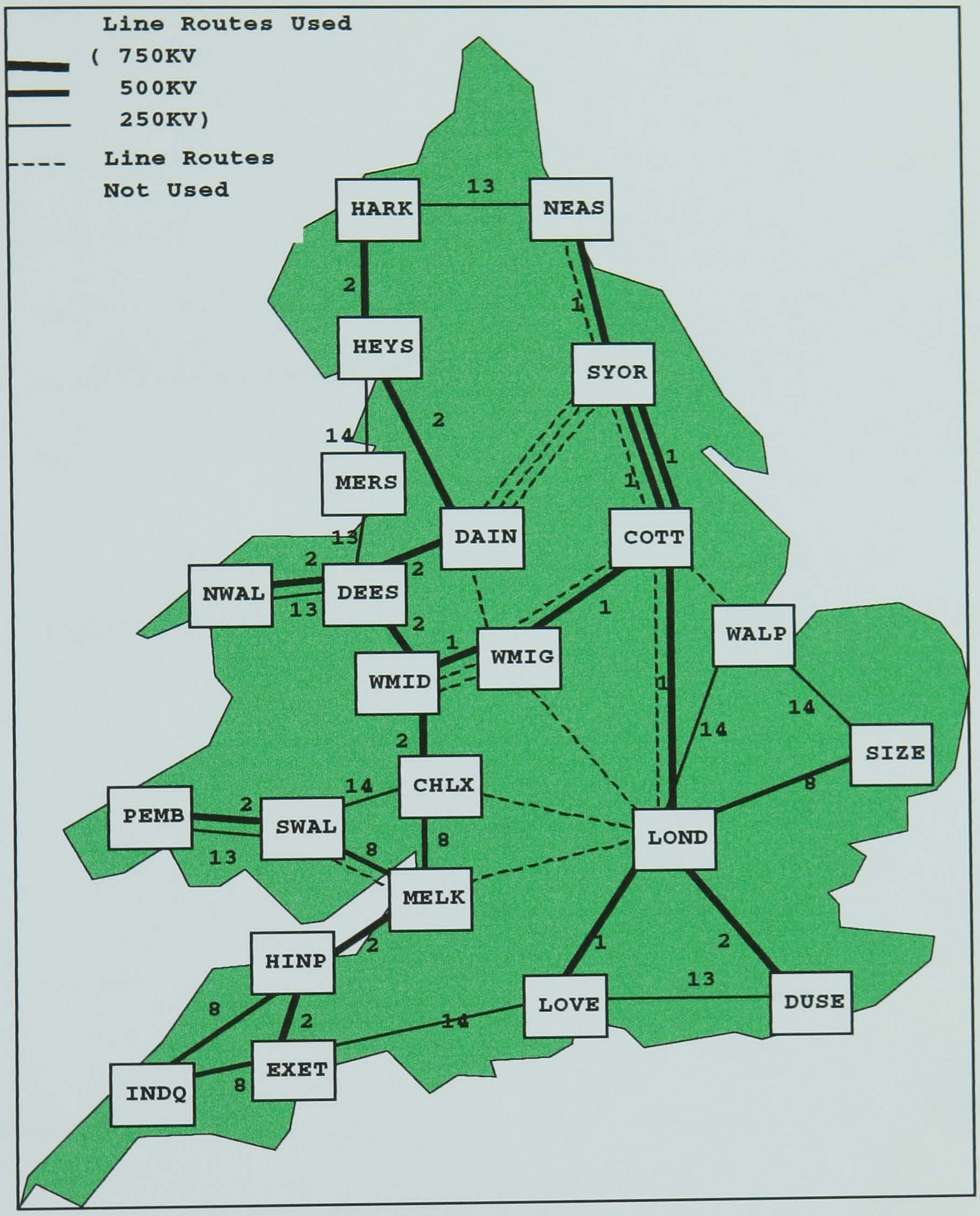


Figure 5.14 The optimum network design obtained by DCGA for problem Class D

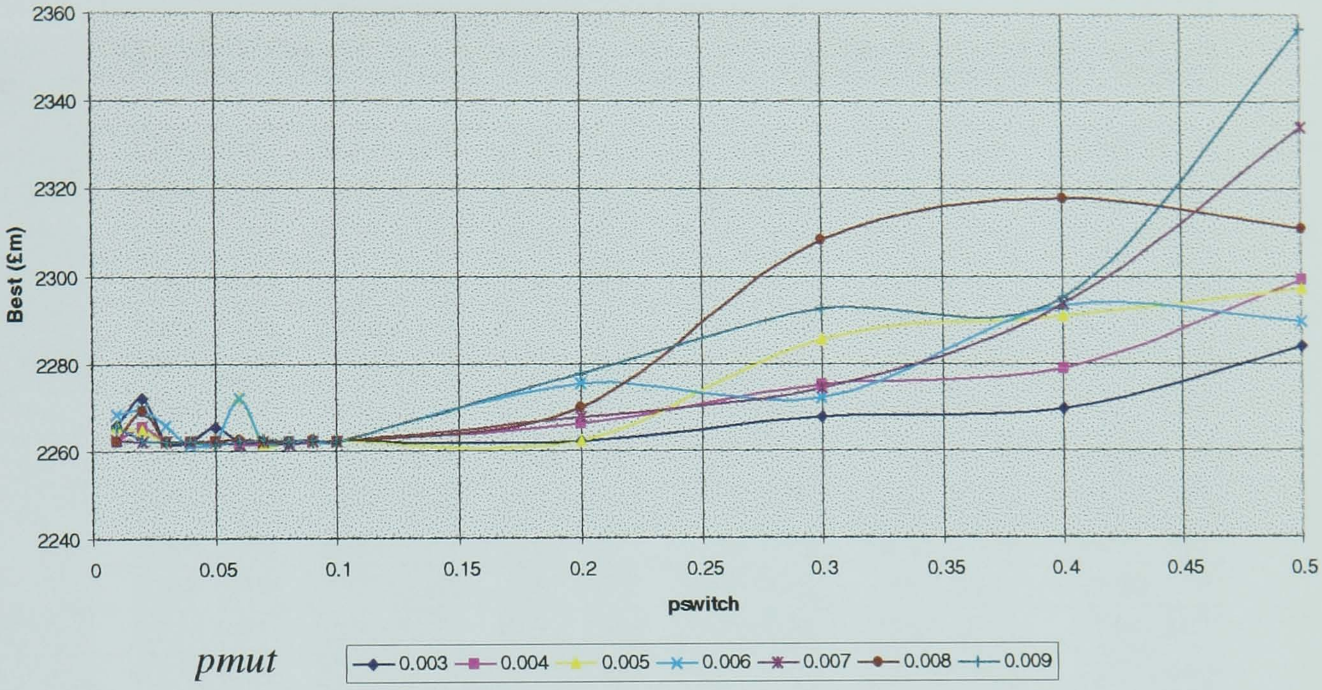


Figure 5.15 Best as a function of *pmut* and *pswitch* for problem Class D

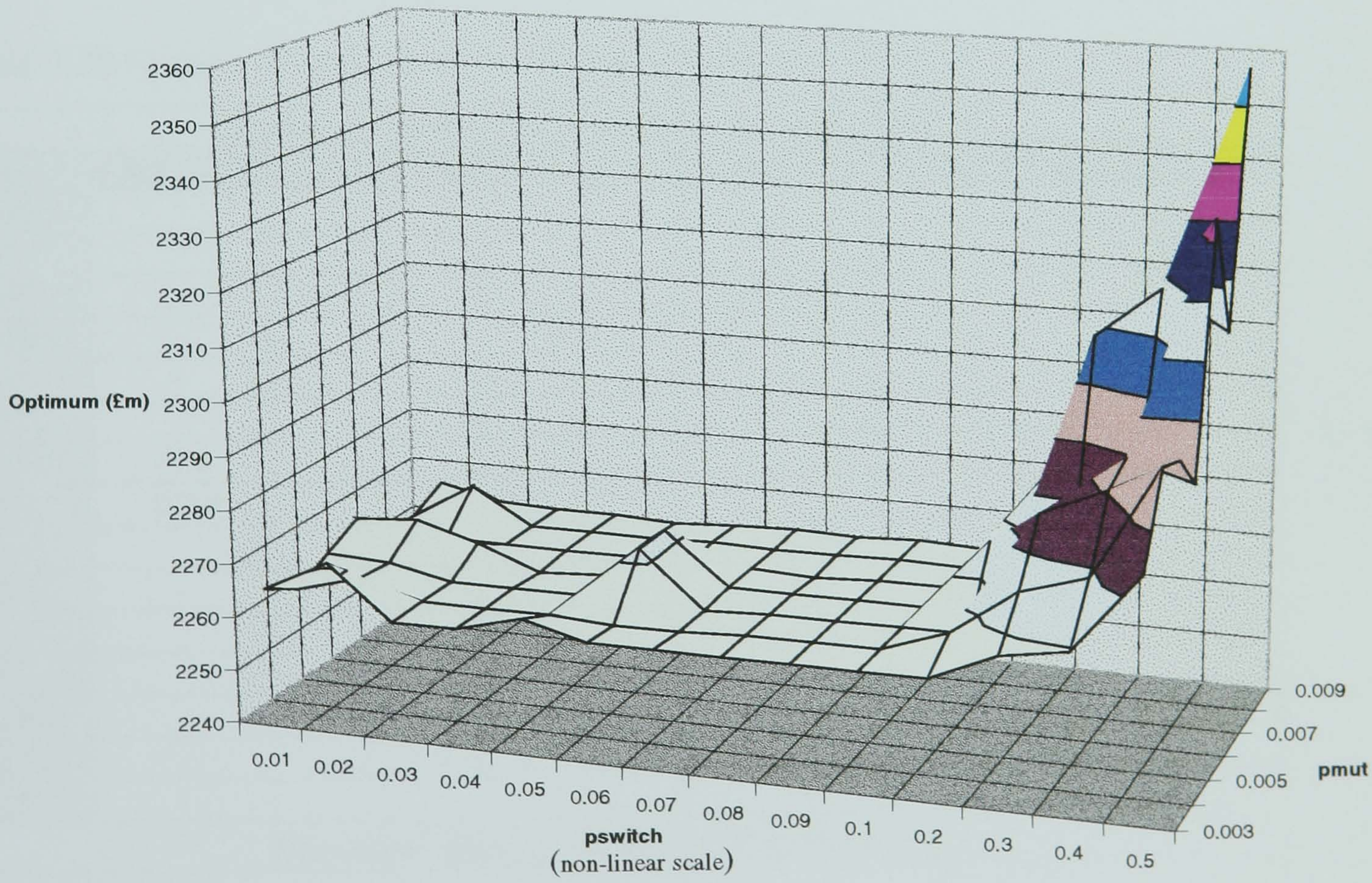


Figure 5.16 Landscape representing the best as a function of *pmut* and *pswitch* for problem Class D

Table 5.19 GA Parameter settings (*pswitch* & *iseed*) and simulation results for Class D

Best (£m) for problem class D						
iseed->	978456333	123456789	246802467	135791357	357991357	654321789
pswitch						
0.01	2262.158	2272.427	2264.781	2262.158	2271.995	**2261.537**
0.02	2265.377	2262.158	2262.158	**2261.537**	2265.952	2262.158
0.03	2262.158	2262.158	**2261.537**	2262.158	2262.158	2274.281
0.04	2262.158	2262.158	2264.781	2265.377	2262.158	2262.158
0.05	2262.158	2262.158	2262.158	2262.158	2262.158	2262.158
0.06	**2261.537**	2269.17	2271.341	2262.158	2262.158	2262.158
0.07	**2261.537**	2265.377	2262.158	2273.662	2262.158	2262.158
0.08	2262.158	2262.158	2262.158	2262.158	2262.158	2262.158
0.09	2262.158	2262.158	2262.158	2262.158	**2261.537**	2271.995
0.1	2262.158	2262.158	2262.158	2262.158	2262.158	2274.281
0.2	2266.397	2265.952	2267.655	2267.655	2267.522	2262.158
0.3	2274.981	2283.675	2279.739	2279.739	2265.952	2266.397
0.4	2278.585	2280.013	2275.718	2275.718	2279.739	2271.449
0.5	2299.145	2279.729	2281.271	2281.271	2284.674	2273.803

iseed is responsible for the randomisation process
pswitch is the probability of crossover

Table 5.20 Number of generations needed to converge for Class D

Number of generation needed to converge (class D)						
iseed-->	978456333	123456789	246802467	135791357	357991357	654321789
pswitch	-	-	-	-	-	
0.01	-	-	-	-	-	831
0.02	-	-	-	857	-	-
0.03	-	-	710	-	-	-
0.04	-	-	-	-	-	-
0.05	-	-	-	-	-	-
0.06	579	-	-	-	-	-
0.07	1024	-	-	-	-	-
0.08	-	-	-	-	-	-
0.09	-	-	-	-	1163	-
0.1	-	-	-	-	-	-
0.2	-	-	-	-	-	-
0.3	-	-	-	-	-	-
0.4	-	-	-	-	-	-
0.5	-	-	-	-	-	-
Run-time (min) ≅ 86 for 2000 generations						

iseed is responsible for the randomisation process
pswitch is the probability of crossover

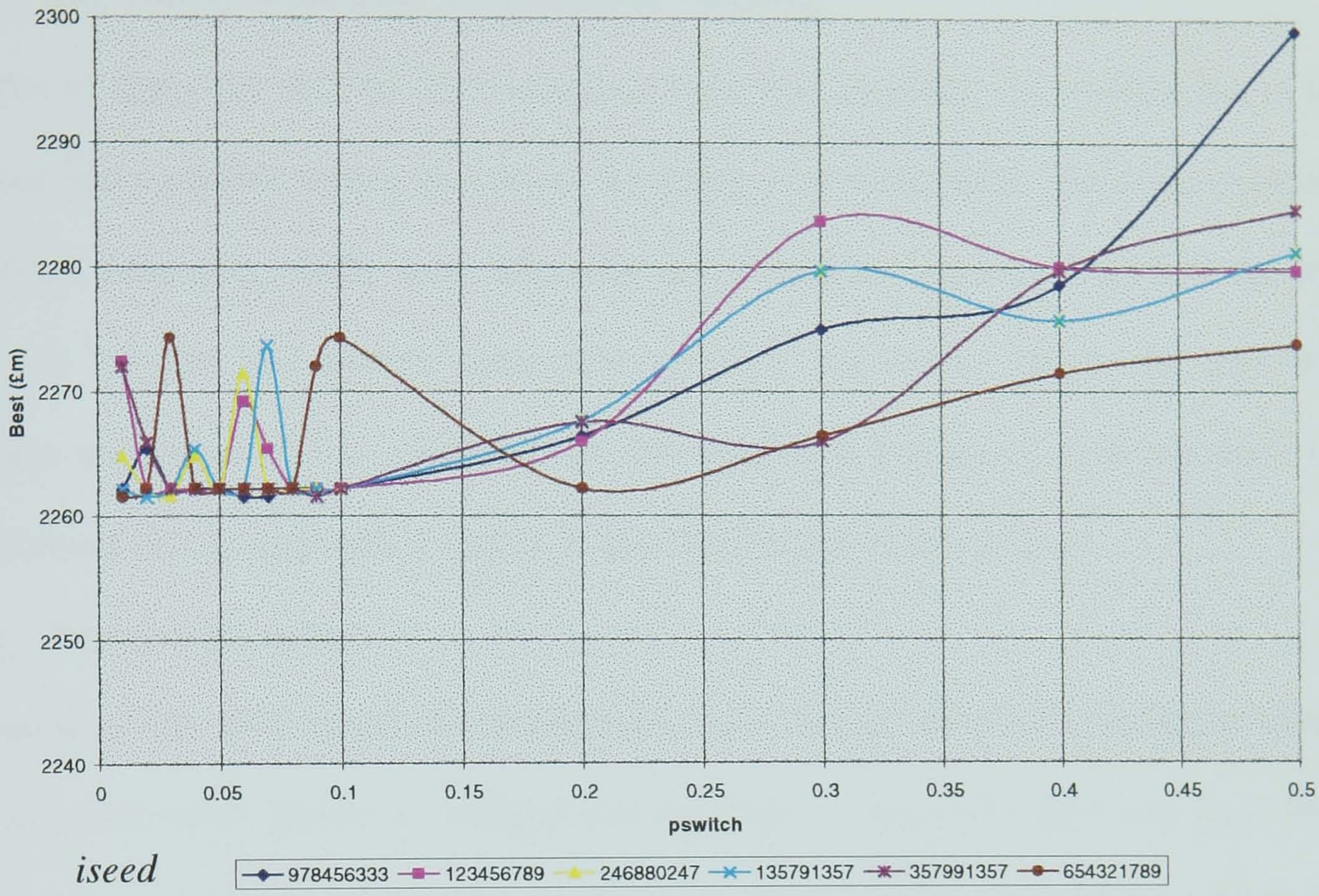


Figure 5.17 Best as a function of *pswitch* and *iseed* for problem Class D

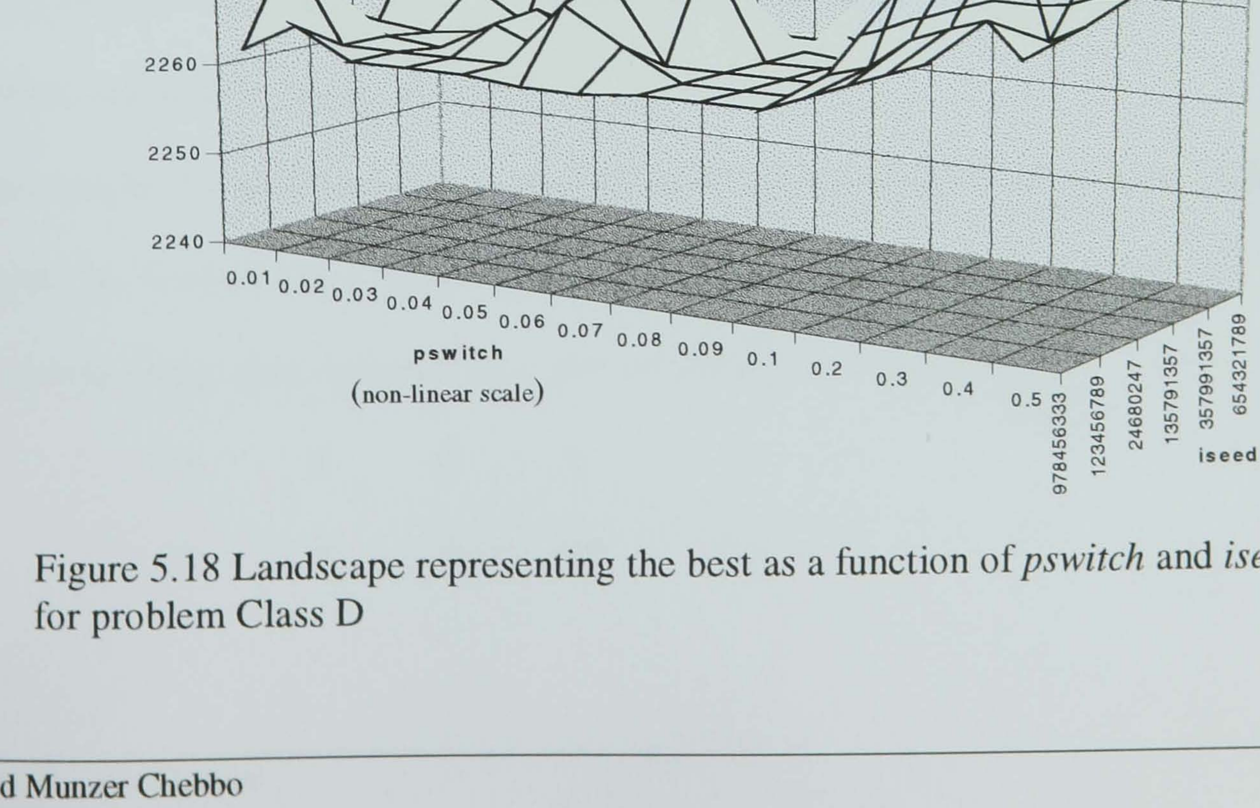


Figure 5.18 Landscape representing the best as a function of *pswitch* and *iseed* for problem Class D

5.8 Ant Colony Search for Transmission Planning

In order to apply the ACS to the transmission-planning problem, the problem has to be mapped into a suitable format that allows the application of the ACS. This is the problem encoding. Moreover, ACS uses indirectly an evaluation function (inspired by the objective function and system constraints) to guide the search. Hence the objective function and some of the constraints of the transmission-planning problem must be transformed into an appropriate fitness as described earlier in this chapter.

5.8.1 Problem Encoding

Before using the ACS model, the transmission network with all-possible routes (*nroute*) and available types (*ntype*) is mapped into a (nroute x (*ntype+1*)) grid (see fig.5.19), where *nroute* and *ntype* represent respectively the vertical and horizontal axis. Then, *m* artificial ants much like those used in the TSP application [Dorigo, Maniezzo, and Colorni, 1996] are recruited to search for good solutions. Each ant corresponds to a string of integers, and represents a given transmission network design.

Every route is represented by an integer number, which represents the line type of that route. In the test model presented earlier, 14 actual line types are available (supplemented by line type 0, to represent an unused route) including various single-circuit and double-circuit line types.

The integers for each route are concatenated to form an overall string which represents a particular network design, and which is a member of the ant population.

A typical string (ant) representing a network design is as follows:

$$\begin{matrix} 0 & 3 & 6 \; \ldots & 9 & \ldots & 14 \\ T_1 & T_2 & T_3 \; \ldots & T_i & \ldots & T_{nroute} \end{matrix}$$

Where

T_i	represent line types in route i
type zero	represents no line in the corresponding route.
nroute	possible routes or wayleaves

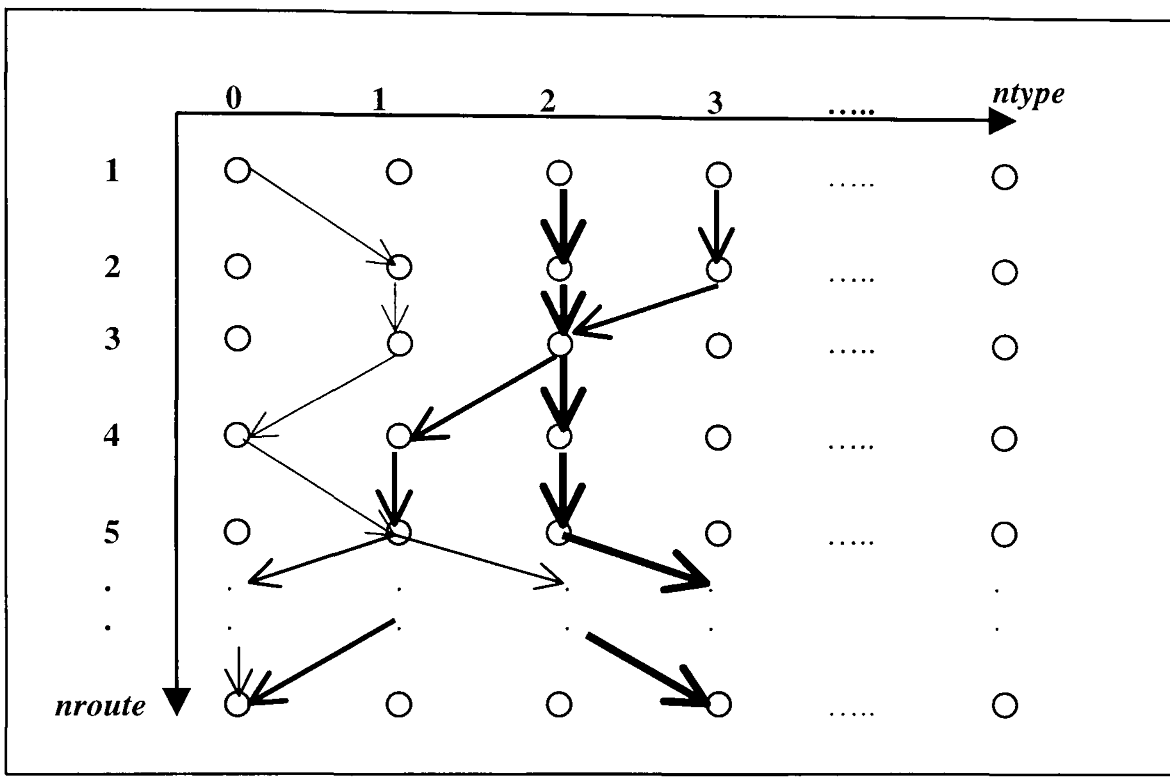


Figure 5.19 Problem mapping: transmission network with all possible routes to (*nroute* x *ntype*+1) grid.

Ant paths (possible networks)

O Node (possible assignment of a type to a route)

5.8.2 Simulation Results

Several experiments that consumed several months of CPU time have been carried out on the 23-bus NGC system described earlier. These tests only consider problems classes A and B. The reason is that the computational time increases with the complexity of the problem, and the ACS appears to be less efficient than the GA.

Only the effect of various combination of the important ACS parameters (α, β, *iseed*, and *m*) have been considered. The convergence criterion is chosen so that a run is terminated after a fixed number of cycles and the best obtained so far is recorded. The number of cycles could be different for the various optimisation problems considered. Table 5.21 reports on the organisation of simulation results (tables and figures) corresponding to the range of problem classes considered and the various combinations of ACS parameters adopted.

For set values of *m*, *iseed* and the remaining parameters, the ACS program is executed for a range of values of α and β. The best results are illustrated in graphs. Every graph is associated with a certain value of β and shows the variation of the best as α increases and corresponds to a series of ACS runs that are generated automatically in the main program Adopting the same *iseed* and the optimum setting of β of previous runs, the ACS is again run for a range of *m* and α values. For those tests, every graph is associated with a certain value of *m* and shows the variation of the best as α increases The optimum setting of β corresponds to the least computational time to achieve the optimum solution. More tests are also carried out by setting *m* and β to the same values considered in previous runs, and running ACS program for a range of *iseed* and α values. In this case, every graph is associated with a certain value of *iseed* and shows the variation of the best as α increases.

Table 5.21 Classification of simulation results corresponding to the range of problems considered and the various combinations of ACS parameters adopted

	Class A	**Class B**
	Figures 5.4 (optimum)	Figure 5.9 (optimum)
α and β	Tables 5.22-5.24 Figures 5.20-5.21	Tables 5.25-27 Figures 5.22-5.23 Figure 5.28 (convergence graph)
α and *m*	-	Tables 5.28-5.29 Figures 5.24-5.25
α and *iseed*	-	Tables 5.30-5.31 Figures 5.26-5.27

5.8.2.1 Problem Class A

The ACS parameter settings and the convergence criteria for this optimisation problem are shown in table 5.22. These parameters remain constant throughout a simulation. Only the variations of α and β are considered in these experiments. Simulation results are summarised in tables 5.23 and 5.24 and illustrated in figures 5.20 and 5.21. Table 5.23 shows that the optimum is obtained with different settings of α and β especially with α in the range 0.5 to 0.75 and β in the range 3 to 9 . The optimum thus occurs in a narrow range with respect to α (which has a direct influence on the performance of the ACS algorithm). However, the computational time is different as can be seen in table 5.24; with the optimum settings the optimum can be obtained very fast (in less than a second). The results obtained in this experiment are consistent with the characteristic of the algorithm. A high value of α (α > 0.75) means that trail is very important and therefore ants tend to choose paths chosen by other ants in the past which might force the algorithm to enter stagnation behaviour very quickly without finding very good solutions. On the other hand, if enough importance was not given to the trail ($\alpha < 0.5$) the algorithm becomes very similar to a stochastic multigreedy algorithm and did not find very good solutions.

Figures 5.21 suggest that the landscape of the loss problem is smooth and the algorithm is more sensitive to α than β.

Table 5.22 Parameter settings for Problem Class A

ACS Parameter settings	
ctetrail	0.5
ctevisib	1
Q	100
elite	1
m	15
ρ	0.8
iseed	123456789
maxcycle	500

ctetrail is initial value of phermone trail
ctevisib is initial value of visibility
Q is a constant quantity per unit length of pheromone trail laid by the ant
elite number of ants reinforcing the trail on the best path of a cycle
ρ is the persistence of the trail
m is the number of ants
iseed is responsible for the randomisation process
maxcycle is the maximum number of cycles for a run.

Table 5.23 Parameter settings (α & β) and simulation results for Class A

Best (£m) obtained after 500 cycles (class A)							
β-->	3	4	5	6	7	8	9
α							
0	279.888	260.824	249.103	244.935	242.736	240.906	239.914
0.05	267.272	252.144	247.808	242.736	241.858	239.979	239.768
0.1	263.994	250.28	243.734	241.006	239.979	239.237	239.166
0.15	258.81	245.977	242.789	239.979	239.561	238.899	238.901
0.2	252.717	242.706	240.554	239.164	239.166	238.901	238.901
0.25	246.553	241.866	239.164	239.158	239.112	238.659	238.453
0.3	242.275	239.883	238.999	238.912	238.632	238.378	238.281
0.35	240.75	239.042	238.73	238.581	238.47	238.302	238.143
0.4	239.199	238.839	238.515	238.416	238.183	238.118	238.149
0.45	238.777	238.686	238.362	238.149	238.119	238.118	238.117
0.5	238.216	238.335	238.117	238.117	238.113	238.113	238.117
0.55	238.375	238.207	238.213	**238.112**	**238.112**	**238.112**	**238.112**
0.6	**238.112**	238.149	**238.112**	**238.112**	**238.112**	**238.112**	238.117
0.65	238.147	238.116	238.14	**238.112**	**238.112**	238.116	238.113
0.7	238.143	**238.112**	238.116	238.14	238.143	238.113	**238.112**
0.75	238.143	238.366	238.113	238.328	238.113	**238.112**	**238.112**
0.8	238.574	238.116	238.328	238.113	238.113	238.113	238.113
0.85	238.116	238.366	238.147	238.147	238.147	238.143	238.147
0.9	238.146	238.573	238.5	238.14	238.14	238.14	238.14
0.95	238.14	238.525	238.524	238.163	238.163	238.163	238.163

Bold numbers represent the **optimum** which can obtained by different combination of α and β.
α and β are parameters that control the relative importance of trail versus visibility

Table 5.24. Convergence cycles needed for the optimum combinations of α & β for Class A

Number of cycles needed to converge (class A)							
β-->	3	4	5	6	7	8	9
α							
0	-	-	-	-	-	-	-
0.05	-	-	-	-	-	-	-
0.1	-	-	-	-	-	-	-
0.15	-	-	-	-	-	-	-
0.2	-	-	-	-	-	-	-
0.25	-	-	-	-	-	-	-
0.3	-	-	-	-	-	-	-
0.35	-	-	-	-	-	-	-
0.4	-	-	-	-	-	-	-
0.45	-	-	-	-	-	-	-
0.5	-	-	-	-	-	-	-
0.55	-	-	-	411	134	61	106
0.6	298	-	61	106	106	106	-
0.65	-	-	-	143	87	-	-
0.7	-	87	-	-	-	-	47
0.75	-	-	-	-	-	-	37
0.8	-	-	-	-	-	-	-
0.85	-	-	-	-	-	-	-
0.9	-	-	-	-	-	-	-
0.95	-	-	-	-	-	-	-
Run-time (sec) ≅ 57.02971 for 500 cycles							

α and β are parameters that control the relative importance of trail versus visibility

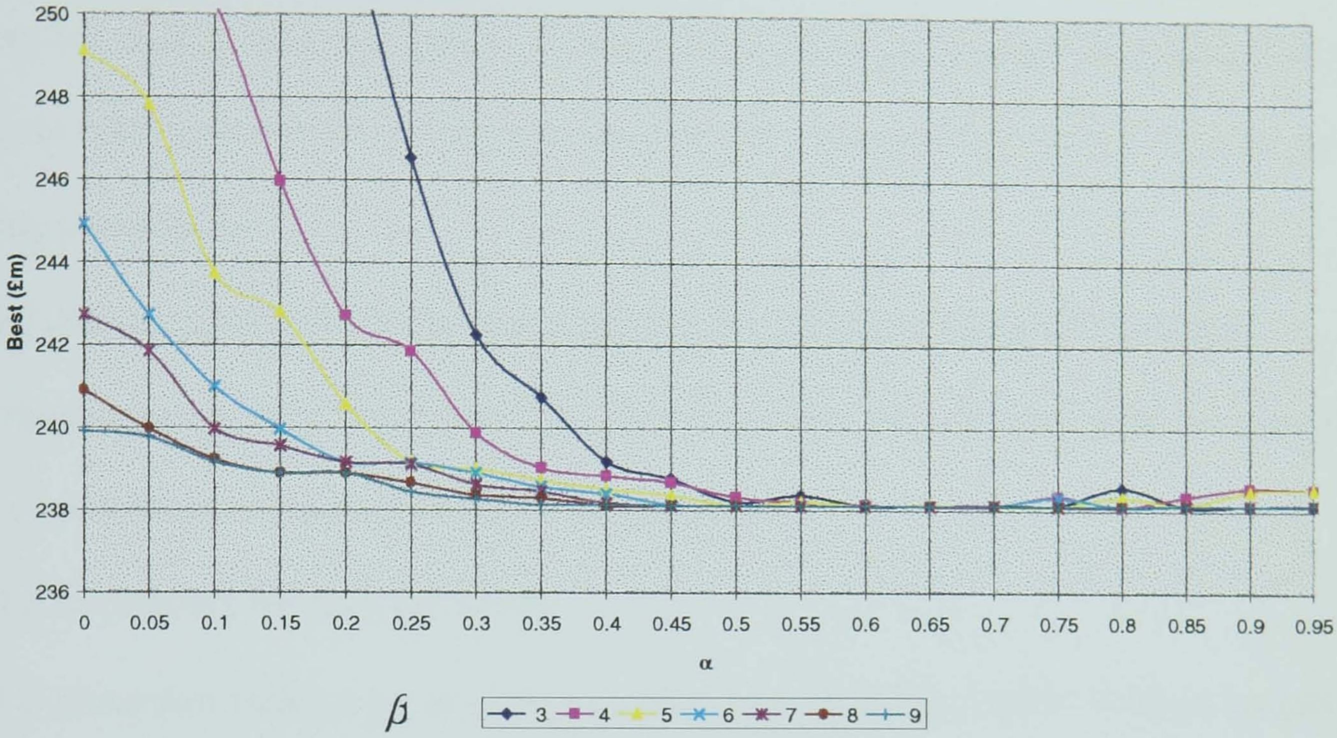


Figure 5.20 Best as a function of α and β for problem Class A

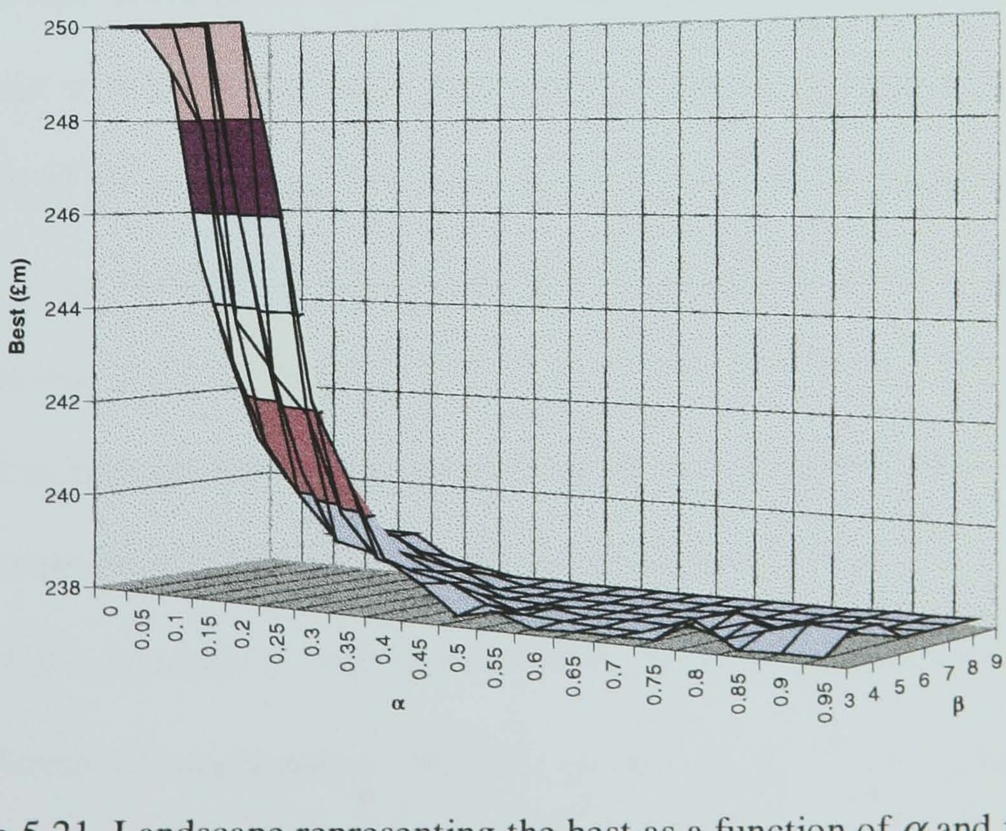


Figure 5.21. Landscape representing the best as a function of α and for problem Class A

5.8.2.2 Problem Class B

This is a relatively difficult problem due to the need for finding the minimum spanning tree network design with the cheapest design cost. More exhaustive tests have been carried out both to search for the optimum and to study the sensitivity of the ACS to the following parameters:

- *iseed* that is responsible for the randomisation process (initial distribution of ants),
- *m,* the number of ants, which has a direct influence on the intensity of the trail used for communication among ants (synergistic effect),
- and β responsible for the local search.

The ACS parameter settings are as shown in table 5.25. These parameter settings remain constant unless otherwise specified.

Figures 5.22 and 5.23 illustrate the performance of the ACS with respect to α and β parameters. The optimum can be obtained when α in the range 0.2 to 0.7 and β in the range 2.5 to 4 but with different computational times as reported in tables 5.26 and 5.27.

The effect of α and *m* on the performance of the ACS is depicted in figure 5.24 and 5.25. Tables 5.28 and 5.29 shows that the optimum can be obtained when α in the range 0.4-0.6 and *m* in the range 18-25. It is important to note the synergistic effect in using many ants and using the trail communication system on the performance of the ACS algorithm; that is, a run with *m* ants (enough ants) is more effective with communication among ants ($\alpha > 0$, α in the range 0.4-0.6) than with no communication ($\alpha = 0$).

Figures 5.26-5.27 also illustrate the performance of the ACS with respect to α and *iseed*. Starting with different initial population ACS was capable to identify the optimum when α in the range 0.35-0.7 and with all *iseed* considered (see table 5.30-5.31).

Simulation results showed that the ACS is more sensitive to α than the other parameters considered (β, *m* or *iseed*). They also showed that there is a slightly variable range of α whereby the optimum can be obtained if we tune the other parameters appropriately.

Figure 5.28 depicts the convergence graphs of a series of runs corresponding to $\beta = 4$ and α in the range {0.3...0.75} respectively. It is important to note that the best is improving rapidly during the first cycles of the evolution (exploration phase and reduction of the search space) and is improving slowly towards the end of the run (where the search space is reduced and the solutions are very similar). It can also be observed that the best remains constant for a number of cycles before it improves during the search for the optimum. Therefore the choice of the number of cycles is critical as it can affect the quality of the solution. This also implies that if the ACS is left running for more cycles the ACS will probably converge for other sets of parameters. Therefore, the landscape resulting from the application of ACS to problem class B would be smoother.

Table 5.25 Parameter Settings for Problem Class B

ACS Parameter settings	
ctetrail	0.5
ctevisib	1
Q	100
elite	1
m	20
ρ	0.8
iseed	123456789
β	3.5
maxcycle	500

ctetrail is initial value of pheromone trail
ctevisib is initial value of visibility
Q is a constant quantity per unit length of pheromone trail laid by the ant
elite number of ants reinforcing the trail on the best path of a cycle
ρ is the persistence of the trail
m is the number of ants
iseed is responsible for the randomisation process
maxcycle is the maximum number of cycles for a run.

Table 5.26 Parameter settings (α & β) and simulation results for Class B

Best (£m) (class B)							
β--->	2.5	3	3.5	4	4.5	5	5.5
α							
0	599.597	491.633	449.724	432.174	436.338	429.368	983.273
0.05	551.158	475.164	435.182	435.334	431.092	430.631	946.448
0.1	526.476	463.803	425.93	431.21	429.067	431.768	958.74
0.15	486.44	449.1	425.93	428.912	426.376	431.768	926.077
0.2	467.952	435.182	427.196	**422.362**	422.808	440.028	865.785
0.25	458.853	433.119	429.01	425.053	422.808	440.224	824.425
0.3	439.91	427.694	428.609	**422.362**	422.808	438.195	686.009
0.35	434.476	426.862	423.7	**422.362**	422.808	438.663	628.915
0.4	426.844	425.363	424.146	**422.362**	422.808	438.195	556.856
0.45	424.248	423.923	**422.362**	**422.362**	426.376	439.756	486.587
0.5	422.585	423.7	**422.362**	422.808	428.829	437.749	441.773
0.55	**422.362**	423.7	**422.362**	422.808	429.052	437.749	435.841
0.6	423.7	**422.362**	423.7	427.045	435.296	440.425	425.975
0.65	423.7	**422.362**	423.7	426.599	432.62	437.08	423.7
0.7	423.7	422.808	**422.362**	426.599	431.059	440.425	423.923
0.75	422.808	422.808	428.383	427.045	435.073	436.857	428.829
0.8	427.491	427.045	432.174	427.045	438.195	437.972	424.146
0.85	423.7	424.146	432.174	427.268	437.749	436.857	424.146
0.9	423.7	428.829	427.937	427.268	441.317	436.857	430.054
0.95	428.829	427.491	432.62	427.268	440.425	434.404	429.275

α and β are parameters that control the relative importance of trail versus visibility

Table 5.27 Convergence cycle needed for the optimum combinations of α & β for Class B

Number of cycle needed to converge (class B)							
β-->	2.5	3	3.5	4	4.5	5	5.5
α							
0	-	-	-	-	-	-	-
0.05	-	-	-	-	-	-	-
0.1	-	-	-	-	-	-	-
0.15	-	-	-	-	-	-	-
0.2	-	-	-	322	-	-	-
0.25	-	-	-	-	-	-	-
0.3	-	-	-	426	-	-	-
0.35	-	-	-	301	-	-	-
0.4	-	-	-	426	-	-	-
0.45	-	-	271	426	-	-	-
0.5	-	-	271	-	-	-	-
0.55	241	-	239	-	-	-	-
0.6	-	79	-	-	-	-	-
0.65	-	240	-	-	-	-	-
0.7	-	-	271	-	-	-	-
0.75	-	-	-	-	-	-	-
0.8	-	-	-	-	-	-	-
0.85	-	-	-	-	-	-	-
0.9	-	-	-	-	-	-	-
0.95	-	-	-	-	-	-	-
Run-time (min) ≅ 210							

α and β are parameters that control the relative importance of trail versus visibility

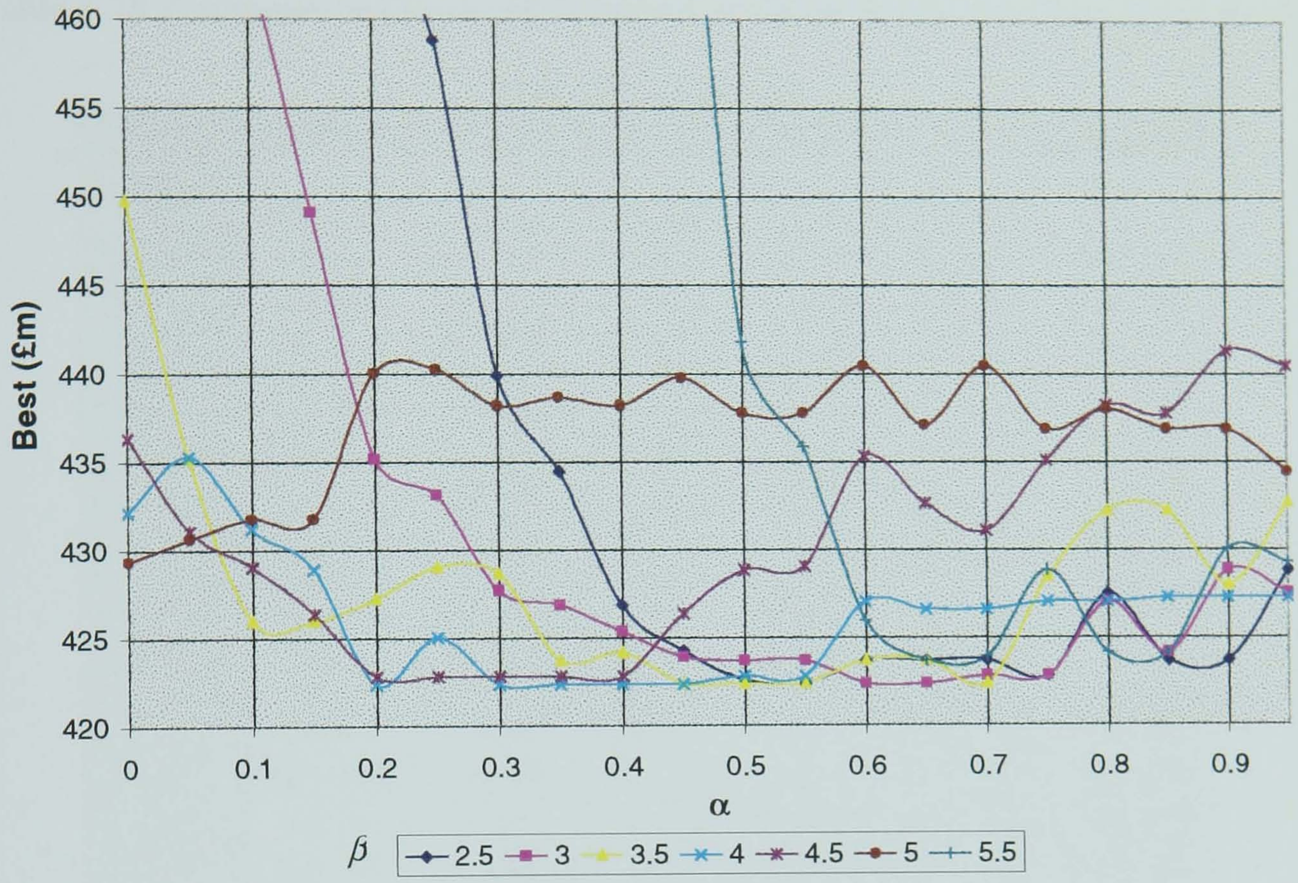


Figure 5.22 Best as a function of α *and* β for problem Class B

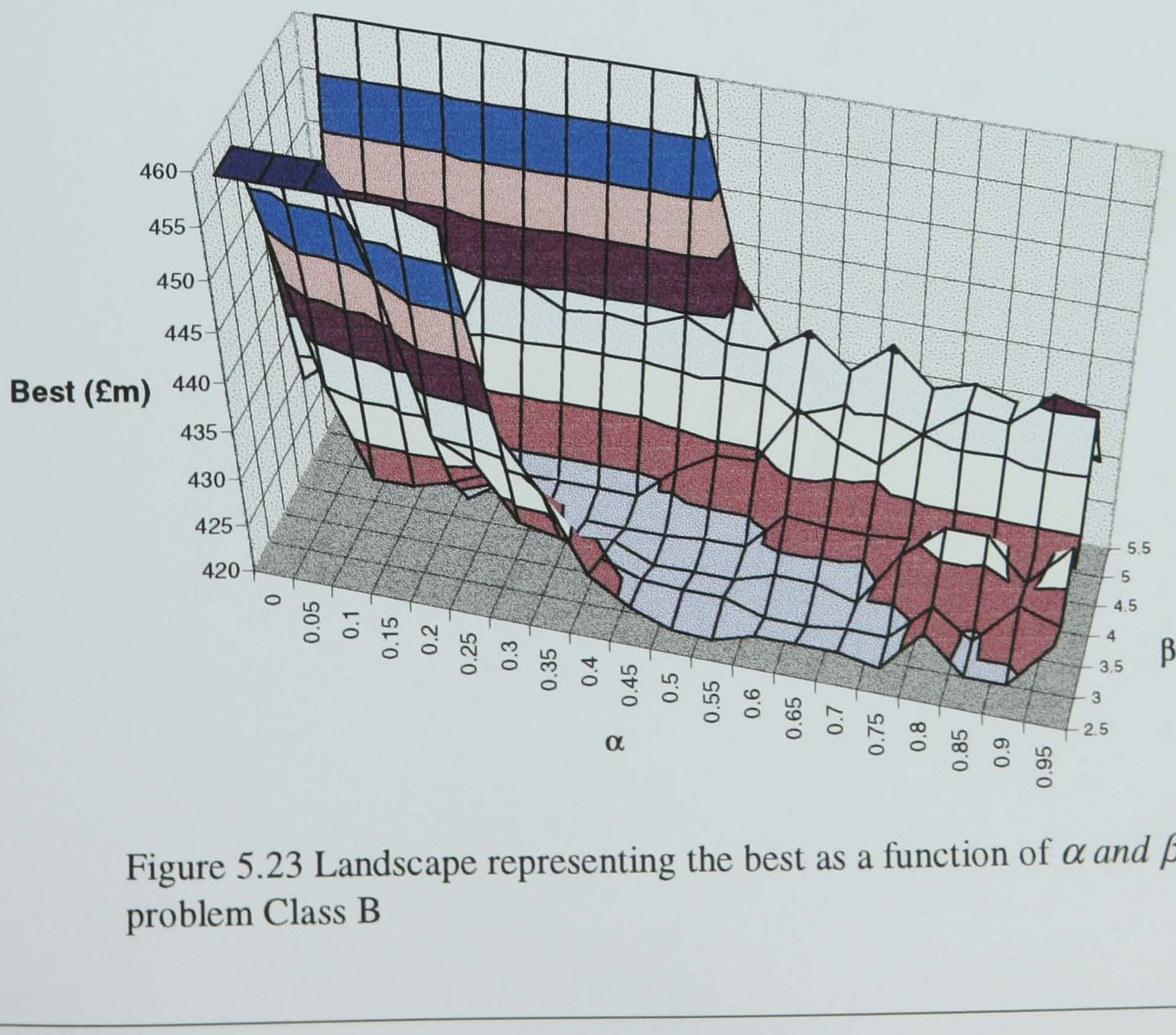


Figure 5.23 Landscape representing the best as a function of α *and* β for problem Class B

Table 5.28 Parameter settings (α & m) and simulation results for Class B for $\beta = 3.5$

Best (£m) (class B)						
m -->	15	18	19	20	22	25
α						
0	456.637	457.799	456.434	449.724	451.244	444.143
0.05	450.69	445.612	453.528	435.182	459.017	450.694
0.1	440.783	436.784	438.326	425.93	432.707	439.205
0.15	430.308	434.332	430.858	425.93	436.226	431.527
0.2	432.668	429.037	427.488	427.196	431.607	432.349
0.25	433.576	427.929	424.329	429.01	426.084	429.52
0.3	426.201	426.644	425.667	428.609	427.138	429.762
0.35	424.168	423.861	422.585	423.7	423.415	423.923
0.4	423.7	422.808	**422.362**	424.146	423.923	424.248
0.45	422.83	**422.362**	423.7	**422.362**	**422.362**	**422.362**
0.5	423.7	**422.362**	423.7	**422.362**	**422.362**	423.7
0.55	423.7	423.031	422.808	**422.362**	423.7	424.146
0.6	424.369	422.808	**422.362**	423.7	423.923	423.923
0.65	424.369	424.369	422.808	423.7	423.923	423.923
0.7	432.62	424.146	422.808	**422.362**	423.923	424.369
0.75	424.146	424.146	422.808	428.383	427.491	424.146
0.8	435.073	428.829	422.808	432.174	427.491	424.146
0.85	433.066	428.829	424.592	432.174	424.369	424.369
0.9	436.634	428.829	424.369	427.937	425.93	424.369
0.95	428.383	427.714	424.369	432.62	429.498	429.275

α is the parameter that control the relative importance of trail

m *is the number of ants*

Table 5.29 Convergence cycles needed for the optimum combinations of α & m for problem class B for $\beta = 3.5$

Number of cycle needed to converge (class B)						
m-->	15	18	19	20	22	25
α	-	-	-	-	-	-
0	-	-	-	-	-	-
0.05	-	-	-	-	-	-
0.1	-	-	-	-	-	-
0.15	-	-	-	-	-	-
0.2	-	-	-	-	-	-
0.25	-	-	-	-	-	-
0.3	-	-	-	-	-	-
0.35	-	-	-	-	-	-
0.4	-	-	354	-	-	-
0.45	-	495	-	271	456	464
0.5	-	466	-	271	294	-
0.55	-	-	-	239	-	-
0.6	-	-	130	-	-	-
0.65	-	-	-	-	-	-
0.7	-	-	-	-	-	-
0.75	-	-	-	-	-	-
0.8	-	-	-	-	-	-
0.85	-	-	-	-	-	-
0.9	-	-	-	-	-	-
0.95	-	-	-	-	-	-
Run-time (min)=between 208 and 259						

α is the parameter that control the relative importance of trail

m is the number of ants

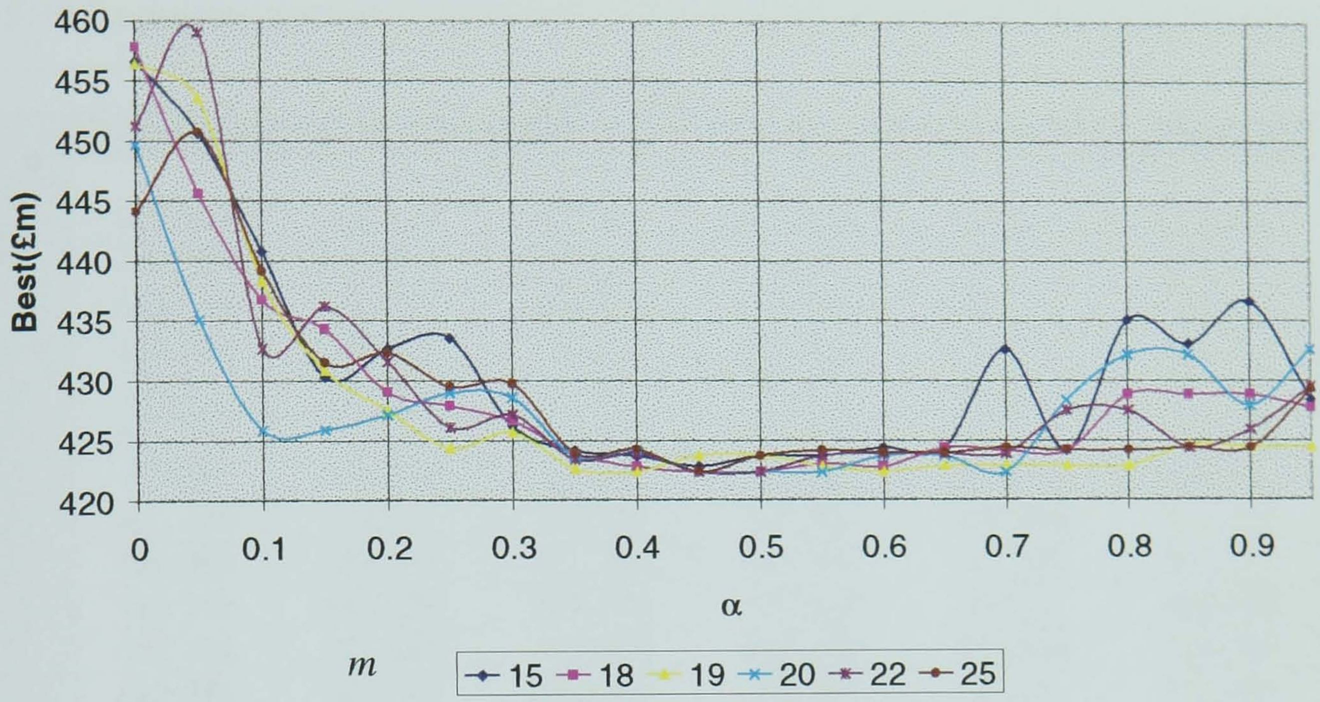


Figure 5.24 Best as a function of α and m for problem class B for $\beta = 3.5$

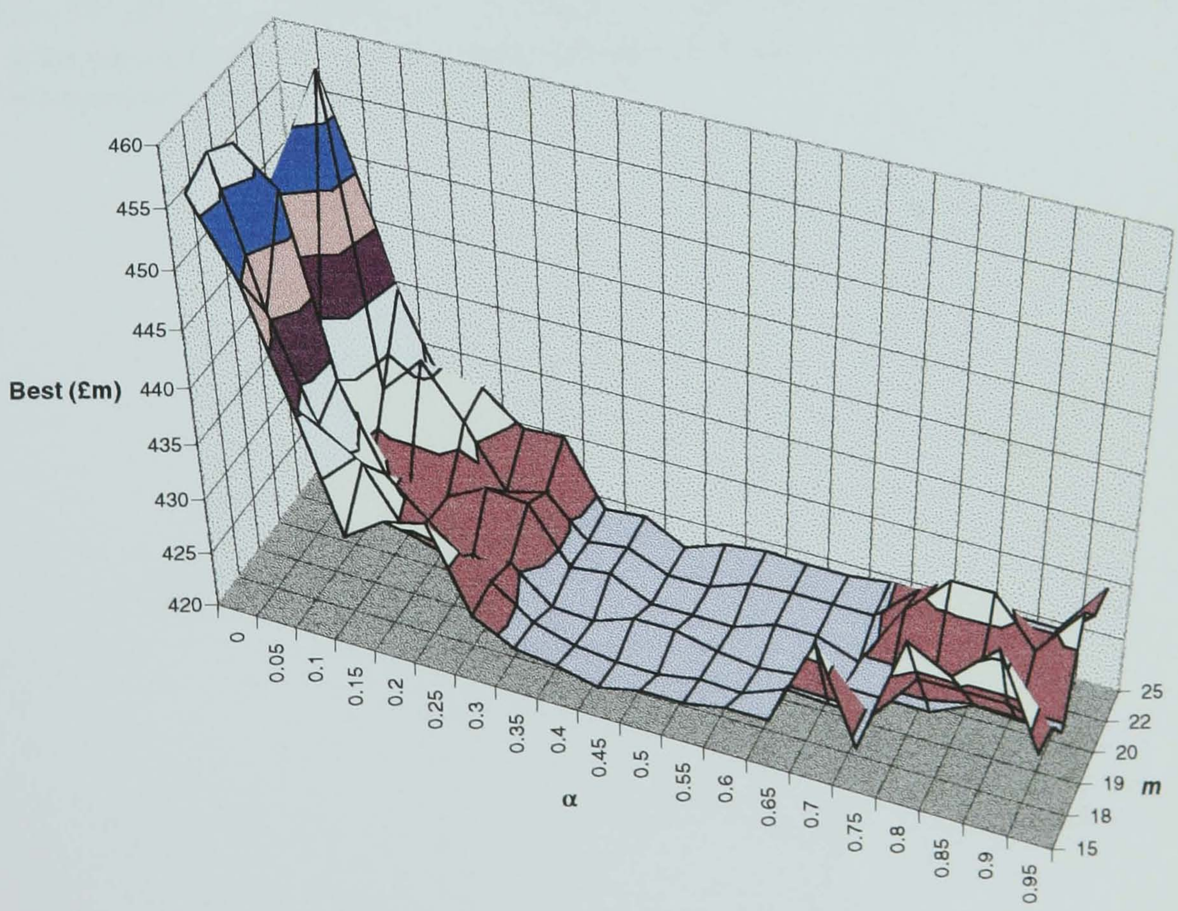


Figure 5.25 Landscape representing the best as a function of α and m for problem class B for $\beta = 3.5$

Table 5.30 Parameter settings (α & *iseed*) and simulation results for Class B and for β=3.5

	Best (£m)					
iseed-->	987654321	123456789	975319753	246802457	531975239	246802467
α						
0	445.901	449.724	447.863	452.009	450.018	459.723
0.05	446.3874	435.182	445.337	443.25	442.722	448.495
0.1	443.775	425.93	439.902	434.819	438.776	435.434
0.15	436.661	425.93	427.937	435.484	432.715	431.716
0.2	429.77	427.196	429.275	431.932	429.987	431.419
0.25	426.775	429.01	424.886	427.513	427.176	428.536
0.3	426.51	428.609	424.837	423.802	422.585	426.391
0.35	423.923	423.7	423.638	422.808	**422.362**	423.415
0.4	**422.362**	424.146	422.808	422.808	422.585	426.391
0.45	**422.362**	**422.362**	**422.362**	422.808	425.93	**422.362**
0.5	**422.362**	**422.362**	422.808	**422.362**	**422.362**	428.606
0.55	**422.362**	**422.362**	422.808	**422.362**	425.93	427.491
0.6	**422.362**	423.7	422.808	422.585	426.153	427.491
0.65	423.923	423.7	422.808	**422.362**	431.059	428.606
0.7	423.923	**422.362**	422.808	422.808	431.505	428.606
0.75	427.937	428.383	429.052	427.491	431.505	428.829
0.8	426.822	432.174	429.052	422.585	431.505	428.829
0.85	426.822	432.174	429.052	423.031	431.505	432.397
0.9	426.822	427.937	430.613	426.376	431.505	432.843
0.95	427.937	432.62	430.613	426.599	431.505	432.62

α is the parameter that control the relative importance of trail
iseed is responsible for the randomisation process

Table 5.31 Convergence cycles needed for the optimum combinations of α & *iseed* for Class B for $\beta = 3.5$

Number of cycle needed to converge						
iseed-->	987654321	123456789	975319753	246802457	531975239	246802467
α						
0	-	-	-	-	-	-
0.05	-	-	-	-	-	-
0.1	-	-	-	-	-	-
0.15	-	-	-	-	-	-
0.2	-	-	-	-	-	-
0.25	-	-	-	-	-	-
0.3	-	-	-	-	-	-
0.35	-	-	-	-	449	-
0.4	364	-	-	-	-	-
0.45	401	271	315	-	-	213
0.5	291	271	-	436	424	-
0.55	426	239	-	470	-	-
0.6	296	-	-	-	-	-
0.65	-	-	-	167	-	-
0.7	-	-	-	-	-	-
0.75	-	-	-	-	-	-
0.8	-	-	-	-	-	-
0.85	-	-	-	-	-	-
0.9	-	-	-	-	-	-
0.95	-	-	-	-	-	-
Run-time (min) ≅ 211.7034						

α is the parameter that control the relative importance of trail

iseed is responsible for the randomisation process

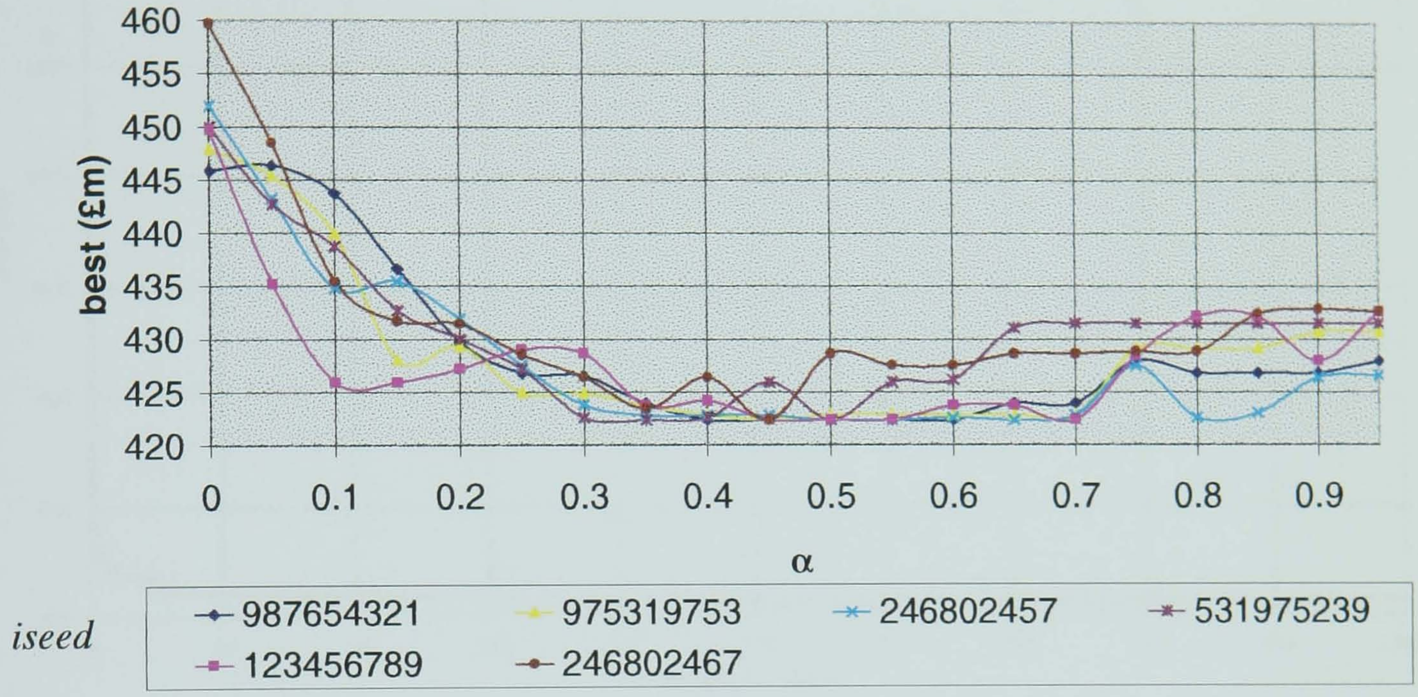


Figure 5.26 Landscape representing the best as a function of α and *iseed* for problem Class B for $\beta = 3.5$

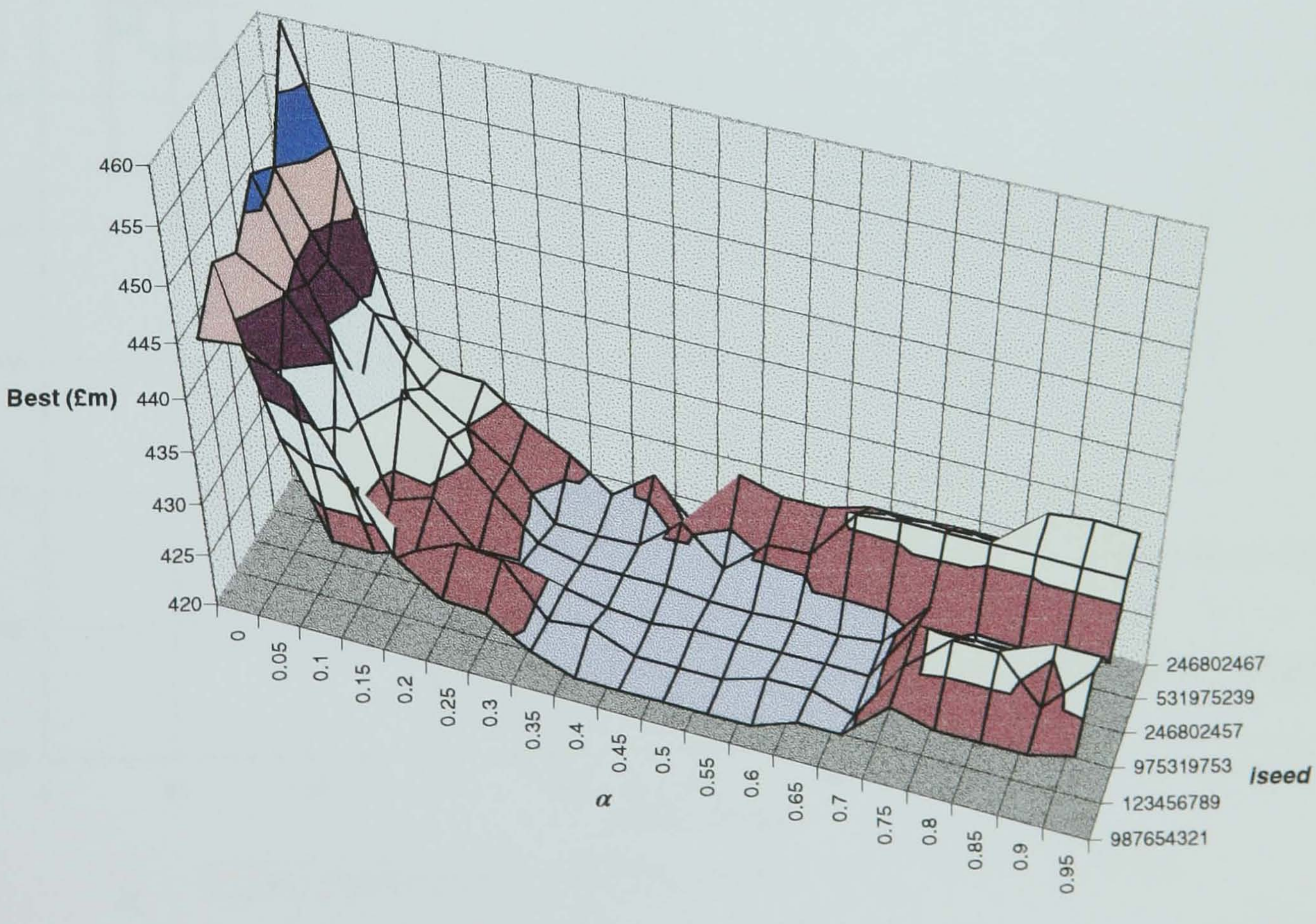
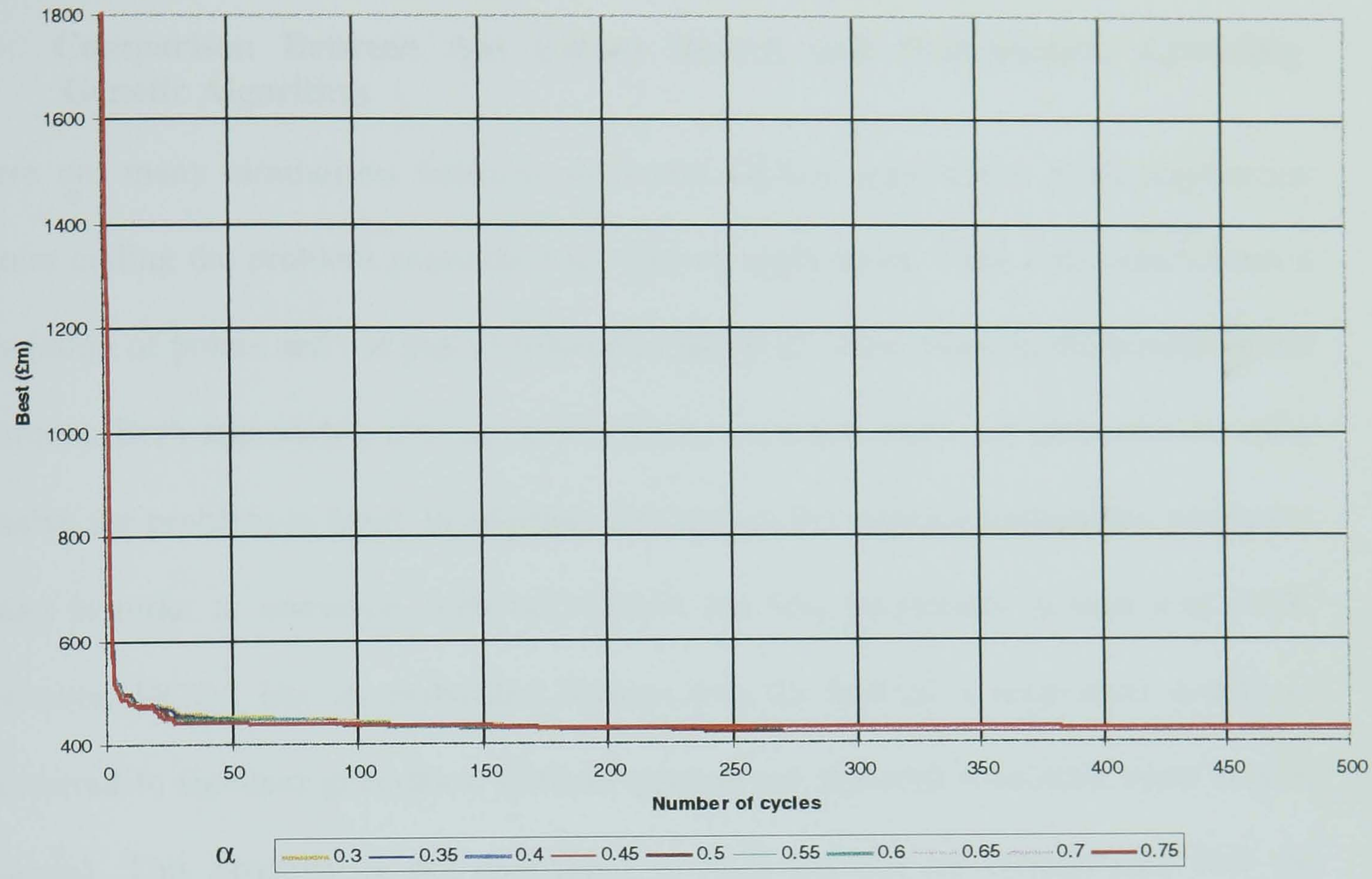
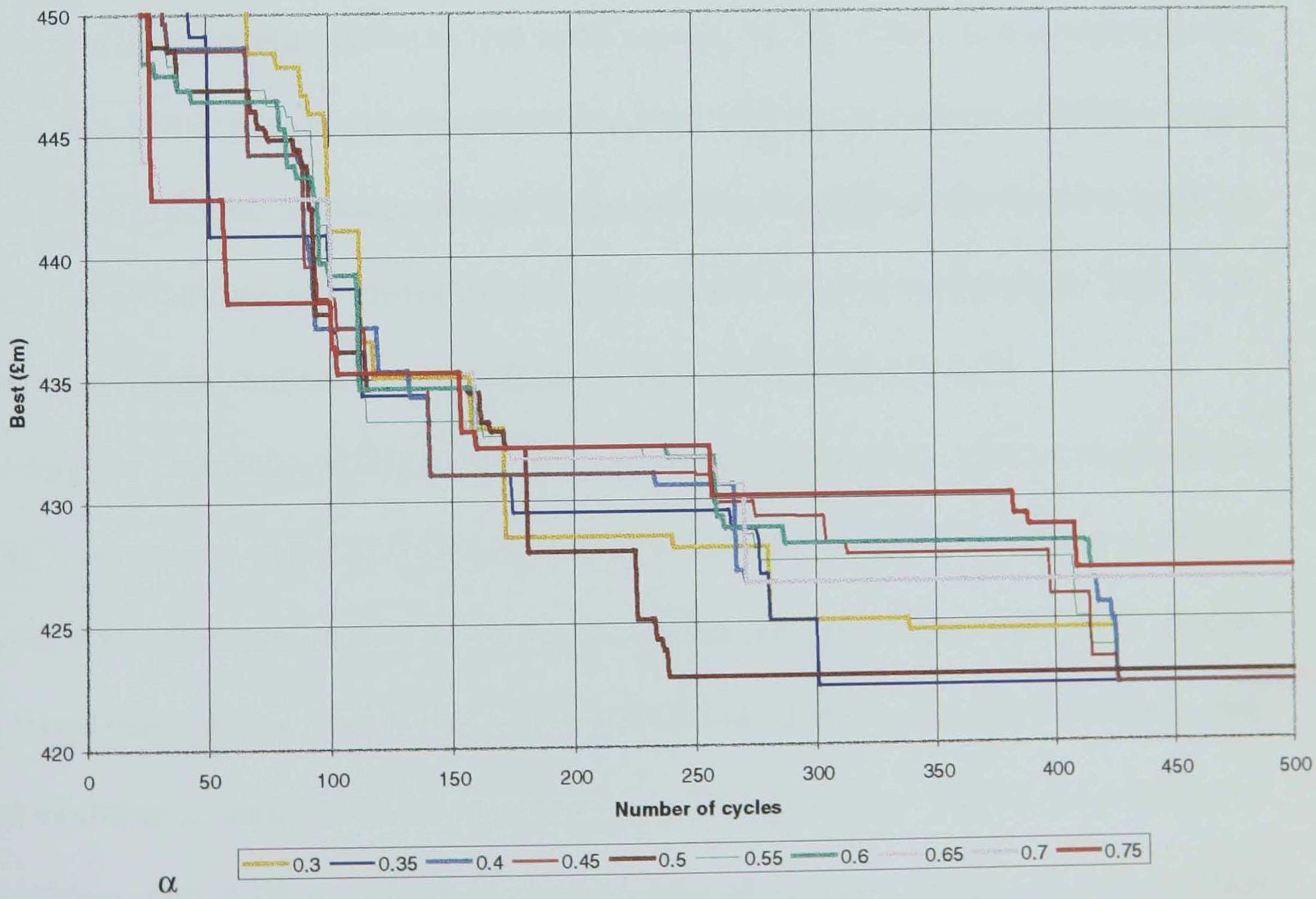


Figure 5.27 Landscape representing the best as a function of α and *iseed* for problem Class B for β=3.5.



a) Full convergence graph (best in the range £m 400-1800)



b) A magnified version of the convergence graph (best in the range £m420-450)

Figure 5.28 Convergence graph for class B for $\beta = 4$ and $\alpha \in \{0.3 \ldots 0.75\}$

5.9 Comparison Between Ant Colony Search and Deterministic Crowding Genetic Algorithm

There are many similarities between ACS and DCGA approaches. Both approaches require coding the problem parameters in order to apply them. They both search from a population of points and use payoff (objective function) information in the pursuit of the optimum. Both approaches also use probabilistic transition rules, not deterministic rules to solve the problem in hand. In addition both techniques requires appropriate parameter tuning in order to converge, however, DCGA has less parameters to tune than ACS. Moreover, DCGA has an embedded elitism; i.e., the best of a generation is always transferred to the next generation (inferior parents are replaced with their more similar children). This property is not embedded in ACS but can be incorporated into the algorithm as explained earlier in chapter 4.

However, both techniques differ in two main aspects. In ACS there is a communication between ants (synergy) through the pheromone trail, which is in contrast to DCGA where there is no information exchange among the population. In addition, the decision made by the ants takes into account global information as well as local information. Table 5.32 reports on the main similarities and differences between DCGA and ACS.

Regarding the performance of the two proposed methods, problem class A results show that ACS converged to the optimum in at least 7 seconds of CPU time where as DCGA achieved convergence in at least 4 minutes. Therefore, for this simple class, ACS appears to be faster than DCGA. This is due to the exploitation of the local information provided by the visibility in addition to the experience gained by the ants. That is, the solution to that problem with ACS is refined as ants construct their paths (solutions) step by step. The decision to move from one node to another is made with a probability, which is function of the pheromone trail accumulated on that node and the visibility of that node (the node with the cheapest cost is more attractive to ants than other nodes).

For problem class B, ACS converged in at least around 59 minutes whereas DCGA converged in at least around 2 minutes. For more realistic problems, the DCGA appear to be more efficient computationally than ACS and further modelling has been carried out with DCGA. Table 5.33 illustrates the comparative performance of DCGA and ACS.

Table 5.32 Comparison between DCGA and ACS approaches

DCGA	ACS
Need for problem representation (coding of the parameter set)	Need for problem representation (coding of the parameter set)
Search from a population of points	Search from a population of points
Iterative process	Iterative process
Need for parameter tuning	Need for parameter tuning
Use of payoff information	Use of payoff information
Use of probabilistic transition rules (through the use of crossover and mutation probabilities)	Use of probabilistic transition rules (through the decision of ants to move to another node)
Embedded elitism	Can include elitism (as it is the case in our implementation)
-	Use of local information provided by the visibility concept
-	Communication among agents (synergy) through pheromone trail

Table 5.33 Comparison of the performance of DCGA and ACS for problem classes A and B

	DCGA	**ACS**
Problem class A	• at least 4 minutes, • more reliable (optimum is achieved with a wide range of parameters	• at least 7 seconds, • less reliable
Problem class B	at least 2 minutes	at least 59 minutes

5.10 Conclusions

In this chapter, two heuristic approaches, namely the Deterministic Crowding Genetic Algorithm and the Ant Colony Search, have been applied to the transmission planning problem. The aim has been to assess both approaches as planning tools to optimise the configuration of the system. Both models have been applied to a 23-node 49-route transmission network design problem, which represents a simplified version of the England and Wales transmission network.

Rigorous experiments that consume months of CPU time have been carried out. These tests have been subdivided into four categories (problem classes A, B, C and D) according to the objective function.

Simulation results have shown the suitability of both approaches to the solution of the transmission-planning problem. They have also demonstrated the effectiveness of both algorithms in the search for the optimum. However, both algorithms require parameter tuning in order to get the best solutions. In addition, the landscape of the transmission planning problem resulting from the application of DCGA is smoother than the one obtained with ACS for the problems considered.

The solutions obtained for problem classes A and B agreed with the known theoretical optima giving a degree of confidence in both approaches. However ACS appears to be more computationally expensive than the GA for more realistic problem classes. Therefore further modelling was carried out using DCGA.

For the realistic problems (classes C and D), with no known optimum, the best solutions obtained by DCGA did appear to be of high quality when inspected manually by experts.

Chapter 6

Application of GA to Network Planning Including Voltage Transformation

6.1 Introduction

Because of the flexibility of Genetic Algorithms, further modelling requirements can be included in the fitness function to further improve the transmission system design.

This chapter illustrates problem class E which includes further enhancement of problem class D, the additional implementation of maintenance and transformation costs in the transmission-planning problem. Several experiments have been carried out to optimise the transmission system design. The aim is to design a cost effective, maintainable, and secure system.

6.2 Transformer Modelling

As mentioned in chapter 5, the GA has the ability to choose a transmission line type in the range 0 to 14 for every available route, with type 0 representing an unused route and types 1 to 14 representing a variety of standard transmission lines operating at voltage levels 275KV, 500KV or 750KV. To provide a more realistic model for the design process it is possible to include a representation of the transformers which are implied by the voltage levels used. The string generated by GA is analysed to infer the transformers which are required. As well as including the transformer costs, the transformer is modelled as a transmission line having nominal resistance and reactance. Two types of transformers are considered namely, 275KV/500KV (and vice versa) and 500KV/750KV (and vice versa). Associated with each transformer is the corresponding capital cost, the electrical resistance, etc. Table 6.1 reports on the types of transformers and their corresponding characteristics.

It is necessary to assess the number, and type, of transformers required at any substation in the network.

Assume v1, v2, and v3 are the available voltages with v1 being the smallest and v3 the largest. Four possible scenarios could be encountered at a node:

i) transmission lines operating at one voltage level (v1, or v2, or v3)

ii) transmission lines operating at two consecutive voltage levels (v1 and v2 or v2 and v3)

iii) transmission lines operating at two voltage levels but not consecutive (v1 and v3)

iv) and transmission lines operating at three voltage levels (v1,v2, and v3).

In the first scenario, there is no need for transformers. The second possibility requires installing one type of transformer depending on the line-voltage levels. One additional node only is required to accommodate the transformers. For the last two scenarios, two additional nodes are required and two types of transformers are required. The various possible scenarios are illustrated in figure 6.1.

The line-voltage levels are assigned to the initial node and the additional node(s) starting with the smallest voltage level at the initial node. To evaluate the number of transformers required at an initial node, the sum of line ratings for lines connected to the initial node and operating at the same voltage level is evaluated. For scenarios two and three, the number of transformers required is equal to the ratio of the minimum sum to the transformer rating chosen. For scenario 4, the number of transformers required is the ratio of the minimum sum between two adjacent voltage levels to the transformer rating chosen. The number of transformers required for the various scenarios are calculated as follows:

Scenario 2:

$$M = Min\left(\sum_{ij,\, j\in i}^{v1} rating_{ij},\ \sum_{ik,\, k\in i}^{v2} rating_{ik}\right)$$

Or

$$M = Min\left(\sum_{ij,\, j\in i}^{v2} rating_{ij},\ \sum_{ik,\, k\in i}^{v3} rating_{ik}\right)$$

Number of transformer required = M / rating of transformer (v1 to v2 or v2 to v3)

Scenario 3:

$$M = Min\left(\sum_{ij,\, j\in i}^{v1} rating_{ij},\ \sum_{ik,\, k\in i}^{v3} rating_{ik}\right)$$

Number of transformer required = M / rating of transformer (v1 to v2)

+ M / rating of transformer (v2 to v3)

Scenario 4:

$$M_1 = Min\left(\sum_{ij,\, j\in i}^{v1} rating_{ij},\ \sum_{ik,\, k\in i}^{v2} rating_{ik}\right)$$

and

$$M_2 = Min\left(\sum_{ij,\, j\in i}^{v2} rating_{ij},\ \sum_{ik,\, k\in i}^{v3} rating_{ik}\right)$$

Number of transformer required = M_1 / rating of transformer (v1 to v2)

+ M_2 / rating of transformer (v2 to v3)

In all scenarios, an additional transformer is added to the number of transformers required for security purposes.

To summarise, Figures 6.1-6.5 illustrate the modelling of transformers.

Table 6.1 Transformer types and characteristics

Types	Reactance(%)	Resistance(%)	Rating(MVA)	Cost(£m)
(275/400 KV)	15	0.15	1000	2
(400/750 KV))	15	0.15	2000	4

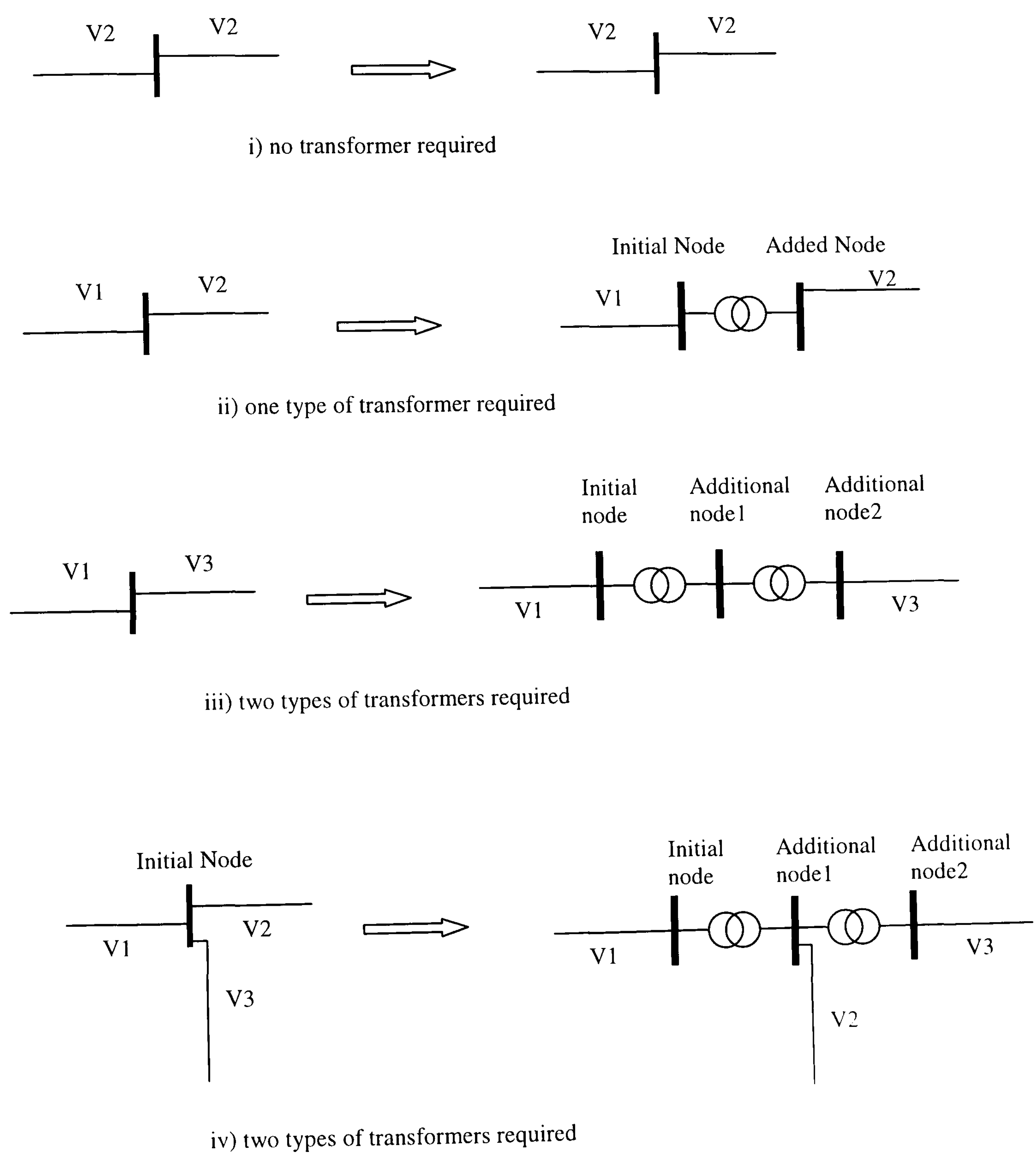


Figure 6.1 Possible voltage scenarios at a node

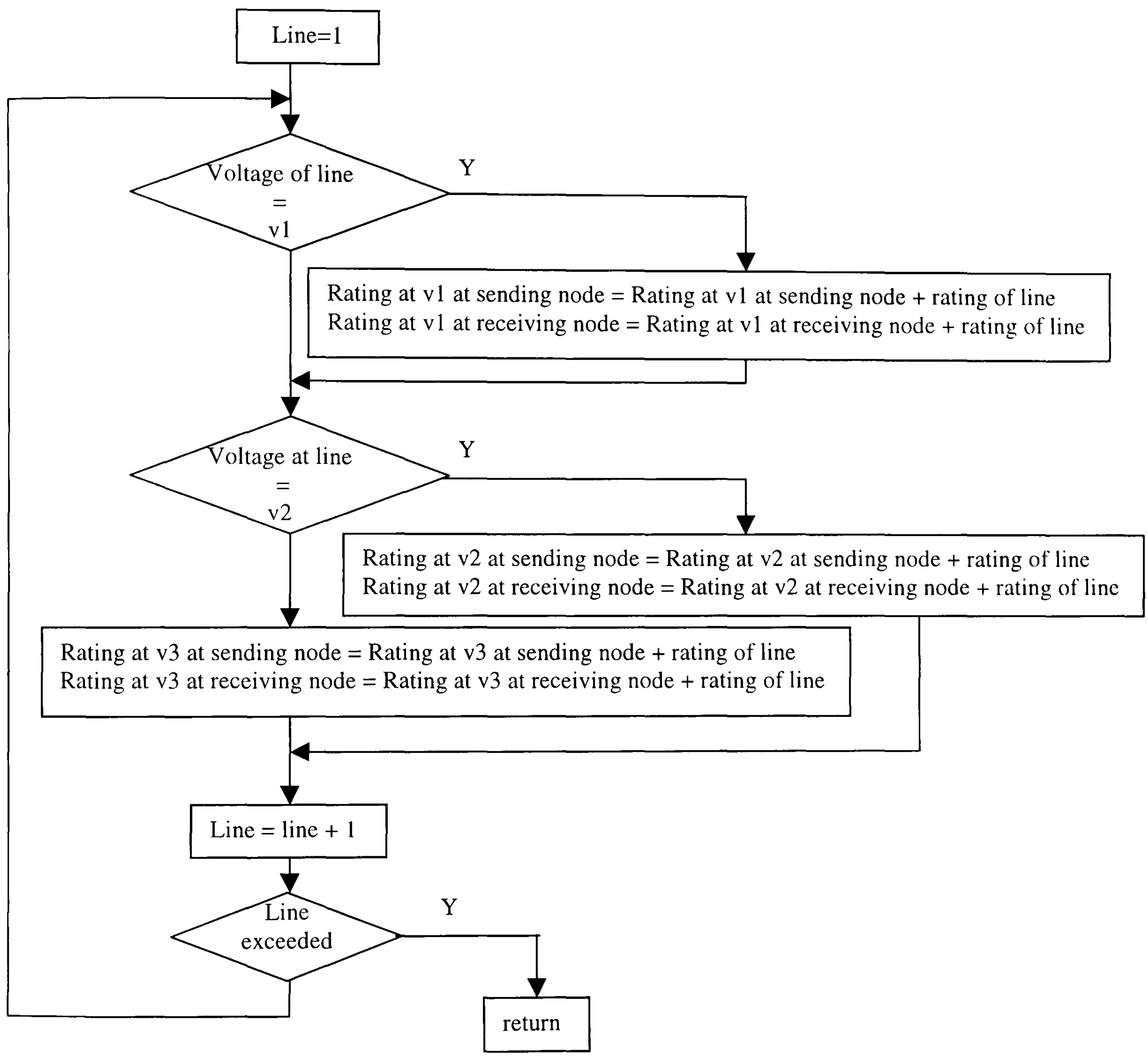


Figure 6.2 Evaluation of the sum of line ratings operating at the same voltage level and connected to the same initial node

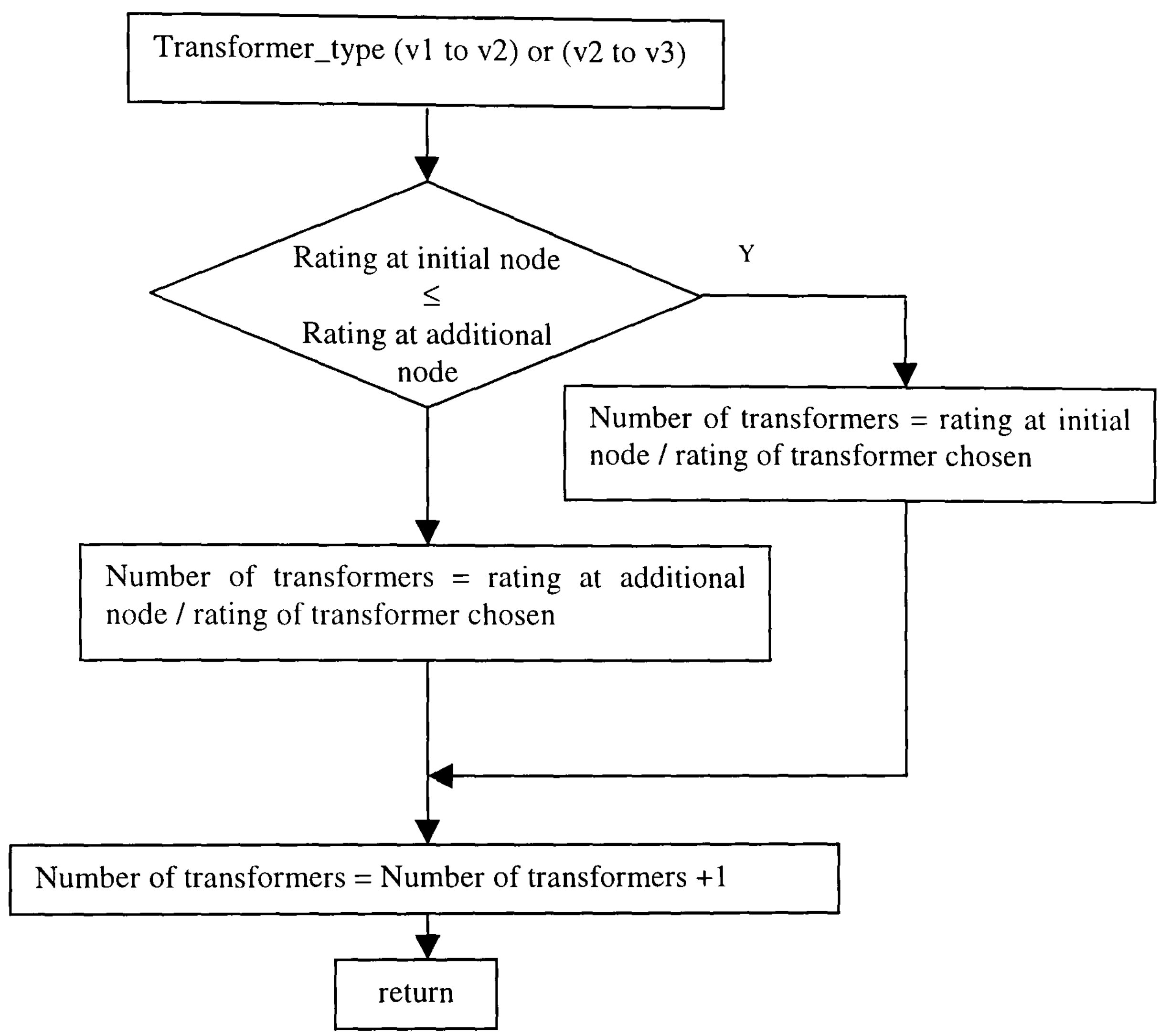


Figure 6.3 Evaluation of the Number of transformers for scenario 2

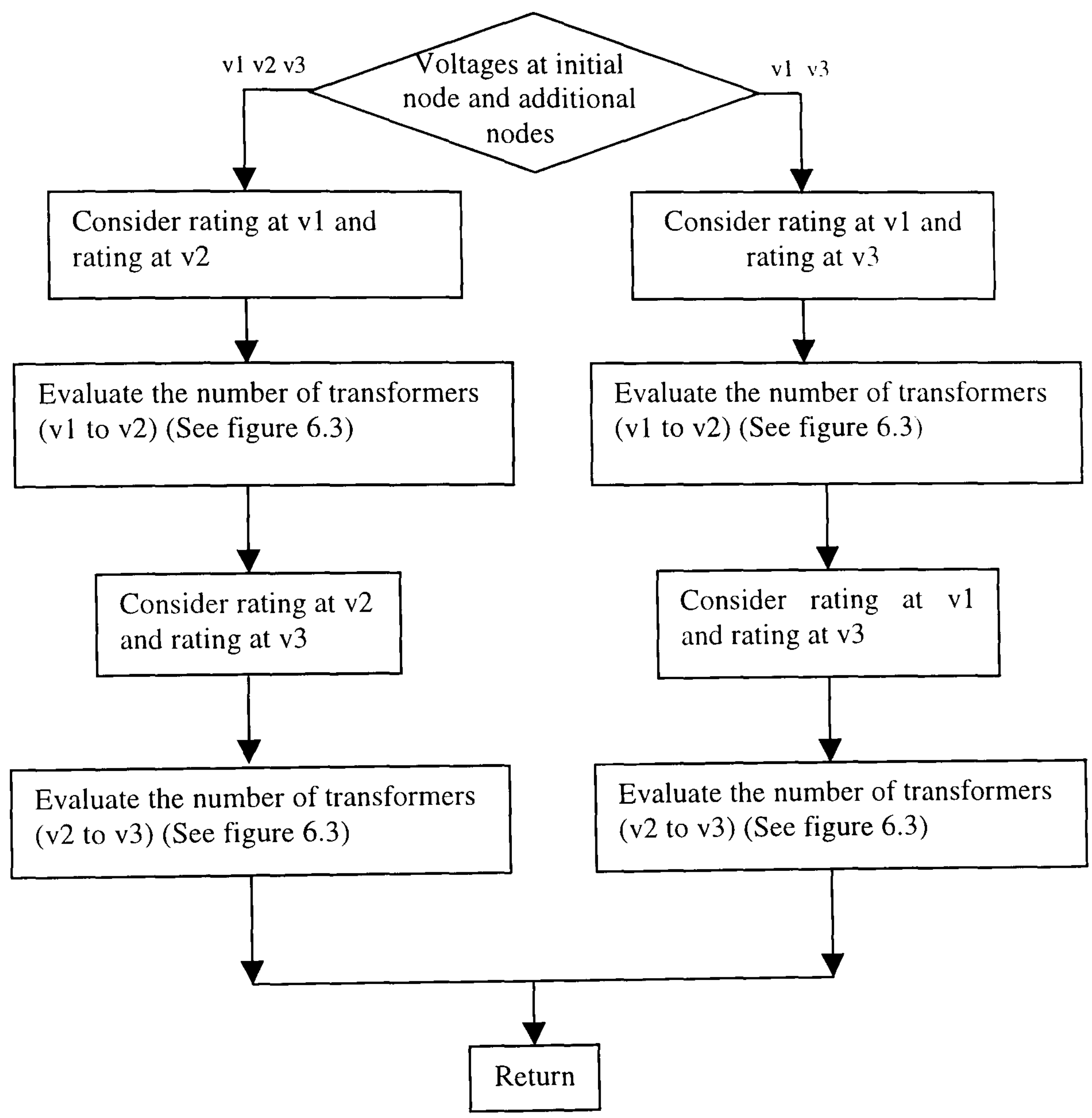


Figure 6.4 Evaluation of the Number of transformers for scenarios 3 and 4

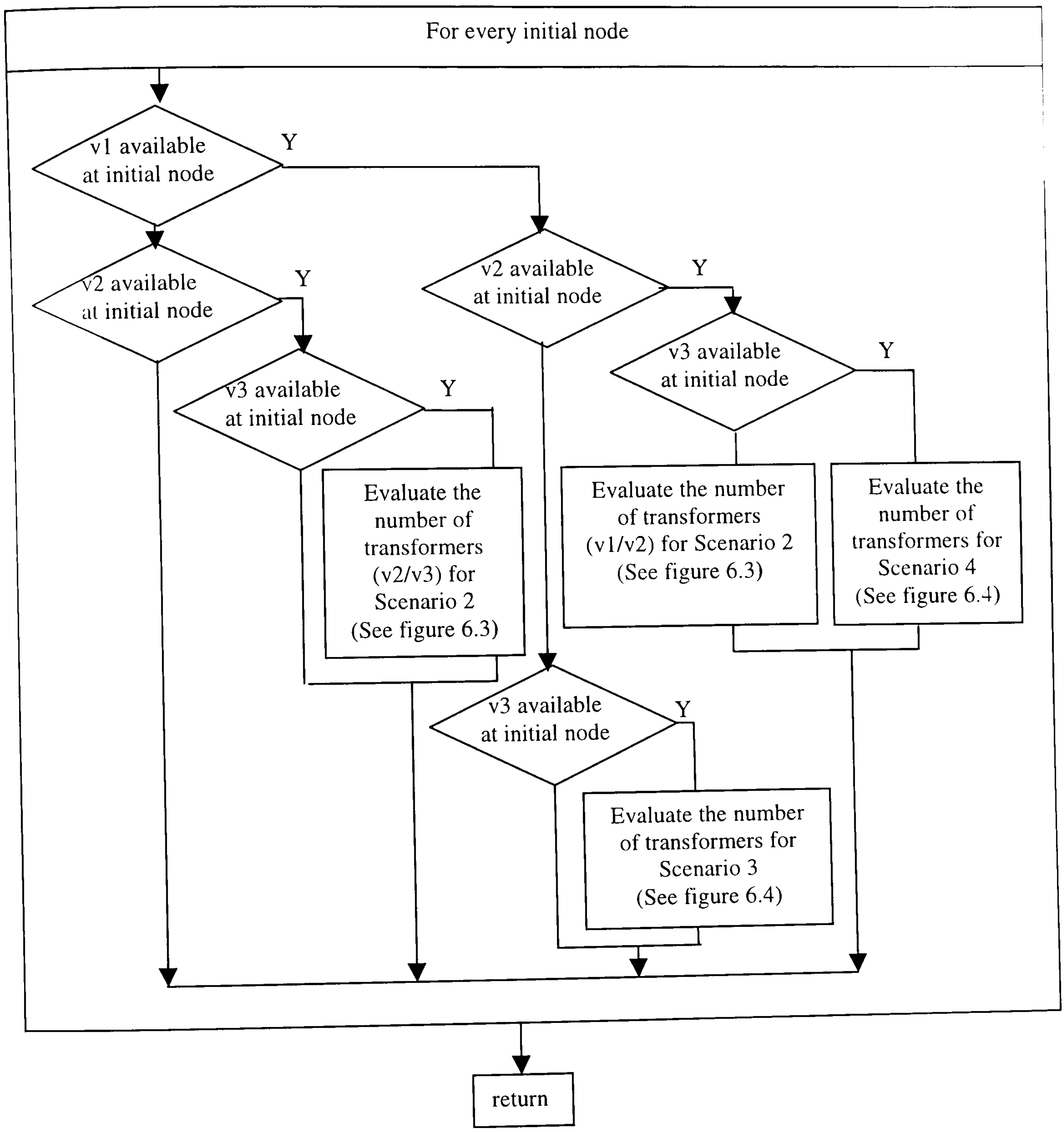


Figure 6.5 Simulation of the transformer modelling for a particular network design

6.3 Network Reconfiguration

Before any further processing of the network design, the network generated by GA is subjected to certain modifications. This includes rearrangement of generators and modification of sending and receiving ends of some of the lines to reflect the changes in the network design caused by the introduction of transformers.

6.3.1 Rearrangement of Generators

After the transformer simulation is accomplished, generation at an initial node is moved to the node with the highest voltage (the first additional node or the second one, if available) and the load is retained at the node with the lowest voltage. Figure 6.6 illustrates the redistribution of generators.

6.3.2 Modification of Sending and Receiving Nodes of the Initial Network

Only lines operating at the lowest voltage level have their sending and receiving ends unchanged. Lines operating at higher voltage level, their sending and/or receiving end will be moved to the additional node(s) according to the scenarios encountered at those ends. Figure 6.7 illustrates this process.

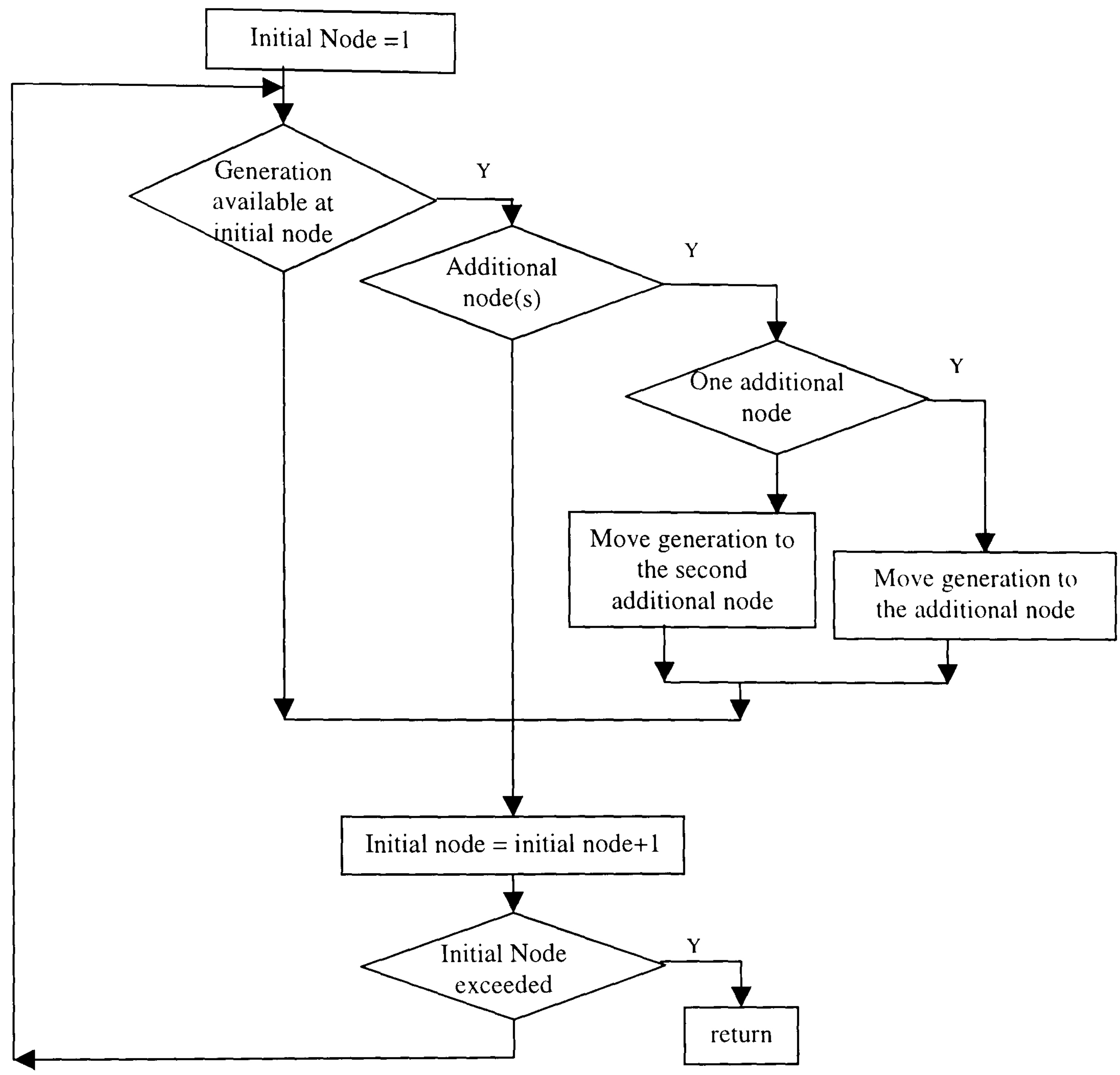


Figure 6.6 Rearrangement of generators (move generation to the node with the highest voltage)

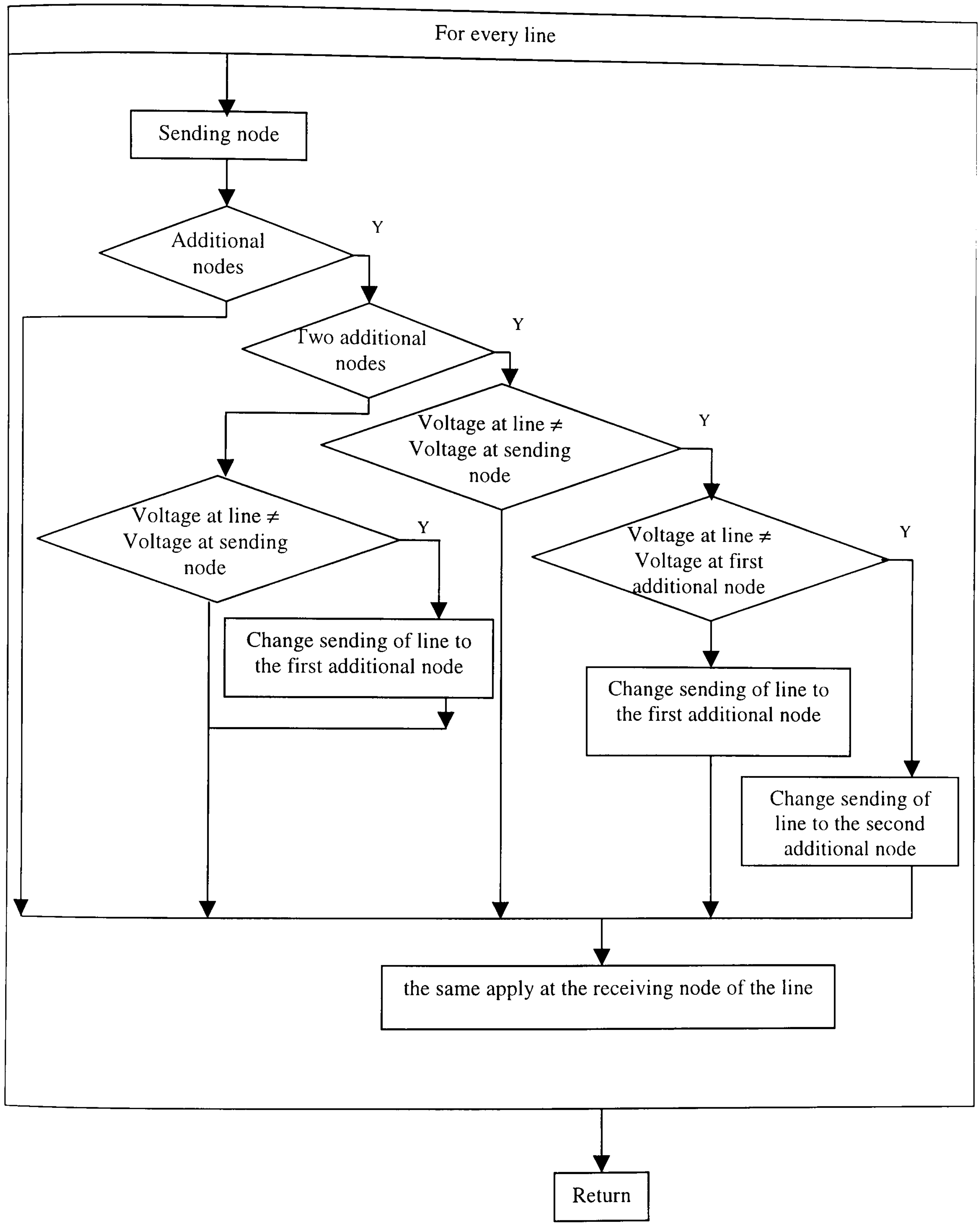


Figure 6.7 Modification of the sending and receiving nodes of lines having voltage level which is the same of the additional node(s)

6.4 Fitness Function

The objective function considered here is to minimise losses and investment including transformer costs while satisfying system constraints. In this chapter, the aim is to design a cost effective, maintainable, and secure system. As stated in chapter 5, any violation of constraints implied by a candidate solution is handled using a penalty function approach, in which penalty costs are incorporated into the fitness function so as to reduce the apparent fitness of infeasible candidates.

The overall objective function considered is therefore as follows:

$$
\begin{aligned}
F \; = \; & \sum l_i \,.\, \mathrm{cos}tl_i \;+\; \sum trans\,\mathrm{cos}t_i \\
& + \; lecf \,.\left(\sum \mathrm{lg}_i \,.\, llf_i + \sum transloss_j \,.\, transllf\right) \;+\; lccf \,.\left(\sum \mathrm{lg}_i + \sum transloss_j\right) \\
& + \; E.\sum eg_i \;+\; \Psi \,.\sum |\, pf_i \,| \;+\; p_0 \,.\, (nisl_0 - 1) \\
& + \; \phi \,.\sum \left(|\, pf_i \,| - ratl_i\right) \\
& + \; \sum \mu_i \,.\sum \left(|\, pf_k \,| - ratl_k\right) \\
& + \; \sum \lambda_j \,.\sum \left(|\, pf_m \,| - ratl_m\right)
\end{aligned}
$$

where:

F	is the overall fitness value,
	(Σ implies summation over the appropriate elements)
$lecf$	is the loss energy cost factor,
lg_i	are the thermal energy losses for each line,
$transloss_j$	are the thermal energy losses for each transformer,
llf_i	are the loss load factors for each line,
$transllf$	is the loss load factor for the transformer,
$lccf$	is the energy loss capacity cost factor,
E	is the environmental impact cost factor,
eg_i	are the environmental impact factors for each line,
Ψ	is a penalty cost factor for unsatisfied loads (based on the power flow that would be required to satisfy the load),

pf_i	are the power flows in the lines,
ϕ	is a penalty cost factor for line overloading,
μ_i, λ_j	are penalty costs for line overloading following outages for maintenance and security respectively,
$ratl_i$	are the power flow ratings of each line,
l_i	are the lengths of each line,
$costl_i$	are the annuitised capital costs (per unit length) of each line,
$transcost_i$	are the annuitised capital costs of each transformer,
p_0	are a penalty cost factors for network islanding for the intact network,
$nisl_0$	are the number of network islands for the intact network.

6.4.1 Security Requirement

In order to meet security requirements, it is necessary that no line shall be overloaded when one or more circuits are removed from the planned intact network. As mentioned in chapter 5, the so-called 'n-line' security analysis is adopted, where the optimum network should withstand any single line outage without overloading. Furthermore the possibility of 'islanding', whereby the network operates as two or more disconnected parts is also precluded (via high penalties) both in the intact network and in the outage case network. Transformers are not checked while considering the security analysis. The reason is that the number of transformers required always includes an additional transformer for security purposes.

6.4.2 Maintenance Requirement

Maintenance of high voltage electricity transmission network aims to improve reliability, however it reduces circuit availability since circuits must be switched out for safety reasons. It is appropriate to take into account the cost of maintenance during the network planning process. In this work, the maintenance is considered at a circuit level. That is, the maintenance procedure is to take out of service one circuit at a time (including transformers). The resultant outage network shall not be overloaded nor islanded. In addition, it should be (n-line) secure. To avoid excessive computing time, the feasibility of the string generated by GA is first checked against a specified criterion. Every node with a generator or a load should be connected to at least three circuits. The reason is to ensure connectivity of a load or generation to the network while the network undergoes maintenance and in the presence of a further outage due to a fault. This will avoid checking unfeasible solutions against maintenance and security and therefore avoid excessive computational time. The feasibility check is illustrated in figure 6.8.

The main steps for the consideration of maintenance are as shown in figure 6.9.

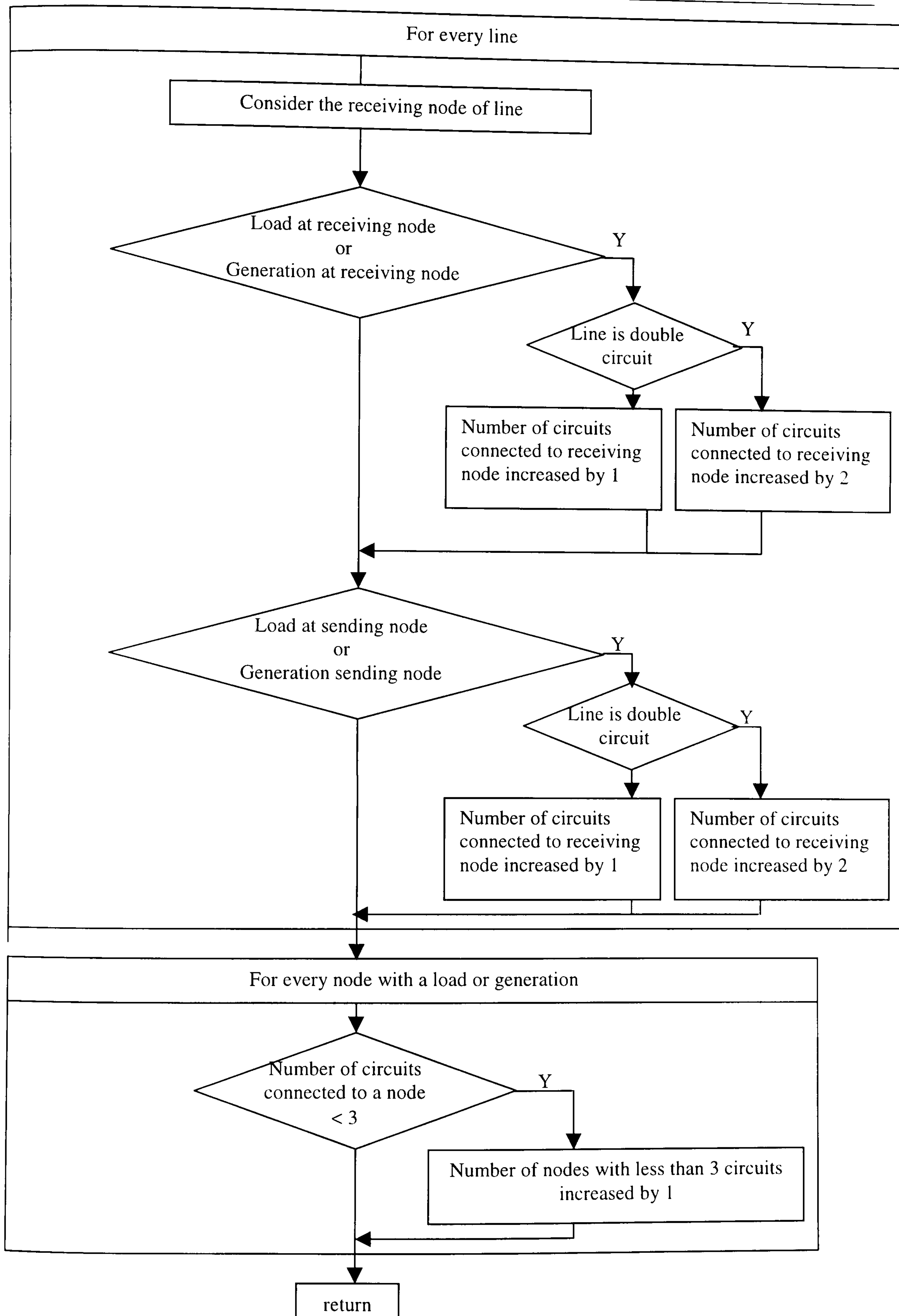


Figure 6.8 Evaluate the number of circuits connected to a node (with a load or generation) and calculate the number of nodes that have less than 3 circuits.

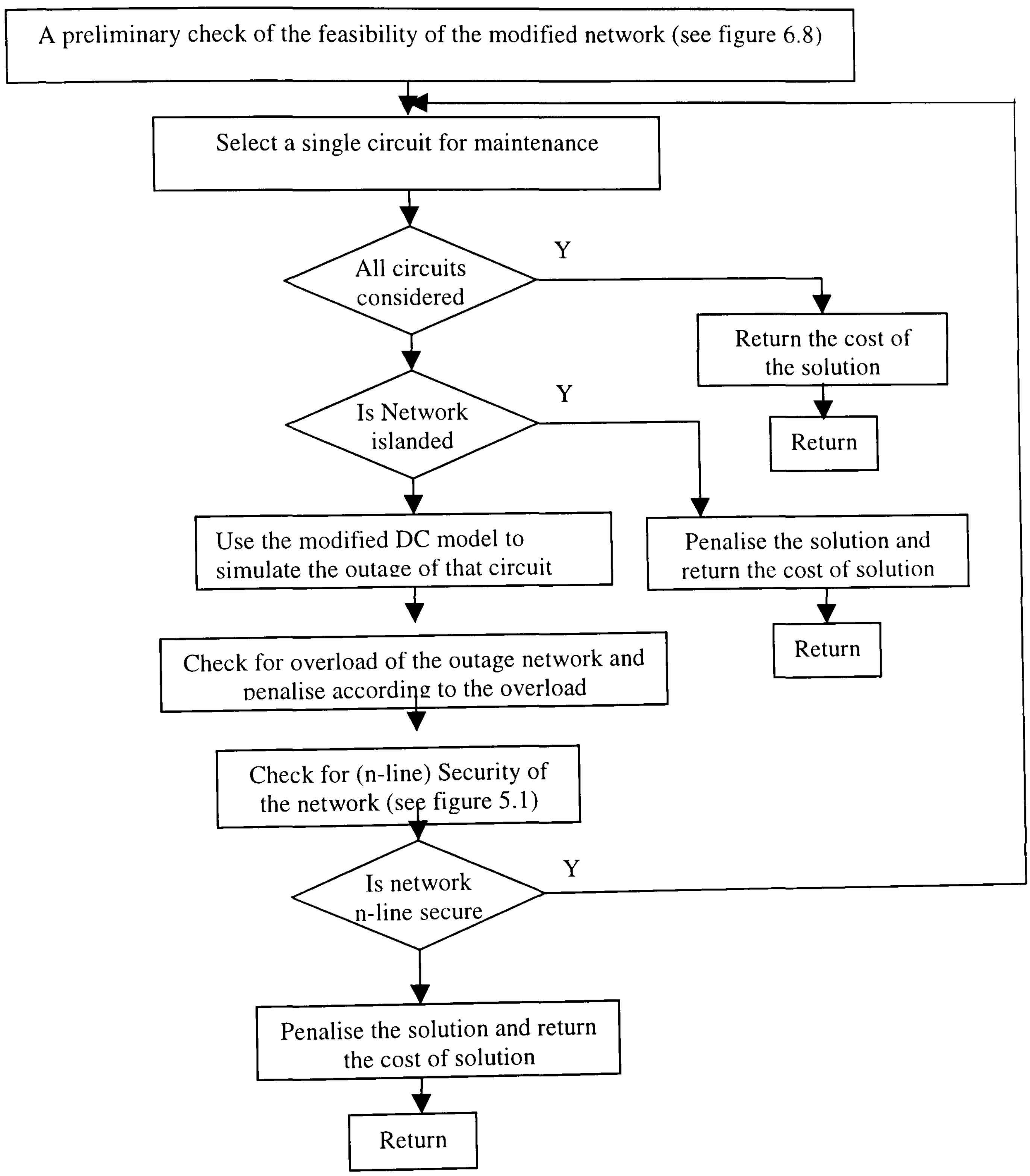


Figure 6.9 Maintenance procedure

6.5 Simulation Results

The DCGA model has been applied to a more realistic problem (including security, maintenance and transformer requirements) derived from the 23-node 49-route transmission network design described in chapter 5. Several tests have been carried out. Only the effect of various combinations of mutation and crossover probabilities (*pmut* and *pswitch* respectively) has been investigated as the computational time is quite large. The convergence criterion is chosen so that a run is terminated after a fixed number of generations (5000 generations in this case) and the best obtained so far is recorded.

For set values of population size and the seed responsible for the randomisation process (*popsize* and *iseed* respectively), the DCGA program is executed for a range of values of *pmut* and *pswitch*. The best results are illustrated in graphs. Every graph is associated with a certain value of *pmut* and shows the variation of the best (outcome of a run) as *pswitch* increases and corresponds to a series of DCGA runs that are generated automatically in the main program. The GA parameter settings and simulation results are reported in tables 6.2 - 6.4 and illustrated in figures 6.10-6.12.

Table 6.3 shows that the best so far occurs when *pmut* is set to 0.007 and *pswitch* to 0.03. However, the computational time needed is very high (around 30 hours per DCGA run) as can be seen in table 6.4. Figures 6.10-6.11 illustrate the variation of the best as a function of *pswitch* and *pmut*. It is noticeable that the GA requires appropriate tuning in order to obtain the best results. Table 6.4 shows that the best results for the various combinations of *pswitch* and *pmut* are obtained between 4600 and 5000 generations. This suggests that the best results remain unchanged only for few generations compared to the maximum number of generation specified for every run. Therefore, if the DCGA model is left to run for more generations, better results might well be found and the graphs in figures 6.10-6.11 would be smoother. However, the computational time would be very high.

Figure 6.12 shows the power network design obtained (so far) by the DCGA model. Table 6.5 reports on the distribution of transformers in the best network design. It is found that the cost of the best design is around £m 2985.096. This encompasses line cost (£m 1661.309), transformer cost (£m 12) and loss cost (£m 13111.787). The comparison of the solutions obtained for problem class E with those obtained for class D justifies the additional cost of designing a maintainable network design that takes into consideration the cost of transformers.

Therefore, for production purposes it is recommended to run the program several times trying different parameter settings to ensure that the solution falls within the best area. Moreover, the GA provides other solutions (slightly more expensive) which may be of interest to the planning engineer. These solutions (network designs) might be worthy of consideration due to additional factors, which were not included in the computer evaluation of the design cost.

Table 6.2 GA parameter settings

GA parameter settings	
Popsize	500
iseed	978456333
maxgen	5000
ptest	0.5

Popsize is population size, *iseed* is responsible for the randomisation process
maxgen is maximum number of generations of a run,
ptest is probability for crossover applied on first bits of the parents.

Table 6.3 Best as function of *pmut* and *pswitch*.

Best (£m) obtained after 5000 generations					
pmut	0.004	0.005	0.006	0.007	0.008
pswitch					
0.01	3140.657	3041.183	3202.701	3172.603	3578.798
0.02	3001.53	3044.042	3498.38	3065.959	3523.269
0.03	3048.657	3059.259	2995.763	**2985.096**	3504.103
0.04	3067.331	3088.8	2993.852	3049.778	3125.739
0.05	3020.203	3007.353	3010.033	3027.641	3059.455
0.06	3006.591	3504.255	3052.292	3088.336	3191.276
0.07	3091.411	3005.971	3076.914	3204.651	3163.86
0.08	3126.148	3086.774	3540.805	3324.707	3283.465
0.09	3020.368	3205.203	3346.989	3407.381	3168.622

bold number represents the optimum
pmut is the probability of mutation
pswitch is the probability of crossover

Table 6.4 Number of generations to obtain the best as function of *pmut* and *pswitch*.

Converged at generation (run stopped after 5000 generations)					
pmut	0.004	0.005	0.006	0.007	0.008
pswitch					
0.01	4996	4799	4971	4958	4996
0.02	4858	4954	4817	4794	4831
0.03	4942	4732	4971	**5000**	4977
0.04	4873	4908	4942	4963	4819
0.05	4860	4992	4928	4973	4972
0.06	4998	4789	4858	4906	4986
0.07	4944	4996	4926	4906	4939
0.08	4954	4797	4832	4643	4975
0.09	4956	4976	4967	4877	4890
Run-time ≈ 30hours after 5000 generations(for every run)					

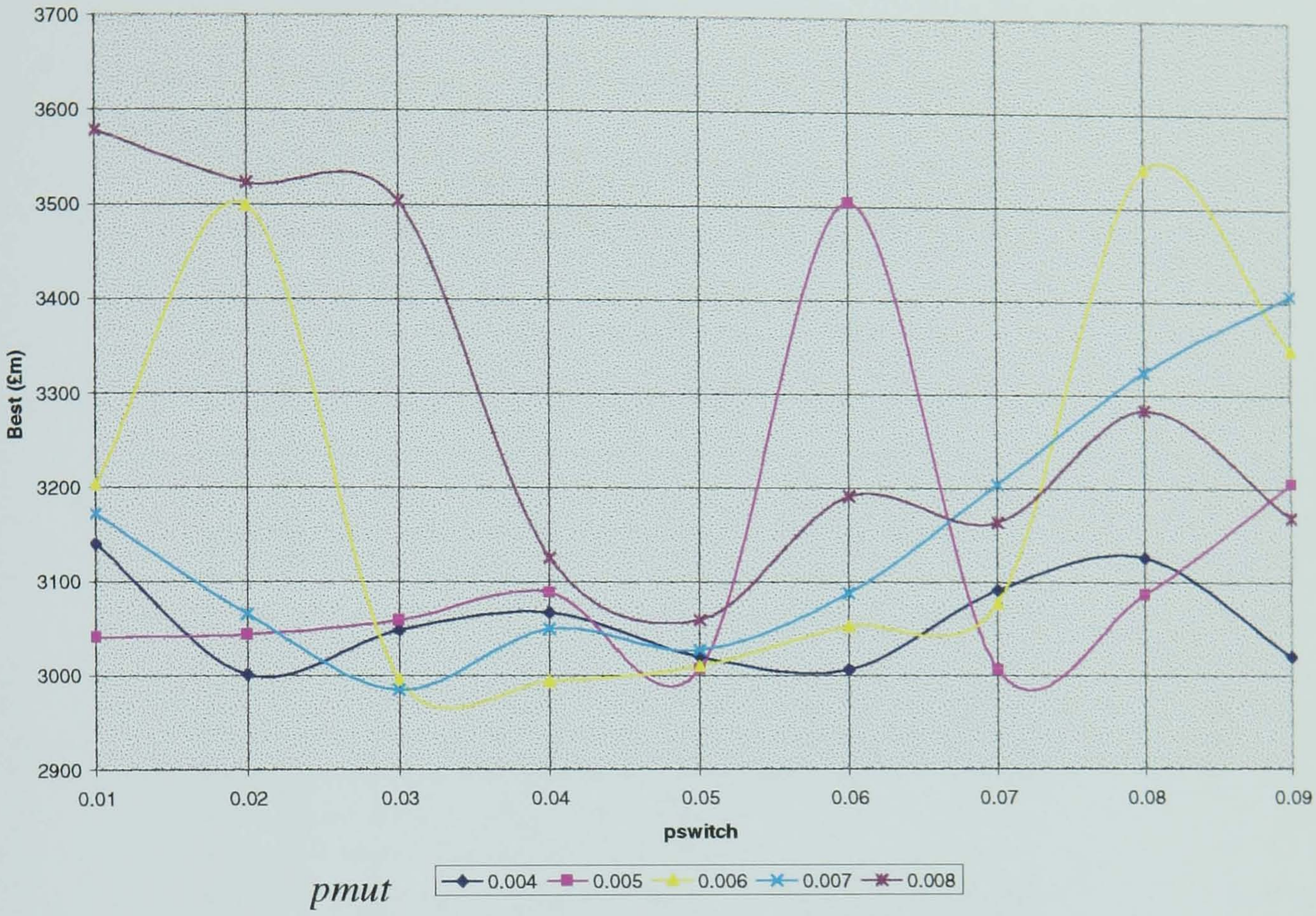


Figure 6.10 Best as function of *pswitch* and *pmut* for problem class E

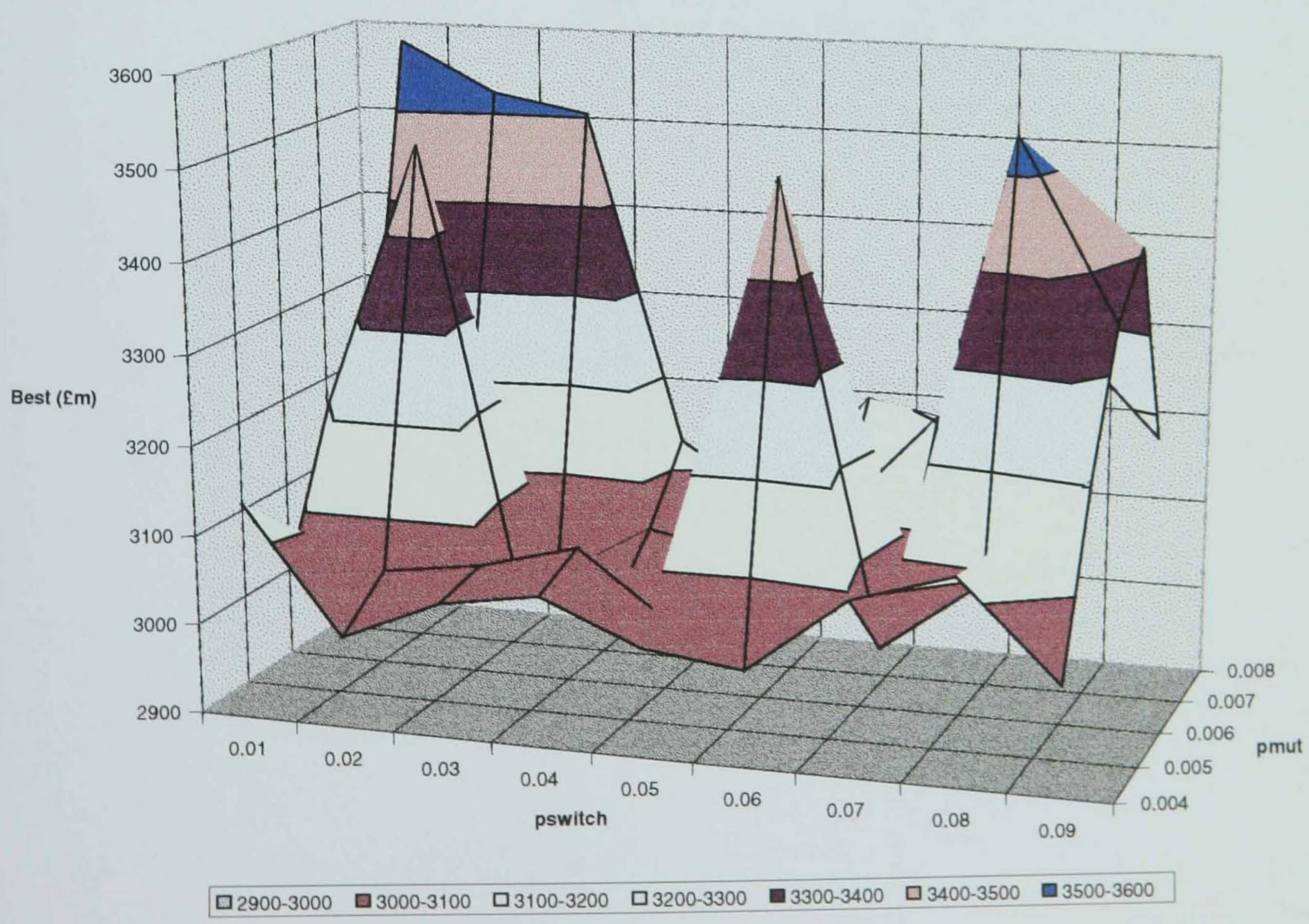


Figure 6.11 Landscape of the best as function of *pswitch* and *pmut* for problem class E

Table 6.5 Distribution of transformers in the best network design for problem class E

Nodes	Transformers of type (275 / 400 KV)	Transformers of type (400 / 750KV)
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	3	0
19	0	0
20	0	0
21	3	0
22	0	0
23	0	0

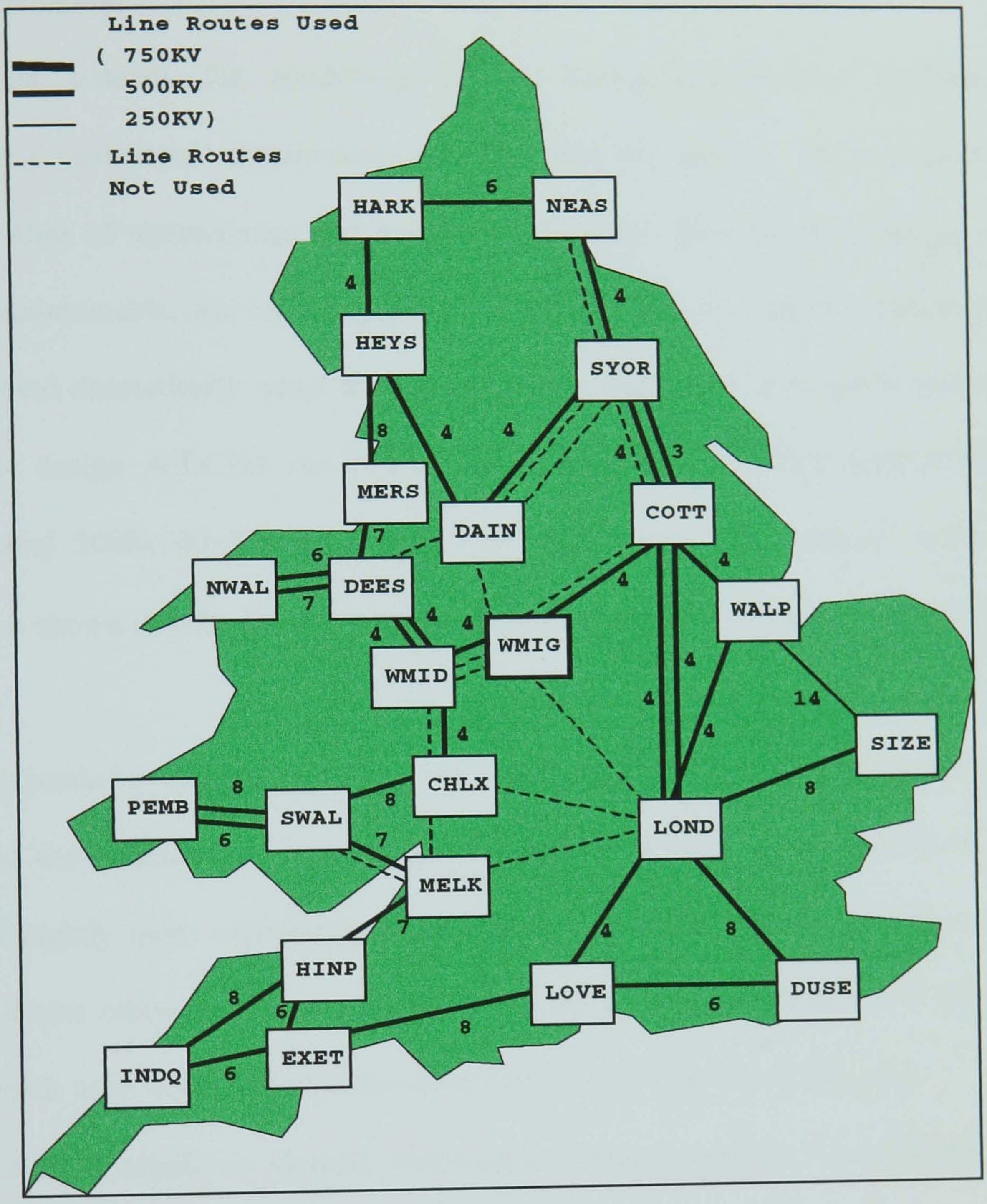


Figure 6.12 The optimum network design obtained by DCGA for problem Class E

6.6 Conclusions

This chapter extends the modelling of the transmission-planning problem by incorporating additional requirements of the network design. This includes the implementation of maintenance and transformation costs. The aim is to design a cost effective, maintainable, and secure system. It is important to note that the computational time increased dramatically when transformer and maintenance costs are considered in the network design. A DCGA run lasts around thirty hours for 5000 generations during which around 5000x500x49x49 ($\cong 6x10^9$) DC load flows are evaluated. Simulation results have shown that the DCGA requires appropriate tuning in order to obtain the best results.

It is recommended to run the program several times trying different parameter settings to ensure that the solution falls within the best area. Moreover, the GA provides other solutions (slightly more expensive) which may be of interest to the planning engineer. These solutions (network designs) might be worthy of consideration due to additional factors, which were not included in the computer evaluation of the design cost.

Because of the flexibility of Genetic Algorithms, further modelling requirements can be included in the fitness function to produce a more realistic model, but the computational time increases with the complexity of the problem. There is a trade-off between having a realistic model and the computational time required achieving this accurate design.

Chapter 7

Interpretation of Results

7.1 Introduction

Projects in the electricity supply industry live for a long time. For most power projects, most expenditure (in the form of operational cost) and income occurs after commissioning. Such future financial flows will fluctuate according to time and circumstances. Correspondingly, these will have a different financial value than flows occurring during project evaluation. Therefore the time value of money (discounting) and the choice of a proper discount rate is highly important for capital-intensive long life projects with high operational costs like those of the electricity supply industry.

In this chapter, two sensitivity analyses based on discount rate and decision variables are carried out independently. Only problem Class D is considered. The reason is that this problem class represents a realistic transmission-planning problem, which can be solved within reasonable computational time. The full-scale problem class described in chapter 6 will not be considered here due to the excessive computational time required.

The aim of these tests is to:

- investigate the effect of the discount rate and decision variables on the best solution provided for the problem,
- test the validity of the DCGA solution obtained for problem D.

It has been observed that:

- for all the tests considered, the DCGA solution obtained at a defined discount rate is better than the evaluation based on this discount rate for DCGA solutions at various other discount rates,

- among all possible combinations of line type assignments to the set of routes considered, the DCGA solution has proven to be the best,
- the solution provided by DCGA is proven to be robust as concluded from the experiments.

7.2 Discount Rate

The life cycle of a project and its feasibility, for a given output, depends on three factors: the investment cost, the operational costs and the discount rate used. Many planners think that the discount rate is the most important of these factors. It greatly affects the whole economics of the project and the decision making, particularly in capital intensive projects like those of the electricity supply industry.

The discount rate is the opportunity cost of capital (as a percentage) which is the return on investment forgone elsewhere by committing capital to the project under consideration. In investment decisions, the opportunity cost of capital is the cut-off rate below which it is not worthwhile to invest in the project.

7.3 Sensitivity Analysis of the Discount Rate on Problem Class D

As mentioned earlier in chapter 5, this problem class represents a realistic problem. No optimal solutions are known in advance. For this case, the objective of transmission planning is to minimise investment cost together with the annuitised cost of energy losses, while satisfying system constraints. The network design obtained should be 'n-1' secure. Furthermore, overloading and the possibility of 'islanding' are also precluded both in the intact network and any of the outage case networks.

Using a discount rate of 7.5%, simulation results (in chapter 5) have shown that the best solution is obtained with a narrow and intermittent range of GA parameters particularly when the probability of crossover (*pswitch*) ∈ {0.04...0.08} and the probability of mutation (*pmut*) ∈ {0.004... 0.009} but with different computational time. The optimum setting is when *pmut* = 0.004 and *pswitch* = 0.06. The optimum parameter settings correspond to the least computational time required to achieve the best design network (with the cheapest cost). Several tests have been performed using the optimum parameter setting of the 'best' network design obtained by DCGA. These tests are classified into two categories:

- run DCGA with a range of discount rates, and
- for each network design obtained from the tests above, also evaluate the cost with the other discount-rate values.

The aim is to investigate the sensitivity to the discount rate and to test the robustness of the solution provided by DCGA.

7.3.1 Test Category A

In these tests, the DCGA parameters are set to the optimum parameter values of the best solution produced with a discount rate of 7.5 % as illustrated in table 7.1. Then the DCGA model is run with various discount rates to search for the 'best' solutions. Different 'best' solutions are produced corresponding to the various discount rates. This process is illustrated in figure 7.1. Simulation results are reported in tables 7.2-7.3. Figures 7.2-7.6 depict the best network designs produced with the various discount rates considered including the best design obtained with a discount rate of 7.5 %. Figure 7.7 illustrates the sensitivity of the cost to discount rate. A high discount rate decreases the cost of the best design whereas cost increases if a low discount rate is chosen. Figure 7.8 illustrates the occurrence of line types in the 'best' solutions as a function of the discount rate. It is important to note how the optimum design changes with the discount rate. It has been observed that if a line type is not chosen at a particular discount rate, it is not usually chosen when different discount rates are applied.

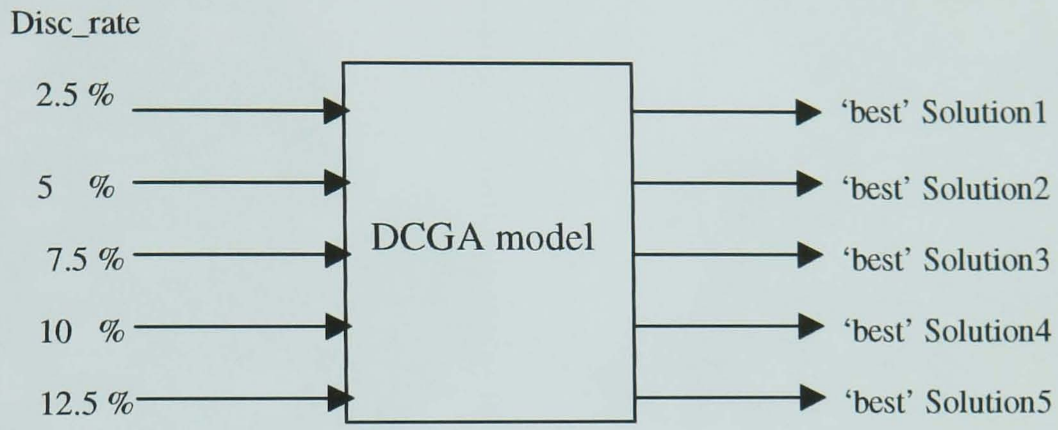


Figure 7.1 Scenarios for test category A

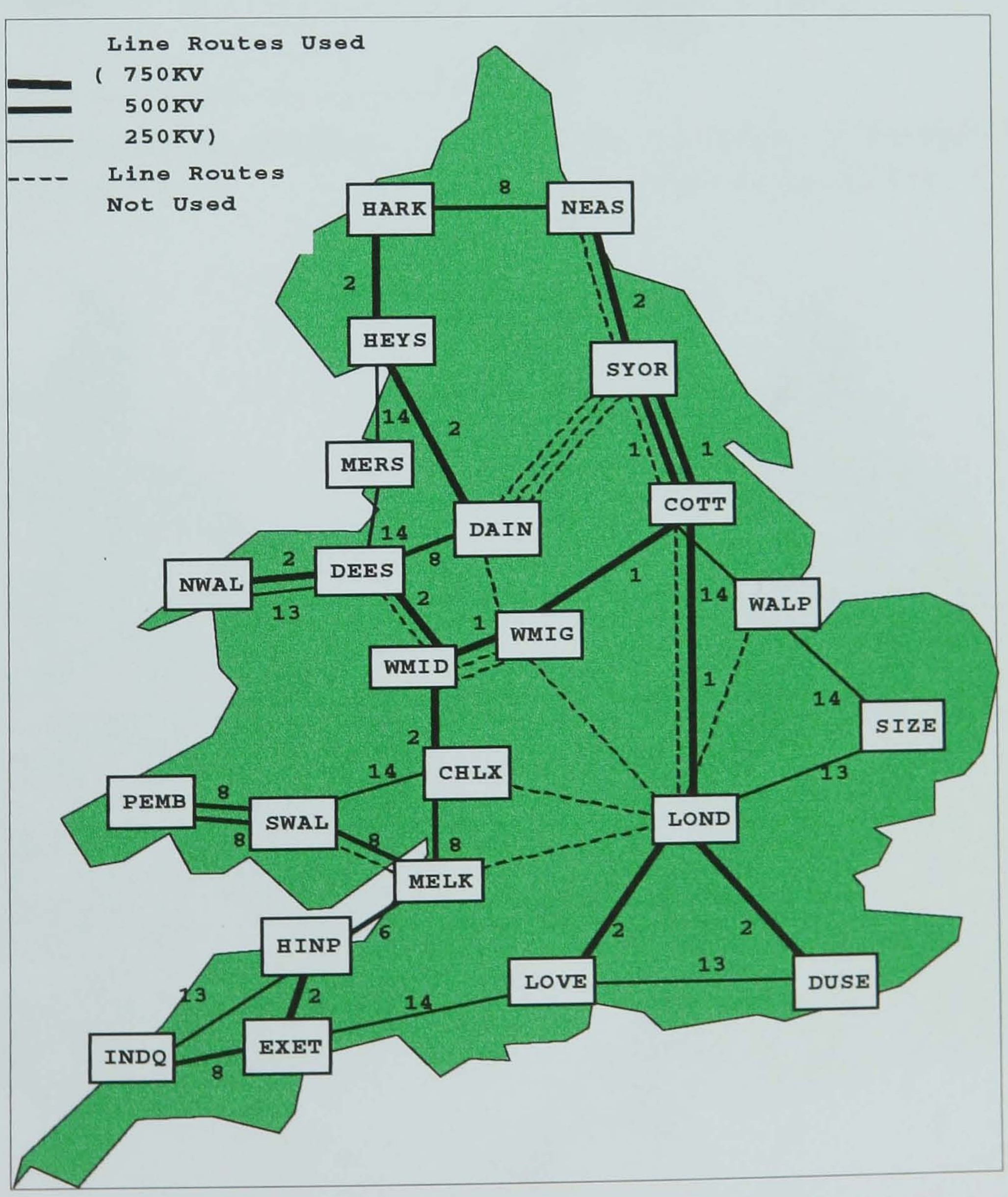


Figure 7.2 Optimum network obtained with a discount rate of 7.5 %

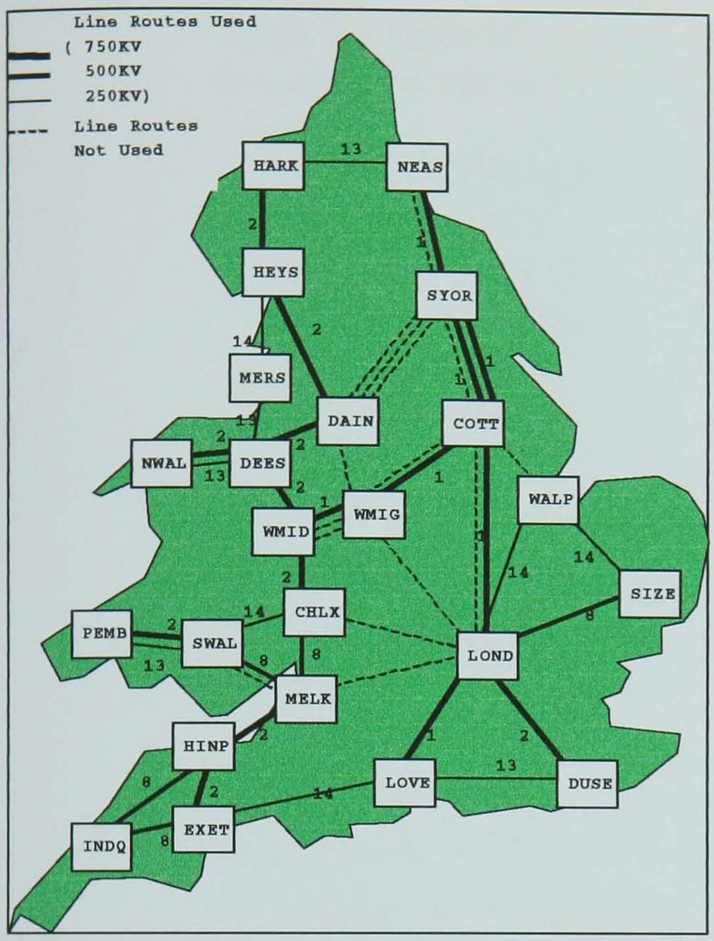


Figure 7.3 Optimum network obtained with a discount rate of 2.5 %

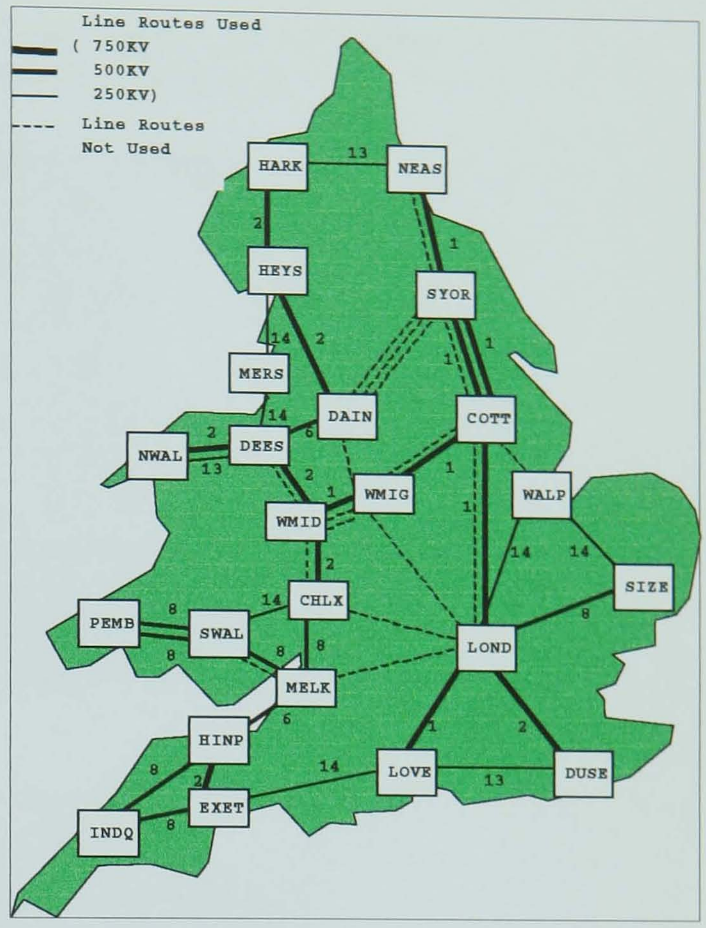


Figure 7.4 Optimum network obtained with a discount rate of 5 %

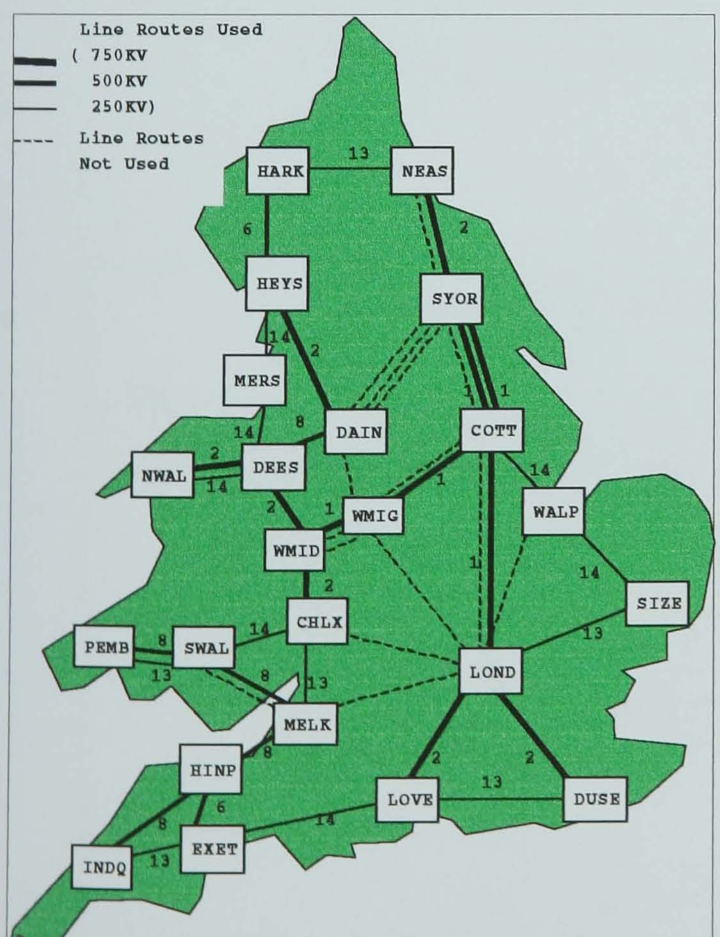


Figure 7.5 Optimum network obtained with a discount rate of 10 %

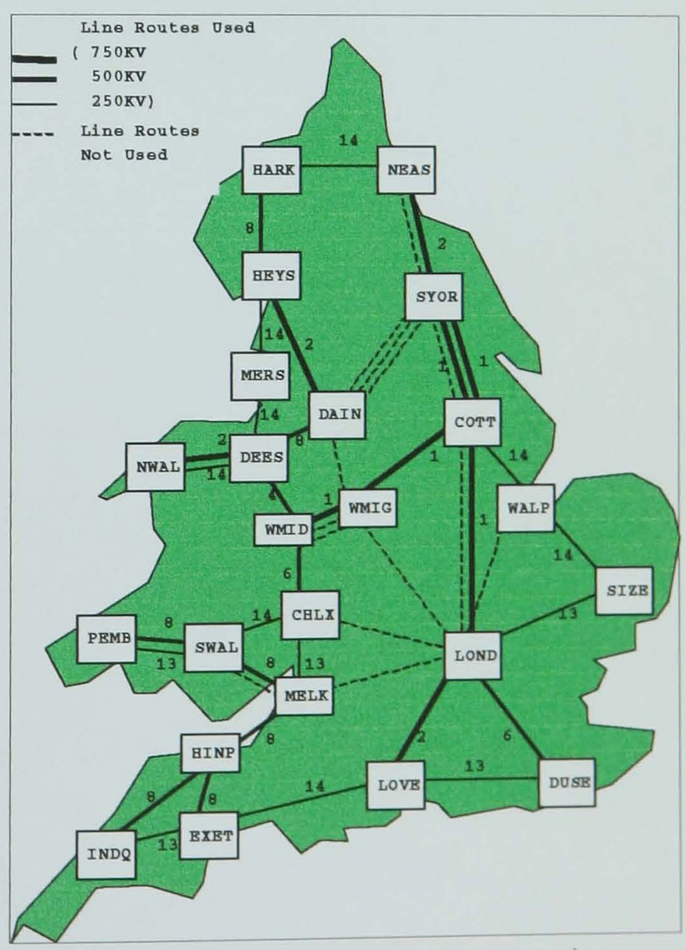


Figure 7.6 Optimum network obtained with a discount rate of 12.5 %

Table 7.1 Optimum DCGA parameter setting corresponding to a discount rate of value 7.5% for problem class D

Optimum GA parameter settings	
Popsize	500
iseed	978456333
maxgen	2000
ptest	0.5
pmut	0.04
pswitch	0.06
disc_rate	7.5 %

popsize is population size, *iseed* is responsible for the randomisation process
maxgen is maximum number of generations of a run,
ptest is probability for crossover applied on first bits of the parents
disc_rate is the discount rate used in the evaluation of the net present value of the losses.

Table 7.2 Sensitivity of best solution produced by DCGA to discount rate (disc_rate) for problem class D

disc_rate	**Best(£m) (total cost)**	**Invest(£m)**	**Loss(£m)**
0.025	2828.035	1631.867	1196.1677
0.05	2445.511	1572.896	872.6155
0.075	2261.537	1483.238	778.29937
0.1	2042.502	1407.095	635.40651
0.125	1917.046	1346.905	570.14065

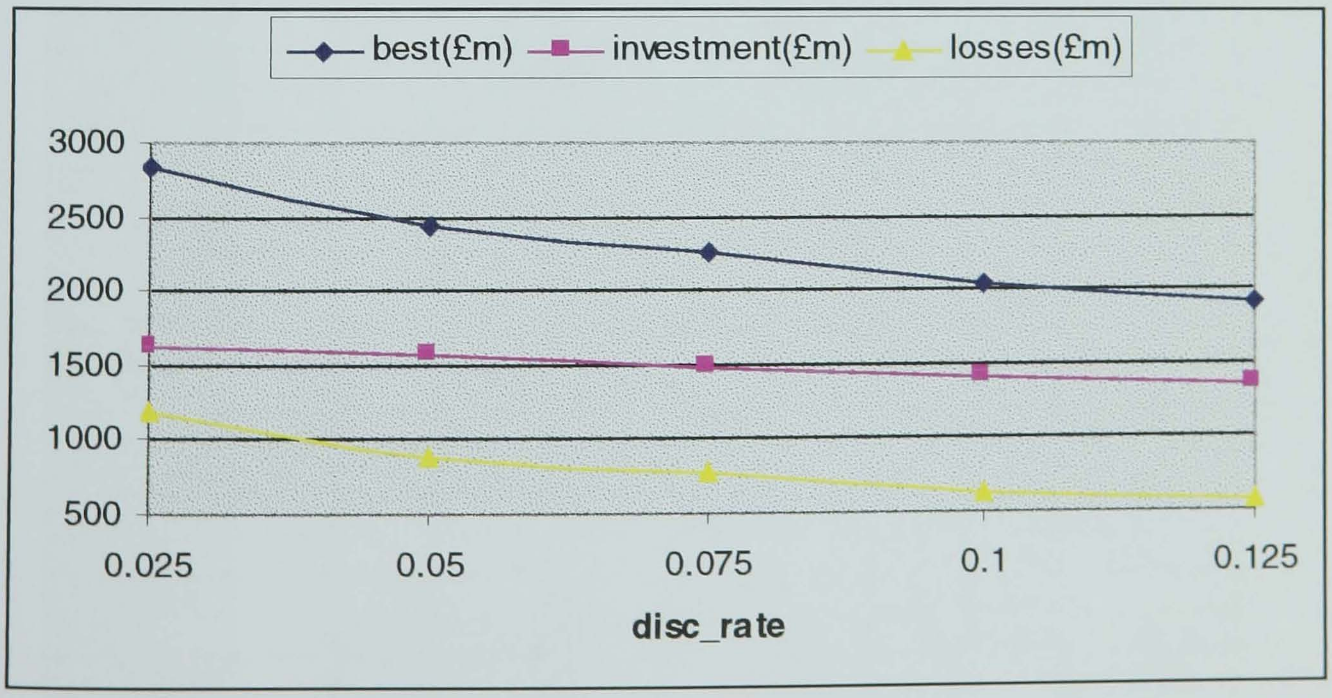
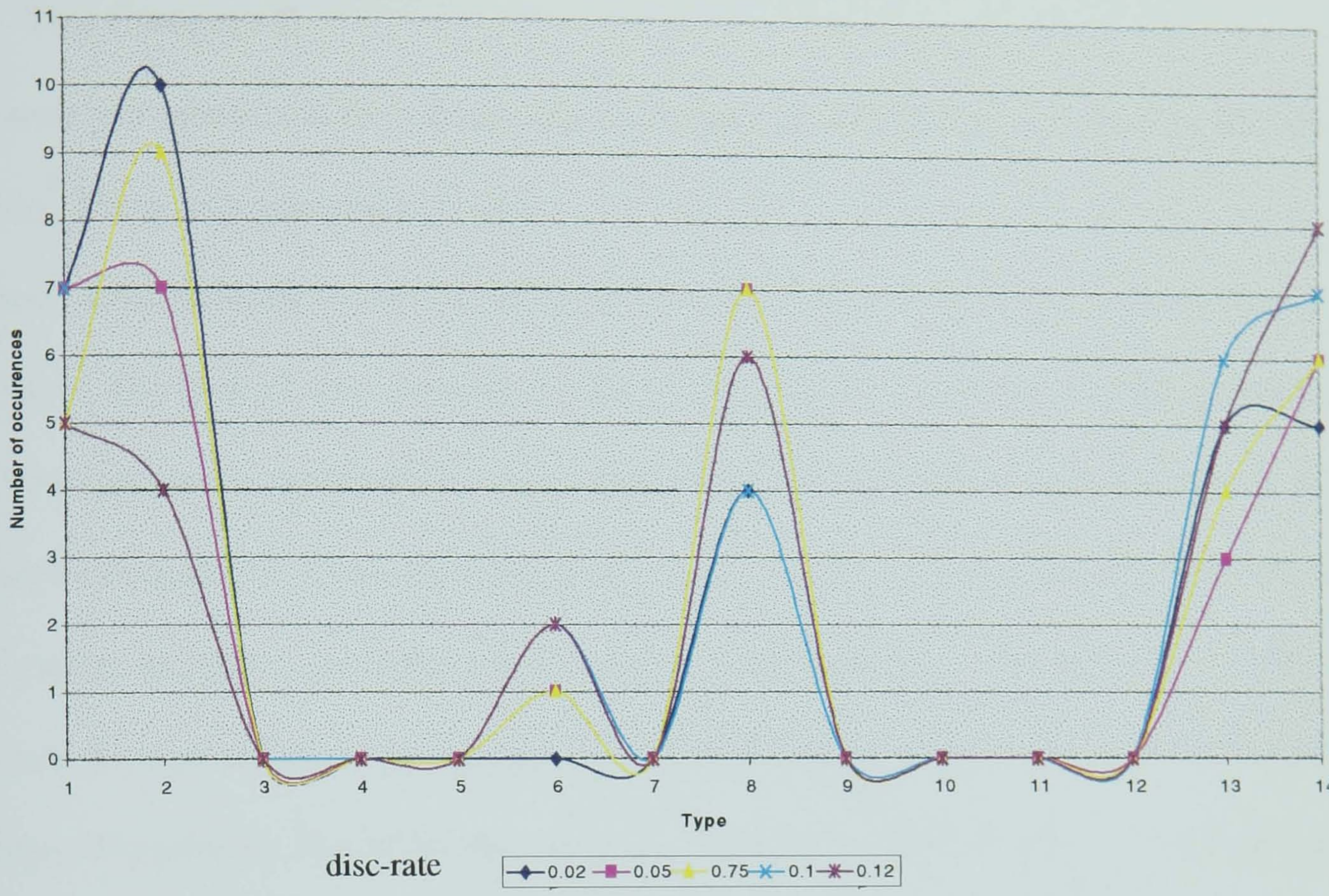


Figure 7.7 Sensitivity of the cost of the network design produced by DCGA to discount rate (disc_rate), for problem class D.

Table 7.3 Line types of the best solutions obtained with the various discount rates considered

Line types					
disc_rate→	0.025	0.05	0.75	0.1	0.125
lines					
1	2	0	0	2	6
2	0	0	0	0	0
3	8	8	8	13	13
4	14	14	14	14	14
5	0	0	0	0	0
6	1	1	1	1	1
7	0	0	0	0	1
8	0	0	0	0	0
9	1	1	1	1	1
10	0	0	14	14	14
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	2	2	2	2	2
15	2	6	8	8	8
16	0	0	0	0	0
17	13	14	14	14	14
18	2	0	2	2	4
19	2	2	2	2	2
20	13	13	13	14	14
21	2	2	2	2	6
22	13	13	13	13	13
23	14	14	14	14	14
24	2	2	2	6	8
25	8	8	8	13	13
26	13	13	8	13	14
27	2	2	2	6	8
28	14	14	14	14	14
29	8	8	13	8	8
30	2	6	6	8	8
31	8	8	13	13	13
32	0	0	0	0	0
33	14	14	0	0	0
34	0	0	0	0	0
35	1	1	2	2	2
36	8	8	8	8	8
37	0	0	0	0	0
38	0	0	0	0	0
39	1	1	2	2	2
40	2	8	8	8	8
41	13	8	8	13	13
42	14	14	14	14	14
43	1	1	1	1	1
44	1	1	1	1	1
45	0	0	0	0	0
46	0	0	0	0	0
47	0	2	0	0	0
48	0	2	2	0	0
49	1	1	1	1	0



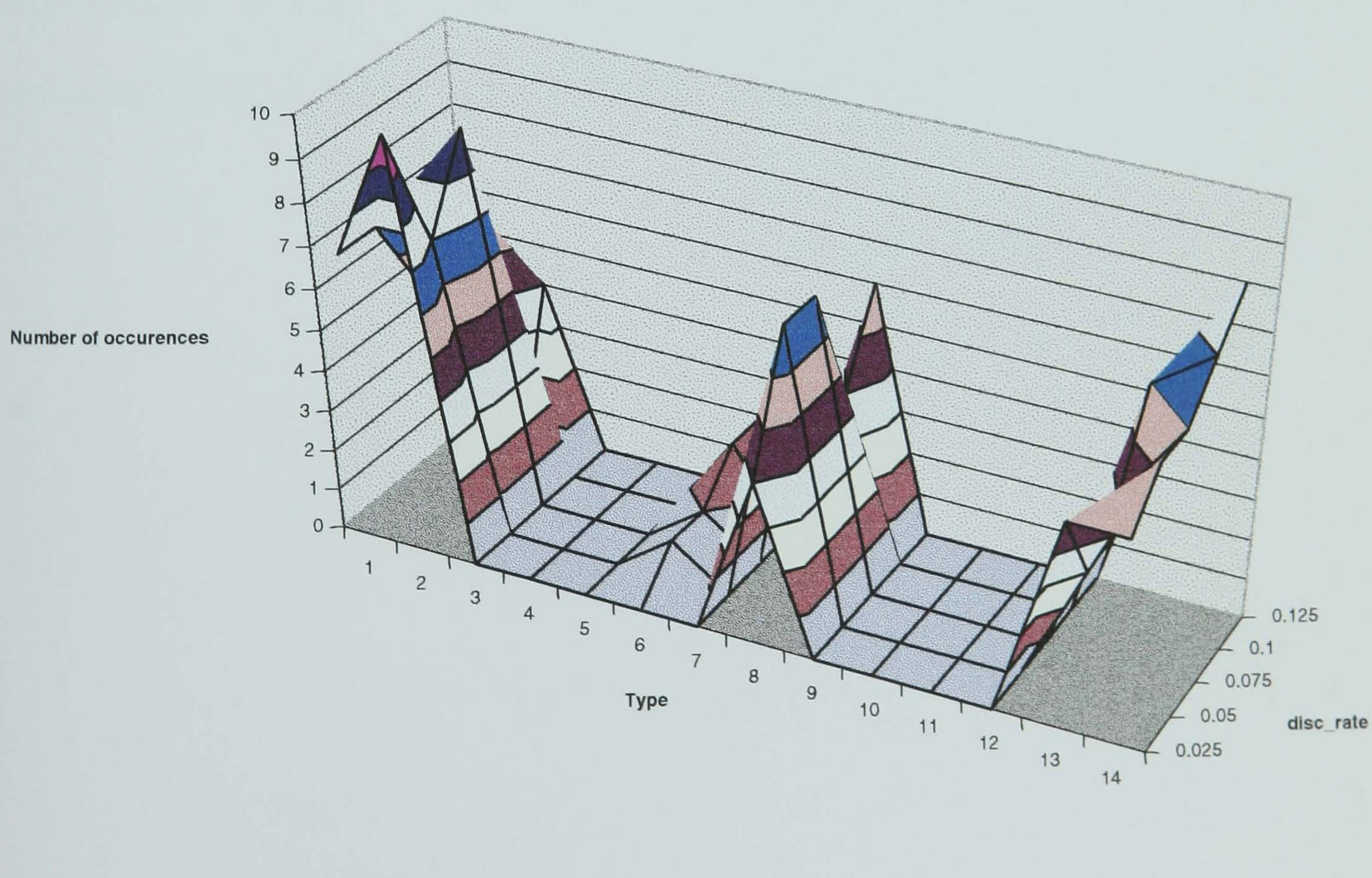


Figure 7.8 Number of occurrences of line types for the best solutions with the various discount rates considered.

7.3.2 Test Category B

These tests are carried out to examine the robustness of the solution provided by DCGA approach. These tests proceed as follows:

The 'best' network design (with the cheapest cost) produced by a specific discount rate is considered. The cost of this network design is then evaluated using the other discount rates. This process is repeated for every 'best' solution obtained in test category A, as illustrated in figure 7.9. The cost evaluations for the various scenarios are grouped into a (5x5) matrix as shown in table 7.4. It can be noted that in a row, the bold number represents the best solution corresponding to the discount rate in the same row (diagonal elements). Therefore it should be cheaper than any other cost in the same row because it has been optimised for this specific discount rate. Figure 7.10 presents a graphical representation of table 7.4. Simulation results demonstrate that the 'best' solution produced by DCGA for problem class D is the optimum among all the cases examined.

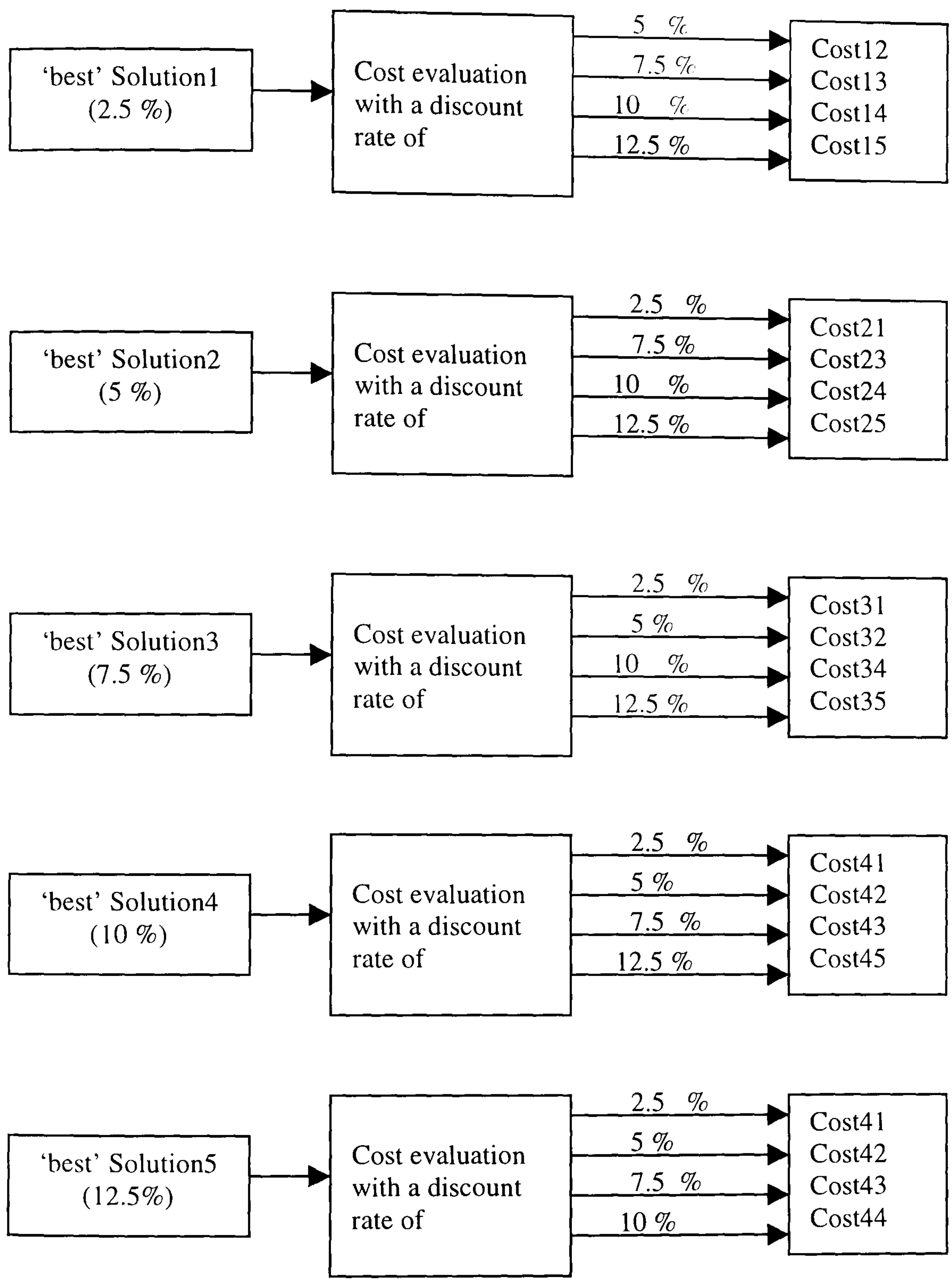


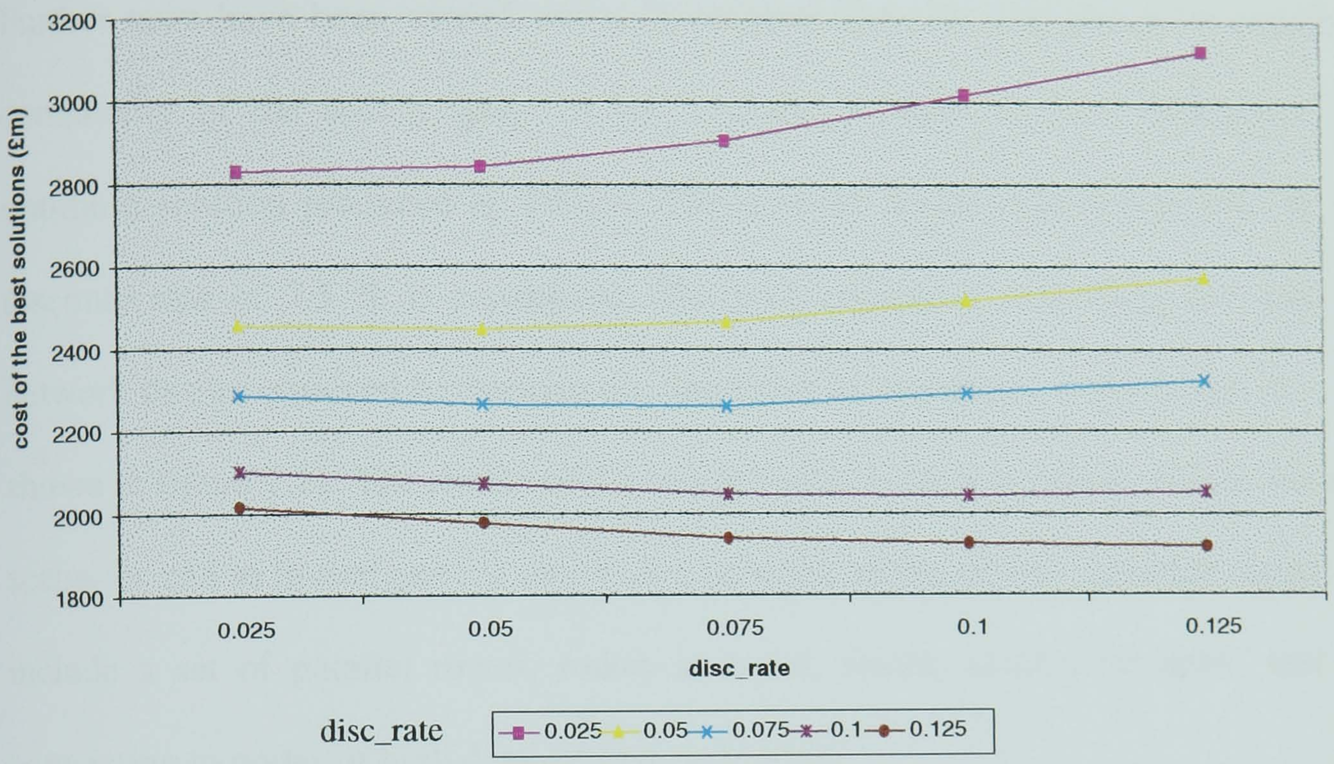
Figure 7.9 Scenarios for test category B

Table 7.4 Evaluation of Scenarios for test category B

disc_rate	optimum	Evaluation with disc_rate	Total cost (£m)	Investment cost (£m)	Loss cost (£m)
0.075	**2261.537**	0.025	2905.1	1483.238	1421.8619
0.075	2261.537	0.05	2462.004	1483.238	978.76578
0.075	2261.537	0.1	2043.894	1483.238	560.65599
0.075	2261.537	0.125	1938.377	1483.238	455.13924
0.05	2445.512	0.025	2840.545	1572.896	1267.6493
0.05	2445.512	**0.075**	**2266.792**	1572.896	693.89557
0.05	2445.512	0.1	2072.73	1572.896	499.83394
0.05	2445.512	0.125	1978.697	1572.896	405.80069
0.025	2828.035	0.05	2455.275	1631.867	823.40753
0.025	2828.035	**0.075**	**2286.631**	1631.867	654.76375
0.025	2828.035	0.1	2103.521	1631.867	471.65449
0.025	2828.035	0.125	2014.775	1631.867	382.90792
0.1	2042.502	0.025	3018.47	1407.095	1611.375
0.1	2042.502	0.05	2516.308	1407.095	1109.2128
0.1	2042.502	**0.075**	**2289.115**	1407.095	882.02025
0.1	2042.502	0.125	1922.857	1407.095	515.76169
0.125	1917.046	0.025	3128.089	1346.905	1781.1841
0.125	1917.046	0.05	2573.014	1346.905	1226.1086
0.125	1917.046	**0.075**	**2321.884**	1346.905	974.9788
0.125	1917.046	0.1	2049.256	1346.905	702.35104

Cost evaluations (£m)					
Optimum at disc_rate	0.025	0.05	0.075	0.1	0.125
Evaluated at disc_rate					
0.025	**2828.035**	2840.545	2905.1	3018.47	3128.089
0.05	2455.275	**2445.511**	2462.004	2516.308	2573.014
0.075	2286.631	2266.792	**2261.537**	2289.115	2321.884
0.1	2103.521	2072.73	2043.894	**2042.502**	2049.256
0.125	2014.775	1978.697	1938.377	1922.857	**1917.046**

Evaluation of Scenarios for test category B organised in a 5x5 matrix



the best solution obtained with a defined discount rate is evaluated with the other disc_rates considered yielding a graph with that defined disc_rate

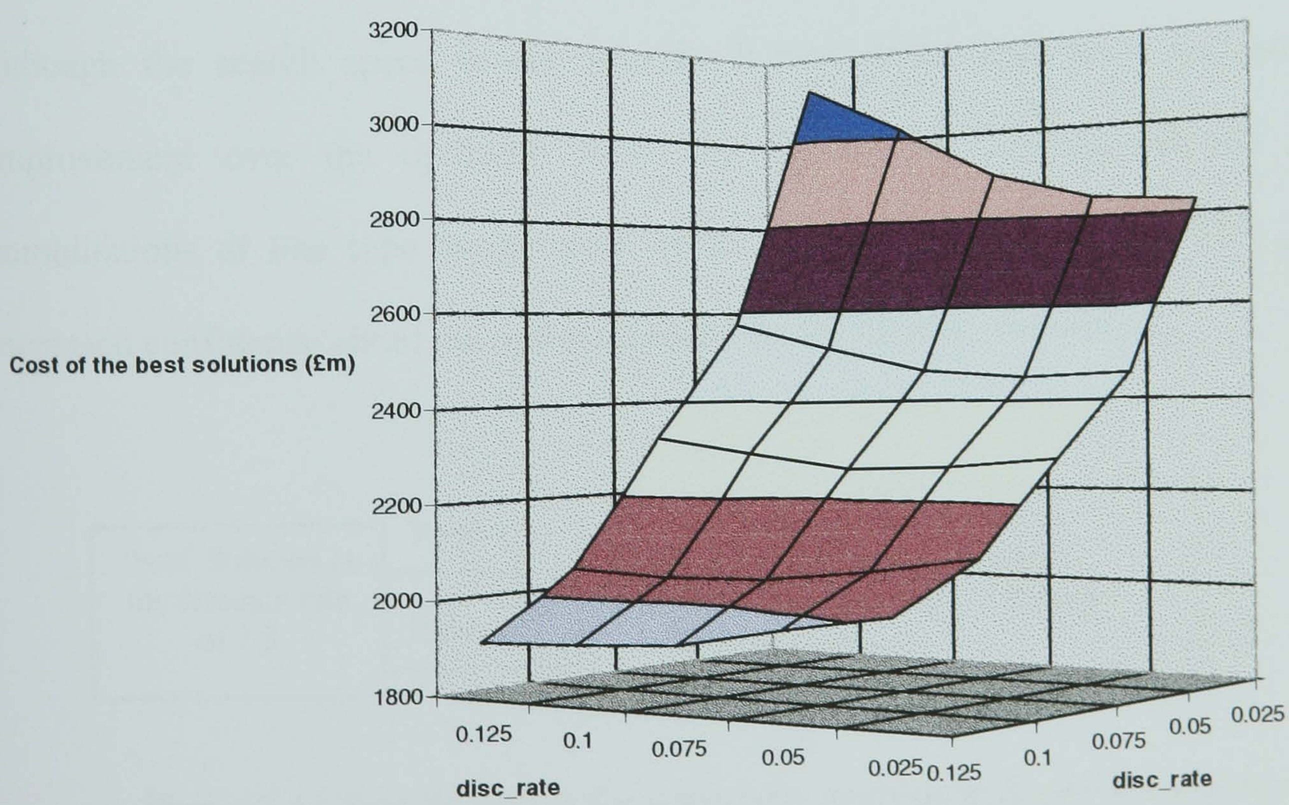


Figure 7.10 Sensitivity of the best solution to discount rate

7.4 Sensitivity Analysis to Problem Class D Decision Variables

Further tests have been carried out with problem class D. The aim is to examine the complexity of the search space of this realistic problem and the robustness of the optimum solution produced by DCGA. The network design obtained by DCGA with a discount rate of 7.5 % is considered. The tests consist of evaluating the cost of the network design obtained by varying the line types of two routes in the range {1...14} as shown in figure 7.11. The choice of the sets of routes is done to reflect various reasonable scenarios and to investigate the interaction between choice of line types for routes. These include a set of parallel routes, routes in series, routes situated far apart, and routes connecting to nodes of high demand and generation.

Simulation results are illustrated in figures 7.12-7.15. It is noticeable that the search space features a number of local optimum. There is symmetry in some figures (figures 7.15 and 7.16). It is important to note that the proposed optimum can be located on any figure although the search space is not smooth. It was found that there was never any improvement over the optimum found by DCGA by carrying out the possible combinations of line type assignments to the set of routes considered. This gives us increased confidence about the optimum found by the Genetic Algorithm.

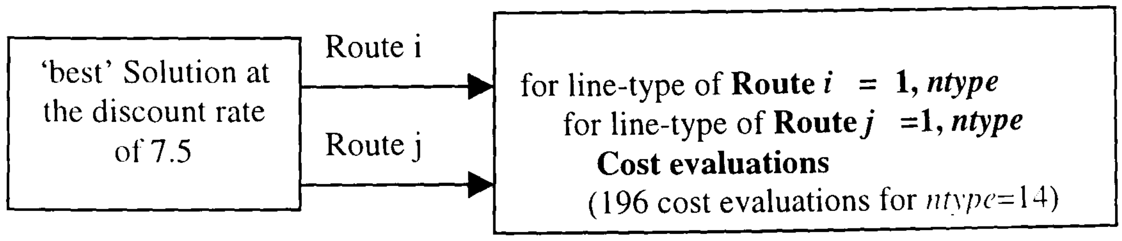


Figure 7.11 Test Scenarios for sensitivity analysis to decision variables

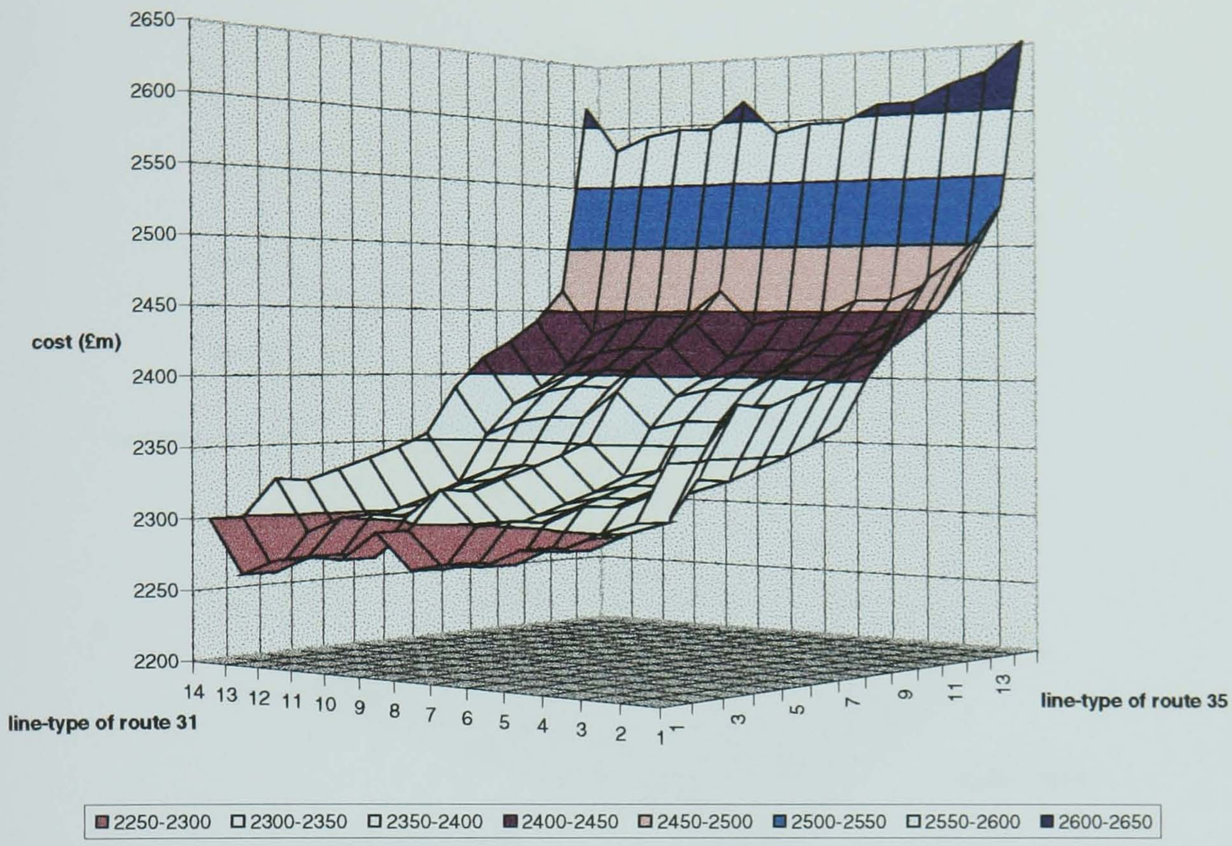


Figure 7.12 Evaluation of the cost of the optimum network design obtained at the discount rate of 7.5 % as a function of two decision variables (choice of line-type of route 31 and route 35). (Optimum choice is type 13 in route 31 and type 2 in route 35).

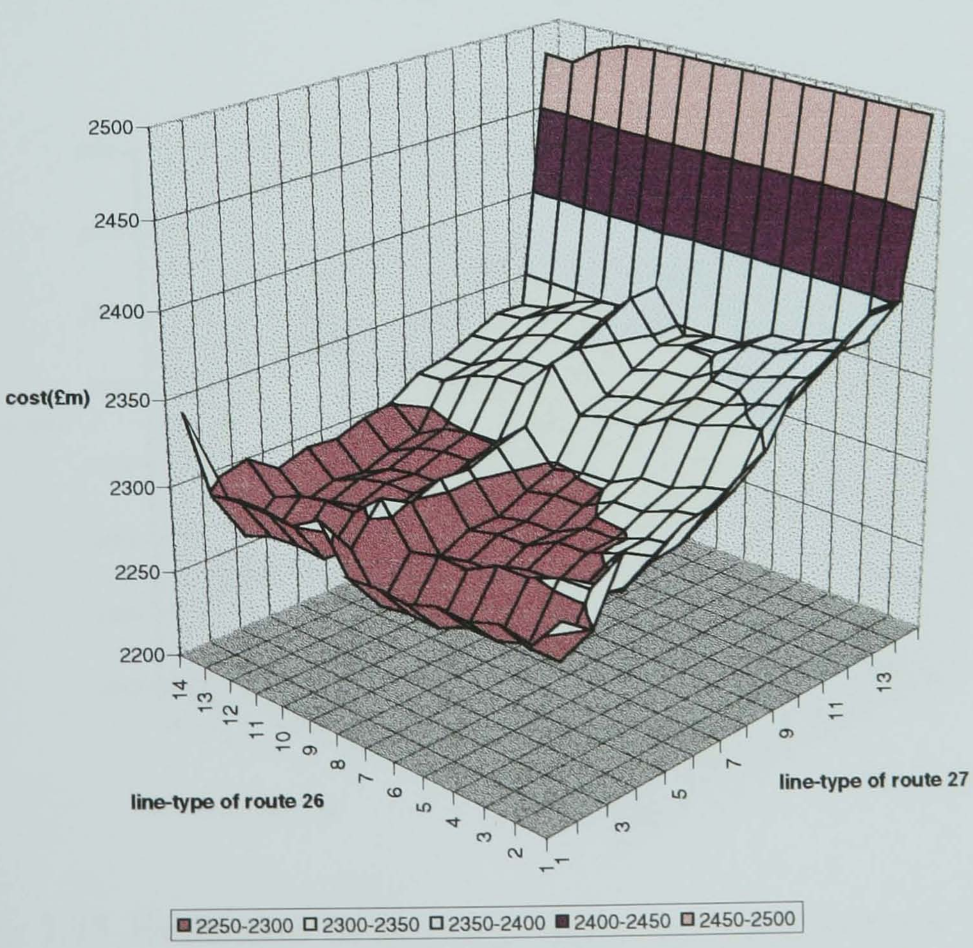


Figure 7.13 Evaluation of the cost of the optimum network design obtained at the discount rate of 7.5 % as a function of two decision variables (choice of line-type of route 26 and route 27). (Optimum choice is type 8 in route 26 and type 2 in route 27).

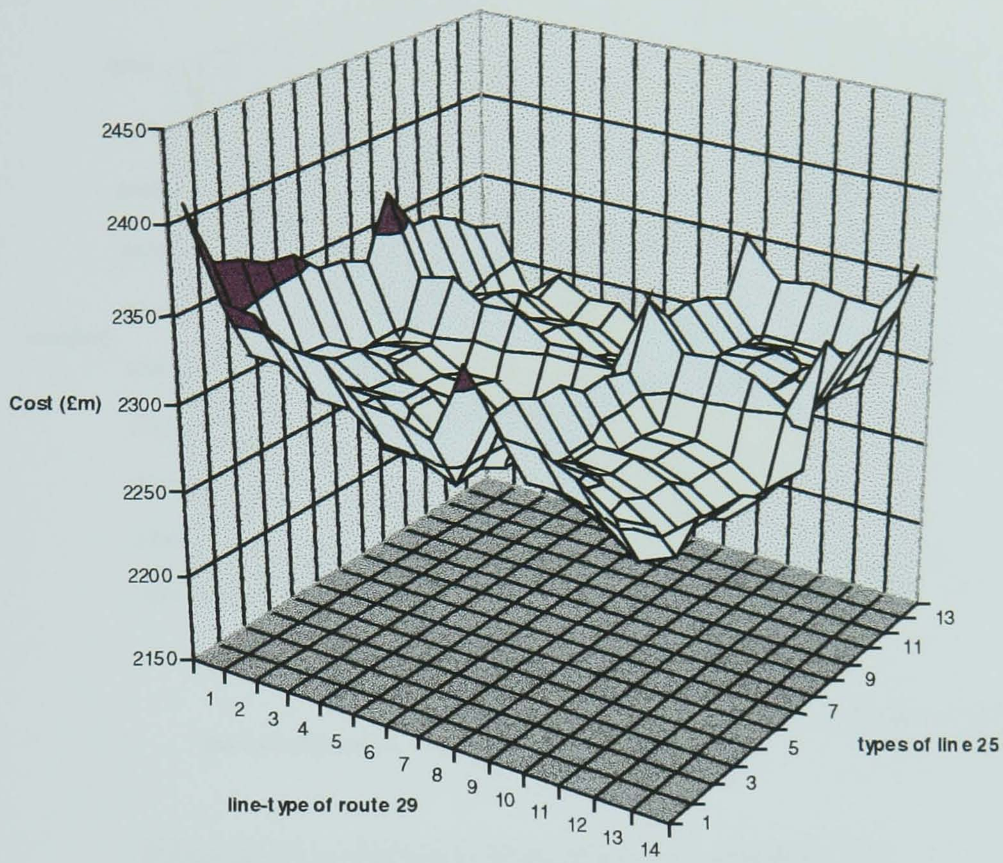


Figure 7.14 Evaluation of the cost of the optimum network design obtained at the discount rate of 7.5 % as a function of two decision variables (choice of line-type of route 25 and route 29). (Optimum choice is type 8 in route 25 and type 13 in route 29).

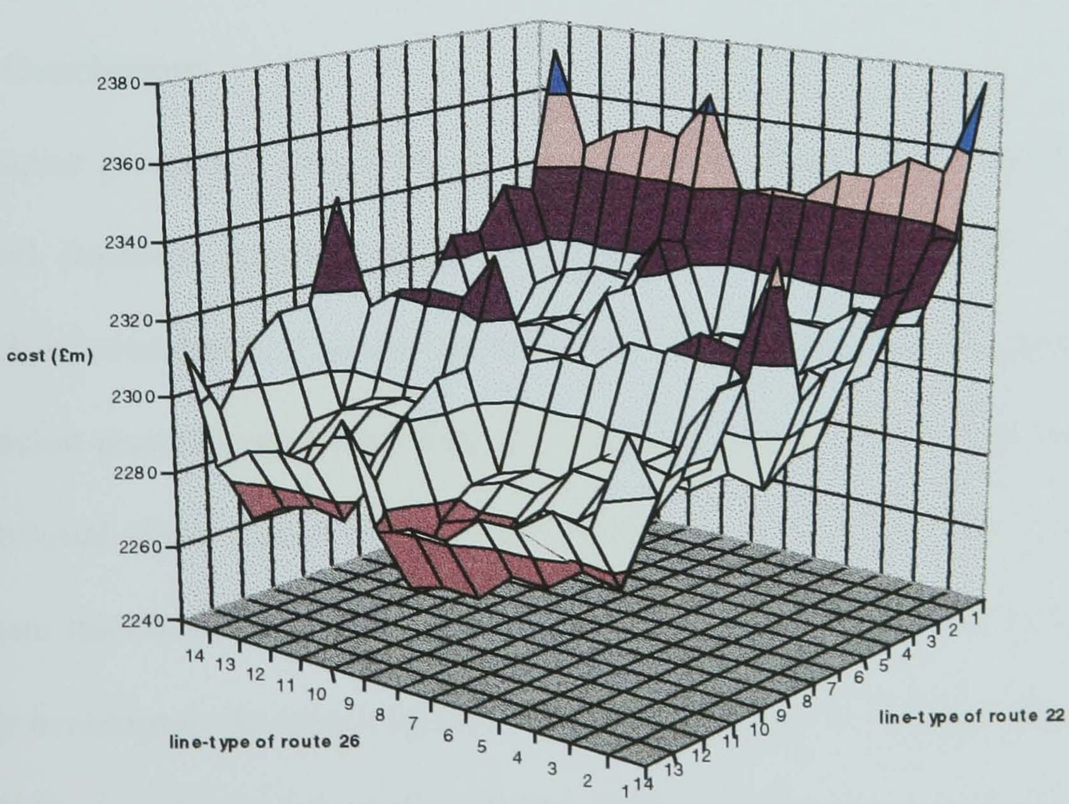


Figure 7.15 Evaluation of the cost of the optimum network design obtained at the discount rate of 7.5 % as a function of two decision variables (choice of line-type of route 22 and route 26). (Optimum choice is type 13 in route 22 and type 8 in route 26).

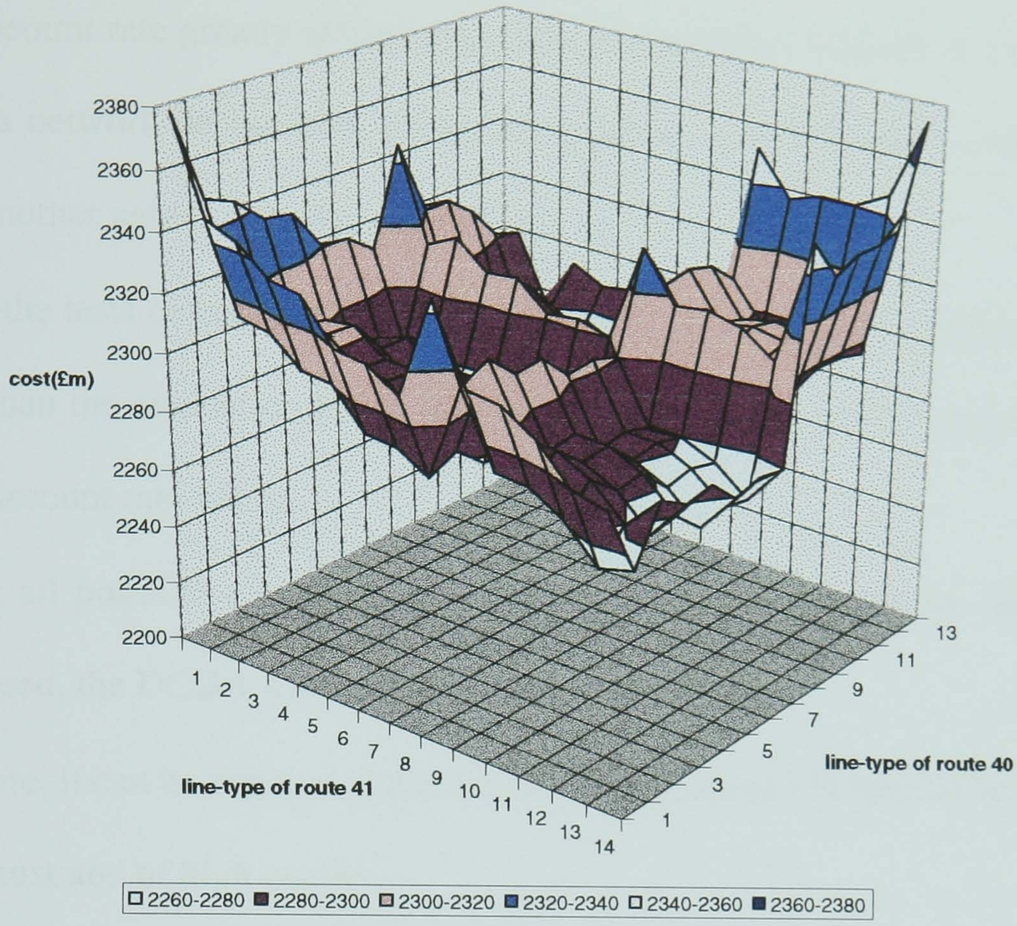


Figure 7.16 Evaluation of the cost of the optimum network design obtained at the discount rate of 7.5 % as a function of two decision variables (choice of line-type of route 40 and route 41). (Optimum choice is type 8 in route 40 and type 8 in route 41).

7.5 Conclusions

This chapter presents a sensitivity analysis of the network design based on the discount rates and decision variables respectively. Only problem Class D with the DCGA approach is considered. The reason being that this problem class represents a realistic transmission-planning problem with no known optima and is solved within reasonable computational time. Three sets of tests have been performed. The objective being to investigate the effect of the discount rate on the best solution provided for problem class D and to investigate the search space of the transmission-planning problem. The tests also provided an opportunity to test the validity of the solution proposed for problem D.

Simulation results showed that:

- The discount rate greatly affects the solution of problem class D. A low discount rate yields a network design with a high net present value whereas a high discount rate gives another network design with a lower net present value.
- For all the tests considered, the DCGA solution obtained at a defined discount rate is better than the evaluation based on this discount rate for DCGA solutions at various other discount rate values.
- Among all possible combinations of line type assignments to the subsets of routes considered, the DCGA solution has proven to be the best.
- Therefore, it can be concluded that the solution proposed by DCGA for problem class D is robust and of high quality.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

There has been considerable interest in Genetic Algorithms, and other heuristic search algorithms (such as Ant Colony Search) rooted in natural and physical processes, among researchers in various application fields. The simplicity, flexibility and robustness of such algorithms has opened up new areas of application, and has also encouraged a re-appraisal of some traditional problems that were either very difficult or even intractable for traditional optimisation techniques. The transmission-planning problem is a complicated, integer-valued, non-convex, non-linear mathematical programming problem. The complexity of the problem arises mainly from the large number of problem variables, combined with the multitude of technical and economical constraints. The goal is to design an electricity transmission network, which is as economical as possible while providing a reliable energy supply.

There are several planning algorithms available for the solution of the long term transmission planning problem, each based on a different interpretation of the system model and choice of the design objective. These include heuristic algorithms and mathematical optimisation techniques.

These techniques can be classified into two main categories:

- 'traditional' or 'incremental' approach,
- 'green-field' or 'non-incremental' approach.

Both categories require:

- load and generation patterns for the target year,

- all possible new routes (lengths and way-leaves),
- the available transmission line types and their corresponding costs.

The main difference is that the traditional approach is restricted by an existing network configuration as opposed to the 'green field' approach which is not constraint by any existing network.

Most planning methods belong to the traditional approach, where transmission planners tend to evaluate a relatively small number of expansion alternatives over a relatively short time horizon.

In this thesis, two novel heuristic methodologies namely, the Deterministic Crowding Genetic Algorithm (DCGA) and Ant Colony Search (ACS), are proposed for the solution of the long term-transmission planning problem based on an original problem formulation termed the 'green field' approach. They also differ from the conventional mathematical counterparts by the following:

- the search proceeds from a population of points,
- use payoff information in the pursuit of the optimum (only the objective function is required),
- use probabilistic transition rules, not deterministic rules, to solve the problem in hand.

Before applying the DCGA (or ACS) model for transmission planning, the problem must be represented in a suitable format that allows the application of the GA operators (or ACS method). This is the problem encoding process. The GA works by maximising a single variable, the fitness function whereas ACS uses an evaluation function indirectly (inspired by the objective function and system constraints) to guide the search.

Therefore, in both methods, the objective function and some of the constraints of the transmission-planning problem must be transformed into some measure of fitness.

Two computer programs have been developed corresponding to ACS and DCGA approaches. They are both implemented on a Pentium 233 MHz processor, under the Windows NT operating system using Fortran 77. Both programs are modular and can be applied to various problems with slight variation of the code to accommodate the new objective function. Two GA representations namely, the binary and integer, are implemented in the DCGA program. However, the binary representation has been adopted.

Two sample-problems are independently considered:

- shortest path problem,
- range of problems derived from a 23-node 49-route transmission network design that represents a simplified version of the England and Wales transmission network.

8.1.1 Shortest Path Problem

We have chosen this particular test because it resembles, to a certain extent, our transmission-planning problem and it has a known optimum. This provides a test bed to investigate the performance of ACS and DCGA algorithms.

Simulation results have shown that both algorithms (DCGA & ACS) are robust and effective in finding the known optimum for the artificial test problem (shortest path problem). Moreover, the ACS algorithm converged in relatively less computational time (a few seconds less).

Experimental results have also demonstrated the importance of synergy in ACS. That is, the search carried out by a given number of ants is more effective with co-operation than without (ants acting independently).

However, the ACS algorithm provides a wider range of convergence regarding the visibility parameter (β, which emphasises the local information). The reason may be that the high visibility in conjunction with low values of trail parameter (α, representing the global information), for this particular test problem, is effective in guiding the search towards convergence. On the other hand, GA does not have the concept of visibility and therefore has less local search capability, resulting in a narrower range of convergence. Consequently, ACS appears to be more reliable than the DCGA model regarding the tuning of the parameter setting for the shortest path problem.

Therefore, it is interesting to apply the ACS & DCGA algorithms to the power transmission-planning problem. However, we can not generalise about the optimum parameter settings nor the behaviour of the ACS & GA from this particular problem. The reason is that the transmission-planning problem is a difficult non-linear, non-convex, discrete-variable constrained optimisation problem.

8.1.2 Cost Optimisation Problems

In order to assess the two proposed models as a planning tool to optimise the configuration of the system, they have been applied to the range of problems (problem classes A, B, C, D, E) derived from a 23-node 49-route transmission network design problem.

The various problem classes are classified according to the objective function and are summarised in the following sections with their corresponding results.

For two of the problem classes the nature of the optimal solution is known in advance from theoretical considerations. These cases provide an opportunity to test the validity of the solutions proposed by GA and ACS approaches. The remaining three problem classes are more realistic, but are also complex and do not have known optima. For these cases, only DCGA has been considered as this approach was found to be significantly more efficient computationally than ACS. Moreover, it is not possible to assess the solutions provided by GA theoretically and it is necessary to rely on numerical experiments.

Any violation of constraints implied by a candidate solution is handled using a penalty function approach, in which penalty costs are incorporated into the fitness function so as to reduce the apparent fitness of an infeasible candidate.

To allow more rapid evaluation of the present network-planning model, the DC load flow approximation has been adopted. The DC load flow provides a linear active power flow model that is sufficiently accurate for the present application and can be used efficiently in conjunction with the Householder modified matrix formula for outage studies.

Extensive tests that consume months of CPU time have been carried out on the sample problem and considering the five classes of cost function.

8.1.2.1 Problem Class A: Cost of Energy Losses Only

For this problem class, the cost of energy losses is to be minimised subject only to the satisfaction of required generation and load levels throughout the network. Other costs and constraints are temporarily neglected. From considerations based on simple electrical network theory, it is apparent that an optimal solution for this case consists of a transmission network design with each available route occupied by the line type having the lowest electrical resistance.

Both methods converged to the known optimum. However, ACS appears to be faster than DCGA. This is due to the exploitation of the local information provided by the visibility in addition to the experience gained by the ants and representing the global information. That is, the solution to that problem with ACS is refined as ants construct their paths (solutions) step by step. The decision to move from one node to another is made with a probability, which is function of the pheromone trail accumulated on that node and the visibility of that node (the node with the cheapest cost is more attractive to ants than other nodes). This problem like the shortest path problem is more suited for ACS than for DCGA.

8.1.2.2 Problem Class B: Investment Cost Only

In this problem class, only the capital cost of transmission lines is considered, subject to satisfying required generation and load levels. Other costs and constraints are temporarily neglected. A theoretical optimal solution for this case consists of a transmission network design based on a minimum length spanning tree (i.e. a radial network with shortest possible total line length) in which the line with lowest capital cost is used throughout.

It has been found that the best solution obtained with both optimisation techniques agreed with the known theoretical optimum for this problem class, which is found to be more challenging for them than problem class A.

Both techniques exhibit comparable convergence characteristic. The best improves fairly quickly in the early stage of evolution, whereas this improvement slows down towards the end of the run. Moreover, it is been observed that the best remains constant for a number of generations (or cycles) before it changes again. This implies that the stopping criteria for a run based on a maximum number of generations (or cycles) should be chosen carefully in order to allow enough exploration. Therefore, if the GA (or ACS) is left running for more generations, it is more likely to get better solutions and the landscape of the best solutions obtained by applying DCGA (or ACS) would be smoother.

However, the DCGA appear to be more efficient computationally than ACS and further modelling has been undertaken with the DCGA approach.

8.1.2.3 Problem Class C: Energy Losses and Investment Cost

This problem class represents a realistic planning problem in which all factors, except security against outages, are considered. In particular, the solutions obtained for this class of problems show how energy loss costs are to be traded-off against initial capital costs. There are no known optimal solutions for this class of problems, and so the solutions obtained by the DCGA approach are of significant interest.

It has been found that the DCGA is very reliable, relatively easy to get the best solutions in the range of parameters specified.

8.1.2.4 Problem Class D: Energy Losses, Investment, and Security Analysis Cost

This problem class represents a more realistic problem with most factors considered. The cost of energy losses and the capital cost of transmission lines are to be minimised subject to the satisfaction of required generation and load levels throughout the network and security constraints. Again no optimal solutions are known in advance. The comparison of the solutions obtained for problem class C with those obtained for class D allow the additional cost of designing a secure network to be assessed.

The DCGA model converged to a best solution which can be located with different parameter settings. The best solution is obtained within reasonable computational time. A sensitivity analysis of the network design based on the discount rates and decision variables respectively is carried out. The objective being to investigate the effect of the discount rate on the best solution provided for problem class D and to investigate the search space of the transmission-planning problem. The tests also provided an opportunity to test the validity of the solution proposed for problem D.

Simulation results showed that:

- There was no further improvement over the best solution obtained by DCGA when it is run with an initial population that includes the best solution.
- The discount rate greatly affects the solution of problem class D. A low discount rate yields a network design with a high net present value whereas a high discount rate gives another network design with a lower net present value.
- For all the tests considered, the DCGA solution obtained at a defined discount rate is better than the evaluation based on this discount rate for DCGA solutions at various other discount rate values.
- Among all possible combinations of line type assignments to the set of routes considered, the DCGA solution has proven to be the best.
- Therefore, it can be concluded that the solution obtained by DCGA for problem class D is robust and of high quality.

8.1.2.5 Problem Class E: Energy Losses, Investment Including Transformers, and Security and Maintenance Analysis Cost

This final problem class represents the full-scale problem with all factors considered. This problem class extends the modelling of the transmission-planning problem D by incorporating additional requirements of the network design. This includes the implementation of maintenance and transformation costs. The aim is to design a cost effective, maintainable, and secure system.

Again no optimal solutions are known in advance. Moreover, it is not possible to assess the solutions provided by GA theoretically and it is necessary to rely on numerical experiments. It is important to note that the computational time increased dramatically when transformer and maintenance costs are considered in the network design. It is been observed that the DCGA requires appropriate tuning in order to obtain the best results.

8.1.3 Global Conclusions

Simulation results have shown the suitability of both approaches to the solution of the transmission-planning problem. They have also demonstrated the effectiveness of both algorithms in the search for the optimum. However, both algorithms require parameter tuning in order to get the best solutions.

The solutions obtained for problem classes A and B agreed with the known theoretical optima giving a degree of confidence in both approaches. However ACS appears to be more computationally expensive than the GA for more realistic problem classes.

For more realistic problems (classes C and D, E), with no known optimum, the best solutions obtained by DCGA are robust and of high quality.

For production purposes, it is recommended to run the programs several times trying different parameter settings to ensure that the solution falls within the area of convergence. Moreover, the DCGA provides other solutions (slightly more expensive) which may be of interest to the planning engineer. These solutions (network designs) might be worthy of consideration due to additional factors, which were not included in the computer evaluation of the design cost.

Because of the flexibility of both methods, further modelling requirements can be included in the fitness function to produce more a realistic model, but the computational time increases with the complexity of the problem. Therefore, there is a trade-off between having a realistic model and the computational time required.

8.2 Proposals for Future Work

This research proposed two heuristic methodologies, namely DCGA and ACS for the solution of the transmission-planning problem based on an original problem formulation termed as the 'green field' approach. This research has sought to answer some of the questions on the DCGA and ACS behaviours on real world problems, however there are a number of areas that provide an opportunity for further work.

- In the present model, the demand/generation profiles represent only the predicted peak values for an immediate future. Future work is required to extend the modelling of transmission planning design to accommodate seasonal load scenarios.
- To allow more rapid evaluation of the present network model, the DC load flow approximation has been adopted. More research can be carried out to enhance the transmission modelling by implementing AC load flow to maintain an acceptable voltage profile and ensure that reactive generation is within the specified limits.
- Because the computational time increases with the size and complexity of the problem, further research can be oriented towards the improvement of the computational efficiency of the optimisation methods without compromising solution quality. A possible solution is to implement DCGA (ACS) on powerful machines that use parallel processing techniques.
- An area of further work is the use of hybrid systems where the merits and trade-offs of two optimisation techniques can combine to optimally solve large-scale transmission systems.

- The GA and ACS models require parameters tuning in order to achieve convergence. This is very time consuming and requires extensive experiments. Further research is required to provide automatic parameter tuning.
- The application of GAs to the 'traditional' or 'incremental' planning problem could be investigated.
- The GA modelling could be extended to allow for the expert user to interact during the evolutionary process and force some interesting changes during the evolutionary process.

References and Bibliography

[1] Aarts E., Lenstra J.K., 'Local Search in Combinatorial Optimisation', *John Willey & Sons,* 1997.

[2] Aitchison P.W., Pereira J. L. R., and Brameller A., 'Extensions and Singularities in Compensated Network Solution Applicable to Security Monitoring', *Proc. IEE,* Vol.134, No.2, March 1987.

[3] Baker J. E., 'Reducing Bias and Inefficiency in the Selection Algorithm', *Proc. of the second International Conference on Genetic Algorithms*, pp 14-21, 1987.

[4] Benders J.F., 'Partitioning Procedures for Solving Mixed-Variables Programming Problems', *Numerische Mathematik,* Vol.4, 1962.

[5] Berrie T.W., 'Electricity Economics and Planning', *IEE Power Series 16, Peter Peregrinus Ltd., London, UK,* 1992.

[6] Bilchev G. and Parmee I., 'Constrained Optimisation with an Ant Colony Search Model', *Adaptive Computing in Engineering Design and Control, Second Conference, Plymouth, UK,* March 1996.

[7] Boese K.D., Kahng A.B., and Muddu S., 'A New Adaptive Multi-Start Technique for Combinatorial Global Optimisation', *Oper. Res. Letters,* 16, pp 101-113, 1994.

[8] Bullnheimer R., Hartl R.F., and Strauss C., 'A New Rank Based Version of the Ant System - A Computational Study', *University of Vienna, Institute of Management Science, Austria,* April 1997.

[9] Boone G. and Chiang H., 'Optimal Capacitor Placement in Distribution Systems by Genetic Algorithms', *International Journal Of Electrical Power and Energy*

Systems, Vol.15, No.3, pp 155-162, 1993.

[10] Chanda R.S. and Bhattacharjee P.K., 'Application of Computer Software in Transmission Expansion Planning Using Variable Load Structure', *Electric Power Systems Research* 31, pp 13-20, 1994.

[11] Chebbo H.M. and Irving M.R., 'Application of Genetic Algorithm to Transmission Planning', *IEE GALISIA Conference*, Glasgow, pp 388-393, 1997.

[12] Chen P.H., and Chang H.C., 'Large-Scale Economic dispatch by Genetic Algorithms', *IEEE Transactions on Power Systems*, Vol.10, No.4, pp 1919-1926, 1995.

[13] Cheng D.T.Y., 'Fast Decoupled Flow with Transformer Representation and Branch Outages', Research Report, *University of Durham, Department of Engineering, Power Systems Control Group*, June 1983.

[14] Chou C.S., and Song Y.H., 'Ant Colony-Tabu Approach for Combined Heat and Power Economic Dispatch', *UPEC 97, Manchester*, pp 605-608, September 1997.

[15] Colorni A., Dorigo M., and Maniezzo V., 'Distributed Optimization by Ant Colonies', *Proceedings of European Conference on Artificial Life*, Paris, France, pp 134-142, 1996.

[16] Dasgupta D., and McGregor D.R., 'Thermal Unit Commitment Using Genetic Algorithms', *IEE Proceedings Part C, Generation, Transmission and Distribution,* 141 (5), pp 459-465, 1994.

[17] Davis L., 'Handbook of Genetic Algorithms', *VanNosttrand Reinhold, New York,* 1991.

[18] De Jong K.A., 'An Analysis of the Behaviour of a Class of Genetic Adaptive

Systems', *Ph.D. Thesis, University of Michigan, Ann Arbor, USA*, 1975.

[19] Di Caro G. and Dorigo M., 'AntNet: A Mobile Agents Approach to Adaptive Routing', *Technical Report 97-12, Iridia, Universite Libre de Bruxelles, Bruxelles, Belgium,* 1997.

[20] Ding H., El-Keib A.A., and Smith R.E., 'Optimal Clustering of Power System Networks Using Genetic Algorithms', *Electrical Power Systems Research*, Vol.30, No.3, pp 209-214, 1994.

[21] Dorigo M., 'Optimisation, Learning, and Natural Algorithms', *PhD Thesis, Politecnico di Milano*, 1992.

[22] Dorigo M.and Gamberdella L.M., 'Ant Colony System: A Co-operative Learning Approach to the Travelling Salesman Problem', *IEEE Transactions on Evolutionary Computation*, Vol.1., No.1, April 1997.

[23] Dorigo M. and Gamberdella L.M., 'Ant-Q: A Reinforcement Learning Approach to the Travelling Salesman Problem', *Proceeding of ML-95, Twelfth International conference on Machine Learning, Morgan Kaufmann*, pp 252-260, 1995.

[24] Dorigo M. and Gamberdella L.M., 'Ant-Q: A Co-operative Learning Approach to Combinatorial Optimisation Problem', *Technical Report, Iridia, Universite Libre de Bruxelles, Bruxelles, Belgium, European Union*, January 1995.

[25] Dorigo M. and Gamberdella L.M., 'A Study of Some Properties of Ant-Q', *Proceedings of PPSN IV- Fourth International Conference on Parallel Problem Solving from Nature, Berlin, Germany,* September 22-27 1996.

[26] Dorigo M. and Gamberdella L.M., 'Ant Colonies for the Travelling Salesman Problem.', *BioSystems Press, International Journal 15-BIOSYS97,* 1997.

[27] Dorigo M., Maniezzo V., and Colorni A., 'The Ant System: Optimisation by a Colony of Co-operating Agents', *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, Vol.26, No.1, pp 1-13., 1996.

[28] Dorigo M., Maniezzo V., and Colorni A., 'The Ant System: Optimisation by a Colony of Co-operating Agents', *IEEE Transactions on Systems, Man, and Cybernetics,* 1995

[29] Dusonchet Y.P. and El-Abiad A., 'Transmission Planning Using Discrete Dynamic Optimization', *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-92, pp 1358-1371, 1973.

[30] El-sobki S.M., El-Metwally M.M., and Farrag M.A., 'New Approach for Planning High-Voltage Transmission Networks', *IEE Proceedings*, Vol.133, Pt.C, No.5, July 1986.

[31] Eschelman L., Caruana R., and Schaffer D., 'Biases in the Crossover Landscape', *Proc. 3rd Int. Conf. on Genetic Algorithms*, 1989.

[32] Feo T.A. and Resende M.G.C., 'Greedy Randomised Adaptive Search Procedures',. *Journal of Global Optimisation*, *6*, pp 109-133, 1995.

[33] Finch J.W. and Besmi M.R., 'Genetic Algorithms Applied to a Power System Stabiliser', *Proc. of First IEE/IEEE Int Conf. on Genetic Algorithms in Engineering Systems: Innovations and Applications*, U.K., IEE Conference Publication No.414, pp 100-105.

[34] Fouad A.A., Venkataraman, and Davis J.A., 'An Expert System for Security Trend Analysis of a Stability-Limited Power System', *IEEE Transaction on Power Systems* , Vol.6, No.3, August 1991.

[35] Francisco D., Galiana, Donald T.M., and Miguel A.M., 'Expert Systems in Transmission Planning', *Proceeding of the IEEE*, Vol.80, No.5, pp 712-725, May 1992.

[36] Gallego R.A., Monticelli A., and Romero R., 'Comparative Studies on Non-Convex Optimization Methods for Transmission Network Expansion Planning', *IEEE Transaction on Power Systems*, Vol.13, No.3, pp 822-828, August 1998.

[37] Gallego R.A., Monticelli A., and Romero R., 'Transmission System Expansion Planning by an Extended Genetic Algorithm', *IEE Proceeding on Generation Transmission and Distribution*, Vol.145, No.3, pp 329-335, May 1998.

[38] Gallego R.A., Alves A.B. and Monticelli A., 'Parallel Simulated Annealing Applied to Long Term Transmission Network Expansion Planning', *IEEE Transaction on Power Systems*, Vol.12, No.1, pp 181-188, 1997.

[39] Gamberdella L.M., Tailard ED., and Dorigo M., 'Ant Colonies for the QAP', *IDSIA, Lugano, Switzerland, Technical Report IDSIA 97-4*, 1997.

[40] Gamberdella L.M. and Dorigo M., 'Solving Symmetric and Asymmetric TSPs by Ant Colonies', *Proc. IEEE Int. Conf. Evolut. Comp. (ICEC'96)*, IEEE Press, 1996.

[41] Garver L.L., 'Transmission Network Estimation Using Linear Programming', *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-89, pp 1688-1697, 1970.

[42] Geoffrion A.M., 'Generalisised Benders Decomposition', *Journal of Optimisation Theory and Applications*, Vol.10, 1972.

[43] Gilles M.L., 'Optimum HVAC Transmission Expansion Planning - A New Formulation', *IEEE Transaction on Power Systems*, PWRS-1, pp 48-56, 1986.

[44] Glover F., 'Tabu Search - part II', *ORSA Journal on Computing*, Vol.2, No.1, pp 4-32, 1990.

[45] Glover F., 'Tabu Search - part I', *ORSA Journal on Computing*, Vol.1, No.3, pp 190-206, 1989.

[46] Glover F., Laguna M., 'Tabu Search', *Kluwer Academic Publishers*, Boston, USA, 1997.

[47] Goldberg D.E., 'Genetic Algorithms in Search, Optimisation and Machine Learning', *Addison-Wesley*, Reading, Mass., 1989.

[48] Granvill S. and Alli, 'Mathematical Decomposition Techniques for Power System Expansion', *EPRI Report* RP 2473-6, Vol.2, 1988.

[49] Holland J.H., 'Adaptation in Natural and Artificial Systems', *MIT Press*, Cambridge, Mass., 1975.

[50] Hsiao Y.T., Chiang H.D., Liu C.C., and Chen Y.L., 'A Computer Package for Optimal Multi-objective VAR Planning in Large Scale Power Systems', *IEEE Transaction on Power Systems*, Vol.9, No.2, May 1994.

[51] Hulselman M., Muller H. and Wertzlinger H., 'Economic Short Term Scheduling for a Run River Plant Chain by Combined Linear Programming and Genetic Optimisation.', *Proc. On Int. Conf. On Power System Control and Management (PSCM 96)*, UK, IEE Conference Publication No.421, pp 42-46, 1996.

[52] Iba K., ' Reactive Power Optimisation by Genetic Algorithm', *IEEE Trans. On PWRS*, Vol.9, No.2, pp 685-692, May 1994.

[53] Ilic'-Spong M. et al., 'Secondary Voltage Control Systems Pilot Point Information', *IEEE Transaction on Power Systems*, Vol.3, No.2, pp 660-668,

May 1988.

[54] Johnson D.S., 'Local Optimisation and the Travelling Salesman Problem', *Proc. 17th coll. Automata, Lang. and Prog.*, Vol.443 of LNCS, pp 446-461, Springer, 1990.

[55] Kaelbling L.P., Littmann L.M., and Moore A.W., 'Reinforcement Learning: A Survey', *J. Artif. Intell. Res.*, vol.4, pp 237-285, 1996.

[56] Kaltenbach J.C., Peschon J. and Gehrig E.H., 'A Mathematical Optimization Technique for the Expansion of Electric Power Transmission Systems', *IEEE Trans. on Power Apparatus and Systems*, Vol.PAS-89, pp 113-119, 1970.

[57] Kern J.K., Young M.P. and Kwang Y.L., 'Optimal Long Term Transmission Expansion Planning Based on Maximum Principle', *IEEE Transaction on Power Systems*, Vol.3, No.4, pp 1494-1501, November 1988.

[58] Khatib H., 'Financial and Economic Evaluation of Projects in the Electricity Supply Industry', *IEE Power Series 23*, London, 1997.

[59] Kirkpatrick S., Gelatt C.GD., and Vecchi M.P., 'Optimization by Simulated annealing.', *Science*, 220, pp 671-680, 1983.

[60] Lai L.L., 'Intelligent System Applications in Power Engineering', *John Wiley & Sons Ltd, Baffins Lane, Chichester, West Sussex, PO19 1UD, England*, 1998.

[61] Lai L.L., Ma J.T., Wong K.P., Yokoyama R., Zhao M., and Sasaki H., 'Application of Evolutionary Programming to Transmission System Planning', *IEE of Japan Power and Energy*, 1996.

[62] Lansbery J.E., Wozniack P.E., and Goldberg D.E., 'Optimal Hydro Generator Governor Tuning with a Genetic Algorithm.', *IEEE Trans. on Energy Conversion*, Vol.7, No.4, pp 623-630, December 1992.

[63] Latorre-Bayona G. and Perez-Arriaga I., 'Chopin, a Heuristic Model for Long Term Transmission Expansion Planning', *IEEE Transaction on Power Systems*, Vol.9, No.4, pp 1886-1894, November 1994.

[64] Lee K.Y., Bai X., and Park Y.M., 'Optimisation Method for Reactive Power by Using a Modified Simple Genetic Algorithm', *IEEE Trans. On PWRS*, Vol.10, No.4, pp 1843-1849, November 1995.

[65] Lee S.T.Y., Hicks K.L., and Hnyilicza E., 'Transmission Expansion by Branch-and-Bound Integer Programming With Optimal Cost-Capacity Curves', *IEEE PES Winter Meeting*, New York, N.Y., January/February 1974.

[66] Leerink L.R., Schultz S.R., and Jabri M.A., 'A Reinforcement Learning Exploration Strategy Based on Ant Foraging Mechanisms', *Proceedings of the sixth Australian Conference on Neural Networks*, Sydney, Australia, 1995.

[67] Levi V.A. and Calovic M.S., 'Linear-Programming-Based Decomposition Method for Optimal Planning of Transmission Network Investments', *IEE Proceedings*-C, Vol.140, No.6, November 1993.

[68] Levi V.A. and Calovic M.S., 'A New Decomposition Based Method for Optimal Expansion Planning of Large Transmission Networks', *IEEE Trans. on PWRS*, pp 937-943, August 1991.

[69] Mahfoud S., 'Crowding and Pre-selection in Genetic Algorithms', *Parallel Problem Solving from Nature 2, R. Manner and B. Manderrick (Ed.s)*, Elsevier, pp 27-36, 1992.

[70] Maifeld T. and Sheble G.B., 'Short Term Load Forecasting by a Neural Network and Refined Genetic Algorithm', *Electrical Power Systems Research*, 31 (3), pp 147-152, 1994.

[71] Maniezzo V., Colorni A., and Dorigo M., 'The Ant System Applied to the Quadratic Assignment Problem', *Universite Libre de Bruxelles, Bruxelles, Belgium*, Technical Report IRIDIA/94-28, 1994.

[72] Monticelli A., santos Jr.A., Pereira M.V.F., Cunha S.H., Parker B.J., and Praca J.G.G., 'Interactive Transmission Network Planning Using a Least-Effort Criterion', *IEEE Transaction on Power Apparatus and Systems*, PAS-101, No.10, pp 3919-925, October 1982.

[73] Meliopoulos A.P. and Debs A.S., ' Impact of Improved on-line Security Dispatching on Long-term Transmission Expansion Planning', *Proc. Of the 1977 PICA Conf.*, Toronto , Ontario, Canada, May 24-27 1977.

[74] Miranda V., Ranito V., and Proenca L.M., 'Genetic Algorithms in Optimal Multi-stage Distribution Network Planning', *IEEE transactions on Power Systems*, Vol.9, No.4, pp 1927-1933, 1994.

[75] Nara K., Satoh T., Kitagawa M., 'Distribution Systems Loss Minimum Reconfiguration Using Genetic Algorithms', *IEEE transactions on Power Systems*, Vol.7, No. 3, pp 1044-1057, August 1992.

[76] Oliveira G.C., Costa A.P.C., and Binato S., 'Large scale Transmission Network Planning Using Optimzation and Heuristic Techniques', *IEEE Transaction on Power Systems*, Vol.10, No.4, November 1995.

[77] Orero S.O. and Irving M.R., 'Large Scale Unit Commitment Using Hybrid Genetic Algorithms', *International Journal of Electric Power and Energy Systems*, 19 (1), pp 45-55, 1997.

[78] Orero S.O., 'Power System Generation Scheduling and Optimisation Using Evolutionary Computation Techniques', *Thesis, Brunel*, June 1996.

[79] Orero S.O., and Irving M.R., 'A Genetic Algorithm for Network Partitioning in Power System State Estimation', *IEE Int. Conf. On Control*, Exeter, UK, 1996.

[80] Pereira M.V.F., Pinto L.M.V.G., Cunha S.H.F., and Oliveira G.C., 'A Decomposition Approach to Automated Generation-Transmission Expansion Planning', *IEEE Trans. On PAS*, Vol.PAS-104, No.11, pp 3074-3083, November 1985.

[81] Puntel W.R., Reppen N.D., Ringlee R.G., Platts J.E., Ryan W.A. and Sullivan P. J., 'An Automated Method for Long Range Planning of Transmission Networks', *Proceeding of the 8th PICA Conference, Minneapolis, Minn*, pp 38-46, 1973.

[82] Reeves C.R., 'Modern Heuristic Techniques for Combinatorial Problems', *McGraw-Hill, London*, 1995.

[83] Reinelt G., 'Fast Heuristics for Large Geometric Travelling Salesman Problems', *ORSA Journal on Computing*, 4, 2, pp 206-217, 1992.

[84] Reinelt G., 'The Travelling Salesman: Computational Solutions for TSP Applications', Vol.840 of *LNCS. Springer Verlag*, 1994.

[85] Renders J.M and Flasse S.P, 'Hybrid Method Using Genetic Algorithms for Global Optimization', *IEEE Transaction on Systems, Man, and Cybernetics_Part B: Cybernetics*, Vol.26, No.2, April 1996.

[86] Romero R., Gallego R.A. and Monticelli A., 'Transmission System Expansion Planning by Simulated Annealing', *IEEE Transaction on Power Systems*, Vol.11, No.1, pp 364-369, February 1996.

[87] Romero R. and Monticelli A., 'A Zero-One Implicit Enumeration Method for Optimizing Investments in Transmission Expansion Planning', *IEEE Transaction on Power Systems*, Vol.9, No.3, August 1994.

[88] Romero R. and Monticelli A., 'A hierarchical Decomposition Approach for Transmission Network Expansion Planning.', *IEEE Transaction on Power Systems*, PWRS-9, No.1, pp 373-380, February 1994.

[89] Rudnick H., Palma R., Cura E. and Silva C., 'Economically Adapted Transmission Systems in Open Access Schemes – Application of Genetic Algorithms', *IEEE Transaction on Power Systems*, Vol.11, No.3, pp 1427-1434, August 1996.

[90] Santos A., Franca P.M. and Said A., 'An Optimization Model for Long Range Transmission Expansion Planning', *IEEE Transaction on Power Systems*, Vol.4, No.1, February 1989.

[91] Sawey R.M. and Zinn C.D., 'A Mathematical Model for Long Range Expansion Planning of Generation and Transmission in Electric Utility Systems', *IEEE Transaction on Power Apparatus and Systems*, Vol.PAS-96, pp 657-666, March/April 1977.

[92] Schoonderwoerd R., Holland O.E., Bruten J.L., and Rothkrantz L.J.M., 'Ant-Based Load Balancing in Telecommunications Networks', *Adaptive Behaviour*, vol.5, No.2, pp 169-207, 1997.

[93] Shebble G.B., Maifeld T.T., Brittig K., and Fahd G., 'Unit Commitment by Genetic Algorithms with Penalty Methods and a Comparison of Lagrangian Search and Genetic Algorithm-Economic Dispatch Algorithm', *International Journal of Electric Power and Energy Systems*, 18 (6), PP 339-346, 1996.

[94] Spears W.M., De Jong K.A., 'On the Virtues of Parameterised Uniform Crossover', *Proc. of the fourth International Conference on Genetic Algorithms*, 1991.

[95] Spears W.M., De Jong K.A., Back T., 'an Overview of Evolutionary Computation', *Proceedings of European Conference on Machine Learning*, Lecture notes in AI 667: Berlin, Springer Verlag, pp 442-459, 1993.

[96] Song Y.H., 'Modern Optimization Techniques in Power Systems', *Kluwer Academic Publishers*, 1999.

[97] Stutzle T., and Hoos H., 'Improvements on the Ant System: Introducing the Max-Min Ant System', *3rd International Conference on Artificial Neural Networks and Genetic Algorithms*, University of East Anglia, Norwich, UK, 1997.

[98] Stutzle T., and Hoos H., 'Max-Min Ant System and Local Search for Combinatorial Optimisation Problems', *2nd International Conference on Metaheuristics - MIC97*, Sophia-Antipolis, France, July 21-24 1997.

[99] Stutzle T., and Hoos H., 'The Max-Min Ant System and Local Search for the Travelling Salesman Problem', *IEEE 4th International Conference on*

Evolutionary Computation, IEEE Press, pp 308- 313, 1997.

[100] Taylor A.J.E., Irving M.R., Sterling M.J.H., 'Power System Partitioning Using Genetic Optimisation', *Universities Power Engineering Conference*, UK, pp 607-610, 1990.

[101] Teive R.C.G., Silva E.L. and Fonseca, 'A Cooperative Expert Systems for Transmission Expansion Planning of Electrical Power Systems', *IEEE Transaction on Power Systems*, Vol.13, No.2, pp 636-640, May 1998.

[102] Tsamasphyrou P., Renaud A. and Carpentier P., 'Transmission Network Planning: An Efficient Benders Decomposition Scheme', *Proceedings 13th Power Systems Computation Conference (PSCC)*, Trondheim, June 28-July2nd 1999.

[103] Van Laarhooven P.J.M. and Aarts E.H., 'Simulated Annealing: Theory and Applications', *D. Reidel Publishing Company, Holland*, 1987.

[104] Walters D.C. and Shebble G.B., 'Genetic Algorithms Solution of Economic Dispatch with Valve Point Loading', *IEEE transactions on Power Systems*, 8 (3), pp 1325-1332, 1993.

[105] Yeh E.C., Venkata S.S. and Sumic Z., 'Improved Distribution System Planning Using Computational Evolution', *Proceeding of IEEE PICA Conf.*, pp 530-560, 1995.

[106] Yin X. and Germay N., 'Investigations on Solving Load Flow Problem by Genetic Algorithms', *Electrical Power Systems Research*, Vol.22 (3), pp 151-163, 1991.

[107] Youssef H.K. and Hackam R., 'New Transmission Planning Model', *IEEE Transaction on Power Systems*, Vol.4, No.1, February 1989.

Publications

[1] Chebbo H.M., Irving M.R., 'Application of Genetic Algorithms to Transmission Planning', *Proc. Of the IEE Genetic Algorithms in Engineering Systems: Innovation and Applications*, University of Strathclyde, pp 388 – 393, 1997.

[2] Irving M.R., Chebbo H.M. and Orero S.O., 'Transmission Network Planning Using Genetic Algorithms', 'Modern Optimisation Techniques in Power Systems', Edited by Y.H. Song, *Kluwer Academic Publishers*. 1999.