Use of decision science to aid selection of genetically superior animals

A Thesis presented in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in

Animal Science

At Massey University, Palmerston North, New Zealand

Ryan Leith Sherriff

2010

Abstract

This thesis is concerned with a theoretical simulation model for pig breeding, as part of the ongoing search for the “perfect” genotype. The starting point is an additive model to investigate how accurately the classical, infinitesimal model predicts genetic gain for traits controlled by few loci and few alleles. This initial investigation demonstrates that the infinitesimal model is robust, providing that at least 15 loci are controlling a trait and there is symmetry in the allele distributions.

A Genotype-Pig (GE-Pig) model is then developed to apply the additive effects of alleles on sub-phenotypic traits like maximum protein deposition, minimum lipid to protein content in the whole body, *ad libitum* digestible energy intake, energy for maintenance requirement and water content in the whole body. These parameters are then used in a nutrient partitioning simulation model to growth a pig and calculate traditional breeding traits such as average daily gain, feed conversion ratio, and backfat thickness for any combination of alleles. Three algorithms, Genetic Algorithm, Tabu Search, and Simulated Annealing, are used to investigate the GE-Pig model and find optimal combination of alleles for different dietary and selection objective situations. The two diets investigated were either of a low or high quality, and the three selection objectives used were, maximising average daily gain, minimizing feed conversion ratio, and minimizing back fat. A graphical method is developed for easy comparison of the genotypes.

Of the algorithms, the Genetic Algorithm performed the best, followed by Tabu Search and finally Simulated Annealing. It is demonstrated that, in general, there is a different, single, optimum for any given selection objective and diet. However under the back fat selection objective, both diets produce the same optimal genotype. Also there are many similarities between the optima for the average daily gain and feed conversion ratio selection objectives. When the theoretical minimum number of generations of selection to the optima is considered, the feed conversion ratio selection objective is the quickest for a breeding program to achieve the optimal solutions, followed by back fat, then

average daily gain. It is demonstrated that diet also has an effect on the theoretical number of generations.

A Multiple selection objective, using relative economic values applied to the individual selection objectives, is also investigated. For both diets, the majority of the multiple selection objective solutions are in the vicinity of the feed conversion ratio optima, indicating that feed conversion ratio is the most prominent factor. It is also demonstrated that the optimal solution is most affected by the objective parameter weights under low diet conditions.

Publications

Studies completed during candidature, some of which are reported in this thesis have been presented in the following communications:

Morel, P.C.H., Alexander, D.L.J., Sherriff, R.L, and Wood, G.R., 2001. Maximisation of gross margin in intensive livestock production. Acta Horticulture. 556: 97-103.

Sherriff, R.L., Blair, H.T., Morel, P.C.H., Sridharan, R., Tweedie, J.W. and Wood, G.R. 2003. Theoretical considerations on genetic selection. In Manipulating Pig Production IX. J. Paterson (Ed). 66.

Sherriff, R.L., Blair H.T., Morel, P.C.H., Sridharan R. and Wood, G.R. 2003. Searching for the Best Genotype(s) to Optimize Breeding Value for Pigs Using Heuristics. INFORMS, Atlanta, USA, October 19-22. http://www.informs.org/Conf/Atlanta2003/ATLANTA%20Tuesday%20PDF.pdf

Acknowledgements

This work would not have been possible without the help and encouragement of a number of people. In particular I would like to thank:

- The Agricultural and Marketing Research and Development Trust (AGMARDT) for their financial support.

- My numerous supervisors, with particular thanks to Patrick Morel and Graham Wood, who have been involved and supported me throughout this work.

- My parents for their support and allowing me to complete this thesis during work hours.

- My wife, for believing in me.

Table of Contents

List of Figures

List of Tables

Chapter 1 – Introduction

1.1 Background

Most current and traditional methods of breeding methods, Figure 1.1, rely on phenotypic traits and their associated heritabilities, genetic variances and genetic correlation (infinitesimal model). The genetic architecture of the traits is treated as a black box, with no knowledge of the number of genes which affect each trait (Dekkers and Hospital 2002). The performance of the selection program is proportional to the accuracy with which the breeding values (genetic merit of animals which breeders wish to improve, for example, average daily gain) can be estimated. It is generally accepted that the link between the genes and observed phenotypic value is largely influenced by genotype by environment interaction (GxE) and imposes difficulties with current breeding methods (Doeschl-Wilson *et al*. 2007).

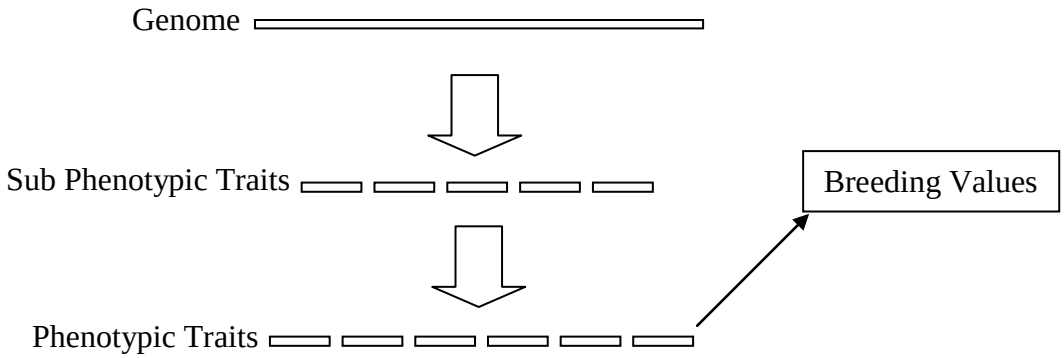

Figure 1.1: Traditional breeding method.

One of the benefits of the genomics revolution is knowledge of genes and genetic markers and the mapping of quantitative trait loci (QTL), Figure 1.2. At present the increased rate of genetic improvement from use of DNA-based tests for QTL is small. However, in the future, with significantly more QTL identified, large gains are possible (Goddard 2003). Although for complex traits that are controlled by several QTL of moderate or low effect, or that are subject to high environmental variation, epistasis between QTL or between QTL and the genetic background, it is risky to carry out selection solely on the basis of marker effects, without confirming the estimated effects by phenotypic evaluation (Dekkers and Hospital 2002).

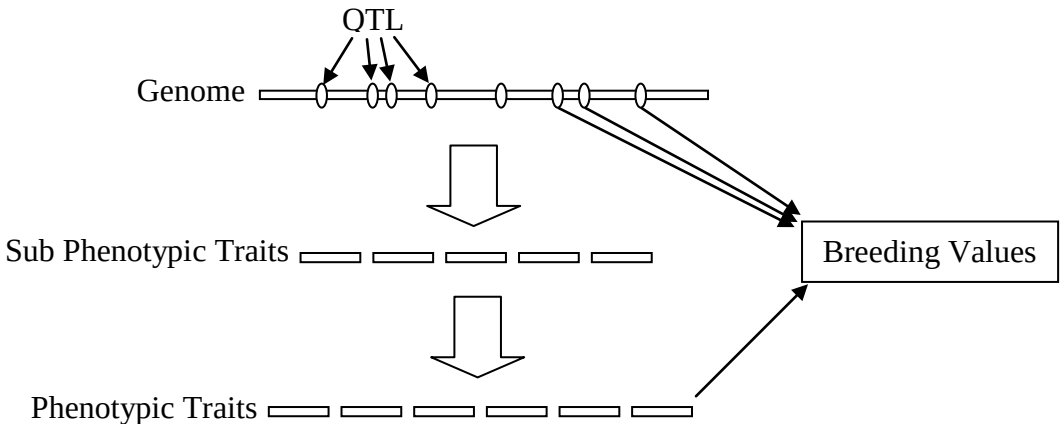

Figure 1.2: Breeding method that includes QTL information.

Recently there has been a new marker type, single nucleotide polymorphism (SNP), which has been gaining higher popularity (Vignal *et al*. 2002). SNP looks for the presence or absence of individual alleles, giving the ability to produce finer genetic maps than QTL. It is foreseeable that in the future SNP will be widely included in breeding programs in a similar way to QTL, Figure 1.3. For example since 2005 SNP have been used in New Zealand in cattle breeding schemes (Spelman *et*. *al*., 2006).

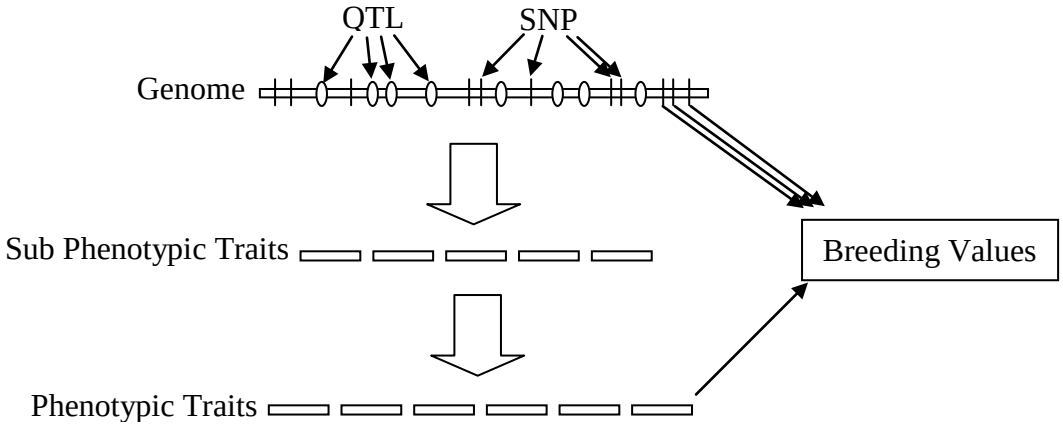

Figure 1.3: Breeding method that includes SNP information.

Mechanical growth models predict phenotypic traits by using sub phenotypic traits that are considered more closely related to the underlying biology, and hence the genes. These sub phenotypic traits are generally accepted to be less influenced by GxE interactions and more closely represent the genes of the animal. Research is in progress into the ability to use model inversion on these mechanical growth models to determine the sub phenotypic traits from the observed phenotypic traits (Doeschl-Wilson *et al*.

2007). This would then allow for the inclusion of the less environmentally influenced sub phenotypic traits in the breeding programs, Figure 1.4.

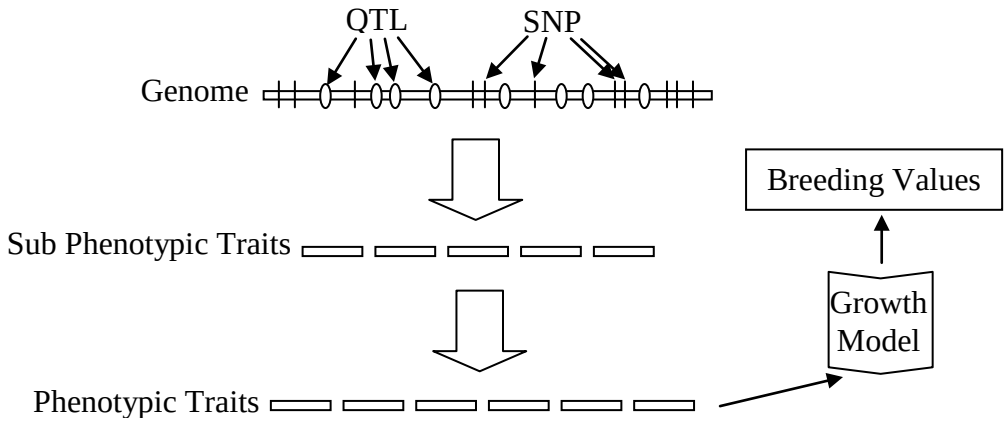

Figure 1.4: Breeding method that includes mechanical growth model inversion.

Another method under research (Kell 2002) is to treat the entire genotype to phenotype mapping as a complex tree of simple functions. Machine learning is then applied to the tree to determine a tree that generates the desired output, Figure 1.5.

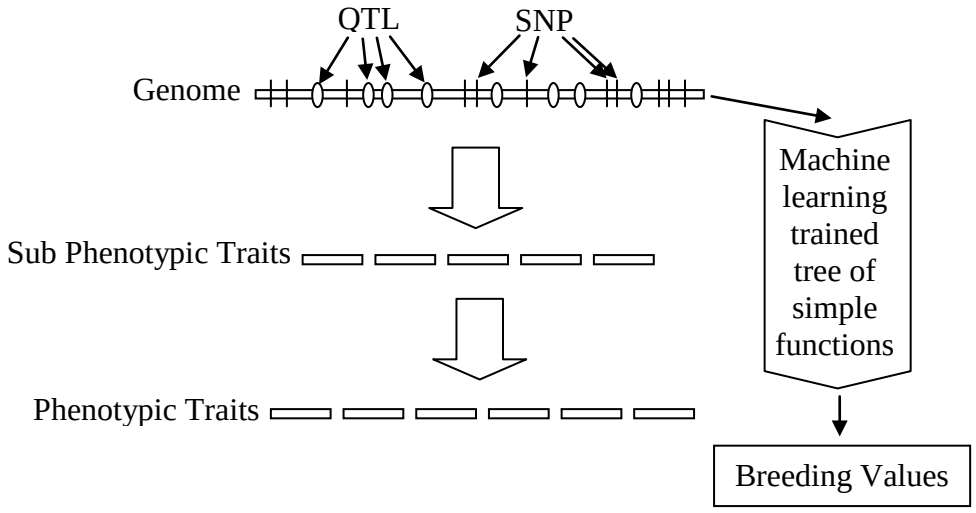

Figure 1.5: Breeding method with machine learning trained tree of simple functions.

1.2 Aim of thesis

The aim of this thesis is to investigate whether current animal growth models can be enhanced to include genetic information, Figure 1.6. This would allow the use of powerful optimization procedures in the search for genetically superior animals. It would also allow for genetic tests to be completed and an animal's potential modelled,

before reproductive age, to provide more accurate information for breeding programs. This research builds on current knowledge in the specialised area of pig modelling. The process of the investigation involved finding a minimum size for the genetic structure that was then developed to attach to the existing pig growth model. Optimization procedures were then written to investigate the feasibility of finding, for various objectives, the optimal genome. The basic concepts of the infinitesimal model, growth models and optimization procedures, Linear Programming, Genetic Algorithms, Tabu Search, and Simulated Annealing, used in the thesis are outlined below in the following sections of this chapter.

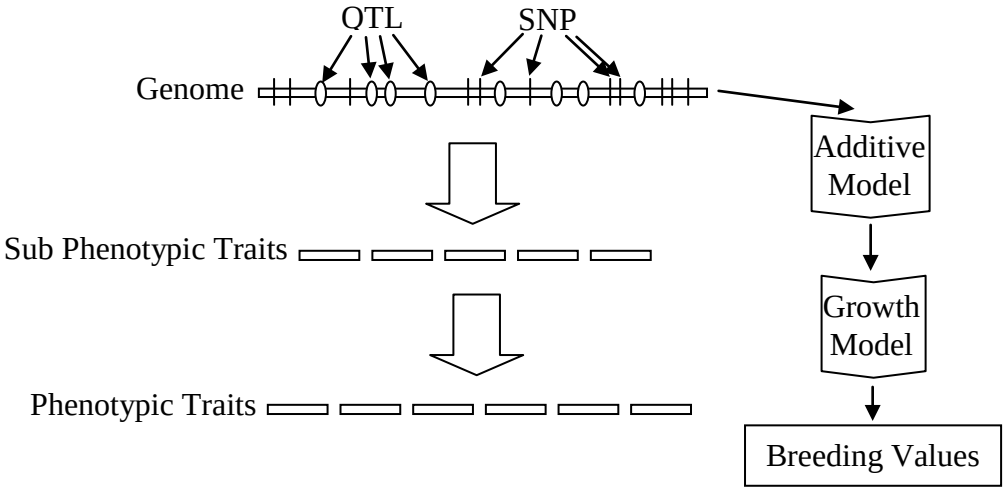

Figure 1.6: Breeding method with genetically enhanced mechanical growth model.

1.3 Infinitesimal model

Accurate predictions of breeding values and the expected genetic gain of various breeding programs are important for animal breeders, in the selection of replacement stock and the comparison of alternative selection schemes. The complete sequencing of the genome of various organisms and the downstream discovery of protein function is leading to an explosion of information that defines new phenotypes. The challenge for quantitative geneticists will be to include these new phenotypes in the prediction of breeding values and expected rates of genetic change.

The infinitesimal model (Fisher 1918) has traditionally been used to predict the expected genetic gain. The model assumes that a trait is controlled by an effectively infinite number of unlinked and non-epistatic loci, each with an infinitesimal effect. Under this model the allele frequencies are unchanged by selection. Large changes in

the mean occur by summing small allele frequency changes at a large number of loci. By assuming the trait under selection shows a normal distribution and that a linear parent-offspring regression underlies selection response, the single-generation response to selection can be predicted from knowledge of the appropriate variance components and the selection intensity of the breeding program (Falconer and MacKay 1996).

The validity of these assumptions (Falconer and MacKay 1996) and predictions (Clayton *et al*. 1957, Hill and Caballero 1992) of the infinitesimal model have been considered when applied to quantitative traits. Effects on inbreeding coefficients (Curik *et al*. 2002) and the effects of having linear and non-linear parent offspring regressions (Mbaga and Hill 1997) have been investigated. Some of this research has been recently reviewed (Goddard 2001). However, there is increasing evidence that a small number of loci or genes may have a strong impact on the inheritance of production traits (Pirchner 1988; Falconer and MacKay 1996; Jeon *et al*. 1999).

1.4 Growth models

Animal growth models have been studied from several different view points, reflecting the intended applications.

Empirical models (Parks 1982, Devir et al. 1995) allow the weight gain of animals to be expressed as a relatively simple function. This type of model allows for experimental comparison of various genetic compositions and feeding regimes (Thompson et al. 1985, Thompson and Barlow 1986). Empirical fetal growth models have also been developed for sheep (Koong et al. 1975) and for pigs (van Oijen et al. 1993). Empirical models however do not give an in depth understanding of the system under study.

Mechanistic models are preferable for research and practical applications (Baldwin and Sainz 1995), as they properly address the underlying mechanisms of a system. A number of mechanistic models have been developed for cattle (Keener 1979, Koong et al. 1982, Oltjen et al. 1986, Korver et al. 1988), for pigs (Pomar et al. 1991, Skorupski et al. 1995a, b, De Lange 1995), and for sheep (Wallach et al. 1986, Finlayson et al. 1995). A generic mechanistic model has also been developed (Vetharaniam et al. 2001).

This current work is based on the pig model from the work of De Lange 1995. Models such as that of De Lange require specific sub phenotypic trait parameter inputs that generalise the genetic merit of the animal. These sub phenotypic trait are:

Pd_{max} maximum protein deposition

LP_{min} minimum lipid to protein ratio

E_m energy requirements for maintenance

p feed intake capacity expressed as a percentage of the standard National Research Council (NRC) digestible energy curve

w chemical vs. physical body composition. Used in determining the amount of water present in the pig.

In this current work the genetic merit for the sub phenotypic traits is supplied directly by the genetic structure, which has the inherent correlations between the sub phenotypic traits as part of the structure. This has the advantage of allowing optimization procedures to be applied with the knowledge that the input parameter correlations are satisfied.

1.5 Linear programming

Linear programming has been used in this work to generate the weighting of the various loci, whilst maintaining the correlations for the input parameters of the pig model.

The field of linear programming, and the Simplex Algorithm (Dantzig and Thapa 1997) for solving linear programming problems, was developed by George B. Dantzig. Since its discovery by Dantzig in 1947, linear programming has grown hugely and is today one of the most widely used tools in industry for planning and scheduling.

Linear programming allowed, for the first time, the structuring of complex allocation and resource problems. The Simplex Algorithm then allows for the optimal solving of the structure.

Dantzig and Thapa 1997, p1, state that “linear programming is concerned with the maximization or minimization of a linear objective function in many variables subject to linear equality and inequality constraints”.

In mathematical terms, the general linear programming problem may be stated as,

$$\begin{aligned} \text{Minimize/Maximize:} \quad & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \text{Subject to:} \quad & \boldsymbol{A}_1\boldsymbol{x} \leq \boldsymbol{b}_1 \\ & \boldsymbol{A}_2\boldsymbol{x} \geq \boldsymbol{b}_2 \end{aligned}$$

where:

$\boldsymbol{c}^{\mathrm{T}}$ is the transpose of the coefficients of the objective function vector c

$\boldsymbol{x}$ is the solution vector

$\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are the coefficient matrices of the constraints

$\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are the right hand side vectors of the constraints.

The goal is to find the optimal combination of $\boldsymbol{x}$ that minimizes/maximizes the objective. The standard form for a linear program is

$$\begin{aligned} \text{Minimize:} \quad & \boldsymbol{c}^{T}\boldsymbol{x} \\ \text{Subject to:} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned}$$

Again, Dantzig and Thapa 1997, p63 state that “The Simplex Method is a very efficient procedure for solving large practical linear programs on the computer”. In general the method starts with a linear program that is in standard form at a known feasible solution, often the origin, in graphical terms. The algorithm then pivots along the boundary of the feasible region from one vertex (extreme point) to the next, until either the optimal solution is found or the problem is determined to be unbounded. Since the 1980’s linear programs that move through the interior of the feasible set of solutions have become popular (interior point methods), but these methods have not been used in this work, as the problem size does not require this level of power.

1.6 Genetic Algorithm

The Genetic Algorithm is the first of the three optimization procedures used in this work to investigate the GE-Pig model. The GE-Pig model is defined and developed in §5.3.

Genetic Algorithms (Holland 1975) use Darwin’s ‘survival of the fittest’ paradigm from natural evolution to efficiently find high quality solutions to a wide range of problems. The problems may be as diverse as constrained optimization (Homifar et al. 1994), multiprocessor scheduling (Hov et al. 1994), jobshop scheduling (Davis 1985), road

maintenance planning (Faw et al. 1994), and computer aided molecular design (Venkatasubramanian et al. 1994).

The Genetic Algorithm is a stochastic algorithm that simulates the evolution process. Although the evolution process is random, it is guided by a selection mechanism based on the fitness of individual structures. Relatively high fitness structures have a larger chance to survive and produce even higher fitness offspring. There is thus an increase in the overall fitness of a population in each new generation. The Genetic Algorithm uses a set of genetic operators to select, recombine, and alter existing structures to direct the search towards improvement.

Although the Genetic Algorithm was based on ideas from evolution, it is today an extremely powerful and adaptable tool which is applied to a wide range of fields.

1.7 Tabu Search

This work also uses Tabu Search as an optimization procedure to investigate the GE-Pig model.

The philosophy of Tabu Search, as given by Glover and Laguna 1997, p1, is to “derive and exploit a collection of principles of intelligent problem solving. In this sense it can be said that Tabu Search is based on selected concepts that unite the fields of artificial intelligence and optimization”.

Tabu Search is based on four early developments. The first development, arising from the work of Fisher and Thompson 1963 and Glover 1963 in jobshop scheduling problems, involved an active restructuring of multiple decision rules to allow new decision combinations, thus giving the possibility of carrying search paths beyond simple local optima. This strategy also involved alternating between decision rules in a probabilistic manner.

The second development, involving work from Glover 1966, 1969, in the integer programming field, was the use of a flexible individualized memory and associated conditional restrictions for controlling the solutions that are allowably generated.

The third development likewise involves work from integer programming and utilizes a strategy of purposely visiting both feasible and infeasible regions in successive waves. This strategic oscillation developed from Glover's 1968 work.

The final development involves the introduction of surrogate constraint ideas from integer programming (Glover 1965). Surrogate constraints are generated by combining constraints, with the goal of yielding information not contained in the original constraints.

Overall, Tabu Search has grown to be concerned with imposing restriction to guide a search process to negotiate otherwise difficult regions. These restrictions are imposed or created by referring to specially designed memory structures. Tabu Search thus incorporates the principles of long term and short term memory. That is, Tabu Search regards information in two different ways, firstly as retained long term data, and secondly as transient information that affects decision making for short periods of time. Tabu Search is so flexible that the definition of long term and short term memory will vary from one implementation to another. This fact can be a strength as it allows for great flexibility and adaptation to a given problem. However this can also be a weakness because a large amount of time can be spent in customising the algorithm to best suit the problem.

Tabu Search as it stands today is an extremely flexible meta-heuristic method which is customized for any given problem.

1.8 Simulated Annealing

The third optimization procedure used in this work to investigate the GE-Pig model is Simulated Annealing.

Simulated Annealing was introduced independently by Kirkpatrick, Gelatt, and Vecchi 1983 and Cerny 1985. Their work was based on an algorithm proposed by Metropolis et al. 1953, for the efficient simulation of the process of a solid coming to thermal equilibrium. Kirkpatrick et al. established an analogy between minimizing the cost function in an optimization and the slow cooling of a solid. Lundy and Mees 1986

proved that Simulated Annealing converges with probability close to one, provided that certain conditions are met, including an infinitely long cooling schedule.

Simulated Annealing can be viewed as a general optimization technique for solving combinatorial optimization problems.

1.9 Thesis structure

Chapter 2 introduces the additive model by firstly describing the genetic and environmental effects, followed by examples of how alleles are passed from one generation to the next and how genetic and phenotypic values are calculated. The chapter then develops the theory behind the model, giving the distribution involved and the formula for calculating genetic gain and expected genetic value. The chapter then go on to theory of the infinitesimal model and concludes with the variance components of the model.

Chapter 3 describes the simulations that were carried out with the model presented in Chapter 2. Two evaluation parameters are used to compare the simulation results to the theory, the first, Λ, compares the simulation results to the infinitesimal model predictions of genetic gain, and the second parameter, δ, compares the simulation results to the prediction from the model analysis in chapter 2. The results are then compared to determine how small the additive model can be while the infinitesimal model still predicts accurate genetic gains.

Chapter 4 introduces a modified additive model that allows the shape of a 5-allele distribution to be altered and compares the results of simulations with the infinitesimal model genetic gain predictions.

Chapter 5 introduces the GE-Pig model that is used throughout the rest of the thesis. The model takes an existing pig growth model and adds an additive genetic structure to the sub phenotypic inputs of the pig model. The desired means, variances and correlations for the additive structure and presented and the method for calculating the various loci weights is developed. The results from Chapters 2 to 4 are used to set minimum sizes on the additive model structure implemented in the GE-Pig model. Also the diets used throughout the rest of the thesis are presented.

Chapter 6 presents the implantations of the search algorithms, Genetic Algorithm, Tabu Search, and Simulated Annealing, used through out the optimizations performed on the GE-Pig model.

Chapter 7 through 9 present the simulation results for three different optimization objectives, maximize average daily gain, maximize feed conversion ratio, and minimize back fat. Results from each of the three search algorithms are presented and compared.

Chapter 10 investigates a multiple selection objective based on relative economic values of average daily gain, feed conversion ratio, and back fat. Low, medium, and high values for each of the three single objectives are used and the results compared. A comparison of the single and multiple objectives is also presented.

Chapter 11 concludes the thesis and presents future possible directions of research.

Chapter 2 – Additive model description and theory

2.1 Introduction

Chapters 2, 3 and 4 investigate the additive model, when there are few alleles and loci present, to determine when the model results are similar to infinitesimal model results. This investigation is achieved by comparing predicted genetic gains from the infinitesimal model with observed genetic gains from additive model simulations.

This chapter gives a description of the additive model, providing the model equation and the *n*-locus, *m*-allele structure along with the model assumptions. A worked example of the model is included, demonstrating the calculation of genetic and phenotypic value for an individual. This chapter also gives the analysis of the additive model, giving the formula for calculating the expected genetic gain for the model. An overview of the infinitesimal model is also provided. Chapter 3 describes the simulations undertaken and presents and discusses the results. Chapter 4 has a further investigation of changes in the underlying allele distribution.

2.2 Additive model description

A simple additive model was built to enable simulations of observed rates of genetic gain. Because further research with the additive model will be in the animal science field, the model organisms were designed to be diploid. The model also allowed for sexual reproduction, however it was not a requirement of the model for the phenotype to be sex-influenced. Generation of offspring, when a male and female individual were bred, was also required, with reproduction passing information from the two parents to the offspring. Since selection is mainly applied to the male population, the model allowed for differential rates of selection for males and females.

The model equation used for the simulations was:

$$Phenotypic\ value_i = \sum_j G_{ij} + E_i ,$$

where G_{ij} is the genetic effect of locus j for individual i, described in §2.2.1, and E_i is the environmental effect for individual i, described in §2.2.2.

2.2.1 Genetic effect

An *n*-locus, *m*-allele additive model mapped onto chromosome pairs was used. The initial assumptions were that there were no dominance, epistatic, linkage or mutation effects present in the model. The genetic value of an individual was defined as the sum of the genetic values of the *n* loci, and the genetic value of a locus was defined as the sum of the genetic values of the two alleles present (of the possible *m* alleles) on the chromosome pair at the locus. All loci were modelled with the same spread of *m* alleles, with equal intervals between the genetic values of each *m* allele at the same locus.

The genetic values of the *m* alleles were spread uniformly over the interval [*a*, *b*] in step sizes of $a/(m-1)$, with the first allele genetic value being *a* and the last *b*. In the initial population, alleles were randomly assigned to individuals with equal probability. This resulted in the initial population having allele values that followed a discrete uniform distribution.

As a consequence, the genetic values of a locus follow a discrete triangular distribution on the interval [2*a*, 2*b*]. The genetic values of the individuals in the initial population therefore take values in the interval [2*an*, 2*bn*], with the distribution being the sum of *n* discrete triangular distributions, which tends towards a normal distribution as *n* increases.

2.2.2 Environmental effect

Environment was considered in the model by way of a normally distributed environmental effect with mean zero and standard deviation, σ_E, added to each individual's genetic value. The value of σ_E was selected such that the heritability (Lush 1937), h^2, was fixed at a predetermined value in the first generation. The way in which the selection of σ_E was achieved is described in §2.5. The environment experienced by an individual was unaffected by the sex of the individual. No systematic environmental effects were modelled.

2.2.3 Model example

An example of how alleles are passed on during breeding is shown in Figure 2.1. This example is a two loci, six allele example on the interval [1.25, 2.5]. The interval of

[1.25, 2.5] is arbitrary, however it provides an interval of [2.5, 5] for each loci, and an interval of [5, 10] across the two loci. For each parent, one of two alleles present at each locus is passed on to the offspring. The allele that is passed on is randomly selected and is independent of alleles passed on from other loci. In Figure 2.1, parent 1 has passed on allele 2 from locus 1 and allele 5 from locus 2, and parent 2 has passed on allele 4 from locus 1 and allele 2 from locus 2. This has given an offspring with the alleles 2 and 4 at locus 1 and alleles 5 and 2 at locus 2.

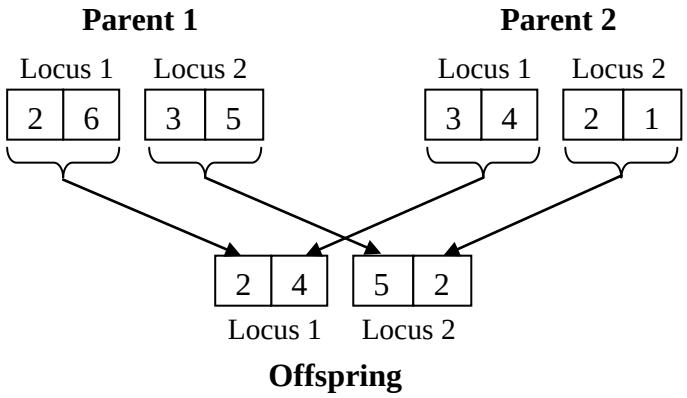

Figure 2.1: Diagram showing the passing on of alleles during breeding.

An example of the calculation of the genetic value and the phenotypic value of the individuals in Figure 2.1 is given in Table 2.1. For a 2-locus, 6-allele model, the possible allele values are 1.25, 1.5, 1.75, 2, 2.25, and 2.5 for alleles 1, 2, 3, …, 6 respectively. For parent 1 at locus 1, allele 2 has an allele value of 1.5 and allele 6 has an allele value of 2.5, giving a genetic value at locus 1 of 1.5 + 2.5 = 4; similarly for the other loci genetic values. The genetic value of parent 1 is the sum of the genetic values of the loci. Therefore the genetic value of parent 1 is 4.00 + 4.00 = 8.00; similarly for parent 2, 3.75 + 2.75 = 6.50, and the offspring, 3.05 + 3.75 = 7.25. The environmental value is a random number from a normal distribution with mean zero and variance appropriately selected for the required heritability. The phenotypic value of parent 1 is the sum of the genetic value and the environment which gives 8.00 + (-0.08) = 7.92; similarly for parent 2, 6.50 + 0.14 = 6.64, and the offspring, 7.25 + (-0.21) = 7.04.

Table 2.1: Calculation of phenotypic values (PV) from the sum of genetic values (GV) and a random environment effect (E) for a 2-locus, 6-allele model.

	Locus 1		**GV_1**	**Locus 2**		**GV_2**	**GV**	**E**	**PV**
	Allele 1	Allele 2		Allele 1	Allele 2				
Parent 1	2	6	4.00	3	5	4.00	8.00	-0.08	7.92
Parent 2	3	4	3.75	2	1	2.75	6.50	0.14	6.64
Offspring	2	4	3.50	5	2	3.75	7.25	-0.21	7.04

A systematic search was undertaken to investigate under what conditions, for an additive model, it would be reasonable to expect little or no difference between simulated genetic gain, from the additive model, and the predicted genetic gain, using infinitesimal theory.

2.3 Additive model theory

2.3.1 Distributions

The *m* alleles in the initial population are equally likely to occur, and are spread uniformly over the interval [*a*, *b*], and therefore have a discrete uniform distribution with a probability mass function, f_A, of

$$f_A(\alpha) = \begin{cases} \dfrac{1}{m} & \alpha \in \left\{ \dfrac{(b-a)i}{m-1} + a \mid i = 0,\ldots,m-1 \right\} \\ 0 & \textit{otherwise} \end{cases}.$$

For a diploid additive model, the genetic value of a locus is the sum of the genetic values of the two alleles present at the loci. For the model under investigation, this means that a locus has a probability mass function that is the sum of two discrete uniform distributions.

For this additive model, the overall genetic value is the sum of the *n* locus genetic values, thus the genetic value distribution is the sum of either *n* discrete triangular

distributions or 2*n* discrete uniform distributions. Therefore the genetic value has probability mass function, f_G, of

$$f_G(\gamma)=\begin{cases} 2n\left(\frac{1}{m}\right)^{2n}\sum_{i=0}^{floor\left(\frac{j}{m}\right)}\left((-1)^i\frac{(2n-1+j-im)!}{i!(2n-i)!(j-im)!}\right) & j=\frac{(\gamma-2na)(m-1)}{b-a}\in\{0,\ldots,2n(m-1)\} \\ 0 & \textit{otherwise} \end{cases}$$

where $floor\left(\frac{j}{m}\right)=\max\left\{k:k\le\frac{j}{m}\right\}$ and $k\in\mathbb{Z}$.

The genetic value probability mass function above was determined, for this work, through a combinatorics counting procedure on integers, which was then scaled to the genetic value range. Thus the variable *j* is the genetic value rescaled to integer values.

The overall genetic value has variance, σ_G^2, given by

$$\sigma_G^2=\frac{(b-a)^2\,n(m+1)}{6(m-1)}.$$

A derivation of this variance is given in §2.5.

Heritability, h^2, is defined as

$$h^2=\frac{\sigma_G^2}{\sigma_P^2}=\frac{\sigma_G^2}{\sigma_G^2+\sigma_E^2},$$

where σ_G^2 is the genetic variance, σ_P^2 is the phenotypic variance and σ_E^2 is the environmental variance. Since heritability has been predetermined, the above equation can be rearranged, and σ_G^2 substituted, to give the environmental variance, σ_E^2, of

$$\sigma_E^2=\frac{1-h^2}{h^2}\frac{(b-a)^2\,n(m+1)}{6(m-1)}.$$

The environmental effect has a normal distribution with a mean of zero, by definition, therefore the environmental effect probability density function, f_E, is

$$f_E(\varepsilon) = \frac{e^{\frac{-3\varepsilon^2 h^2 (m-1)}{(1-h^2)(b-a)^2 n(m+1)}}}{\sqrt{\pi \frac{1-h^2}{h^2} \frac{(b-a)^2 n(m+1)}{3(m-1)}}}.$$

The phenotypic value is the sum of the genetic value and the environmental value, which gives a phenotypic probability density function, f_P, of

$$\begin{aligned} f_P(\phi) &= \sum_{\gamma} f_G(\gamma) f_E(\phi - \gamma) \\ &= \sum_{j=0}^{2n(m-1)} f_G\left(\frac{j(b-a)}{m-1} + 2na\right) f_E\left(\pi - \frac{j(b-a)}{m-1} - 2na\right) \\ &= \frac{2n\left(\frac{1}{m}\right)^{2n}}{\sqrt{\pi \frac{1-h^2}{h^2} \frac{(b-a)^2 n(m+1)}{3(m-1)}}} \sum_{j=0}^{2n(m-1)} \left[\left(\sum_{i=0}^{floor\left(\frac{j}{m}\right)} (-1)^i \frac{(2n-1+j-im)!}{i!(2n-i)!(j-im)!} \right) e^{\frac{-3\left(\phi - \frac{j(b-a)}{m-1} - 2na\right)^2 h^2 (m-1)}{(1-h^2)(b-a)^2 n(m+1)}} \right] \end{aligned}$$

where $floor\left(\frac{j}{m}\right) = \max\left\{k : k \le \frac{j}{m}\right\}$ and $k \in \square$.

This phenotypic probability density function, f_P, is summed over all possible genetic values, γ, and the genetic value is rescaled to the integer variable j.

The Phenotypic value has a variance, σ_P^2, of

$$\sigma_P^2 = \sigma_G^2 + \sigma_E^2 = \frac{(b-a)^2 n(m+1)}{6h^2(m-1)},$$

since the genetic value and environmental value are independent.

2.3.2 Genetic gain

Genetic gain is the observed change in the mean of a population from one generation to the next, whilst predicted genetic gain is the prediction of the expected genetic gain for one generation.

Predicted genetic gain, ΔG, is the difference between the average of the expected genetic values for the male and female population means and the expected genetic value for the whole population, and is given by

$$\Delta G = \frac{E\left[G|_{P=\mu_m}\right] + E\left[G|_{P=\mu_f}\right]}{2} - E\left[G|_{P=\mu_P}\right],$$

where μ_m is the mean phenotypic value of the male population, μ_f is the mean phenotypic value of the female population, and μ_p is the mean phenotypic value of the whole population.

For the simulations carried out in Chapter 3, the male population is selected from the population to form a distinct genotypic population, while the female population is taken to be identical to the entire population, giving the female population

$$E\left[G|_{P=\mu_f}\right] = E\left[G|_{P=\mu_P}\right].$$

Hence when using the analysis of the additive model, the predicted genetic gain, ΔG_A, is

$$\Delta G_A = \frac{E\left[G|_{P=\mu_m}\right] - E\left[G|_{P=\mu_P}\right]}{2}.$$

For the infinitesimal model, ΔG_I, this simplifies to,

$$\Delta G_I = \overline{i}h^2\sigma_p,$$

as a result of the normality and linear parent-offspring regression assumptions, where

$$\bar{i} = \frac{\mu_m - \mu_p}{\sigma_p}$$

is the selection intensity for the breeding program.

Since

$$\Delta G_I = \mu_o - \mu_p$$

where μ_o is the mean of the offspring population, an alternative calculation for heritability under the infinitesimal model is

$$h^2 = \frac{\mu_o - \mu_p}{\mu_m - \mu_p}$$

For the additive model, the individual expected genetic values must be calculated for the mean phenotypic values of the male population, the female population, and the whole population.

2.3.3 Expected genetic value

Under the additive model, the theoretical expected genetic value for a given phenotypic value π, $E\left[G|_{P=\pi}\right]$, is given by

$$E\left[G\left|_{P=\pi}\right.\right]=\frac{\int\gamma f_G(\gamma)\,f_E(\pi-\gamma)\,d\gamma}{f_P(\pi)}$$

$$=\frac{\displaystyle\sum_{j=0}^{2n(m-1)}\left[\left(\frac{j(b-a)}{m-1}+2na\right)\left(\sum_{i=0}^{floor\left(\frac{j}{m}\right)}\left((-1)^i\frac{(2n-1+j-im)!}{i!(2n-i)!(j-im)!}\right)\right)e^{\frac{-3\left(\pi-\frac{j(b-a)}{m-1}-2na\right)^2h^2(m-1)}{(1-h^2)(b-a)^2n(m+1)}}\right]}{\displaystyle\sum_{j=0}^{2n(m-1)}\left[\left(\sum_{i=0}^{floor\left(\frac{j}{m}\right)}(-1)^i\frac{(2n-1+j-im)!}{i!(2n-i)!(j-im)!}\right)e^{\frac{-3\left(\pi-\frac{j(b-a)}{m-1}-2na\right)^2h^2(m-1)}{(1-h^2)(b-a)^2n(m+1)}}\right]}$$

where $floor\left(\frac{j}{m}\right)=\max\left\{k:k\le\frac{j}{m}\right\}$ and $k\in\square$.

2.4 Infinitesimal theory overview

The joint assumptions of normality and linear parent-offspring regressions underlie most simple models of selection response, as shown in Figure 2.2 below. With these assumptions it is possible to predict the single-generation response to selection pressure when truncated selection is applied.

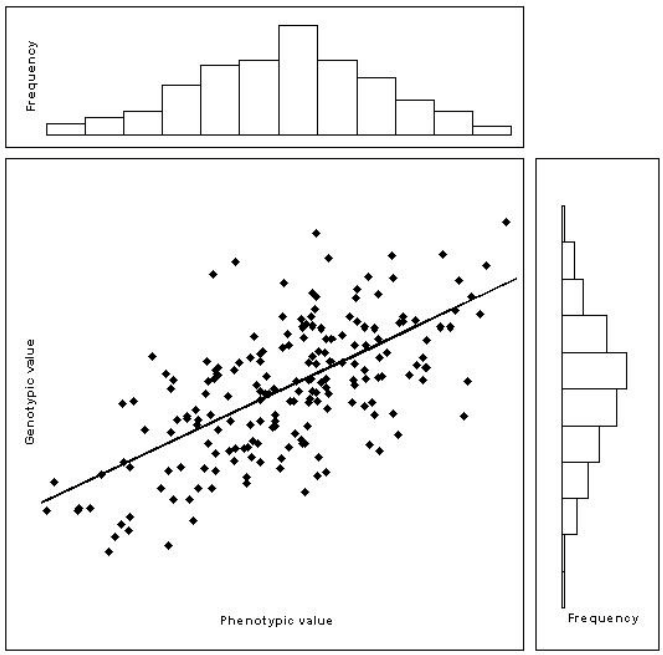

Figure 2.2: Relationship between individual genotypic and phenotypic values for the infinitesimal model. Both the genotypic and phenotypic distributions are normally distributed and there is a linear relationship.

The infinitesimal model assumes a very large, effectively infinite, number of unlinked and non-epistatic loci, each with infinitesimal effect. This satisfies the assumptions of normality and linearity. As a result, most models of selection response are based on the assumption that the infinitesimal model sufficiently describes the situation. The infinitesimal model is not taken as an exact description of biological reality, but it does represent one extreme of assumptions about the underlying loci, allowing the effects of allele frequency changes to be ignored.

The infinitesimal model is a very robust model as significant changes in the mean, with little change in the other moments, can be achieved by very small allele frequency changes over a large number of loci. Under the infinitesimal model, it is assumed that the difference between the selected animals and the complete population is significant, but since the genotypic values of the animals are the sum of an effectively infinite number of loci, the expected difference at each individual locus is very small. Thus the expected difference in allele frequencies at each locus is very small also, resulting in the expected change in allele frequencies over a generation of selective breeding being even smaller. Thus when summed over a large number of loci, these very small allele frequency changes allow for significant changes in the mean with little change in the other moments.

The infinitesimal prediction of single-generation response (Lush 1937), which is predicted genetic gain, ΔG_p, is given by

$$\Delta G_p = \bar{i} h^2 \sigma_p ,$$

where $\bar{i}$ is the selection index for the breeding program, h^2 is the heritability of the trait and also the slope of the line in Figure 2.2 as per below, and σ_p is the phenotypic standard deviation of the trait.

The slope of the line in Figure 2.2 is

$$\frac{Cov(G,P)}{\text{var}(P)} = \frac{\text{cov}(G,G+E)}{\text{var}(P)} = \frac{\text{cov}(G,G)+\text{cov}(G,E)}{\text{var}(P)} = \frac{\text{var}(G)}{\text{var}(P)} = h^2$$

2.5 Variance components

The infinitesimal model assumes that the trait subject to selection is normally distributed and there is a linear parent-offspring regression underlying the selection response. With these assumptions the single-generation response can be predicted from the knowledge of the phenotypic variance and selection intensity (Falconer and McKay, 1996) of the breeding program. The variances required to predict the single-generation response are described below.

The underlying distribution of the genetic component of the phenotypic value is a discrete uniform distribution (Beyer 1987). The discrete uniform density function on $n = 1, \ldots, N$ is

$$P(n) = \frac{1}{N}.$$

The variance, σ_N^2, of the discrete uniform distribution is

$$\sigma_N^2 = \frac{1}{12}(N-1)(N+1).$$

The allele genetic values in the initial population follow a discrete uniform distribution on the interval [a, b] with a step size of $(a–b)/(m–1)$. This domain is a linear transformation of the domain of the discrete uniform distribution above, and therefore the allele values have a variance, σ_{Al}^2, of

$$\sigma_{Al}^2 = \frac{(a-b)^2}{(m-1)^2}\frac{1}{12}(m-1)(m+1) = \frac{(a-b)^2(m+1)}{12(m-1)}.$$

Since the genetic value distribution at a locus is the sum of two independent allele value distributions, the genetic value distribution has a variance, σ_L^2, of

$$\sigma_L^2 = 2\sigma_{Al}^2 = \frac{(a-b)^2(m+1)}{6(m-1)},$$

in the initial population.

The genetic value of an individual is defined as the sum of the genetic values of the n loci. This gives the genetic value a variance, σ_G^2, in the initial population of

$$\sigma_G^2 = n\sigma_L^2 = \frac{(a-b)^2 n(m+1)}{6(m-1)}.$$

A definition of heritability, h^2, (Lush 1937) is

$$h^2 = \frac{\sigma_G^2}{\sigma_P^2},$$

where σ_P^2 is the phenotypic variance (Lush 1937). This results in the variance in phenotypic values, σ_P^2, of individuals in the initial generation being

$$\sigma_P^2 = \frac{\sigma_G^2}{h^2} = \frac{(a-b)^2 n(m+1)}{6h^2(m-1)}.$$

Since, by the definition of the model, the phenotypic value is equal to the sum of the genetic value and the environmental effect, we have

$$\sigma_P^2 = \sigma_G^2 + \sigma_E^2,$$

where σ_E^2 is the environmental variance. In this simplified model the genetic and environmental effects are independent and hence the covariance is zero.

Combining the above equations gives

$$\sigma_E^2 = \sigma_P^2 - \sigma_G^2 = \frac{1-h^2}{h^2}\frac{(a-b)^2 n(m+1)}{6(m-1)},$$

which is the environmental variation modelled.

Chapter 3 – Simulation method and results

3.1 Introduction

This chapter describes the search space over which the simulation were performed, the population simulated and the method of simulation for the model introduced in Chapter 1. It then gives a description of the parameters used to evaluate the model and compares the results to the infinitesimal model predictions. The results of the simulation are then presented and followed by a discussion.

3.2 Search space

The additive model was investigated for number of loci n = 1, 2, 3, 4, 5, 10, 15, 20, 30, 40 and 50, number of alleles m = 2, 3, 4, 5, 6, 20, 40 and 100, and heritability h^2 = 0.1, 0.3, 0.4, 0.6 and 0.8. This resulted in 440 different combinations for investigation. The interval [a, b] was arbitrarily chosen to be [1.25, 2.5] to give each locus an interval of [2.5, 5].

3.3 Simulated population and mating strategy

Populations of 500 individuals were simulated for one breeding cycle. One breeding cycle was simulated, as multiple breeding cycles can result in movements in allele frequency and hence the variances of genetic and phenotypic values change. These changing variances then result in varying heritability. This on-flowing variability over multiple breeding cycles is often referred to as the Bulmer effect (Bulmer 1971).

The individuals in the initial population were randomly generated such that it was equally likely that each of the m alleles at the n loci would appear in the population. A single mating strategy, that of truncated phenotypic selection, was simulated. The top 1%, 5 individuals, was selected as the males of the next generation. Every individual in the population, regardless of sex, was then mated with one of the five males, so each male is mated with 100 individuals. This simplification of sex results in a slight bias in the results as the males are present in the system as females when they are not the current male being bred. However there was no selfing. This was thus an asexual, closed population.

3.4 Method

One thousand simulation runs were performed for each simulation setup, and 95% confidence intervals on the evaluation parameters Λ, the standardised difference between the observed genetic gain from the simulations and the prediction from the infinitesimal model, and δ, the standardised difference between the observed genetic gain from the simulations and the expected genetic gain from additive model analysis, as defined in §3.6, were calculated. Over the 1000 simulation runs, the observed genetic gains, ΔG_O, were approximately normally distributed, and hence Λ and δ were approximately normally distributed. The additive model was investigated with the alleles on the interval [1.25, 2.5].

A program was written to calculate the values of δ as a special data structure was needed to cope with values in excess of 100 factorial.

3.5 Simulation software and procedure

The simulation was programmed in Visual C++, with a Mersenne Twister (Matsumoto and Nishimura 1998) random number generator. The data files generated contained the mean genetic gain observed over the 1000 replicates and the standard deviation of the observed genetic gains. The predicted genetic gain and the evaluation parameter Λ were calculated using a Microsoft Excel spreadsheet.

3.6 Evaluation parameters

A commonly used measure of the performance of a breeding program is genetic gain. Genetic gain is defined as the change in the mean phenotypic value from one generation to the next.

The infinitesimal prediction of single-generation response (Lush 1937), predicted genetic gain, ΔG_P, is given by

$$\Delta G_P = \bar{i} h^2 \sigma_p .$$

This gives a predicted genetic gain of

$$\Delta G_P = \bar{i} h^2 \sqrt{\frac{1.25^2 n(m+1)}{6h^2(m-1)}} = 1.25 \bar{i} h \sqrt{\frac{n(m+1)}{6(m-1)}} ,$$

where $\bar{i}$ is the standardised selection differential. Standardised selection differential is the difference between the mean of the population and the mean of the selected population, in standard deviation units. The value of $\bar{i}$ is 1.305 for the selection of the top 5 males from 500, with no selection pressure applied to the females, and allowing for small numbers selected.

The observed genetic gain from the simulations, ΔG_O, is the difference between the mean observed phenotypic values after one breeding cycle, $\bar{X}_{P_2}$, and the mean observed phenotypic values of the initial population, $\bar{X}_{P_1}$, giving

$$\Delta G_O = \bar{X}_{P_2} - \bar{X}_{P_1}.$$

The first evaluation parameter, Λ, of interest is the standardised difference between the observed and predicted genetic gains

$$\Lambda = \frac{\Delta G_O - \Delta G_P}{\sigma_P} = \frac{\bar{X}_{P_2} - \bar{X}_{P_1}}{\sigma_P} - \bar{i}h^2.$$

The second evaluation parameter, δ, is defined as the standardised difference between the observed genetic gain, ΔG_O, from the simulation, and the predicted genetic gain of the additive model analysis, ΔG_A. Therefore

$$\delta = \frac{\Delta G_O - \Delta G_A}{\sigma_P} = \frac{\bar{X}_{P_2} - \bar{X}_{P_1}}{\sigma_P} - \frac{E\left[G\,|_{P=\mu_m}\right]}{2\sigma_P} + \frac{E\left[G\,|_{P=\mu_p}\right]}{2\sigma_P}.$$

Assuming normal distributions for Λ and δ, confidence intervals of 95% were calculated from the 1000 simulations for each simulation setup. Anderson-Darling normality tests were carried out on a significant selection of the setups to confirm the validity of the normality assumption. For the majority of the setups investigated, the evaluation parameters were normally distributed. The setups that failed the normality test had high heritability and one or two loci.

Parameters Λ and δ are zero when there is no divergence between the simulation and theoretical predictions. In this case the confidence interval would contain zero. Similarly, Λ and δ are non-zero when a difference exists, and hence the confidence

interval would not contain zero. Since the simulated observations are the only source of variation in Λ and δ for a particular setup, the placement of the confidence interval to the left of zero, around zero, or the right of zero, depends on the magnitude of the simulation results relative to the theoretical predictions.

3.7 Results

The following graphs, in Figure 3.1, display the standardised difference between the simulation and the predicted genetic gain from the additive model analysis. The evaluation parameter δ is displayed on the y-axis. The heritabilities of 0.3 and 0.6 are shown on separate graphs. The multiple plots on the graphs allow for various numbers of alleles at each locus.

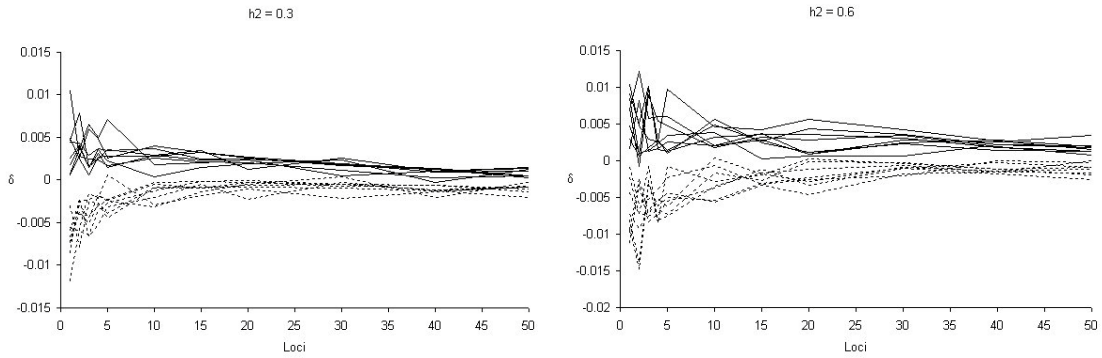

Figure 3.1: Simulation results showing 95% confidence intervals for δ with the number of loci, *n*, on the x-axis. Various numbers of alleles have been plotted on each graph. The solid lines are the upper bounds on the confidence interval and the dotted lines are the lower bounds.

As can be seen the confidence intervals contain zero. This indicates that the simulation is producing results in line with the additive model theory.

The two graphs, in Figure 3.2 below, display the standardised difference between the simulation and the infinitesimal predictions. The evaluation parameter Λ is displayed on the y-axis. The heritabilities of 0.3 and 0.6 are shown on separate graphs. The multiple plots on the graphs allow for various numbers of alleles at each locus.

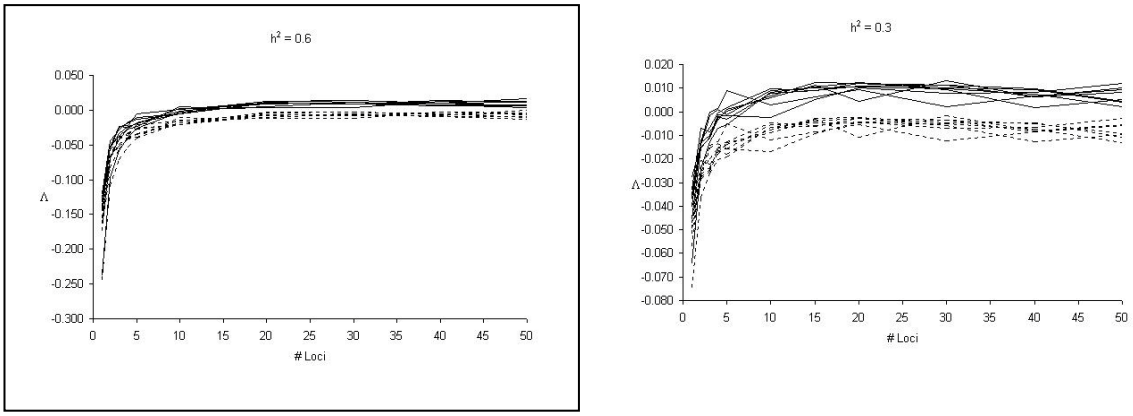

Figure 3.2: Simulation results showing 95% confidence intervals for Λ with the number of loci, *n*, on the x-axis. The solid lines are the upper bounds on the confidence interval and the dotted lines are the lower bounds.

When truncated selection pressure was applied to the male population, the results had a consistent trend, as shown in Figure 3.3. When the number of loci is low, the observed confidence interval for Λ resulting from the simulation did not contain zero. Hence for a hypothesis test at the 5% significance level there was a significant difference between the observed genetic gain, ΔG_O, and the infinitesimal theory predicted genetic gain, ΔG_P, with the infinitesimal model overestimating the observed genetic gain.

However as the number of loci increases the observed genetic gain, ΔG_O, from the simulation, produced a confidence interval for Λ that contained zero. This means that hypothesis test at the 5% significance level there was no significant difference between the infinitesimal theory predicted genetic gain, ΔG_P, and the observed genetic gain, ΔG_O.

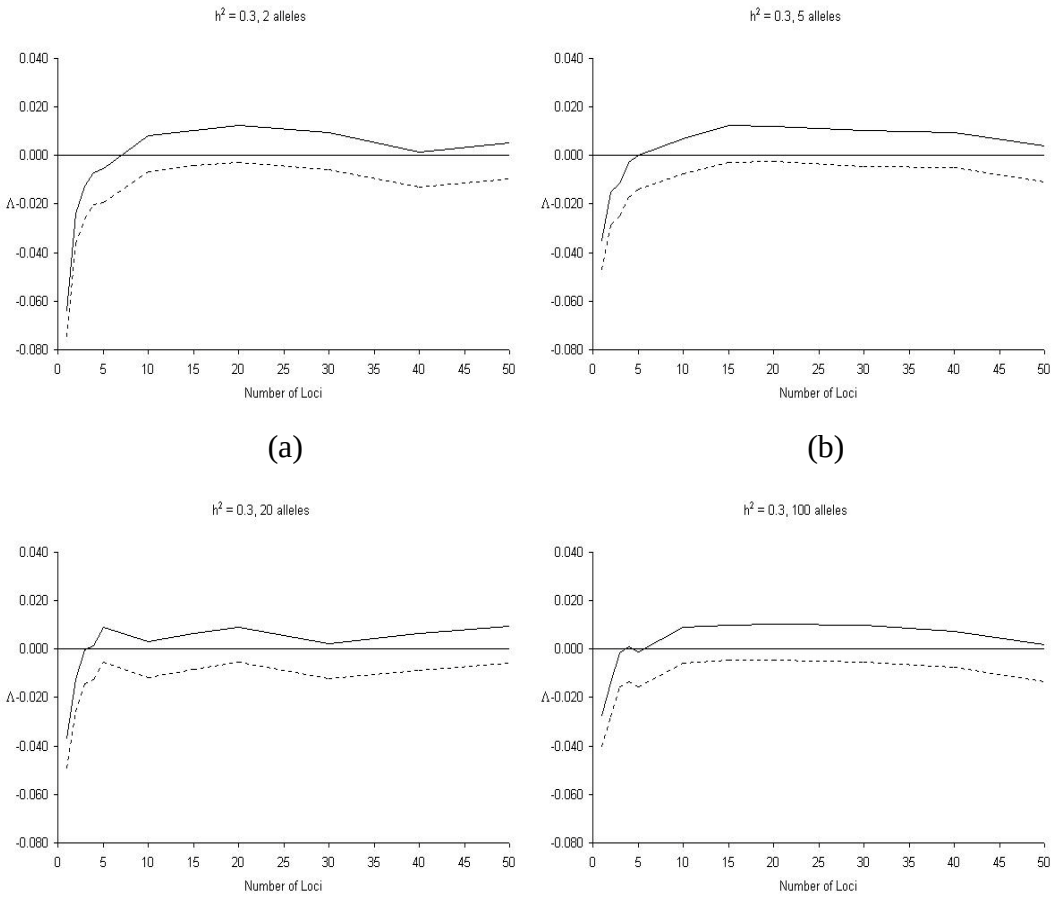

Figure 3.3: Simulation 95% confidence interval for Λ.

h^2 = 0.3 and varying number of loci and alleles are graphed. A result for each allele number is displayed on a separate graph. The solid lines represent the upper bounds, and the dotted lines represent the lower bounds on the confidence intervals.

A subset of results for heritability of 0.3 and 100 alleles is summarised in Table 3.1. These results show the trend that was repeatedly observed as the number of loci increased from 1 to 50, columns 1 to 4 in the table. As the number of loci increases, the genetic and phenotypic standard deviations increase due to the increasing number of loci contributing to the genetic value, however the contribution of each locus to the variances is constant, resulting in the influence of an individual loci decreasing as the number of loci increases. The values of Λ are not influenced by this increasing variation as they have been standardised by the standard deviation. The environmental standard deviation also increases with the number of loci, as heritability was set to 0.3. The infinitesimal genetic gain prediction increases, as the number of loci increases as a result of the increasing phenotypic standard deviation. The mean and standard deviation

of the genetic gains are from 1000 repetitions of the simulation. The resultant mean and standard deviations for Λ are also shown along with the 95% confidence intervals. The values of Λ over the 1000 repetitions were approximately normally distributed. This subset of results clearly shows that for low numbers of loci the confidence intervals are to the left of zero. As the number of loci increases, the confidence intervals move to contain zero.

Table 3.1: Summary table of a subset of simulation results, for heritability 0.3 and 100 alleles, of infinitesimal predictions and 95% confidence intervals for Λ.

Number of loci, n	**1**	**5**	**10**	**50**
Genetic value S.D., σ_G	0.515	1.153	1.630	3.645
Environmental effect S.D., σ_E	0.787	1.761	2.490	5.567
Phenotypic value S.D., σ_p	0.941	2.104	2.976	6.654
Predicted genetic gain, ΔG_p	0.368	0.824	1.165	2.605
Mean in silico genetic gain	0.337	0.806	1.170	2.568
S.D. of in silico genetic gain	0.096	0.249	0.353	0.819
Mean Λ	-0.034	-0.008	0.002	-0.006
S.D. of Λ	0.010	0.029	0.042	0.101
95% confidence interval for Λ	(-0.040, -0.027)	(-0.016, -0.001)	(-0.006, 0.009)	(-0.013, 0.002)

The simulation results for heritability of 0.3, as displayed in Figure 3.3, show that, in general, when the number of loci *n* was above 10 the confidence intervals contained zero, and hence there was no significant difference between the infinitesimal prediction of genetic gain and the observed genetic gain. When the number of loci fell below this level there was a significant overestimation of the genetic gain by the infinitesimal model, relative to that observed in the simulation of the additive model.

A similar trend was observed in the results for the other heritability levels investigated, as shown in Figure 3.4. For a heritability of 0.1 the results change from overestimation by the infinitesimal model to accurate prediction of the observed gain at about 5 loci. When the heritability was increased to 0.4, the infinitesimal model accurately predicted the observed genetic gain from about 10 loci, and for heritability of 0.8, from about 15

to 20 loci. This indicates that the magnitude of the heritability had some influence on the accuracy of the infinitesimal predictions of ΔG_P. This is caused by the decreasing environmental noise, and hence decreasing normality of the trait, as heritability increases.

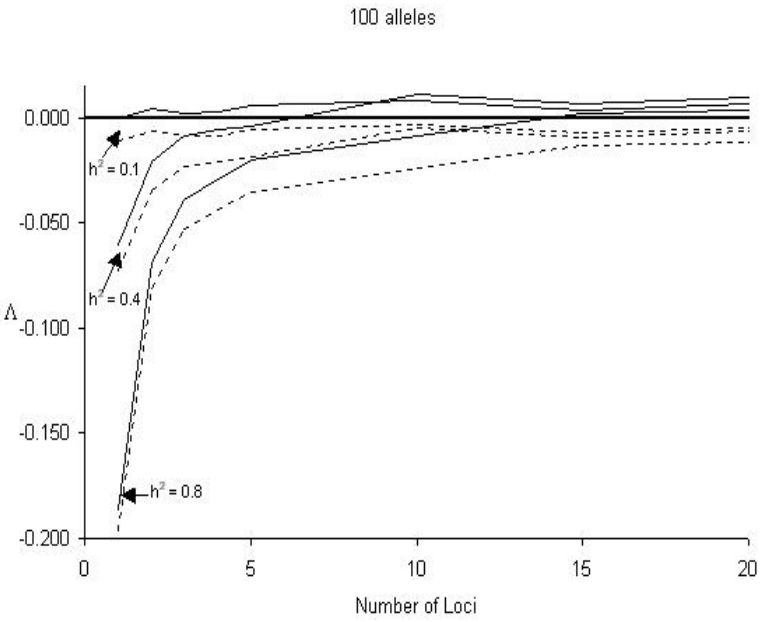

Figure 3.4: Simulation results showing the 95% confidence intervals for Λ. The results are for 100 alleles and 20 or fewer loci. The solid lines represent the upper bounds, and the dotted lines represent the lower bounds on the confidence intervals.

The influence of heritability on the observations is shown in Figure 3.4. This figure shows that the difference between the simulation results and the infinitesimal predictions is exaggerated as the value of heritability increases.

When the heritability is high, with only a few loci, the confidence intervals do not overlap for the various allele combinations and they become significantly different from each other, as delineated in Figure 3.5. Thus if there are few loci and few alleles, both the number of loci and the number of alleles influence the accuracy of the infinitesimal predictions.

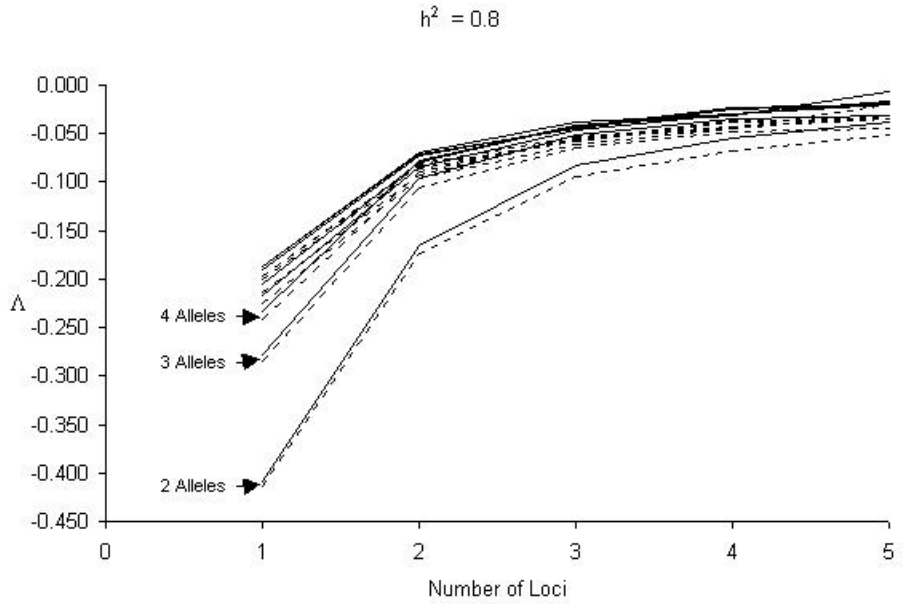

Figure 3.5: Simulation results showing the 95% confidence intervals for Λ.

The results are for a heritability h^2 of 0.8. The solid lines represent the upper bounds, and the dotted lines represent the lower bounds on the confidence intervals.

3.8 Discussion

As can be seen in Figure 3.1 when the additive model analysis is used for predicting the genetic gain, there is no significant over or under estimation of the genetic gain generated by the simulation. This indicates that the simulation is producing results in line with the additive model theory.

This discussion first addresses the situation where only a few loci control the trait. As can be seen in Figure 3.2, when the infinitesimal model is used for predicting genetic gain, there is a clear over-estimation of the genetic gain generated by the simulation. This over-estimation becomes greater as the number of loci reduces. When the simulated genetic gain is compared to the predicted genetic gain from the model analysis, as in Figure 3.1, there is no discrepancy between the observed and predicted genetic gain.

This over-estimation from Figure 3.2 arises from a violated assumption of the infinitesimal model, namely that there are an infinite number of loci and alleles. With this assumption, the relationship between genetic gain and the selected population mean is linear, with heritability being constant and equal to the slope of the line. Without this

infinitesimal assumption, and using the additive model analysis, this relationship becomes non-linear, and heritability becomes non-constant as shown in Figure 3.6 below. This is intuitively clear for this model, as the minimum genetic value that can occur is $2na$ and the maximum genetic value is $2nb$. Hence there are upper and lower bounds on the genetic gain that can occur, and genetic gain is asymptotic to these bounds as the selected population mean becomes more extreme. The infinitesimal model does not allow for the existence of these bounds.

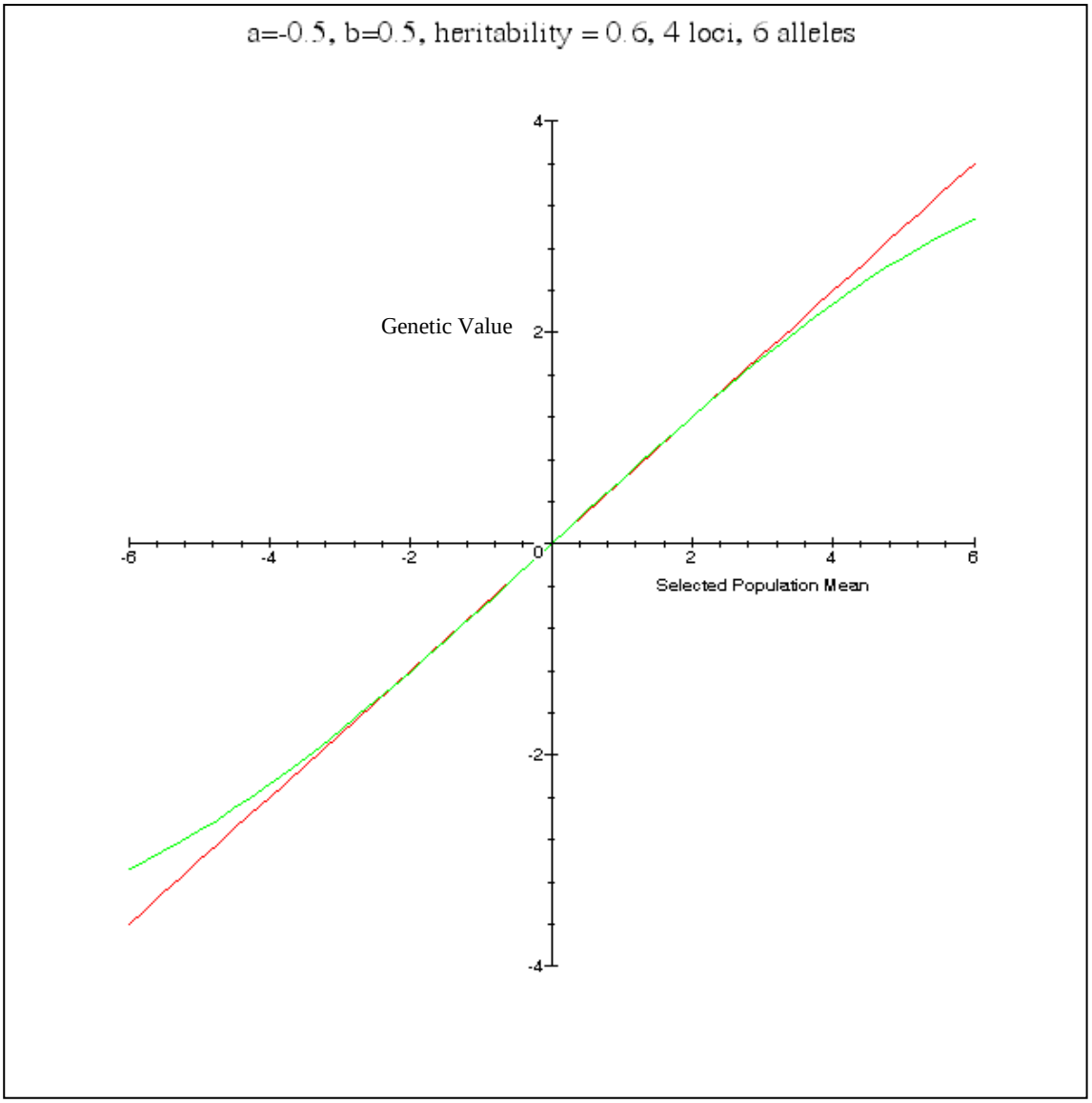

Figure 3.6: Graph of expected genetic value given the phenotypic value for the additive model with heritability of 0.6, 4 loci and 6 alleles at each loci, with the allele distribution being on the range [-0.5, 0.5]. The red straight line is the result of the

infinitesimal model assumptions and the green curved line is the result of the additive model analysis.

It is clear from Figure 3.2 that bounds on genetic gain start to influence the prediction of the infinitesimal model when there are few loci and when selection pressure is high, as was the case in this simulation. It should be noted that this is an extreme situation.

An *n*-locus, *m*-allele additive model with no dominance, epistatic, linkage or mutation effects and equal frequency of each allele is unlikely to accurately represent a biological trait. Such a model channels all of the biology of gene expression through to the final observed trait into a predetermined value and a random environmental noise value, with no interaction between the two. There exist equations representing the flow of substrates in the biological pathways of cells that could be used to give a more accurate representation. One of the simplest pathway equations for the conversion of one substrate to another is the Michaelis-Menten equation (Michaelis and Menten 1913)

$$\textit{Substrate conversion velocity} = \frac{V_{\max}\left|S\right|}{K+\left|S\right|},$$

where V_{max} is the maximum velocity at which the substrate may be converted, K is the concentration of the substrate at which the substrate conversion velocity is equal to half of V_{max} and $\left|S\right|$ is the substrate concentration.

Basing the simulation model on this equation, or similar, would be more biologically correct. This could be achieved by making V_{max} and/or K genetically controlled, but because of the interaction between genes and substrate levels, the definition of the genetic value becomes ambiguous. As well, the calculations of theoretical genetic values become difficult, and for some initial substrate distributions, impossible (Sherriff unpublished). As exact and unambiguous genetic values were wanted for this investigation, the *n*-loci, *m*-allele additive model was used.

The main assumptions of the infinitesimal model are that the underlying distributions are normally distributed and that there is a linear parent-offspring regression. When there are 2-alleles and *n*-loci, the genetic value distribution is the sum of 2*n* Bernoulli distributions. The Central Limit Theorem (Feller 1945) gives the result that the sum of 2*n* Bernoulli distributions will converge to a normal distribution as *n* increases. The

situation when there are more than 2 alleles is a generalisation of the Bernoulli distribution. From the results presented in this chapter, the genetic value distribution has converged sufficiently to the normal distribution for the infinitesimal model to give accurate predictions when there are more than about 15 loci in the model, as expected from the central limit theorem. As can be seen in Figure 3.2 the number of alleles has little influence on this with allele numbers ranging from 2 to 100 all following a similar pattern.

When the number of loci is small the upper and lower tails of the genetic value distribution diverge from that of the normal distribution. Under the additional condition of low environmental variation, the phenotypic distribution also diverges from that of the normal distribution (Sherriff unpublished). This situation violates one of the primary assumptions of the infinitesimal model. When the genetic or phenotypic distributions are non-normal in the tails, the relationship between the genetic and phenotypic distributions is non-linear in the tail region. This results in violation of the primary classical assumption of a linear parent-offspring regression. The non-normality of the upper tails for small numbers of loci plays a significant role in the observed discrepancy between the observations and the infinitesimal predictions of genetic gain.

When a discrepancy exists between the observed and predicted results, the 95% confidence intervals for Λ's are negative, indicating that the infinitesimal model predictions are over-estimating the expected increase in mean phenotypic value of the population. The reason for over-estimation, as opposed to under-estimation, can be explained by the choice of distribution used for the allele value distribution. With a discrete uniform distribution for the allele value distribution, the genetic value distribution is a sum of discrete triangular distributions. This sum of discrete triangular distributions has higher probabilities in the centre of the distribution and the tails are truncated when compared to a normal distribution. As can be seen in Figure 3.6 this causes the heritability at the extremes to be less than that predicted by the infinitesimal model. This results in an over-prediction of heritability by the infinitesimal model when extreme selection is applied, which overall results in an over prediction of genetic gain.

Chapter 4 – Additive model alternative allele distributions

4.1 Alternative allele distributions

4.1.1 Introduction

This chapter expands on the simulations in Chapter 3 by looking at the affects of varying the underlying allele distribution of the additive model. Firstly Chapter 4 describes the allele structure and the parameters for which the allele structure is simulated. It then goes on to present the results of the simulations and finally discusses these results.

4.1.2 Model description

The model investigated here is an *n*-loci, 5-allele additive diploid model with a Mendelian system of genetic interaction. Five alleles were chosen for this section for ease of calculation of probability values for the distribution. The allele values are determined by parameters α and θ, where the angle θ controls the frequency of the alleles in the population and therefore the general shape of the distribution, and α controls the skew of the allele value distribution. The number of alleles at each locus is 5, with the allele value distribution has a mean of zero and a variance of 0.5. The value of a locus is the sum of the values of the two alleles present at the locus, and the genetic value is the sum of the *n* loci values. The allele value distribution is shown in Figure 4.1 below.

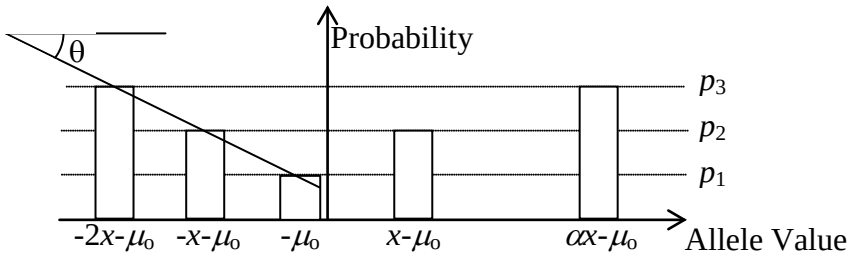

Figure 4.1: Allele value distribution.

There are six parameters to this model, α, θ, p_1, p_2, p_3, and x. The two parameters α and θ, which control the skewness and shape of the distribution are input parameters

and are used to determine the values of the other four parameters. The parameter x controls the distance between the allele values, p_3 is the probability of allele one and five occurring, p_2 is the probability of allele two and four occurring, and p_1 is the probability of allele three occurring.

The parameters p_1, p_2, p_3, x are determined so that the allele value distribution has a mean of zero and variance of 0.5. A mean of zero and variance of 0.5 was desired to give each locus, which is the sum of two allele value distributions, a zero mean and variance of one.

Since the allele value distribution has a mean of zero,

$$\mu_o = (\alpha - 2)\, x p_3$$

Therefore, for the allele value distribution to have a variance of 0.5,

$$4x^2 p_3 + 2x^2 p_2 + \alpha^2 x^2 p_3 - (\alpha - 2)^2 x^2 p_3^2 = 0.5$$

There are also the additional constraints of

$$p_3 = 2p_2 - p_1$$

$$\tan\theta = \frac{p_3 - p_1}{2x}$$

$$2p_3 + 2p_2 + p_1 = 1$$

When these equations are solved simultaneously, x is the roots of

$$(32\alpha^2 - 128\alpha + 128)\tan^2\theta\, x^4 + (-24\alpha^2 - 64\alpha - 76)\tan\theta\, x^3 + (-8\alpha^2 - 8\alpha - 52)x^2 + 25 = 0$$

and

$$p_2 = \frac{1 - x\tan\theta}{5}$$

$$p_3 = 1 - 4p_2$$

$$p_1 = 2p_2 - p_3$$

As in §2.2 and §2.3 there is a normally distributed random environmental effect added to the genetic value to produce the phenotypic value of an individual. The variance of the environmental effect is selected so that heritability, h^2, (Lush 1937) is fixed at a predetermined value of 0.3. There are no dominance, epistatic, mutation or sex effects modelled.

Linkage is explicitly modelled, unlike §2.2 or §2.3, with parameter β. Linkage is modelled by controlling, with parameter β, the probability of there being a crossover between loci. A crossover is when from one locus to the next there is a change from selecting the first/(second) allele present at the first locus to selecting the second/(first) allele present at the second locus. A crossover probability of zero gives complete linkage of the loci. In this situation a second generation offspring has either half of the grandparents' genetics or none of the grandparents' genetics. Thus a second generation offspring has genetic information from only two of the four grandparents. A crossover probability of 0.5 gives no linkage between loci. This means that a second generation offspring will, on average, obtain a quarter of its genetic information from each of its four grandparents.

The initial population is randomly produced with each allele being equally likely to occur. The top 1% of the population is selected as male and is bred randomly with the whole population such that each male is bred with the same number of females. Discrete generations are modelled so that the complete population is replaced when breeding occurs. Each randomly produced population is bred for one generation and the observed genetic gain is recorded. A population size of 500 is used and 1000 replications are made. The number of loci ranged from 1 to 15 in steps of 2.

The same evaluation parameter, Λ, as in §2.2 and §2.3 is used for comparison purposes, to give a standardised difference between observed genetic gain and predicted genetic gain using the infinitesimal model.

4.1.3 Method

The initial investigations were made with $\alpha=2$, to give symmetric allele value distributions. The angle, θ, was then varied as follows:

i) $\theta = -10$, producing a close model to a normal distribution, with a low frequency of extreme allele values in the population.
ii) $\theta = 0$, giving a uniform distribution model with equal allele frequencies in the population.
iii) $\theta = 10$, producing a very non-normal distribution model with high frequencies of extreme allele values in the population.

Setting $\theta = 0$ produces an equivalent model to that in §2.2 and §2.3 above, with 5 alleles. This gives an *m*-loci, 5-allele, additive model with equal allele frequencies.

Investigations were also made with α increasing from 2 to 4. Here, the frequencies of the alleles were non-symmetric and right skewed.

The linkage parameter, β, ranges from zero, representing complete linkage of the loci, to 0.5, representing no linkage.

4.1.4 Results

Figure 4.2 below shows the simulated difference between the observed and predicted genetic gain for varying θ when there is a symmetric distribution and no linkage.

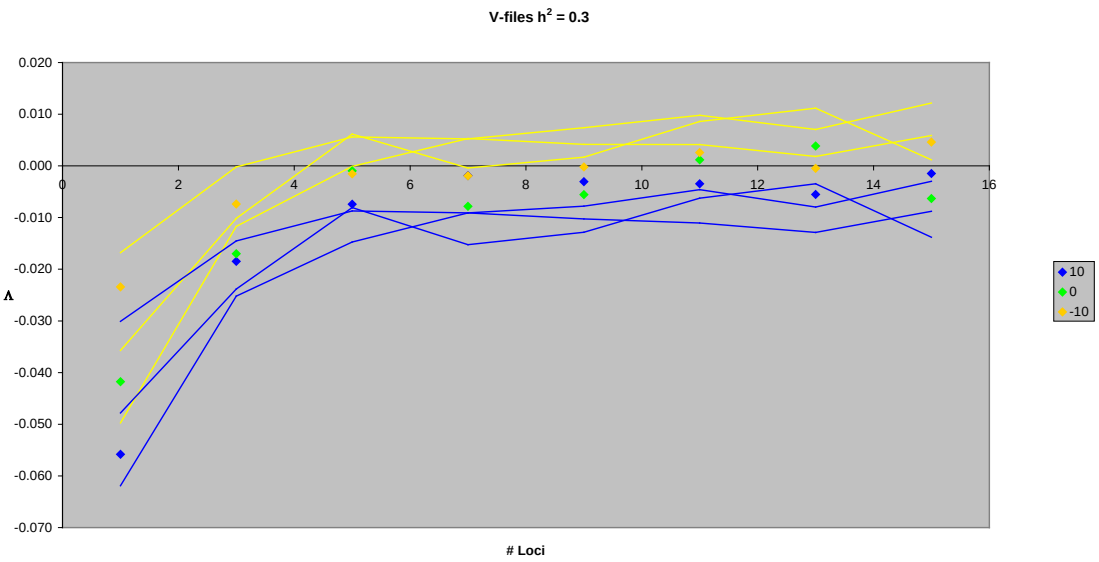

Figure 4.2: Effects of varying allele frequencies

varying θ, α=2, β=0.5, 95% confidence intervals, yellow lines are the upper value of the interval, blue lines are the lower value of the interval.

As can been seen, at low loci numbers, the confidence intervals for the different values of θ are distinct. As the number of loci increases, the confidence intervals merge. Therefore the allele distribution shape has an effect on the accuracy of the infinitesimal prediction when there are a few loci.

Figure 4.2 also shows that at low loci levels, low values of θ are most accurate and high values of θ are least accurate. This is as expected, since the infinitesimal prediction is based on the normal distribution. When the angle is -10^{o}, and hence the allele value distribution most resembles a normal distribution, the discrepancy between observed and predicted genetic gain is at its smallest. When the angle is $+10^{o}$ the allele value distribution is least like a normal distribution and the discrepancy is at its biggest.

Figure 4.3 and Figure 4.4 below plot the difference between observed and predicted genetic gain under varying skew, α, for θ = -10 and θ = 10 respectively.

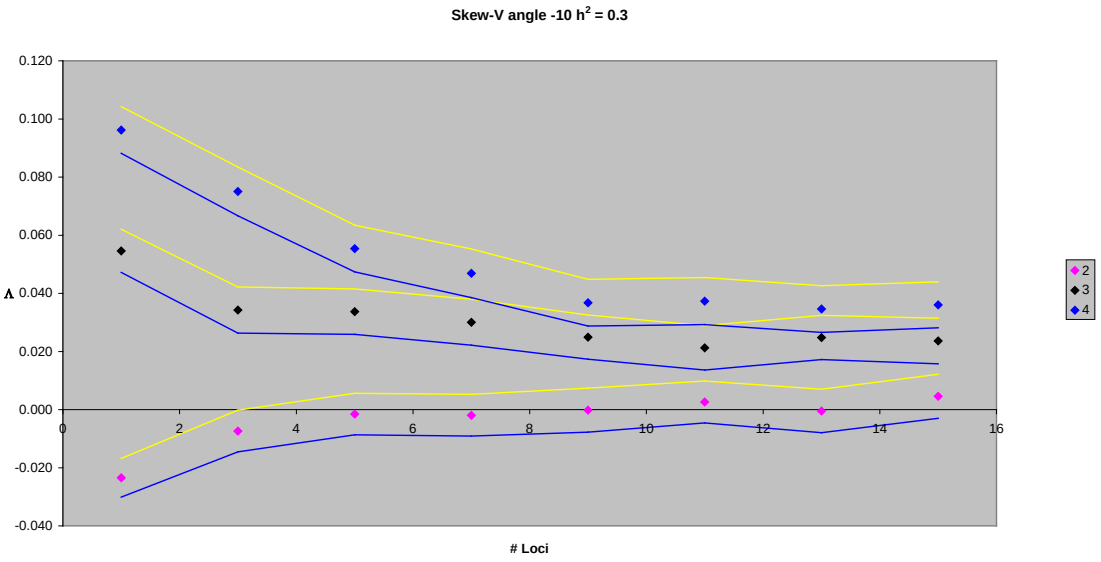

Figure 4.3: Skewed allele distribution

varying α, β=0.5, θ = -10, 95% confidence intervals, yellow lines are the upper value of the interval, blue lines are the lower value of the interval.

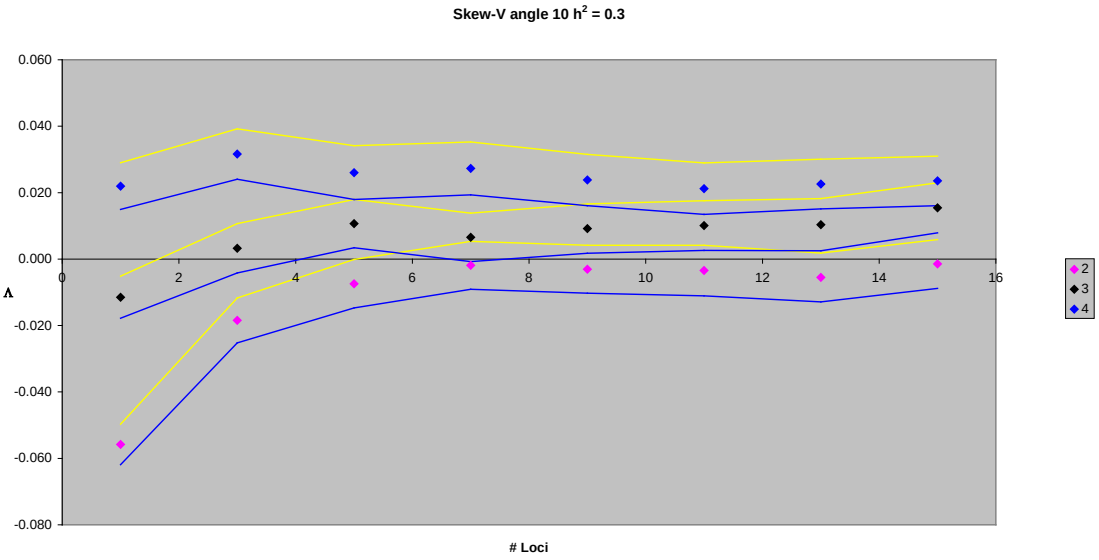

Figure 4.4: Skewed allele distribution

varying α, β=0.5, θ = 10, 95% confidence intervals, yellow lines are the upper value of the interval, blue lines are the lower value of the interval.

Figure 4.3 and Figure 4.4 address the issue of distribution skew. When α is increased from 2 the distribution becomes non-symmetric and right skewed. As can be seen in these two graphs, the infinitesimal predictions of genetic gain are very sensitive to the allele distribution being right skewed. Under right skew conditions it is possible to observe genetic gains that are greater than predicted genetic gains. Even when there are sufficient loci for a symmetric distribution to give equivalent observed and predicted genetic gains, right skew distributions gives consistently higher observed genetic gains to predicted genetic gains. This holds for both extremes of allele value frequencies investigated.

Figure 4.5, Figure 4.6, and Figure 4.7 below display the effects of linkage for the first and second generation of breeding.

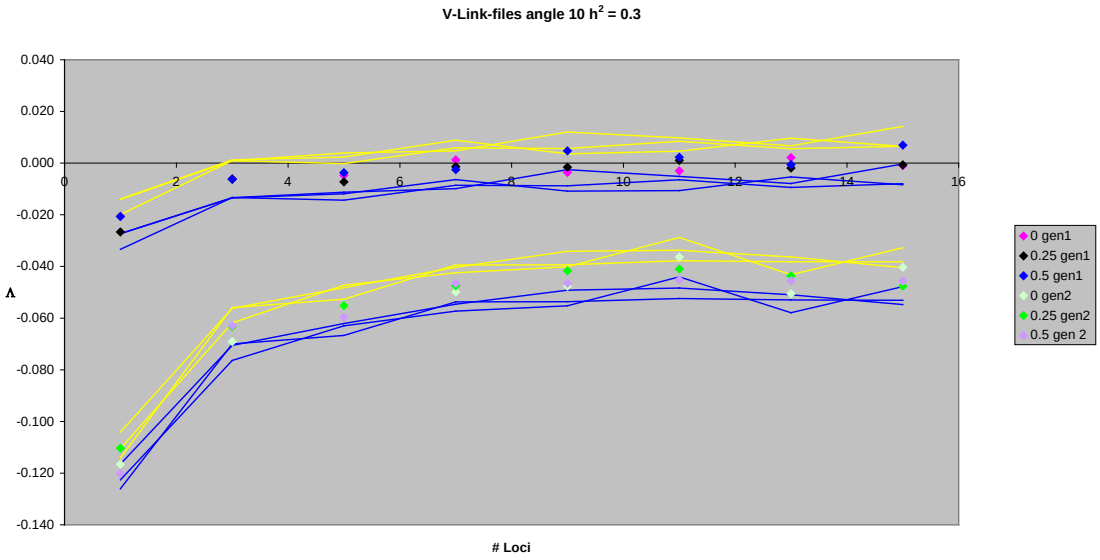

Figure 4.5: Linkage effects

varying β, α=2, θ =10, 2 generations, 95% confidence intervals, yellow lines are the upper value of the interval, blue lines are the lower value of the interval.

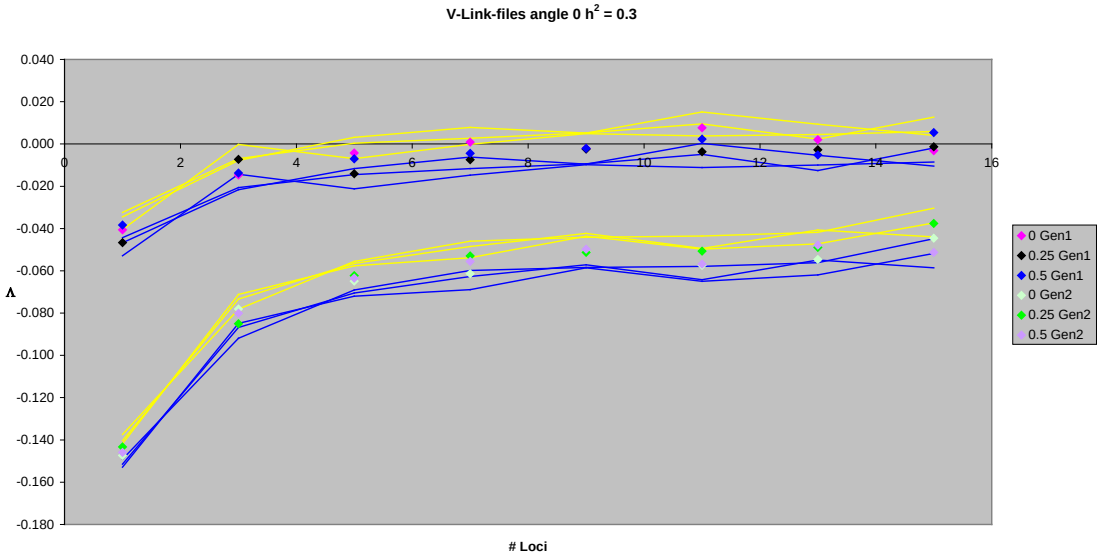

Figure 4.6: Linkage effects

varying β, α=2, θ =0, 2 generations, 95% confidence intervals, yellow lines are the upper value of the interval, blue lines are the lower value of the interval.

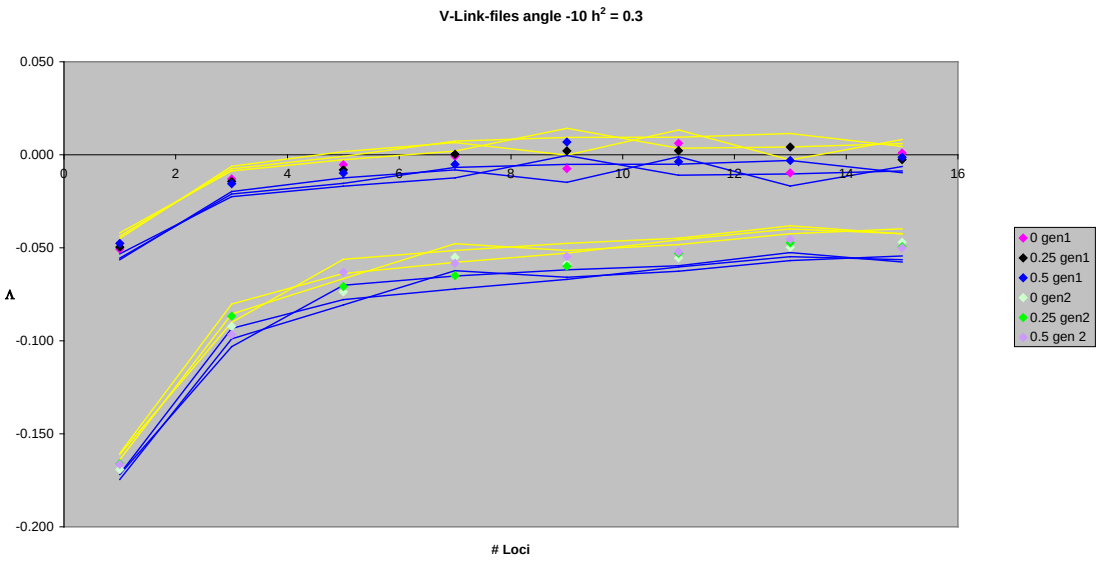

Figure 4.7: Linkage effects

varying β, α=2, θ =-10, 2 generations, 95% confidence intervals, yellow lines are the upper value of the interval, blue lines are the lower value of the interval.

For all three distribution shapes, the crossover probability, and therefore linkage, has no effect on the accuracy of the infinitesimal predictions in the first and second generations of breeding. The greater difference between observed and predicted genetic gain in the second generation is a result of the Bulmer effect, the decrease of genetic variation due to breeding and changing allele frequencies. The creation of left skew in allele distributions, resulting from the first generation of breeding, may also be a factor in this observed greater difference. This could be explicitly explored by making infinitesimal predictions with the Bulmer effect calculated.

4.1.5 Discussion

It was observed in Figure 4.2 that where there are few loci the distribution of allele frequencies can have an effect on the accuracy of the infinitesimal predictions. Verrier *et al.* (1991) observed in their review of prediction response to selection that the limit to selection response depends on the distributions of gene effects and gene frequencies in the base population when discussing the one locus model. That is selection response is affected by the shape of the allele distribution when there is one locus, as was observed.

When there is mutation with large effects, generally asymptotic response is reached much earlier than under the infinitesimal model (Keightley, 2004). That is, the observed

genetic gains are inconsistent with those predicted by the infinitesimal mode. Also for traits showing genetic asymmetry due to asymmetrical gene frequencies, non-linear offspring-parent regressions and heritabilities are expected (Frankham, 1990). Non-linear offspring-parent regressions and heritabilities violate a main assumption of the infinitesimal model, and can result in inaccurate infinitesimal prediction. These situations are consistent with the results seen in Figure 4.3 and Figure 4.4, where the allele distribution was skewed, or asymmetrical, and it was seen that infinitesimal prediction of genetic gain was biased. When there is a mutation with a large effect present, you get skewed allele distributions as in these simulations.

When linkage was investigated it was observed that linkage had no effect on the accuracy of the infinitesimal predictions in the first and second generations of breeding. This is consistent with Keightley and Hill (1987) who investigated mutation and linkage and found that when there are low mutation levels, which is the case here, linkage has little effect on selection response.

4.2 Summary

In Chapter 3, it was seen in the situation when there are few loci controlling a trait, to which selection pressure is applied, predictions of genetic gain given by the infinitesimal model have the potential to be biased and over estimate the genetic gain. This bias becomes greater with the interaction of fewer loci, fewer alleles and greater heritability of the trait. The bias starts to become apparent in one generation of selection, when the number of loci is less than 15. When using the infinitesimal model for predicting genetic gain, when there are few loci controlling the trait, this potential bias needs to be kept in mind.

It is thus found that the infinitesimal model is an accurate and useful tool for genetic gain prediction, but breaks down in the extreme situation of low loci count and high selection pressure. In this extreme situation the model consistently overestimates genetic gain. This may lead to potentially disappointing breeding results. When the non-linearity of the genetic value-phenotypic value regression is taken into account, for the cases where there are few alleles and few loci, more accurate prediction of the genetic gain can be made.

It is also found that the relationship between genetic gain and the selected population mean is non-linear and thus that heritability is not constant for all possible selection pressures. This is in spite of the linearity and constant heritability expectations of the infinitesimal model.

In Chapter 4, where the five allele distribution was investigated further, it has been shown that the allele distribution shape, as long as it is symmetric, has no effect on the quality of the infinitesimal prediction, when there are more than 5 loci present. The infinitesimal model requires symmetry in the allele distributions. The least acceptable requirement for reliable predictions is average symmetry over the allele distributions. If the allele distributions are generally skewed in a single direction, as can happen after several generations of breeding, the simulation results show that the infinitesimal predictions will become unreliable. The simulation results also show that linkage has no effect on the accuracy of the infinitesimal prediction for the first and second generations of breeding.

Chapter 5 – GE-Pig model

5.1 Introduction

This chapter introduces the selected animal biological growth model and the specially developed genetic structure. The motivation for adding a genetic structure to an animal growth model is first discussed and then the selected pig growth model is described. The newly developed genetic structure is introduced along with a method for generation of loci and allele values. The actual values used for the simulations are then presented. The two diets used during the simulations are delineated and the performance of the complete model, GE-Pig, on the two diets is considered.

5.2 Model motivation

With time, more and more genetic information is being discovered. This information includes locations of loci on chromosomes, the alleles present at each locus, and the effects of each of the alleles. At present there is a drive to find this information. However although molecular genetic information has been used in industry programs for several decades, primarily through gene-assisted selection and linkage disequilibrium markers-assisted selection, and is growing, the extent of use has not lived up to initial expectations, with most applications have been integrated into existing programs on an ad hoc basis (Dekkers, 2004). There exist breeding indices that include marker DNA, but it is possible that over the long term these methods can have detrimental effects on genetic gains obtained (Gibson 1994). This suggests that there is a need for a model that can incorporate this information as it becomes available. The model could then be used to predict the “best” combination of loci and alleles that animal breeders should be aiming for, and to see the effects various breeding methods may have on the animal population once loci and alleles have been identified.

The genetic information that is most accessible is average and interactive effects or additive and dominant effects of various alleles. Average and interactive effects compared with additive and dominant effects are different methods of summarizing the same information. Both these methods involve comparing, in a population of animals, the average performance of the animals that contain the allele of interest to the average

performance of the population. An example of a calculation of average and interactive effects is given in §5.2.1 and an example of additive and dominance effects is given in §5.2.2.

For this work additive and dominance effects have been included. As there is little information about the actual values for the additive and dominance effects, since research into specific loci and allele effects is a very new area, dominance effects have been assumed to be zero.

There currently exist growth models that will take information about an animal, such as maximum protein deposition and minimum lipid to protein ratio, and predict the growth of the animal on a given diet. These models have been used successfully to increase the performance of the animals, by optimising the diet for the genotype that is being grown. However these models currently require observed genetic traits of the animal as an input.

It would be desirable if these models could be expanded so that instead of starting with observed genetic traits, the genotype is supplied. This would then allow for applications in breeding simulations and the ability to search for the best genotype for a given diet. Overall this would make a model more flexible and powerful with a wider range of applications.

5.2.1 Example calculation of average and interactive effects

The following is an example of the calculation of average and interactive effect values for a 3-allele 1-locus setup for a trait, as summarised in Table 5.1. The three alleles are denoted A_1, A_2, and A_3, with two of the alleles present at the locus, notated, for example, by A_3A_2. Note that A_1A_2 is the same as A_2A_1. The average weight of the trait for each allele combination is given, along with the population average, μ, of the trait. The difference from the population average is also shown.

Table 5.1: Initial genotype data – equal allele frequencies.

		Genotype					
	μ	A_1A_1	A_1A_2	A_1A_3	A_2A_2	A_2A_3	A_3A_3
Average weight in grams	11	14	12	16	6	10	8
Difference from mean		3	1	5	–5	–1	–3

In general the average effect of A_i is given by,

$$\text{Average effect of } A_i = \frac{\sum_j \left(A_iA_j - \mu\right)}{j}.$$

The specific average effects for the example in Table 5.1 are thus,

$$\text{Average effect of } A_1 = \frac{3+1+5}{3} = 3,$$

$$\text{Average effect of } A_2 = \frac{1-5-1}{3} = -1.67,$$

$$\text{Average effect of } A_3 = \frac{5-1-3}{3} = 0.33.$$

The interactive effect of A_iA_j is given by,

$$\text{Interactive effect of } A_iA_j = \textit{Difference of } A_iA_j \textit{ from mean} - \left(\textit{Average effect of } A_i + \textit{Average effect of } A_j\right)$$

The interactive effect of A_1A_1, for the example in Table 5.1, is thus as follows,

$$A_1A_1 = 3-(3+3) = -3$$

The interactive effects for the example in Table 5.1 are summarised in Table 5.2 below.

Table 5.2: Calculated average and interactive effects.

		Genotype					
	μ	A_1A_1	A_1A_2	A_1A_3	A_2A_2	A_2A_3	A_3A_3
Average weight in grams	11	14	12	16	6	10	8
Difference from mean		3	1	5	–5	–1	–3
Combined average effects		6	1.33	3.33	–3.33	–1.34	0.66
Interactive effects		–3	–0.33	1.67	–1.67	0.34	–3.66

The average weight for a genotype is then calculated as follows

$$A_iA_j = \mu + \left(Average\,effect\,of\ A_i + Average\,effect\,of\ A_j \right) + Interactive\ effect\ of\ A_iA_j .$$

For the worked example in Table 5.1 the average weight for A_1A_1 is thus,

$$\begin{aligned} A_1A_1 &= 11 + (3+3) + (-3) \\ &= 14 \end{aligned}$$

5.2.2 Example calculation of additive and dominance effects

The following is an example of the calculation of additive and dominance values for the example in §5.2.1, a 3-allele 1-locus setup for a trait, with two of the alleles present at the locus, as summarised in Table 5.3. The average weight of the trait for each allele combination is given, along with the population average, μ, of the trait. The difference from the population average is also shown.

Table 5.3: Initial genotype data – equal allele frequencies.

		Genotype					
	μ	$\mathbf{A_1A_1}$	$\mathbf{A_1A_2}$	$\mathbf{A_1A_3}$	$\mathbf{A_2A_2}$	$\mathbf{A_2A_3}$	$\mathbf{A_3A_3}$
Average Weight in grams	11	14	12	16	6	10	8
Difference from mean		3	1	5	–5	–1	–3

In general the additive effect of A_i is given by,

$$\text{Additive effect of } A_i = \frac{A_iA_i - \mu}{2}.$$

The specific additive effects for the example in Table 5.1 are thus,

$$\text{Additive effect of } A_1 = \frac{14-11}{2} = 1.5,$$

$$\text{Additive effect of } A_2 = \frac{6-11}{2} = -2.5,$$

$$\text{Additive effect of } A_3 = \frac{8-11}{2} = -1.5.$$

The dominance effect of A_iA_j is given by,

$$\begin{aligned} \text{Dominance effect of } A_iA_j = {} & Difference\,of\ A_iA_j\ \ from\,mean \\ & -\left(Additive\,effect\,of\ A_i + Additive\,effect\,of\ A_j \right) \end{aligned}$$

The dominance effect of A_1A_2, for the example in Table 5.1, is thus,

$$\begin{aligned}\text{Dominance effect of } A_1A_2 &= 1-\left(1.5+\left(-2.5\right)\right)\\ &= 2\end{aligned}$$

The dominance effects for the example in Table 5.3 are summarised in Table 5.4.

Table 5.4: Calculated additive and dominance effects.

		Genotype					
	μ	A_1A_1	A_1A_2	A_1A_3	A_2A_2	A_2A_3	A_3A_3
Average Weight in grams	11	14	12	16	6	10	8
Difference from mean		3	1	5	–5	–1	–3
Combined Additive effects		3	–1	0	–5	–4	–3
Dominance effects		0	2	5	0	3	0

The average weight for a genotype is then calculated as follows

$$A_iA_j = \mu + \left(\textit{Additive effect of } A_i + \textit{Additive effect of } A_j\right) + \textit{Dominance effect of } A_iA_j .$$

For the worked example in Table 5.1 the average weight for A_1A_2 is thus,

$$\begin{aligned}A_1A_2 &= 11+\left(1.5+\left(-2.5\right)\right)+\left(2\right)\\ &= 12\end{aligned}$$

5.3 GE-Pig model overview

Combining a growth model with the genetic information about the effects of various alleles and loci would make the growth model a more powerful tool. This would allow the expanded model to be used for finding good/optimal combinations of alleles which produce desired performances, as well as allowing for the prediction of the results of breeding programs.

A specific pig model, as described in §5.3.1, was chosen and then augmented with a genetic structure, as described in §5.3.3, giving the full, genetically enhanced, GE-Pig model. The genetic structure will allow the pig model to produce the desired genetic performance described in §5.3.2 when the enhanced model parameters are initialised as described in §5.3.4. An overview of the full GE-Pig structure is shown below in Figure 5.1.

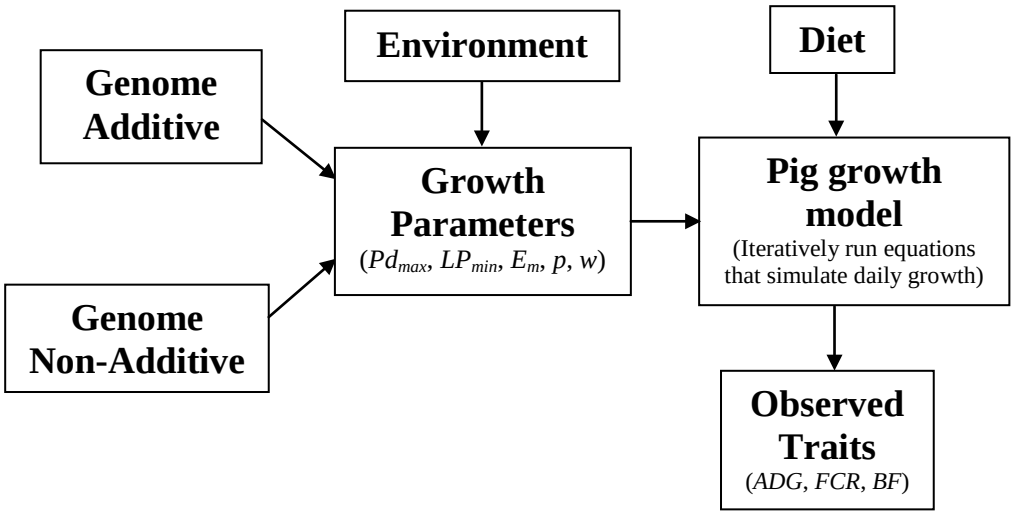

Figure 5.1: GE-Pig model

The pig growth model requires growth parameters (sub phenotypic traits) to determine the growth of the pig. To combine the pig growth model and genetic information, the required parameters have been modelled with additive, non-additive, and environmental components. The additive genetic effects are used to model additive effects of alleles as they become available in the literature. Currently these have been calculated with the method described in §5.3.4 as insufficient additive genetic information is currently known. Dominance effects may easily be added to this section of the model. The non-additive genetic effect is used to model all the genetic information that is unexplained by the additive genetic effects. The environmental effects model the influences that the environment has on the observed values of the growth parameters of the growth model. These environmental effects are used to control the heritabilities of the growth parameters.

Assuming normal distributions will sufficiently describe the non-additive and environmental effects and they are independent of each other and the additive effects is a significant assumption. In reality, very complex interactions occur in creating these effects which are not fully understood. However the means, variances, correlations and heritabilities of the sub phenotypic traits are maintained by the structure chosen. It is expected that this is sufficient to limit the effects of the chosen structure, compared to other possible structures.

5.3.1 Pig growth model

The pig growth model (De Lange 1995), as shown in Figure 5.2 and Figure 5.3, has five parameters, supplied by the user, to describe the type of pig to be grown. These parameters are

Pd_{max}	maximum protein deposition
LP_{min}	minimum lipid to protein ratio
E_m	energy requirements for maintenance
p	feed intake capacity expressed as a percentage of the NRC digestible energy curve
w	chemical vs. physical body composition. Used in determining the amount of water present in the pig.

These growth parameters drive how the model grows the pig on a given diet.

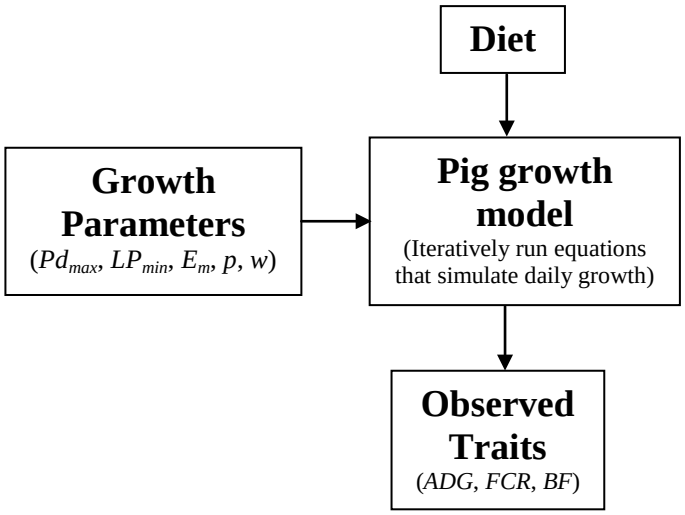

Figure 5.2: Pig Growth model

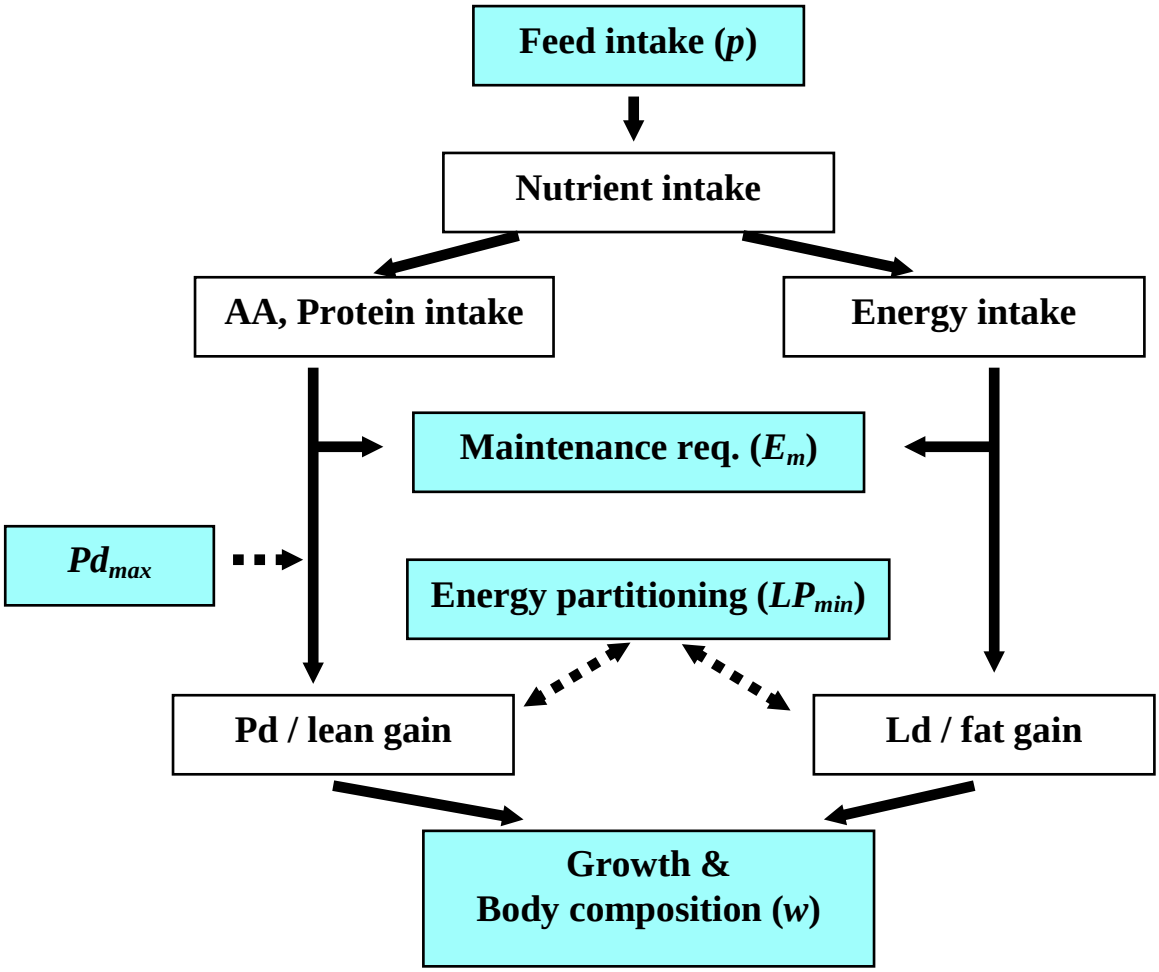

Figure 5.3: Pig growth model.

The coloured sections are influenced by the five growth parameters.

The model also requires information on the composition of the diet. The pig model implemented takes into account six amino acids in the diet when growing the pigs. These are Lysine, Methionine, Methionine and Cystoline, Threonine, Tryptophane, and Isoleucine. The distribution of amino acids, *AA*, as given below in Table 5.5 (ideal amino acid balance), is the requirement for optimum growth (Commonwealth Agricultural Bureaux 1981).

Table 5.5: Ideal amino acid balance.

Amino Acid	Balance (g/kg)
Lysine	0.07
Methionine	0.0175
Meth + Cyst	0.035
Threonine	0.042
Tryptophane	0.01
Isoleucine	0.038

The model calculates the growth on a daily basis, along with information which includes back fat, *BF*, average daily gain, *ADG*, and feed conversion ratio, *FCR*, of interest to pig breeders and farmers. The equations for calculating the growth are supplied in Appendix A. For the purposes of this research the pigs are grown from a live weight of 20kg to a live weight of 90kg.

5.3.2 Growth parameter information

Currently there is very limited information on the generic parameters for Pd_{max}, LP_{min}, E_m, p and w. For the simulation model values for the heritabilities, means, standard deviations, and correlations of this five parameters were pieced together from various work done on pig growth (Pd_{max}: Whittemore *et al.,* 2001a; LP_{min}: from data published later in De Lange *et al.,* 2008; *Em*: Whittemore *et al.,* 2001b; *p*: from data collected on NZ farms; *w*: De Lange *et al.,* 2003;) as well as other similar traits (feed intake, lean tissue growth rate, percent lean, feed conversion ratio) (Morel, 1988; Morel *et al.,* 1997). The mean parameters chosen correspond to a modern lean pig genotype.

Table 5.6 below summarises the values used during this research for the means, standard deviations and heritabilities of the five growth parameters, Pd_{max}, LP_{min}, E_m, p, and w.

Table 5.6: Mean, standard deviation, and heritability of growth parameters.

	Model Assumptions			Literature	
	Mean	**SD**	**Heritability**	**Min**	**Max**
Pd_{max}	170	25	0.45	115	260
LP_{min}	0.7	0.12	0.3	0.3	1.2
E_m	0.485	0.03	0.4	0.434	0.523
p	0.9	0.08	0.25	0.8	1.0
w	5.05	0.15	0.5	4.90	5.62

Table 5.7 below summarises the values used during this research for the correlations of the five growth parameters, Pd_{max}, LP_{min}, E_m, p, and w.

Table 5.7: Correlations of growth parameters

	Pd_{max}	LP_{min}	E_m	p	w
Pd_{max}	1	-0.55	0.2	0.25	0.3
LP_{min}		1	-0.15	0.3	-0.2
E_m			1	-0.2	0.3
p				1	-0.15
w					1

5.3.3 Growth parameter structure

The five growth parameters in the pig growth model are an attempt to describe the genetic makeup of the pig that is to be grown. This section develops a genetic structure that determines the growth parameter values from genetic information.

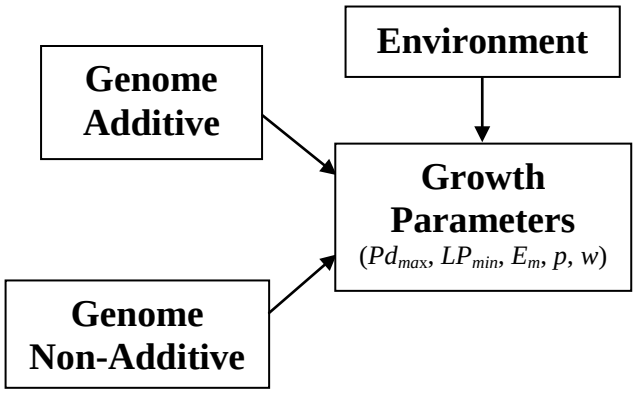

Figure 5.4: Growth parameter structure

Each growth parameter is derived from a combination of three influences, two of which are genetic and one environmental. The three influences, Additive effects, Non-Additive effects, and Environmental effects, are each a weighted sum of random variables that have mean zero and variance one. These individual structures are then scaled to the appropriate variance, as determined in §5.3.5.1. The growth parameter value is then the sum of the three influences and the growth parameter mean, which will give the desired growth parameter information as given in §5.3.2. That is,

$$\text{Growth Parameter}_i = \mu_i + \text{Additive effect}_i + \text{Non-Additive effect}_i + \text{Environmental effect}_i$$

$$\text{Growth Parameter}_i = \mu_i + \sigma_{A_i} \sum_{\text{loci } j} w_{ij} \left(g\left(X_{j,1}\right) + g\left(X_{j,2}\right) \right) + \sigma_{N_i} \sum_{\text{distributions } k} \omega_{ik} Y_k + \sigma_{E_i} \sum_{\text{distributions } l} \nu_{il} Z_l$$

with

$$\sum_{\text{loci } j} w_{ij}^2 = 1, \quad \sum_{\text{distributions } k} \omega_{ik}^2 = 1, \quad \sum_{\text{distributions } l} \nu_{il}^2 = 1$$

where

- μ_i is the mean of growth parameter *i*, as given in §5.3.2
- σ_{A_i}, σ_{N_i}, and σ_{E_i} are the standard deviations for the Additive, Non-Additive, and Environmental effects respectively, as given in §5.3.5.1
- w_{ij}, ω_{ik}, and ν_{il} are weights used to create the desired correlations, as calculated in §5.3.5.2, between the growth parameters
- $g\left(X_{j,1}\right)$ and $g\left(X_{j,2}\right)$ are the genetic effects of alleles $X_{j,1}$ and $X_{j,2}$ respectively at loci *j* from the additive structure. These genetic effects are independent of the growth parameters, and each is uniformly distributed with mean zero and variance 0.5. For the model implemented 6 alleles were present at each locus.
- Y_k and Z_l are random variables whose distributions are normal with mean zero and variance one from the Non-Additive structure and the Environmental structure respectively. Y_k and Z_l are independent from each other.

Each of the two genetic influences and the environmental influence has the same general structure,

$$Effect_i = \sum_{j=1}^{s} x_{ij} X_j$$

with

$$\sum_{j=1}^{s} x_{ij}^2 = 1, \text{ for all } i$$

where

- s is the number of distributions
- $X_1, \ldots, X_s$ are independent with mean zero and variance one.
- x_{ij}^2 is the proportion of the variation of $Effect_i$ attributable to distribution j

Since $X_1, \ldots, X_s$ are independent with mean zero and variance one, and $\sum_{j=1}^{s} x_{ij}^2 = 1$, for all i, $Effect_i$ also has mean zero and variance one. Also, when x_{ij} is positive/negative, $Effect_i$ is positively/negatively correlated with Xj.

With the additional assumption that

$$x_{ij}^2 = \begin{cases} x_j^2 & \text{if distribution } j \text{ contributes to effect } i \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } i \text{ and } j$$

that is, each random variable Xj has a weighting of $-x_j$, 0, or x_j for each $Effect_i$, it follows that

$$\left|Correlation\left(Effect_h, Effect_i\right)\right| = \sum_{j=1}^{s} \begin{cases} x_j^2 & if\ x_{hj}^2 = x_{ij}^2 \\ 0 & otherwise \end{cases}$$

given that:

- $x_{ij} = x_{hj}$ for all j if the correlation between $Effect_i$ and $Effect_h$ is positive,
- $x_{ij} = -x_{hj}$ for all j if the correlation between $Effect_i$ and $Effect_h$ is negative.

The Environmental and Non-Additive effects structures are modelled with individual sets of independent standard normal distributions.

The Additive effects structure is modelled with an n-locus, m-allele additive model mapped onto chromosome pairs. The m alleles at each of the n loci are distributed with a discrete uniform distribution with mean zero and variance 0.5. It was shown in

Chapter 4 that the infinitesimal model required symmetric distributions for the alleles. There are two alleles present at any one time at each locus, giving

$$X_j = g\left(X_{j,1}\right) + g\left(X_{j,2}\right)$$

which results in the genetic value at each locus having a mean zero and variance one as required for the general structure.

Dominance effects were not modelled but could be included in the additive effects structure with the following change.

$$Effect_i = \sum_{j=1}^{s} x_{ij} X_{ij} = \sum_{j=1}^{s} x_{ij} \left[g_{ij}\left(X_{j,1}\right) + g_{ij}\left(X_{j,2}\right) + d_{ij}\left(X_{j,1}, X_{j,2}\right) \right]$$

5.3.4 Generation of weights - method

For a given set of correlations between the five growth parameters, there exist many possible solutions for the weights x_{ij} in §5.3.3. These solutions will have varying numbers of distributions for any given set of growth parameter correlations. For the purpose of this model, only one solution of the many possible solutions is required. This solution is produced using the linear programming method detailed below.

A linear program cannot be formulated directly on the variables x_{ij} as the number of x_{ij} is dependent on the number of random variables *j*, which varies over all possible solutions. Thus to enable the formulation of the linear program, a new parameter x_{abcde} and new random variables X_{abcde} are defined, where *a*, *b*, *c*, *d*, and *e* are binary variables, to replace x_{ij} and *Xj*. The parameter, x_{abcde}, is then defined as the proportion that random variable X_{abcde} contributes to the growth parameters Pd_{max}, LP_{min}, E_m, *p*, and *w* when *a*, *b*, *c*, *d*, and *e* respectively are one. Since the variables *a*, *b*, *c*, *d*, and *e* are defined as binary, the variable x^2_{abcde} is indexing 2^5 = 32 variables, each with its own random variable X_{abcde}. The distribution for each growth parameter *i* can then be restated from the original parameters and random variables

$$\text{Growth parameter}_i = \sum_{j=1}^{s} x_{ij} X_j$$

with

$$\sum_{j=1}^{s} x_{ij}^2 = 1, \text{ for all } i$$

to the new parameters and random variables

$$\text{Growth parameter}_i = \sum_{a=\begin{cases}1 & i=1\\ 0 & otherwise\end{cases}}^{1} \sum_{b=\begin{cases}1 & i=2\\ 0 & otherwise\end{cases}}^{1} \sum_{c=\begin{cases}1 & i=3\\ 0 & otherwise\end{cases}}^{1} \sum_{d=\begin{cases}1 & i=4\\ 0 & otherwise\end{cases}}^{1} \sum_{e=\begin{cases}1 & i=5\\ 0 & otherwise\end{cases}}^{1} x_{abcde} X_{abcde}$$

with

$$\sum_{a=\begin{cases}1 & h \text{ or } i=1\\ 0 & otherwise\end{cases}}^{1} \sum_{b=\begin{cases}1 & h \text{ or } i=2\\ 0 & otherwise\end{cases}}^{1} \sum_{c=\begin{cases}1 & h \text{ or } i=3\\ 0 & otherwise\end{cases}}^{1} \sum_{d=\begin{cases}1 & h \text{ or } i=4\\ 0 & otherwise\end{cases}}^{1} \sum_{e=\begin{cases}1 & h \text{ or } i=5\\ 0 & otherwise\end{cases}}^{1} x_{abcde}^2 = 1$$

This new set of parameters has a fixed number of random variables, thus allowing a linear program to be formed. The linear program constraints can be stated as

$$\sum_{a=\begin{cases}1 & h \text{ or } i=1\\ 0 & otherwise\end{cases}}^{1} \sum_{b=\begin{cases}1 & h \text{ or } i=2\\ 0 & otherwise\end{cases}}^{1} \sum_{c=\begin{cases}1 & h \text{ or } i=3\\ 0 & otherwise\end{cases}}^{1} \sum_{d=\begin{cases}1 & h \text{ or } i=4\\ 0 & otherwise\end{cases}}^{1} \sum_{e=\begin{cases}1 & h \text{ or } i=5\\ 0 & otherwise\end{cases}}^{1} x_{abcde}^2 = \left|\text{Corr}\left(h,i\right)\right| \quad \textit{for all } h,i = 1,2,3,4,5$$

$$x_{abcde}^2 \geq 0 \quad \textit{for all } a,b,c,d,e = 0,1$$

$$x_{abcde}^2 \leq 1 \quad \textit{for all } a,b,c,d,e = 0,1$$

$$x_{abcde}^2 = 0 \quad \textit{if there are opposing correlations}$$

where h, i = 1, 2, 3, 4, 5 correspond to the growth parameters Pd_{max}, LP_{min}, E_m, p, and w respectively.

The first set of constraints enforces the correlation matrix requirements on the variables. The second set of constraints is the standard non-negativity constraint. The third set of constraints, $x_{abcde}^2 \leq 1$, is included to make explicit the range of x_{abcde}^2, although the constraint is technically redundant. The last set of constraints, $x_{abcde}^2 = 0$ if there are opposing correlations, is concerned with excluding situations where there are oppositions between the correlations of three growth parameters. These opposing correlations occur when the first and second growth parameters are positively/(negatively) correlated, the second and third growth parameters are positively/(negatively) correlated, but the first and third growth parameters are negatively/(negatively) correlated. These situations are illustrated in Figure 5.5. Variables that contribute to all three growth parameters that have opposing correlations are explicitly excluded from the formulation.

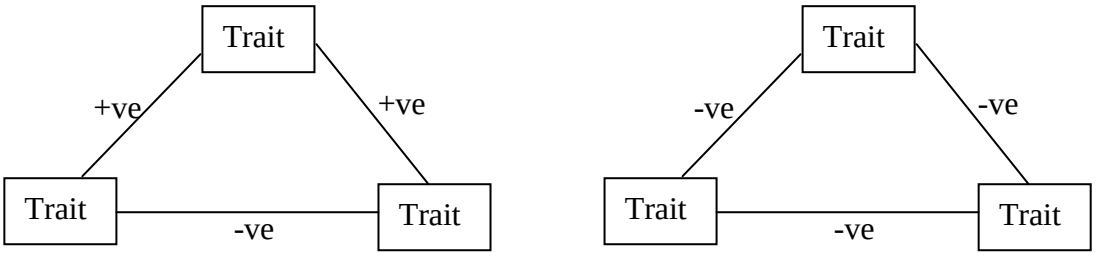

Figure 5.5: Opposing correlations.

The objective can now be added to the linear program. A simple objective is to minimize the sum of the variables x^2_{abcde}, which gives the following linear program.

$$\min \sum_{a=0}^{1}\sum_{b=0}^{1}\sum_{c=0}^{1}\sum_{d=0}^{1}\sum_{e=0}^{1} x^2_{abcde}$$

s.t.

$$\sum_{a=\begin{cases}1 & h \text{ or } i=1\\ 0 & otherwise\end{cases}}^{1}\ \sum_{b=\begin{cases}1 & h \text{ or } i=2\\ 0 & otherwise\end{cases}}^{1}\ \sum_{c=\begin{cases}1 & h \text{ or } i=3\\ 0 & otherwise\end{cases}}^{1}\ \sum_{d=\begin{cases}1 & h \text{ or } i=4\\ 0 & otherwise\end{cases}}^{1}\ \sum_{e=\begin{cases}1 & h \text{ or } i=5\\ 0 & otherwise\end{cases}}^{1} x^2_{abcde} = \left|\mathrm{cor}\left(h,i\right)\right| \quad \textit{for all } h,i = 1,2,3,4,5$$

$$x^2_{abcde} \geq 0 \quad \textit{for all } a,b,c,d,e = 0,1$$

$$x^2_{abcde} \leq 1 \quad \textit{for all } a,b,c,d,e = 0,1$$

$$x^2_{abcde} = 0 \quad \textit{if there are opposing correlations}$$

This linear program will minimize the sum of the variables x^2_{abcde}. The minimization will in part be achieved by maximizing the number of variables that contribute to multiple growth parameters. This results in the linear program indirectly minimizing the number of random variables that have weights greater than zero and maximizing the number of random variables that contribute to more than one growth parameter.

For this linear program the weights x^2_j in §5.3.3 are the various values of x^2_{abcde}. If x^2_{abcde} equals zero then distribution X_{abcde} can be dropped from the solution.

For example, if $x^2_{10110} = 0.1$ and there are correlations between the traits as shown in Table 5.8 below, and x^2_{10110} corresponds to the first distribution, $j = 1$, then $x^2_{11} = x^2_{31} = x^2_{41} = 0.1$.

Table 5.8: Signs of the example correlations for traits 1, 3 and 4.

	Trait 3	**Trait 4**
Trait 1	+ve	–ve
Trait 3		–ve

There are then two possible solutions for the x_{ij}, as shown below in Figure 5.6.

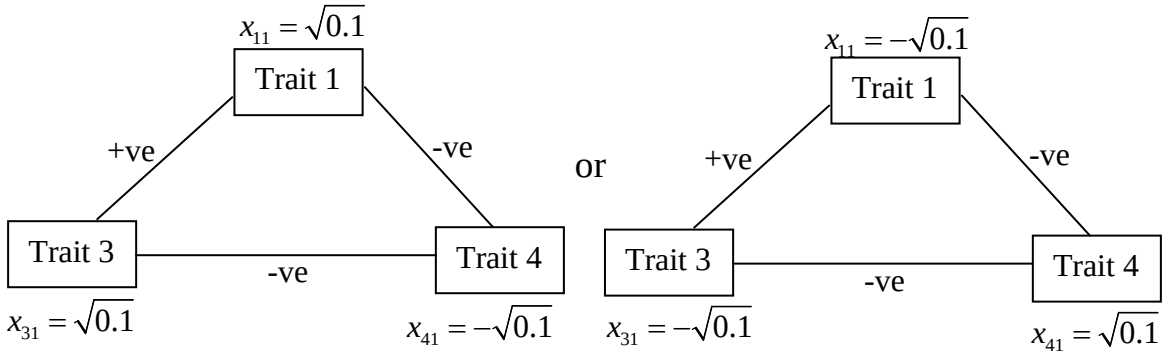

Figure 5.6: Example assignments of weights

5.3.4.1 Generation of weights – practical example

Suppose there are three traits with the correlation matrix in Table 5.9.

Table 5.9: Example correlation matrix

Trait	**1**	**2**	**3**
1	1	0.1	0.2
2		1	-0.05
3			1

Then the possible variables in the linear program and the traits that they contribute to are shown in Table 5.10.

Table 5.10: Example linear program variables

	Trait		
Variable	**1**	**2**	**3**
x^2_{100}	✓		
x^2_{010}		✓	
x^2_{001}			✓
x^2_{110}	✓	✓	
x^2_{101}	✓		✓
x^2_{011}		✓	✓
x^2_{111}	✓	✓	✓

There exists an opposing correlation in this example between traits 1, 2, and 3, therefore $x^2_{111} = 0$. This gives rise to a linear program for this example of

$$\min\ x^2_{100} + x^2_{010} + x^2_{001} + x^2_{110} + x^2_{101} + x^2_{011} + x^2_{111}$$

s.t.

$$\begin{aligned}
x^2_{100} + x^2_{101} + x^2_{011} + x^2_{111} &= 1\\
x^2_{110} + x^2_{111} &= 0.1\\
x^2_{101} + x^2_{111} &= 0.2\\
x^2_{010} + x^2_{110} + x^2_{011} + x^2_{111} &= 1\\
x^2_{011} + x^2_{111} &= 0.05\\
x^2_{001} + x^2_{101} + x^2_{111} &= 1\\
x^2_{100} &< 1\\
x^2_{010} &< 1\\
x^2_{001} &< 1\\
x^2_{110} &< 1\\
x^2_{101} &< 1\\
x^2_{011} &< 1\\
x^2_{111} &< 1\\
x^2_{111} &= 0
\end{aligned}$$

The solution for this linear program is given in Table 5.11 below.

Table 5.11: Example linear program solution

	Trait			
Variables	**1**	**2**	**3**	**value**
x^2_{100}	✓			0.7
x^2_{010}		✓		0.85
x^2_{001}			✓	0.75
x^2_{110}	✓	✓		0.1
x^2_{101}	✓		✓	0.2
x^2_{011}		✓	✓	0.05
x^2_{111}	✓	✓	✓	0

This then gives rise to two possible assignments of the weights as shown in Table 5.12 and Table 5.13.

Table 5.12: Example assignment of weights - 1

	Trait		
Distribution	**1**	**2**	**3**
1	$\sqrt{0.7}$		
2		$\sqrt{0.85}$	
3			$\sqrt{0.75}$
4	$\sqrt{0.1}$	$\sqrt{0.1}$	
5	$\sqrt{0.2}$		$\sqrt{0.2}$
6		$\sqrt{0.05}$	$-\sqrt{0.05}$

Table 5.13: Example assignment of weights - 2

	Trait		
Distribution	**1**	**2**	**3**
1	$\sqrt{0.7}$		
2		$\sqrt{0.85}$	
3			$\sqrt{0.75}$
4	$\sqrt{0.1}$	$\sqrt{0.1}$	
5	$\sqrt{0.2}$		$\sqrt{0.2}$
6		$-\sqrt{0.05}$	$\sqrt{0.05}$

5.3.5 Growth parameter information subdivision

The following two sub-sections, §5.3.5.1 and §5.3.5.2, subdivide the means, variances and correlations given in §5.3.2 into three sets, one for each of the three structures, Additive, Non-Additive and Environmental, given in §5.3.3.

5.3.5.1 Means and variances

Table 5.14 (a copy of a portion of Table 5.6 from §5.3.2 for convenience) below summarises the values used during this research for the means, standard deviations and heritability of the five growth parameters, Pd_{max}, LP_{min}, E_m, p, and w.

Table 5.14: Mean, standard deviation, and heritability of growth parameters.

	Mean	**SD**	**Heritability**
$\boldsymbol{Pd_{max}}$	170	25	0.45
$\boldsymbol{LP_{min}}$	0.7	0.12	0.3
$\boldsymbol{E_m}$	0.485	0.03	0.4
$\boldsymbol{p}$	0.9	0.08	0.25
$\boldsymbol{w}$	5.05	0.15	0.5

These means, standard deviations and heritabilities need to be separated for the three structures, Additive, Non-Additive, and Environmental, that will be used to determine the growth parameter values.

Heritability may be used to separate the standard deviation for the Environmental structure from the Additive and Non-Additive genetic structures, as heritability relates genetic and phenotypic variance such that

$$h^2 = \frac{\sigma_G^2}{\sigma_P^2}$$

Also the phenotypic, genetic and environmental variances have the relationship

$$\sigma_P^2 = \sigma_G^2 + \sigma_E^2 .$$

Rearranging these two equations gives

$$\sigma_G^2 = h^2 \sigma_p^2$$
$$\sigma_E^2 = \sigma_p^2 - \sigma_G^2$$

which, when applied to the figures in Table 5.14, gives the standard deviations in Table 5.15.

Table 5.15: Genetic and environmental standard deviations of growth parameters

	Genetic SD	**Environmental SD**
$\boldsymbol{Pd_{max}}$	16.8	18.5
$\boldsymbol{LP_{min}}$	0.0657	0.100
Em	0.0189	0.0232
p	0.04	0.0693
w	0.106	0.106

The genetic standard deviations need to be further separated between the Additive and Non-Additive. However there is little information available as to the proportion of genetic variation that is additive. Therefore for the purpose of being able to generate a genetic structure, it has been assumed that the proportion of additive genetic variance to non-additive genetic variance is $k = 0.7$. The variable k effectively controls the amount of known additive genetic information in the system. Under this assumption the standard deviations in Table 5.16 are obtained for the additive genetic structure, non-additive genetic structure, and environmental structure.

Table 5.16: Standard deviations of growth parameters.

	Additive SD	**Non-Additive SD**	**Environmental SD**
$\boldsymbol{Pd_{max}}$	14.0	9.19	18.5
$\boldsymbol{LP_{min}}$	0.0550	0.036	0.100
$\boldsymbol{Em}$	0.0159	0.0104	0.0232
$\boldsymbol{p}$	0.0335	0.0219	0.0693
$\boldsymbol{w}$	0.0887	0.0581	0.106

5.3.5.2 Correlations

Table 5.17 (a copy of Table 5.7 from §5.3.2 for convenience) below, summarises the values used during this research for the correlations of the five growth parameters, Pd_{max}, LP_{min}, E_m, p, and w.

Table 5.17: Correlations of growth parameters

	$\boldsymbol{Pd_{max}}$	$\boldsymbol{LP_{min}}$	$\boldsymbol{E_m}$	$\boldsymbol{p}$	$\boldsymbol{w}$
$\boldsymbol{Pd_{max}}$	1	-0.55	0.2	0.25	0.3
$\boldsymbol{LP_{min}}$		1	-0.15	0.3	-0.2
$\boldsymbol{E_m}$			1	-0.2	0.3
$\boldsymbol{p}$				1	-0.15
$\boldsymbol{w}$					1

Little information is known as to how these correlations are separated between genetic and environmental factors, therefore it was assumed that the proportion of environmental covariance was q = 0.35, and the genetic (additive and non-additive) covariance was $(1 – q)$ = 0.65. The value of 0.35 for q was chosen as it lies within the range of heritabilities for the growth parameters. For the separation between the genetic covariances it was assumed that a proportion of r = 0.7 was additive and $(1 – r)$ = 0.3 was non-additive. This results in the same correlation tables for the Additive and Non-Additive structures, since the genetic variances were also split with a proportion of 0.7. The equations and process used to separate the correlation tables are summarised in Figure 5.7.

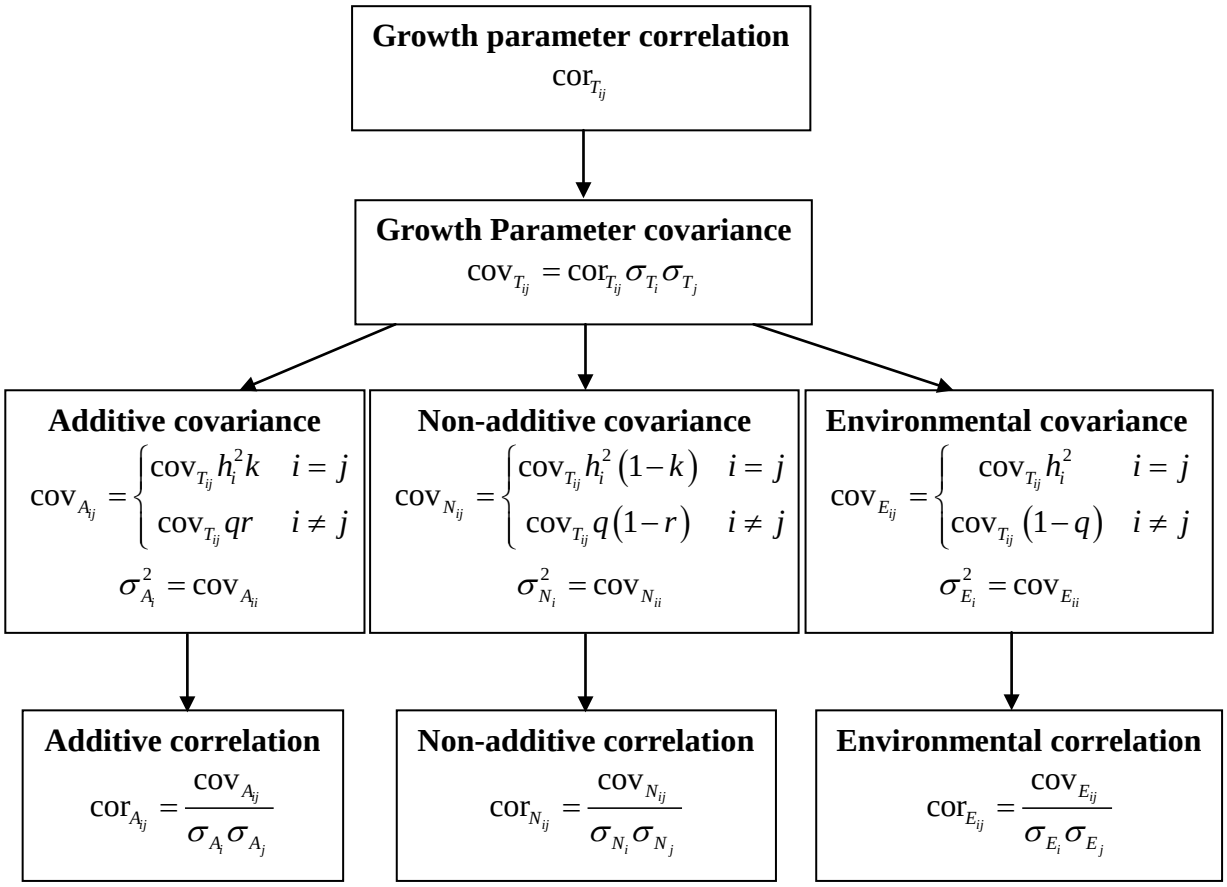

Figure 5.7: Equations and process for separating correlation table.

Thus using the means, standard deviations, and heritabilities in Table 5.14, correlations in Table 5.17, and values of k = 0.7, q = 0.35, and r = 0.7, gives the environmental correlations shown in Table 5.18 and the additive and non-additive correlations in Table 5.19.

Table 5.18: Environmental correlations of growth parameters

	Pd_{max}	LP_{min}	E_m	p	w
Pd_{max}	1	-0.57616	0.226301	0.253012	0.37185
LP_{min}		1	-0.15045	0.269126	-0.21974
E_m			1	-0.19379	0.35602
p				1	-0.15922
w					1

Table 5.19: Additive and non-additive genetic correlations of growth parameters

	Pd_{max}	LP_{min}	E_m	p	w
Pd_{max}	1	-0.52392	0.164992	0.260875	0.221359
LP_{min}		1	-0.15155	0.383406	-0.18074
E_m			1	-0.22136	0.234787
p				1	-0.14849
w					1

5.3.6 Calculation of weights – simulation values

Before applying the linear program in §5.3.4 to the environmental, additive and non-additive correlations, all the opposing correlations need to be identified. The existing opposing correlations are Pd_{max}-LP_{min}-p, Pd_{max}-E_m-p, and Pd_{max}-p-w, resulting in the linear program for the three formulations having the following opposing correlation constraints

$$x^2_{11010} = 0 \qquad x^2_{10110} = 0 \qquad x^2_{10011} = 0 \qquad x^2_{11110} = 0 \qquad x^2_{11011} = 0$$
$$x^2_{10111} = 0 \qquad x^2_{11111} = 0$$

When the linear program in §5.3.4, is applied to the environmental correlations, the solution is

$$x^2_{11000} = 0.300946784 \quad x^2_{10100} = 0.018714120 \quad x^2_{10010} = 0.253012157$$
$$x^2_{11100} = 0.055476529 \quad x^2_{11001} = 0.219740106 \quad x^2_{10101} = 0.152110304$$
$$x^2_{01000} = 0.059741775 \quad x^2_{01100} = 0.094969237 \quad x^2_{01010} = 0.269125568$$
$$x^2_{00100} = 0.281027893 \quad x^2_{00110} = 0.193792558 \quad x^2_{00101} = 0.203909359$$
$$x^2_{00010} = 0.124852884 \quad x^2_{00011} = 0.159216833 \quad x^2_{00001} = 0.265023398$$

This linear program solution has 15 normal distributions with the weighting structure shown in Table 5.20.

Table 5.20: Environmental structure weights and standard deviations

Normal Distribution	Pd_{max}	LP_{min}	E_m	P	w
1	$\sqrt{0.30095}$	$-\sqrt{0.30095}$			
2	$\sqrt{0.01871}$		$\sqrt{0.01871}$		
3	$\sqrt{0.25301}$			$\sqrt{0.25301}$	
4	$\sqrt{0.05548}$	$-\sqrt{0.05548}$	$\sqrt{0.05548}$		
5	$\sqrt{0.21974}$	$-\sqrt{0.21974}$			$\sqrt{0.21974}$
6	$\sqrt{0.15211}$		$\sqrt{0.15211}$		$\sqrt{0.15211}$
7		$\sqrt{0.05974}$			
8		$-\sqrt{0.09497}$	$\sqrt{0.09497}$		
9		$\sqrt{0.26913}$		$\sqrt{0.26913}$	
10			$\sqrt{0.28103}$		
11			$-\sqrt{0.19379}$	$\sqrt{0.19379}$	
12			$\sqrt{0.20391}$		$\sqrt{0.20391}$
13				$\sqrt{0.12485}$	
14				$-\sqrt{0.15922}$	$\sqrt{0.15922}$
15					$\sqrt{0.26502}$
σ	18.5	0.100	0.0232	0.0693	0.106

Applying the linear program to the additive and non-additive correlations gives a solution of

$x^2_{11000} = 0.352774384$ $x^2_{10100} = 0.106112707$ $x^2_{10010} = 0.260874597$

$x^2_{10001} = 0.109094056$ $x^2_{11100} = 0.058878875$ $x^2_{11001} = 0.112265380$

$x^2_{01100} = 0.038333975$ $x^2_{01010} = 0.369273542$ $x^2_{01101} = 0.054341595$

$x^2_{01011} = 0.014132248$ $x^2_{00100} = 0.340527869$ $x^2_{00110} = 0.221359436$

$x^2_{00101} = 0.180445543$ $x^2_{00011} = 0.134360176$ $x^2_{00001} = 0.395361002$

This linear program solution also has 15 normal distributions with the weighting structure shown in Table 5.21 for the non-additive effect.

Table 5.21: Non-Additive structure weights and standard deviations

Normal Distribution	Pd_{max}	LP_{min}	E_m	P	w
1	$\sqrt{0.35277}$	$-\sqrt{0.35277}$			
2	$\sqrt{0.10611}$		$\sqrt{0.10611}$		
3	$\sqrt{0.26087}$			$\sqrt{0.26087}$	
4	$\sqrt{0.10909}$				$\sqrt{0.10909}$
5	$\sqrt{0.05888}$	$-\sqrt{0.05888}$	$\sqrt{0.05888}$		
6	$\sqrt{0.11227}$	$-\sqrt{0.11227}$			$\sqrt{0.11227}$
7		$-\sqrt{0.03833}$	$\sqrt{0.03833}$		
8		$\sqrt{0.36927}$		$\sqrt{0.36927}$	
9		$-\sqrt{0.05434}$	$\sqrt{0.05434}$		$\sqrt{0.05434}$
10		$\sqrt{0.01413}$		$\sqrt{0.01413}$	$-\sqrt{0.01413}$
11			$\sqrt{0.34053}$		
12			$-\sqrt{0.22136}$	$\sqrt{0.22136}$	
13			$\sqrt{0.18045}$		$\sqrt{0.18045}$
14				$-\sqrt{0.13436}$	$\sqrt{0.13436}$
15					$\sqrt{0.39536}$
σ	9.19	0.036	0.0104	0.0219	0.0581

For the additive structure it is desirable, see Chapter 3, that there are 15 loci contributing to each growth parameter. This ensures that when genetic selection is applied to the growth parameter, one generation genetic gains match the predicted genetic gains given by infinitesimal theory. However the linear program solution had between 5 and 7 loci for each growth parameter.

Thus the weights for the additive structure from the linear program need to be further divided to give 15 loci for each growth parameter. As the square of the columns values in Table 5.21 sum to one, the square of the values of the figures in the table were multiplied by 15 to give the expected number of loci for each variable in the linear program, Table 5.22. The values were then adjusted to whole numbers to give a total of 15 for each column, giving the values in Table 5.22.

Table 5.22: Expected number of loci (left) and number of loci (right)

Expected number of loci				
Pd_{max}	LP_{min}	E_m	P	w
5.29	5.29			
1.59		1.59		
3.91			3.91	
1.64				1.64
0.88	0.88	0.88		
1.68	1.68			1.68
	0.58	0.58		
	5.54		5.54	
	0.82	0.82		0.82
	0.21		0.21	0.21
		5.11		
		3.32	3.32	
		2.71		2.71
			2.02	2.02
				5.93

Number of loci				
Pd_{max}	LP_{min}	E_m	P	w
4	4			
2		2		
4			4	
2				2
1	1	1		
2	2			2
	1	1		
	5		5	
	1	1		1
	1		1	1
		4		
		3	3	
		3		3
			2	2
				4

The weights for each variable were then split equally over the number of loci to give the additive structure weights in Table 5.23.

Table 5.23: Additive structure weights and standard deviations

Loci	Pd_{max}	LP_{min}	E_m	P	w
1-4	$\sqrt{0.08819}$	$-\sqrt{0.08819}$			
5-6	$\sqrt{0.05306}$		$\sqrt{0.05306}$		
7-10	$\sqrt{0.06522}$			$\sqrt{0.06522}$	
11-12	$\sqrt{0.05455}$				$\sqrt{0.05455}$
13	$\sqrt{0.05888}$	$-\sqrt{0.05888}$	$\sqrt{0.05888}$		
14-15	$\sqrt{0.05613}$	$-\sqrt{0.05613}$			$\sqrt{0.05613}$
16		$-\sqrt{0.03833}$	$\sqrt{0.03833}$		
17-21		$\sqrt{0.07385}$		$\sqrt{0.07385}$	
22		$-\sqrt{0.05434}$	$\sqrt{0.05434}$		$\sqrt{0.05434}$
23		$\sqrt{0.01413}$		$\sqrt{0.01413}$	$-\sqrt{0.01413}$
24-27			$\sqrt{0.08513}$		
28-30			$-\sqrt{0.07379}$	$\sqrt{0.07379}$	
31-33			$\sqrt{0.06015}$		$\sqrt{0.06015}$
34-35				$\sqrt{0.06718}$	$-\sqrt{0.06718}$
36-39					$\sqrt{0.09884}$
σ	14.0	0.0550	0.0159	0.0335	0.0887

This setup of the genetic structure has produced a 39-loci, 6-alleles additive structure which has $6^{2^{39}} = 4.96 \times 10^{60}$ possible allele combinations. The number of possible allele combinations is further discussed in §6.2.

An alternative method to achieve the desired number of loci is to apply a second optimization to minimize difference between the expected number of loci and the number of loci to evenly spread the weights over, as below, where y_{abcde} is the number of loci to evenly spread weight x^2_{abcde} over.

$$\min \sum_{a=0}^{1}\sum_{b=0}^{1}\sum_{c=0}^{1}\sum_{d=0}^{1}\sum_{e=0}^{1}\left|15x_{abcde}^{2} - y_{abcde}\right|$$

s.t.

$$\sum_{a=\begin{cases}1 & j=1\\ 0 & otherwise\end{cases}}^{1}\ \sum_{b=\begin{cases}1 & j=2\\ 0 & otherwise\end{cases}}^{1}\ \sum_{c=\begin{cases}1 & j=3\\ 0 & otherwise\end{cases}}^{1}\ \sum_{d=\begin{cases}1 & j=4\\ 0 & otherwise\end{cases}}^{1}\ \sum_{e=\begin{cases}1 & j=5\\ 0 & otherwise\end{cases}}^{1} y_{abcde} = 15 \qquad for\ all\ j = 1,2,3,4,5$$

$$y_{abcde} \geq x_{abcde}^{2}$$

$$y_{abcde} \qquad integer$$

This optimization gives the results in Table 5.24 below. The results of the optimization differ in from the values used in the simulations throughout Chapters 7 to 10 in five places as highlighted in yellow. Given the results observed in Chapters 7 to 10, that all the loci in a block are optimal when every locus in the block is maximized or every locus in the block is minimized, it is unlikely that the differences would significantly affect the results.

Table 5.24: Number of loci from additional optimization

Number of loci used in simulations				
Pd$_{max}$	***LP$_{min}$***	***E$_m$***	***P***	***w***
4	4			
2		2		
4			4	
2				2
1	1	1		
2	2			2
	1	1		
	5		5	
	1	1		1
	1		1	1
		4		
		3	3	
		3		3
			2	2
				4

Number of loci from additional optimization				
Pd$_{max}$	***LP$_{min}$***	***E$_m$***	***P***	***w***
5	5			
2		2		
4			4	
2				2
1	1	1		
1	1			1
	1	1		
	5		5	
	1	1		1
	1		1	1
		5		
		3	3	
		2		2
			2	2
				6

5.3.7 Conformation of simulation values

To confirm that the model with the weights calculated in §5.3.6 is generating the desired correlations in Table 5.7 and standard deviations in Table 5.6, 65,000 random genomes were simulated and evaluated.

Table 5.25 gives the observed correlations and standard deviations from the 65,000 random genomes for the environmental section of the model compared with the desired values as calculated in §5.3.6. The maximum difference between the desired correlations and the observed correlation is 0.00718. Also all the standard deviations are accurate to two significant figures with some small variations seen at the third significant figure.

Table 5.25: Environmental correlations of growth parameters
top desired correlation, bottom simulated correlations (65,000 genomes)

	Pd_{max}	LP_{min}	E_m	p	w	**SD**
Pd_{max}	1	-0.57616	0.226301	0.253012	0.37185	18.5
LP_{min}		1	-0.15045	0.269126	-0.21974	0.100
E_m			1	-0.19379	0.35602	0.0232
p				1	-0.15922	0.0693
w					1	0.106

	Pd_{max}	LP_{min}	E_m	p	w	**SD**
Pd_{max}	1	-0.57586	0.22274	0.24971	0.37251	18.6
LP_{min}		1	-0.15347	0.27358	-0.22289	0.101
E_m			1	-0.20097	0.35564	0.0233
p				1	-0.16286	0.0696
w					1	0.106

Table 5.26 are presents the observed additive genetic correlations compared with the desired correlation. For this section of the model the maximum difference in the correlations is 0.00681. Again for the standard deviations, differences are not seen until the third significant figure.

Table 5.26: Additive genetic correlations of growth parameters
top desired correlations, bottom simulated correlations (65,000 genomes)

	Pd_{max}	LP_{min}	E_m	p	w	**SD**
Pd_{max}	1	-0.52392	0.164992	0.260875	0.221359	14.0
LP_{min}		1	-0.15155	0.383406	-0.18074	0.0550
E_m			1	-0.22136	0.234787	0.0159
p				1	-0.14849	0.0335
w					1	0.0887

	Pd_{max}	LP_{min}	E_m	p	w	**SD**
Pd_{max}	1	-0.52451	0.17181	0.26162	0.22033	14.1
LP_{min}		1	-0.15391	0.38060	-0.18101	0.0548
E_m			1	-0.21543	0.23386	0.0158
p				1	-0.14730	0.0334
w					1	0.0884

The results for the non-additive genetic correlation are presented in Table 5.27. The maximum difference in the correlations for the non-additive genetic section of the model is 0.00981. Differences in the standard deviations are not seen again until the third significant figure.

Table 5.27: Non-additive genetic correlations of growth parameters
top desired correlations, bottom simulated correlations (65,000 genomes)

	Pd_{max}	LP_{min}	E_m	p	w	**SD**
Pd_{max}	1	-0.52392	0.164992	0.260875	0.221359	9.19
LP_{min}		1	-0.15155	0.383406	-0.18074	0.036
E_m			1	-0.22136	0.234787	0.0104
P				1	-0.14849	0.0219
W					1	0.0581

	Pd_{max}	LP_{min}	E_m	p	w	**SD**
Pd_{max}	1	-0.52386	0.16587	0.26620	0.22715	9.22
LP_{min}		1	-0.14487	0.37359	-0.17798	0.036
E_m			1	-0.22001	0.23405	0.0104
p				1	-0.14437	0.0218
w					1	0.0581

Finally Table 5.28 presents the correlation for the complete model. For the complete model the maximum difference seen in the correlations is 0.00316, which is less than the original significant figures provided in Table 5.7. For the standard deviations and means there are no differences for the significant figures in the original values in Table 5.6.

Table 5.28: Correlations of growth parameters

top original correlations, bottom simulated correlations (65,000 genomes)

	Pd_{max}	LP_{min}	E_m	p	w	**Mean**	**SD**
Pd_{max}	1	-0.55	0.2	0.25	0.3	170	25
LP_{min}		1	-0.15	0.3	-0.2	0.7	0.12
E_m			1	-0.2	0.3	0.485	0.03
p				1	-0.15	0.9	0.08
w					1	5.05	0.15

	Pd_{max}	LP_{min}	E_m	p	w	**Mean**	**SD**
Pd_{max}	1	-0.54684	0.19849	0.25285	0.30207	170	25
LP_{min}		1	-0.15182	0.30164	-0.20224	0.7	0.12
E_m			1	-0.20291	0.30098	0.485	0.03
p				1	-0.15118	0.9	0.08
w					1	5.05	0.15

Overall the model is generating the desired correlations, standard deviations, and means for the five growth parameters. Differences seen are small and do not occur until the third significant figure, where the majority (all but the means of E_m and w) of the desired values are specified to two significant figures.

5.4 Diets

Table 5.29 below delineates the two diets used in the simulation experiments. The extremes of diet were investigated. It is expected that if there is a measurable difference in optimal genetics it will be most visible under extreme conditions.

Table 5.29: Simulation Diets (*AA*, *DP* in g/kg; *DE* in MJ/kg).

	High Quality	**Low Quality**
Lysine	17.3	7.1
Methionine	6.9	2.2
Meth + Cyst	8.8	3.6
Threonine	10.6	4.9
Tryptophane	2.9	1.1
Isoleucine	8.3	4.1
Digestible Energy (*DE*)	14.5	12.8
Digestible Protein (*DP*)	211	113

5.4.1 Model Performance on high and low diets

The means and standard deviations in Table 5.30, correlations in Table 5.32 and matrix plot in Figure 5.8 and Figure 5.9 show the performance of the growth model (produced from 32,000 randomly generated pigs) with the high and low diet conditions.

The factors being considered for correlations are:

- DTS – days to slaughter from a live weight 20kg to a live weight of 90kg
- BF - back fat depth (mm)
- CW – carcass weight (kg)
- FCR – feed conversion ratio
- ADG – average daily gain (kg/day)
- ADFi – average daily feed intake (kg).

The mean performances (ADG, FCR, BF, and ADFi) of the pigs simulated with the model are consistent with experimental results (Cameron, N. D. *et al*., 1999) and values observed on commercial farms.

Means and standard deviation for ADG, FCR and BF in Large White, Landrace, Composite and Duroc pigs tested at the Swiss National Pig Breeding Centre (MLP) in 2006 and 2007 (SUISAG, 2006 and 2007) are presented in Table 5.31. At the MLP the pigs are growth between 30 kg and 103 kg and fed *ad libitum* 80kg of a grower diet (13.3MJ DE per kg and 8.5g Lysine per kg) then a finisher diet (12.90MJ DE per kg and

7.86g Lysine per kg). These two diets are closer to the low diet situation (12.8 MJ DE per kg and 7.1g lysine per kg) than the high diet situation (14.5MJ DE per kg and 17.3g lysine per kg). The coefficient of variation of ADG, FCR and BF obtained in the simulated populations with the two diets are in the range of those measured at the Swiss National Pig Breeding Centre.

From the correlations in Table 5.32 it is clear that the addition of the genetic component to the model has not affected consistency. Currently, as there is very limited information on correlations between Pd_{max}, LP_{min}, E_m, p and w assumptions had to be made (see 5.3.2). The phenotypic correlations generated by the model, base on those assumptions, are in the range or very close to the values published in the literature for *ad libitum* fed pigs. This means that the assumptions made were reasonable. However, if specific phenotypic correlations between performance traits have to be generated, the correlation matrix between the model inputs parameters Pd_{max}, LP_{min}, E_m, p and w (Table 5.17) can be adapted.

Further performance results can be found in Appendix B and Appendix C for the high and low diet, respectively.

Table 5.30: Model Means and Standard Deviations for high and low diet (32,000 pigs with randomly generated genes)

	High Diet			**Low Diet**		
	Mean	**Std Dev**	**CV**	**Mean**	**Std Dev**	**CV**
DTS	77.59	10.41	13.4	77.28	8.87	11.5
BF	12.29	2.56	20.8	14.64	2.30	15.7
CW	68.50	0.369	0.5	68.78	0.361	0.5
FCR	2.175	0.207	9.5	2.419	0.172	7.1
ADG	0.9241	0.119	12.9	0.9241	0.103	11.1
ADFi	1.993	0.182	9.1	2.226	0.203	9.1

Table 5.31: Means and standard deviation for Swiss National Pig Breeding Centre

		ADG		FCR		BF	
		2006	**2007**	**2006**	**2007**	**2006**	**2007**
Largewhite	**Mean**	0.855	0.844	2.62	2.61	17	18
	Std Dev	0.094	0.095	0.22	0.2	3	3
Landrace	**Mean**	0.857	0.866	2.63	2.62	17	16
	Std Dev	0.1	0.096	0.23	0.24	4	3
Composite	**Mean**	0.87	0.862	2.55	2.52	16	16
	Std Dev	0.103	0.1	0.19	0.2	3	3
Duroc	**Mean**	0.897	0.877	2.61	2.63	17	18
	Std Dev	0.094	0.125	0.19	0.2	3	3
CV Range	**min**	10.5	11.1	7.3	7.6	17.6	16.7
	max	11.8	14.3	8.7	9.2	23.5	18.8

Table 5.32: Model Correlations for high and low diet compared with values published in the literature under *ad libitum* feeding (32,000 pigs with randomly generated genes)

		Diet		Literature[1]	
		High	**Low**	**Min**	**Max**
ADG	**ADFi**	0.694	0.789	0.43	0.79
ADG	**FCR**	-0.704	-0.574	-0.4	-0.55
ADG	**BF**	-0.151	0.091	-0.05	0.27
ADFi	**FCR**	0.009	0.042	-0.04	0.61
ADFi	**BF**	0.566	0.642	0.12	0.56
FCR	**BF**	0.763	0.708	0.02	0.56
DTS	**BF**	0.141	-0.09	-	-
DTS	**CW**	-0.005	-0.177	-	-
DTS	**FCR**	0.709	0.583	-	-
DTS	**ADG**	-0.981	-0.985	-	-
DTS	**ADFi**	-0.688	-0.777	-	-
CW	**BF**	0.745	0.694	-	-
CW	**FCR**	0.486	0.394	-	-
CW	**ADG**	0.019	0.205	-	-
CW	**ADFi**	0.524	0.546	-	-

[1]Range of phenotypic correlations for Large white, Landrace and Pietrain in Germany, England and Switzerland from Wafler (1982), Blum (1983), Krieter (1986), Schmidt (1988), Morel (1988) and Cameron and Curan (1994).

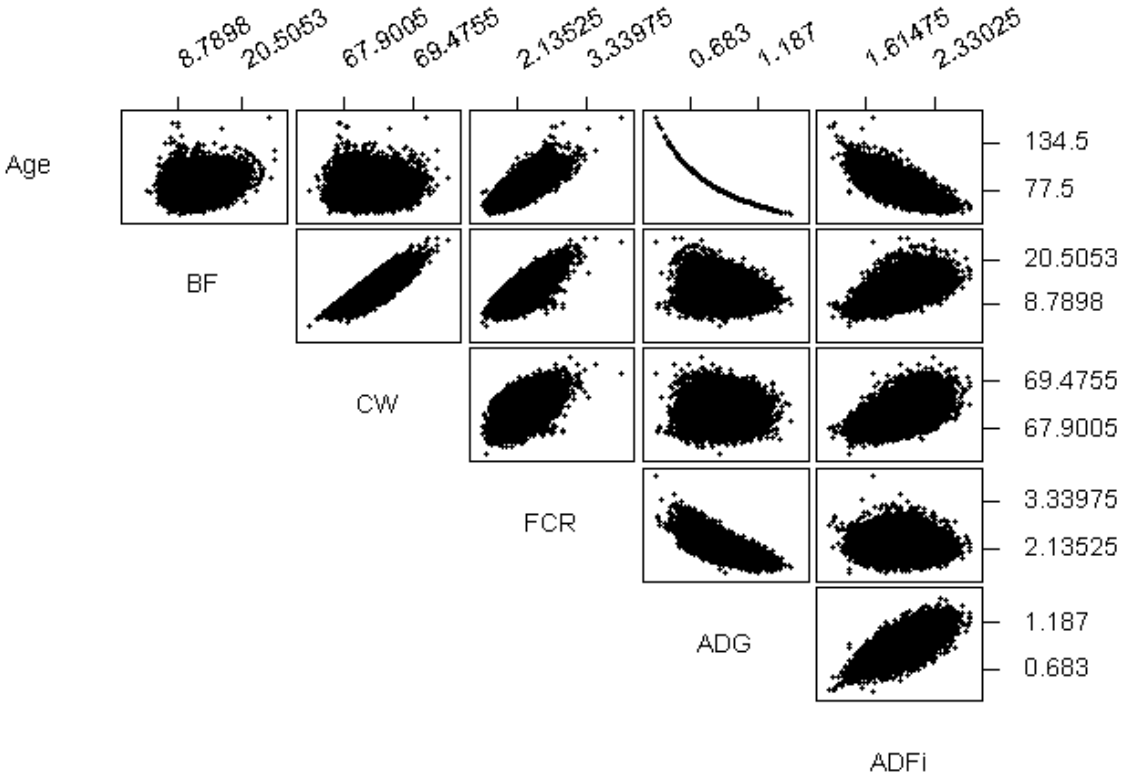

Figure 5.8: Matrix scatter plot – high diet
32,000 pigs with randomly generated genes.

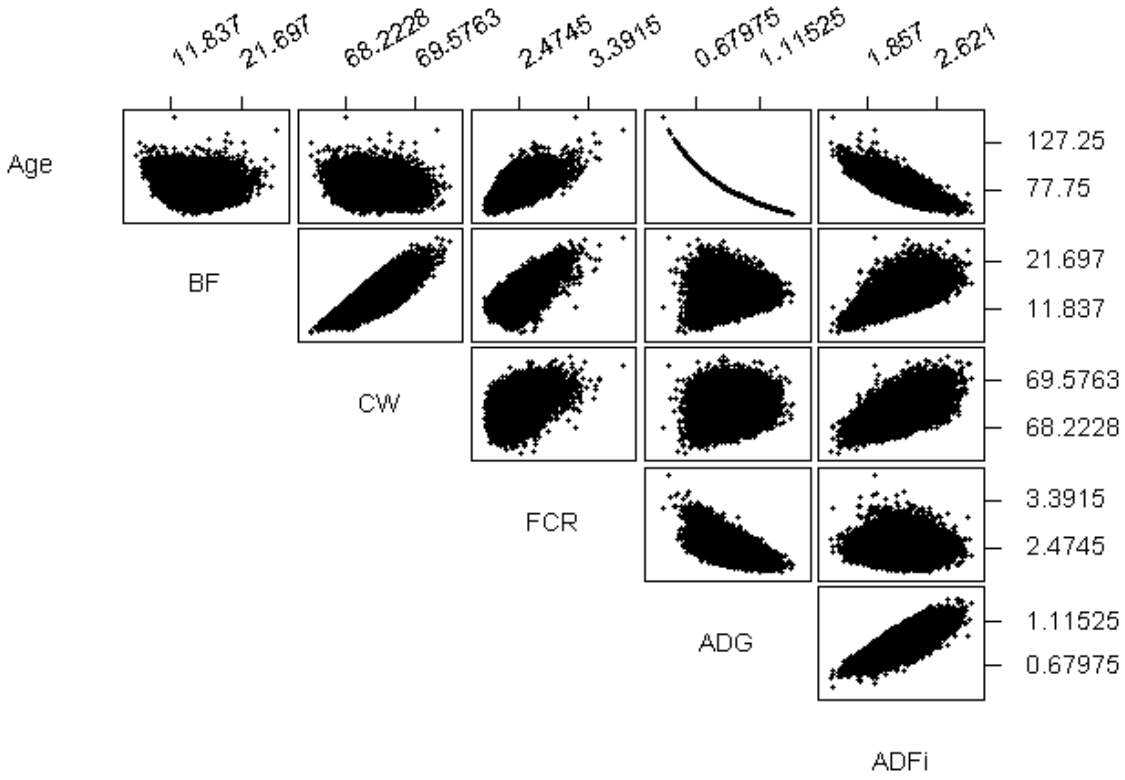

Figure 5.9: Matrix scatter plot – low diet
32,000 pigs with randomly generated genes.

5.5 Summary

This chapter is concerned with the description of the GE-Pig model. Over time the knowledge about genes, and their influences in the genotype performance, is increasing. Soon there will be a need for ways to incorporate this information into various breeding methods and models to take advantage of this new knowledge. One new area of information is concerned with the additive effects of alleles at various loci. The GE-Pig model has been developed to take these additive effects of alleles and combine them with a pig growth model. This allows for the performance of various combinations of the alleles to be predicted. The GE-Pig model gives the capability for optimization algorithms to be applied, thus allowing optimal combination of alleles for any given objective to be found.

Chapter 6 – Search algorithms description

6.1 Introduction

This chapter discusses the three algorithms, Genetic Algorithm, Tabu Search, and Simulated Annealing, used to optimize the extended pig growth model as described in Chapter 5. Each algorithm is described in depth with the parameters and their setup for each method discussed. A general heuristic description is given for each method, with the specific heuristics included as appendices D to F. A discussion is included on comparative aspects of the different heuristics. These three algorithms have been implemented to optimize the model as they each take different approaches to optimization and each have different advantages and weaknesses. These algorithms are also the three leading stochastic optimization methods. The results in the following Chapters have also been compared to a simple Monte Carlo run, or a pure random search as a baseline comparison.

Stochastic optimization methods have been implemented to optimize the model as for each combination of the genetic setup there is a distribution of possible model results due to the environmental and non-additive effects being randomly generated each time the combination is evaluated. It was also desired to maintain the model has a whole and not approximate the results of the model with analytical expressions which would have allowed deterministic methods to be used.

6.2 Domain size

This setup of the genetic structure has produced a 39-loci, 6-alleles additive structure which has $6^{2^{39}} = 4.96 \times 10^{60}$ possible allele combinations. That is there are two alleles present at each locus, each of which can be one of six alleles, giving $6^2 = 36$ possible allele combinations at each locus, and as there are 39 loci there is in total $36^{39} = 4.96 \times 10^{60}$ possible allele combinations.

However as the allele structure is additive in nature it can be noted that, for example, a locus with alleles A1A2 has the same evaluated value as the same locus with alleles A2A1 and therefore would be observed as the same. Thus the $6^2 = 36$ possible allele

combinations at each locus have only 11 observable values. Therefore the domain size over the 39 loci could be reduced to $11^{39} = 4.11 \times 10^{40}$.

It can be further noted that there are groups of loci which have the same effects that could be used to further reduce the domain. These loci can be grouped together as the loci in the group have the same variances and influences in the growth parameters. These groups and their total number of observable values are listed in Table 6.1. When these loci groups are taken into account, the domain size can be reduced to 3.94×10^{20}, being the product of the observable values.

Table 6.1: Loci groups and number of observable values.

Loci	Observable values
1-4	41
5-6	21
7-10	41
11-12	21
13	11
14-15	21
16	11
17-21	51
22	11
23	11
24-27	41
28-30	31
31-33	31
34-35	21
36-39	41

6.3 Genetic Algorithm

6.3.1 Genetic Algorithm description

The Genetic Algorithm uses a move structure that is based on what happens to the DNA strings when breeding occurs. The process starts with a base population and ‘breeds’ various members of the population together to generate an offspring population. The

new base population is then selected from the offspring population and the procedure begins again.

Genetic algorithms have been used extensively in a wide variety of areas. Some examples of their use in the animal growth modelling area include maximising gross margin in intensive livestock production for the pig and poultry industry (Morel *et. al*., 2001), optimize the nutrient use of the diets and minimize nitrogen excretion in pig meat production systems (Morel and Wood, 2005). They have also been used to optimize pastoral dairying scenarios that incorporates sub-models of pasture growth and animal metabolism (Liu *et. al*., 2003).

The additive structure, incorporating the two alleles that are present at each of the 39 loci of the model, is used to describe each member of the base population. Different orderings of the same alleles at a locus have been treated as different. This results in the Genetic Algorithm having a domain of size $6^{2^{39}} = 4.96 \times 10^{60}$ for the setup of the model being investigated.

A genetic algorithm would run more efficiently on the reduced domains discussed in §6.2, however in this case the larger domain was used to limit the number of data structures required and to allow for easy changing of the input data. The relatively small number of loci in the setup of the model used also allowed the larger domain to be considered. If the number of loci became too large then the reduction of the domain would have to be considered.

The breeding procedure in the Genetic Algorithm takes two members of the base population as parents and combines the additive structures from each parent together to create an offspring, as shown in Figure 6.1.

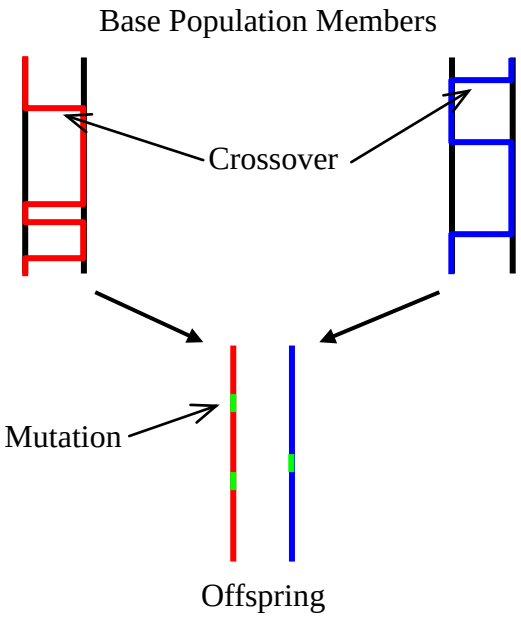

Figure 6.1: Illustration of breeding procedure.

For each of the two selected parents, the first or second allele at the first locus is selected randomly with equal probability. For each consecutive locus the same allele is selected unless a crossover occurs.

Crossover occurs at each locus with probability *C*. When crossover occurs the allele that is being selected switches, first to second or second to first. The selected alleles at each locus for each of the two base population members are then passed to the offspring and mutation is applied.

Mutation occurs at each allele of the offspring with probability *M*. When mutation occurs it is equally likely that that allele value is increased or decreased. When an allele value is increased/(decreased), it is increased/(decreased) to the next possible allele value. If the allele value is at it's maximum/(minimum) value then no mutation occurs at that allele.

The number of crossovers that occur when selecting from each parent follows a binomial distribution with probability *C* and number of trials being the number of loci in the system less one. The average number of crossovers for the setup being investigated is (number of loci – 1)*C*, which for the model parameters under investigation is 38*C*.

The number of mutations that occur in an offspring when the parents are randomly generated follows a binomial distribution with probability of success being $\frac{5}{6}M$ when there are 6 alleles at each locus. The number of trials is twice the number of loci in the system. For the setup of the model being investigated, the average number of mutations in a randomly generated offspring is $65M$.

The Genetic Algorithm was programmed with a base population size of N, which is initialized with the N best of $N(N-1)/2+1$ randomly generated members. Breeding is applied to all possible pairings of base population members, with one offspring generated from each pairing. This results in the creation of $N(N-1)/2$ pairings and offspring. The offspring are then supplemented by the member of the base population that has been evaluated as having the best objective value, thus giving an offspring population size of $N(N-1)/2+1$. The N offspring that are evaluated as having the best N objective values are then selected as the new base population and the procedure repeats. The stopping 'criteria' for the algorithm has been set to 'no change in the best evaluated for a pre-selected number, S, of iterations'.

6.3.2 Parameter explanation

The parameters in this formulation of the Genetic Algorithm are:

N	The number of members in the base population
C	Probability of a crossover
M	Probability of a mutation
S	Number of iterations for which the best solution remains unchanged before the algorithm stops.

Setting the value of N is a trade off between the running time of the algorithm and the amount of diversity investigated and retained at each iteration.

It is possible for C to take any value between zero and one. When $C = 0$, no crossover occurs. This results in the second generation of offspring receiving information from only two of the four parents in the initial base population as only one side of the structure is passed to the offspring.

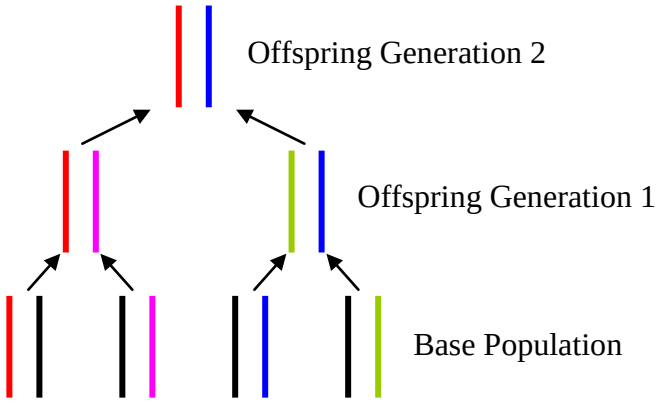

Figure 6.2: Illustration of crossover when $C = 0$.

When $C = 1$, crossover occurs between every loci, resulting in 38 crossovers for each parent for the setup being investigated. This results in the second generation of offspring receiving a quarter of its information from each of the four parents in the initial base population.

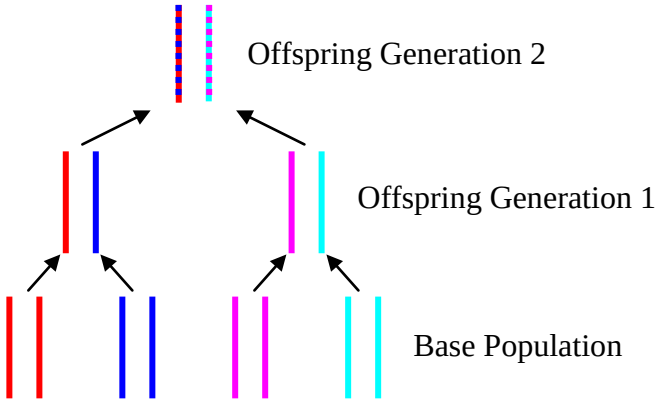

Figure 6.3: Illustration of crossover when $C = 1$.

The Value for M can be any value between zero and one. The value of one represents mutation of both alleles at every locus. A zero value is indicative of no mutation at all. The inclusion of mutation broadens the possible range of investigation by increasing the allele population.

Setting the value of S is a trade off between the running time of the algorithm and the algorithm terminating before it has found a quality solution.

6.3.3 Parameter setup

The crossover rate was set at C = 0.25, which has an average of 9.5 crossovers per parent.

The mutation rate was set at M = 0.1, which has an average of 6.5 mutations in an offspring that has randomly generated parents.

The following combinations, in Table 6.2, of stopping times, *S*, and base populations sizes, *N*, were considered.

Table 6.2: Stopping time and base population size combinations.

N* \ *S	5	10
10	✓	✓
30	✓	✓
50	✓	✓

6.3.4 Genetic Algorithm heuristic description

The genetic algorithm heuristic requires the input of *N*, the number of genomes to carry from one iteration to the next, the mutation probability, *M*, and the crossover probability, *C*. Initialization consists of generating $N(N-1)/2+1$ randomly generated genomes, selecting the *N* best genomes to be the current population, and setting the best genome to be the best from the current population.

The heuristic then starts a breeding process in which all possible pairs in the current population form the offspring population. The best genome is then added to the offspring population. The *N* best from this offspring population then become the current population and the best genome becomes the best genome in the current population. This breeding process is repeated either for a set number of iterations, or until there has been no improvement in the best solution for a set number of iterations. The heuristic then outputs the best solution genome.

The breeding process above, for each parent, uses a binary variable which controls allele selection, at each locus, to be passed to the offspring. A change in the binary

variable is tested for between loci and is trigged by the crossover probability. Mutation of each selected allele occurs with the probability of the mutation probability.

A pseudo code version of the genetic algorithm heuristic may be found in Appendix D.

6.4 Tabu Search

6.4.1 Tabu Search description

A Tabu Search is a meta-heuristic that guides a local heuristic search procedure to explore the solution space beyond local optimality. The definition of the neighbourhood of any given solution is an important part of the Tabu Search, as is the use of short term and long term memory, which creates a more flexible search behaviour. In general, memory based strategies are the hallmark of Tabu Search approaches. Tabu Searches can be very intricate with a great deal of embedding of different search procedures with different neighbourhood definitions. This particular Tabu Search heuristic is based on ideas as delineated by Glover & Laguna (1997), and has a relatively simple profile.

One of the mainstays of the Tabu Search meta-heuristic is the Tabu list. This list delineates the moves that the search procedures declare illegal or tabu. Moves declared tabu are generally on the list for a specified time and are then removed from the list.

This Tabu Search has a single neighbourhood definition §6.4.2, the same as that used in the Simulated Annealing Algorithm in §6.5, but uses path search ideas to force the solution to move away form the current location until either a better solution is found or the path search is abandoned. The algorithm calls on four internal search procedures, a line search, a path search, diversify search and a local search. For simplicity, the order of the four procedures is predetermined.

The algorithm begins with generating a set of the initial elite solutions by starting with randomly generated genomes which are line searched, as in §6.4.3, until there is no improvement. The procedure then repeats a diversify search, which is a local search with added constraints on the search area as in §6.4.6, for each elite solution, and path searches, as in §6.4.4, between the elite solutions, until there is no improvement in the best solution. Each elite solution then undergoes a local search, as described in §6.4.5. The best solution found is then reported.

6.4.2 Neighbour Generation

Each of the search procedures uses a neighbour generation process. A neighbour of a current solution is generated by changing each of the alleles at each locus with a random probability. The change will occur only if it is not tabu. The change is limited to the next or previous allele in the allele list.

For example, if the current solution has alleles 2 and 6 at locus 1, and alleles 3 and 5 at locus 2 as in Table 6.3, then two of the possible neighbours are shown.

Table 6.3: Examples of neighbours.

	Locus 1		**Locus 2**	
	Allele 1	Allele 2	Allele 1	Allele 2
Current Solution	2	6	3	5
Neighbour 1	3	6	2	6
Neighbour 2	2	5	3	4

6.4.3 Line Search

The line search procedure begins by generating neighbours around the starting point. The best of the generated neighbours is then selected. The direction of the line search is then defined as the difference between the selected neighbour and the starting point. This direction is not a distance but a string of values that give a measure of how the alleles changed at each locus, as can been seen in the example in Table 6.4.

Table 6.4: Example of direction calculation and a direction neighbour.

	Locus 1		**Locus 2**	
	Allele 1	Allele 2	Allele 1	Allele 2
Current Solution	2	6	3	5
Best Neighbour	3	6	2	6
Direction	1	0	-1	1

The line search procedure then adds the direction to the new current solution and any alleles that fall outside the allele range are brought back within the range. The new direction is then calculated and the process repeated as shown in Figure 6.4. The process is repeated until the there is no change in the current solution.

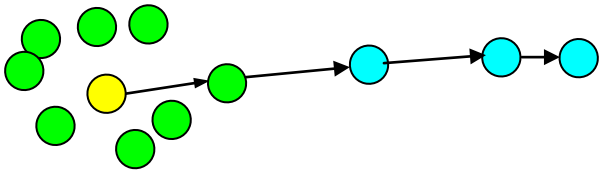

Figure 6.4: Pictorial view of line search procedure.

6.4.4 Path Search

The path search requires a given starting solution and destination solution. The search procedure then declares tabu any allele move that will move an allele away from the destination allele at each locus, as covered in the example in Table 6.5. This forces the search to move overall in the direction of the destination solution. Neighbours of the current solution are generated, the best is selected and the tabu list updated. The process is then repeated until the stopping criteria are met, as shown in Figure 6.5. The stopping criteria for this search are stop if the destination solution is reached or if there is improvement in the solution for *S* iterations.

Table 6.5: Example of tabu moves.

	Locus 1		**Locus 2**	
	Allele 1	Allele 2	Allele 1	Allele 2
Current Solution	2	5	3	5
Destination Solution	6	2	1	6
Tabu	decrease	increase	increase	decrease

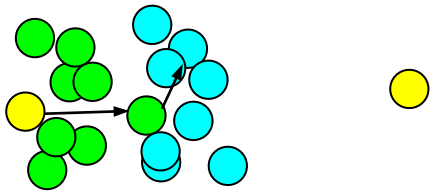

Figure 6.5: Pictorial view of the path search procedure.

6.4.5 Local Search

The local search procedure generates and evaluates the neighbours of the current solution. It then moves to the best neighbour and make it tabu for a set time period to undo any part of that move, as shown in Figure 6.6. This process is repeated until the stopping criteria are met.

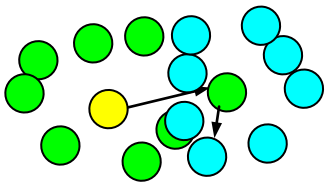

Figure 6.6: Pictorial view of the local search procedure.

6.4.6 Diversify Search

The diversify search was added to the tabu search algorithm when it became clear, during program testing, that it was necessary to improve the range of solutions under consideration. The search procedures above tend to gravitate toward the centre of the solution plane. This search procedure was added to force the solutions toward the extremes of the solution plane.

The search starts by declaring tabu any move in the alleles that will move them toward the centre of the allele range, as can be seen in Table 6.6. It then generates neighbours and moves to the best found. It then updates the tabu list and repeats until the stopping criteria are met. The stopping criterion for this search is stop if there is no improvement for *S* iterations.

Table 6.6: Tabu moves for diversify search.

Allele	Tabu
1	increase
2	increase
3	
4	
5	decrease
6	decrease

6.4.7 Parameter setup

The neighbour generation probability was set to 0.1. All the remaining variables were investigated at two levels. The number of elite solutions, *E*, was set at either 3 or 5. The number of neighbours, *N*, was at either 10 or 30 and the stopping criteria, *S*, was either 5 or 10 iterations. These configurations are summarised in Table 6.7 below.

Table 6.7: Parameter setups investigated for Tabu Search.

***E* = 3**		
N* \ *S	5	10
10	✓	✓
30	✓	✓

***E* = 5**		
N* \ *S	5	10
10	✓	✓
30	✓	✓

6.4.8 Tabu Search heuristic description

This heuristic begins by inputting the number of elite solutions and the number of neighbours and setting the neighbourhood probability. Then for each elite solution the current genome is randomly generated and a line search is repeated until there is no improvement on the current solution. The elite genome is then set to the current genome.

The heuristic then repeatedly performs a local search, with diversify enabled, for each elite solution, and a path search between the elite solutions, until there is no improvement. A local search, with diversify disabled, follows for each of the elite solutions. The best genome is then reported as the solution found.

The line search sets both the current best genome and the line best genome to equal the provided genome value. Neighbours of the current genome are generated and the best is selected. A direction is then defined as the difference between the current genome gene structure and the best neighbour genome gene structure. The search then repeatedly sets the new genome to the current genome plus the direction, adjust the alleles back into the allele range if required, sets line best genome to the new genome if the new genome is better, and then recalculates the direction and sets the current genome to the new genome. This process continues until the direction is zero. The output from the line search is the line best genome.

The local search sets the local best genome and the current genome to the input genome. If the diversify option has been specified, the tabu list is set up as specified in §6.4.6 with infinite tabu tenure. The local search then repeats the following actions until the stopping criteria is met:

Generate n neighbours and select the best as the best neighbour

Update the local best genome if the best neighbour is better

Calculate the direction

Set the current genome to the best genome.

It is then made tabu to undo the direction of movement for a set number of iterations. The algorithm returns the local best genome.

The path search starts by generating elite pairs so that each elite is a source once and a destination once, and no destination elite of the current destination elite is the current elite. A path search is then performed for each pair. The destination genome is set to the destination of the pair and the current genome is set to the source of the pair. The path search, repeated until the stopping criteria is met when applied to the current genome, then makes tabu any move that will increase the distance between the current genome and the destination genome, generates neighbours, selects the best and moves to the best selected if this is a better solution. At the end of each search, the new elite is set to the current genome. After completing the path search for each pair, the new elite solutions are returned.

The neighbour generation takes the current genome and at each locus and increases or decreases the alleles by one with probability of the neighbourhood probability. Any alleles that are out of range after being changed are brought back within the valid range.

A pseudo code version of the Tabu Search heuristic may be found in Appendix E.

6.5 Simulated Annealing

6.5.1 Simulated Annealing description

Simulated Annealing is a derivative of the primitive decent search. Simulated annealing is a local search algorithm that generates neighbours and moves to the best neighbour with a probability that is dependent on the current temperature of the system. If the best

neighbour is better than the current solution it is selected with a probability of one, otherwise it is selected with probability

$$e^{\frac{best\_neighbor - current}{temperature}}.$$

The temperature of the algorithm follows a preset pattern and generally decreases with time. This algorithm was developed from the industrial idea of heating and cooling metal to obtain the optimal internal structure for strength. Simulated annealing is effectively a type of descent search that has the ability to climb upwards on the search hyper-plane.

This implementation generates neighbours in the same way as the Tabu search algorithm in §6.4.2. The temperature in this implementation cools to $(\text{temperature} \times \text{cool\_factor})$ at regular intervals.

6.5.2 Parameter setup

The cooling factor was set to 0.9 with a starting temperature of 1 and an ending temperature of 0.01 or 0.05. The number of neighbours and the number of iterations between cooling were set to various values, as outlined in Table 6.8.

Table 6.8: Parameter setups investigated for Simulated Annealing.

***S* = 0.01**				
N* \ *T	5	10	30	50
10	✓	✓	✓	
30	✓	✓		✓
50	✓	✓	✓	

***S* = 0.05**		
N* \ *T	5	10
10	✓	✓
30	✓	✓
50	✓	✓

The two configurations that are highlighted in yellow have the same number of genomes evaluated, giving the ability to investigate the interaction of number of neighbours and number of iterations between cooling. The same is true for the two configurations highlighted in green.

6.5.3 Simulated Annealing heuristic description

The simulated annealing algorithm begins with the input of the number of neighbours, maximum number of iterations spent at each temperature, stopping temperature, neighbour probability, start temperature, and cooling factor. The current genome is initialised to a randomly generated genome. The best genome is then set to the current genome, the temperature of the algorithm is set to the start temperature and the cool time is set to the number of iterations spent at each temperature. The algorithm then repeats the following until the stopping temperature is reached:

Generate neighbours of the current genome and select the best neighbour

If the best neighbour is better than the current genome then set the current genome to the best neighbour

If the current genome is better than the best neighbour then set the current genome to the best neighbour with probability $e^{\frac{best\_neighbor - current}{temperature}}$

If the current genome is better than the best genome then set the best genome to the current genome

Decrease the cool time by one

If the cool time is zero, decrease the temperature by the cooling factor and reset the cool time to the number of iterations spent at each temperature

The neighbour generation takes the current genome and at each locus and increases or decreases the alleles by one with probability of the neighbourhood probability. Any alleles that are out of range after being changed are brought back within the valid range.

A pseudo code version of the Simulated Annealing heuristic may be found in Appendix F.

6.6 Discussion of aspects of the algorithms

Genetic Algorithms are known for their very good exploration abilities and they can be powerful and efficient global optimizers (Leung and Wang, 2001, Mitchell, 2001 and Sarker et al., 2002). However if the exploration is not balanced out by exploiting results produced by the search, it can lead to excessive computational expense. Also if results are over exploited the search is in danger of premature convergence, or of turning simply into a local optimizer. Keeping the balance between the two and preserving the

selection pressure relatively constant through the whole run is important characteristic of any Genetic Algorithm technique (Mitchell, 2001 and Ali et al., 2005). Other problems associated with Genetic Algorithms are their relatively slow convergence and low accuracy of the found solutions (Yao et al., 1999 and Ali et al., 2005). The Genetic Algorithm convergence depends on a number of factors, initial conditions, selection mechanism, choice of reproductive operators, stopping conditions (Rudolph, 1994).

The Genetic algorithm implemented here has four inputs, the parent population size, the stopping criteria, probability of crossover, and probability of mutation. This implementation starts with a large set $N(N-1)/2+1$ of randomly generated genomes, from which the *N* best are chosen as the parent population. The time that it will take for the algorithm cannot be predicted as the number of iteration performed is dependant of the results of the search. However the time it takes to perform an iteration can be predicted with some certainty, once the evaluation time for a single genome is known. This is due to their being a set number of genomes evaluated each iteration.

Simulated Annealing is known for being able to escape local optima. It can be shown under certain conditions Simulated Annealing will escape all local optima and will find the global optimum (Hajek, 1988, Belisle, 1992, and Locatelli, 2001). This is achieved by the use of a temperature variable that allows the algorithm to accept, with a probability, a solution that is worse than the current solution and in effect climb out of local optima.

The Simulated Annealing algorithm implemented here starts with a single randomly generated genome and proceeds from that point through neighbours of the genomes. This implementation has six inputs, number of neighbours, neighbourhood probability, stopping criteria, start temperature, cooling factor, and time at each temperature. The last three control the temperature of the algorithm. The running time for this implementation can be predicted with some certainty, once the evaluation time for a single genome is known. This prediction can be made because for any particular setup of the algorithm there are a known number of genomes evaluated.

Tabu Search can be a very powerful and in a variety of problem settings had found solutions equal or superior to the best previously obtained by alternative methods

(Fouskakis and Draper, 2002). The major drawback of Tabu Search is the large number of potentially crucial user-defined choices that need to be made in implementing the method. Also there is little advice in the literature about how to make these choices.

The Tabu Search implemented here starts with a randomly generated genome for each elite solution, giving a set of solutions as a starting point. It has four main inputs, namely the number of elite solutions, number of neighbours, neighbourhood probability, and stopping criteria. However, the order of the subroutines performed by the Tabu search and the period of time for which moves are taboo, also affect the algorithms efficiency. This implantation could get caught in local optima, but the path search between elite solutions reduces the chances of this by forcing the movement of an elite solution towards another elite. This implementation also has a changing tabu list that prohibits the heuristic from retracing previous steps. Predicting the running time of the Tabu Search algorithm is not possible because of the complexity of the sub-algorithm structures and the unknown number of iterations that these sub-algorithms will perform.

Each of the algorithms implemented have different starting points, with the Genetic Algorithm with a possible advantage on the start conditions, with a multiple start set and ‘best’ criteria already having been applied. They each employ different methods in an attempt to escape local optima so that the global optima can be found. The inputs to the algorithms vary significantly across the algorithms, however Tabu Search and Simulated Annealing do share the inputs relating to the neighbourhood structure. Only Simulated Annealing’s total running time can be predicted with any certainty.

6.7 Summary

This chapter has discussed the algorithms used in the process to find the best genetic string for the model described in Chapter 5. The discussion and comparison of the actual simulation results is contained in Chapters 7, 8, 9, and 10.

The three methods of investigation were chosen to give a range of approaches to the problem. The Tabu search has a forced movement around the objective plane, Simulated Annealing has a local search movement and the Generic Algorithm makes use of the biological breeding ideas on which the problem is based.

It would appear that the Genetic Algorithm heuristic is a better theoretical fit for this application than either of the other two heuristics with a natural match in structure. The Tabu Search can be a very powerful tool because of the range in possible sub-search options, but most of these searches require some form of movement structure which is not well defined in this application. The main problem with Simulated Annealing is that again there is no well defined movement structure.

Chapter 7 – Average daily gain objective

7.1 Introduction

This chapter presents the results from the optimization algorithms when applied to the GE-Pig model for the first objective investigated, to maximize average daily gain. Each run of the algorithms reports the best found combination of alleles for the additive genetic structure for the given objective. Firstly the objective is described, followed by a description of the graphs presented throughout the Chapter. The results of the algorithm runs are then presented, broken down into two sections, high diet and low diet. These diet sections are further broken down by algorithm. The results for the two diets are compared and the theoretical minimum number of generations to the optimal solution is calculated. Throughout, the results are interpreted for the practical farming situation.

7.2 Objective description

One of the many factors that pig farmers wish to optimize or improve is Average Daily Gain (ADG). The higher ADG the more body weight the pigs gain each day and the quicker they get to target weight, meaning more pigs can be grown within a year. ADG is defined as the average weight the pig puts on each day over the growth period. For the purposes of this optimization the ADG is calculated per pig simply by dividing the total weight gain over the number of days to grow the pig. For the maximization process the starting weight is set to 20kg and the pigs are grown to 90kg, so that the only variable is the number of day of growth. One thousand pigs are grown to evaluate the objective. The resultant pigs however will have varying carcass weights, *CW*, as carcass weight is dependant on live weight, *LW*, and back fat, *BF*, as per the following equation

$$CW = 0.01LW\left(65.9 + 0.092LW + 0.12BF\right)$$

7.3 Description of graph structures

There are two types of graphs used in this Chapter and the following Chapters to present the results of the simulations. The first graph type is a standard 2-dimensional graph presenting the performance of the 20 repeated simulation runs for each setup of the algorithm parameters investigated. Also graphed is the result of a Monte Carlo run, or a

pure random search, which was set to search though 80,000 random generated genomes, for a baseline comparison, and the mean of the population. The y-axis is the best value found by the algorithm for each repeat of the algorithm setup and the x-axis plots the number of genomes the algorithm investigated to find the best solution reported. This setup for the axes means that the x-axis is a measure of time taken (time itself was not used as the simulations were carried out on more than one computer with different processing speeds) and the y-axis is a measure of the algorithm's performance. Using the number of solutions investigated as a measure of time is fairly accurate as the time to evaluate a solution significantly outweighs the time it takes the algorithms to determine which solutions to evaluate. This is due the solution evaluation involving an iterative process of growing a group of 1000 pigs from a start weight of 20kg to an end weight of 90kg on a day by day basis.

The other type of graph in this and the following Chapters was designed specifically for this work to give a visual representation of the solutions produced by the optimization procedures. An Excel 2-dimensional contour plot graph has been used with the x-axis being the loci number, the y-axis being the simulation run number and the z-axis being the combined value of the alleles (when the alleles are mapped to the range [0, 5], i.e. numerically ordered from 0 to 5 by their value, and when summed a range of [0, 10]) present at each locus in the solution. Each setup of the algorithms was repeated 20 times to allow the consistency of the algorithms performance to be observed and to allow for the observation of multiple optimal solutions. This means that each graph consists of 20 horizontal sets of solutions, each a final solution presented by the optimization algorithm for the particular setup of the algorithm being graphed. This gives a coloured graph like a coloured weather map and allows for visual vertical comparison of each locus over the simulation runs.

Figure 7.1 and Figure 7.2 give a general indication as to how results will look with one solution repeated 20 times (repeatability value of 1.000) representing the ideal situation of the same optimal solution being generated by every run of the algorithm, and one solution repeated 20 times with random variation added (repeatability value of 0.948). Figure 7.3 depicts 20 randomly generated solutions (repeatability of 0.014). The repeatability values have been calculated as

$$r = \frac{\frac{MSA - MSE}{K}}{\frac{MSA - MSE}{K} + MSE}$$

where K is the number of loci in the model, i.e. 39, MSA is the mean square between the alleles, and MSE is the mean square within the alleles. Repeatability is a measure of the consistency of the results from each setup of the heuristic, where a value of 1 is complete consistency and a value of 0 is no consistency.

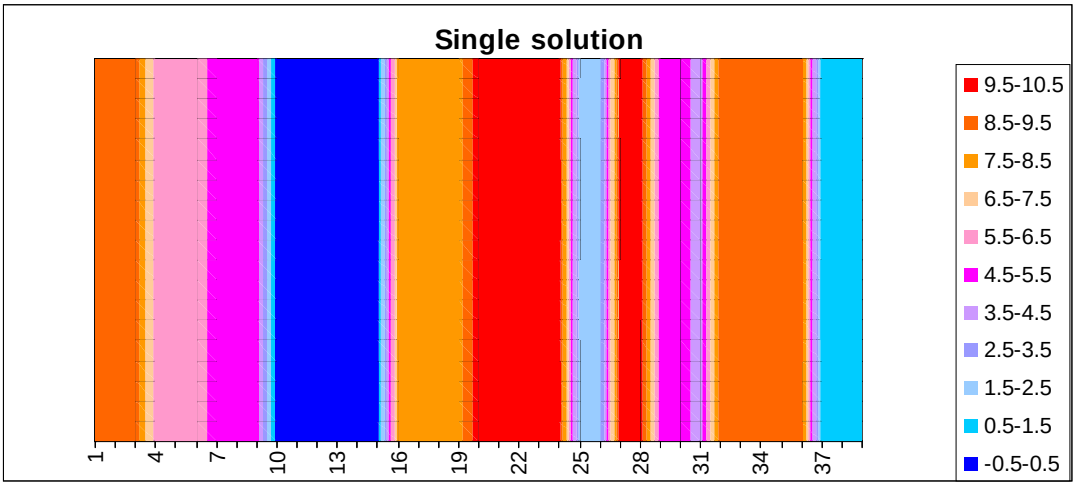

Figure 7.1: Same solution for every run, repeatability 1.000.

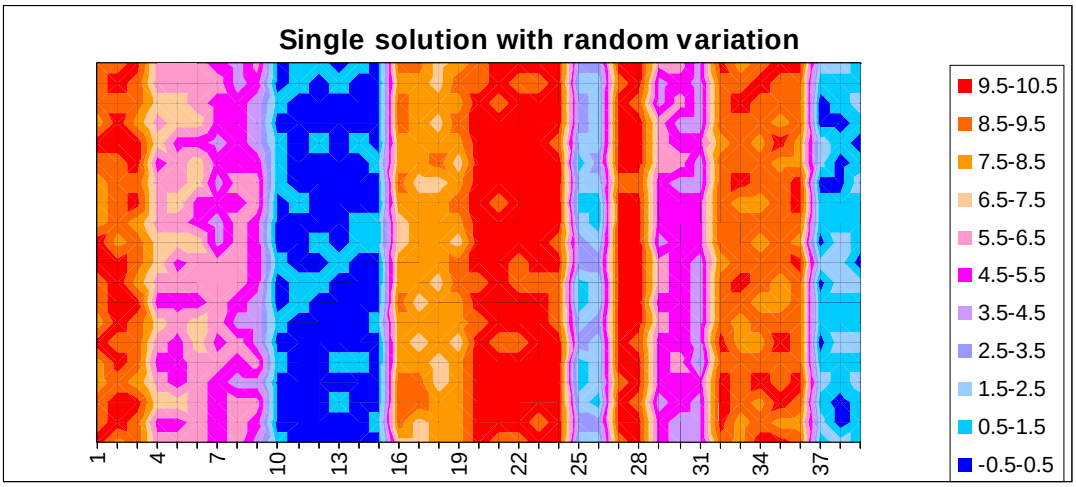

Figure 7.2: Same solution with random variation added, repeatability 0.948.

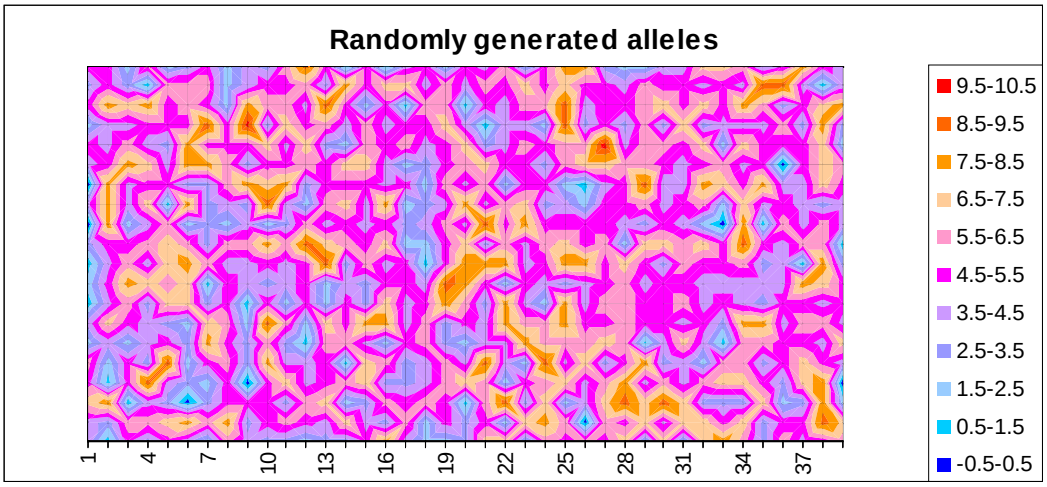

Figure 7.3: Randomly generated alleles, repeatability 0.014.

It is also possible for there to be multiple optimal solutions, representing two or more completely different genetic optimal solutions. This situation would require further investigation to determine whether there is actually a significant difference in. Figure 7.4 depicts a situation where there are two solutions generated by the algorithm, with a repeatability of 0.355, and Figure 7.5 shows the same situation with added random variation, with a repeatability of 0.335.

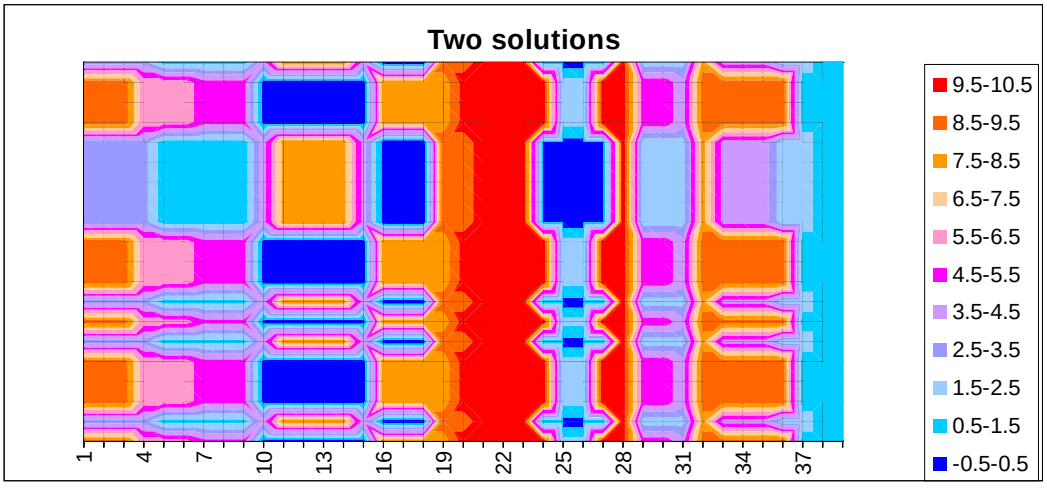

Figure 7.4: Two solutions, repeatability 0.355.

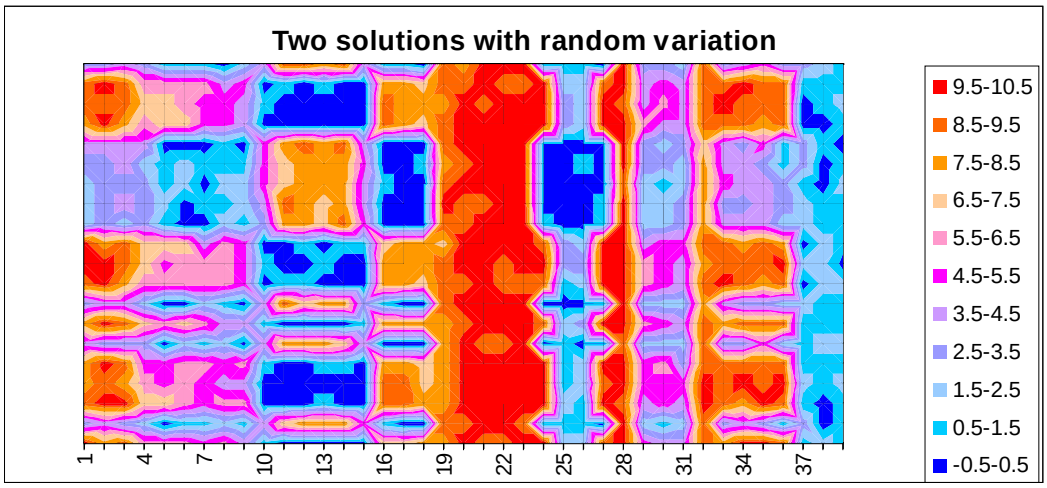

Figure 7.5: Two solutions with random variation, repeatability 0.335.

Given the nature of the model, with only a portion (the additive structure) being optimized, with the remainder (non-additive and environmental structures) being left as random variables, I would expect the results of the multiple runs of the optimization heuristics to be similar to Figure 7.2 or Figure 7.5, that is a single solution with random variation, or multiple solutions with random variation. This will likely occur because with this model it is quite possible for two solutions, say *A* and *B*, which are very similar to each other to in one instant to be evaluated such that *A* is more optimal than *B* even if in long term evaluations (over a large number of pigs, or over multiple evaluation) *B* is more optimal than *A*. To minimize the occurrence of this 1,000 pig simulation are performed to evaluate each solution investigated by the algorithms, however this will not eliminate the occurrence of this.

7.4 Results

7.4.1 High diet

7.4.1.1 Genetic Algorithm

As can been seen in Figure 7.6 and Table 7.1, the Genetic Algorithm is performing significantly better than the Monte Carlo run. When the high diet is being fed, the Genetic Algorithm results for ADG lie beyond 5 standard deviations from the mean. It can also be seen that the algorithm has very little improvement in solution quality from setups that investigated 20,000 genomes to setups that investigated 60,000 genomes, with the parent population size of 30 and stopping criteria of no change for 10 iterations setup, which investigated an average of 19206 genomes, observing a mean ADG of

1.623kg/d, and the parent population size of 50 and stopping criteria of no change for 10 iterations setup, which investigated an average of 50410 genome, observing a mean ADG of 1.626kg/d. The Genetic Algorithm, in a very short time frame, is producing high quality solutions.

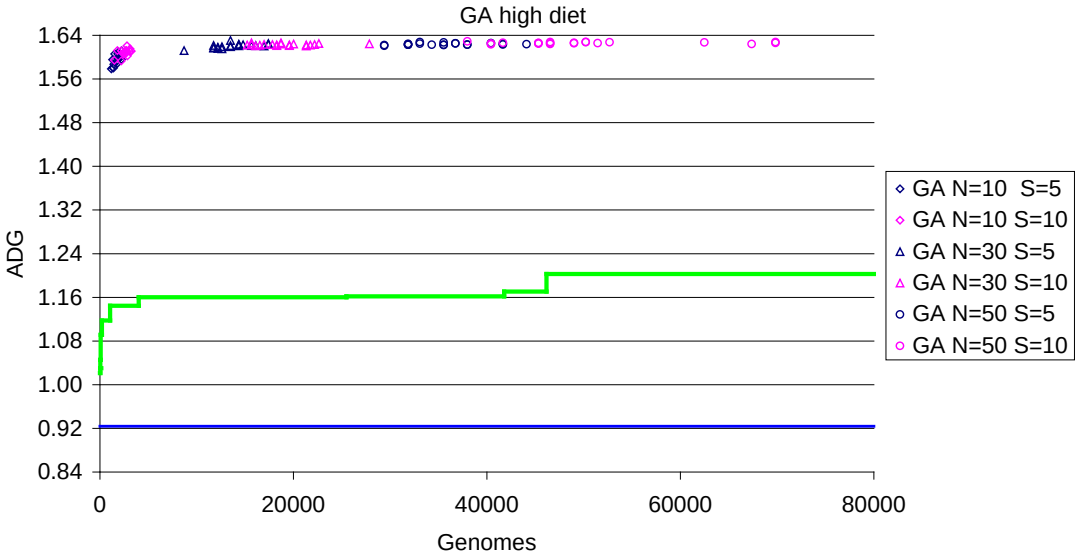

Figure 7.6: Genetic Algorithm results when feeding the high diet.
Green line is the Monte Carlo simulation results, Blue line is the mean.

Table 7.1: Genetic Algorithm results when feeding the high diet.

		Genomes Evaluated		ADG (kg/d)		
Configuration		**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
GA N=10	S=5	1808	355	1.597	0.00987	0.782
	S=10	2577	416	1.608	0.00635	0.849
GA N=30	S=5	13769	2219	1.620	0.00402	0.957
	S=10	19206	3064	1.623	0.00164	0.968
GA N=50	S=5	35526	4168	1.624	0.00169	0.975
	S=10	50410	9599	1.626	0.00143	0.989

The graphs below in Figure 7.7, the worst performance (parent population size of 10 and stopping criteria of no change for 5 iterations), and Figure 7.8, the best performance (parent population size of 50 and stopping criteria of no change for 10 iterations), clearly show that the algorithm is finding only one optimal solution. The improvement

in solution quality between the two setups of the algorithm can clearly be seen with reduction of variation of allele values, which is supported by the increase in repeatability from 0.782 to 0.989 and the decrease on standard deviation in the best genomes found from 0.00987kg/d to 0.00143kg/d. There is also an increase in the mean reported best ADG from 1.597kg/d to 1.626kg/d. The graphs for the other Genetic Algorithm setups can be found in Appendix H-1.

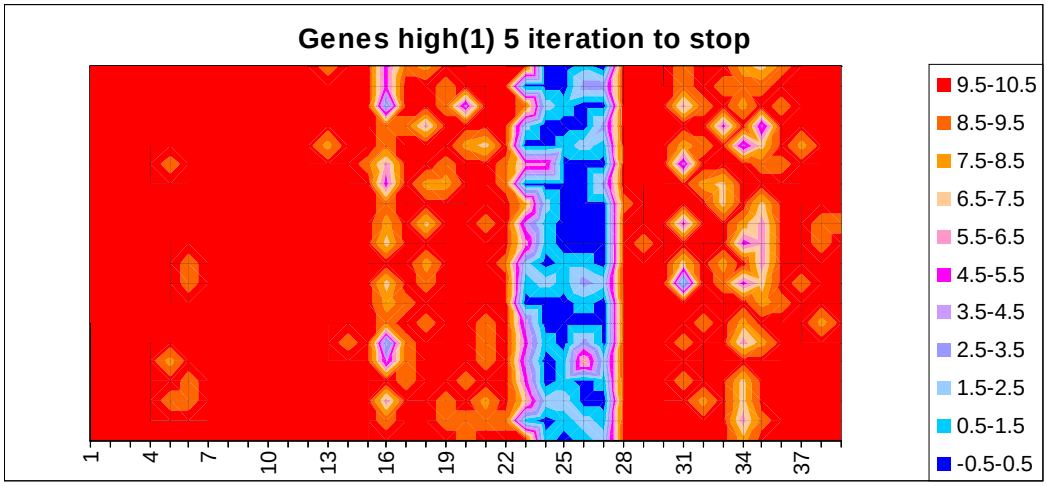

Figure 7.7: Genetic Algorithm high diet, repeatability 0.782.
Parent population size of 10 and stopping criteria of no change for 5 iterations.

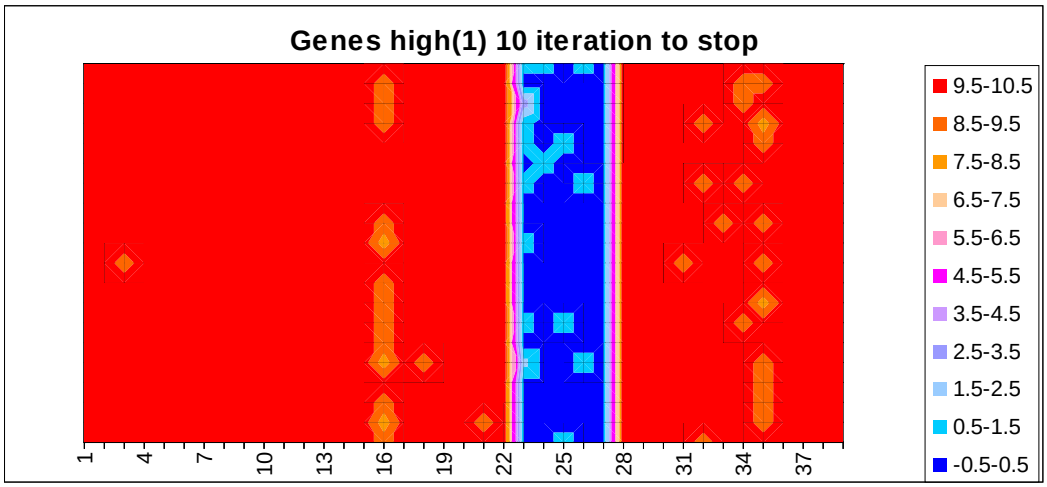

Figure 7.8: Genetic Algorithm high diet, repeatability 0.989.
Parent population size of 50 and stopping criteria of no change for 10 iterations.

7.4.1.2 Tabu Search

The Tabu Search algorithm, results in Figure 7.9 and Table 7.2, has performed significantly better than the Monte Carlo run and is producing results that are beyond 5

standard deviations from the mean. There appears to be only a small gain in solution quality in extending the number of genomes investigated, with the best observed ADG ranging from a mean of 1.615kg/d and standard deviation of 0.00733kg/d for an average of 9308 investigated genomes to a mean of 1.626kg/d and standard deviation of 0.00423kg/d for an average of 58023 investigated genomes.

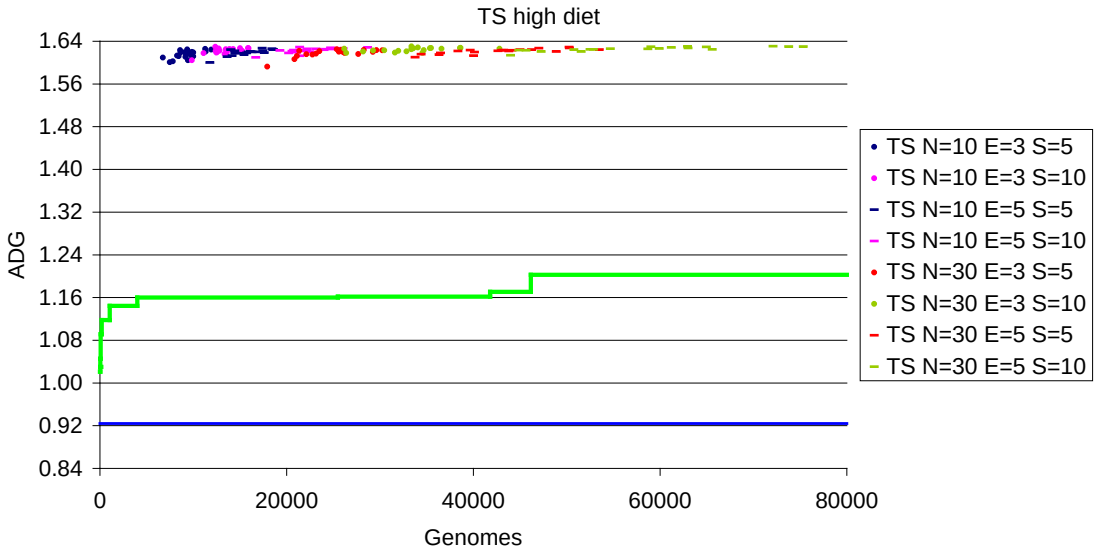

Figure 7.9: Tabu Search results when feeding the high diet.
Green line is the Monte Carlo simulation results, Blue line is the mean.

Table 7.2: Tabu Search results when feeding the high diet.

			Genomes Evaluated		ADG (kg/d)		
Configuration			**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
TS N=10	E=3	S=5	9308	1284	1.615	0.00733	0.849
		S=10	13417	1498	1.622	0.00556	0.915
	E=5	S=5	15828	1843	1.619	0.00608	0.898
		S=10	22526	2693	1.623	0.00498	0.901
TS N=30	E=3	S=5	24937	3287	1.618	0.00751	0.866
		S=10	33393	4129	1.624	0.00358	0.921
	E=5	S=5	42852	5519	1.621	0.00486	0.898
		S=10	58023	9509	1.626	0.00423	0.934

The graphs below in Figure 7.10, the worst performance (10 neighbours, 3 elite solutions and a stopping criteria of no change for 5 iterations), and Figure 7.11, the best performance (30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations), clearly show that the algorithm is finding only one optimal solution. The improvement in solution quality between the two setups of the algorithm can be seen by the reduction in variation of the reported alleles, which is supported by the increase in repeatability from 0.849 to 0.934. The graphs for the other Tabu Search setups can be found in Appendix H-2.

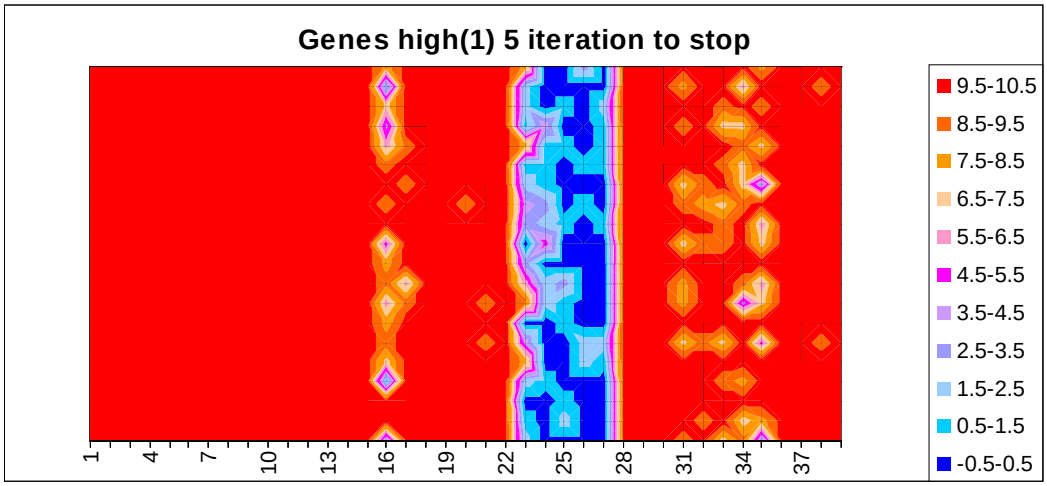

Figure 7.10: Tabu Search high diet, repeatability 0.849.
10 neighbours, 3 elite solutions and a stopping criteria of no change for 5 iterations.

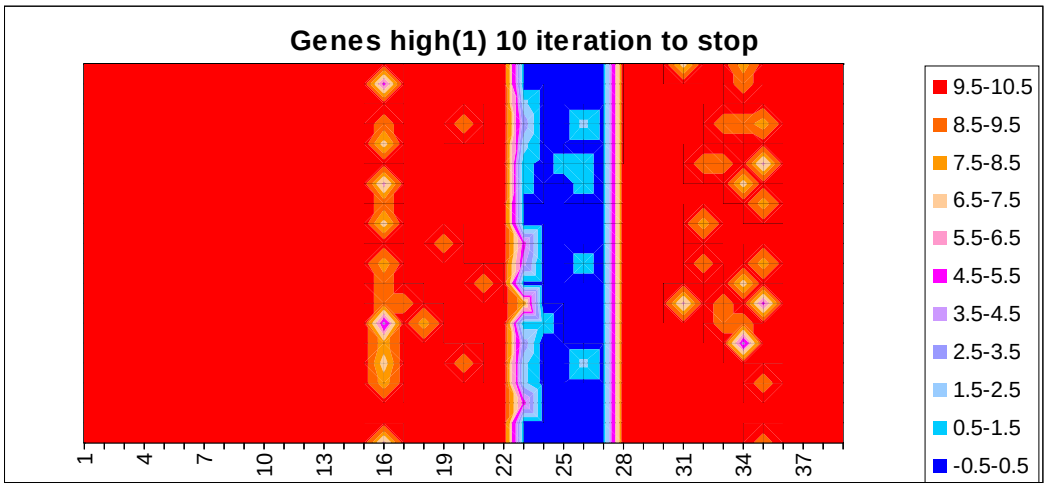

Figure 7.11: Tabu Search high diet, repeatability 0.934.
30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations.

7.4.1.3 Simulated Annealing

The Simulated Annealing algorithm in Figure 7.12 and Table 7.3 below has produced quick results, but with large variation in quality of solution, with for the quickest running setup, which investigated 1451 genomes, giving a repeatability of 0.510 with a mean ADG of 1.523kg/d and standard deviation of 0.01946kg/d, and the longest running setup, which investigated 66001 genomes, giving a repeatability of 0.895 with mean ADG of 1.610kg/d and standard deviation of 0.00301kg/d. However the solution quality does lie outside 4.5 standard deviations.

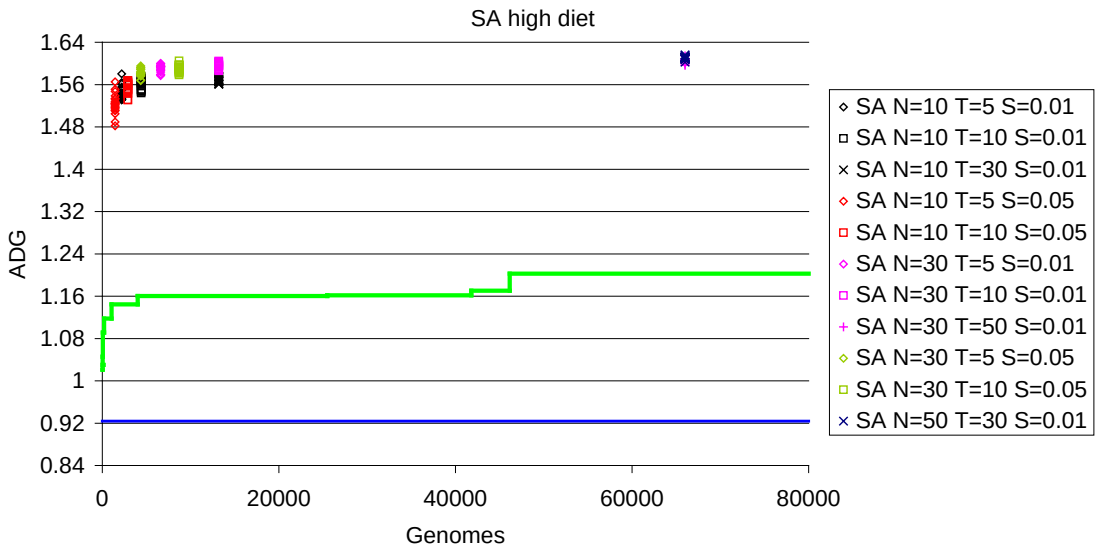

Figure 7.12: Simulated Annealing results when feeding the high diet.
Green line is the Monte Carlo simulation results, Blue line is the mean.

Table 7.3: Simulated Annealing results when feeding the high diet.

			Genomes Evaluated		ADG (kg/d)		
Configuration			**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
SA N=10	T=5	S=0.01	2201	0	1.547	0.01270	0.633
	T=10		4401	0	1.561	0.00929	0.701
	T=30		13201	0	1.568	0.00471	0.726
	T=5	S=0.05	1451	0	1.523	0.01946	0.510
	T=10		2901	0	1.554	0.01114	0.674
SA N=30	T=5	S=0.01	6601	0	1.591	0.00678	0.809
	T=10		13201	0	1.596	0.00493	0.836
	T=50		66001	0	1.604	0.00354	0.894
	T=5	S=0.05	4351	0	1.582	0.00745	0.712
	T=10		8701	0	1.590	0.00642	0.800
SA N=50	T=30	S=0.01	66001	0	1.610	0.00301	0.895

Figure 7.13, which presents the worst performing setup (neighbourhood size of 10, cooling time of 5 iterations and stop temperature of 0.05), shows that these results are very clear or consistent. This is supported by the low repeatability of 0.510 and a standard deviation of 0.01270kg/d in best ADG reported. However Figure 7.14, the best performance (neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01), with a repeatability of 0.895 is producing results showing a single optimal solution. However, there is still reasonable variation being produced. The graphs for the other Simulated Annealing setups can be found in Appendix H-3.

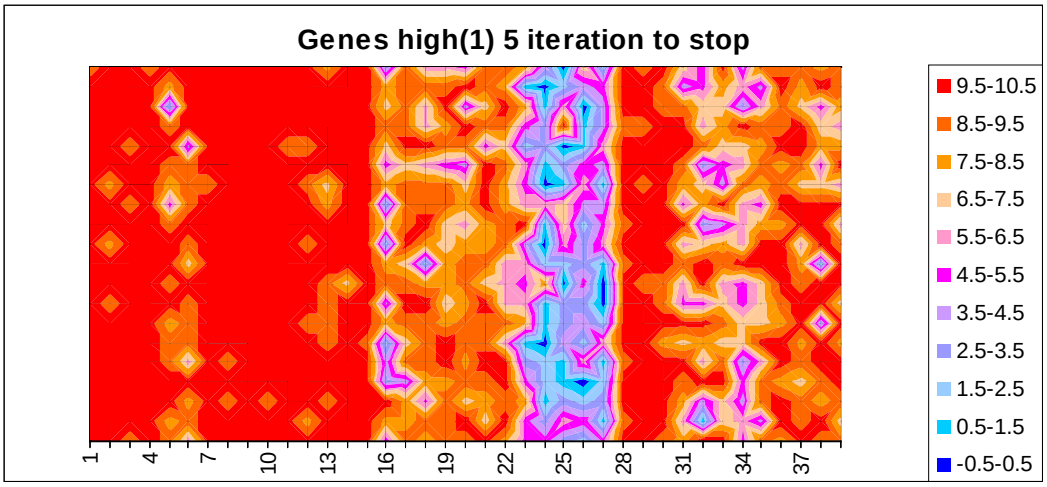

Figure 7.13: Simulated Annealing high diet, repeatability 0.510.
Neighbourhood size of 10, cooling time of 5 iterations and stop temperature of 0.05.

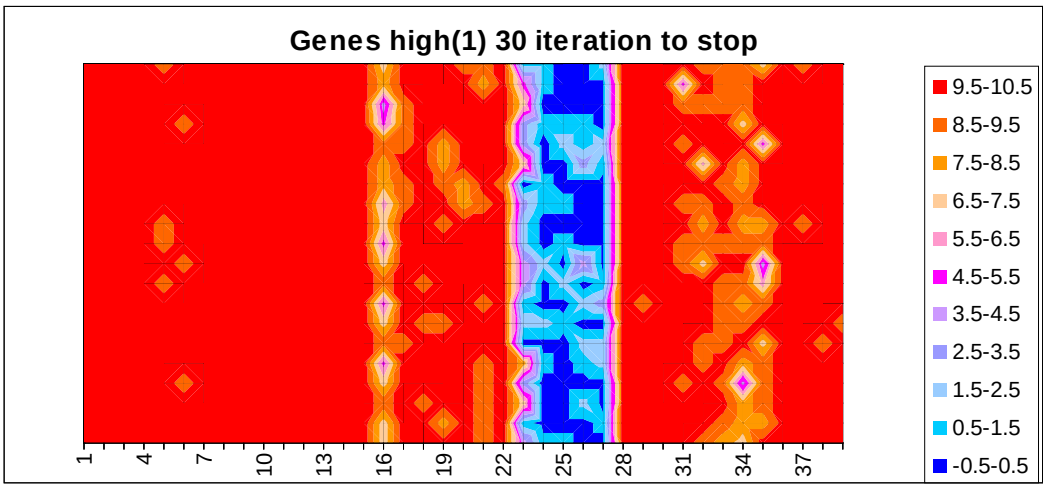

Figure 7.14: Simulated Annealing high diet, repeatability 0.895.
Neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01.

7.4.1.4 Algorithm comparison

Figure 7.15 below shows a comparison between the three algorithms. When comparing results for particular numbers of genomes investigated, the Genetic algorithm gives results with less variation than the Tabu Search, although they have similar average performance. This can be seen when the results from the Genetic algorithm setup with N=30 and S=5, which investigated an average of 13769 genomes with a mean best ADG of 1.620kg/d and standard deviation of 0.00402kg/d, and setup N=50 and S=10, which investigated an average of 50410 genomes with a mean best ADG of 1.626kg/d and standard deviation of 0.00143kg/d, are respectively compared to the Tabu Search setup with N=10, E=3, and S=10, which investigated on average 13417 genomes with a mean

best ADG of 1.622kg/d with standard deviation of 0.00556kg/d, and setup *N*=30, *E*=5, and *S*=10, which investigated 58023 genomes on average with a mean best ADG of 1.626kg/d with standard deviation of 0.00423kg/d. It can also be seen in this figure that Simulated Annealing has a lower performance level than both Genetic Algorithm and Tabu Search, which can be observed when the Simulated Annealing setup with *N*=10, *T*=30, *S*=0.01, which investigated 13201 genomes with mean best ADG of 1.568kg/d and standard deviation of 0.00471kg/d, and setup *N*=50, *T*=30, *S*=0.01, which investigated 66001 genomes with mean best ADG of 1.610kg/d and standard deviation of 0.00301kg/d, are compared with the Genetic Algorithm and Tabu Search results above.

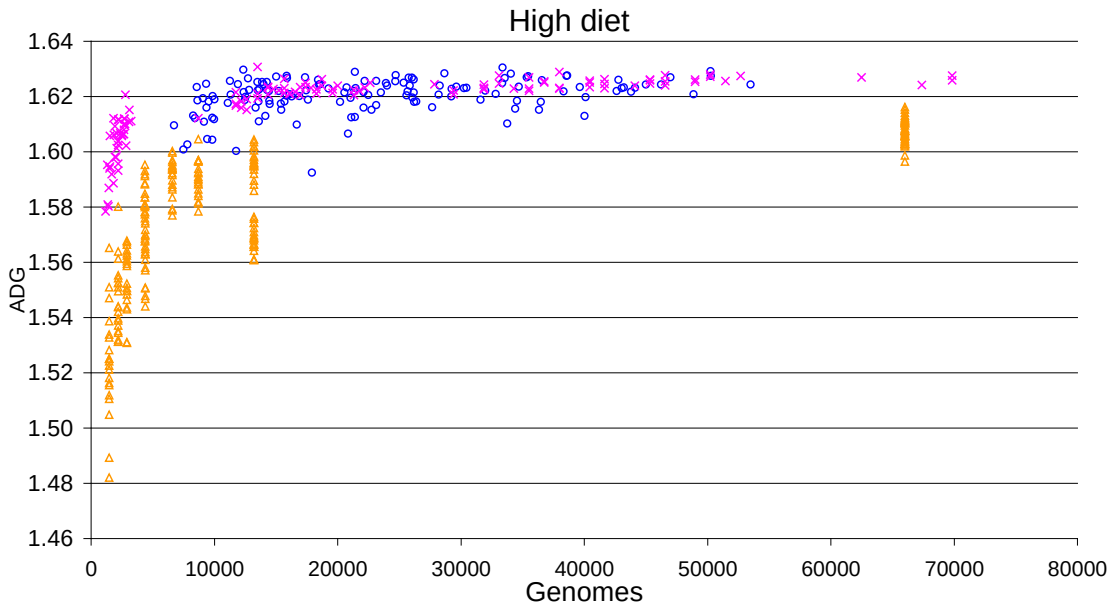

Figure 7.15: Comparison of the three algorithm results on high diet.
Pink crosses – Genetic Algorithm, Blue circles – Tabu Search, Orange triangles – Simulated Annealing.

The graphs below in Figure 7.16, Figure 7.17, and Figure 7.18, show that the algorithms are producing the same general solution. These graphs show that the alleles at loci 1 to 22 and 28 to 39 are maximized and the alleles at loci 23 to 27 are minimized to achieve optimal performance. The results are clearest in the Genetic Algorithm display, with the results being the most consistent between runs and with the highest repeatability value of 0.989. The Tabu Search had problems optimising locus 23. Tabu Search sometimes did not completely minimize the loci, which is shown by the ragged edge between the

red and blue regions. Compare this to the straight edge between loci 27 and 28 where all the loci are at the extreme. The Simulated Annealing also shows the optimal solution but is showing more inconsistencies between runs with a repeatability of 0.895.

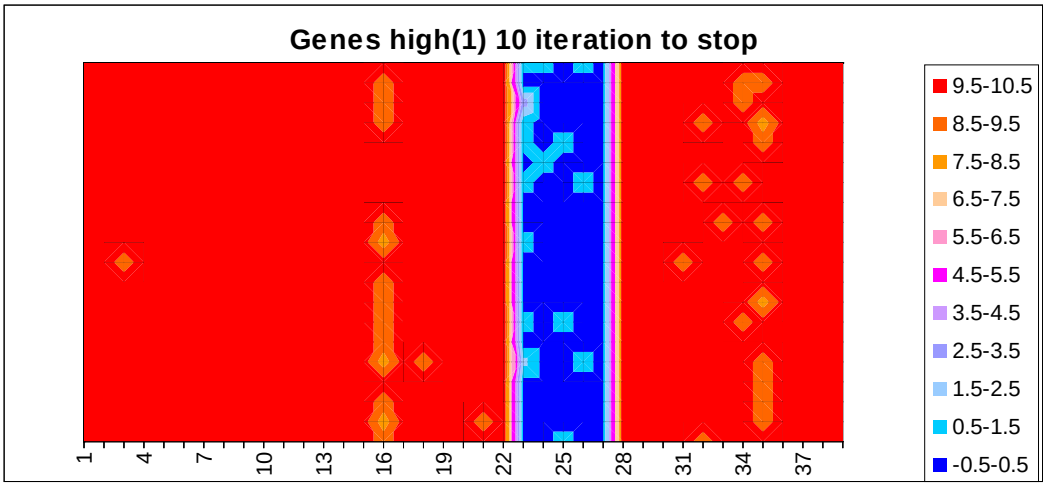

Figure 7.16: Genetic Algorithm high diet, repeatability 0.989.
Parent population size of 50 and stopping criteria of no change for 10 iterations.

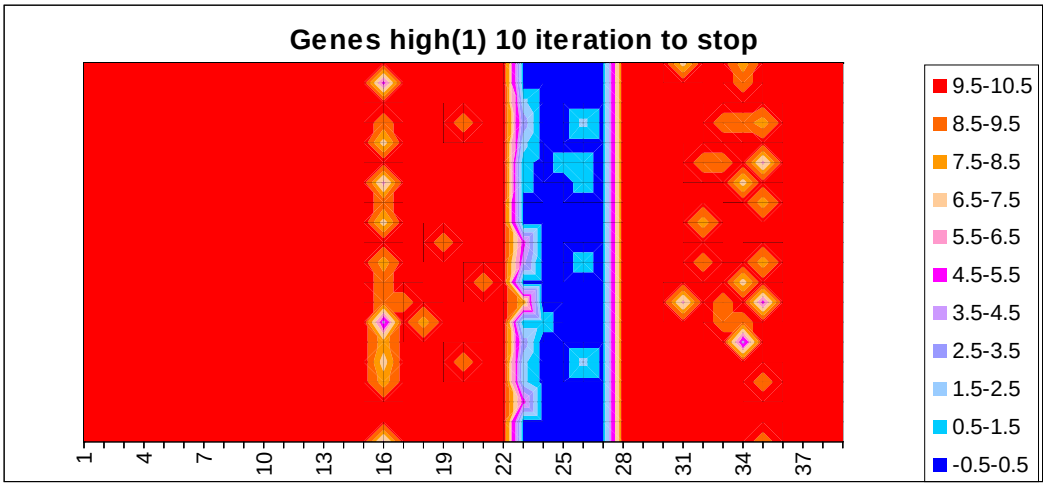

Figure 7.17: Tabu Search high diet, repeatability 0.934.
30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations.

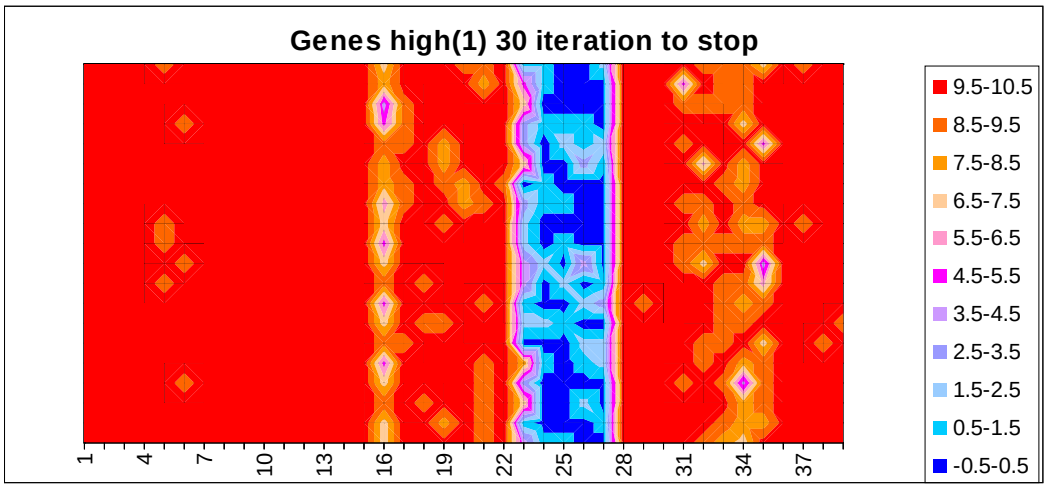

Figure 7.18: Simulated Annealing high diet, repeatability 0.895.
Neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01.

Table 7.4 below presents which loci have been maximized and which have been minimized along with the resultant effect on the growth parameters. This table clearly shows that Pdmax has been maximized, with a mean of 281.94, 10 loci have been minimized for LP_{min} and 5 maximized, to give a mean of 0.576, 8 loci of Em have been maximized and 7 have been minimized, with a mean of 0.481. All but one locus for feed intake (p) have been maximized with a mean of 1.149, and all but two loci for *w* have been maximized giving a mean of 5.558.

For the pig farmer this would mean that, given a high diet, the optimal pig has:

- high protein deposition (high muscle development)
- below average lipid to protein ratio (little fat to protein ratio, i.e. a lean pig)
- average energy requirements for maintenance
- high ad libitum feed intake (the pig is willing to eat a lot of food)
- high water content.

Table 7.4: Optimal genome for average daily gain with high diet being fed.

ADG optimal for High diet						
Loci	**Optimal**	Pd_{max}	LP_{min}	E_m	p	w
1-4	max	max	min			
5-6	max	max		max		
7-10	max	max			max	
11-12	max	max				max
13	max	max	min	max		
14-15	max	max	min			max
16	max		min	max		
17-21	max		max		max	
22	max		min	max		max
23	min		min		min	max
24-27	min			min		
28-30	max			min	max	
31-33	max			max		max
34-35	max				max	min
36-39	max					max
Mean		281.94	0.576	0.481	1.149	5.558

Table 7.5 below presents the performance of the optimal genome for average daily gain with the high diet being fed. The mean value of 1.6162kg/day for ADG is over 5 standard deviations from the population mean of 0.9241kg/day presented in Table 5.30. This is an extreme values for ADG and could be attributed to the maximum possible value for LP_{min}, Pd_{max} and w in the model being possibly to large. In particular a Pd_{max} of 281.94 is beyond values observed in the literature listed in Table 5.6. This combined with possibly of the correlations between LP_{min}, Pd_{max} and w being too strong has resulted in this extreme observation. In the future when data on these parameters are estimated and published in the literature, the data could be used to calibrate the model.

Table 7.5: Performance of optimal genome for ADG with high diet being fed.

	ADG optimal for High diet	
	Mean	**Std Dev**
FCR	1.5946	0.0952
ADG	1.6162	0.1233
BF	9.7895	1.4428
DTS	44.075	3.3795
Mortality	0	

7.4.2 Low diet

7.4.2.1 Genetic Algorithm

Figure 7.19 and Table 7.6 shows that the Genetic Algorithm performs very consistently throughout the range, repeatability ranging from 0.860 to 0.987, and mean best ADG ranging from 1.467kg/d to 1.486kg/d with standard deviations ranging from 0.00655kg/d to 0.00135kg/d respectively. The Genetic Algorithm results for ADG on the low diet lie beyond 5 standard deviations of the mean.

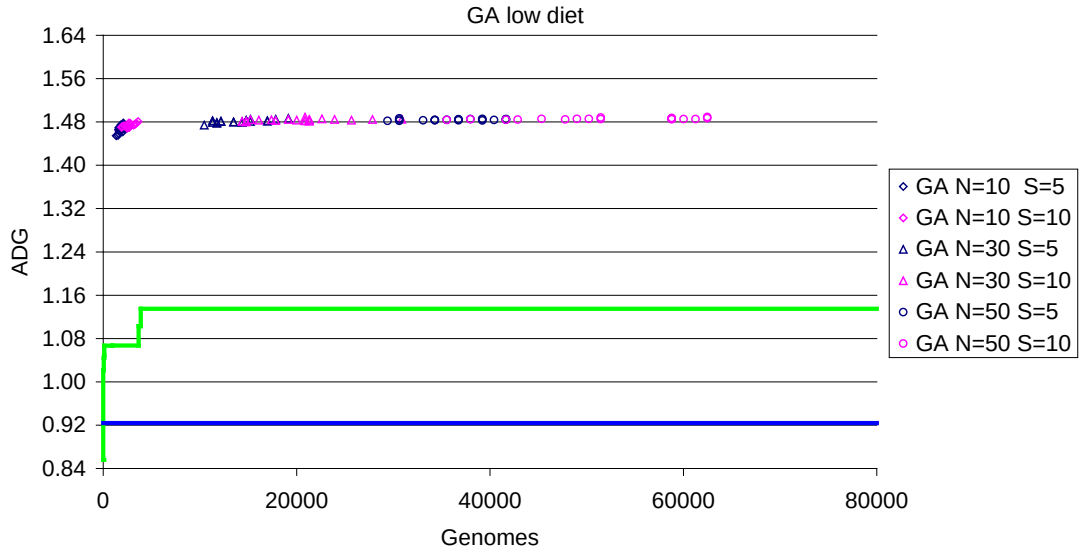

Figure 7.19: Genetic Algorithm results when feeding the low diet.
Green line is the Monte Carlo simulation results, Blue line is the mean.

Table 7.6: Genetic Algorithm results when feeding the low diet.

		Genomes Evaluated		ADG (kg/d)		
Configuration		**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
GA N=10	S=5	1923	320	1.467	0.00655	0.860
	S=10	2690	438	1.473	0.00324	0.884
GA N=30	S=5	13942	2459	1.481	0.00274	0.974
	S=10	20359	4446	1.484	0.00196	0.981
GA N=50	S=5	36200	3964	1.484	0.00120	0.985
	S=10	52615	8501	1.486	0.00135	0.987

The graphs below in Figure 7.20, which presents the worst performing setup (parent population size of 10 and stopping criteria of no change for 5 iterations), and Figure 7.21, the best performing setup (parent population size of 50 and stopping criteria of no change for 10 iterations), show that the algorithm is only finding one optimal solution. The improvement in solution quality between the two setups of the algorithm can clearly be seen with reduction variation of the alleles in the best found solutions, which is supported by the increase in repeatability from 0.860 to 0.987, and reduction in standard deviation from 0.00655kg/d to 0.00135kg/d. The graphs for the other Genetic Algorithm setups can be found in Appendix H-1.

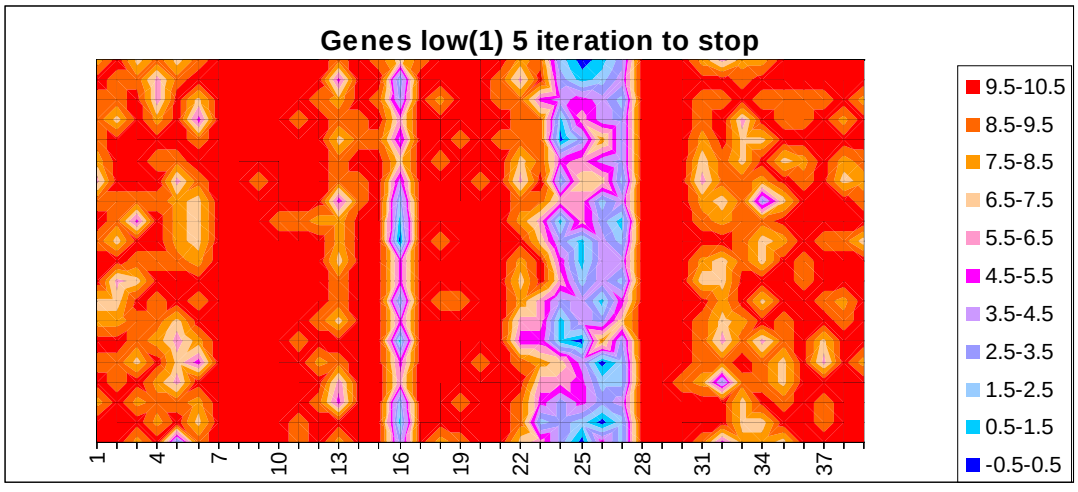

Figure 7.20: Genetic Algorithm low diet, repeatability 0.860.
Parent population size of 10 and stopping criteria of no change for 10 iterations.

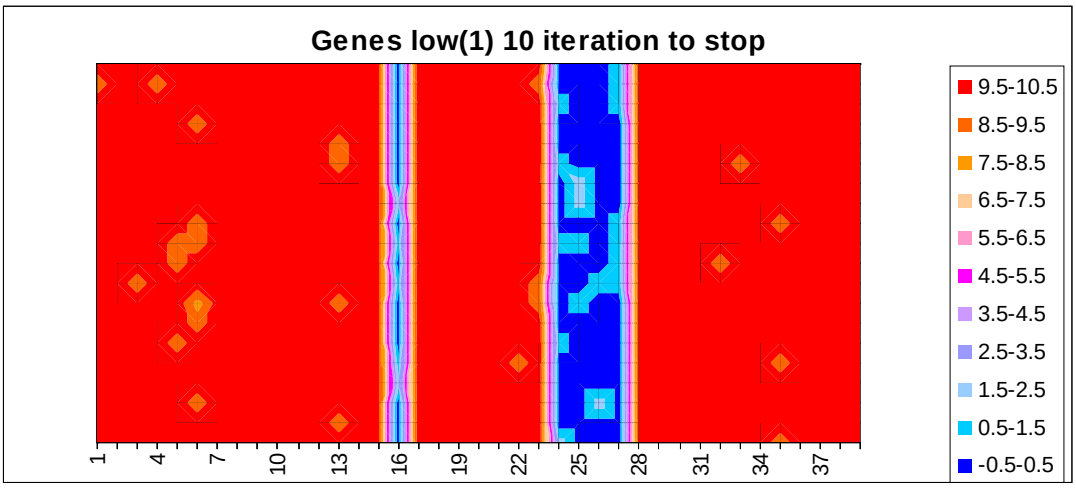

Figure 7.21: Genetic Algorithm low diet, repeatability 0.987.
Parent population size of 50 and stopping criteria of no change for 10 iterations.

7.4.2.2 Tabu Search

The Tabu Search results in Figure 7.22 and Table 7.7 show a consistent and good quality solution production that is beyond 5 standard deviations from the mean. The mean best ADG range from 1.472kg/d with a standard deviation of 0.00667kg/d and repeatability of 0.738 when an average of 8430 genomes are investigated to 1.485kg/d with a standard deviation of 0.00375kg/d and repeatability of 0.956 when an average of 59375 genomes are investigated.

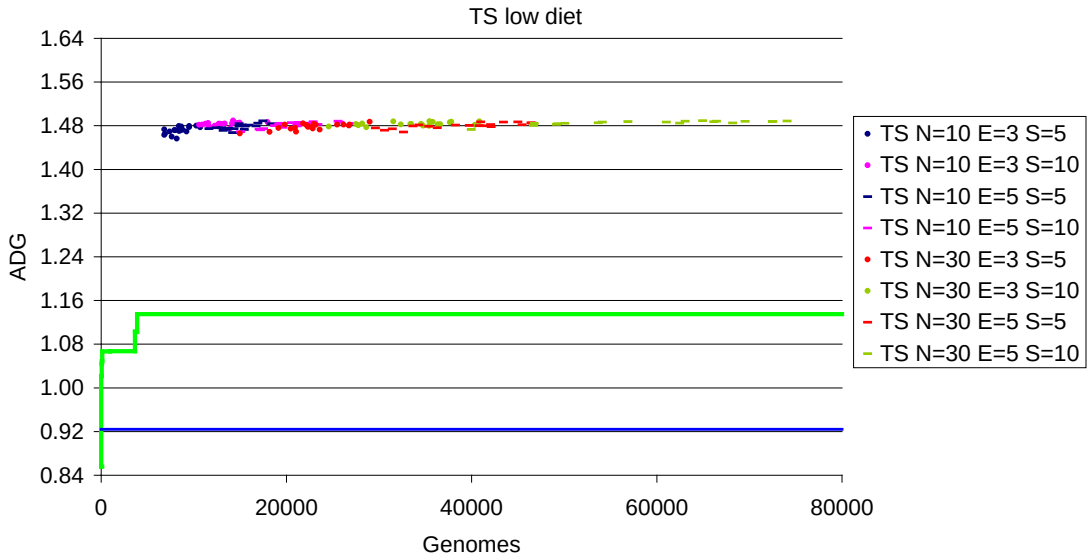

Figure 7.22: Tabu Search algorithm results when feeding the low diet.
Green line is the Monte Carlo simulation results, Blue line is the mean.

Table 7.7: Tabu Search results when feeding the low diet.

Configuration			Genomes Evaluated		ADG (kg/d)		
			Mean	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
TS N=10	E=3	S=5	8430	1063	1.472	0.00667	0.738
		S=10	13095	1740	1.483	0.00330	0.930
	E=5	S=5	14901	1719	1.478	0.00518	0.874
		S=10	20977	3172	1.481	0.00529	0.916
TS N=30	E=3	S=5	22506	3293	1.478	0.00574	0.868
		S=10	33655	4443	1.483	0.00322	0.916
	E=5	S=5	38797	5466	1.480	0.00496	0.884
		S=10	59375	10132	1.485	0.00375	0.956

The graphs below in Figure 7.23, the worst performance (10 neighbours, 3 elite solutions and a stopping criteria of no change for 5 iterations), and Figure 7.24, the best performance (30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations), clearly show that the algorithm is finding only one optimal solution. The improvement in solution quality between the two setups of the algorithm can be seen with reduction the in variation in reported best alleles, which is supported by the increase in repeatability from 0.738 to 0.956 and the reduction in variation from 0.00667kg/d to 0.00375kg/d. The graphs for the other Tabu Search setups can be found in Appendix H-2.

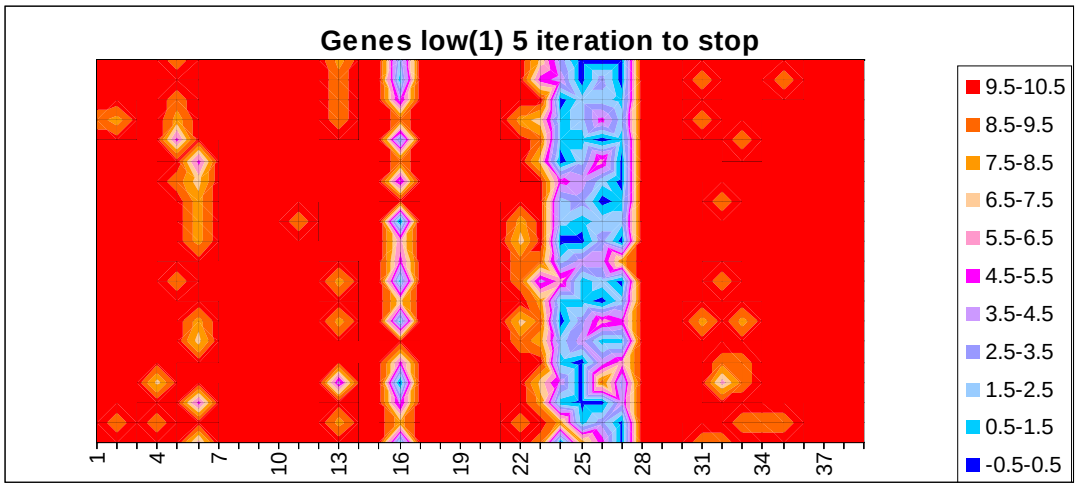

Figure 7.23: Tabu Search low diet, repeatability 0.738.

10 neighbours, 3 elite solutions and a stopping criteria of no change for 5 iterations.

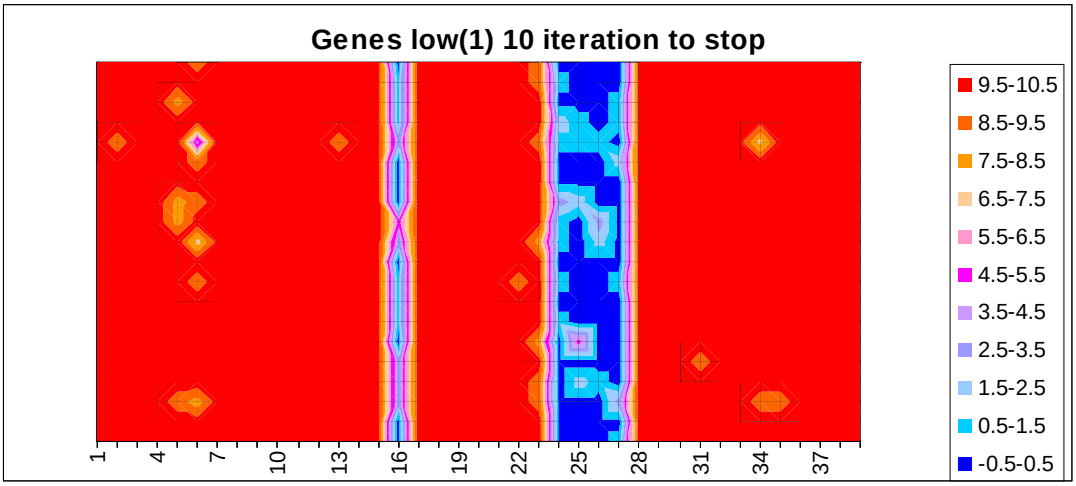

Figure 7.24: Tabu Search low diet, repeatability 0.956.

30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations.

7.4.2.3 Simulated Annealing

As can be seen in Figure 7.25 and Table 7.8 the Simulated Annealing results show some variation, with mean best ADG found ranging from 1.408kg/d to 1.472kg/d, standard deviations ranging from 0.01039kg/d to 0.00201kg/d, and the repeatability of the performances ranging from 0.593 to 0.921 for the N=10, T=5, and S=0.05 setup and N=50, T=30 and S=0.01 setup respectively. However the results lie beyond 4.5 standard deviations from the mean.

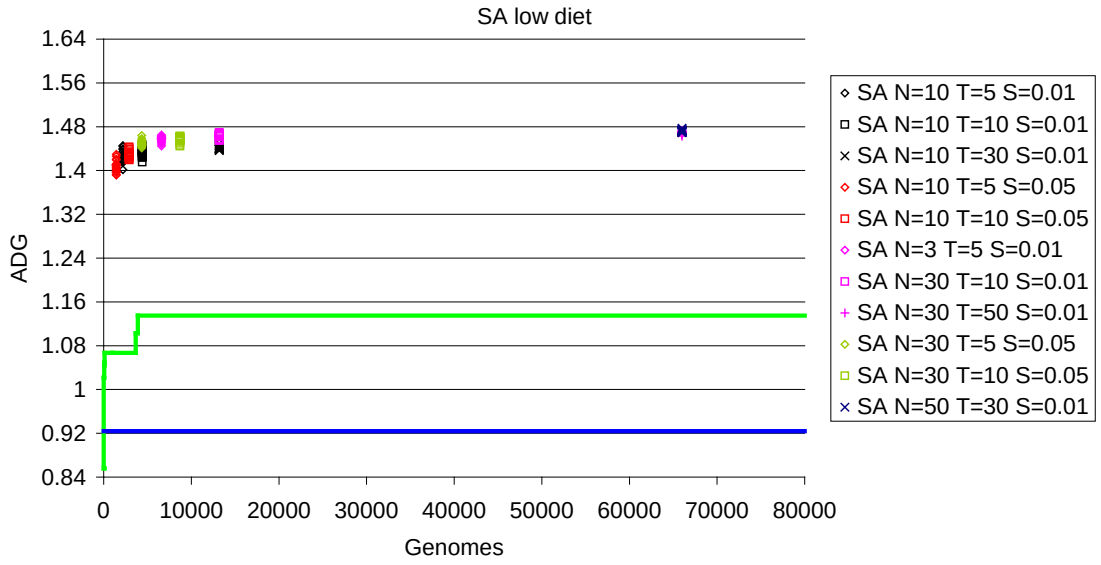

Figure 7.25: Simulated Annealing algorithm results when feeding the low diet. Green line is the Monte Carlo simulation results, Blue line is the mean.

Table 7.8: Simulated Annealing results when feeding the high diet.

			Genomes Evaluated		ADG (kg/d)		
Configuration			**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
SA N=10	T=5	S=0.01	2201	0	1.425	0.01102	0.606
	T=10		4401	0	1.433	0.00753	0.697
	T=30		13201	0	1.442	0.00380	0.712
	T=5	S=0.05	1451	0	1.408	0.01039	0.593
	T=10		2901	0	1.430	0.00694	0.601
SA N=30	T=5	S=0.01	6601	0	1.457	0.00485	0.771
	T=10		13201	0	1.463	0.00408	0.819
	T=50		66001	0	1.469	0.00241	0.881
	T=5	S=0.05	4351	0	1.450	0.00544	0.666
	T=10		8701	0	1.456	0.00516	0.763
SA N=50	T=30	S=0.01	66001	0	1.472	0.00201	0.921

Figure 7.26, which presents the results from the worst performing setup (neighbourhood size of 10, cooling time of 5 iterations and stop temperature of 0.05), shows that this setup is not producing very clear or consistent results, which is supported by the low repeatability of 0.593 and standard deviation in best ADG of 0.01102kg/d. However Figure 7.27, the best performing setup (neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01), with a repeatability of 0.921 is producing results that show a single optimal solution, even though there is still reasonable variation in the best genome being found. The graphs for the other Simulated Annealing setups can be found in Appendix H-3.

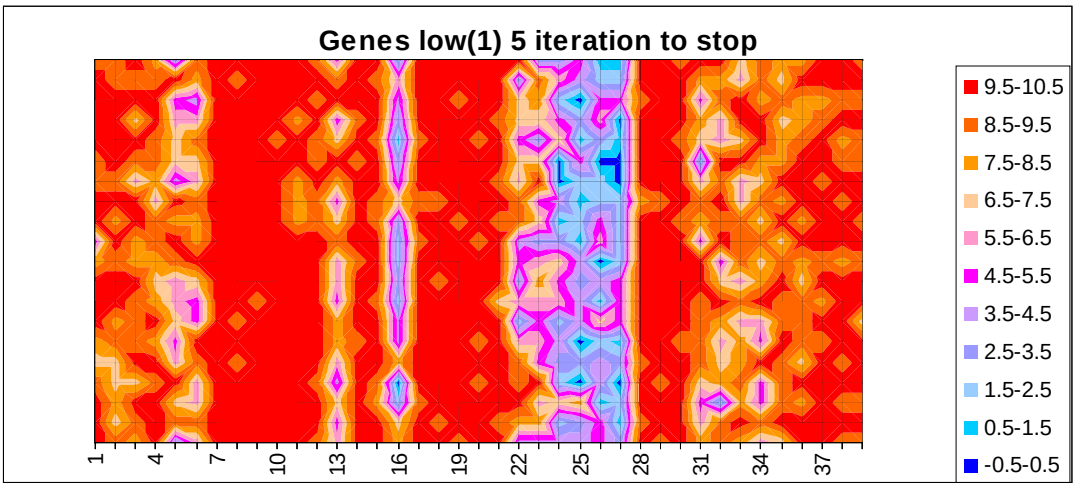

Figure 7.26: Simulated Annealing low diet, repeatability 0.593.
Neighbourhood size of 10, cooling time of 5 iterations and stop temperature of 0.05.

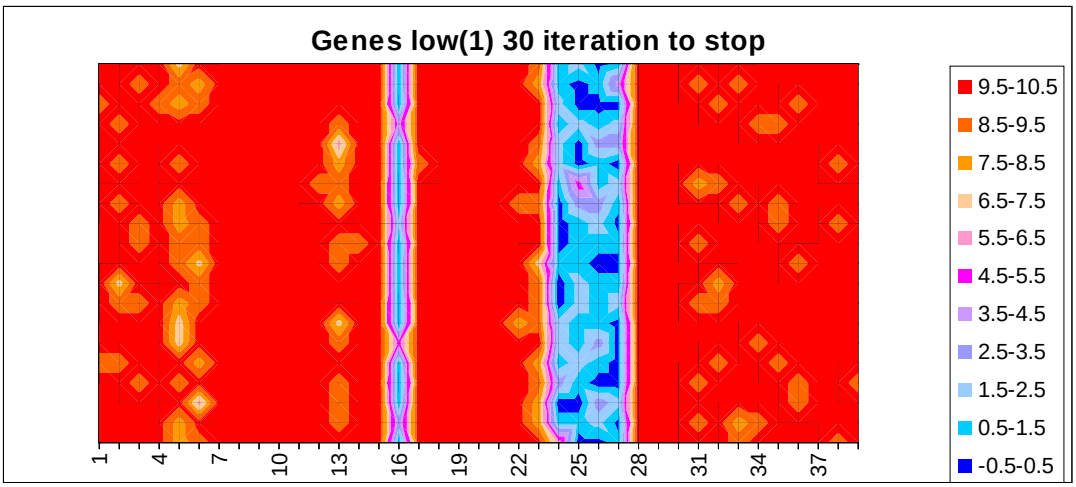

Figure 7.27: Simulated Annealing low diet, repeatability 0.921.
Neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01.

7.4.2.4 Algorithm comparison

As can be seen in Figure 7.28 below, the Genetic Algorithm and Tabu Search algorithm produced very similar results, although the Tabu Search solutions have more variation. This can be seen when the Genetic Algorithm setup with N=30, and S=5, which evaluated on average 13942 genomes with mean best ADG of 1.481kg/d and standard deviation of 0.00274kg/d, and setup N=50, and S=10, which evaluated on average 52615 genomes with a mean best ADG of 1.486kg/d and standard deviation of 0.00135kg/d, is compared respectively with the Tabu Search setup with N=10, E=3, and S=10, which investigated an average of 13095 genomes with a mean best ADG of 1.483kg/d and standard deviation of 0.00330kg/d, and setup N=30, E=5, and S=10,

which investigated an average of 59375 genomes with a mean best ADG of 1.485kg/d and standard deviation of 0.00375kg/d. Both algorithms outperform the Simulated Annealing algorithm, which can be seen when the Simulated Annealing setup N=10, T=30, and S=0.01, which evaluated 13201 genomes with a mean best ADG of 1.442kg/d and standard deviation of 0.00380kg/d, and setup N=50, T=30, and S=0.01, which evaluated 66001 genomes with a mean best ADG of 1.472kg/d and standard deviation of 0.00201kg/d, are compared respectively to the Genetic Algorithm and Tabu Search setups above.

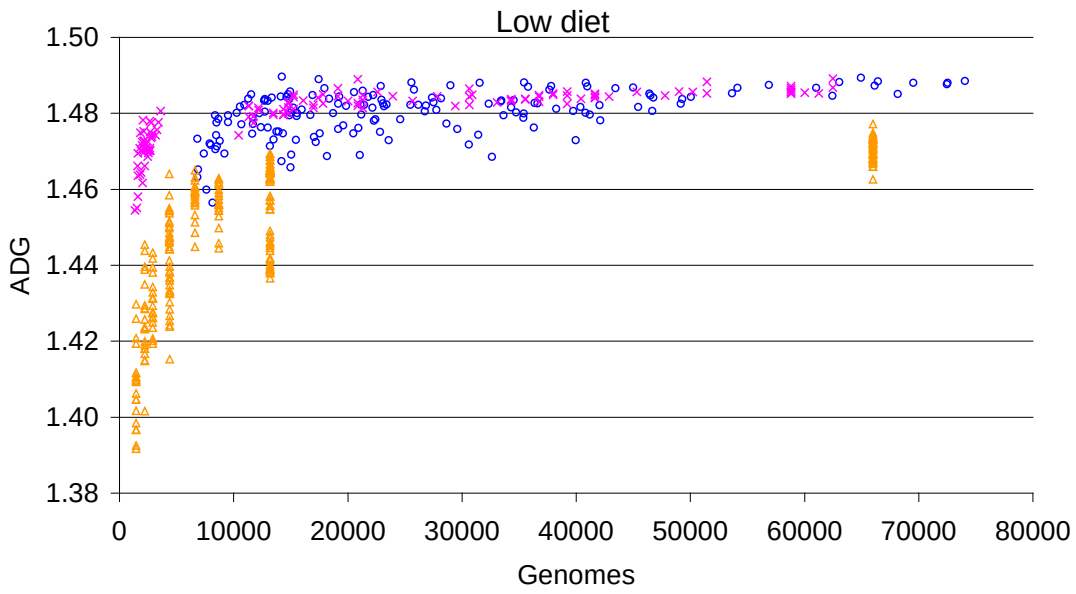

Figure 7.28: Comparison of the three algorithm results on low diet.
Pink crosses – Genetic Algorithm, Blue circles – Tabu Search, Orange triangles – Simulated Annealing.

The graphs in Figure 7.29, Figure 7.30, and Figure 7.31 show that, again, the algorithms are converging to a common solution. For optimal performance on the low diet, the alleles at loci 1 to 15, 17 to 23, and 28 to 39 are maximized, and locus 16 and loci 24 to 27 are minimized. The Genetic Algorithm has produced the most consistent results, with repeatability of 0.987, and hence the clearest graphical representation and the least variation in the reported best alleles. All three algorithms had problems with optimizing locus 16, with the simulated annealing having the most trouble and the Genetic Algorithm the least. As was seen with the high diet results there is significant variation in the reported best alleles in the Simulated Annealing solutions.

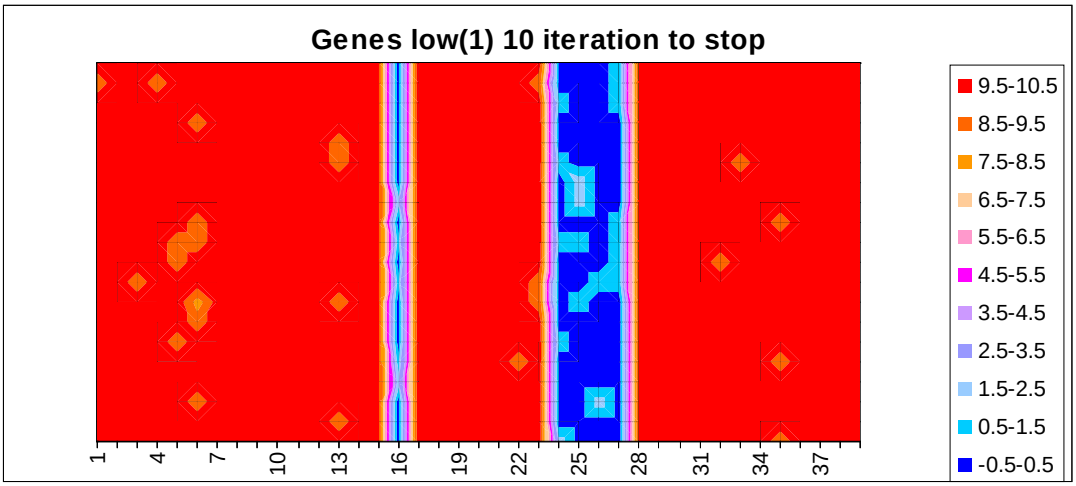

Figure 7.29: Genetic Algorithm low diet, repeatability 0.987.

Parent population size of 50 and stopping criteria of no change for 10 iterations.

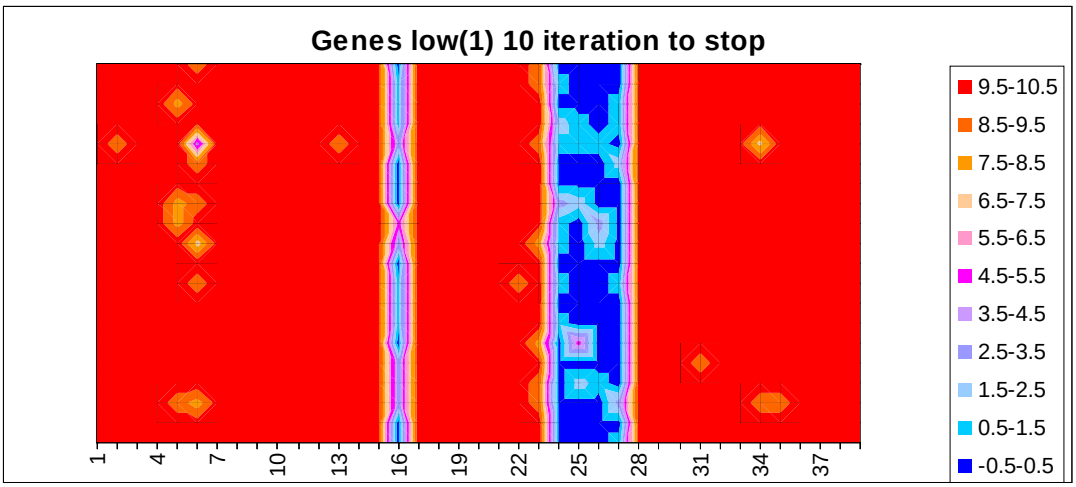

Figure 7.30: Tabu Search low diet, repeatability 0.956.

30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations.

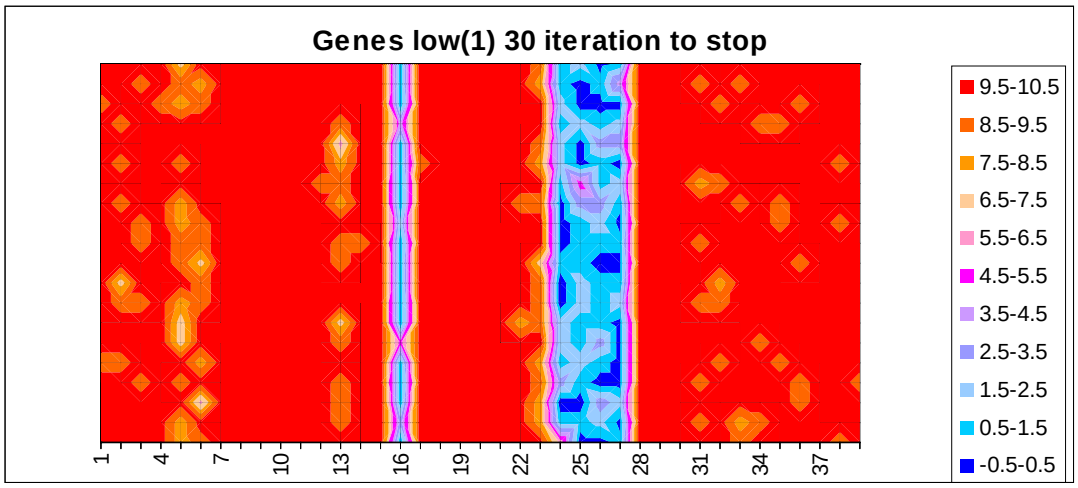

Figure 7.31: Simulated Annealing low diet, repeatability 0.921. Neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01.

The factors for this optimal performance are shown in Table 7.9 below. This table clearly shows that Pd_{max} has been maximized with a value of 281.94, 8 loci have been minimized for LP_{min} with a value of 0.647 and 7 have been maximized, 7 loci of Em have been maximized giving a value of 0.468 and 8 have been minimized, feed intake (p) has been maximized for a value of 1.166, and 3 loci have been minimized for w and 12 have been maximized giving a value of 5.515.

For the pig farmer this would mean that, given a low diet, the optimal pig has

- high protein deposition (high muscle development)
- average lipid to protein ratio (little fat to protein ratio, i.e. a lean pig)
- average energy requirements for maintenance
- high ad libitum feed intake (the pig is willing to eat a lot of food)
- high water content.

Table 7.9: Optimal genome for average daily gain with low diet being fed.

ADG optimal for Low diet						
Loci	**Optimal**	Pd_{max}	LP_{min}	E_m	p	w
1-4	max	max	min			
5-6	max	max		max		
7-10	max	max			max	
11-12	max	max				max
13	max	max	min	max		
14-15	max	max	min			max
16	min		max	min		
17-21	max		max		max	
22	max		min	max		max
23	max		max		max	min
24-27	min			min		
28-30	max			min	max	
31-33	max			max		max
34-35	max				max	min
36-39	max					max
Mean		281.94	0.647	0.468	1.166	5.515

Table 7.10 below presents the performance of the optimal genome for average daily gain with the low diet being fed. The mean value of 1.4766kg/day for ADG is over 5 standard deviations from the population mean of 0.9241kg/day presented in Table 5.30. As for the high diet, this is an extreme value for ADG and could be attributed to the maximum possible values for LP_{min}, Pd_{max} and w in the model being possibly to large. In particular a Pd_{max} of 281.94 is beyond values observed in the literature listed in Table 5.6. This combined with possibly of the correlations between LP_{min}, Pd_{max} and w being too strong has resulted in this extreme observation. As more data on these parameters become available in the literature, the model could be further calibrated.

Table 7.10: Performance of optimal genome for ADG with low diet being fed.

	ADG optimal for Low diet	
	Mean	**Std Dev**
FCR	1.9459	0.0536
ADG	1.4766	0.102
BF	13.66	0.9746
DTS	48.226	3.3425
Mortality	0	

7.4.3 Diet comparison

Figure 7.32 shows that the comparative performance between the three algorithms is unaffected by the type of diet that is being fed.

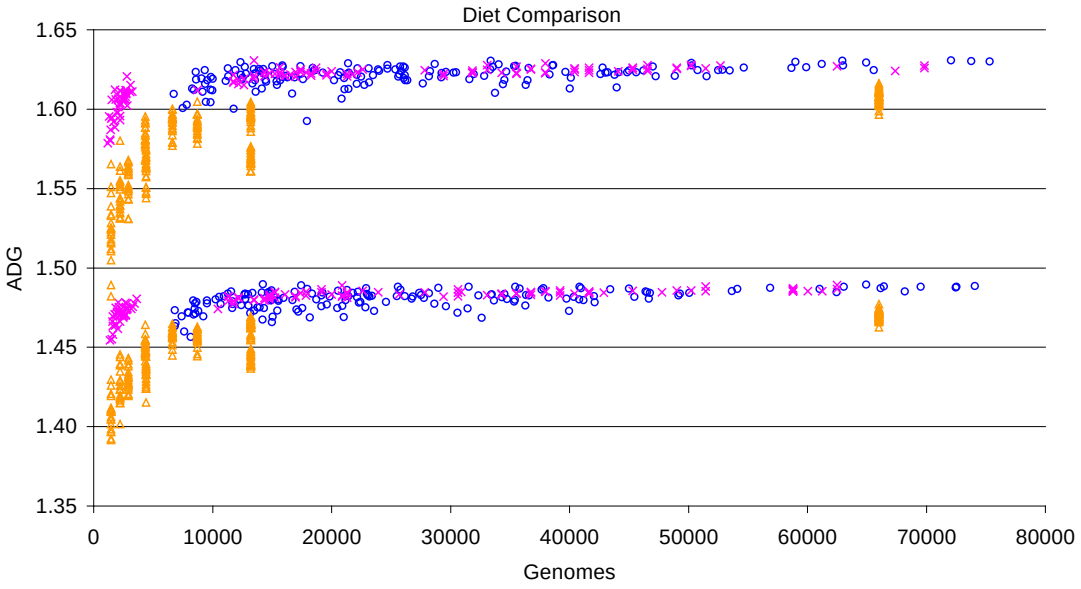

Figure 7.32: Comparison of the three algorithm results on low diet and high diet. Pink crosses – Genetic Algorithm, Blue circles – Tabu Search, Orange triangles – Simulated Annealing.

Figure 7.33 below provides a visual comparison of the optimal results for the high and low diet when optimizing average daily gain. The results shown are from the Genetic Algorithm, which had the least variation in reported best alleles of the three algorithms.

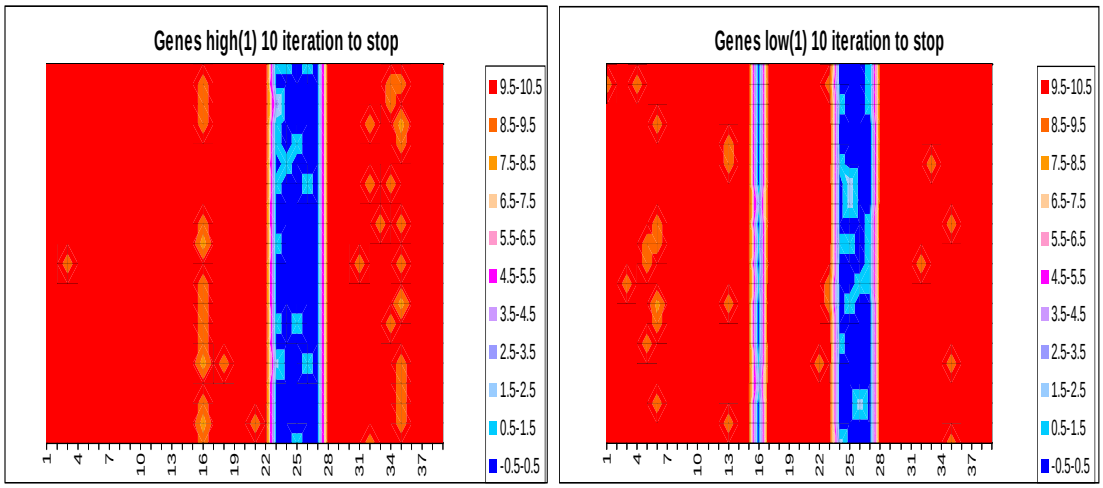

Figure 7.33: Genetic Algorithm diet comparison.
Parent population size of 50 and stopping criteria of no change for 10. High diet results are on the left and low diet results on the right.

The opposing diets produce results that differ at only two loci. Locus 16 is maximized for the high diet but minimized for the low diet. Locus 16 is negatively correlated to LP_{min} and positively correlated to E_m. This indicates that for the high diet, at this locus, LP_{min} has been minimized and E_m maximized, but for the low diet E_m had been minimized and LP_{min} maximized. When attempting to maximize ADG, it is desirable to minimize, where possible, the energy requirements for maintenance, allowing more energy in the diet to be used for growth. This desire is being observed at locus 16 for the low diet where there is limited energy available, however for the high diet, where energy in the diet is significantly less restricting, there more advantage in minimizing LP_{min} for locus 16 than E_m, allowing for a better partitioning between lipids and protein.

The other locus affected by the diet is locus 23. This locus is positively correlated to LP_{min} and feed intake, p, and negatively correlated to w. For the high diet, the locus was minimized, resulting in a minimization of LP_{min} and p at the locus, whilst w was maximized. For the low diet, the locus was maximized, resulting in a maximization of LP_{min} and p at the locus, whilst w was minimized. When the objective is to maximize ADG, it is generally of benefit to maximize the feed intake, allowing more energy to be available for growth. For the low diet the maximization of p at locus 23 is being observed, however for the high diet, low LP_{min} and high w are more important than high p.

From the point of view of the farmer, this means that the lower the quality of the feed, the more important the energy requirements for maintenance, and the more important the pig ad libitum feed intake. The high diet effectively contains a surplus of energy which is not available in the low diet situation, allowing other parameters to be improved.

Table 7.11: Optimal genome performances.
p values are for a hypothesis test at the 5% significance level.

		Optimal High		Optimal Low			Average Pig	
		Mean	**Std Dev**	**Mean**	**Std Dev**	**p**	**Mean**	**Std Dev**
Diet	**High**	1.62	0.12	1.61	0.12	0.00%	0.92	0.12
	Low	1.46	0.10	1.48	0.10	0.00%	0.92	0.10
	p	0.00%		0.00%				

The figures in Table 7.11 show that when the optimal genome for the high diet is fed the low diet, under a hypothesis test at the 5% significance level, the genome performs statistically significantly worse than the low diet optimal genome. Similarly when the optimal genome for the low diet is fed the high diet the genome performs statistically significantly worse than the high diet optimal.

For the farmer these differences in performances of the optimal diets are extremely small and probably not observable. However both optimal genotypes produce an average daily gain which is significantly above the performance of the population average. The performance of a population of random pigs is largely unaffected by the diet, however the performance of the optimal pigs improves as the diet quality increases.

7.4.4 Theoretical generations to optimal solution

This section is based on the published paper in Appendix G and is concerned with the prediction of the minimum number of theoretical generations of breeding needed to reach the optimal solution. The minimum number of generations is calculated as the difference between the mean of the population and the mean of the optimal. The difference is then divided by the expected genetic gain in the first generation of breeding. This calculation does not take into account the Bulmer effect. The Bulmer

effect, if included, would result in the genetic gain decreasing with each generation of breeding and subsequently the expected genetic gain decreasing, and hence increasing the number of generations to the optimal solution. Given that under selection the heritability will tend towards zero under the Bulmer effect, the optimal solution will likely never be achieved under selection programs. This means that the number of generation reported here is the minimum theoretical number required to reach the optimal solution.

The heritability figures of average daily gain, generated by the model, for each diet has been estimated by the simulation of 50,000 breeding pairs over one generation of breeding.

For the high diet, average daily gain has a mean of 0.924kg/d, a standard deviation of 0.119kg/d and an optimal genome average of 1.62kg/d, as shown in Figure 7.34. The heritability of average daily gain on the high diet is 0.317. Therefore, a breeding program with an i-bar (selection intensity) of 1 will have a theoretical gain per generation of 0.0379kg. This indicates that it would take a minimum of 18.4 generations of breeding, with no Bulmer effect, for the optimal solution to be reached.

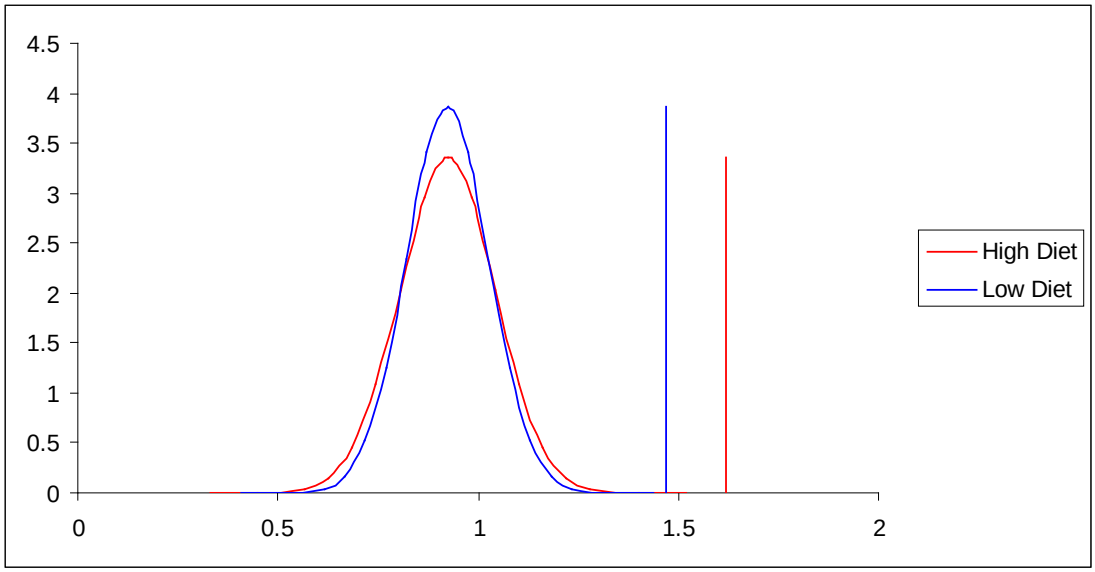

Figure 7.34: Optimal genome average plotted with population normal curve.

For the low diet, also shown in Figure 7.34, average daily gain has the same mean of 0.924kg/d, but has a standard deviation of 0.103kg/d and an optimal solution average of

1.47kg/d. The heritability for the low diet is 0.299, giving a breeding program, with an i-bar of 1, a theoretical gain per generation of 0.0307kg. Therefore this indicates that for the low diet it would take a minimum of 17.4 generation to reach the optimal, again if the Bulmer effect is ignored.

Both the low and high diet information is summarised in Table 7.12 below.

Table 7.12: Summary data

Diet	Mean	Std Dev	Heritability	Genetic gain	Optimal	Generations
High	0.924	0.119	0.317	0.0379	1.62	18.4
Low	0.924	0.103	0.299	0.0307	1.47	17.8

As can be seen, the diet has a small effect on the theoretical minimum number of generations to reach the optimal solution for average daily gain. In this case, more generations are required to reach the optimal solution on the high diet than the low diet.

7.5 Summary

In summary, it is found that if the farmer's overall objective is to maximize ADG, then the simulations produce clear evidence that there is an optimal genotype for any particular diet, which changes as the quality of the diet changes. It also needs to be taken into account that the low diet optimal solution had a performance increase when fed the high diet compared to the low diet, even though the diet had no effect on the mean performance of randomly generated pigs.

The Genetic Algorithm has the best overall performance amongst the three algorithms. It is able to produce quick, good quality solutions, and can be set to run for a longer period, which produces consistent, high quality solutions. Tabu Search also produces high quality solutions, although with slightly more variation than the Genetic Algorithm. However Tabu Search is unable to produce quick solutions. The Simulated Annealing solutions were of a lower quality when compared to the other two algorithms, although it would likely be possible to visually find the optimal when several solutions are graphed together. This could be achieved by noting, on average over multiple runs of the algorithm, the loci with alleles near the bottom of the allele range and loci with alleles near the top of the allele range. The optimal solution will

likely have the alleles at their minimum value for the loci with alleles near bottom of the allele range, and alleles at their maximum value for loci with alleles near the top of the allele range.

It has been calculated that the diet has a small effect on the theoretical minimum number of generations it would take to reach an optimal solution. More generations are required to reach the high diet optimal solution when feeding the high diet, than to reach the low diet optimal solution when feeding the low diet.

Chapter 8 – Feed conversion ratio objective

8.1 Introduction

This chapter covers the second objective investigated, to minimize feed conversion ratio. The results are presented, as in Chapter 7, in two sections, high and low diet, which are further broken down by algorithm. The results consist of the best found combination of alleles for the additive genetic structure for the given objective. The diets are compared, the theoretical minimum number of generations to the optimal solution is calculated, and the results interpreted for the practical situation. Please refer to §7.3 for a description and discussion of the graphs presented in this chapter.

8.2 Objective description

One of the many factors that pig farmers are trying to optimize or improve is feed conversion ratio (FCR). The feed conversion ratio is defined as the total feed intake over the total weight gain. The lower the FCR, the more weight the pig gains for each kilo of feed eaten, meaning the pig is more efficient at processing the diet and therefore costs less to feed. For the minimization process the starting weight is set to 20 kg and the pigs are grown to 90 kg, so that the only variable is the feed intake itself. One thousand pigs are grown to evaluate the objective for each genome investigated by the algorithms. In order to minimize feed conversion ratio within the maximization algorithms, negative feed conversion ratio has been used for the objective.

8.3 Results

8.3.1 High Diet

8.3.1.1 Genetic Algorithm

Figure 8.1 and Table 8.1 below shows that the Genetic Algorithm is performing significantly better than the Monte Carlo run. The Genetic Algorithm results lie beyond 3.5 standard deviations from the mean, and the algorithm has very little improvement in solution quality from 15,000 genomes onwards, which can be seen when the N=30, S=10 setup, which investigated an average of 15313 genomes with a mean best FCR of 1.349, standard deviation of 0.00125, and repeatability of 0.989, is compared to the

N=50, *S*=10 setup, which investigated an average of 51084 genomes for a mean best FCR of 1.347 and standard deviation of 0.00116 with repeatability of 0.987. In a very short time frame the Genetic Algorithm is producing high quality solutions.

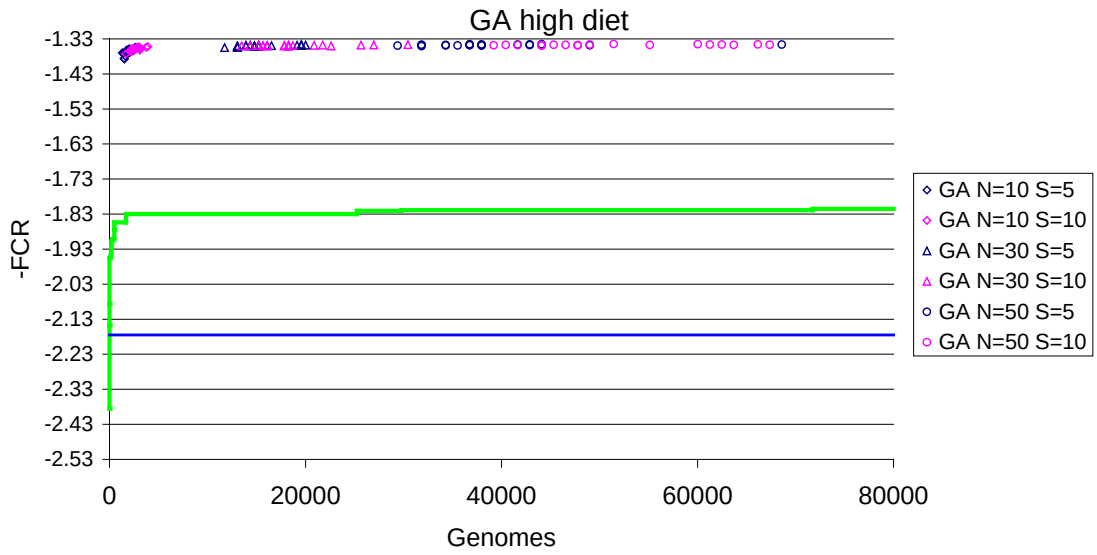

Figure 8.1: Genetic Algorithm results when feeding the high diet.
Green line is the Monte Carlo simulation results, Blue line is the mean.

Table 8.1: Genetic Algorithm results when feeding the high diet.

		Genomes Evaluated		FCR		
Configuration		**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
GA N=10	S=5	1835	328	1.367	0.00855	0.906
	S=10	2688	554	1.359	0.00515	0.935
GA N=30	S=5	15313	2459	1.350	0.00222	0.983
	S=10	19119	4433	1.349	0.00125	0.989
GA N=50	S=5	39324	8533	1.347	0.00148	0.989
	S=10	51084	9177	1.347	0.00116	0.987

The graphs below in Figure 8.2, the worst performing setup (parent population size of 10 and stopping criteria of no change for 5 iterations), and Figure 8.3, the best performing setup (parent population size of 50 and stopping criteria of no change for 10 iterations) clearly show that the algorithm is finding only one optimal solution. The improvement in solution quality between the two setups of the algorithm can clearly be seen with reduction in variation in best alleles found, which is supported by the increase

in repeatability from 0.906 to 0.987 and a decrease in standard deviation from 0.0855 to 0.00116. The graphs for the other Genetic Algorithm setups can be found in Appendix H-4.

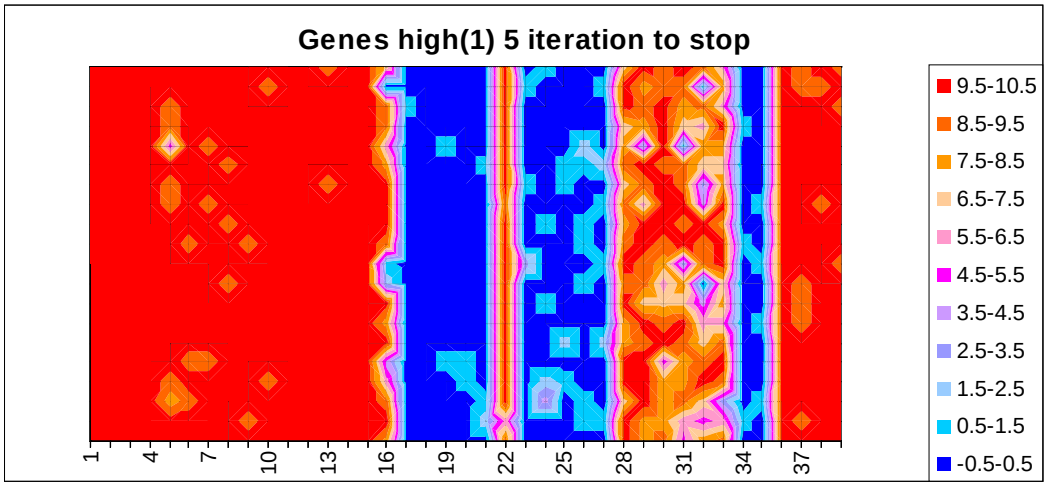

Figure 8.2: Genetic Algorithm high diet, repeatability 0.906.
Parent population size of 10 and stopping criteria of no change for 5 iterations.

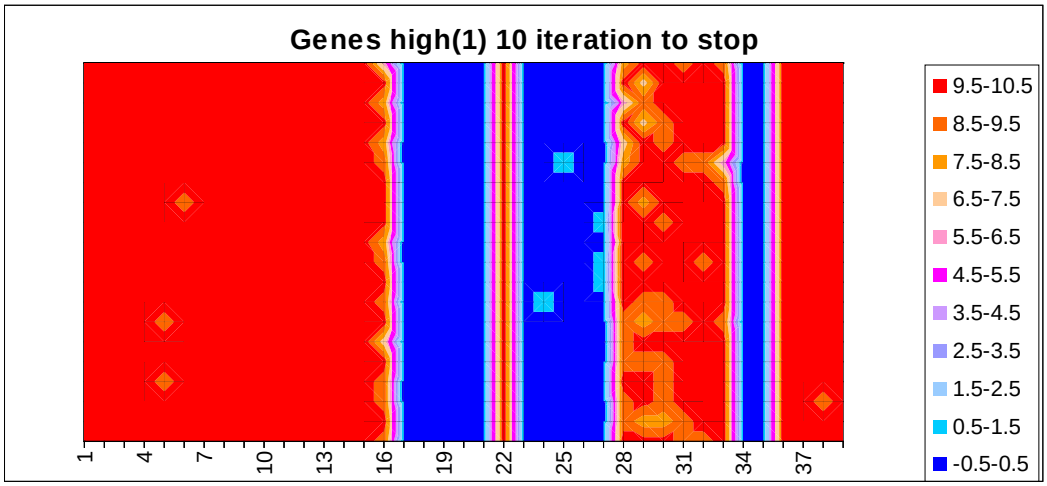

Figure 8.3: Genetic Algorithm high diet, repeatability 0.987.
Parent population size of 50 and stopping criteria of no change for 10 iterations.

8.3.1.2 Tabu Search

Figure 8.4 and Table 8.2 below shows the performance of the Tabu Search algorithm. It can be seen that Tabu Search has performed significantly better than the Monte Carlo run and is producing results that are beyond 3.5 standard deviations from the mean. There is little gain in solution quality in extending the number of genomes investigated, with the quickest running setup (N=10, E=3, and S=5) having a mean best FCR of 1.354

and the longest running setup (*N*=30, *E*=5, and *S*=10) having a mean best FCR of 1.349, although the standard deviation has been over halved from 0.00511 to 0.00206.

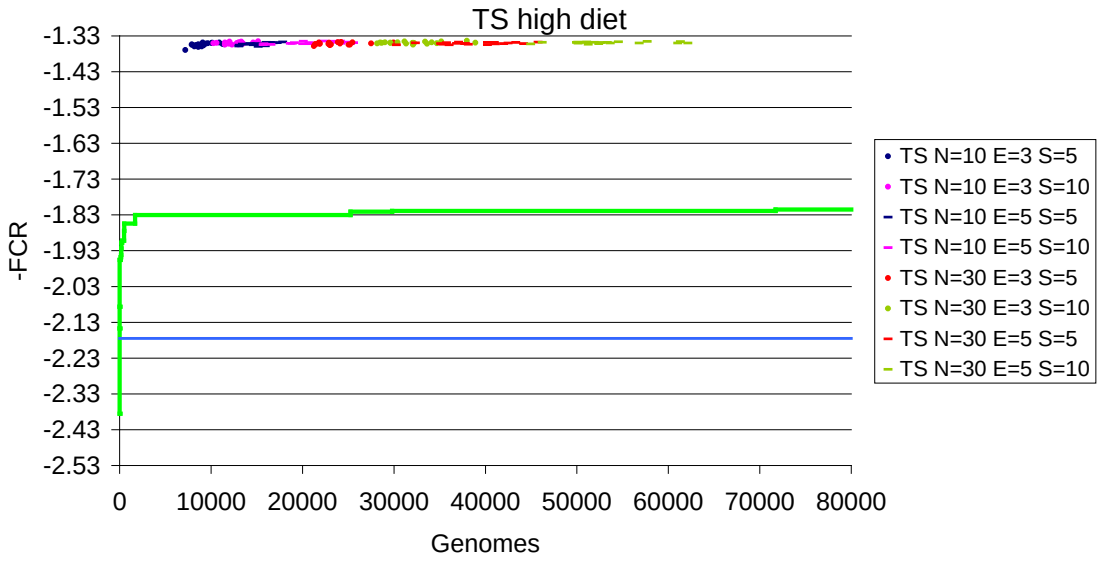

Figure 8.4: Tabu Search results when feeding the high diet.
Green line is the Monte Carlo simulation results, and the Blue line is the mean.

Table 8.2: Tabu Search results when feeding the high diet.

			Genomes Evaluated		**FCR**		
Configuration			**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
TS N=10	E=3	S=5	9162	1149	1.354	0.00511	0.931
		S=10	12703	1317	1.350	0.00263	0.959
	E=5	S=5	15369	1348	1.352	0.00358	0.947
		S=10	21193	3094	1.349	0.00319	0.958
TS N=30	E=3	S=5	23633	2189	1.352	0.00301	0.943
		S=10	32138	3004	1.349	0.00246	0.957
	E=5	S=5	38628	3983	1.351	0.00245	0.952
		S=10	53216	4623	1.349	0.00206	0.965

The graphs below in Figure 8.5, the worst performance (10 neighbours, 3 elite solutions and a stopping criteria of no change for 5 iterations), and Figure 8.6, the best performance (30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations), clearly show that the algorithm is finding only one optimal solution. The

improvement in solution quality between the two setups of the algorithm can clearly be seen with reduction in variation in the best alleles found, which is supported by the increase in repeatability from 0.931 to 0.965. The graphs for the other Tabu Search setups can be found in Appendix H-5.

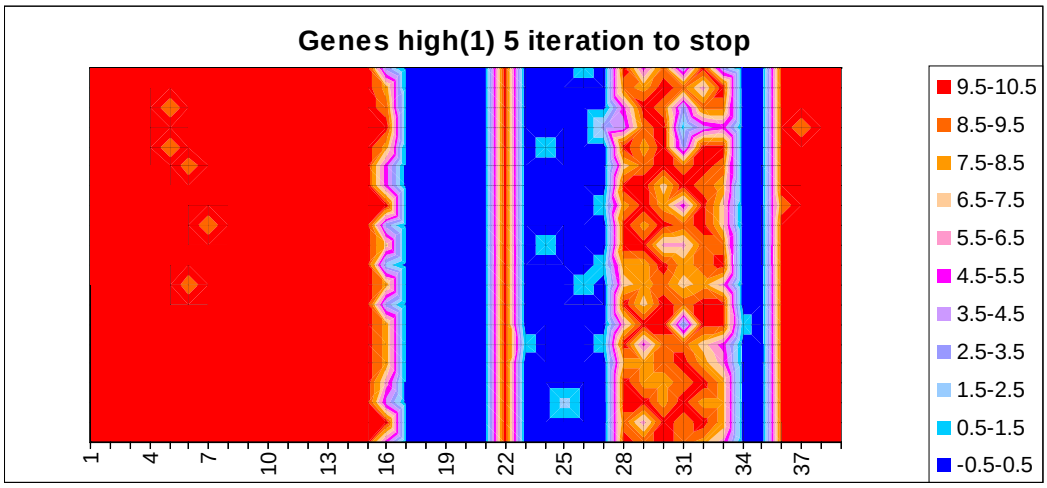

Figure 8.5: Tabu Search high diet, repeatability 0.931.
10 neighbours, 3 elite solutions and a stopping criteria of no change for 5 iterations.

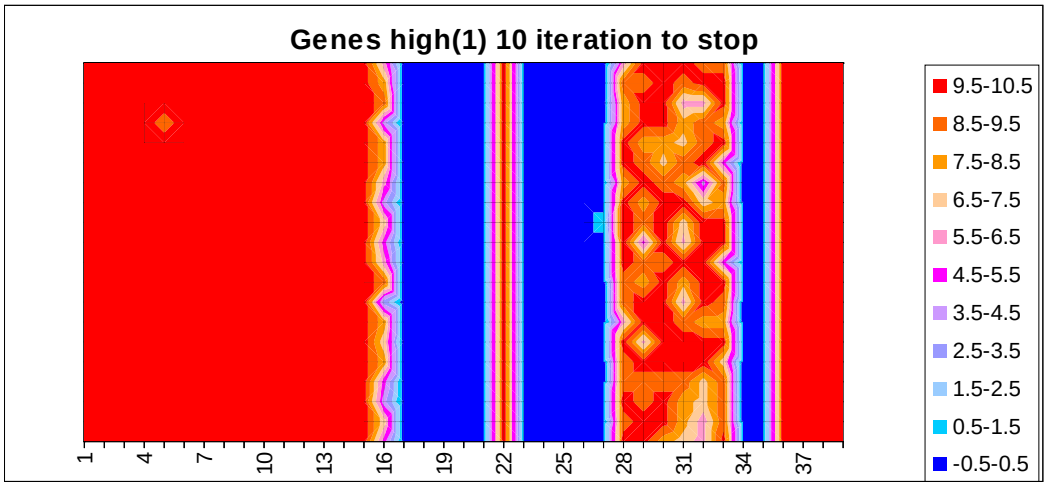

Figure 8.6: Tabu Search high diet, repeatability 0.965.
30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations.

8.3.1.3 Simulated Annealing

As shown in Figure 8.7 and Table 8.3 below, the Simulated Annealing algorithm produced fast results, but with large variation in quality of solution, with repeatability ranging from 0.796 to 0.961, standard deviation ranging from 0.01156 to 0.00245, and mean best FCR ranging from 1.427 to 1.357 for the N=10, T=5, and S=0.5 setup and

N=50, *T*=30, and *S*=0.01 setup respectively. However the solution quality does lie outside 3.5 standard deviations.

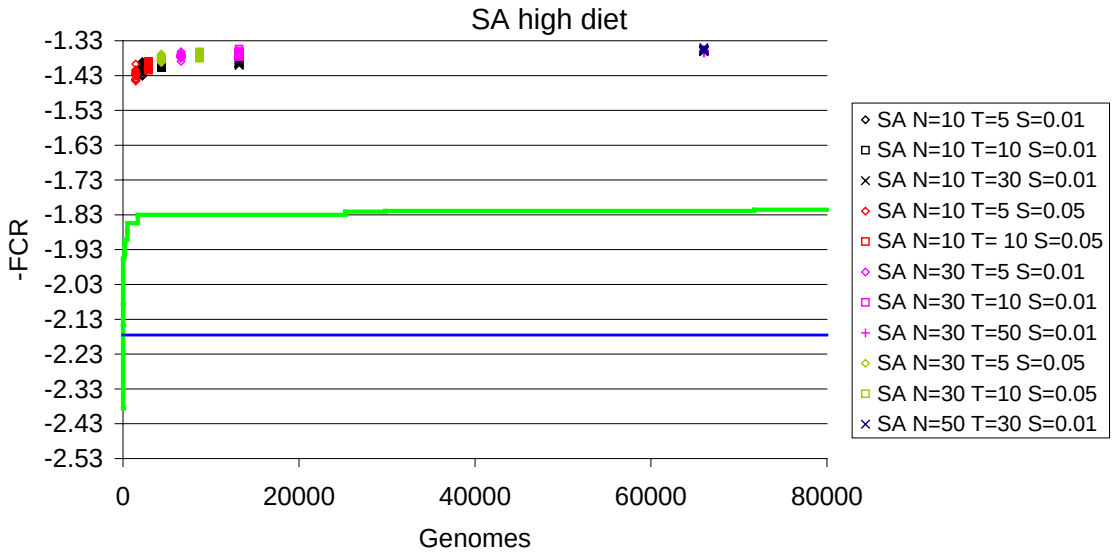

Figure 8.7: Simulated Annealing results when feeding the high diet.
Green line is the Monte Carlo simulation results, the Blue line is the mean.

Table 8.3: Simulated Annealing results when feeding the high diet.

			Genomes Evaluated		FCR		
Configuration			**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
SA N=10	T=5	S=0.01	2201	0	1.407	0.01092	0.843
	T=10		4401	0	1.397	0.00659	0.872
	T=30		13201	0	1.391	0.00554	0.900
	T=5	S=0.05	1451	0	1.427	0.01156	0.796
	T=10		2901	0	1.401	0.00755	0.855
SA N=30	T=5	S=0.01	6601	0	1.374	0.00561	0.917
	T=10		13201	0	1.367	0.00508	0.943
	T=50		66001	0	1.361	0.00262	0.960
	T=5	S=0.05	4351	0	1.380	0.00744	0.918
	T=10		8701	0	1.371	0.00514	0.928
SA N=50	T=30	S=0.01	66001	0	1.357	0.00245	0.961

Figure 8.8, the worst performance (neighbourhood size of 10, cooling time of 5 iterations and stop temperature of 0.05), shows that the worst performance is not

producing very clear or consistent results, which is supported by the low repeatability of 0.796 and standard deviation of 0.1156. However Figure 8.9, the best performance (neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01), with a repeatability of 0.961 is producing results showing a single optimal solution. However, there is still reasonable variation in the alleles. The graphs for the other Simulated Annealing setups can be found in Appendix H-6.

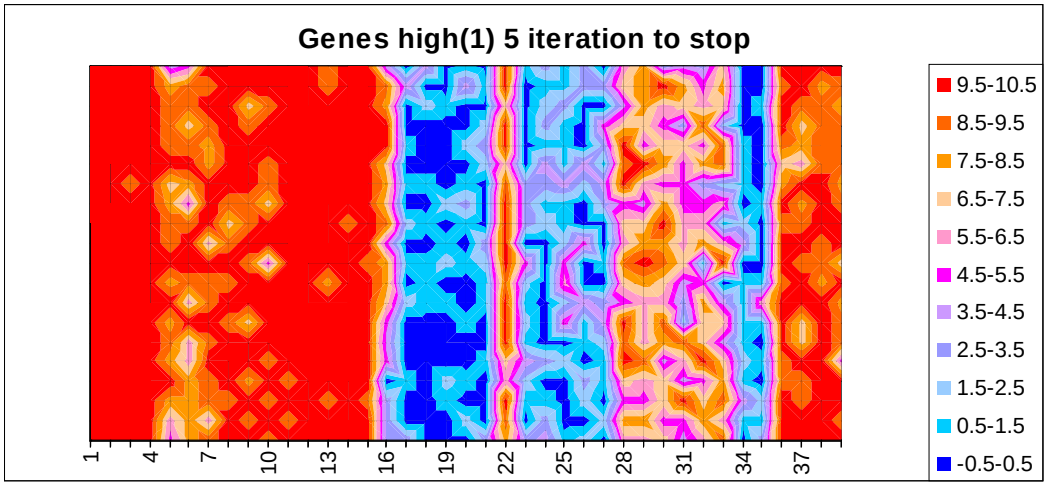

Figure 8.8: Simulated Annealing high diet, repeatability 0.796.
Neighbourhood size of 10, cooling time of 5 iterations and stop temperature of 0.05.

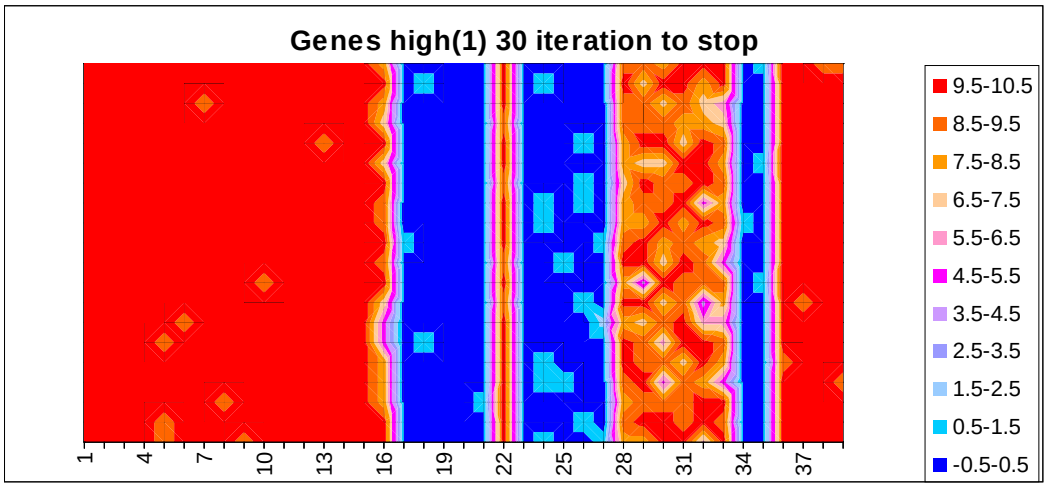

Figure 8.9: Simulated Annealing high diet, repeatability 0.961.
Neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01.

8.3.1.4 Algorithm comparison

The three algorithms are compared below in Figure 8.10. If the results are compared, for particular numbers of genomes investigated, the Genetic Algorithm gives results with less variation than the Tabu Search, although both algorithms produce approximately the same average performance. This can be seen when the Genetic Algorithm setup N=30 and S=5, which investigated on average 15313 genomes, and the Tabu Search setup N=10, E=5, and S=5, which investigated on average 15369 gnomes, are compared. These setups had mean best FCR of 1.350 and 1.352 respectively, which are very similar, but standard deviations of 0.00222 and 0.00358 respectively. Also a similar pattern can be seen when the Genetic Algorithm setup N=50 and S=10, which investigated 51084 genomes on average, and the Tabu Search setup N=30, E=5, and S=10, which investigated 53216 genomes, are compared. These two setups had mean best FCR of 1.347 and 1.349, and standard deviations of 0.00116 and 0.00206 respectively. Figure 8.10 also clearly shows that the Simulated Annealing algorithm has a lower performance level than both the Genetic Algorithm and the Tabu Search algorithm. This can be seen when the Simulated Annealing setups N=10, T=30, and S=0.01, and N=50, T=30, and S=0.01, which investigated 13201 and 66001 genomes, with mean best FCR of 1.391 and 1.357, and standard deviation of 0.00554 and 0.00245 respectively, are compared to the Genetic Algorithm and Tabu Search results above.

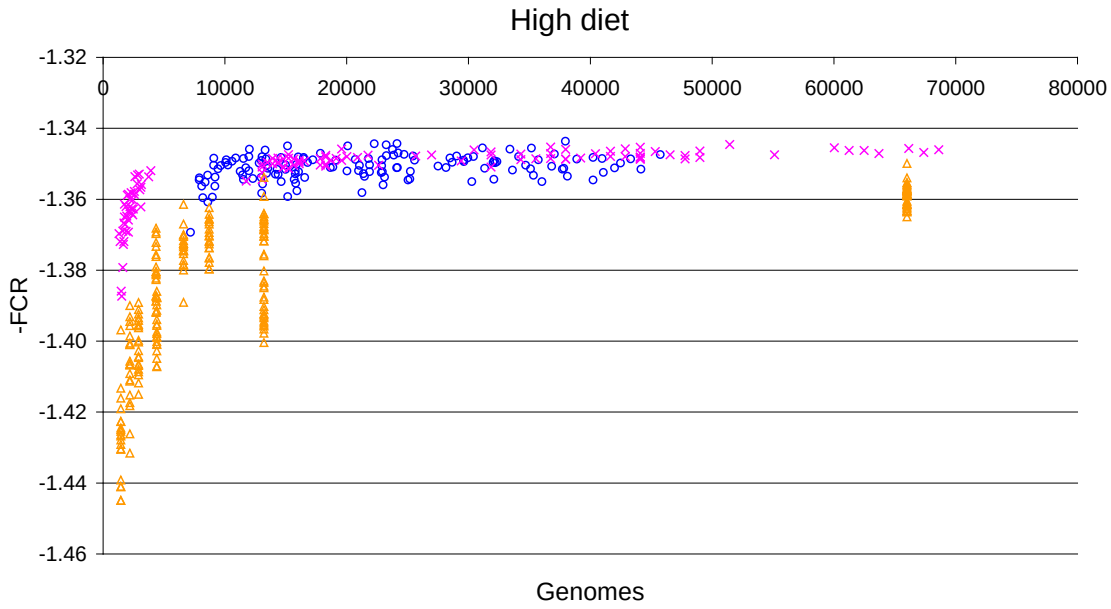

Figure 8.10: Comparison of the three algorithm results on high diet.
Pink crosses – Genetic Algorithm, Blue circles – Tabu Search, Orange triangles – Simulated Annealing.

The graphs below in Figure 8.11, Figure 8.12, and Figure 8.13 show that the algorithms are producing the same general solution. These graphs clearly show that the alleles at loci 1 to 16, 22, 28 to 33 and 36 to 39 are maximized and the alleles at loci 17 to 21, 23 to 27 and 34 to 35 are minimized to achieve optimal performance. The results are clearest in the Genetic Algorithm display, with the results being the most consistent between runs with a repeatability of 0.987. The Tabu Search had problems with loci 16 and 28 to 33. The algorithm sometimes did not always completely maximize the loci, which is shown by the ragged edge between the red and blue regions and the variegated patterns. Compare this to the straight edge between loci 21 and 22 and the solid pattern at loci 36 to 39 where all the loci are at the extreme. The Simulated Annealing also shows the optimal solution but is the most inconsistencies between runs, with a repeatability of 0.961.

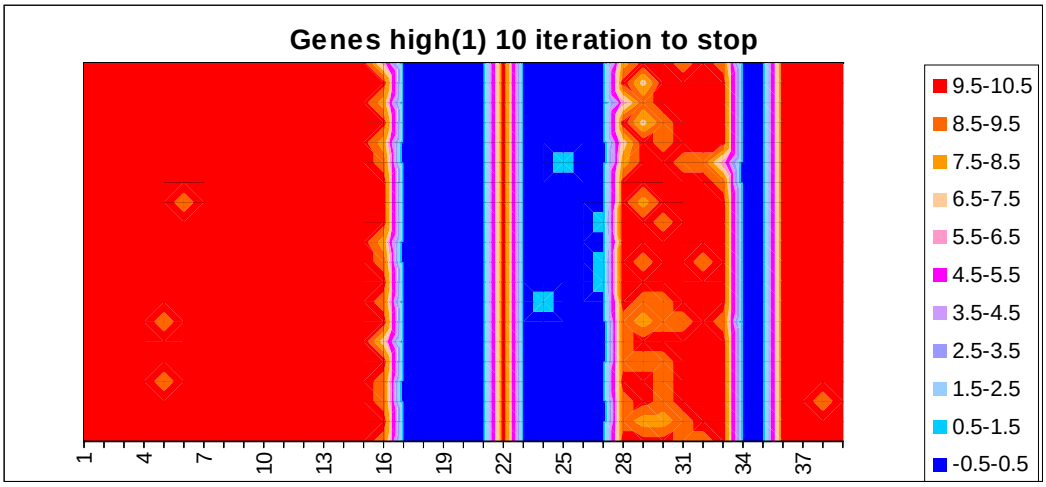

Figure 8.11: Genetic Algorithm high diet, repeatability 0.987.

Parent population size of 50 and stopping criteria of no change for 10 iterations.

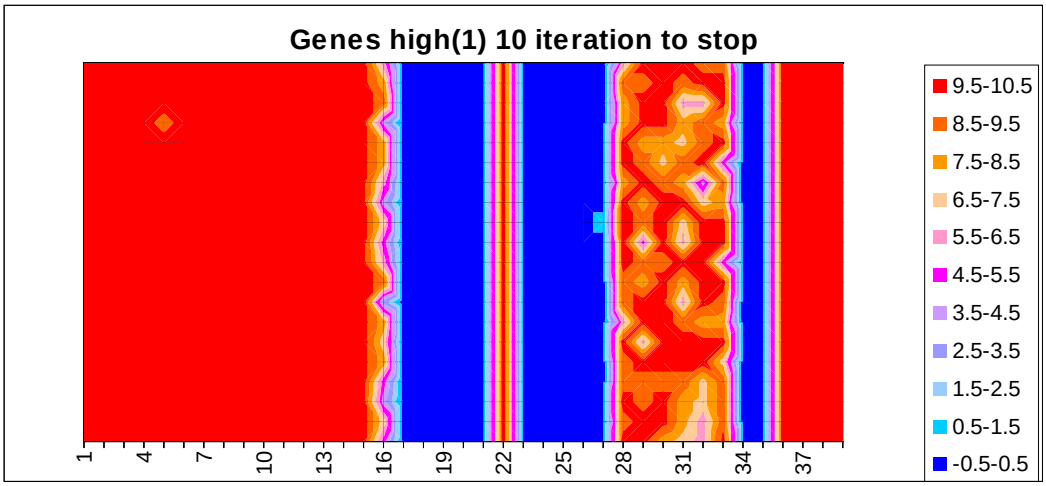

Figure 8.12: Tabu Search high diet, repeatability 0.965.

30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations.

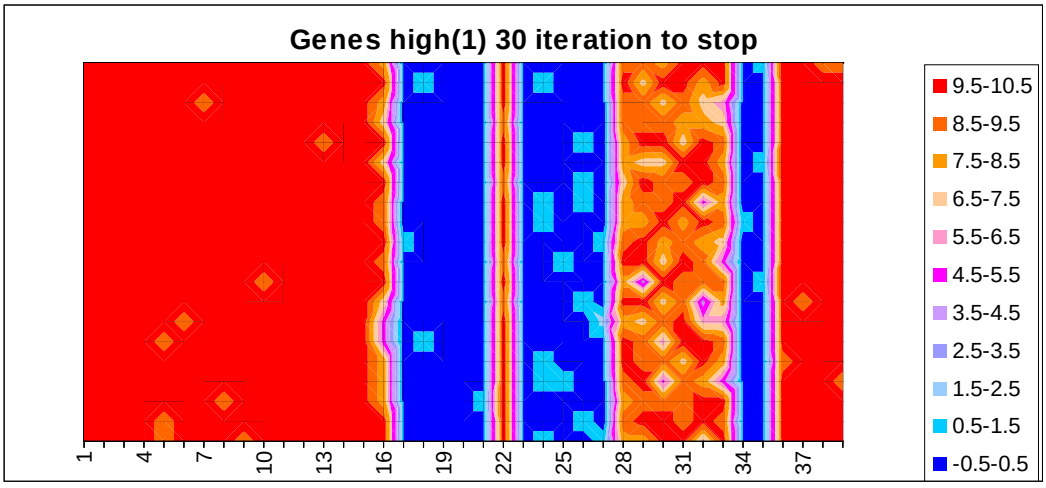

Figure 8.13: Simulated Annealing high diet, repeatability 0.961.

Neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01.

The factors for this optimal performance are shown in Table 8.4 below. This table clearly shows that Pdmax and w have been maximized and LP_{min} has been maximized with values of 281.94, 0.266, and 5.749 respectively. E_m and feed intake, p are in the middle of their ranges with 8 loci of Em maximized and 7 minimized and 7 loci of feed intake maximized and 8 minimized and values of 0.481 and 0.889 respectively. For the pig farmer this would mean that the pig has:

- high protein deposition (high muscle development)
- low lipid to protein ratio (little fat to protein ratio, i.e. a lean pig)
- average energy requirements for maintenance
- average ad libitum feed intake
- high water content.

Table 8.4: Optimal genome for feed conversion ratio with high diet being fed.

FCR optimal for High diet						
Loci	**Optimal**	Pd_{max}	LP_{min}	E_m	p	w
1-4	max	max	min			
5-6	max	max		max		
7-10	max	max			max	
11-12	max	max				max
13	max	max	min	max		
14-15	max	max	min			max
16	max		min	max		
17-21	min		min		min	
22	max		min	max		max
23	min		min		min	max
24-27	min			min		
28-30	max			min	max	
31-33	max			max		max
34-35	min				min	max
36-39	max					max
Mean		281.94	0.266	0.481	0.889	5.749

Table 8.5 below presents the performance of the optimal genome for feed conversion ratio with the high diet being fed. The mean value of 1.354 for FCR is nearly 4 standard deviations from the population mean of 2.175 presented in Table 5.30. This is an extreme value for FCR and could be attributed to the maximum possible values for LP_{min}, Pd_{max} and w in the model being possibly to large. In particular values of 281.94 for Pd_{max}, and 0.266 for LP_{min} are beyond values observed in the literature listed in Table 5.6. This combined with possibly of the correlations between LP_{min}, Pd_{max} and w being too strong has resulted in this extreme observation. As more data on these parameters become available in the literature, the model could be further calibrated.

Table 8.5: Performance of optimal genome for FCR with high diet being fed.

	FCR optimal for High diet	
	Mean	**Std Dev**
FCR	1.354	0.0908
ADG	1.4792	0.1383
BF	4.9548	1.4472
DTS	48.273	4.7708
Mortality	0	

8.3.2 Low Diet

8.3.2.1 Genetic Algorithm

The Genetic Algorithm, as shown in Figure 8.14 and Table 8.6, produces consistent results throughout the range, with repeatability ranging from 0.931 to 0.992, mean best FCR ranging from 1.887 to 1.877 and standard deviation ranging from 0.00402 to 0.00065 for the N=10 and S=5 setup and N=3 and S=10 setup respectively. These results lie beyond 3 standard deviations of the mean.

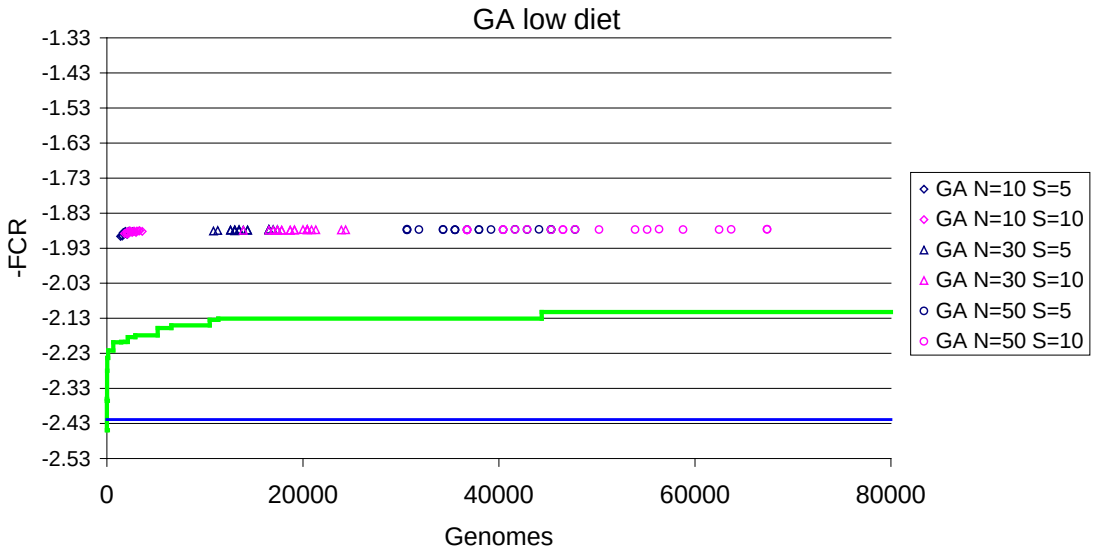

Figure 8.14: Genetic Algorithm results when feeding the low diet.
Green line is the Monte Carlo simulation results, Blue line is the mean.

Table 8.6: Genetic Algorithm results when feeding the low diet.

		Genomes Evaluated		FCR		
Configuration		**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
GA N=10	S=5	1893	294	1.887	0.00402	0.931
	S=10	2537	504	1.884	0.00318	0.923
GA N=30	S=5	14160	1932	1.878	0.00112	0.990
	S=10	19293	2446	1.878	0.00081	0.993
GA N=50	S=5	37915	5295	1.877	0.00058	0.994
	S=10	49859	10227	1.877	0.00065	0.992

The graphs below in Figure 8.15, which presents the worst performing setup (parent population size of 10 and stopping criteria of no change for 5 iterations), and Figure 8.16, the best performing setup (parent population size of 50 and stopping criteria of no change for 10 iterations), show that the algorithm is finding only one optimal solution. The improvement in solution quality between the two setups of the algorithm can clearly be seen by the reduction in variation on best alleles found, which is supported by the increase in repeatability from 0.931 to 0.992, and reduction in standard deviation from 0.00402 to 0.00065. The graphs for the other Genetic Algorithm setups can be found in Appendix H-4.

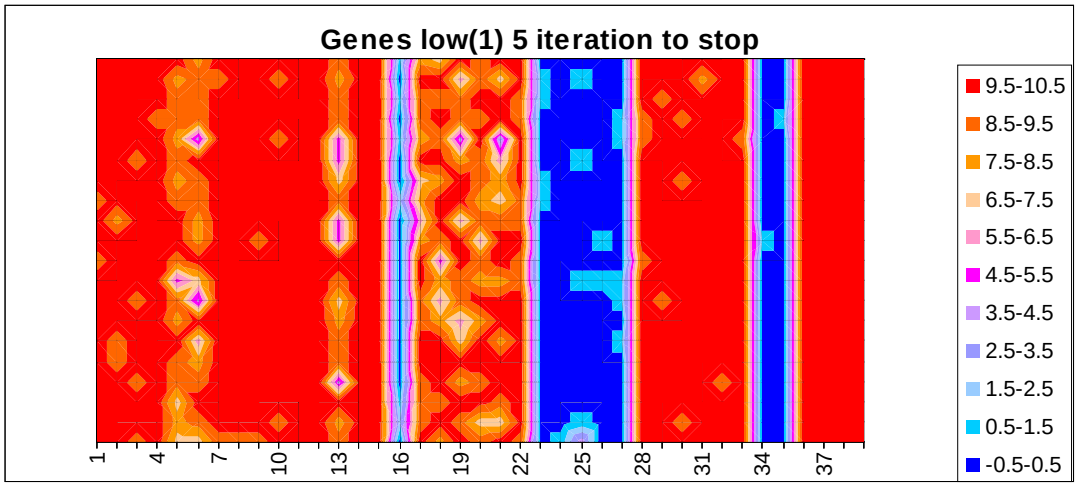

Figure 8.15: Genetic Algorithm low diet, repeatability 0.931.

Parent population size of 10 and stopping criteria of no change for 10 iterations.

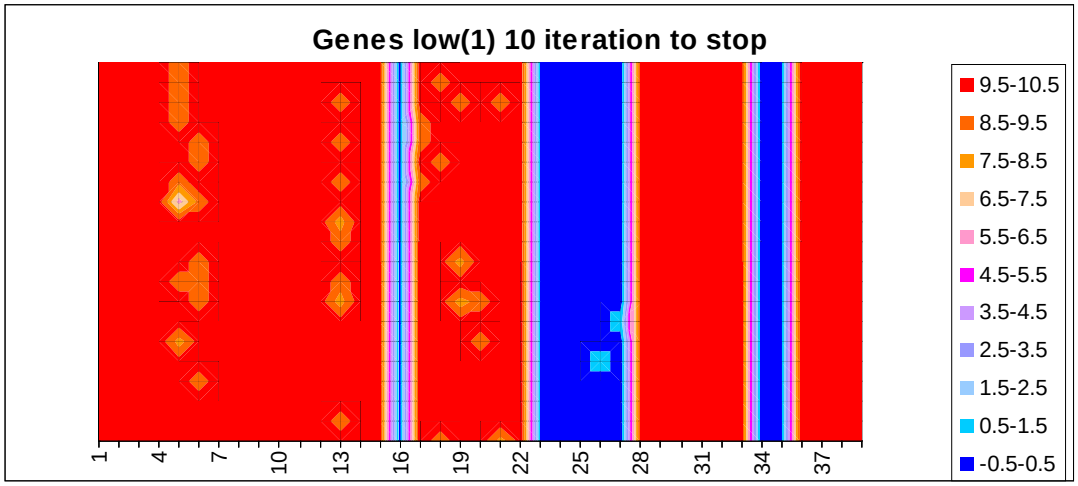

Figure 8.16: Genetic Algorithm low diet, repeatability 0.992.

Parent population size of 50 and stopping criteria of no change for 10 iterations.

8.3.2.2 Tabu Search

Figure 8.17 and Table 8.7 below shows that the Tabu Search results in a consistent and good quality solution production, with repeatability, mean best FCR, and standard deviation ranging from 0.917 to 0.986, 1.880 to 1.876, and 0.00273 to 0.00084 respectively. These solutions lie beyond 3 standard deviations from the mean.

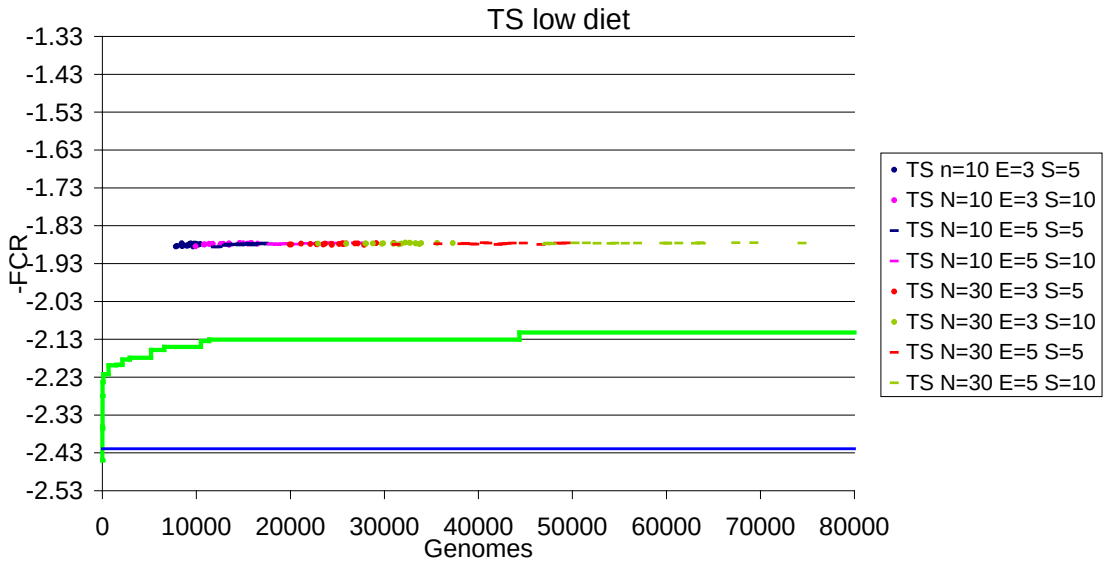

Figure 8.17: Tabu Search algorithm results when feeding the low diet. Green line is the Monte Carlo simulation results, Blue line is the mean.

Table 8.7: Tabu Search results when feeding the low diet.

			Genomes Evaluated		FCR		
Configuration			**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
TS N=10	E=3	S=5	9243	753	1.880	0.00273	0.917
		S=10	13197	1876	1.878	0.00207	0.963
	E=5	S=5	17984	1646	1.880	0.00279	0.947
		S=10	21718	2863	1.877	0.00130	0.979
TS N=30	E=3	S=5	24745	2880	1.878	0.00150	0.961
		S=10	31175	3333	1.877	0.00117	0.978
	E=5	S=5	41684	4549	1.877	0.00138	0.974
		S=10	58041	7592	1.876	0.00084	0.986

The graphs below in Figure 8.18, the worst performance (10 neighbours, 3 elite solutions and a stopping criteria of no change for 5 iterations), and Figure 8.19, the best performance (30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations), clearly show that the algorithm is finding only one optimal solution. The improvement in solution quality between the two setups of the algorithm can be seen by the reduction in the allele variation, which is supported by the increase in repeatability

from 0.917 to 0.986. The graphs for the other Tabu Search setups can be found in Appendix H-5.

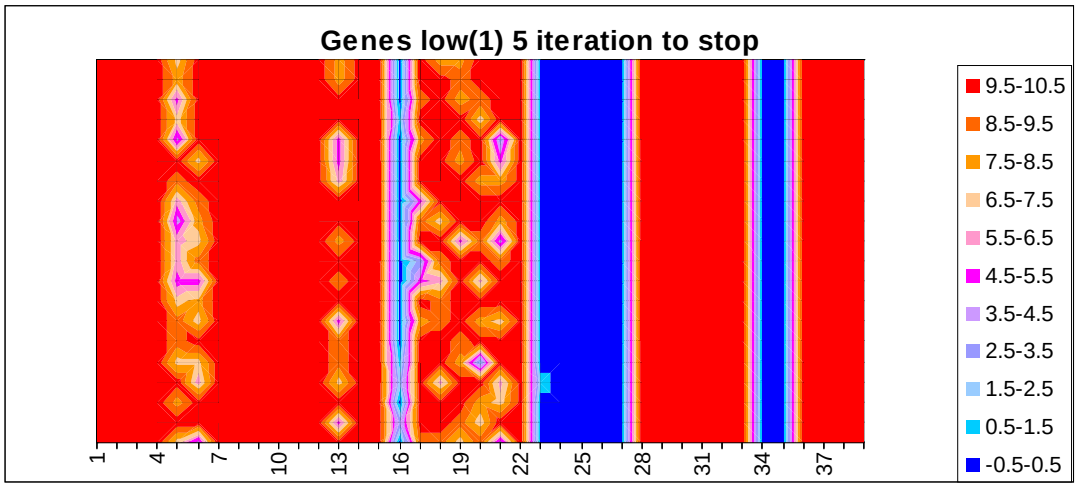

Figure 8.18: Tabu Search low diet, repeatability 0.917.
10 neighbours, 3 elite solutions and a stopping criteria of no change for 5 iterations.

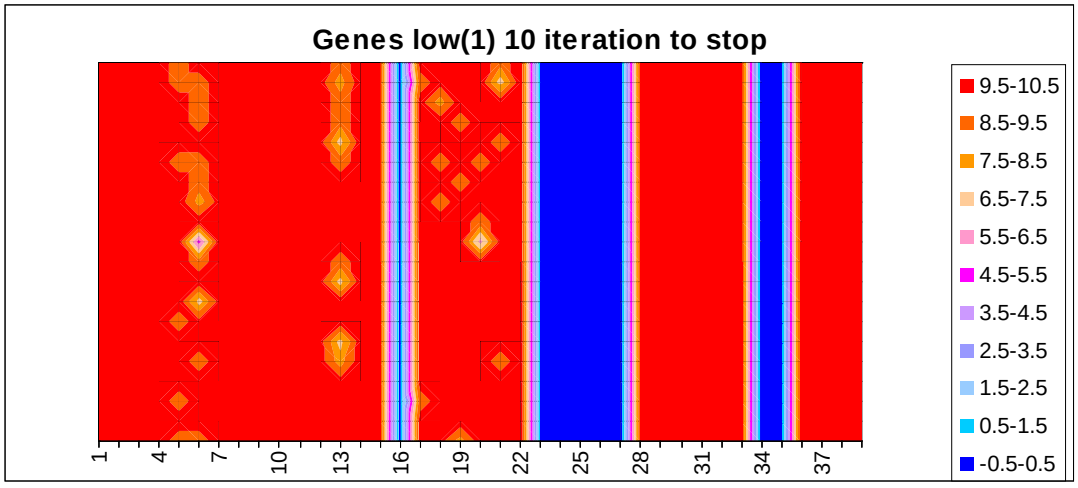

Figure 8.19: Tabu Search low diet, repeatability 0.986.
30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations.

8.3.2.3 Simulated Annealing

The Simulated Annealing results are shown below in Figure 8.20 and Table 8.8. Some variation is present, with repeatability ranging from 0.773 to 0.961 and standard deviation ranging from 0.00876 to 0.00104 for the N=10, T=5, and S=0.05 and N=50, T=30, S=0.01 setups respectively. These results lie beyond 2.5 standard deviations from the mean.

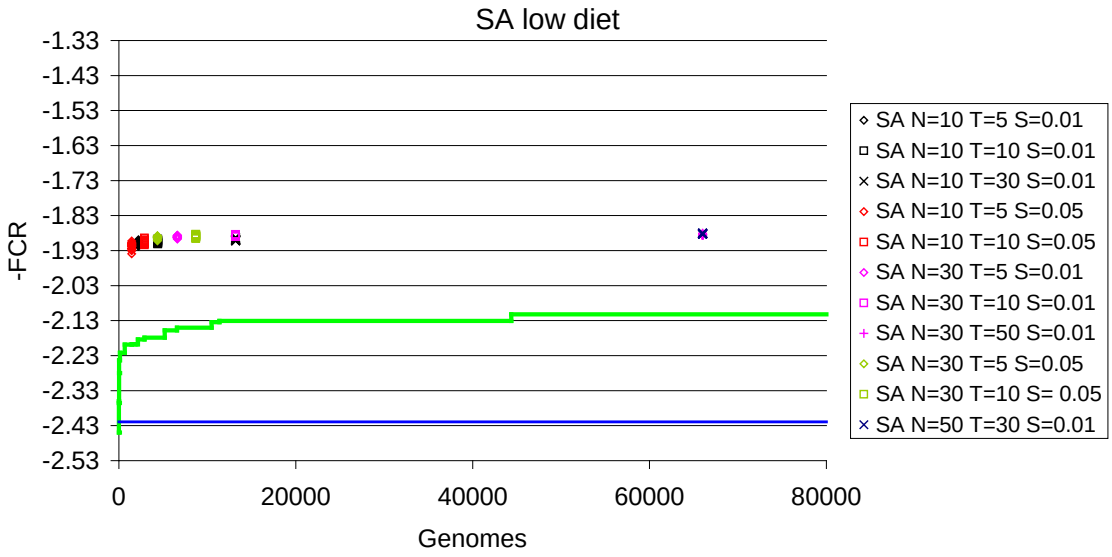

Figure 8.20: Simulated Annealing algorithm results when feeding the low diet. Green line is the Monte Carlo simulation results, Blue line is the mean.

Table 8.8: Simulated Annealing results when feeding the low diet.

			Genomes Evaluated		FCR		
Configuration			**Mean**	**Std Dev**	**Mean**	**Std Dev**	**repeatability**
SA N=10	T=5	S=0.01	2201	0	1.908	0.00496	0.811
	T=10		4401	0	1.904	0.00413	0.846
	T=30		13201	0	1.899	0.00268	0.882
	T=5	S=0.05	1451	0	1.918	0.00876	0.773
	T=10		2901	0	1.904	0.00479	0.849
SA N=30	T=5	S=0.01	6601	0	1.891	0.00290	0.890
	T=10		13201	0	1.887	0.00161	0.933
	T=50		66001	0	1.884	0.00146	0.941
	T=5	S=0.05	4351	0	1.893	0.00388	0.878
	T=10		8701	0	1.889	0.00328	0.918
SA N=50	T=30	S=0.01	66001	0	1.882	0.00104	0.961

Figure 8.21, the worst performing setup (neighbourhood size of 10, cooling time of 5 iterations and stop temperature of 0.05), shows that the worst performance is not producing very clear or consistent results, which is supported by the low repeatability of 0.773 and standard deviation of 0.00876. However Figure 8.22, the best performance (neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01),

with a repeatability of 0.961 is producing results that show a single optimal solution, even though there is still reasonable variation present in the best alleles found. The graphs for the other Simulated Annealing setups can be found in Appendix H-6.

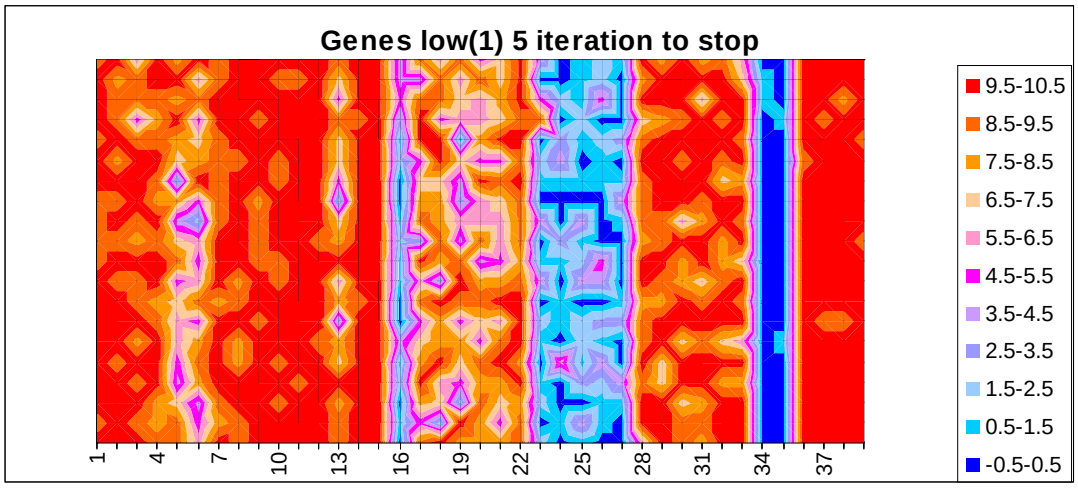

Figure 8.21: Simulated Annealing low diet, repeatability 0.773.
Neighbourhood size of 10, cooling time of 5 iterations and stop temperature of 0.05.

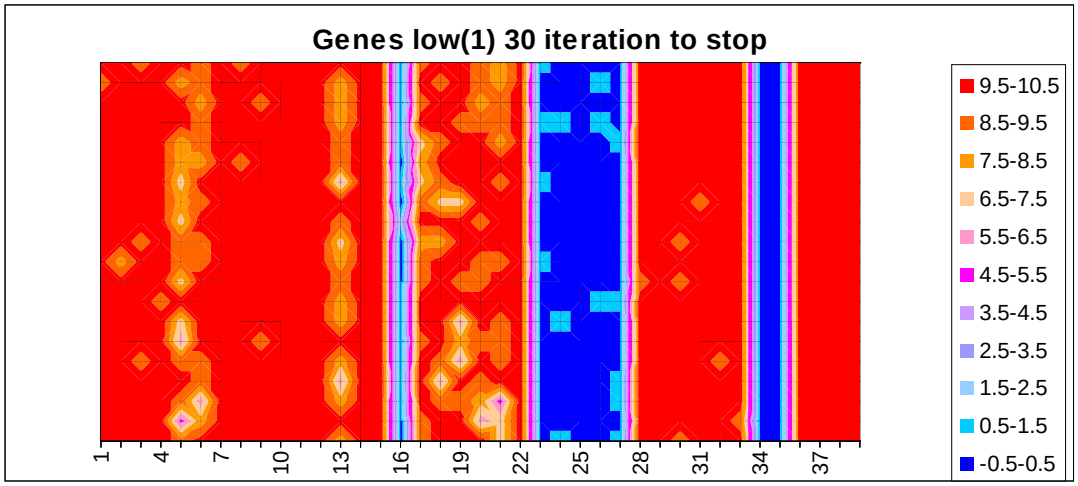

Figure 8.22: Simulated Annealing low diet, repeatability 0.961.
Neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01.

8.3.2.4 Algorithm comparison

The results from all three algorithms are plotted below in Figure 8.23. It is clear that the Genetic Algorithm and Tabu Search algorithm produced very similar results, which can be seen by the mean best FCR's of 1.878 and 1.878, and 1.877 and 1.876 for the Genetic Algorithm setup N=30, S=5, Tabu Search setup, N=10, E=3, S=10, Genetic algorithm setup N=50, S=10, and Tabu Search setup N=30, E=5, S=10 respectively, which investigated 14160, 13197, 49859, and 58041 genomes on average respectively.

However the Tabu Search solutions have more variation which can be seen when the standard deviations of 0.00112, 0.00207, 0.00065, and 0.00084 respectively for the above setup are compared and the repeatabilities of 0.990, 0.963, 0.992 and 0.986 respectively are also compared. Both these algorithms outperform the Simulated Annealing algorithm. This can be seen when the mean best FCR's of 1.899 and 1.882, standard deviations of 0.00268 and 0.00104, and repeatabilities of 0.882 and 0.961, for the Simulated Annealing setups of N=10, T=30, S=0.01, and N=50, T=30, S=0.01, which evaluated 13201 and 66001 genomes respectively, are respectively compared to the Genetic Algorithm and Tabu Search results above.

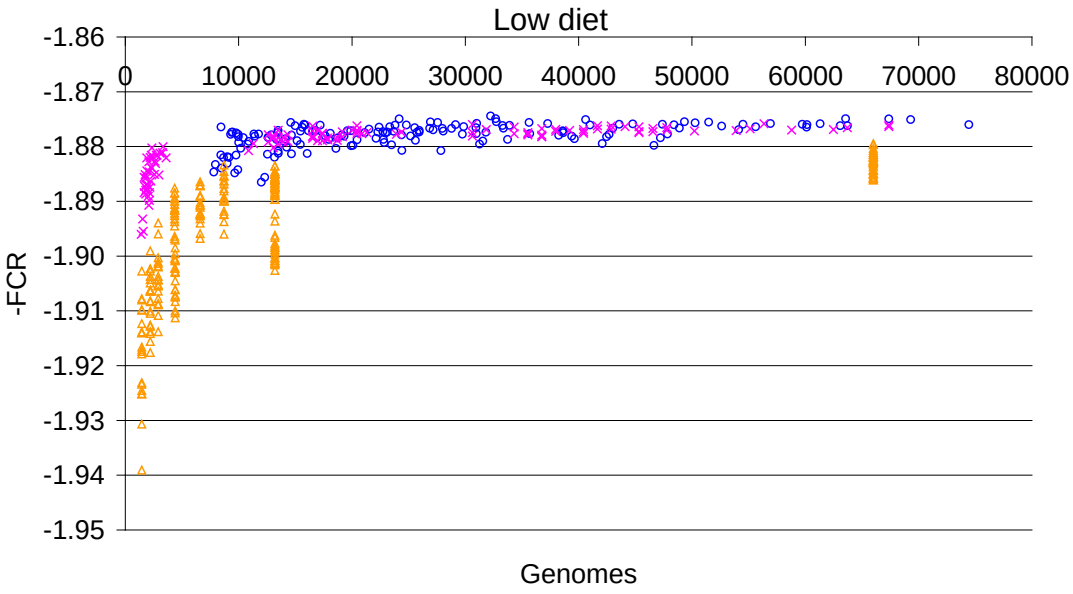

Figure 8.23: Comparison of the three algorithm results on low diet.
Pink crosses – Genetic Algorithm, Blue circles – Tabu Search, Orange triangles – Simulated Annealing.

The graphs in Figure 8.24, Figure 8.25, and Figure 8.26 show that, again, the algorithms are converging to a common solution. For optimal performance, the alleles at loci 1 to 15, 17 to 22, 28 to 33 and 36 to 39 are maximized, and loci 16, 23 to 27 and 34 to 35 are minimized. The Genetic Algorithm has produced the most consistent results, and hence the clearest graphical representation with a repeatability of 0.992. All three algorithms had some variation in the alleles in the first 22 loci, with the simulated annealing having the most and the Genetic Algorithm the least. As was seen with the high diet results there is significant variation in the alleles in the Simulated Annealing solutions.

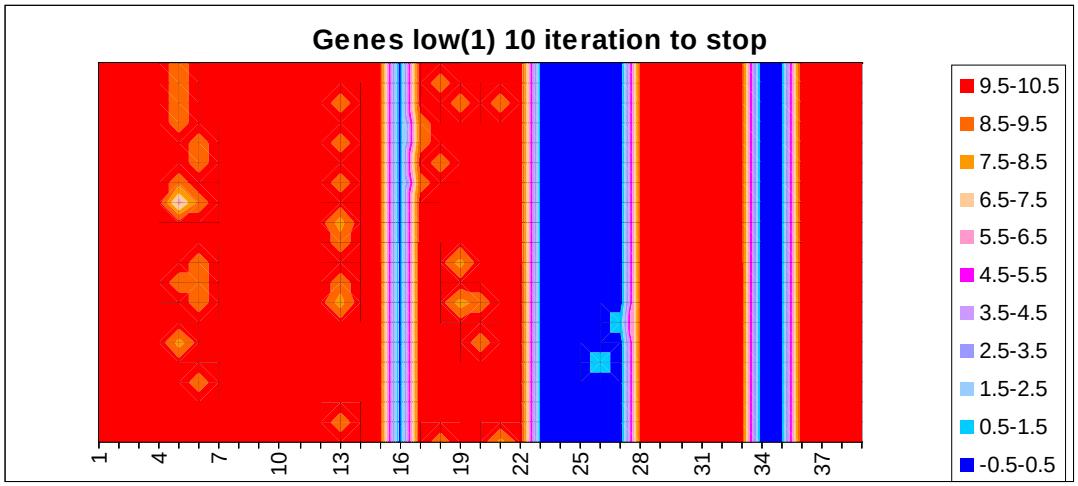

Figure 8.24: Genetic Algorithm low diet, repeatability 0.992.

Parent population size of 50 and stopping criteria of no change for 10 iterations.

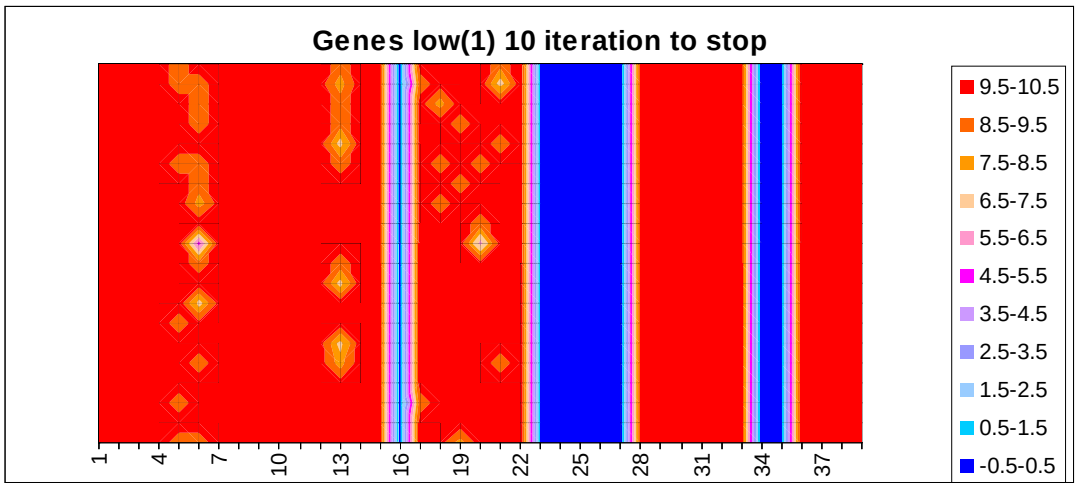

Figure 8.25: Tabu Search low diet, repeatability 0.986.

30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations.

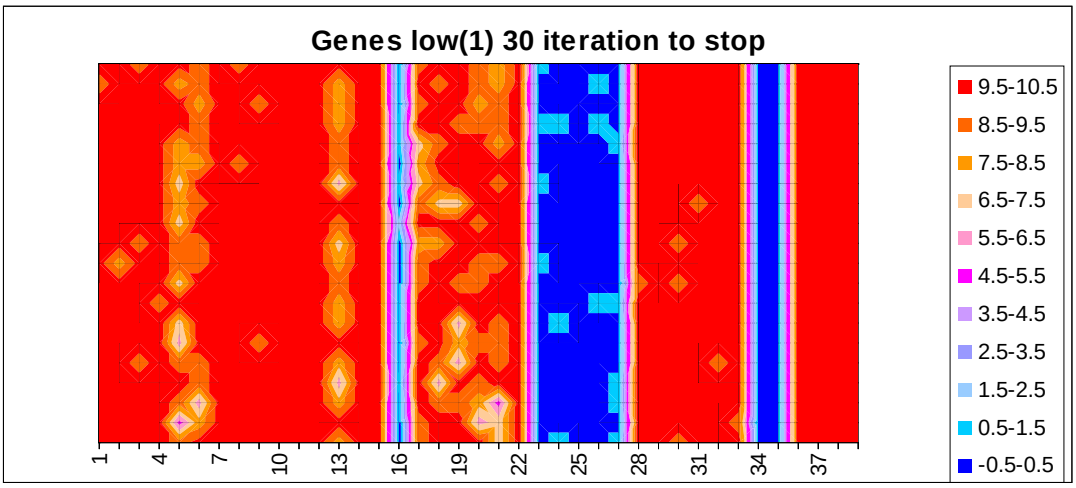

Figure 8.26: Simulated Annealing low diet, repeatability 0.961.
Neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01.

The factors for this optimal performance are shown in Table 8.9 below. This table clearly shows that Pd_{max} and w have been maximized with values of 281.94 and 5.749 respectively, 9 loci have been minimized for LP_{min} and 6 have been maximized giving a value of 0.62, 7 loci of E_m have been maximized and 8 have been minimized for a value of 0.468, and 3 loci have been minimized for feed intake, p, and 12 have been maximized for a value of 1.077. For the pig farmer this would mean that the pig had:

- high protein deposition (high muscle development)
- below average lipid to protein ratio (little fat to protein ratio, i.e. a lean pig)
- average energy requirements for maintenance
- above average ad libitum feed intake (the pig is willing to eat a lot of food)
- high water content.

Table 8.9: Optimal genome for feed conversion ratio with low diet being fed.

FCR optimal for Low diet						
Loci	**Optimal**	Pd_{max}	LP_{min}	E_m	p	w
1-4	max	max	min			
5-6	max	max		max		
7-10	max	max			max	
11-12	max	max				max
13	max	max	min	max		
14-15	max	max	min			max
16	min		max	min		
17-21	max		max		max	
22	max		min	max		max
23	min		min		min	max
24-27	min			min		
28-30	max			min	max	
31-33	max			max		max
34-35	min				min	max
36-39	max					max
Mean		281.94	0.62	0.468	1.077	5.749

Table 8.10 below presents the performance of the optimal genome for feed conversion ratio with the low diet being fed. The mean value of 1.8799 for FCR is over 3 standard deviations from the population mean of 2.419 presented in Table 5.30. This is less extreme value for FCR than that seen for the high diet, despite Pd_{max} of 281.94, and w of 5.749 being beyond values observed in the literature listed in Table 5.6. This indicates that low quality diet significantly affects the ability to observe extreme FCR values.

Table 8.10: Performance of optimal genome for FCR with low diet being fed.

	FCR optimal for Low diet	
	Mean	**Std Dev**
FCR	1.8799	0.0444
ADG	1.42	0.1081
BF	12.576	0.8487
DTS	50.185	3.8599
Mortality	0	

8.3.3 Diet comparison

Figure 8.27 shows that the comparative performance between the three algorithms is unaffected by the type of diet that is being fed, although the performance of the pigs is significantly affected, as would be expected.

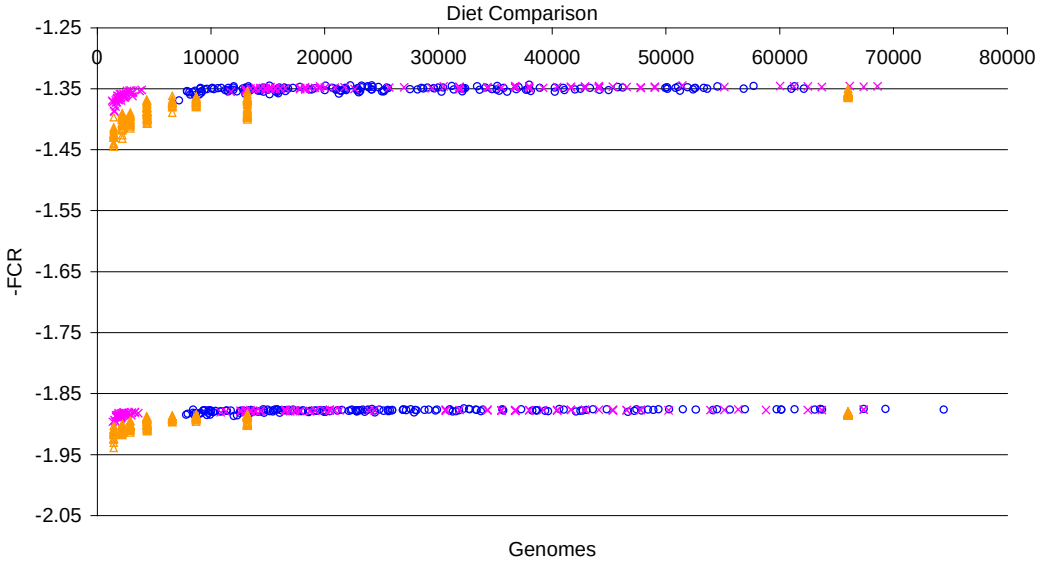

Figure 8.27: Comparison of the three algorithm results on low diet and high diet. Pink crosses – Genetic Algorithm, Blue circles – Tabu Search, Orange triangles – Simulated Annealing.

Figure 8.28 below provides a visual comparison of the optimal results for the high and low diet when optimizing feed conversion ratio. The results shown are from the Genetic Algorithm, which had the least variation in alleles of the three algorithms.

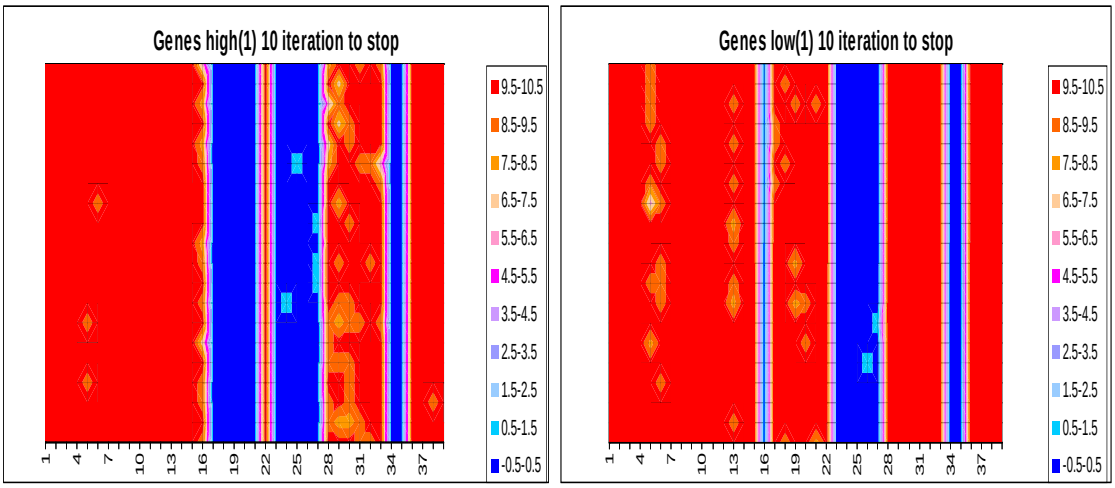

Figure 8.28: Genetic Algorithm diet comparison.
Parent population size of 50 and stopping criteria of no change for 10 iterations. High diet results are on the left and low diet results on the right.

The opposing diets produce results that differ at loci 16 to 21. Locus 16 is maximized for the high diet but minimized for the low diet. Locus 16 is negatively correlated to LP_{min} and positively correlated to E_m. This indicates that for the high diet, at this locus, LP_{min} has been minimized and E_m maximized, but for the low diet E_m had been minimized and LP_{min} maximized. Generally it would be of benefit to minimize energy requirement for maintenance when minimizing FCR, as this will generally allow more for more energy in the diet to be used for growth. It is also beneficial to minimize LP_{min} to allow better partitioning between lipids and proteins. In the low diet, where energy is very limiting, reducing the energy requirements for maintenance provides better growth benefit, and therefore better FCR than reducing LP_{min}. However in the high diet, where there is a surplus of energy in the diet, reducing LP_{min} provides better benefits the reducing E_m. This is because the pig is already eating surplus energy which can be better utilised when LP_{min} is lower, but reducing E_m only increases the surplus energy.

The other loci, 17 to 21 affected by the diet are positively correlated to LP_{min} and feed intake, p. For the high diet, the loci were minimized, resulting in a minimization of LP_{min} and feed intake at the loci. For the low diet, the loci were maximized, resulting in a maximization of LP_{min} and feed intake at the locus. Generally in minimizing FCR it is beneficial to minimize LP_{min} and also reduce p to reduce excess eating that is providing little growth benefit. For the high diet at these loci, since there is excess energy in the diet, it is possible to reduce p with little effect on growth and this gives the benefit of

reduced LP_{min}. However for the low diet, where energy is restricting, increasing p allows more energy to be available for growth, which is more than the benefits lost by increasing LP_{min}.

Table 8.11: Optimal genome performances.
p values are for a hypothesis test at the 5% significance level.

		Optimal High		Optimal Low			Average Pig	
		Mean	**Std Dev**	**Mean**	**Std Dev**	**p**	**Mean**	**Std Dev**
Diet	**High**	1.35	0.09	1.52	0.09	0.00%	2.18	0.21
	Low	1.95	0.05	1.88	0.04	0.00%	2.42	0.17
	P	0.00%		0.00%				

The figures in Table 8.11 show that when the optimal genome for the high diet is fed the low diet, the genome performs statistically significantly worse than the low diet optimal genome. Similarly when the optimal genome for the low diet is fed the high diet the genome performs statistically significantly worse than the high diet optimal. From the farmer's point of view these differences in performances of the optimal diets are small but would likely be observable. Both optimal genotypes produce a feed conversion ratio which is significantly below the performance of the population average on both diets.

8.3.4 Theoretical generations to optimal solution

As in §7.4.4, this section is concerned with the prediction of the minimum number of theoretical generations of breeding needed to reach the optimal solution. The number of generations is calculated as the difference between the mean of the population and the mean of the optimal. The difference is then divided by the genetic gain. This calculation does not take into account the Bulmer effect. The Bulmer effect, if included, would result in the genetic gain decreasing with each generation of breeding, and hence increasing the number of generations to the optimal solution.

The heritability figures of feed conversion ratio, generated by the model, for each diet has been estimated by the simulation of 50,000 breeding pairs over one generation of breeding.

For the high diet, feed conversion ratio has a mean of 2.18, a standard deviation of 0.207 and an optimal genome average of 1.35, as shown in Figure 8.29. The heritability of feed conversion ratio on the high diet is 0.389. This then gives a breeding program, with an i-bar (selection intensity) of 1, a theoretical gain per generation of 0.0806. The indication is thus that it would take a minimum of 10.3 generation of breeding, with no Bulmer effect, for the optimal solution to be reached.

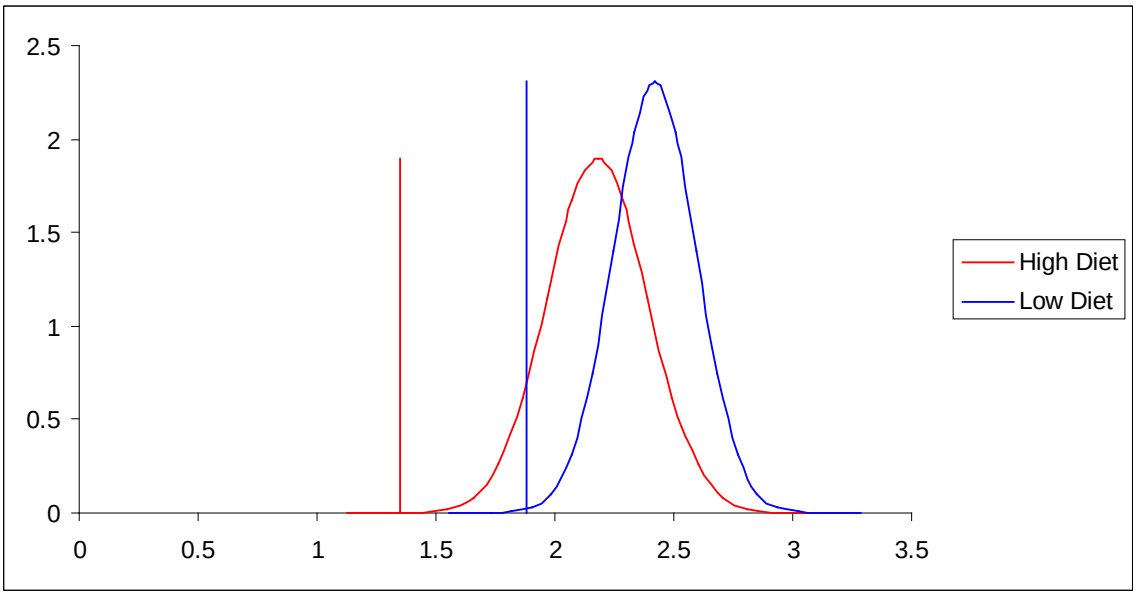

Figure 8.29: Optimal genome average plotted with population normal curve.

For the low diet, also shown in Figure 8.29, feed conversion ratio has a mean of 2.42, a standard deviation of 0.173 and an optimal solution average of 1.88. The heritability for the low diet is 0.393, giving a breeding program, with an i-bar of 1, a theoretical gain per generation of 0.0680. Therefore this indicates that for the low diet it would take a minimum of 7.9 generation to reach the optimal, again if the Bulmer effect is ignored.

The information for both the low and high diets is summarised in Table 8.12 below.

Table 8.12: Summary data

Diet	Mean	Std Dev	Heritability	Genetic gain	Optimal	Generations
High	2.18	0.207	0.389	0.0806	1.35	10.3
Low	2.42	0.173	0.393	0.0680	1.88	7.9

It is clear that the diet has an effect on the minimum theoretical number of generations to reach the optimal solution for feed conversion ratio. In this case, it takes more generations to reach the optimal solution on the high diet than the low diet.

8.4 Summary

When the overall objective is to minimize feed conversion ratio, the algorithms produce clear evidence that there is an optimal genotype for any particular diet. The optimal genotype changes as the quality of the diet changes. Generally, for any particular genotype, the high diet produces better feed conversions ratios than the low diet.

The Genetic Algorithm has the best overall performance of the three algorithms. The Genetic Algorithm produces solutions less variation but the same or better quality as the other two algorithms. Quick, good quality solutions are also an option with the Genetic Algorithm. Tabu Search produces the same high quality solutions as the Genetic Algorithm, but with more variability. However Tabu Search is unable to produce quick solutions. The Simulated Annealing solutions are of a lower quality in terms of variation and genetics, when compared to the other two algorithms.

When the theoretical minimum number of generations to the optimal solution is compared for the two diets, more generations are required on the high diet than the low diet.

Chapter 9 – Back fat objective

9.1 Introduction

Chapter 9 describes the third objective investigated, that of minimizing back fat. The results are presented, as in Chapter 7, in two sections, high and low diet, which are further broken down by algorithm. The results consist of the best found combination of alleles for the additive genetic structure for the given objective. The diets are compared and the theoretical number of generations to the optimal solution is calculated. The practical farming perspective for this objective is also presented. Please refer to §7.3 for a description and discussion of the graphs presented in this chapter.

9.2 Objective description

The third factor that pig farmers are working to optimize or improve is Back Fat (BF). Back Fat is defined as the depth of fat, in millimetres, with the measurement being taken at a specific point on the back of the pig. Lower BF values tend to achieve higher carcass prices than those with higher BF values. The objective is therefore to minimize BF. For the minimization process the starting weight is set to 20kg and the pigs are grown to 90kg, as for the other factors. One thousand pigs are grown to evaluate the objective. In order to minimize BF within the maximization algorithms, negative BF has been used for the objective.

9.3 Results

9.3.1 High Diet

9.3.1.1 Genetic Algorithm

Figure 9.1 and Table 9.1 below shows that the Genetic Algorithm is performing significantly better than the Monte Carlo run. When the high diet is fed, the Genetic Algorithm results lie beyond 3 standard deviations from the mean. It can also be seen that the algorithm has very little improvement in solution quality from 15,000 genomes onwards, and in a very short time frame it is producing high quality solutions. This can be seen by comparing the N=30, S=5 setup, which investigated an average of 14116 genomes, with a mean best BF of 3.285mm, standard deviation of 0.0453mm and

repeatability of 0.932, with the *N*=50, *S*=10 setup, which investigated an average of 55922 genomes, with a mean best BF of 3.194mm, standard deviation of 0.0247mm and repeatability of 0.984. The Genetic Algorithm thus appears to become reliable, for this objective, in a very short time frame.

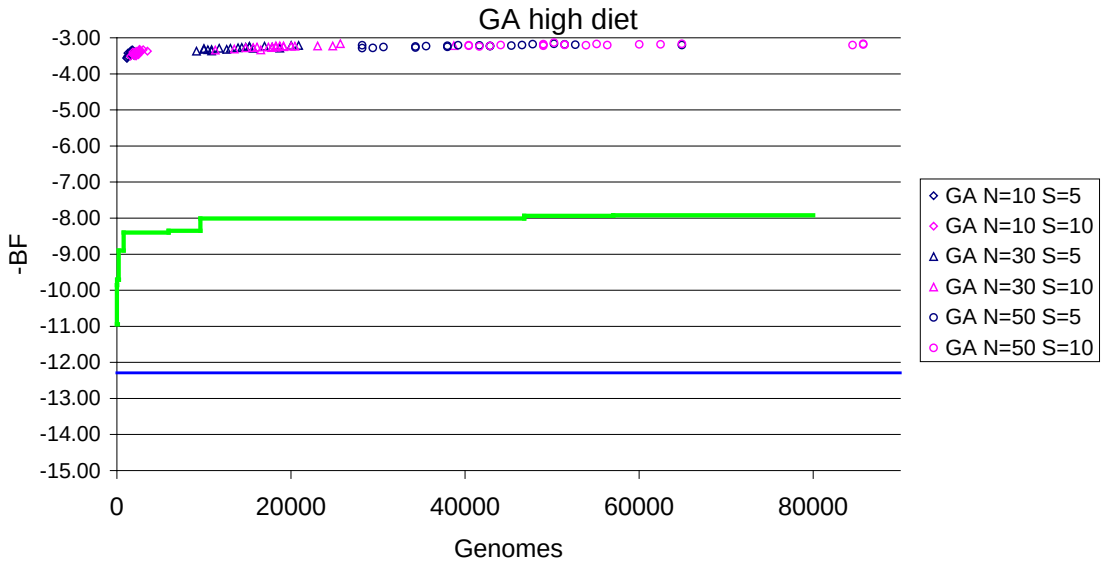

Figure 9.1: Genetic Algorithm results when feeding the high diet.
Green line is Monte Carlo simulation results, Blue line is the mean.

Table 9.1: Genetic Algorithm results when feeding the high diet.

		Genomes Evaluated		BF		
Configuration		**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
GA N=10	S=5	1569	260	3.449	0.0617	0.786
	S=10	2296	453	3.418	0.0567	0.803
GA N=30	S=5	14116	3487	3.285	0.0453	0.932
	S=10	19259	5727	3.252	0.0418	0.945
GA N=50	S=5	41100	9509	3.224	0.0334	0.967
	S=10	55922	14425	3.194	0.0247	0.984

The graphs below in Figure 9.2, the worst performance (parent population size of 10 and stopping criteria of no change for 5 iterations), and Figure 9.3, the best performance (parent population size of 50 and stopping criteria of no change for 10 iterations), clearly show that the algorithm is finding only one optimal solution. The improvement

in solution quality between the two setups of the algorithm can clearly be seen with reduction in variation of reported best alleles, which is supported by the increase in repeatability from 0.786 to 0.984, reduction in standard deviation from 0.0617mm to 0.0247mm, and reduction in mean best BF from 3.449mm to 3.194mm. The graphs for the other Genetic Algorithm setups can be found in Appendix H-7.

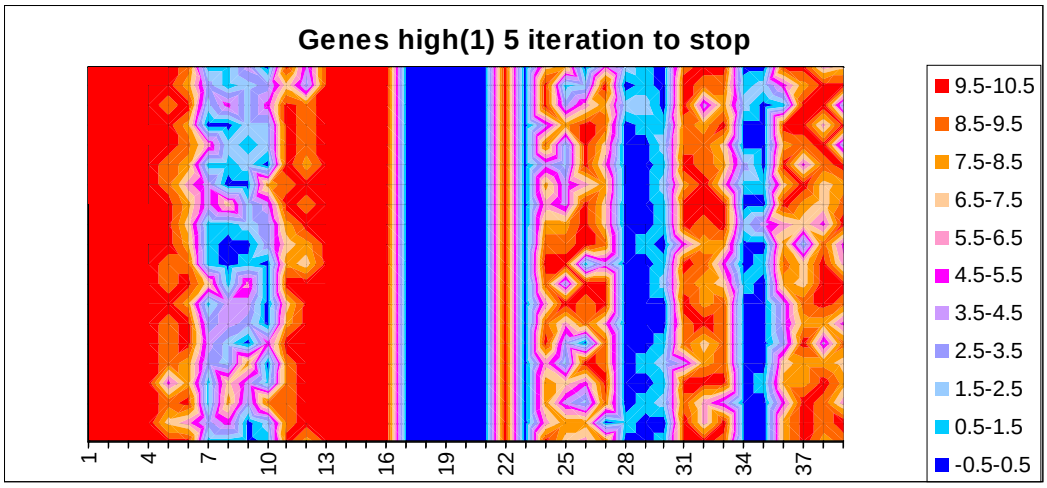

Figure 9.2: Genetic Algorithm high diet, repeatability 0.786.
Parent population size of 10 and stopping criteria of no change for 5 iterations.

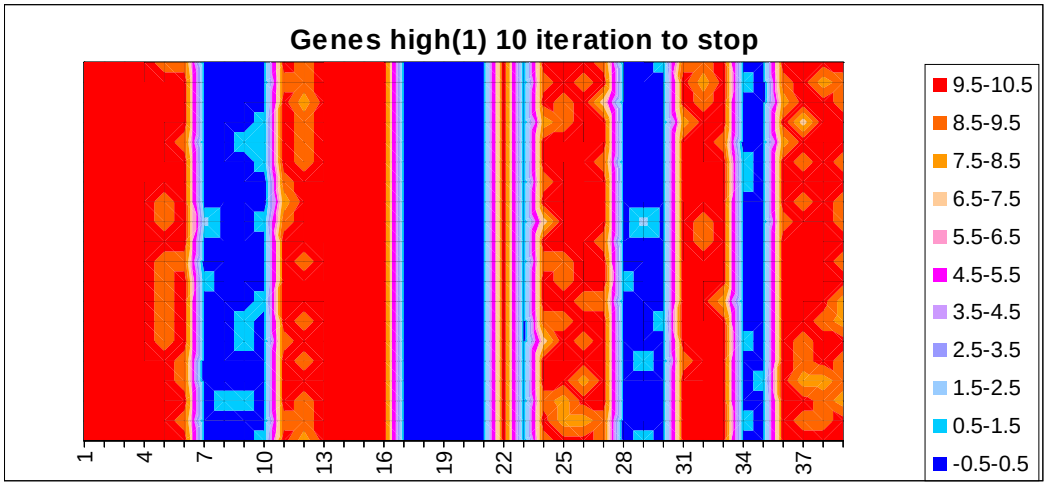

Figure 9.3: Genetic Algorithm high diet, repeatability 0.984.
Parent population size of 50 and stopping criteria of no change for 10 iterations.

9.3.1.2 Tabu Search

Figure 9.4 and Table 9.2 below shows the performance of the Tabu Search algorithm. It can be seen that Tabu Search has performed significantly better than the Monte Carlo

run and is producing results that are beyond 3 standard deviations from the mean. There appears to be little gain in solution quality in extending the number of genomes investigated. This can be seen by comparing the *N*=10, *E*=3, *S*=5 setup with the *N*=30, *E*=5, *S*=10 setup, which investigated 6689 and 44625 genomes on average respectively, and respectively had mean best BF of 3.475mm and 3.384mm, standard deviation of 0.0501mm and 0.0494mm, and repeatability of 0.731 and 0.771.

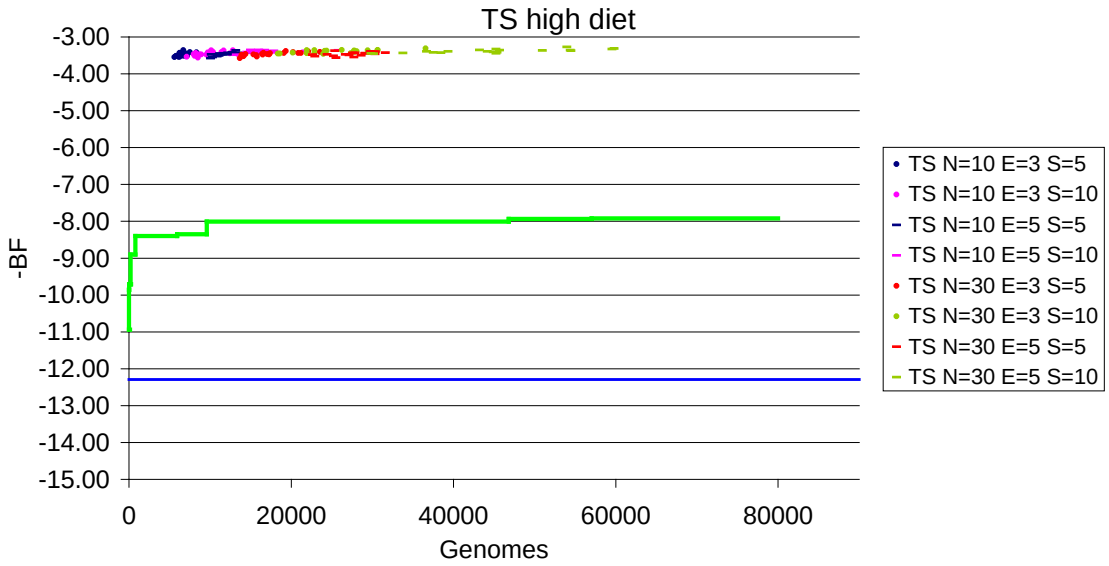

Figure 9.4: Tabu Search results when feeding the high diet.
Green line is Monte Carlo simulation results, Blue line is the mean.

Table 9.2: Tabu Search results when feeding the high diet.

			Genomes Evaluated		BF		
Configuration			**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
TS N=10	E=3	S=5	6689	745	3.475	0.0501	0.731
		S=10	9641	1569	3.447	0.0551	0.759
	E=5	S=5	11769	1208	3.466	0.0467	0.745
		S=10	16538	2835	3.408	0.0403	0.774
TS N=30	E=3	S=5	17517	2896	3.459	0.0477	0.747
		S=10	25566	4525	3.392	0.0432	0.781
	E=5	S=5	27587	2579	3.460	0.0509	0.739
		S=10	44625	8895	3.384	0.0494	0.771

The graphs below in Figure 9.5, the worst performing setup (10 neighbours, 3 elite solutions and a stopping criteria of no change for 5 iterations), and Figure 9.6, the best performing setup (30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations), show that the algorithm is finding only one optimal solution. The lack of improvement in solution quality between the two setups of the algorithm can clearly be seen by there being little reduction in variation of the alleles, which is supported by the small movement in repeatability from 0.731 to 0.771. The graphs for the other Tabu Search setups can be found in Appendix H-8.

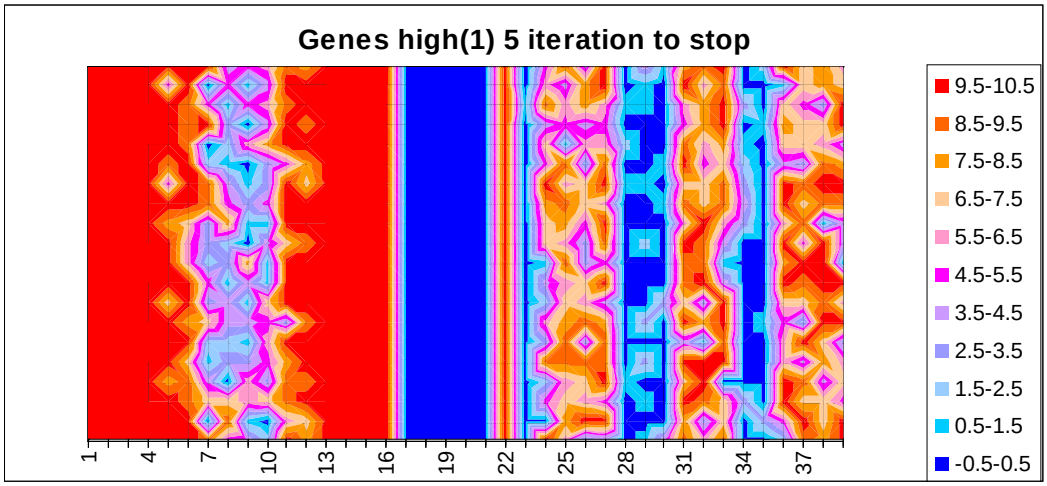

Figure 9.5: Tabu Search high diet, repeatability 0.731.
10 neighbours, 3 elite solutions and a stopping criteria of no change for 5 iterations.

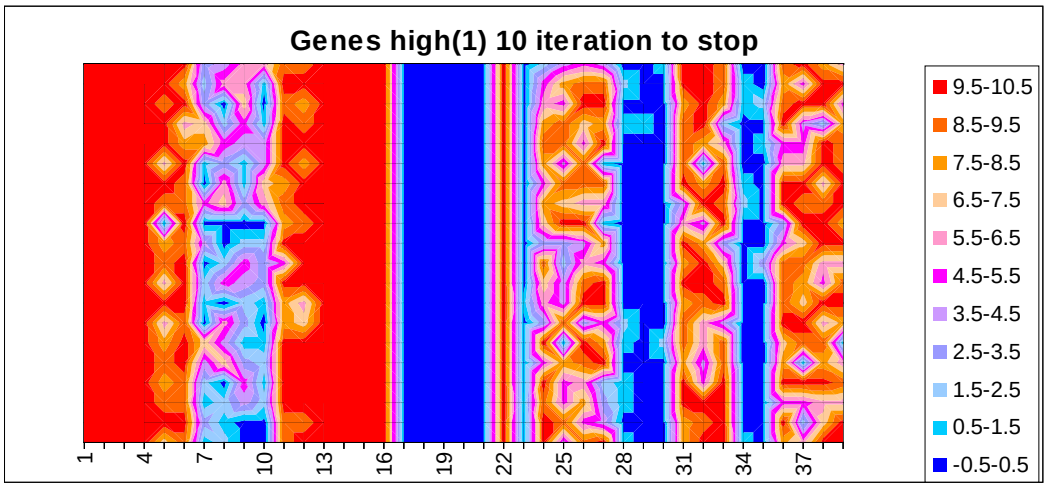

Figure 9.6: Tabu Search high diet, repeatability 0.771.
30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations.

9.3.1.3 Simulated Annealing

As shown in Figure 9.7 and Table 9.3 below, the Simulated Annealing algorithm produced fast results, but with large variation in quality of solution, with mean best BF ranging from 3.745 to 3.419, standard deviation ranging from 0.1055 to 0.0249 and repeatability ranging from 0.709 to 0.844, for the *N*=10, *T*=5, *S*=0.05 setup and *N*=50, *T*=30, *S*=0.01 setup respectively. However the solution quality does lie outside 3 standard deviations.

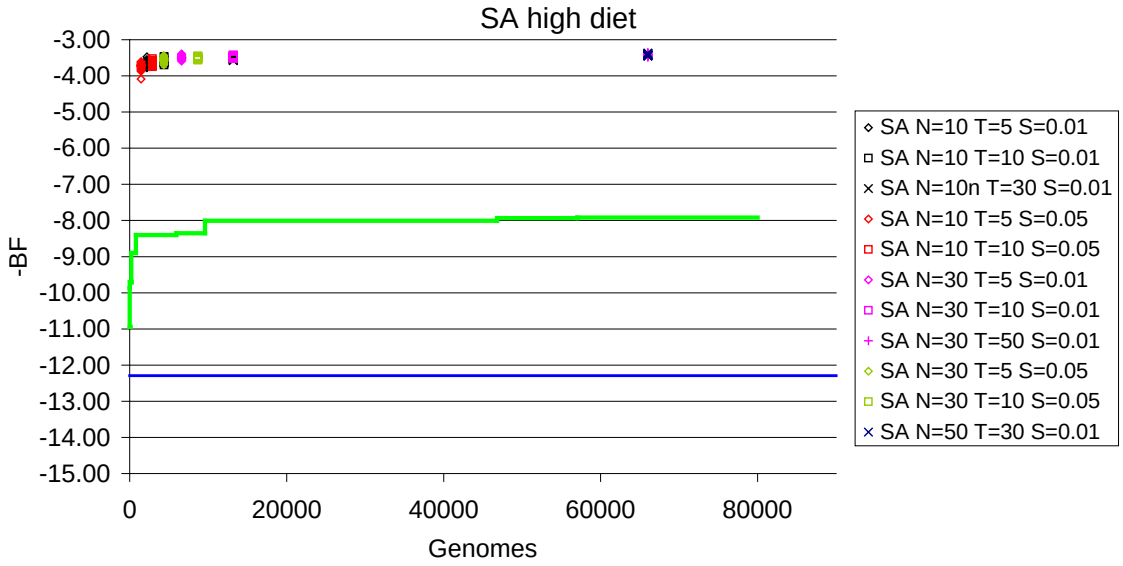

Figure 9.7: Simulated Annealing results when feeding the high diet. Green line is Monte Carlo simulation results, Blue line is the mean.

Table 9.3: Simulated Annealing results when feeding the high diet.

			Genomes Evaluated		BF		
Configuration			**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
SA N=10	T=5	S=0.01	2201	0	3.625	0.0743	0.735
	T=10		4401	0	3.593	0.0531	0.736
	T=30		13201	0	3.520	0.0328	0.789
	T=5	S=0.05	1451	0	3.745	0.1055	0.709
	T=10		2901	0	3.643	0.0581	0.757
SA N=30	T=5	S=0.01	6601	0	3.495	0.0484	0.809
	T=10		13201	0	3.477	0.0307	0.804
	T=50		66001	0	3.423	0.0308	0.829
	T=5	S=0.05	4351	0	3.551	0.0615	0.770
	T=10		8701	0	3.506	0.0334	0.791
SA N=50	T=30	S=0.01	66001	0	3.419	0.0249	0.844

Figure 9.8, which presents the worst performing setup (neighbourhood size of 10, cooling time of 5 iterations and stop temperature of 0.05), shows that this setup is not producing very clear or consistent results, which is supported by the low repeatability of 0.735, and standard deviation of 0.1055. However Figure 9.9, the best performance (neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01), with a repeatability of 0.844 is producing results that show a single optimal solution, even though there is still reasonable variation in best alleles being found. The graphs for the other Simulated Annealing setups can be found in Appendix H-9.

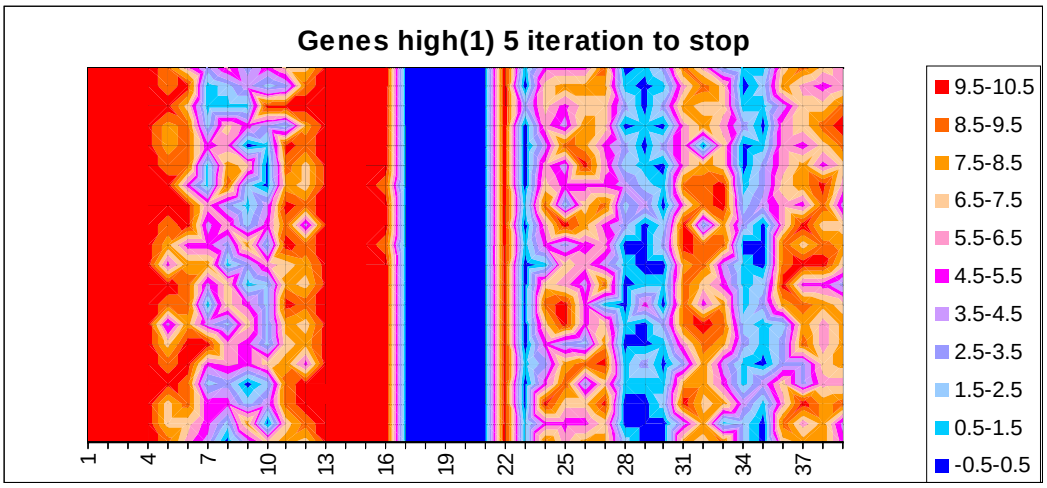

Figure 9.8: Simulated Annealing low diet, repeatability 0.735.
Neighbourhood size of 10, cooling time of 5 iterations and stop temperature of 0.05.

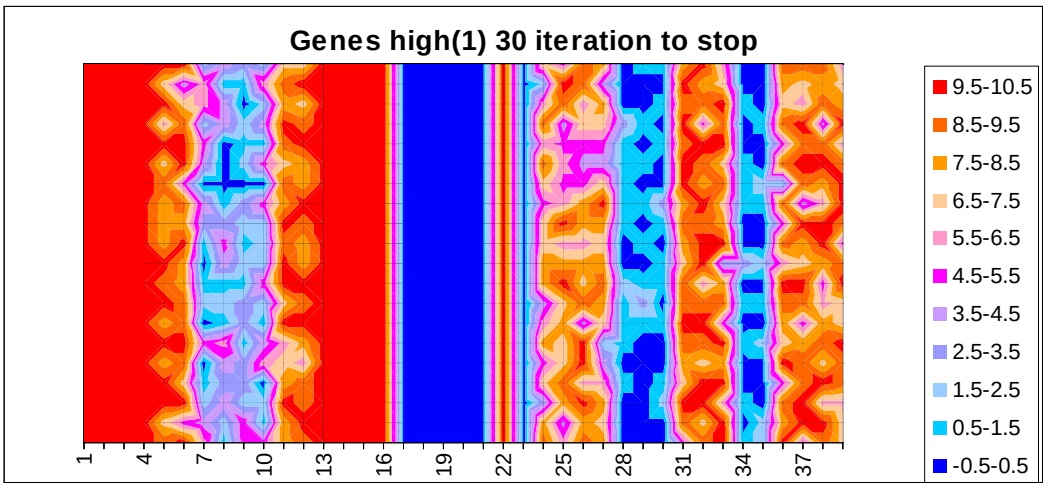

Figure 9.9: Simulated Annealing low diet, repeatability 0.844.
Neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01.

9.3.1.4 Algorithm comparison

The three algorithms are compared below in Figure 9.10. If the results are compared, for particular numbers of genomes investigated, the Genetic Algorithm gives results with higher repeatability than the other two algorithms, with repeatability ranging from 0.786 to 0.984 compared to the Tabu Search repeatability range of 0.731 to 0.771 and Simulated Annealing's range of 0.709 to 0.844 The Genetic Algorithm also produces greater average performance with mean best BF ranging from 3.449mm to 3.194mm compared to Tabu Search ranging from 3.475mm to 3.384mm and Simulated Annealing ranging from 3.745mm to 3.419mm. Figure 9.10 also shows that the Simulated Annealing algorithm and Tabu search have similar performance levels.

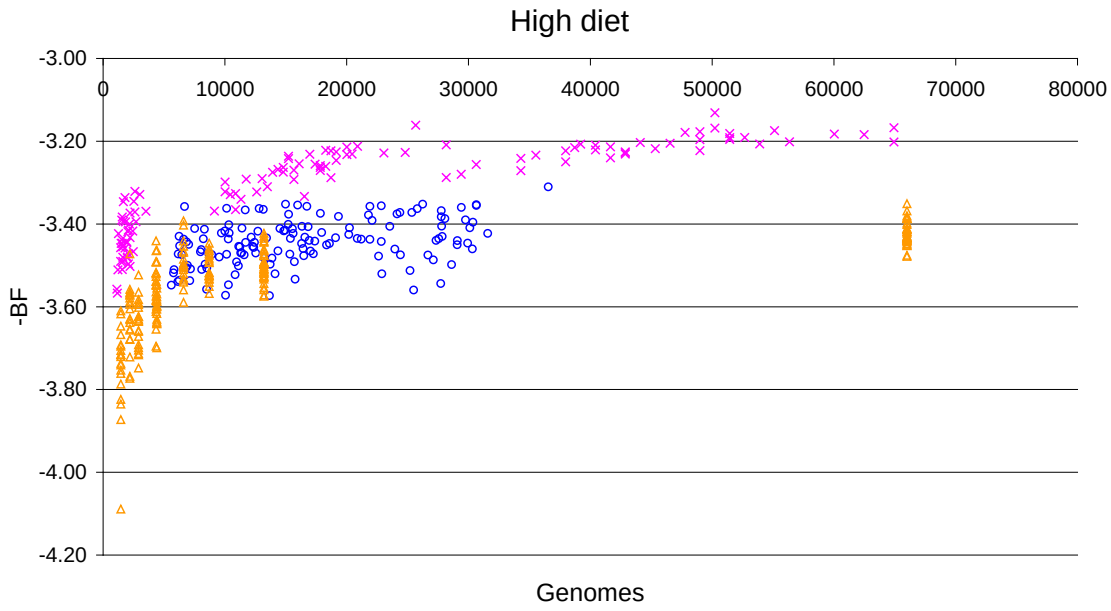

Figure 9.10: Comparison of the three algorithm results on high diet. Pink crosses – Genetic Algorithm, Blue circles – Tabu Search, Orange triangles – Simulated Annealing.

The graphs below in Figure 9.11, Figure 9.12, and Figure 9.13 show that the algorithms are producing the same general solution. These graphs clearly show that the alleles at loci 1 to 6, 11 to 16, 22, 24 to 27, 31 to 33 and 36 to 39 are maximized and the alleles at loci 7 to 10, 17 to 21, 23, 28 to 20 and 34 to 35 are minimized to achieve optimal performance. The results are clearest in the Genetic Algorithm display, with the results being the most consistent between runs. Tabu Search and Simulated Annealing had problems with loci 7 to 10 and loci 23 onwards, with these loci showing significant variation in the reported best alleles.

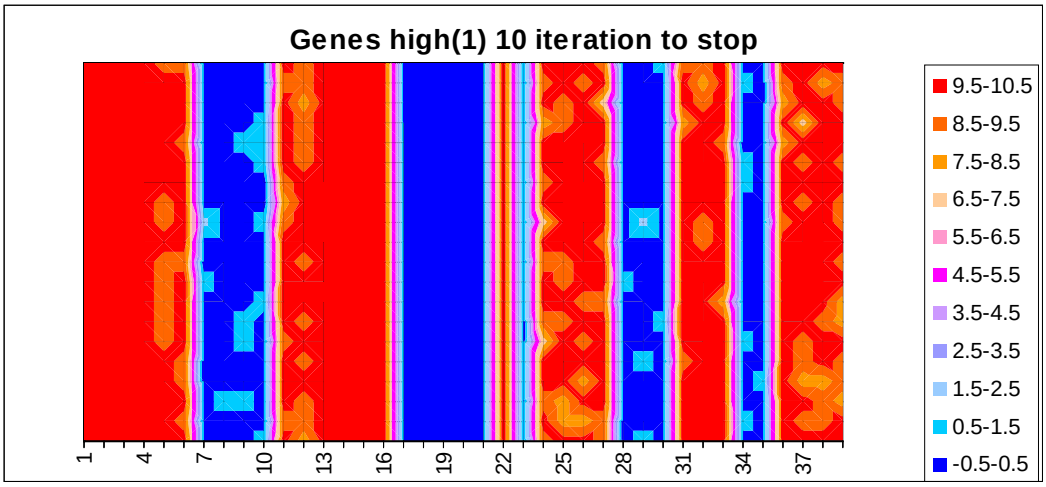

Figure 9.11: Genetic Algorithm high diet, repeatability 0.984.

Parent population size of 50 and stopping criteria of no change for 10 iterations.

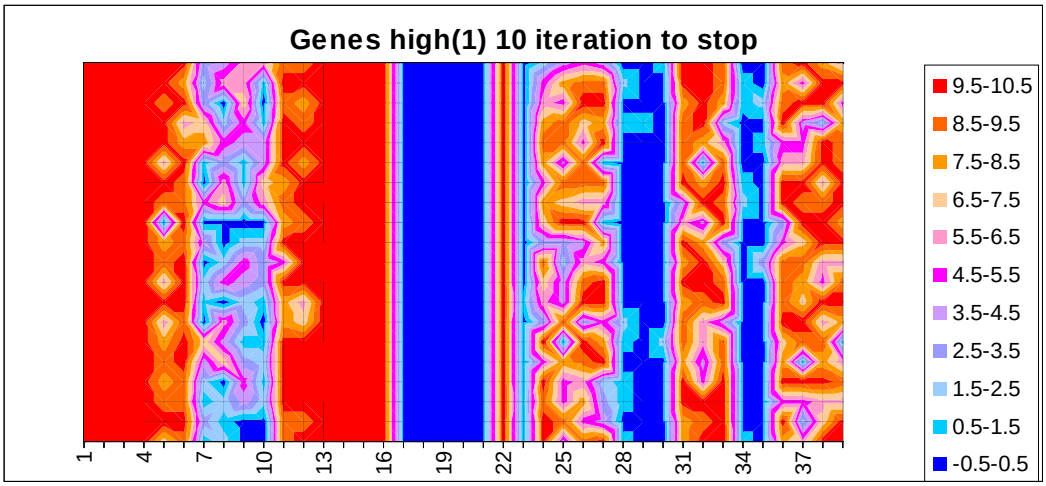

Figure 9.12: Tabu Search high diet, repeatability 0.771.

30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations.

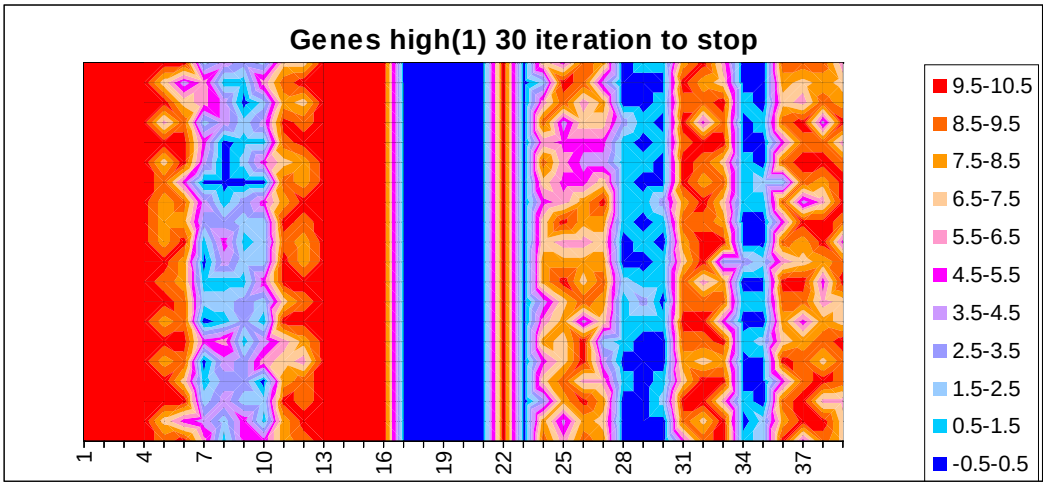

Figure 9.13: Simulated Annealing high diet, repeatability 0.844.

Neighbourhood size of 50, cooling time of 30 and stopping temperature of 0.01.

The factors for this optimal performance are shown in Table 9.4 below. This table clearly shows that E_m and w have been maximized and LP_{min} with values of 0.612 and 5.749 respectively and feed intake, *p*, have been minimized with a value of 0.634. Pd_{max} is in the upper portion of its ranges with 11 loci maximized and 4 minimized giving a value of 222.595. For the pig farmer this would mean that the pig had:

- Above average protein deposition (high muscle development)
- low lipid to protein ratio (little fat to protein ratio, i.e. a lean pig)
- high energy requirements for maintenance
- low ad libitum feed intake
- high water content.

Table 9.4: Optimal genome for back fat with high diet being fed.

BF optimal for High diet						
Loci	**Optimal**	Pd_{max}	LP_{min}	E_m	*p*	*w*
1-4	max	max	min			
5-6	max	max		max		
7-10	min	min			min	
11-12	max	max				max
13	max	max	min	max		
14-15	max	max	min			max
16	max		min	max		
17-21	min		min		min	
22	max		min	max		max
23	min		min		min	max
24-27	max			max		
28-30	min			max	min	
31-33	max			max		max
34-35	min				min	max
36-39	max					max
Mean		222.595	0.266	0.612	0.634	5.749

Table 9.5 below presents the performance of the optimal genome for back fat with the high diet being fed. The mean value of 3.4772mm for BF is nearly 3.5 standard deviations from the population mean of 12.29mm presented in Table 5.30. This is an

extreme value for BF and could be attributed to the maximum possible values for LP_{min}, E_m and w in the model being possibly to large. In particular values of 0.266 for LP_{min}, 0.612 for E_m, and 5.749 for w are beyond values observed in the literature listed in Table 5.6. This combined with possibly of the correlations between LP_{min}, E_m and w being too strong has resulted in this extreme observation. As more data on these parameters become available in the literature, the model could be further calibrated.

Table 9.5: Performance of optimal genome for BF with high diet being fed.

	BF optimal for High diet	
	Mean	**Std Dev**
FCR	2.874	1.3976
ADG	0.5887	0.2164
BF	3.4772	1.316
DTS	143.14	79.241
Mortality	475	

9.3.2 Low Diet

9.3.2.1 Genetic Algorithm

The Genetic Algorithm, as shown in Figure 9.14 and Table 9.6, produces consistent results throughout the range with repeatability ranging from 0.936 to 0.991, however there is more variation in the results for the quicker running setups than the longer running setups, with the standard deviation ranging from 0.1105mm to 0.0208mm. These results lie beyond 4 standard deviations of the mean.

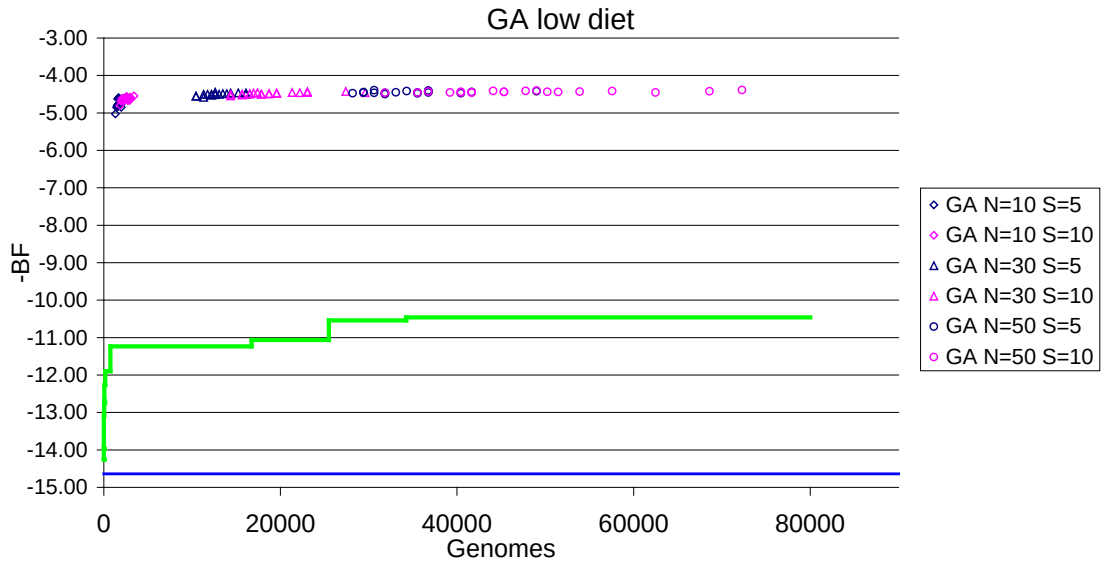

Figure 9.14: Genetic Algorithm results when feeding the low diet. Green line is Monte Carlo simulation results, Blue line is the mean.

Table 9.6: Genetic Algorithm results when feeding the low diet.

		Genomes Evaluated		BF (mm)		
Configuration		**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
GA N=10	S=5	1839	288	4.718	0.1105	0.936
	S=10	2508	428	4.629	0.0557	0.952
GA N=30	S=5	13203	1809	4.502	0.0329	0.983
	S=10	19315	4085	4.480	0.0289	0.984
GA N=50	S=5	34852	4956	4.447	0.0276	0.989
	S=10	47776	10790	4.435	0.0208	0.991

The graphs below in Figure 9.15, the worst performance (parent population size of 10 and stopping criteria of no change for 5 iterations), and Figure 9.16, the best performance (parent population size of 50 and stopping criteria of no change for 10 iterations), show that the algorithm is finding only one optimal solution. The improvement in solution quality between the two setups of the algorithm can clearly be seen with reduction in variation in the alleles, which is supported by the increase in repeatability from 0.936 to 0.991, and the reduction in standard deviation from 0.1105mm to 0.0208mm. The graphs for the other Genetic Algorithm setups can be found in Appendix H-7.

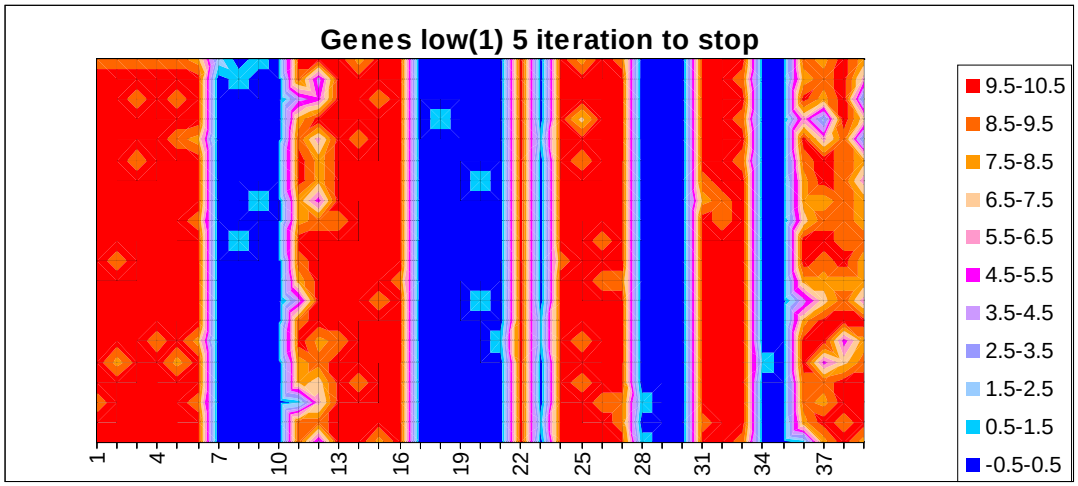

Figure 9.15: Genetic Algorithm low diet, repeatability 0.936.

Parent population size of 10 and stopping criteria of no change for 10 iterations.

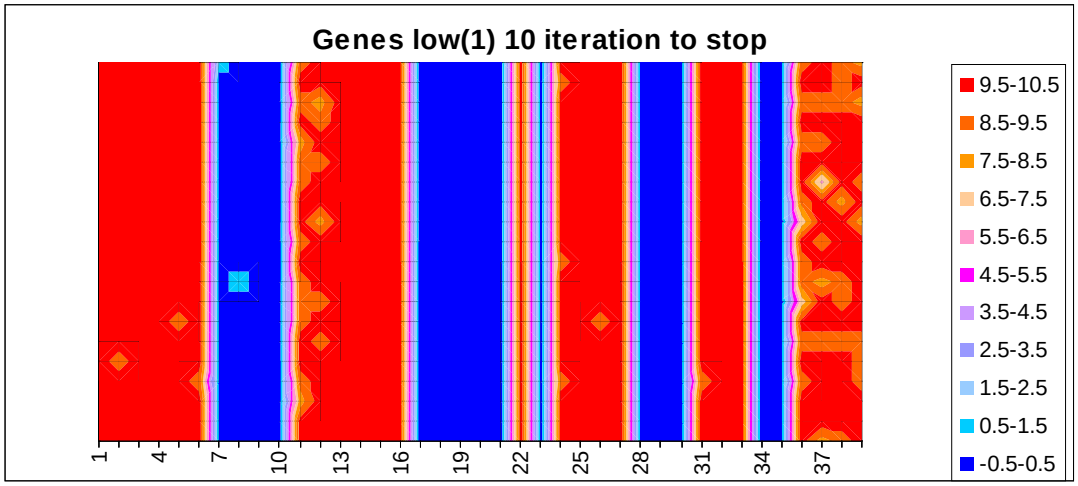

Figure 9.16: Genetic Algorithm low diet, repeatability 0.991.

Parent population size of 50 and stopping criteria of no change for 10 iterations.

9.3.2.2 Tabu Search

Figure 9.17 and Table 9.7 below shows that the Tabu Search results in a consistent and good quality solution production, with repeatability ranging from 0.937 to 0.954, mean best BF ranging from 4.554mm to 4.434mm and standard deviation ranging from 0.0650mm to 0.0308mm for the N=10, E=3, S=5 setup and N=30, E=5, S=10 setup respectively. These solutions lie beyond 4 standard deviations from the mean.

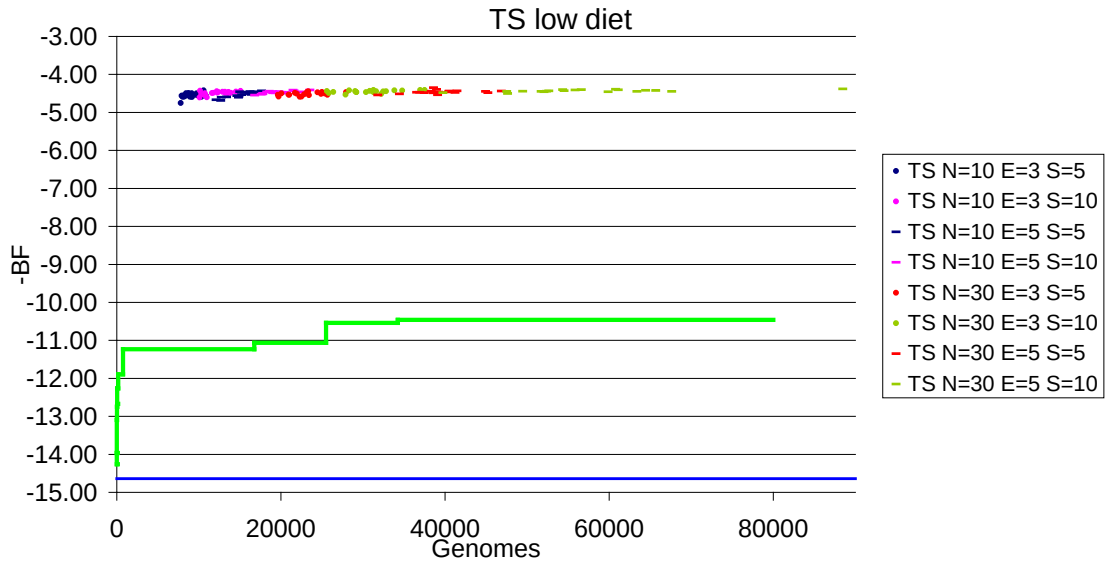

Figure 9.17: Tabu Search algorithm results when feeding the low diet. Green line is Monte Carlo simulation results, Blue line is the mean

Table 9.7: Tabu Search results when feeding the low diet.

			Genomes Evaluated		BF		
Configuration			**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
TS N=10	E=3	S=5	9017	920	4.554	0.0650	0.937
		S=10	12193	1552	4.492	0.0495	0.947
	E=5	S=5	15378	1629	4.527	0.0739	0.939
		S=10	19661	1708	4.473	0.0338	0.951
TS N=30	E=3	S=5	23076	2252	4.512	0.0440	0.932
		S=10	30871	3343	4.451	0.0336	0.967
	E=5	S=5	39431	3508	4.462	0.0433	0.953
		S=10	57287	10137	4.434	0.0308	0.954

The graphs below in Figure 8.18, the worst performance (10 neighbours, 3 elite solutions and a stopping criteria of no change for 5 iterations), and Figure 8.19, the best performance (30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations), clearly show that the algorithm is finding only one optimal solution. The lack of improvement in solution quality between the two setups of the algorithm can clearly be seen by there being little reduction in allele variation, which is supported by

the small movement in repeatability from 0.937 to 0.954. The graphs for the other Tabu Search setups can be found in Appendix H-8.

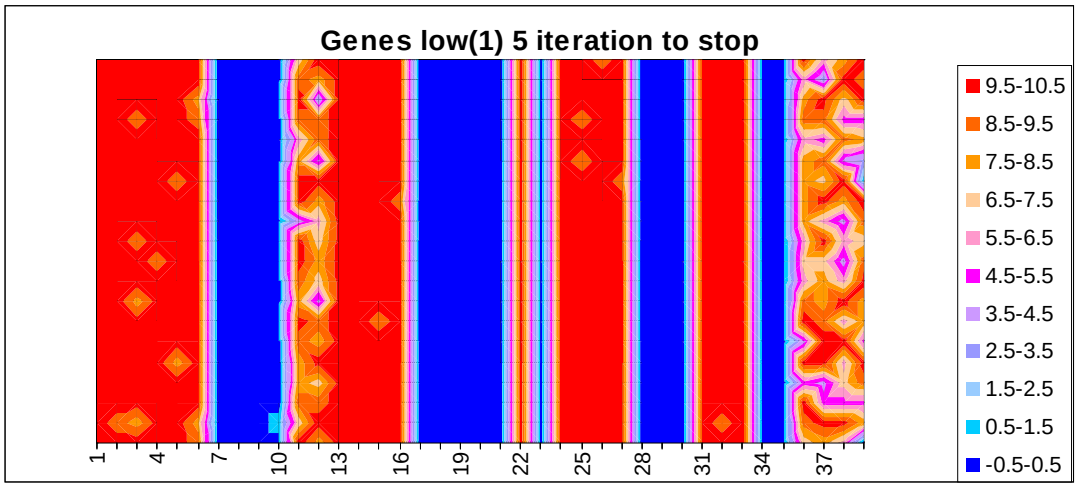

Figure 9.18: Tabu Search low diet, repeatability 0.937.
10 neighbours, 3 elite solutions and a stopping criteria of no change for 5 iterations.

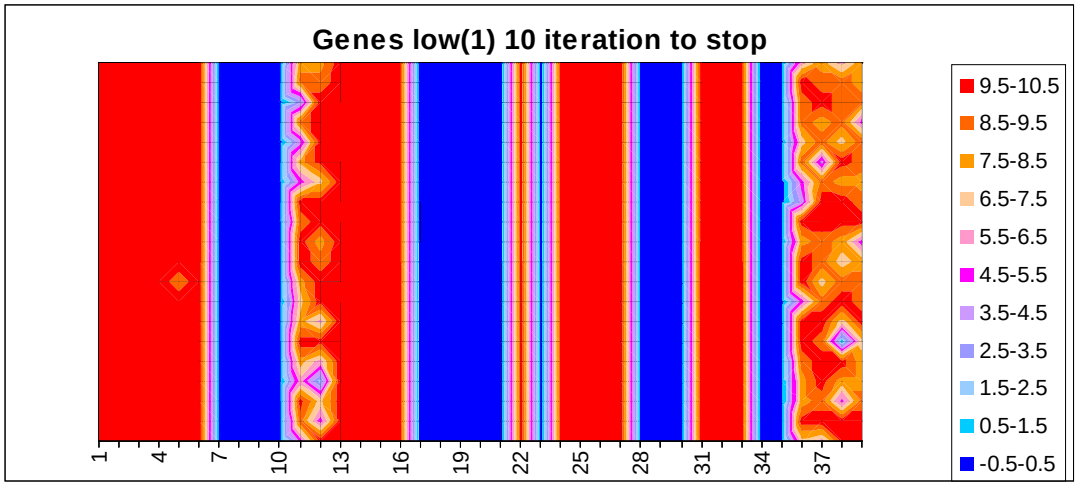

Figure 9.19: Tabu Search low diet, repeatability 0.954.
30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations.

9.3.2.3 Simulated Annealing

The Simulated Annealing results are shown below in Figure 9.20 and Table 9.8. Some variation is present with the mean best BF ranging from 5.570mm to 4.4584mm, standard deviation ranging from 0.1813mm to 0.0427mm and repeatability ranging from 0.848 to 0.953 for the N=10, T=5, S=0.05 setup and N=50, T=30, S=0.01 setup respectively. However the results lie beyond 3.5 standard deviations from the mean.

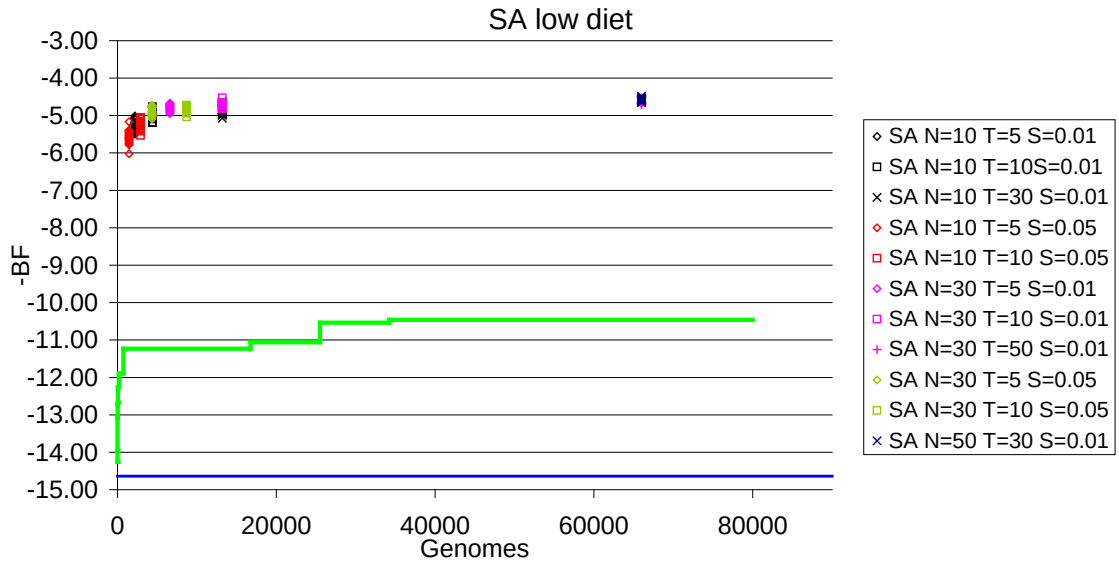

Figure 9.20: Simulated Annealing algorithm results when feeding the low diet. Green line is Monte Carlo simulation results, Blue line is the mean.

Table 9.8: Simulated Annealing results when feeding the low diet.

			Genomes Evaluated		**BF**		
Configuration			**Mean**	**Std Dev**	**Mean**	**Std Dev**	**Repeatability**
SA N=10	T=5	S=0.01	2201	0	5.262	0.1539	0.873
	T=10		4401	0	4.983	0.1045	0.919
	T=30		13201	0	4.840	0.1178	0.907
	T=5	S=0.05	1451	0	5.570	0.1813	0.848
	T=10		2901	0	5.256	0.1344	0.880
SA N=30	T=5	S=0.01	6601	0	4.800	0.0850	0.914
	T=10		13201	0	4.706	0.0667	0.942
	T=50		66001	0	4.623	0.0454	0.945
	T=5	S=0.05	4351	0	4.901	0.1115	0.911
	T=10		8701	0	4.828	0.0731	0.916
SA N=50	T=30	S=0.01	66001	0	4.584	0.0427	0.953

Figure 9.21, the worst performance (neighbourhood size of 10, cooling time of 5 iterations and stop temperature of 0.05), shows that this setup is not producing very clear or consistent results, which is supported by the low repeatability of 0.873 and standard deviation of 0.1813mm. However Figure 9.22, the best performance (neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01),

with a repeatability of 0.953 is producing results that show a single optimal solution, even though there is still reasonable variation in the reported best alleles. The graphs for the other Simulated Annealing setups can be found in Appendix H-9.

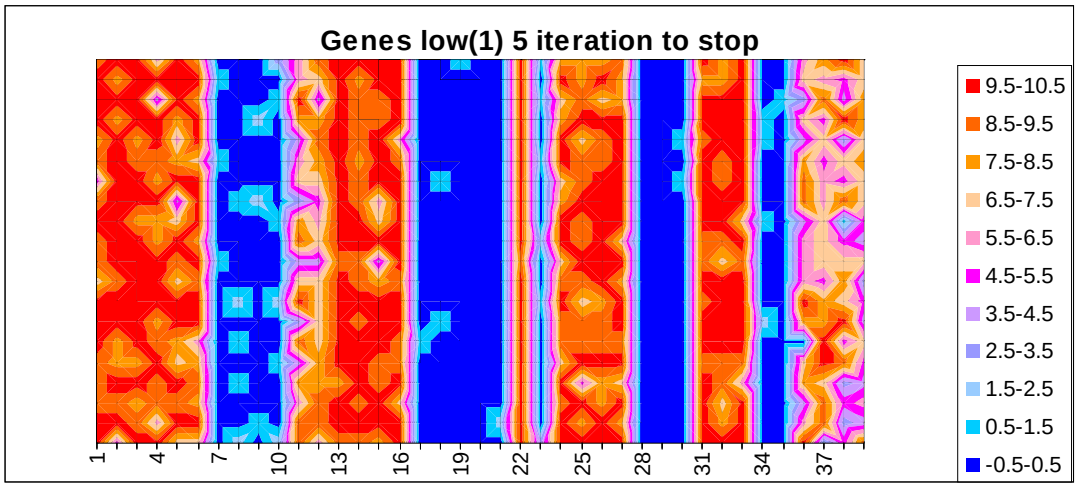

Figure 9.21: Simulated Annealing low diet, repeatability 0.873.
Neighbourhood size of 10, cooling time of 5 iterations and stop temperature of 0.05.

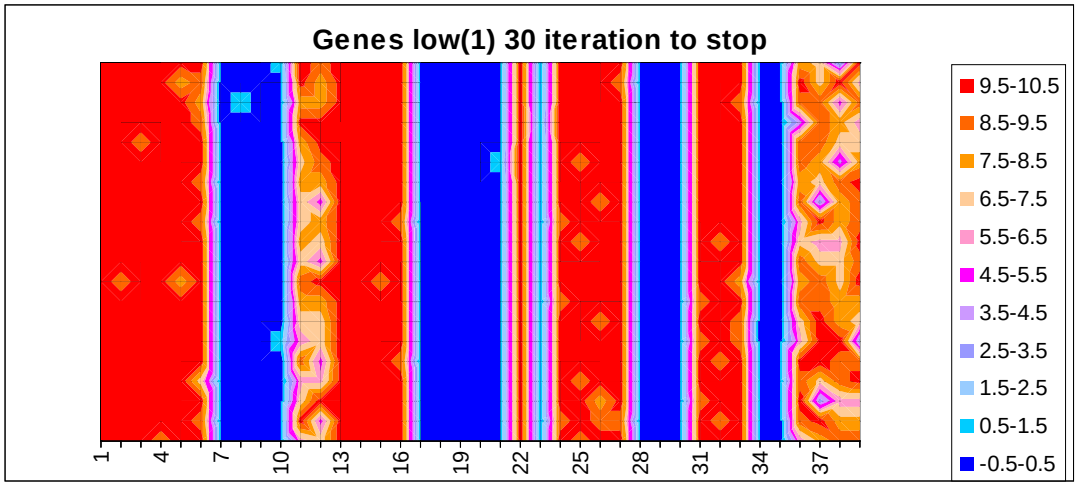

Figure 9.22: Simulated Annealing low diet, repeatability 0.953.
Neighbourhood size of 50, cooling time of 30 iterations and stop temperature of 0.01.

9.3.2.4 Algorithm comparison

The results from all three algorithms are plotted below in Figure 9.23. It is clear that the Genetic Algorithm and Tabu Search algorithm produced very similar results. This can be seen when mean best BF of 4.502mm, 4.492mm, 4.435mm, and 4.434mm are compared for the Genetic Algorithm setup N=30, S=5, Tabu Search setup N=10, E=3, S=10, Genetic Algorithm setup N=50, S=10, and Tabu Search setup N=30, E=5, S=10 respectively which evaluated on average 13203, 12193, 47776, and 57287 genomes.

However the Tabu Search solutions have more variation, which can be seen when the standard deviations of 0.0329mm, 0.0495mm, 0.0208mm, and 0.0308mm respectively are compared. Both these algorithms outperform the Simulated Annealing algorithm, which had mean best BF of 4.840mm and 4.584mm, and standard deviation of 0.1178mm and 0.0427mm for setup N=10, T=30, S=0.01, and setup N=50, T=30, S=0.01 respectively, which evaluated 13201 and 66001 genomes respectively.

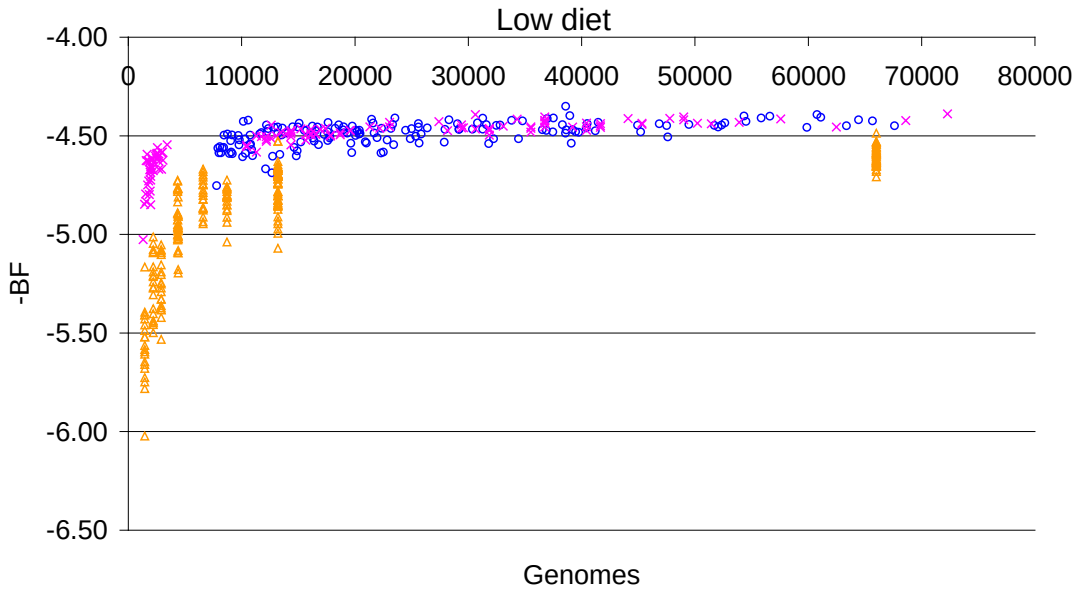

Figure 9.23: Comparison of the three algorithm results on low diet.
Pink crosses – Genetic Algorithm, Blue circles – Tabu Search, Orange triangles – Simulated Annealing.

The graphs in Figure 9.24, Figure 9.25, and Figure 9.26 show that, again, the algorithms are converging to a common solution. For optimal performance, the alleles at loci 1 to 6, 11 to 16, 22, 24 to 27, 31 to 33 and 36 to 39 are maximized, and loci 7 to 10, 17 to 21, 23, 28 to 20 and 34 to 35 are minimized. The Genetic Algorithm has produced the most consistent results, and hence the clearest graphical representation with a repeatability of 0.991. The Genetic Algorithm has some variation in the alleles at loci 11 to 12 and 36 to 39 with the Tabu Search and Simulated Annealing having more, which is reflected by the lower repeatabilities of 0.954 and 0.953 respectively.

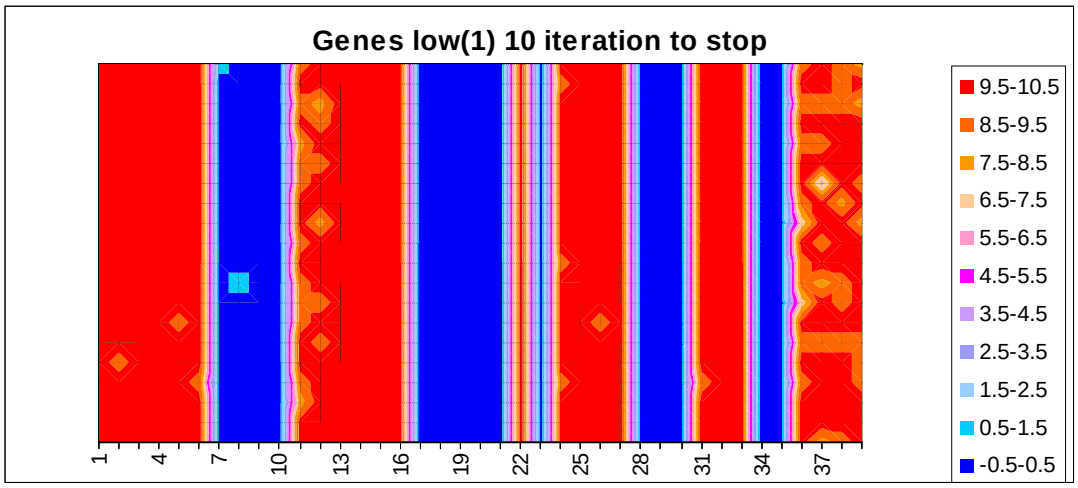

Figure 9.24: Genetic Algorithm low diet, repeatability 0.991.

Parent population size of 50 and stopping criteria of no change for 10 iterations.

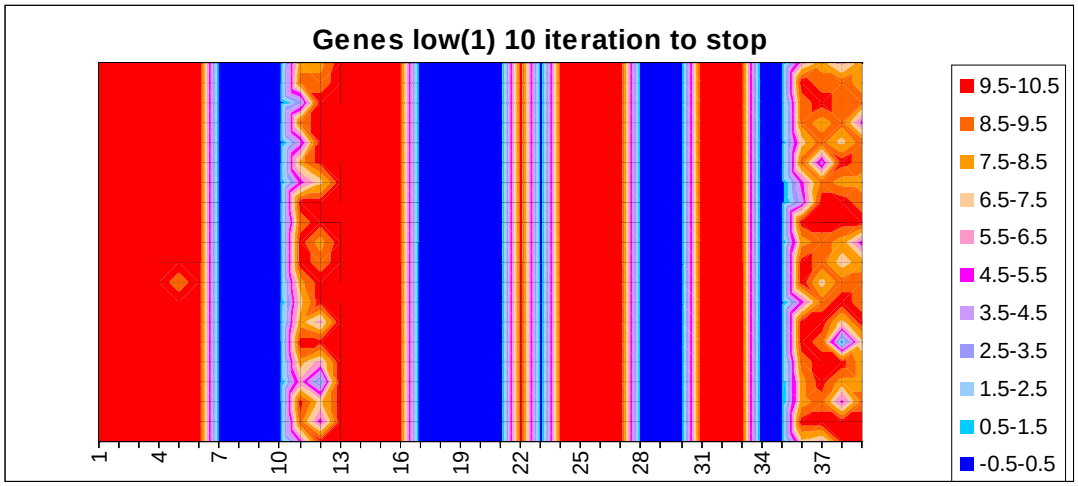

Figure 9.25: Tabu Search low diet, repeatability 0.954.

30 neighbours, 5 elite solutions and a stopping criteria of no change for 10 iterations.

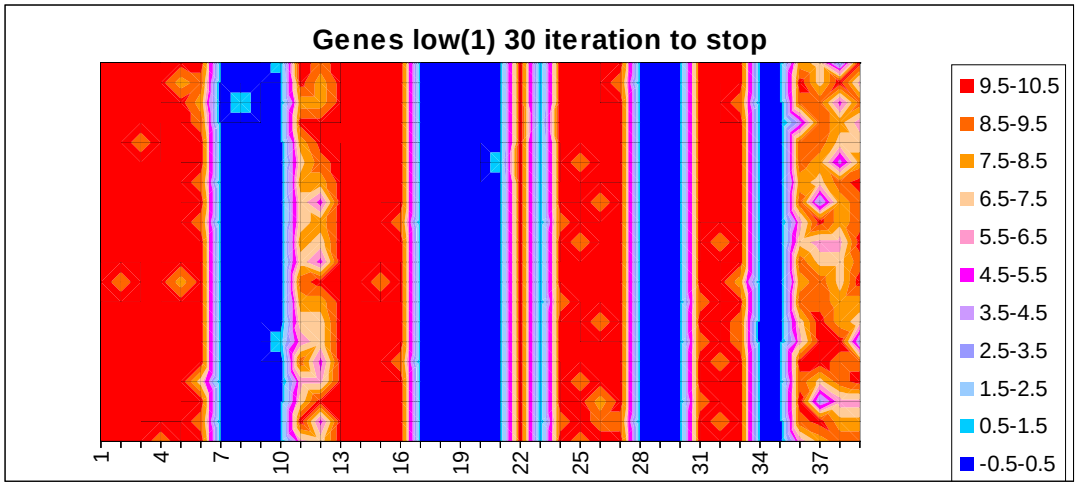

Figure 9.26: Simulated Annealing low diet, repeatability 0.953.

Neighbourhood size of 50, cooling time of 30 and stopping temperature of 0.01.

The factors for this optimal performance are shown in Table 9.9 below. This table clearly shows that E_m and w have been maximized with respective value of 0.612 and 5.749, and LP_{min} and feed intake, *p*, have been minimized with values of 0.266 and 0.634 respectively. Pd_{max} is the upper portion of its ranges with 11 loci maximized and 4 minimized, with a value of 222.595. For the pig farmer this would mean that the pig had:

- Above average protein deposition (high muscle development)
- low lipid to protein ratio (little fat to protein ratio, i.e. a lean pig)
- high energy requirements for maintenance
- low ad libitum feed intake
- high water content.

Table 9.9: Optimal genome for back fat with low diet being fed.

BF optimal on Low diet						
Loci	**Optimal**	Pd_{max}	LP_{min}	E_m	*p*	*w*
1-4	max	max	min			
5-6	max	max		max		
7-10	min	min			min	
11-12	max	max				max
13	max	max	min	max		
14-15	max	max	min			max
16	max		min	max		
17-21	min		min		min	
22	max		min	max		max
23	min		min		min	max
24-27	max			max		
28-30	min			max	min	
31-33	max			max		max
34-35	min				min	max
36-39	max					max
Mean		222.595	0.266	0.612	0.634	5.749

Table 9.10 below presents the performance of the optimal genome for back fat with the low diet being fed. The mean value of 4.6178mm for BF is over 4.5 standard deviations

from the population mean of 14.64mm presented in Table 5.30. This is an extreme value for BF and could be attributed to the maximum possible values for LP_{min}, E_m and w in the model being possibly to large. In particular values of 0.266 for LP_{min}, 0.612 for E_m, and 5.749 for w are beyond values observed in the literature listed in Table 5.6. This combined with possibly of the correlations between LP_{min}, E_m and w being too strong has resulted in this extreme observation. As more data on these parameters become available in the literature, the model could be further calibrated.

Table 9.10: Performance of optimal genome for BF with low diet being fed.

	BF optimal for Low diet	
	Mean	**Std Dev**
FCR	2.6447	1.002
ADG	0.6509	0.157
BF	4.6178	1.6089
DTS	119.01	54.899
Mortality	173	

9.3.3 Diet comparison

Figure 9.27 shows that the comparative performance between the three algorithms is affected by the type of diet that is being fed. Tabu Search has a performance similar to the Genetic Algorithm on the low diet, however with the high diet the performance similar to the Simulated Annealing, with the Genetic Algorithm generally giving better solutions.

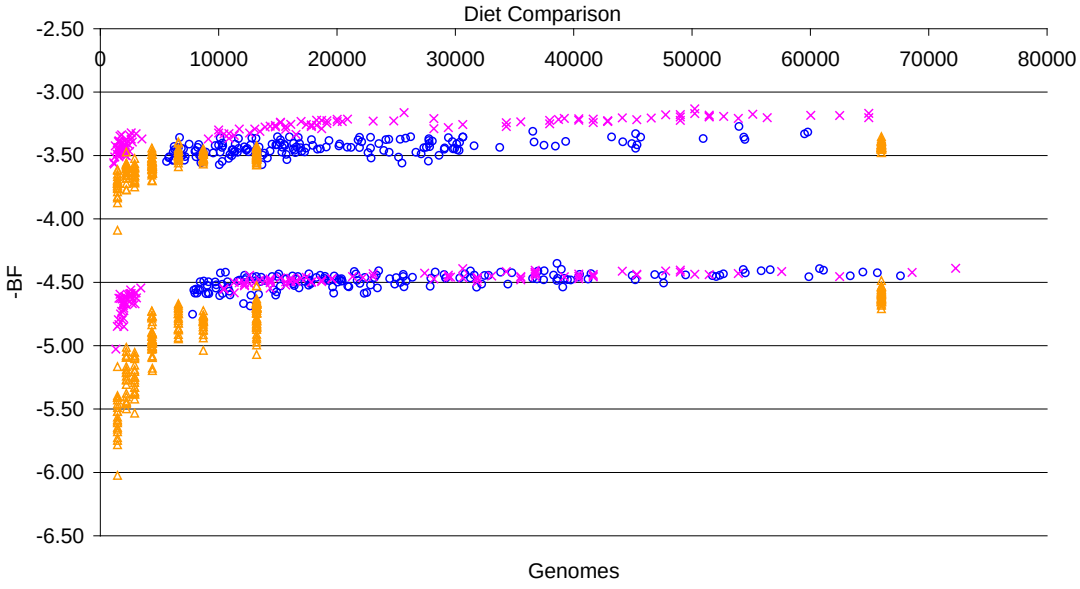

Figure 9.27: Comparison of the three algorithm results on low diet and high diet. Pink crosses – Genetic Algorithm, Blue circles – Tabu Search, Orange triangles – Simulated Annealing.

The figure below provides a visual comparison of the optimal results for the high and low diet when optimizing back fat. The results shown are from the Genetic Algorithm, which had the lease noise of the three algorithms.

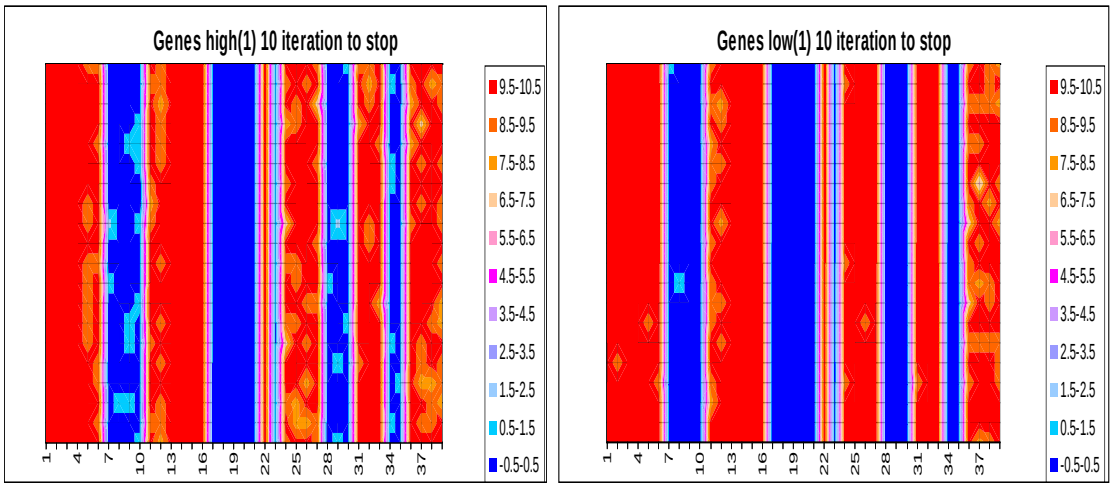

Figure 9.28: Genetic Algorithm diet compression.
Parent population size of 50 and stopping criteria of no change for 10. High diet results are on the left and low diet results on the right.

The optimal solution for the two diets is the same. This indicates that when optimizing back fat the diet has no effect on the optimal solution. However, there is an accompanying rise in mortality rate, as the pig growth model fails to grow up to five percent of the pigs with this additive genotype. Effectively this genotype has a tendency to be unable to maintain its body weight. The pig eats so little food that there is insufficient energy eaten to meet energy requirements for maintenance.

This increase in observed mortality in the model provides the important information that extreme selection on BF only will likely, in the long term, lead to poor animal fitness and potentially increased mortality rates. There are numerous examples of undesirable effects of selection for high productivity (Rauw, 2007). It is possible to adjust the objective function so that increases in mortality are penalised, so that they are no longer desirable. However this would hide which selection objectives are potentially leading to increased mortality rates. This increased risk of mortality would likely decrease genetic gains observed as the offspring generation would be missing some of top performing animals due to the increased mortality, and thereby reducing the mean performance of the offspring population compared to what would have been the mean performance if there was no increased mortality.

9.4 Theoretical generations to optimal solutions

As in §7.4.4, this section is concerned with the prediction of the minimum number of theoretical generations of breeding needed to reach the optimal solution. The minimum number of generations is calculated as the difference between the mean of the population and the mean of the optimal. The difference is then divided by the genetic gain. This calculation does not take into account the Bulmer effect. The Bulmer effect, if included, would result in the genetic gain decreasing with each generation of breeding, and hence increasing the number of generations to the optimal solution.

The heritability figures of back fat, generated by the model, for each diet has been estimated by the simulation of 50,000 breeding pairs over one generation of breeding.

As shown in Figure 9.29, back fat for the high diet has a mean of 12.3mm, a standard deviation of 2.56mm and an optimal genome average of 3.48mm. Back fat on the high diet achieves a theoretical gain per generation of 0.85mm, with a heritability of 0.332

and a breeding program with an i-bar (selection intensity) of 1. This indicates that a minimum of 10.4 generations of breeding would be required, if there was no Bulmer effect, for the optimal back fat solution to be reached.

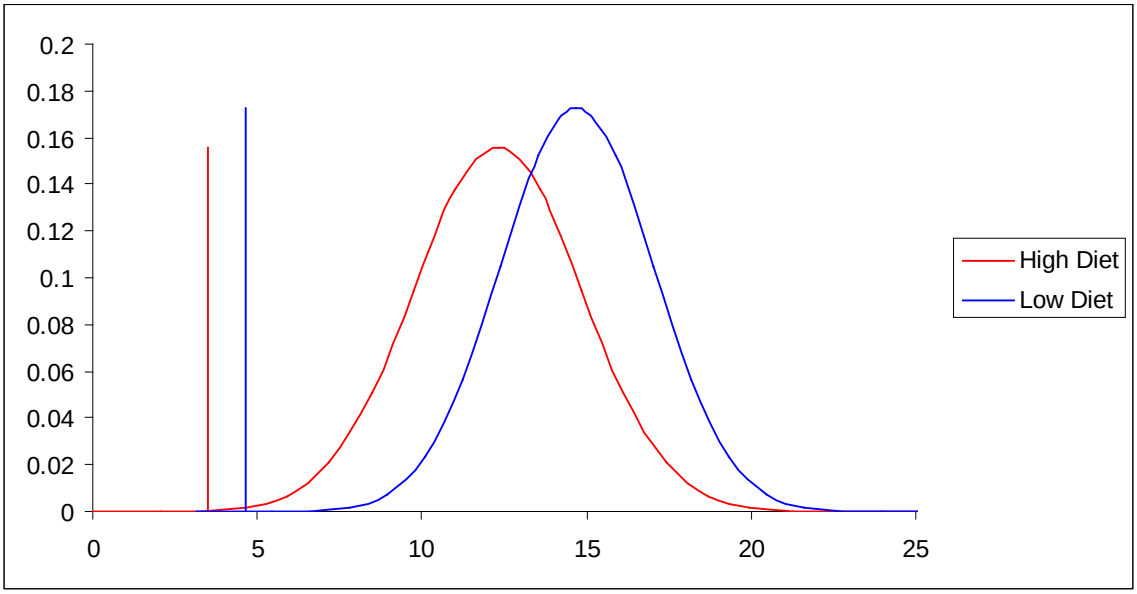

Figure 9.29: Optimal genome average plotted with population normal curve.

Figure 9.29 also show the low diet for back fat, which has a mean of 14.7mm, a standard deviation of 2.31mm and an optimal solution average of 4.62mm. The heritability is now 0.321. Therefore, given a breeding program with an i-bar of 1, back fat on the low diet has a theoretical gain per generation of 0.741. This indicates that for the low diet it would take a minimum of 13.6 generation to reach the optimal solution, again if the Bulmer effect is ignored.

This information for both diets is summarised in Table 9.11 below.

Table 9.11: Summary data

Diet	Mean	Std Dev	Heritability	Genetic gain	Optimal	Generations
High	12.3	2.56	0.332	0.850	3.48	10.4
Low	14.7	2.31	0.321	0.741	4.62	13.6

As can be seen, the diet has an effect on the theoretical minimal number of generations to reach the optimal solution for back fat. In this case, it takes longer to reach the optimal solution on the low diet than the high diet.

9.5 Summary

It has been found that there is clear evidence that the algorithms produce an optimal genotype when the overall objective is to minimize back fat. This optimal genotype is unaffected by the diet. The optimal genotype produced by this objective has a raised risk of mortality, with the GE-Pig model unable to successfully grow some of the pigs.

The Genetic Algorithm produced the best overall performance of the three algorithms, as it produces solutions with less variation. The solutions produced also have the same or better quality as those from the other two algorithms. The Genetic Algorithm also has the ability to be set to produce quick, good quality solutions. Tabu Search produces the same high quality solutions as the Genetic Algorithm on the low diet, but with more variability. However when on the high diet, the quality of solutions from Tabu Search is significantly worse than the Genetic Algorithm solutions. Also, Tabu Search is unable to produce quick solutions. The Simulated Annealing solutions were of a lower quality when compared to the Genetic Algorithm in terms of variation and genetics.

If minimizing back fat is the objective, the theoretical minimal number of generations to the optimal solution is higher on the low diet than the high diet.

Chapter 10 – Multiple objective

10.1 Introduction

In this chapter a multiple objective is created using a weighted sum of the three individual objectives, and the weightings for this multiple objective are discussed. Interpreting the results layout is described, followed by the results for the high diet and the low diet. This is then followed by a general discussion of the single and multiple objective solutions.

10.2 Multiple selection objective

Up to this point the three factors, average daily gain, feed conversion ratio, and back fat, have been investigated in isolation. It is rare though, in breeding programs, for these factors to be considered in isolation. More commonly, the interaction of at least two of the factors is explored. For example, high average daily gain is investigated whilst at the same time back fat is maintained at a low level.

This section combines the three factors into a single multiple selection objective and investigates the effect of varying the weightings of the factors on the solution.

10.2.1 Objective Selection

The multiple selection objective is compiled from several relative economic values that have previously been presented in the literature, as summarised in Table 10.1 below. Economic values of traits are intended to represent the expected change in producer profit for a unit change in the trait. There are ranges of estimates of economic values for traits, and they very across regions, markets, and production systems. They also vary from farm to farm. Relative economic values are calculated by setting one trait by which all comparisons will be made.

Table 10.1: Relative Economic Values.

Source	ADG	FCR	BF
Voegeli 1978	40	-60	-4.50
Voegeli 1978	130	-83	-6.5
Allan B. B *et al*. 1985	5 to 75		-1
Morris *et al*.1978	486.6	-816	-40.8
Skorupski *et al.* 1995	1100 to 5000		-51 to -118
Morel P. *et al.* 1988	100	-29.31	-5.69

The various relative economic values in Table 10.1 above, have varying basis from which they are compared to, therefore they have been converted to a basis of back fat in Table 10.2 below.

Table 10.2: Relative Economic Values converted to ratios.

Source	ADG	FCR	BF
Voegeli 1978	9	-13	-1
Voegeli 1978	20	-13	-1
Allan B. B *et al*. 1985	5 to 75		-1
Morris *et al*.1978	12	-20	-1
Mariuz Skorupski 1995	9 to 98		-1
Morel P. *et al.* 1988	18	-5	-1

The ratios for use with the multiple selection objective analysis are selected to be representative of the relative economic values in Table 10.2. The resulting multiple selection objective ratios are shown in Table 10.3.

Table 10.3: Multiple selection objective ratios.

	ADG	FCR	BF
Low	5	-5	
Med	20	-12	-1
High	50	-20	

The multiple objective is then

$$\text{Multiple Objective} = \alpha ADG + \beta FCR + \varepsilon BF$$

where α is either the low, med or high value for ADG in Table 10.3, β is the low, med, or high value for FCR in Table 10.3, and ε is -1.

The simulations were performed using all the possible combinations of the ratios between the three traits, ADG, FCR, and BF. This results in a total of nine multiple objectives being investigated over the two diets.

For the simulations, the starting weight is set to 20kg and the pigs are grown to 90kg with 1000 pigs grown to evaluate the objective.

10.2.2 Interpretation of graphs

The graphs used in the analysis are the solution graphs of the results, as described in §7.3, laid out in a three by three matrix to allow for easy comparison. Increasing average daily gain weighting is laid out left to right and increasing feed conversion ratio is bottom to top. The overall graphic display is as shown below in Figure 10.1.

The main effects of average daily gain and feed conversion ratio lie horizontally and vertically respectively. The back fat effect, after the main effects of average daily gain and feed conversion ratio have been removed, lies along the diagonal.

In addition, for extra comparison of trends, the three single objective optimal solutions are displayed around the layout in the increasing direction of their main effect.

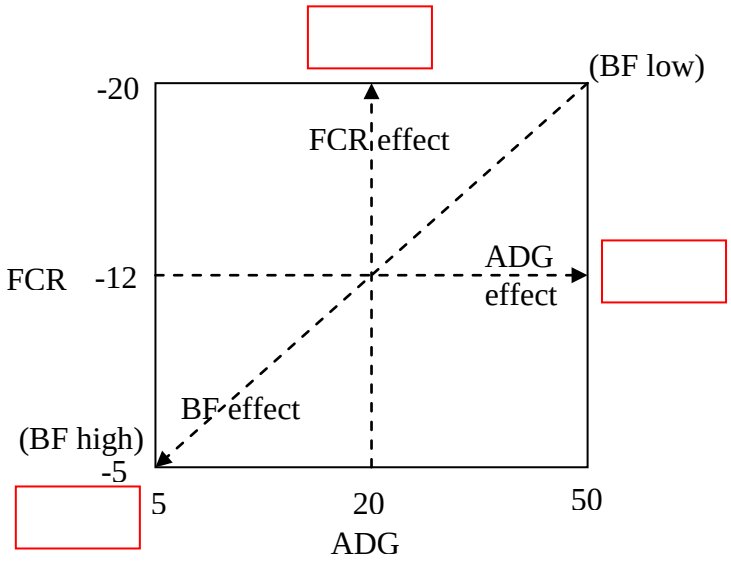

Figure 10.1: Layout of multiple objective graphs with single objective optimal solutions.

10.2.3 Results

10.2.3.1 High diet

In the results in Figure 10.2, a horizontal visual comparison of the solution graphs, from left to right, shows that there is evidence of an effect as the weighting of average daily gain increases. This is most visible when the weighting of feed conversion ratio is low (bottom row). This can also be seen in the change in values in the corresponding table, Table 10.4.

In a vertical comparison, there is little or no main feed conversion ratio effect. However when average daily gain is high (right column), there is movement away from the feed conversion ratio solution as weighting for FCR decreases.

The diagonal comparison provides little indication that back fat has an effect on the optimal solution when the high diet is being fed.

It can be seen that as the weight of average daily gain increases, the solution is trending towards the single objective average daily gain optimal solution. However the weighting of average daily gain would need to be very high before the single objective optimal solution is achieved.

When the weighting of average daily gain and feed conversion ratio decrease, and hence the weighting of back fat increases, there is little indication that the solution is trending towards the back fat optimal solution. This indicates that the weighting of back fat in the objective would need to be at an extreme level before the back fat single objective optimal solution is achieved.

There is evidence of the similarity of the feed conversion ratio optimal solution to the solutions for the nine multiple objectives. In general, then, it can be seen that the feed conversion ratio is the main driving factor in the multiple selection objective.

Table 10.4: Mean performances for multiple selection objective solutions, high diet.

Population			
Pd_{max}	FCR	170	2.175
LP_{min}	ADG	0.7	0.924
E_m	BF	0.485	12.29
p	DTS	0.9	77.59
w		5.05	

FCR Optimal			
Pd_{max}	FCR	281.94	1.35
LP_{min}	ADG	0.266	1.480
E_m	BF	0.481	4.95
p	DTS	0.889	48.27
w		5.75	

			ADG: Low		ADG: Med		ADG: High	
FCR	High	Pd_{max}	281.47	1.37	281.67	1.36	281.83	1.42
		LP_{min}	0.269	1.421	0.275	1.468	0.360	1.513
		E_m	0.495	4.61	0.484	4.99	0.483	6.27
		p	0.867	50.41	0.889	48.69	0.954	47.15
		w	5.71		5.73		5.70	
	Med	Pd_{max}	281.05	1.38	281.81	1.37	281.47	1.46
		LP_{min}	0.269	1.404	0.276	1.474	0.421	1.536
		E_m	0.499	4.49	0.486	5.06	0.484	7.12
		p	0.859	51.07	0.895	48.47	0.997	46.43
		w	5.72		5.72		5.68	
	Low	Pd_{max}	281.12	1.39	281.66	1.38	281.86	1.49
		LP_{min}	0.267	1.385	0.283	1.477	0.474	1.561
		E_m	0.511	4.34	0.487	5.21	0.483	7.90
		p	0.855	51.83	0.904	48.35	1.038	45.67
		w	5.71		5.71		5.66	

ADG Optimal			
Pd_{max}	FCR	281.94	1.59
LP_{min}	ADG	0.576	1.620
E_m	BF	0.481	9.79
p	DTS	1.149	44.07
w		5.56	

BF Optimal			
Pd_{max}	FCR	222.60	2.87
LP_{min}	ADG	0.266	0.590
E_m	BF	0.612	3.48
p	DTS	0.634	143.14
w		5.75	

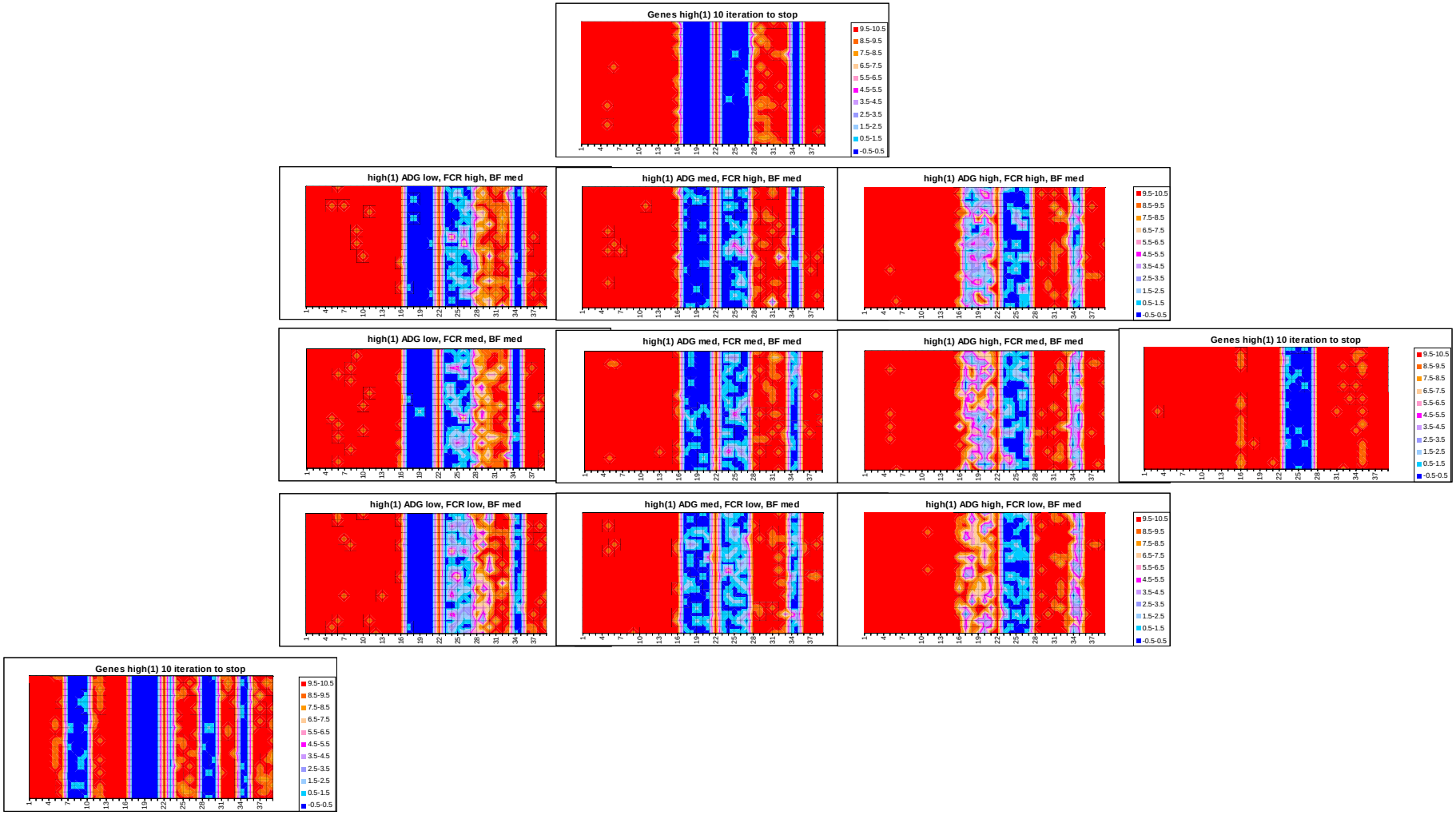

Figure 10.2: Solutions for multiple selection objectives, high diet.

10.2.3.2 Low diet

In the results in Figure 10.3, a horizontal visual comparison of the solution graphs, from left to right, shows that there is evidence of an effect as the weighting of average daily gain increases. This is most visible when the weighting is changed from medium to high. This can also be seen in the change in values in the corresponding table, Table 10.5.

In a vertical comparison, a small feed conversion ratio effect is evidenced. The effect is most clearly seen in the middle column.

The diagonal comparison shows strong evidence that back fat has an effect on the optimal solution when the low diet is being fed.

It can be seen that as the weight of average daily gain increases, the solution is trending towards the single objective average daily gain optimal solution. The weighting of average daily gain would not need to be increased much before the single objective optimal solution is achieved.

When the weighting of average daily gain and feed conversion ratio decrease, and hence the weighting of back fat increases, there is strong evidence that the solution is trending towards the back fat optimal solution. Again the weighting would not need to be changed much for the back fat single objective optimal solution to be achieved.

In general, under low diet conditions, feed conversion ratio is the dominant factor. However this is not the case when the weighting for average daily gain is low, as the back fat factor is more prominent in this situation.

Table 10.5: Mean performances for multiple selection objective solutions, low diet.

Population			
Pd_{max}	**FCR**	170	2.419
LP_{min}	**ADG**	0.7	0.924
E_m	**BF**	0.485	14.64
p	**DTS**	0.9	77.28
w		5.05	

FCR Optimal			
Pd_{max}	**FCR**	281.94	1.88
LP_{min}	**ADG**	0.620	1.420
E_m	**BF**	0.468	12.58
p	**DTS**	1.077	50.18
w		5.75	

			ADG: Low		ADG: Med		ADG: High	
FCR	**High**	Pd_{max}	279.87	1.91	280.53	1.90	279.71	1.92
		LP_{min}	0.512	1.333	0.597	1.410	0.619	1.444
		E_m	0.507	11.56	0.493	12.31	0.484	12.96
		p	1.027	53.51	1.080	50.54	1.124	49.32
		w	5.75		5.73		5.62	
	Med	Pd_{max}	280.16	1.95	279.91	1.91	279.60	1.94
		LP_{min}	0.458	1.265	0.592	1.412	0.618	1.450
		E_m	0.549	10.65	0.498	12.29	0.487	13.08
		p	0.993	56.43	1.085	50.47	1.136	49.14
		w	5.75		5.72		5.58	
	Low	Pd_{max}	261.29	2.23	280.17	1.92	279.25	1.95
		LP_{min}	0.272	0.836	0.588	1.415	0.626	1.456
		E_m	0.607	6.35	0.516	12.23	0.490	13.25
		p	0.735	87.20	1.098	50.34	1.150	48.90
		w	5.75		5.69		5.55	

ADG Optimal			
Pd_{max}	**FCR**	281.94	1.95
LP_{min}	**ADG**	0.647	1.480
E_m	**BF**	0.468	13.66
p	**DTS**	1.166	48.23
w		5.52	

BF Optimal			
Pd_{max}	**FCR**	222.60	2.64
LP_{min}	**ADG**	0.266	0.650
E_m	**BF**	0.612	4.62
p	**DTS**	0.634	119.01
w		5.75	

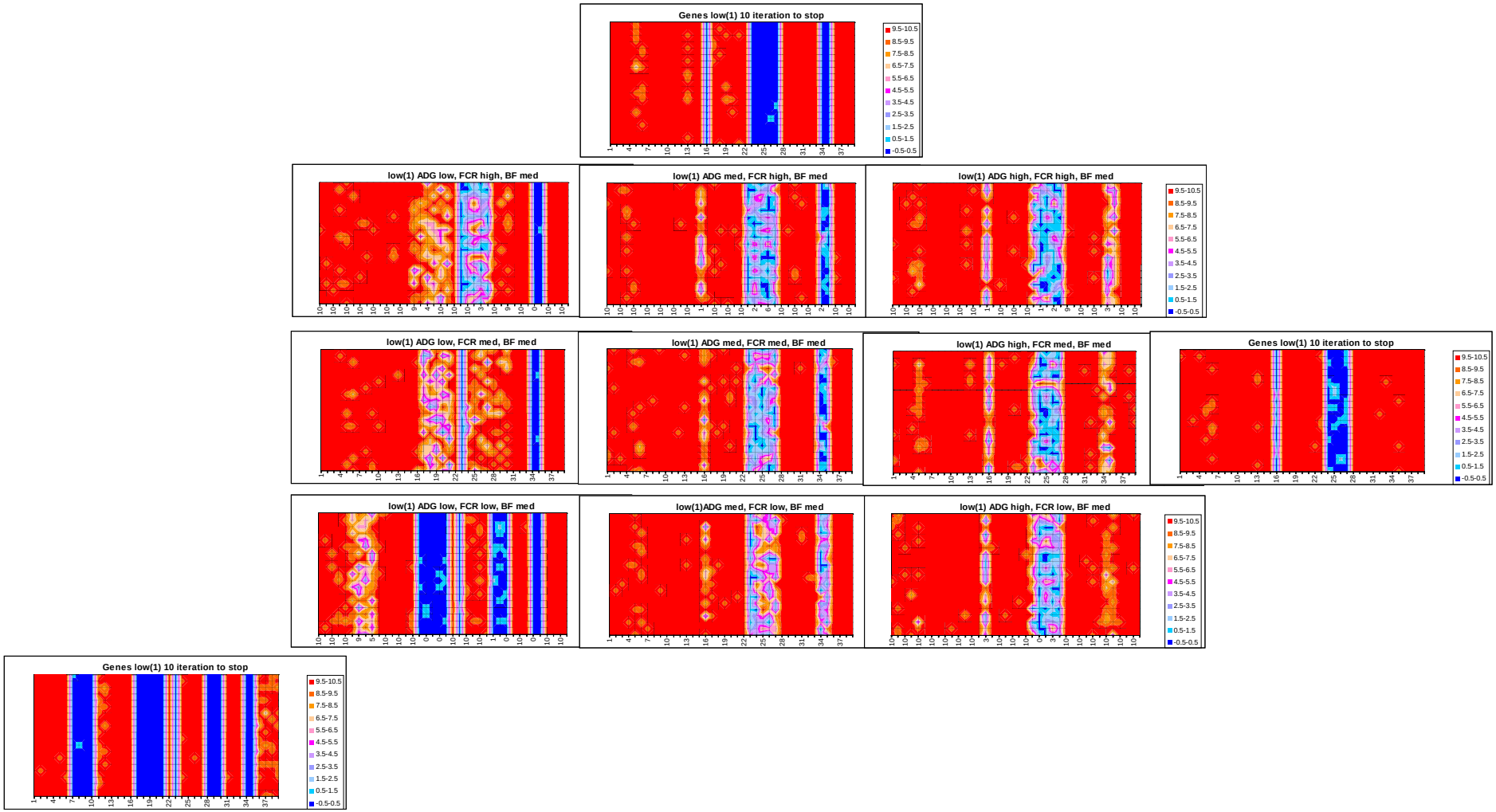

Figure 10.3: Solutions for multiple selection objectives, low diet.

10.2.3.3 Comparison of low and high diets

For both the high and low diets, the majority of the optimal solutions are in the vicinity of the single objective feed conversion ratio optimal solution for the diet. This emphasizes the importance that the feed conversion ratio plays in the optimal solution. Feed conversion ratio is a main driver for the genome as FCR combines and summarises all the sub phenotypic traits. FCR is feed intake over average daily gain which is approximately p over ADG, and ADG is a function of Pd_{max} and w, and to some extend ADG is also a function of E_m and LP_{min}. However average daily gain and back fat have relatively weak links to some of the sub phenotypic traits.

The scale of the effects for the high diet is less than for the low diet. That is, when a low quality diet is being fed, the weighting of the parameters in the objective have a larger effect on the solution than when a high quality diet is being fed. In particular, a low quality diet allows the back fat factor to become more influential.

On the high diet, the fact that back fat has little influence on the solution suggests that there is very little movement in back fat value for solutions around the feed conversion ratio optimal. On the low diet, back fat has more influence, suggesting that there is greater change in back fat performance of genotypes around the feed conversion ratio optimal solution.

When a low diet is being fed, the selection of the objective ratios can have a large influence on the ultimate results of a breeding program, with different ratios giving different genotypes. Whereas, when feeding a high diet, breeding programs will produce the same general genotype regardless of quite large variations in ratios.

What is clear is that the optimal solution is dependent on the diet fed. The same breeding program with different diets will result in different genotypes.

10.3 Comparison of single and multiple objectives

As seen previously in Table 10.4 and Table 10.5, higher ADG, better FCR, and lower BF are always observed, independent of the selection objective, in the high diet optimal solutions when compared to the low diet optimal solutions. The only exception is when

pigs are selected on BF alone, in which case higher ADG and better FCR are observed with the low diet. It is well recognized that that nutrient intake should not be limiting when pigs are selected for maximal growth performances. An example is a selection experiment (Cameron 1994, Cameron and Curran 1994a,b, Cameron et al. 1994) on lean growth rates under ad-libitum or restricted feeding in Large White and Landrace pigs. A greater response to selection for average daily gain and lean growth rate was obtained with ad-libitum feeding.

On the high diet, across the section objectives, the observed values of Pd_{max} ranged between 281.05 and 281.94 (Table 10.4) and therefore Pd_{max} is always at or close to the maximum value. The only exception is when selection is on minimal back fat alone. This means that selecting for high Pd_{max}, which is closely related to lean growth rate, will improve ADG, FCR, and BF. This has been observed in practice. A general analysis of 686,541 records of Yorkshire, Duroc, Hampshire, and Landrace pigs (Chen et al. 2002) reported genetic correlations ranging between -0.80 and -0.86 for growth rate and days to slaughter, and between -0.32 and -0.41 for lean growth rate and back fat thickness.

Figure 10.4, Figure 10.5, and Figure 10.6 give the relationship between the sub-phenotypic traits (LP_{min}, E_m, p, and w) and the best ADG, FCR, and BF, respectively, obtained for the different selection objectives when the high diet is fed.

As the response on ADG increases up to 1.45kg/d, there are no major changes observed in the sub-phenotypic traits (Figure 10.4). However higher ADG is achieved by increasing energy intake, p, decreasing water in the whole body, w, and increasing fat, increasing LP_{min}. Energy requirements for maintenance is also slightly reduced, E_m.

Improving FCR, Figure 10.5, is achieved by a reduction in energy intake, p, having less fat in the whole body, LP_{min}, and more water, w. The energy requirement for maintenance, E_m, is also reduced, which means that more feed is used for growth. A similar pattern is observed when selection pressure to reduce back fat is increased, Figure 10.6, except for the energy requirements for maintenance, which increases as back fat increases. This means that less energy is available to be used for lipid growth.

Overall the optimisation results show that a decrease in energy intake, *p*, is observed when selection pressure on back fat and feed conversion ratio is increased. This has been observed in many selection experiments conducted under ad libitum feeding. Feed intake was reduced when pigs were selected for low back fat (Bereskin et al. 1975), low back fat and low FCR and high growth rate (Wood et al. 1983, Waefler 1982, Ellis et al. 1979), and low back fat and high growth rate (Sather and Freeden 1978, Cleveland et al 1982 and 1983).

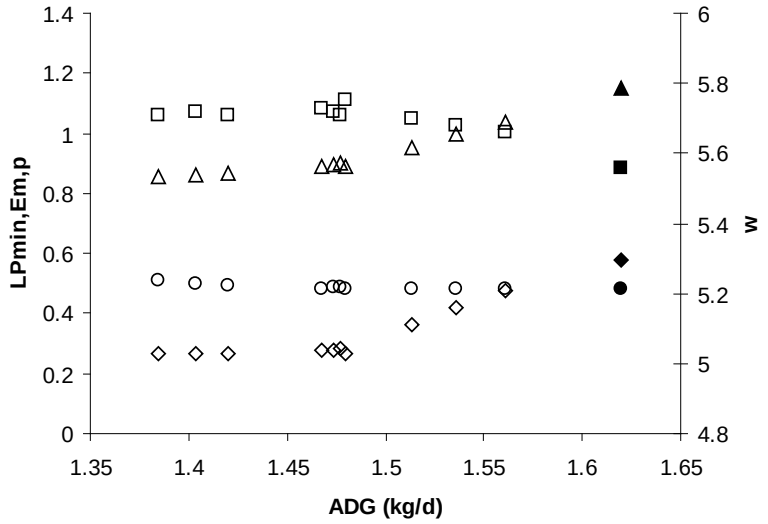

Figure 10.4: Sub-phenotypic changes as ADG selection response increases. Sub-phenotypic traits LP_{min} (◊), w (□), E_m (○), and p (Δ). ADG selection response increases from left to right. The values for the single objective selection on ADG are indicated by filled symbols.

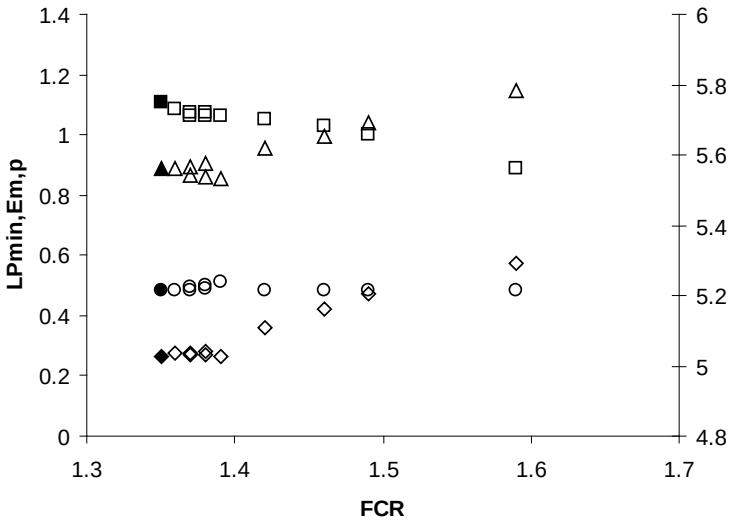

Figure 10.5: Sub-phenotypic changes as FCR selection response increases. Sub-phenotypic traits LP_{min} (◊), w (□), E_m (○), and p (Δ). FCR selection response increases from right to left. The values for the single objective selection on FCR are indicated by filled symbols.

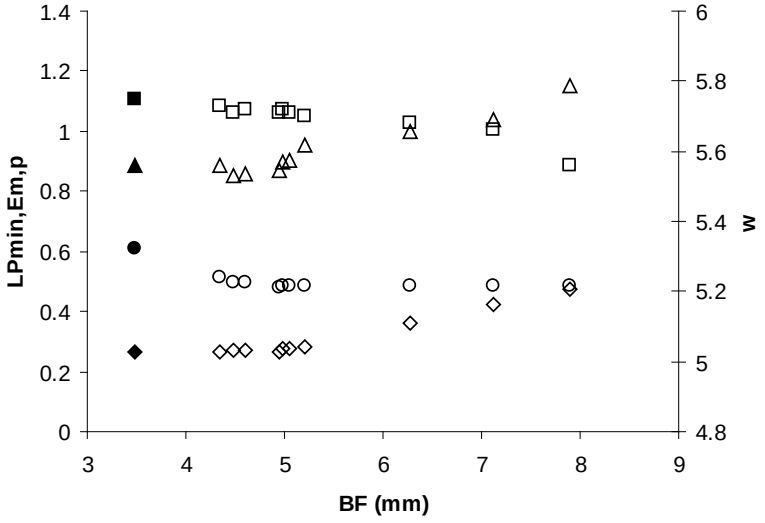

Figure 10.6: Sub-phenotypic changes as BF selection response increases. Sub-phenotypic traits LP_{min} (◊), w (□), E_m (○), and p (Δ). BF selection response increases from right to left. The values for the single objective selection on BF are indicated by filled symbols.

Chapter 11 – Conclusions

11.1 Conclusions

Chapter 2 and Chapter 3 contain an investigation into the effects of a small number of loci and alleles on the additive model, during one generation of breeding, when compared to infinitesimal predictions. It is concluded that when there are less than approximately 15 loci, bias starts to become apparent in one generation of breeding, with the infinitesimal model overestimating the genetic gain that will be achieved. When the additive model was analysed it was found that heritability is non-constant when the number of loci is small, as opposed to constant heritability in the infinitesimal model. When the non-constant heritability was taken into account for the prediction, the bias is removed.

In Chapter 4, various distribution shapes for the allele distribution were also investigated and it is concluded that as long as the allele distributions are symmetric on average, there is no effect on the quality of the infinitesimal predictions. It is noted that this requirement for symmetry is independent of the number of loci and alleles. Therefore if the allele distributions are skewed by a breeding program, the infinitesimal model will not be able to produce accurate prediction, even if Bulmer effects are taken into account.

Chapter 4 also contained an investigation into the possibility of linkage effects on the accuracy of the infinitesimal predictions, where linkage is a measure of the distance between loci on the chromosomes. It is concluded from the results that linkage has no effect on the accuracy of the prediction in the first and second generation of breeding.

Chapter 5 introduced and described the GE-Pig model. This model has been developed to take the additive effects of alleles and combine them with a pig growth model, thus allowing for the prediction of the performance of various combinations of the alleles. GE-Pig gives the capability for optimization algorithms to be applied. The optimal combination of alleles can thus be found for any given objective.

Three optimization algorithms, Genetic Algorithm, Tabu Search, and Simulated Annealing, which were used throughout the optimizations of GE-Pig, are introduced in Chapter 6. These three algorithms were chosen to give a range of approaches to the problem. The Tabu search has a forced movement around the objective plane, Simulated Annealing has a local search movement and the Generic Algorithm makes use of the biological breeding ideas on which the problem is based.

In assessing the performance of the three algorithms, it is concluded that, for all three objectives, the Genetic Algorithm produces the least variable results. The solutions from the Genetic Algorithm are of similar or better quality than those from the other two algorithms. The Genetic Algorithm is also able to produce good quality solutions quickly. Tabu Search generally produced the same quality solutions as those from the Genetic Algorithm, although the results were more variable. Tabu Search however did not perform as well with the back fat objective when on the high diet. This particular version of Tabu Search does not have the ability to produce quick solutions. The performance of Simulated Annealing was more variable, and of a lower quality, than the Genetic Algorithm. The recommendation, in a practical situation, would be to use the Genetic Algorithm, as this algorithm has the superior performance. Also the Genetic Algorithm was originally based on biological breeding ideas, and thus more naturally fits the data structure present in the GE-Pig model.

In Chapter 7, Chapter 8, and Chapter 9, three different objectives, maximising average daily gain, minimizing feed conversion ratio, and minimizing back fat, are investigated. These objectives were investigated over two different quality diets, a high quality and a low quality.

In all the results of the simulations there were no indications of multiple optima. This is of great benefit to the farmer, as it indicates a single target genotype for a given objective. It can also be seen that all the optimal solutions have the alleles at the extremes of their distributions. In addition, for each optimal solution, the alleles in the same grouping created during the parameter generation have the same optimal values.

In general, the optimal solution for each diet and objective is different. The differences between the optimal solutions across the diets for each objective are examples of

genetic by environmental interactions. The only two optimal solutions that are identical are the solutions for the back fat on both the high and low diet. However there are many similarities between the average daily gain solutions and the feed conversion ratio solutions.

When the theoretical minimum number of generations of selection to the optimal solution is considered, the optimal solutions for average daily gain are genetically the furthest away from the population average, whilst the optimal solutions for feed conversion ratio are genetically the closest. Thus, for the farmer, it will take longer for a breeding program to achieve the optimal solutions for average daily gain, followed by the back fat optimal solutions, then the feed conversion ratio optimal solutions. Also, for the back fat optimal solutions, the low diet optimal solution is genetically further from the mean than the high diet solution. This is not true for the other two objectives.

Chapter 10 introduce a multiple selection objective, being a range of relative economic values (REV) applied to the three individual selection objectives, with various REV's investigated. For both the high and low diet, the majority of the multiple selection objective solutions are in the vicinity of the respective optimal solutions for the feed conversion ratio selection objective. It is thus concluded that feed conversion ratio is the most prominent factor. It is also seen that when a low quality diet is fed, the REV in the selection objective has a larger effect on the optimal solution than when a high quality diet is fed. In particular, it has been shown that a low quality diet allows the back fat factor to become more influential.

It is concluded that diet plays a major role when multiple selection objectives are under consideration. When a low diet is fed, the selection of the REV can have a large influence on the ultimate results of a breeding program, with different REV's giving different genotypes. However under high diet conditions, breeding programs will produce the same general genotype regardless of large variations in REV's. It is thus clear that selection and consistency of diet is of importance in the performance of any breeding program. A change in diet will result in changes in the optimal genotype and hence the performance of the breeding program, and ultimately the economic return to the farmer.

11.2 Future directions

In the future, as additive information regarding alleles and their effects becomes available, the theoretical values used for the alleles in this model could be augmented, and eventually replaced, by the real world values. If GE-Pig could then also be combined with DNA testing of the pigs on farms, then simulations of various breeding combinations could be made and optimal breeding strategies for specific farms developed.

The GE-Pig model could also be extended by the inclusion of a linear program, for diet optimization, and a price schedule, as shown in Figure 11.1. Optimization algorithms could then be applied to both the genome and the diet specification to look at maximising the profit of the pig farmer. This extension would also allow for investigation of the effect that prices have on the optimal genome and the best diet.

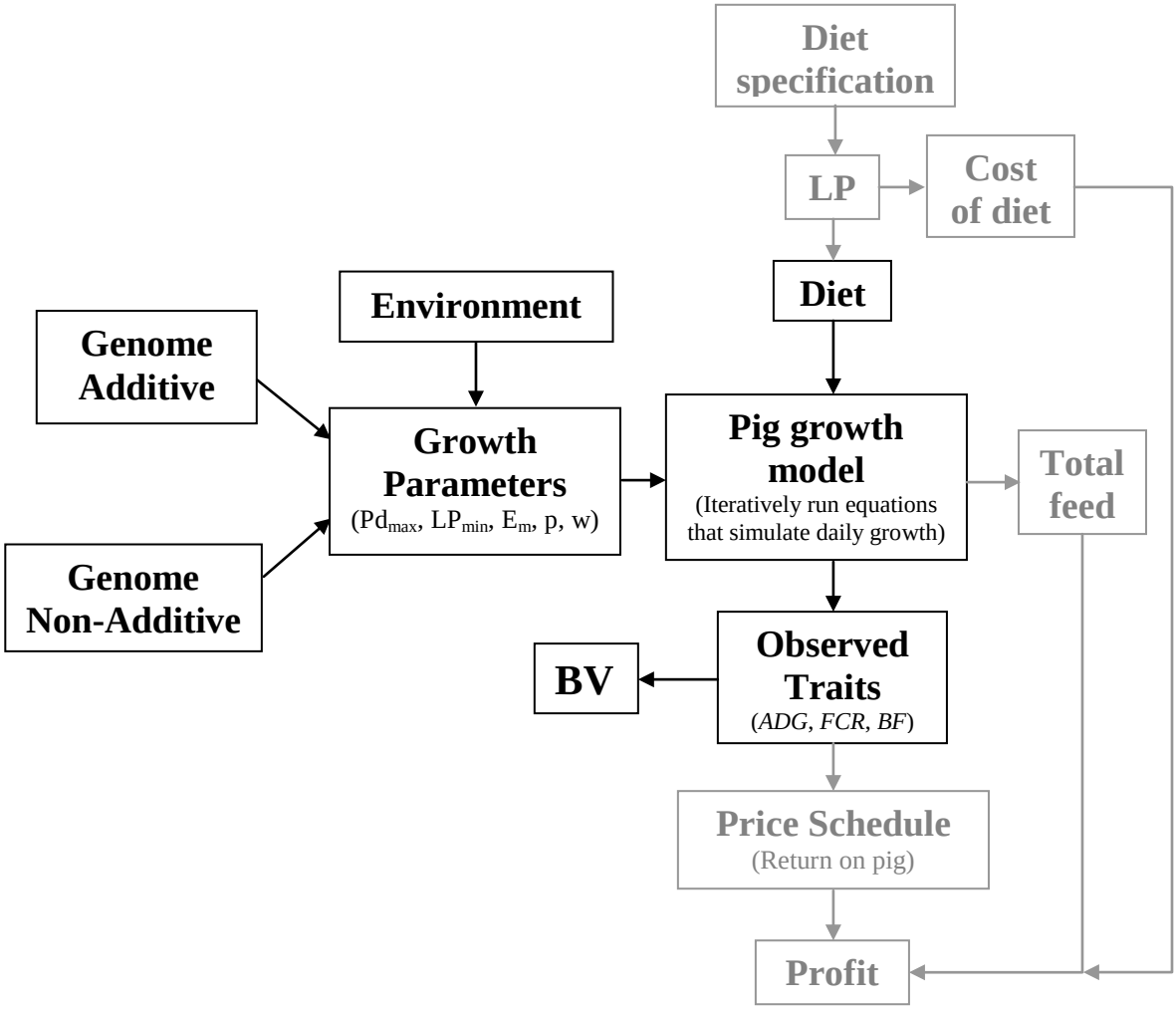

Figure 11.1: Possible GE-Pig model extension.
GE-Pig model shown in black, possible extension is shown in grey.

All the optimal solutions produced to date have all the alleles at the extremes of their distributions. With this knowledge, a new algorithm could be developed that restricts the search space so that each locus has two possible values. For the GE-Pig model setup that has been used, this would dramatically reduce the search space from $6^{2^{39}} = 4.96 \times 10^{60}$ to $2^{39} = 5.50 \times 10^{11}$.

It has also been noted that, for each optimal solution, the alleles in the same grouping created during the parameter generation have the same optimal values. This information could be further used to reduce the search space, by optimizing on the groups rather than optimizing each locus. This would result in a search space size of $2^{15} = 32768$ for the given setup of the GE-Pig model. A search space of this size is sufficiently small for the possibility of every point to be evaluated for every combination, removing the need for an optimization algorithm. Table 11.1 below presents the results from a Simple Search, which does a complete evaluation of the reduced search space size of 32768, and compares it to the Genetic Algorithm results which had a similar running time. As can be seen the Genetic Algorithm results are very close to the Simple Search results and only varied in a few loci by a few allele values. The Simple Search results also did not find multiple optimal solutions. It needs to be noted however that future enhancements, such as dominance effects, to the model would first need to be evaluated before using a Simple Search method to confirm that the optimal solutions still lie at the extremes of their distributions.

Table 11.1: Comparison of Simple Search results with Genetic Algorithm results.

Genetic Algorithm results are listed for the N=50 S=5 setup which investigated on average a similar number of genomes.

	ADG High		ADG Low		FCR High		FCR Low		BF High		BF Low	
	SS	GA	SS	GA	SS	GA	SS	GA	SS	GA	SS	GA
Loci		Mean (Range)		Mean (Range)		Mean (Range)		Mean (Range)		Mean (Range)		Mean (Range)
1-4	10	10.00 (10-10)	10	9.91 (9-10)	10	10.00 (10-10)	10	10.00 (10-10)	10	10.00 (10-10)	10	9.99 (9-10)
5-6	10	10.00 (10-10)	10	9.68 (8-10)	10	10.00 (10-10)	10	9.48 (8-10)	10	9.48 (8-10)	10	9.95 (9-10)
7-10	10	10.00 (10-10)	10	10.00 (10-10)	10	9.99 (9-10)	10	10.00 (10-10)	0	0.45 (0- 3)	0	0.01 (0- 1)
11-12	10	10.00 (10-10)	10	10.00 (10-10)	10	10.00 (10-10)	10	10.00 (10-10)	10	9.43 (7-10)	10	9.33 (8-10)
13	10	10.00 (10-10)	10	9.70 (8-10)	10	10.00 (10-10)	10	9.80 (9-10)	10	10.00 (10-10)	10	10.00 (10-10)
14-15	10	10.00 (10-10)	10	10.00 (10-10)	10	10.00 (10-10)	10	10.00 (10-10)	10	10.00 (10-10)	10	10.00 (10-10)
16	10	8.40 (6-10)	0	0.50 (0- 2)	10	9.15 (5-10)	0	0.10 (0- 1)	10	10.00 (10-10)	10	10.00 (10-10)
17-21	10	9.89 (9-10)	10	9.99 (9-10)	0	0.00 (0- 0)	10	9.83 (8-10)	0	0.00 (0- 0)	0	0.00 (0- 0)
22	10	9.95 (9-10)	10	10.00 (10-10)	10	10.00 (10-10)	10	10.00 (10-10)	10	10.00 (10-10)	10	10.00 (10-10)
23	0	1.15 (0- 3)	10	9.85 (9-10)	0	0.00 (0- 0)	0	0.00 (0- 0)	0	0.00 (0- 0)	0	0.00 (0- 0)
24-27	0	0.10 (0- 1)	0	0.28 (0- 2)	0	0.05 (0- 1)	0	0.01 (0- 1)	10	9.09 (4-10)	10	9.95 (9-10)
28-30	10	9.98 (9-10)	10	10.00 (10-10)	10	9.40 (7-10)	10	9.98 (9-10)	0	0.18 (0- 2)	0	0.00 (0- 0)
31-33	10	9.95 (9-10)	10	9.97 (9-10)	10	9.58 (8-10)	10	10.00 (10-10)	10	9.82 (8-10)	10	9.93 (9-10)
34-35	10	9.53 (7-10)	10	9.90 (9-10)	0	0.03 (0- 1)	0	0.00 (0- 0)	0	0.25 (0- 1)	0	0.00 (0- 0)
36-39	10	9.98 (9-10)	10	10.00 (10-10)	10	9.98 (9-10)	10	10.00 (10-10)	10	9.46 (6-10)	10	9.38 (6-10)

References

Ali, M. C.; Khompatraporn; Zabinsky, Z., 2005: A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems, Journal of Global Optimization, Vol. 31, pp. 635–672.

Allan, B. B.; Fredeen, H. T.; Weiss, G. M., 1985: The sensitivity of two-trait selection index to changes in economic weights and genetic parameter estimates. Canadian Journal of Animal Science 65 (1): 21-29

Baldwin, R. L.; Sainz, R. D., 1995: Energy partitioning and modelling in animal nutrition. Animal review in nutrition, Vol. 15, pp. 191-211.

Belisle, C. J., 1992: Convergence theorems for a class of simulated annealing algorithms on $\mathbb{R}^d$. Journal of Applied Probability. Vol. 29, pp. 885-895.

Bereskin, B.; Davey, R. J.; Peters, W. H.; Hetzer, H. D., 1975: Genetic and environmental effects and interaction in swine growth and feed utilisation. Journal of Animal Science. Vol. 40, pp. 53-60.

Beyer, W. H., 1987: CRC Standard Mathematical Tables, 28th ed.. Boca Raton, FL: CRC Press, pp. 531 and 533.

Blum, J. K., 1983. Populationanalyse der schweizerischen Schweinerassen. Eidgenossischen Technischen Horschule Zurich, Dissertation 7412.

Bulmer, M. G., 1971: The effect of selection on genetic variability. American Naturalist. 105:943, 201-211.

Cameron, N.D. 1994a: Selection for component of efficient lean growth rate in pigs. 1. Selection pressure applied and direct responses in Large White herd. Animal Production. Vol. 59, pp. 251-262.

Cameron, N. D.; Curran, M. K., 1994a: Selection for component of efficient lean growth rate in pigs. 1. Selection pressure applied and direct responses in Landrace herd. Animal Production. Vol. 59, pp. 263-269.

Cameron, N. D.; Curran M. K., 1994b: Selection for components of efficient lean growth rate in pigs. 4. Genetic and phenotypic parameters estimates and correlated responses in performance test traits with *ad libitum* feeding. Animal Production, Vol. 59, pp. 281-291.

Cameron, N. D.; Curran, M. K.; Kerr, J. C., 1994: Selection for component of efficient lean growth rate in pigs. 3. Responses to selection with a restricted feeding regime. Animal Production. Vol. 59 pp. 271-279.

Cameron, N. D.; Penman, J. C.; Fisken, A. C.; Nute, G. R.; Perry, A. M.; Wood, J. D., 1999: Genotype with nutrition interactions for carcass composition and meat quality in pig genotypes selected for components of efficient lean growth rate. Animal Science, Vol. 69 (1), pp. 69-80.

Cerny, V., 1985; Thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm. Journal of optimization theory and its applications, Vol. 45, pp. 41-55.

Chen, P.; Baas, T. J.; Mabry, J. W.; Dekkers, J. C. M.; Koehler K. J., 2002: Genetic parameters and trend for lean growth rate and its components in US Yorkshire, Duric, Hampshire, and Landrace pis. Journal of Animal Science. Vol. 80, pp. 2062-2070.

Clayton, G. A.; Morris, J. A.; Robertson, A., 1957: An experimental check on quantitative genetic theory. I. Short-term responses to selection. J. Genet. 55:131-151.

Cleveland, E. R.; Cunningham, P. S.; Peo, E. R., 1982: Selection for lean growth in Swine. Journal of Animal Science Vol. 54, pp. 719-727.

Cleveland, E. R.; Johnson, R. K.; Mandigo, R. W., 1983: Index selection and feed intake restriction in swine. I. Effect on rate and composition of growth. Journal of Animal Science. Vol. 56, pp. 560-569.

Commonwealth Agricultural Bureaux, 1981: The Nutrient requirements of pigs : technical review. Slough, England

Curik, I.; Solkner, J.; Stipic, N., 2002: Effects of models with finite loci, selection, dominance, epistasis and linkage on inbreeding coefficients based on pedigree and genotypic information. J. Anim. Breed. Genet. 119:101-115.

Dantzig, G. B.; Thapa, Mukund N., 1997: Linear Programming 1: Introduction. Springer-Verlag New York, Inc.

Davis, L., 1985: Job-shop scheduling with genetic algorithm. Proceedings of the international conference on genetic algorithms and their applications, pp. 136-140.

Dekkers, J. C. M.; Hospital, F., 2002: The use of molecular genetics in the improvement of agricultural populations. Nature, Vol. 3, pp. 22-32.

Dekkers, J. C. M., 2004. Commercial application of marker- and gene-assisted selection in livestock: Strategies and lessons. J Anim Sci, Vol. 82, pp. 313-328.

De Lange, C. F. M., 1995: Modelling growth in the pig. Eds. P.J Moughan, M.W.A. Verstegen and M.I. Visser-Reyneveld. (Wageningen Pers: Wageningen). pp 71-85.

De Lange, C.F.M., Morel, P.C.H. and Birkett, S.H., 2003: Modelling chemical and physical body composition of the growing pig. Journal of Animal Science 81 (E. Suppl. 2):E159-E165.

De Lange, C.F.M., Morel, P.C.H. and Birkett, S.H., 2008: Mathematical representation of the partitioning of retained energy in the growing pig. Chapter 14. pp. 316-338. In. (Ed. J. France and E Kebreab): Mathematical Modelling in Animal Nutrition. CABI Publishing, Cambridge, MA 02139, USA.

Devir, S.; Zur, B.; Maltz, E.; Genizi, A.; Antler, A., 1995: A model for the prediction of dairy cow body weight based on a physiological time scale. Journal of agricultural science, Cambridge, Vol. 125, pp. 13-21.

Doeschl-Wilson, A. B.; Knap, P. W.; Kinghorn, B. P.; Van der Steen, H. A. M., 2007: Using mechanistic animal growth models to estimate genetic parameters of biological traits. Animal, Vol. 1, pp. 489-499.

Falconer, D. S.; MacKay, T. F. C., 1996: Introduction to Quantitative Genetics Ed. 4. Longman, London, UK.

Faw, T. F.; Tan, C. Y.; Chan, W. T., 1994: Road-maintenance planning using genetic algorithms. Journal of transportation engineering, Vol. 120, pp. 693-722.

Feller, W., 1945: The Fundamental Limit Theorems in Probability. Bull. Amer. Math. Soc. 51, 800-832.

Finlayson, J. D.; Cacho, O. J.; Bywater, A. C., 1995: A simulation model of grazing sheep: I. Animal growth intake. Agricultural systems, Vol. 48, pp. 1-25.

Fisher, H.; Thompson, G. L., 1963: Probabilistic learning combinations of local jobshop scheduling rules. Industrial scheduling, J. F. Muth and G. L. Thompson (eds.), Prentice-Hall, pp. 225-251.

Fisher, R.A., 1918. The correlation between relatives on the supposition of Mendelian inheritance. Transactions of the Royal Society of Edinburgh 52:399-433.

Fouskakis, D.; Draper, D., 2002. Stochastic Optimization: A Review. International Statistical Review, Vol. 70, No. 3, pp. 315-349.

Frankham, R., 1990: Are responses to artificial selection for reproductive fitness characters consistently asymmetrical? Genet. Res. Vol. 56 (1), pp. 35-42

Gibson, J. P., 1994: Short-term gain at the expense of long-term response with selection on identified loci. Proc. 5th World cong. Genet. Appl. Livest. Prod. Guelph. Vol. 21, pp. 201-204.

Glover, F., 1963: Parametric combinations of local jobshop rules. Chapter IV, ONR Research Memorandum No. 117, GSIA, Carnegie-Mellon University, Pittsburgh, PA.

Glover, F., 1965: A multiphase-dual algorithm for the zero-one integer programming problem. Operations research, Vol. 13, pp. 879-919.

Glover, F., 1966: An algorithm for solving the linear integer programming problem over a finite additive group, with extensions to solving general and certain non-linear integer programs. CRC 66-29, University of California at Berkeley.

Glover, F., 1968: Surrogate constraints. Operations research, Vol. 16, pp. 741-749.

Glover, F., 1969: Integer programming over a finite additive group. SIAM Journal on control, Vol. 7, pp. 213-231.

Glover F.; Laguna M., 1997: Tabu Search. Kluwer Academic Publishers, Boston.

Goddard, M. E., 2001: The validity of genetic models underlying quantitative traits. Livestock Production Science 72:117-127.

Goddard, M. E., 2003: Animal breeding in the (post-) genomic era. Animal Science. Vol. 76 (3), pp. 353-365.

Hajek, B., 1988: Cooling Schedules for optimal annealing. Mathematics of Operations Research. Vol. 13, pp. 311-329.

Hill, W. G.; Caballero, A., 1992: Artificial selection experiments. Annu. Rev. Ecol. Syst. 23:287-310.

Holland, J.H., 1975: Adaptation in natural and artificial systems. The University of Michigan press, Ann Arbor, MI.

Homifar, A.; Qi, C. X.; Lai S. H., 1994: Constrained optimization via genetic algorithms. Simulation, Vol.62, pp. 242-253.

Hov, E. S. H.; Angari, N.; Hong, R., 1994: A genetic algorithm for multiprocessor scheduling. IEEE Transactions on Parallel and Distributed Systems, Vol. 5, pp. 113-120.

Jeon, J. T.; Carlborg, O.; Tornsten, A.; Giuffra, E.; Amarger, V.; Chardon, P.; Andersson-Elkund, L.; Andersson, K.; Hansson, I.; Lundstrom, K.; Andersson, L., 1999: A paternally expressed QTL affecting skeletal and cardiac muscle mass in pig maps to the IGF2 locus. Nat. Genet 21:157-158.

Keener, H. M., 1979: Simulation of energy utilisation of bovine animals. Agricultural systems, Vol. 4, pp. 79-100.

Keightley, P. D., 2004: Mutational variation and long-term selection response. Pland Breeding Reviews, Vol. 24 (1), pp.227-247

Keightley, P. D.; Hill, W. G., 1987: Directional Selection and Variation in Finite Populations. Genetics, Vol. 117, pp. 573-582.

Kell, D. B., 2002: Genotype-phenotype mapping: genes as computer programs. Trends in Genetics Vol. 18 No. 11, pp. 555-559.

Kirkpatrick, S.; Gelatt, C.; Vecchi, M., 1983: Optimization by Simulated Annealing. Science Vol. 220, pp. 671-180.

Koong, L. J.; Garrett, E. N.; Rattray, P. V., 1975: A description of the dynamics of fetal growth in sheep. Journal of animal science, Vol. 41 (4), pp.1065-1068.

Koong, L. J.; Falter, K. H.; Lucas, H. L., 1982: A mathematical model for the joint metabolism of nitrogen and energy in cattle. Agricultural systems, Vol. 9, pp. 301-324.

Korver, S.; Tess, M. W.; Johnson, T., 1988: A model of growth and growth composition for beef bulls of different breeds. Agricultural systems, Vol. 27, pp. 279-294.

Krieter, J., 1986. Entwicklung von selectionsmethoden fur das Wachstums- und Futteraufnamevermogen beim Schwein. Christian-Albrechts-Universitat zu Kiel: Heft 31.

Leung, Y.; Wang, Y., 2001: An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Transactions on Evolutionary Computation, Vol. 5, pp. 41–53.

Liu, D. S.; Post, E.; Kulasiri, D.; Sherlock, R. A., 2003: Optimization of a complex simulation model. MODSIM 2003: International congress on modelling and simulation, vols1-4 pp. 1817-1822.

Locatelli, M., 2001: Convergence and first hitting time of simulated annealing algorithms for continuous global optimization. Mathematical Methods of Operations Research. Vol. 54, pp. 171-199.

Lundy, M.; Mees, A., 1986: Convergence of an annealing algorithm. Math Prog. Vol. 34, pp. 111-124.

Lush, J. L., 1937: Animal breeding plans and succeeding editions. Iowa State College Press, Ames.

Matsumoto, M.; Nishimura, T., 1998: Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator, ACM Trans. on Modelling and Computer Simulation Vol. 8, No. 1, January pp.3-30.

Mitchell, M., 2001: An Introduction to Genetic Algorithms, MIT Press, Massachusetts.

Mbaga, S. H.; Hill, W. G., 1997: Linear versus non-linear offspring parent regression in unselected random bred mice. J. Anim. Breed. Genet. 114:299-307.

Metropolis, N.; Rosenbluth, A.; Teller, A.; Teller, E., 1953: Equation of state calculation by fast computing machines. Journal of chemical physics. Vol. 21, pp. 1087-1091.

Michaelis, L.; Menten, M., 1913: Die Kinetik der Invertinwirkung. Biochem. Z. Vol. 49, pp. 333–369.

Morel, P.; Schwoerer, D; Rebsamen, A., 1988. Prise en consideration de la graisse intramusculaire dans l'indice de selection de Sempach. Der Kleinviehzuechter 36: 1341-1350.

Morel, P. C. H.; Alexander, D. L. J.; Sherriff, R. L.; Wood, G. R., 2001. Maximisation of gross margin in intensive livestock production. ACTA Horticulture Vol. 566, pp. 79-103

Morel, P. C. H.; Wood, G. R., 2005. Optimisation of nutrient use to maximize profitability and minimise nitrogen excretion in pig meat production systems. ACTA Horticulture Vol. 674, pp. 269-275

Morel, P. C. H.; Lopez-Villalobos, N.; Prince, Z., 1997. Genetic Relationship between pH45 and lean growth in pigs. In Manipulating Pig Production VI. P.D. Cranwell (Ed). 176

Morris, C. A.; Adam, J. L.; Johnson, D. L., 1978: Breeding objectives and selection indices for New Zealand pig improvement. New Zealand Journal of Experimental Agriculture Vol. 6 (4), pp. 259-266.

Noblet, J.; Herpin, P.; Dubois, S., 1992: Effect of recombinant porcine somatotropin on ernegy and protein utilization by growing pigs: interaction with capacity for lean tissue growth. Journal of Animal Science. Vol. 70, pp. 2471-2484.

Oltjen, J. W.; Bywater, A. C.; Baldwin, R. L.; Garret, W. N., 1986: Development of a dynamic model of beef cattle growth and composition. Journal of animal science Vol. 62, pp. 86-97.

Oljen, M. A. A. J. van; Koops, W. J.; Zandstra, T.; Kemp, B., 1993: Modelling foetal growth in pigs. Animal Production, Vol. 57, pp. 447-453

Parks, J. R., 1982: A theory of feeding and growth of animals. Springer-Verlag, Berlin.

Pirchner, F., 1988: Finding genes affecting quantitative traits in domestic animals. In: Weir, B. S.; Eisen, E. J.; Goodman, M. M.; Namkoong, G. (eds), Proceedings of Second International Conference on Quantitative Genetics, Sinauer Associates, Sunderland, MA.

Pomar, C.; Harris, D. L.; Minvielle, F., 1991: Computer simulation model of swine production systems: I. modelling the growth of young pigs. Journal of animal science, Vol. 69, pp. 1468-1488.

Rauw, W. M., 2007: Physiological consequences of selection for increased performance. 17th Conference of the Association for the Advancement of Animal Breeding and Genetics (Plenary speaker), Armidale, Australia, Proceedings AAABG: Vol. 17, pp. 240-247

Rudolph, G., 1994: Convergence analysis of canonical genetic algorithms. IEEE Transactions on Neural Networks, Vol. 5, pp. 96–101.

Sarker, R.; Mohammadian, M.; Yao, X., 2002: Editors, Evolutionary Optimization. Kluwer Academic Publishers, Boston.

Sather, A. P.; Freeden, H. T., 1978: Effect of selection for lean growth rate upon feed utilisation by market hog. Canadian Journal of Animal Science. Vol. 58, pp. 285-289.

Schmidt, E., 1988. Analyse des Wachstums und seine Beziehungen zum Schlachtkorperwert beim Schwein. Christian-Albrechts-Universitat zu Kiel: Heft 48.

Sherriff, R. L., 2000: Unpublished PhD dissertation work. Massey University.

Skorupski, M. T.; Garrick, D. J.; Blair, H. T.; Smith, W. C., 1995a: Economic values of traits for pig improvement. I. A simulation model. Australian journal of agricultural research, Vol. 46, pp. 285-303.

Skorupski, M. T.; Garrick, D. J.; Blair, H. T.; Smith, W. C., 1995b: Economic values of traits for pig improvement. II. Estimates for New Zealand conditions. Australian journal of agricultural research, Vol. 46, pp. 305-318.

Spelman, R. J.; Keehan, M.; Obolonkin, V.; Coppieters, W., 2006. Application of genetic information in a dairy cattle breeding scheme. Proc. Assoc. Advmt, Breed. Genet. Vol. 17, pp. 471-478.

SUISAG, 2006. Chiffres et projets 2006.

SUISAG, 2007. Zahlen und Projekte 2007.

Thompson, J. M.; Parks, J. R.; Perry, D., 1985: Food intake, growth and body composition in Australian merino sheep selected for high and low weaning weight. Animal Production, Vol. 40, pp. 55-70.

Thompson, J. M.; Barlow, R., 1986: The relationship between feeding and growth parameters and biological efficiency in cattle and sheep. Dikerson, G. E., Johnson, R. K. (Eds.), 3rd world congress in genetics applied to livestock production, Vol. XI. Lincoln, pp. 271-282.

Tjalling, J. Y., 1995: Historical development of the Newton-Raphson method. SIAM Review 37 vol. 4, pp. 531–551.

Venkatasubramanian, V.; Chank, K.; Caruthurs, J. M., 1994: Computer-aided molecular design using genetic algorithms. Computer & chemical engineering, Vol. 118, pp. 833-844.

Verrier E.; Colleau, J. J.; Foulley, J. L., 1991: Methods for predicting response to selection in small populations under additive genetic models: a review. Livest. Prod. Sci., Vol. 26, pp. 93-114.

Vetharaniam, I.; McCall, D. G.; Fennessy, P. F.; Garrick, D. J., 2001: A model of mammalian energetics and growth: model development. Agricultural systems, Vol. 68, pp. 55-68.

Vignal, A; Milan, D.; SanCristobal, M.; Eggen, A., 2002: A review on SNP and other types of molecular markers and their use in animal genetics. Genet. Sel. Evol. Vol. 34, pp. 275-305

Wafler, P. 1982. Indexslection in entgegensetzter Richung- ein Selektionsexperiment mit Schweinen. Eidgenossischen Technischen Horschule Zurich, Dissertation 6979.

Wallach, D.; Charpenteau, J. L., Elsen, J. M., 1986: Weight gain in grazing sheep: a detailed comparison between models. Agricultural systems, Vol. 19, pp. 211-248.

Whittemore, C.T. Green, D.M. and Knap, P.W., 2001 a: Technical review of energy and protein requirements of growing pigs: protein. Animal Science Vol. 73, pp. 363-373.

Whittemore, C.T. Green, D.M. and Knap, P.W., 2001 b: Technical review of energy and protein requirements of growing pigs: energy. Animal Science Vol. 73, pp. 199-215.

Wood, J. D.; Wheleman, O. P.; Ellis, M.; Smith, W. C.; Laird, R., 1983: Effect of selection for low back fat thickness in pigs on the sites of tissue deposition in the body. Animal Production. Vol. 36, pp. 389-397.

Yao, X.; Liu, Y.; Lin, G., 1999. Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation, Vol. 3, pp. 82–102.

Appendix A – Pig growth equations

An overview of the pig growth model is given in Figure A-1.

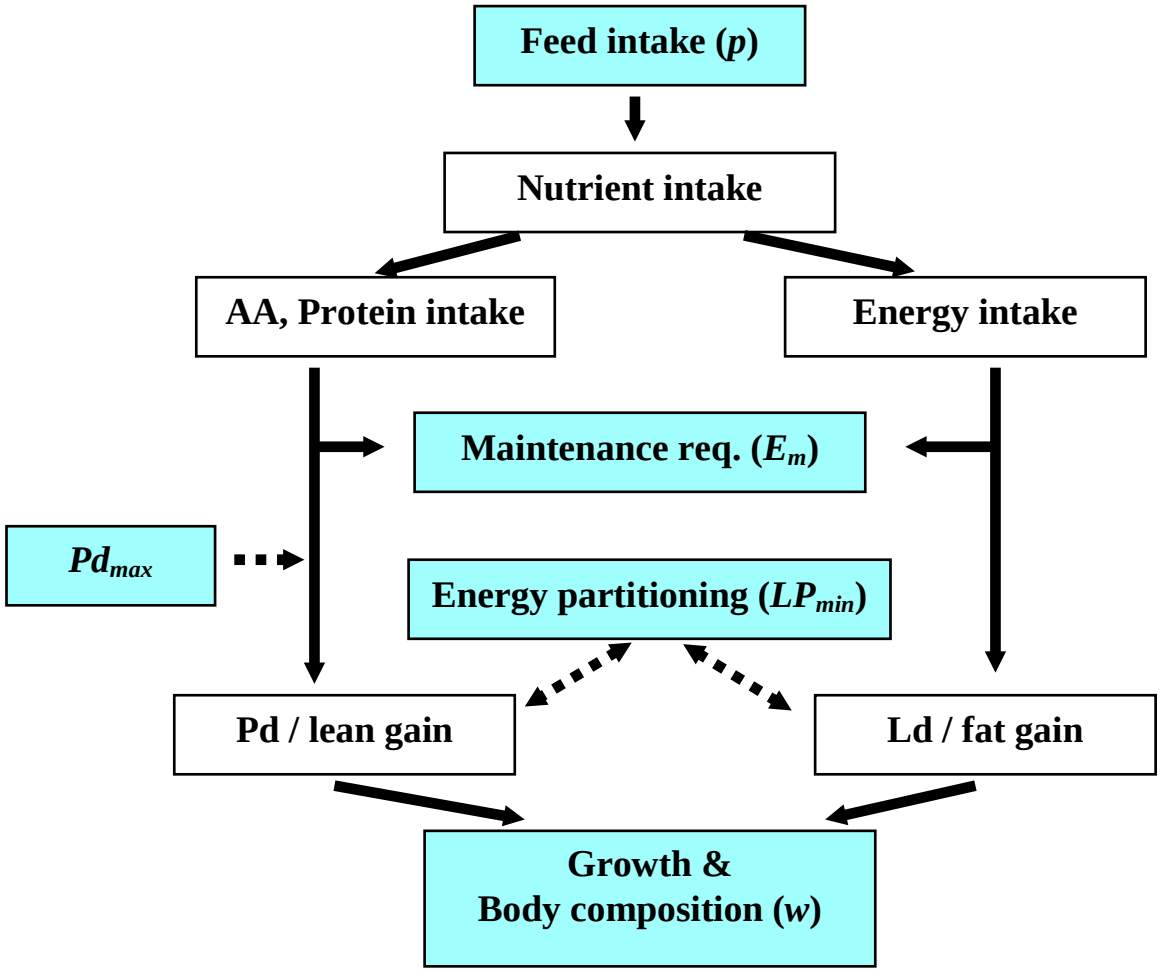

Figure A-1: Pig growth model.

The coloured sections are influenced by the five growth parameters.

The initial composition of the pig is calculated by firstly using a Newton-Raphson (Tjalling, 1995) method for calculating the protein in the pig in kg's, P_{pig}. This is achieved by repeatedly calculating the following equations in sequence until $-0.005 < F < 0.005$ where P_{pig} is initialised with the value of 3.09.

$$F = (1.189 + LP_{\min})P_{pig} + 4.89P_{pig}^{0.855} - 0.95LW_{pig}$$

$$\nabla F = (1 + 0.189 + LP_{\min}) + (4.899 \times 0.855)\, P_{pig}^{(0.855-1)}$$

$$P_{pig} = P_{pig} - \frac{F}{\nabla F}$$

Next lipid in the pig in kg's, $L_{pig,}$ is calculated along with the ash, A, water, W, empty body weight, EBW, and live weight, LW, all in kg's as follows

$$L_{pig} = P_{pig} LP_{min}$$
$$A = 0.189 P_{pig}$$
$$W = w P_{pig}^{0.855}$$
$$EBW = P_{pig} + A + W + L_{pig}$$
$$LW = \frac{EBW}{0.95}$$

The growth of the pig in a day is calculated as follows.

The first stage, as shown in Figure A-1, is the calculation of the feed intake. Firstly digestible energy intake in MJ, DE_{in}, is calculated as

$$DE_{in} = \left(55.07\left(1 - e^{-0.0176 LW}\right)\right) \min\left\{p, p_{diet}\right\}$$

where p is the feed intake capacity, determined by the genotype of the pig, and p_{diet} is the feed intake restriction of the farmer.

Digestibility energy intake is then used to determine the feed intake in kg's, F_{in}, as

$$F_{in} = \frac{DE_{in}}{DE_{diet}}$$

where DE_{diet} is the digestible energy in the diet in MJ per kg.

The second stage consists of determining the protein intake in g's, P_{in}, amino acid intake over the j amino acids in g's, $AA_{in,j}$, and energy protein free intake in MJ per kg, EPF_{in}, as grouped in Figure A-1, where

$$P_{in} = F_{in} DP_{diet}$$
$$AA_{in,j} = F_{in} AA_{diet,j} \quad for\ all\ j$$
$$EPF_{in} = DE_{in} - E_p P_{in}$$

and DP_{diet} is digestible protein in the diet in g's per kg, $AA_{diet,j}$ is the amino acid in the diet for each amino acid j in g's per kg, and E_p is energy content of protein (0.0236 MJ/g).

Thirdly, as shown on the left of Figure A-1, the potential protein deposition in g's, Pd_{pot}, is addressed. The protein requirements for maintenance in g's, P_{main}, and the amino acid requirements for maintenance over the j amino acids in g's, $AA_{main,j}$, are calculated. Also the protein available for growth in g's, P_{growth}, and the amino acid available for growth over the j amino acids in g's, $AA_{growth,j}$, are determined. The balanced protein available for growth in g's, BP_{growth}, is also calculated, where

$$P_{main} = 0.9375 LW^{0.75}$$
$$AA_{main,j} = P_{main} AA_{bal,j} \quad for\ all\ j$$
$$P_{growth} = 0.85\left(P_{in} - P_{main}\right)$$
$$AA_{growth,j} = 0.85\left(AA_{in,j} - AA_{main,j}\right) \quad for\ all\ j$$
$$BP_{growth} = \min\left\{P_{growth}, \frac{AA_{growth,j}}{AA_{bal,j}}\right\} \quad for\ all\ j$$

and $AA_{bal,j}$ is the ideal amino acid balance for each amino acid j.

The potential protein deposition in g's, Pd_{pot}, can then be determined as

$$Pd_{pot} = \min\left\{BP_{growth}, Pd_{\max}\right\}$$

The fourth main section is shown on the right of Figure A-1, and determines the potential lipid deposition in g's, Ld_{pot}. Energy requirements for maintenance in MJ, E_{main}, energy available from excess protein catabolism in MJ, E_{cat}, and energy available for growth in MJ, E_{growth}, are first calculated, where

$$E_{main} = E_m LW^{0.75}$$
$$E_{cat} = E_{pc}\left(P_{in} - P_{main} - Pd_{pot}\right)$$
$$E_{growth} = EPF_{in} + E_{cat} + E_p Pd_{pot} - E_{main}$$

and E_{pc} is the available energy derived from amino acid catabolism (0.0115 MJ/g).

The potential lipid deposition in g's, Ld_{pot}, may then be calculated as

$$Ld_{pot} = \frac{E_{growth} - E_{pd} Pd_{pot}}{E_{pl}}$$

where E_{pd} is the energy content of body protein deposition (0.0439 MJ/g), and E_{pl} is the energy content of body lipid deposition (0.0528 MJ/g).

The fifth section, of energy partitioning, as in Figure A-1, starts with the potential protein weight in kg's, P_{pot}, and the potential lipid weight in kg's, L_{pot}, calculations. This allows for the calculation of the potential lipid to protein ratio, $LP_{min,pot}$, where

$$P_{pot} = P_{pig} + \frac{Pd_{pot}}{1000}$$
$$L_{pot} = L_{pig} + \frac{Ld_{pot}}{1000}$$
$$LP_{min,pot} = \frac{L_{pot}}{P_{pot}}$$

and P_{pig}, and L_{pig}, are the protein and lipid in the pig in kg's at the start of the day's growth.

Adjustments to potential protein deposition in g's, Pd_{red}, and potential lipid deposition in g's, Ld_{inc}, are calculated if the potential lipid to protein ratio is less than the lipid to protein ratio of the pig, LP_{min}, as determined by the genetics of the pig, so that

$$\begin{cases} Pd_{red} = \dfrac{1000\left(LP_{min}P_{pot} - L_{pot}\right)}{\frac{E_{pd}-E_p+E_{pc}}{E_{pl}} + LP_{min}} \\ Ld_{inc} = \frac{E_{pd}-E_p+E_{pc}}{E_{pl}} Pd_{red} \end{cases} \quad if\ LP_{min,pot} < LP_{min}$$

$$\begin{cases} Pd_{red} = 0 \\ Ld_{inc} = 0 \end{cases} \quad otherwise$$

Finally the protein deposition in g's, *Pd*, and lipid deposition in g's, *Ld*, can then be calculated as

$$Pd = Pd_{pot} - Pd_{red}$$
$$Ld = Ld_{pot} + Ld_{inc}$$

The growth and body composition in the final section in Figure A-1 determines the protein and lipid growth for the day which is to be added to the composition of the pig

$$P_{pig} \Leftarrow P_{pig} + 1000Pd$$
$$L_{pig} \Leftarrow L_{pig} + 1000Ld$$

The ash, *A*, water, *W*, empty body weight, *EBW*, and live weight, *LW*, at the end of the day is then calculated, where

$$A = 0.189P_{pig}$$
$$W = wP_{pig}^{0.855}$$
$$EBW = P_{pig} + A + W + L_{pig}$$
$$LW = \frac{EBW}{0.95}$$

where *w* is the chemical vs. physical body composition.

The evaluation parameters, back fat in mm, *BF*, average daily gain in kg's, *ADG*, and feed conversion ratio, *FCR*, are then determined as

$$TF_{in} \Leftarrow TF_{in} + F_{in}$$

$$BF = 0.8L_{pig}$$

$$ADG = \frac{LW - LW_{start}}{DTS}$$

$$ADF_{in} = \frac{TF_{in}}{DTS}$$

$$FCR = \frac{ADF_{in}}{ADG}$$

where TF_{in} is the total feed intake in kg's, ADF_{in} is the average daily feed intake in kg's, DTS is the number of cycles the model has run, and LW_{start} is the live weight of the pig before any growth cycles are processed in kg's.

Appendix B – Model performance on high diet

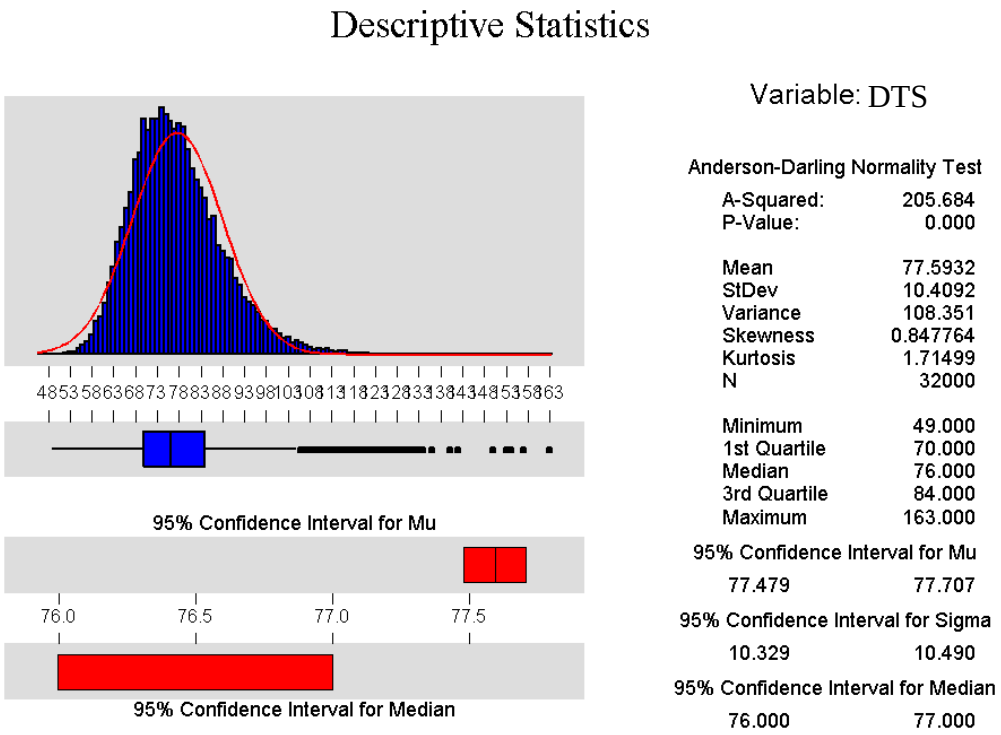

Figure B-1: DTS statistics – high diet

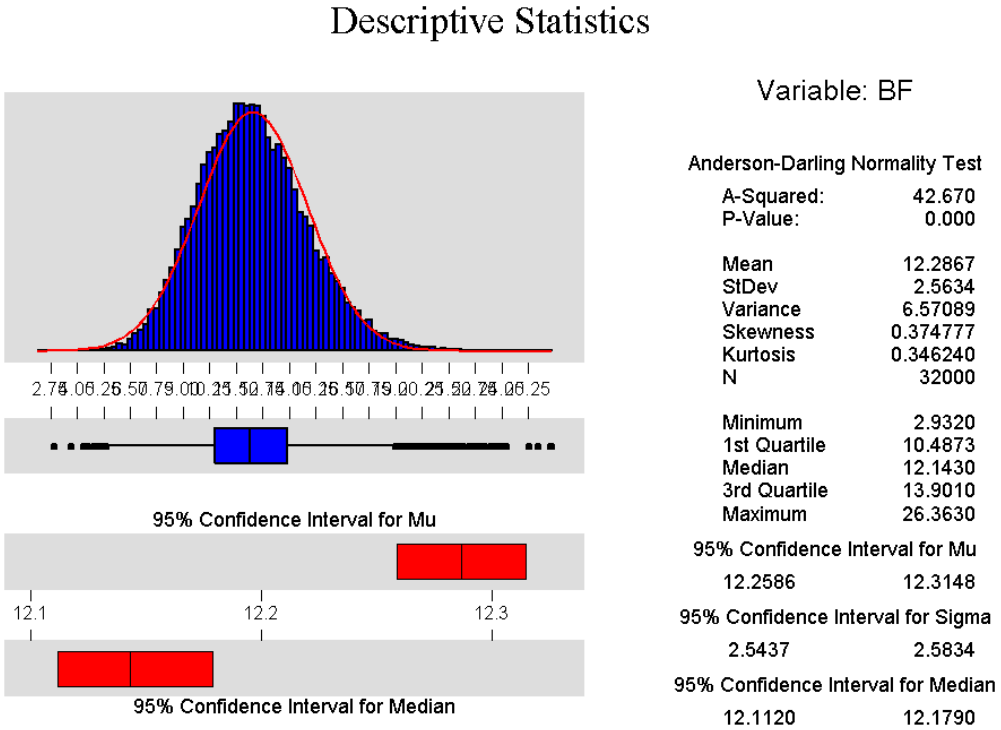

Figure B-2: BF statistics – high diet

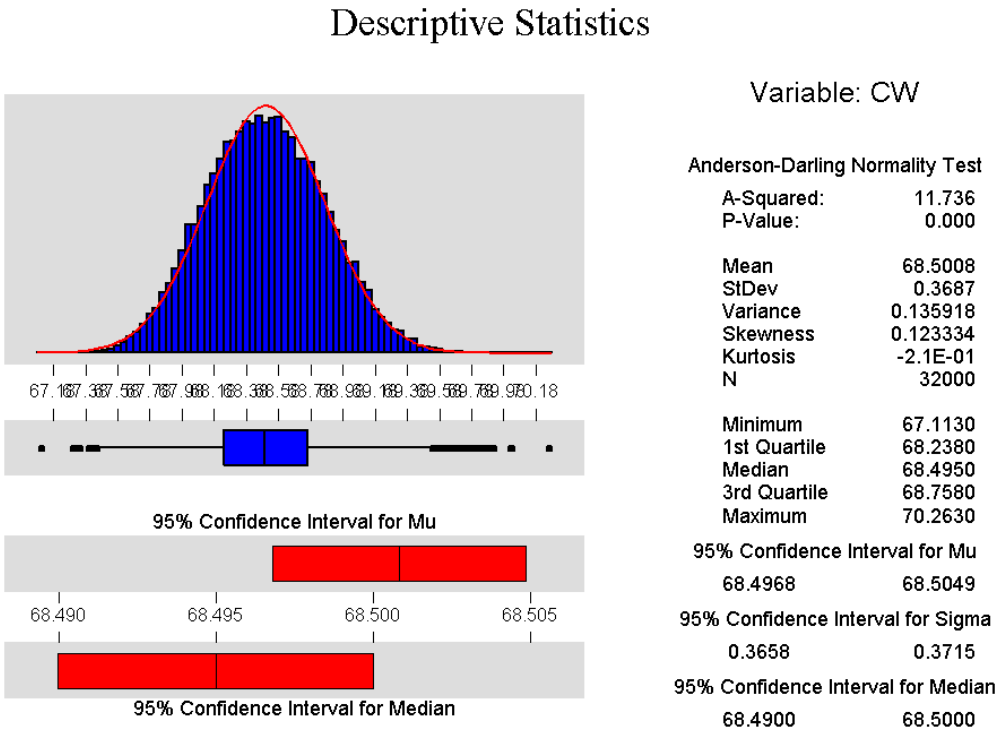

Figure B-3: CW statistics – high diet

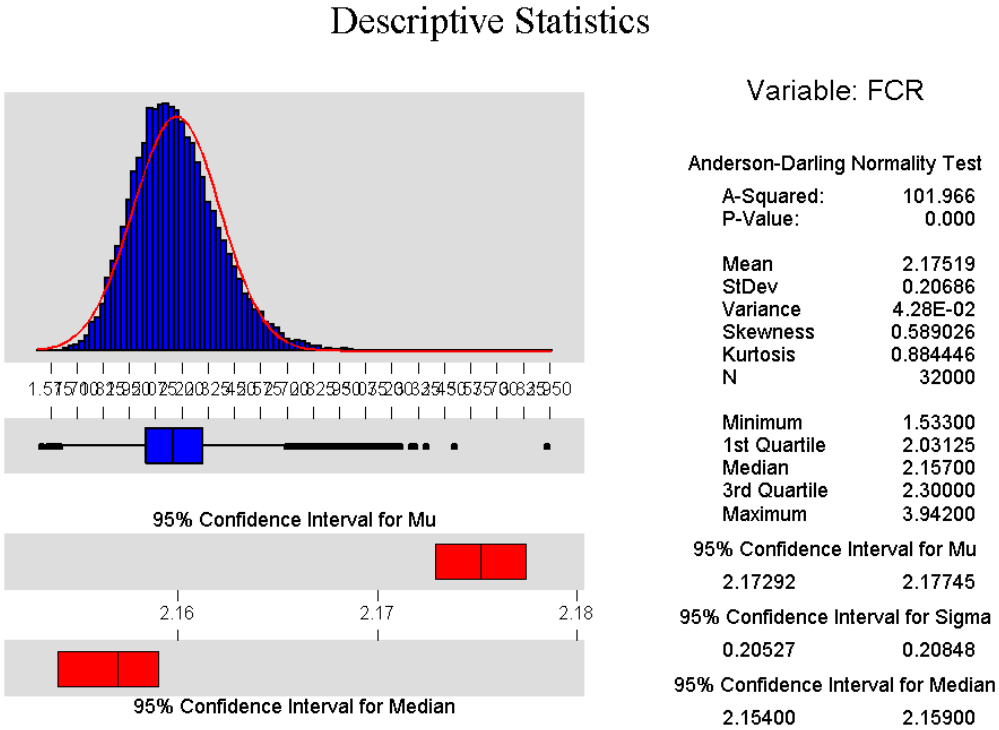

Figure B-4: FCR statistics – high diet

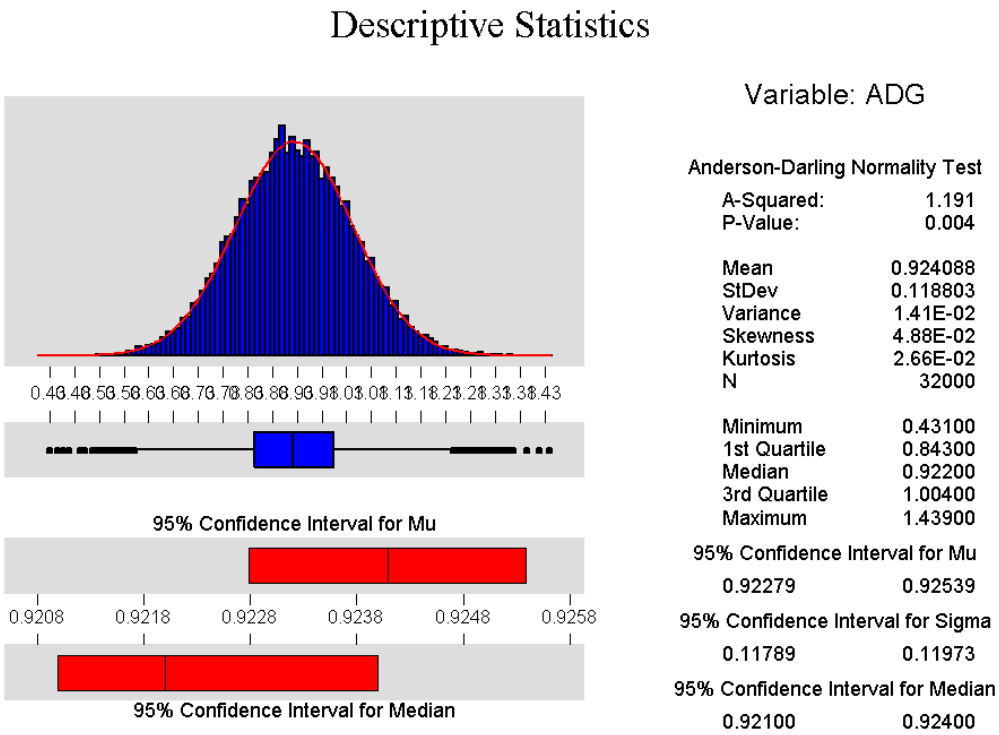

Figure B-5: ADG statistics – high diet

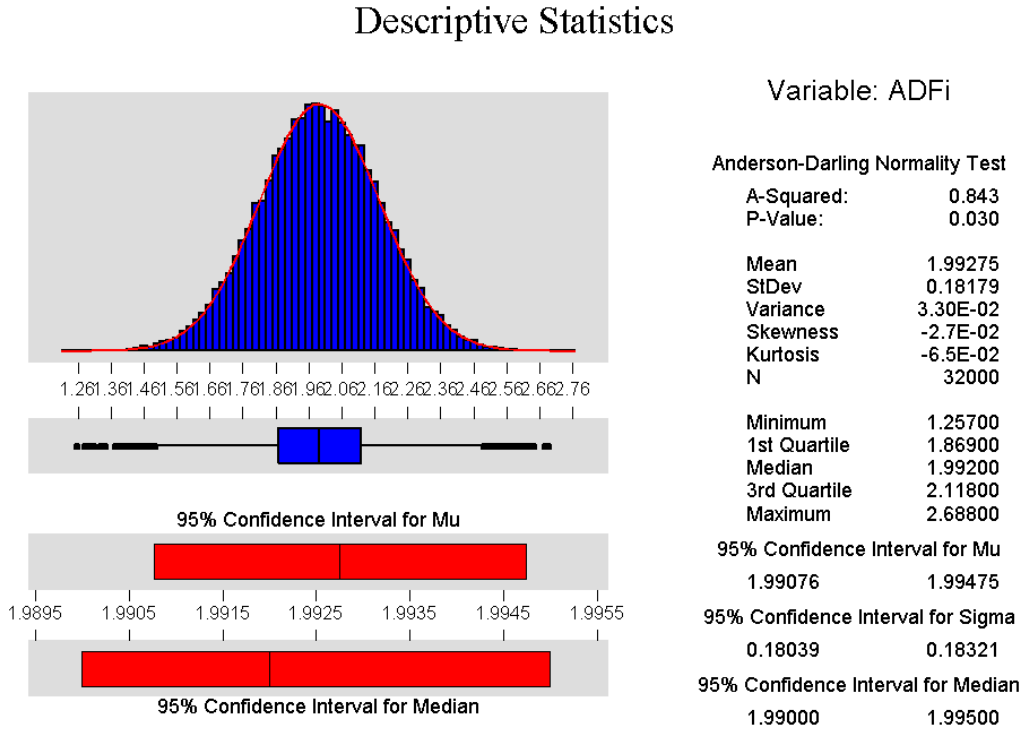

Figure B-6: ADFi statistics – high diet

Appendix C – Model performance on low diet

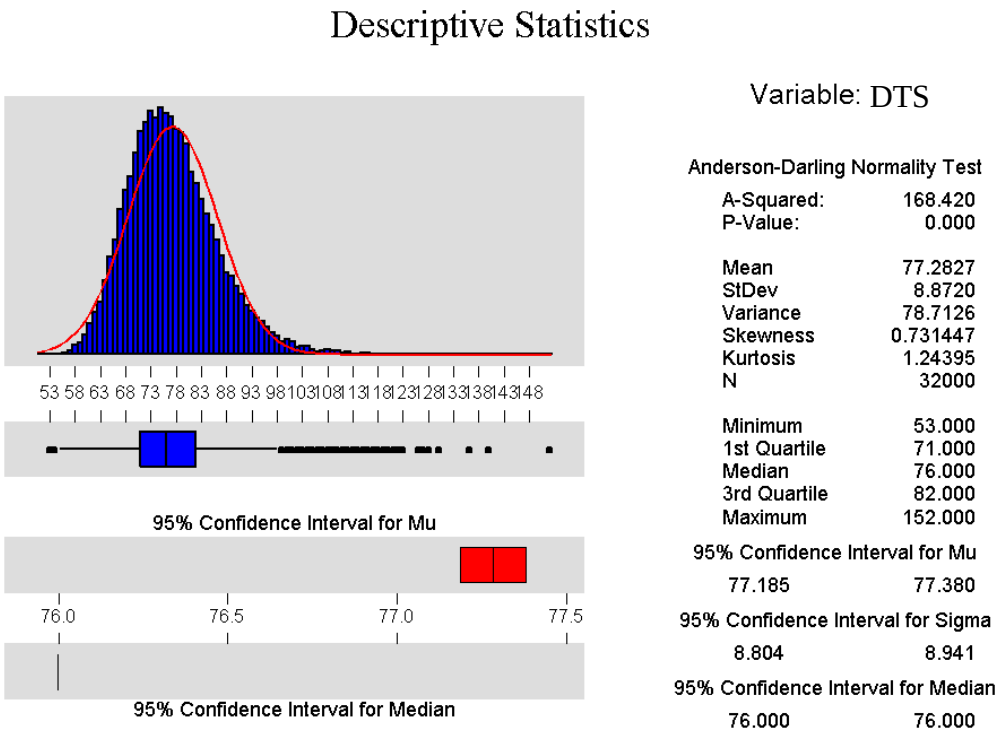

Figure C-1: DTS statistics – low diet

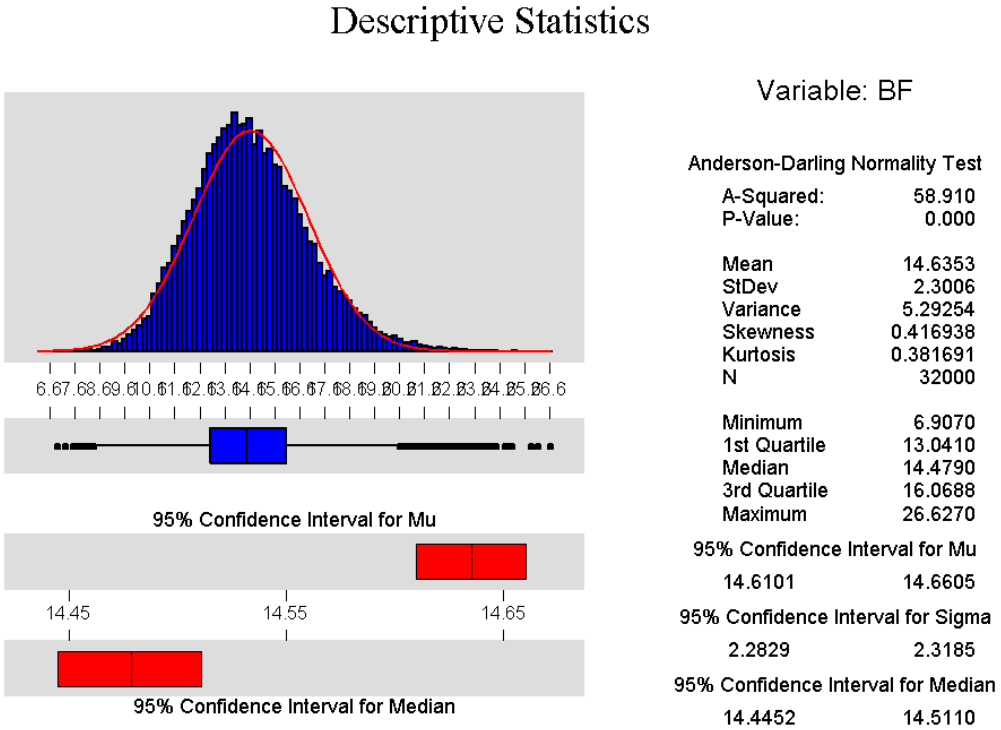

Figure C-2: BF statistics – low diet

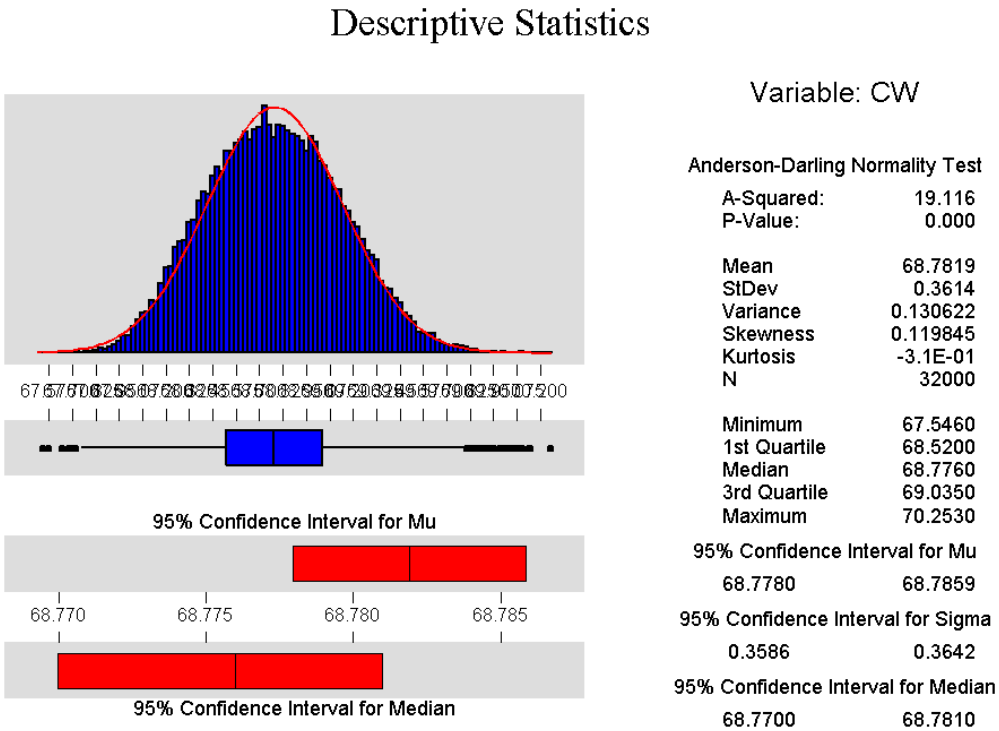

Figure C-3: CW statistics – low diet

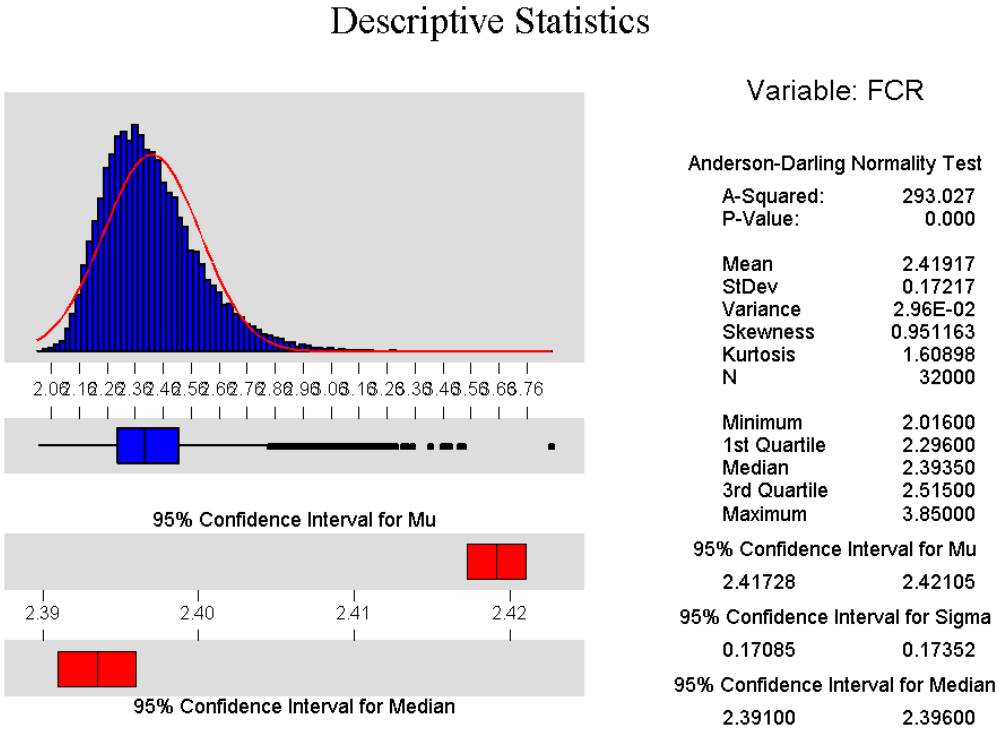

Figure C-4: FCR statistics – low diet

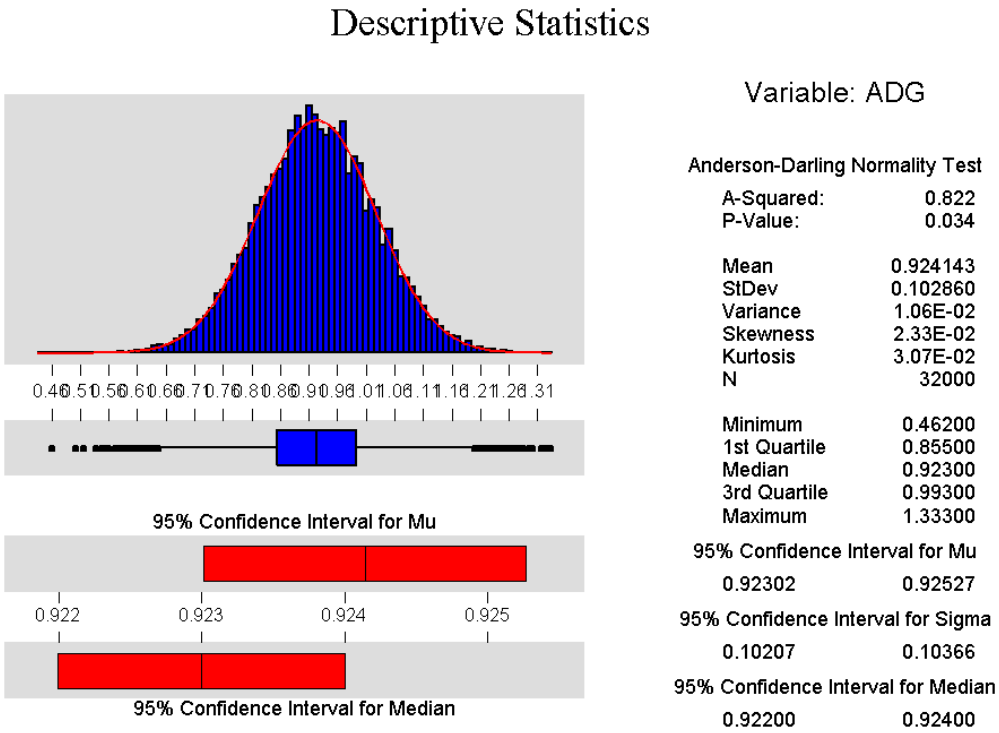

Figure C-5: ADG statistics – low diet

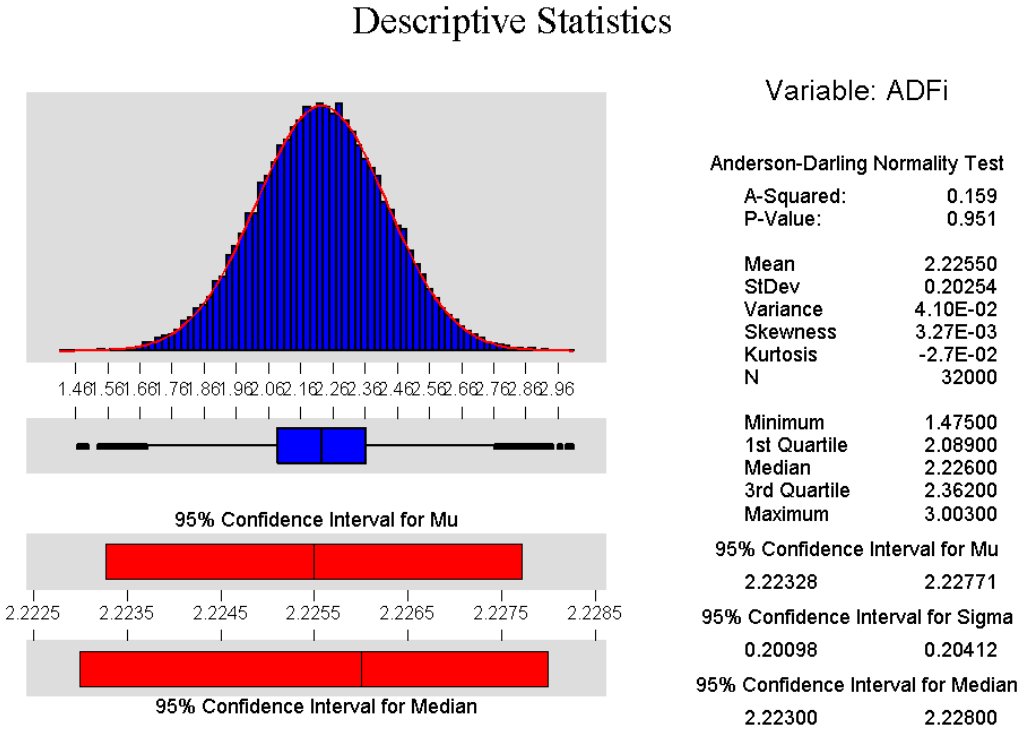

Figure C-6: ADFi statistics – low diet

Appendix D – Genetic Algorithm heuristic

Input:

N number of genomes to carry from one iteration to the next

mutation_prob = 0.1

crossover_prob = 0.25

Initialize:

Generate $N(N-1)/2+1$ genomes

Select *N* best => current_pop

Best_genome = best in current_pop

Repeat:

Breed all possible pairs of genomes in current_pop => offspring_pop

(There are $N(N-1)/2$ pairs)

Add Best_genome to offspring_pop

Select N best from offspring_pop => Current_pop

Best_genome = best in current_pop

Until stopping criteria met

Stopping criteria:

Run for p iterations

or Run until no improvement in best solution for p iterations

Output:

Best_genome

Breeding procedure:

Let n_1 be 1 or 2 randomly

Let n_2 be 1 or 2 randomly

For locus = 1 to max loci

allele 1 at offspring_locus locus = allele n_1 at parent1_locus locus

If (rand_prob < mutation_prob)

allele 1 at offspring_locus locus is increase or decreased by 1.

If (allele 1 at offspring_locus locus < 0)

allele 1 at offspring_locus locus = 0

If (allele 1 at offspring_locus locus > 5)

allele 1 at offspring_locus locus = 5

If (rand_prob < crossover_prob)

$n_1 = 2 – n_1 + 1$

allele 2 at offspring_locus locus = allele n_2 at parent2_locus locus

If (rand_prob < mutation_prob)

allele 2 at offspring_locus locus is increase or decreased by 1.

If (allele 2 at offspring_locus locus < 0)

allele 2 at offspring_locus locus = 0

If (allele 2 at offspring_locus locus > 5)

allele 2 at offspring_locus locus = 5

If (rand_prob < crossover_prob)

$n_2 = 2 – n_2 + 1$

Next locus

Appendix E – Tabu Search heuristic

Input:

E number of elite solutions

N number of neighbours

neighbour_prob = 0.1

Main procedure:

For each elite solution

Current_genome = random genome

Repeat:

Line_Search(Current_genome)

Until no improvement

Current_Elite_genome = Current_genome

Repeat:

Best_genome = best Elite

For each elite

Local_Search(Current_Elite_genome, diversify=True, neighbours=N)

Path_Search(Elite)

Until Best_genome ≥ best Elite

For each elite

Local_Search(Elite, diversify=False, neighbours=2N)

Best_genome = best Elite

Output:

Best_genome

Line Search:

```
    Line_best_genome = current_genome = input_genome

    generate N neighbours of current_genome => Neighbours
    select best Neighbour => best_neighbour

    direction = best_neighbour_genes – current_genes

    Repeat:
        new_genome = current_genome + direction
        If any allele is out of range
            bring it back into range

        If new_genome > Line_best_genome
            Line_best_genome = new_genome

        direction = new_genome – current_genome
        current_genome = new_genome

    Until direction = 0

Return Line_best_genome


Local_Search:
    local_best_genome = current_genome = input_genome

    If diversify = True
        For each locus in current_genome
            If locus value < 1/3 max value
                Make increasing locus value Tabu for 6 iterations

            If locus value > 2/3 max value
                Make decreasing locus value Tabu for 6 iterations
```

Repeat

generate N neighbours of current_genome => Neighbours

select best Neighbour => best_neighbour

If best_neighbour > local_best_genome

local_best_genome = best_neighbour

direction = current_genome – best_neighbour

current_genome = best_neighbour

Decrease all Tabu tenures by 1

For each locus in direction

If locus value > 0

Make increasing locus value Tabu for 3 iterations

If locus value < 0

Make decreasing locus value Tabu for 3 iterations

Until stopping criteria meet

Return local_best_genome

Path_Search:

Generate elite pairs so that each elite is a source once and a destination once and no destination elite of the current destination elite is the source of the current destination elite (no circuits of length 2, or if 1 goes to 2, 2 cannot go to 1)

For each elite pair

destination_genome = destination_elite

current_genome = source_elite

Repeat

Make Tabu any move that will increase distance between current_genome and destination_genome

generate N neighbours of current_genome => Neighbours
select best Neighbour => best_neighbour

If best_neighbour > current_genome
current_genome = best_neighbour

Until stopping criteria meet when applied to current_genome

current_new_elite = current_genome

Return new_elite_genomes

Neighbour Generation:

For locus = 1 to max loci
allele 1 at neighbour_locus locus = allele 1 at current_locus locus

If (rand_prob < neighbour_prob)
allele 1 at neighbour_locus locus is increase or decreased by 1.
If direction chosen is Tabu
make no change
If allele 1 at neighbour_locus locus < 0
allele 1 at neighbour_locus locus = 0
If (allele 1 at neighbour_locus locus > 5)
allele 1 at neighbour_locus locus = 5

allele 2 at neighbour_locus locus = allele 2 at current_locus locus

If (rand_prob < neighbour_prob)

allele 2 at neighbour_locus locus is increase or decreased by 1.

If direction chosen is Tabu

make no change

If allele 2 at neighbour_locus locus < 0

allele 2 at neighbour_locus locus = 0

If allele 2 at neighbour_locus locus > 5)

allele 2 at neighbour_locus locus = 5

Appendix F – Simulated Annealing heuristic

Input:

N	number of neighbours
max_cool_time	number of iterations spent at each temperature
stop_temp	temperature to stop at

neighbour_prob = 0.1

temperature = 1

cool_factor = 0.9

Stopping criteria:

temperature <= stop_temp

Output:

Best_genome

Initialize:

current_genome = random genome

Best_genome = current_genome

cool_time = max_cool_time

Repeat:

generate N neighbours of current_genome -> Neighbours

select best Neighbour -> best_neighbour

If (best_neighbour > current_genome)

current_genome = best_neighbour

Else If (random_prob < $e^{\frac{best\_neighbor - current}{temperature}}$)

current_genome = best_neighbour

If (current_genome > Best_genome)

```
            Best_genome = current_genome

        cool_time = cool_time – 1

        If (cool_time = 0)
            temperature = temperature * cool_factor
            cool_time = max_cool_time
Until stopping criteria met

Neighbour Generation:
    For locus = 1 to max loci
        allele 1 at neighbour_locus locus = allele 1 at current_locus locus

        If (rand_prob < neighbour_prob)
            allele 1 at neighbour_locus locus is increase or decreased by 1.
            If direction chosen is Tabu
                make no change
            If allele 1 at neighbour_locus locus < 0
                allele 1 at neighbour_locus locus = 0
            If (allele 1 at neighbour_locus locus > 5)
                allele 1 at neighbour_locus locus = 5

        allele 2 at neighbour_locus locus = allele 2 at current_locus locus

        If (rand_prob < neighbour_prob)
            allele 2 at neighbour_locus locus is increase or decreased by 1.
            If direction chosen is Tabu
                make no change
            If allele 2 at neighbour_locus locus < 0
                allele 2 at neighbour_locus locus = 0
            If allele 2 at neighbour_locus locus > 5)
                allele 2 at neighbour_locus locus = 5
```

Appendix G – Theoretical considerations on genetic selection

R.L. Sherriff, H.T. Blair, P.C.H. Morel, R. Sridharan , J. W. Tweedie and G.R. Wood*

College of Sciences, Massey University, Palmerston North, New Zealand. *Statistics Department, Macquarie University, NSW 2109, Australia.

Pigs are traditionally selected on phenotypic traits such as feed conversion ratio (FCR), back fat thickness (BF) or average daily gain (ADG) using classical population genetic methods. After several generations of breeding a plateau is reached and no further genetic gain is achieved. In this paper, we have used a simulation model to investigate what the effect of different environments is on the number of generations of breeding it takes to get to the plateau.

A simulation model linking individual gene actions, pig growth modelling and non-linear optimization has been developed. In this model, 39 loci each with six alleles (4.96×10^{60} genomic combinations) are controlling five traits, which are used to characterize the genotype in a pig growth model (De Lange 1995). The growth model then simulates FCR, BF and ADG for this given genotype and a given diet. The effect of dietary nutrient density (High: 14.5 MJDE/kg and 17.4 g available lysine/kg and Low: 12.8 and 7.1, respectively) during growth on theoretical genetic gain (ΔG with i-bar=1) was investigated. A search of the genome for the best value for FCR, BF or ADG was made using a genetic algorithm. To get estimates of the theoretical genetic gain, genomes for 50000 sire/dam pairs were generated at random as the base population, with each pair bred to produce one offspring. FCR, BF and ADG were simulated with the model for each diet and genetic parameters estimated. The theoretical number of generations (N), assuming no Bulmer effect (Bulmer 1971), between the base population and the best value genome was then calculated for each trait and diet combination (Table 1).

Table 1. Population mean, phenotypic standard deviation (sp), heritabilities (h^2), genetic gain (ΔG), best genome performance (BG), theoretical number of generations (N) for FCR, BF and ADG with High and Low diets (SE in bracket).

Trait	Diet	Mean	sp	h^2	ΔG	BG	N
FCR	High	2.18	0.207	0.389	0.0806	1.39 (0.008)	9.8
	Low	2.42	0.173	0.393	0.0680	1.89 (0.005)	7.8
BF	High	12.3	2.56	0.332	0.850	3.51 (0.151)	10.3
(mm)	Low	14.7	2.31	0.321	0.741	5.34 (0.161)	12.1
ADG	High	0.924	0.119	0.317	0.0379	1.62 (0.014)	18.4
(kg/d)	Low	0.924	0.103	0.299	0.0307	1.47 (0.009)	17.8

The nutrient density in the diet had only a small effect on heritability, but a more significant effect on ΔG values. Based on ΔG, it would take about 9.8, 10.3 and 18.4 generations for FCR, BF and ADG, respectively, for the base population to move from its starting performance to the best possible genome when the high diet is fed. But when the low diet is fed, the number of generations are 7.8, 12.1 and 17.8 for FCR, BF, and ADG, respectively. This provides an indication that diet can play an important role

in how far the traits are from their biological limit. This demonstrates that genetic improvement for a given trait is achieved through different pathways in different environments.

Supported by AGMARDT, New Zealand.

References

BULMER, M.G. (1971). American Naturalist. 105: 944, 201-211.

DE LANGE, C.F.M. (1995). In: "Modelling growth in the pig". pp 71-85. Eds. P.J Moughan, M.W.A. Verstegen and M.I. Visser-Reyneveld. (Wageningen Pers: Wageningen).

Appendix H – Additional results

In the following sections, the search algorithm optimal additive gene results for the high diets are graphed on the left with the equivalent low diet results on the right. The graph headings, e.g. Genes high(1) 6 iterations to stop, indicate that the additive genes have been graph, whether the graph is for high or low diet results, and the stopping criteria of the algorithm.

H-1 Genetic Algorithm – Average daily gain

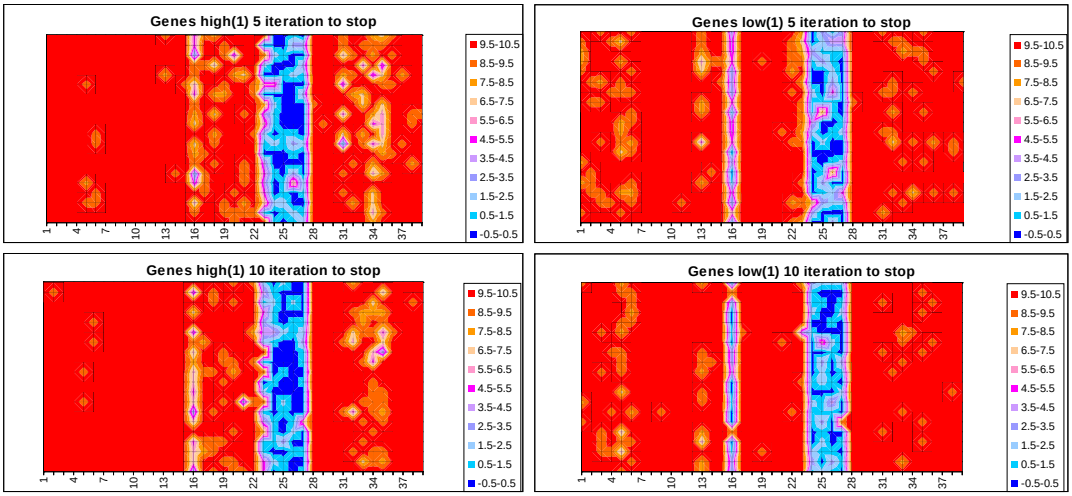

Figure H-1: Genetic Algorithm 10 Genomes

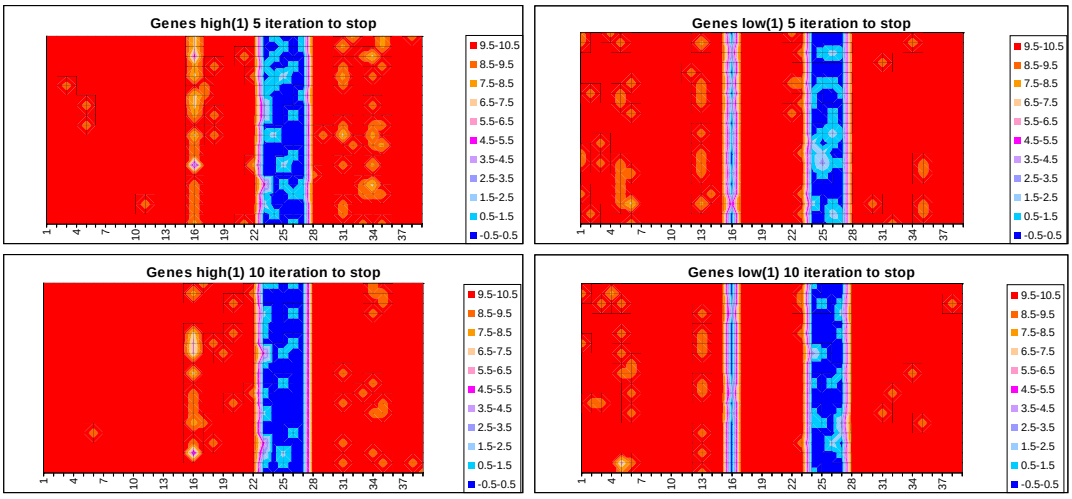

Figure H-2: Genetic Algorithm 30 Genomes

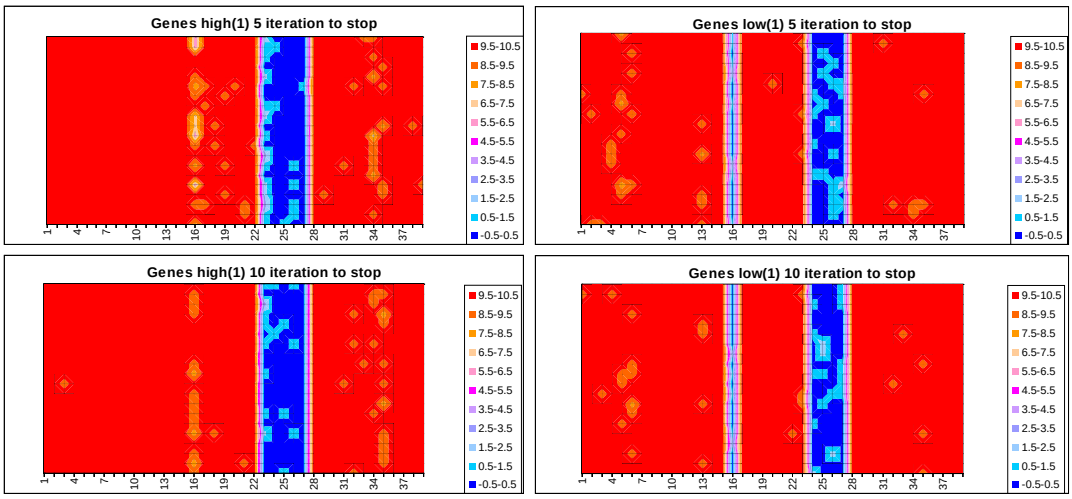

Figure H-3: Genetic Algorithm 50 Genomes

H-2 Tabu Search – Average daily gain

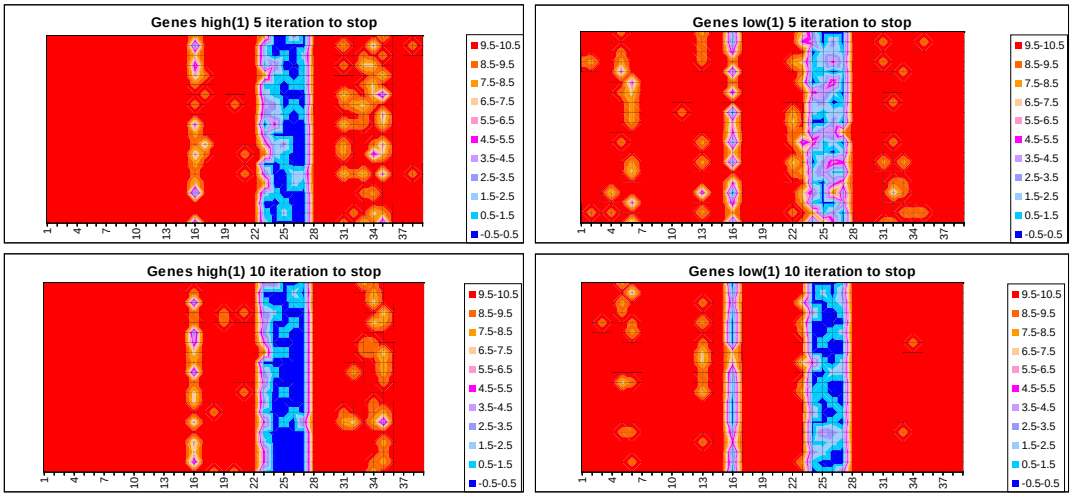

Figure H-4: Tabu Search 10 Neighbours 3 Elite

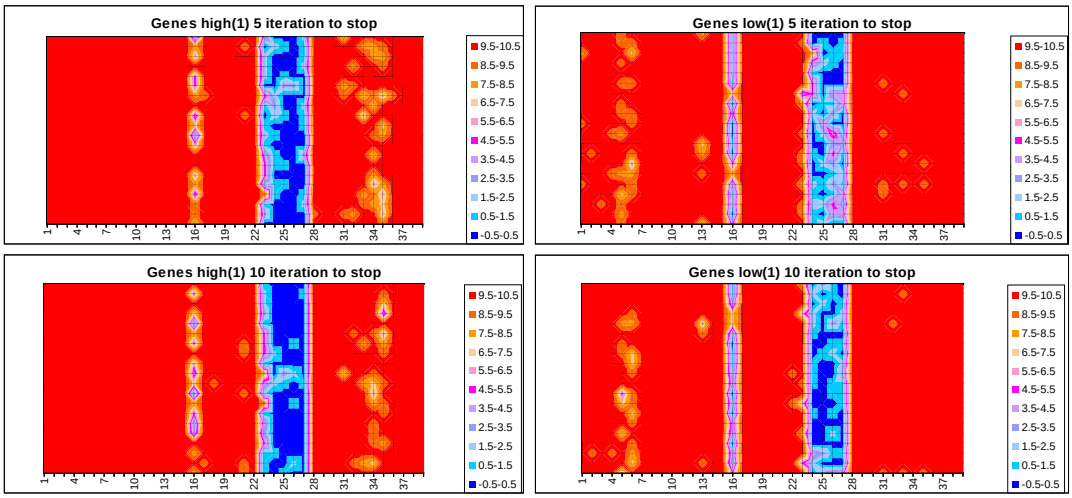

Figure H-5: Tabu Search 10 Neighbours 5 Elite

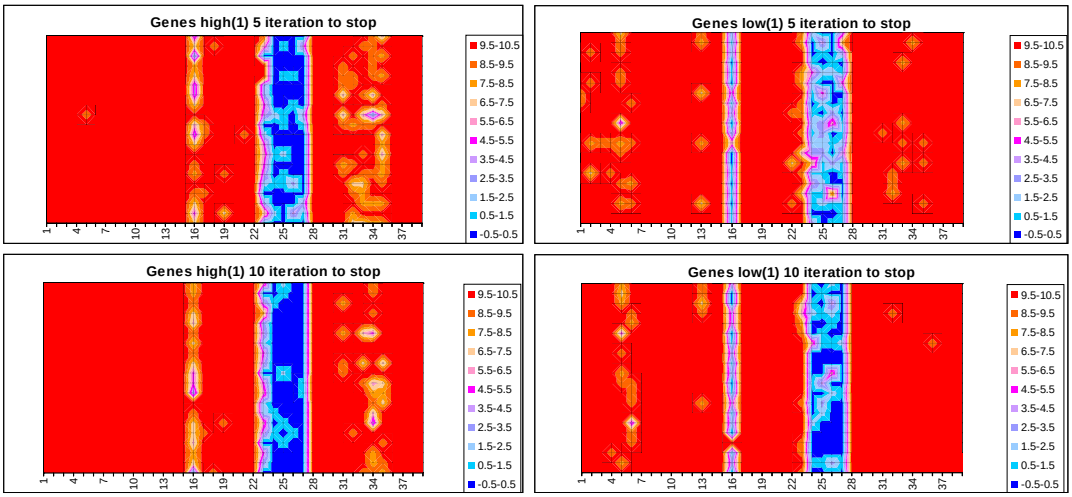

Figure H-6: Tabu Search 30 neighbours 3 elite

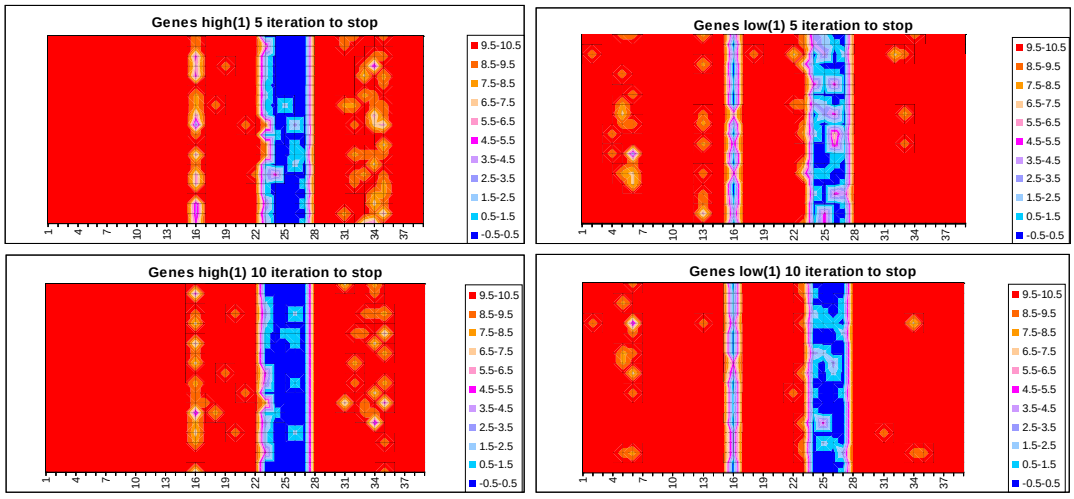

Figure H-7: Tabu Search 30 Neighbours 5 elite

H-3 Simulated Annealing – Average daily gain

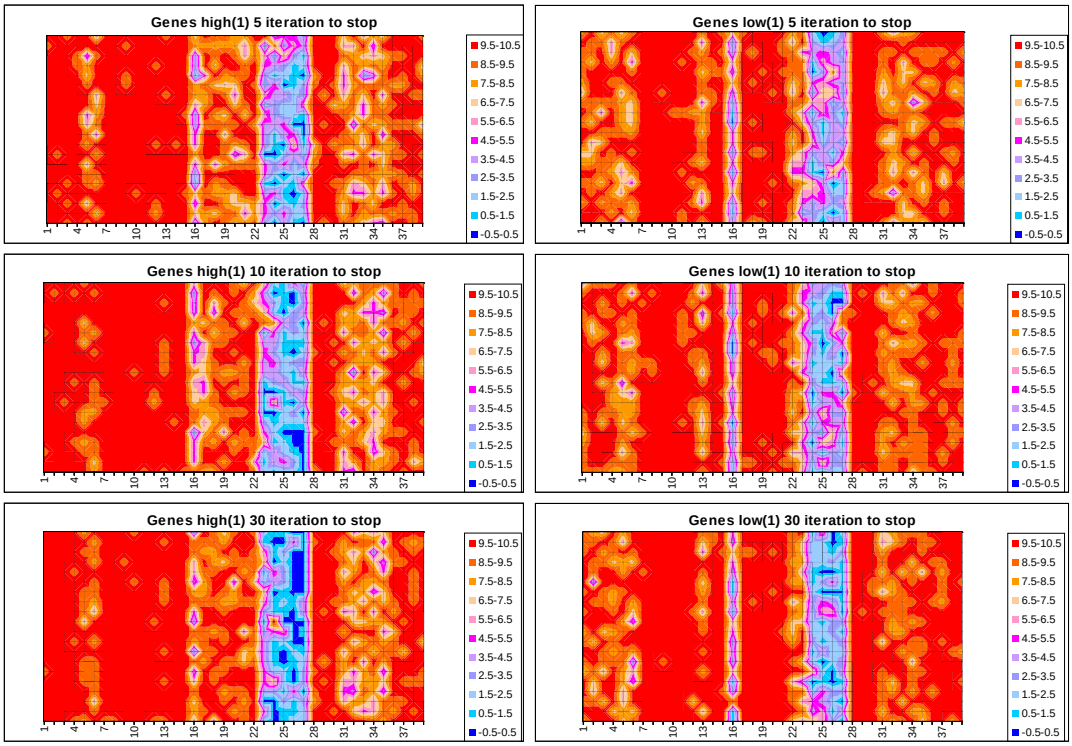

Figure H-8: Simulated Annealing 10 Neighbours 10 Stop temp

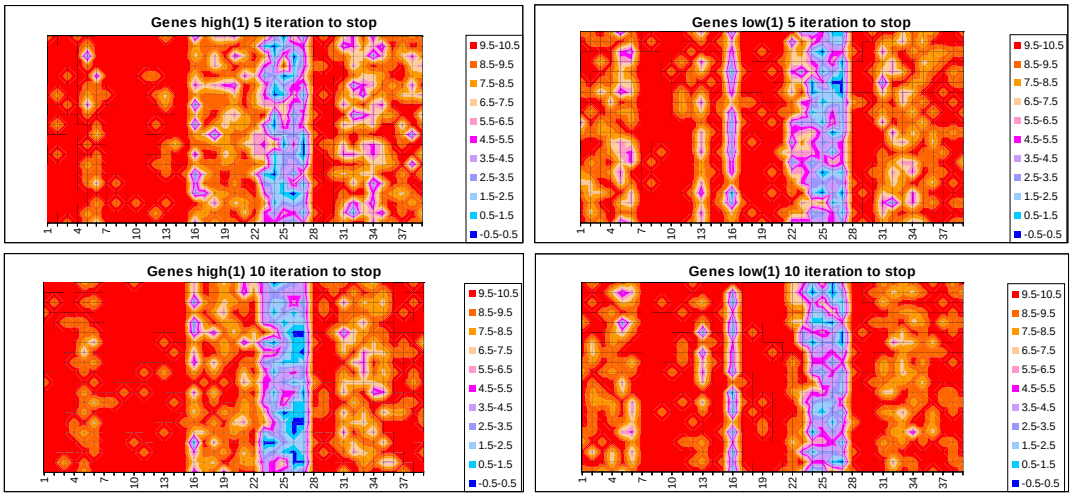

Figure H-9: Simulated Annealing 10 neighbours 50 stop temp

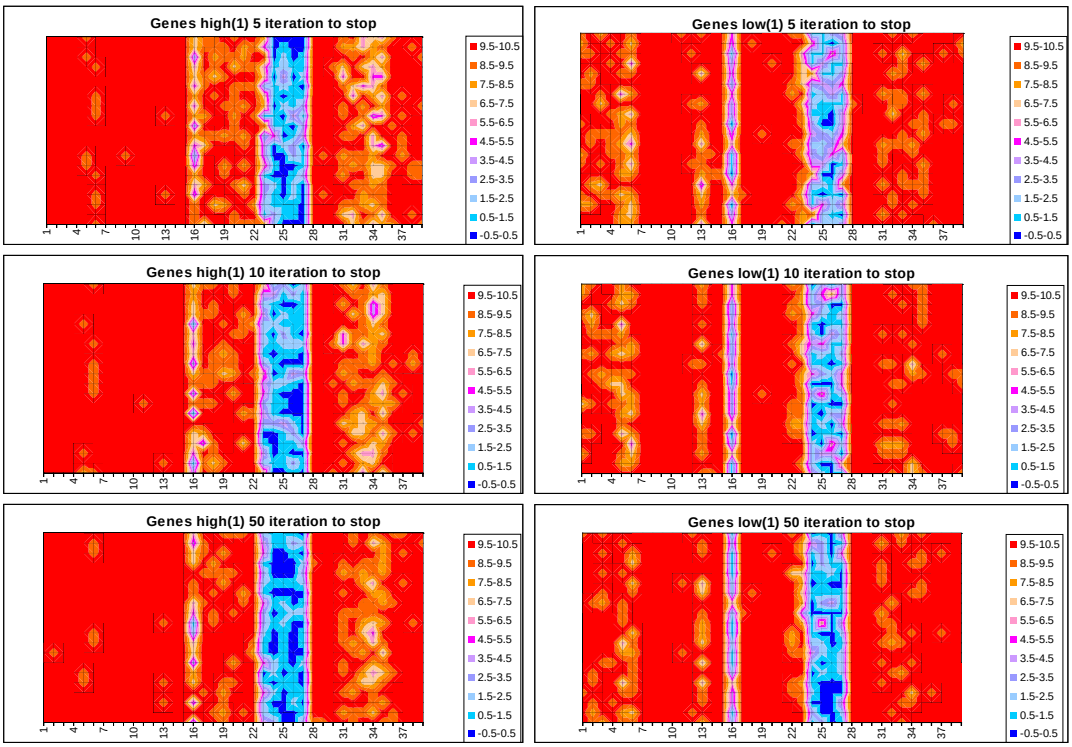

Figure H-10: Simulated Annealing 30 neighbours 10 stop temp

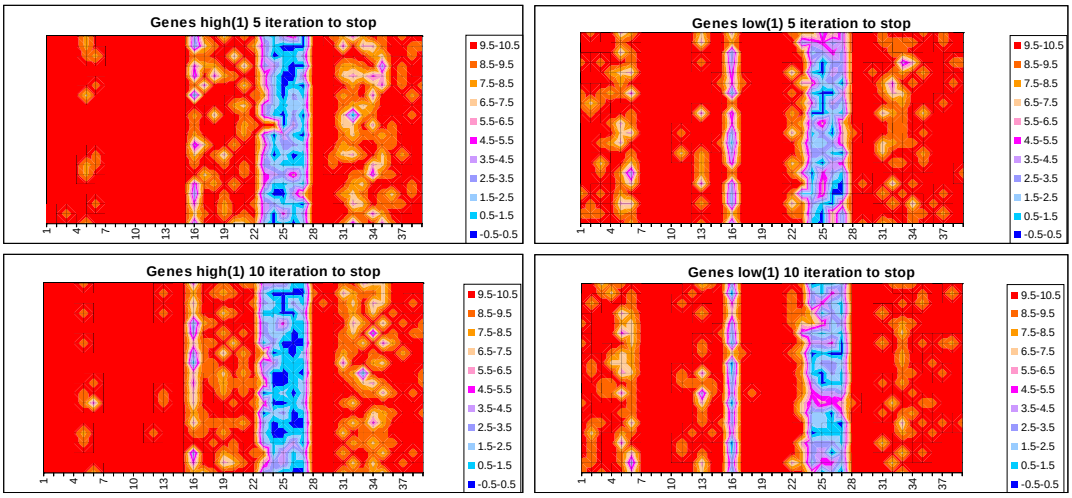

Figure H-11: Simulated Annealing 30 neighbours 50 stop

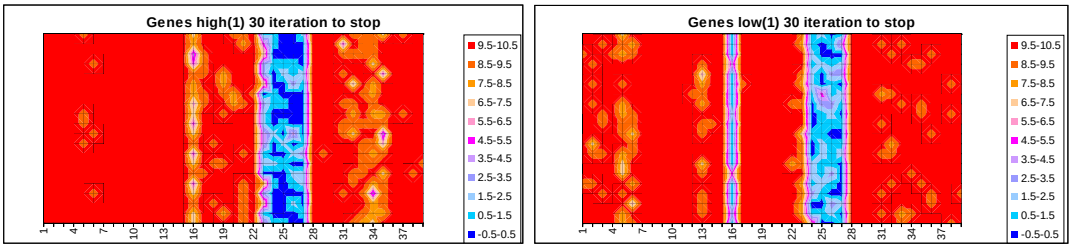

Figure H-12: Simulated Annealing 50 neighbours 10 stop

H-4 Genetic Algorithm – Feed conversion ratio

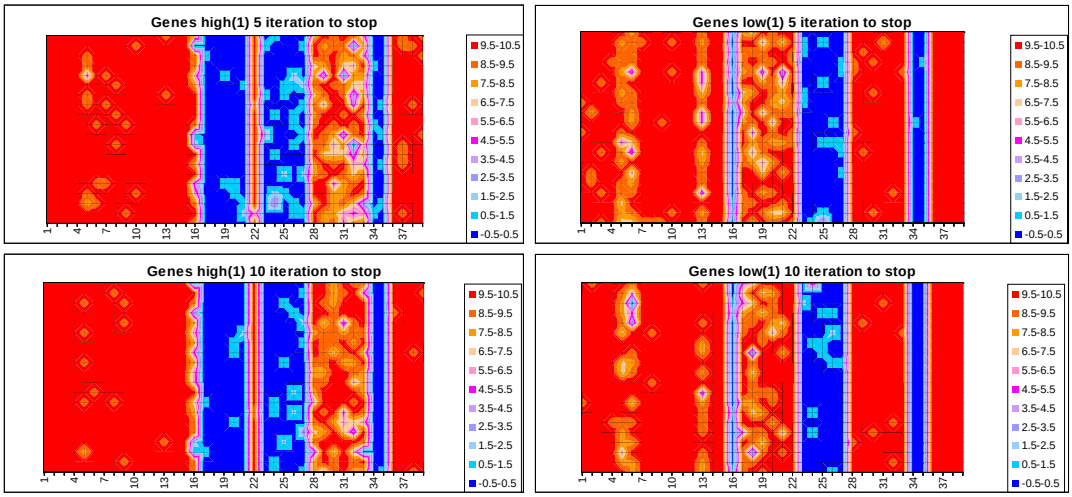

Figure H-13: Genetic Algorithm 10 Genomes

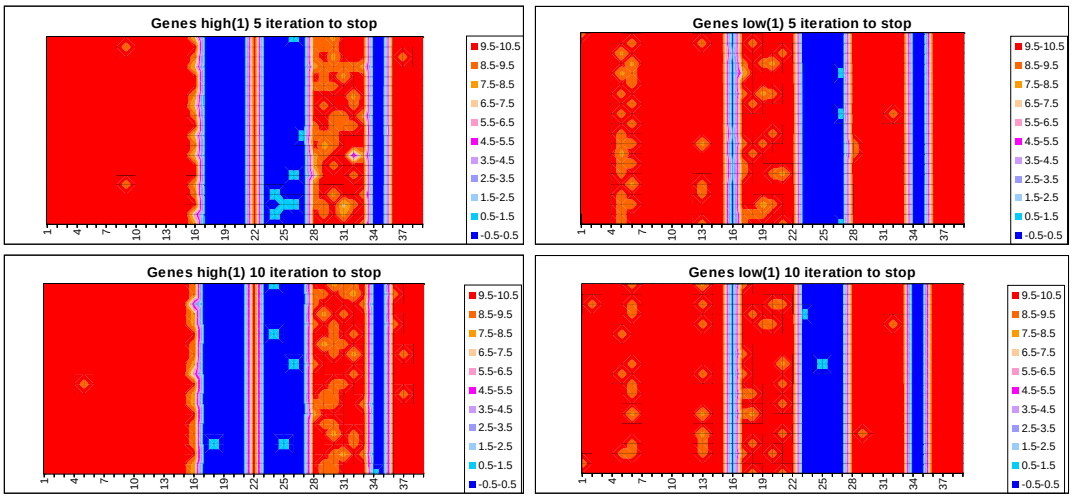

Figure H-14: Genetic Algorithm 30 Genomes

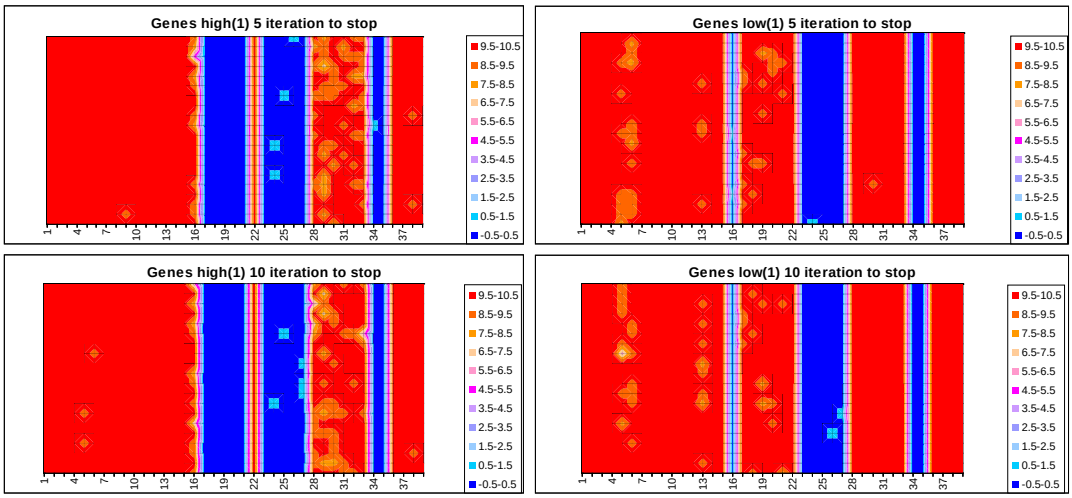

Figure H-15: Genetic Algorithm 50 Genomes

H-5 Tabu Search – Feed conversion ratio

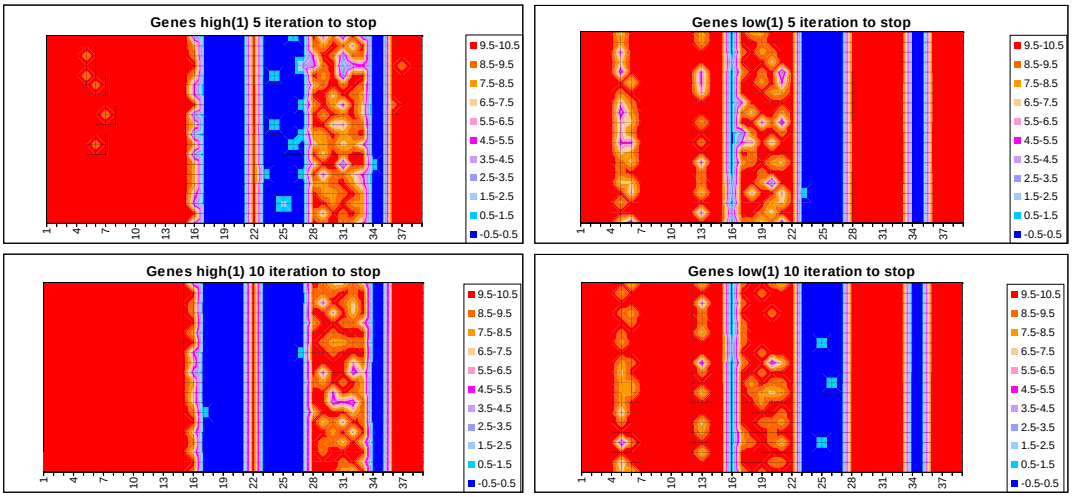

Figure H-16: Tabu Search 10 neighbours 3 elite

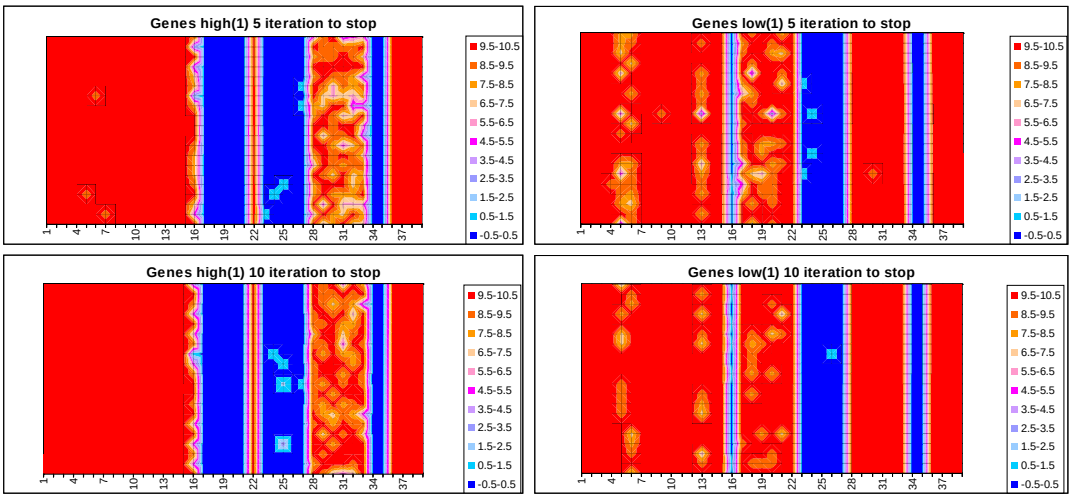

Figure H-17: Tabu Search 10 neighbour 5 elite

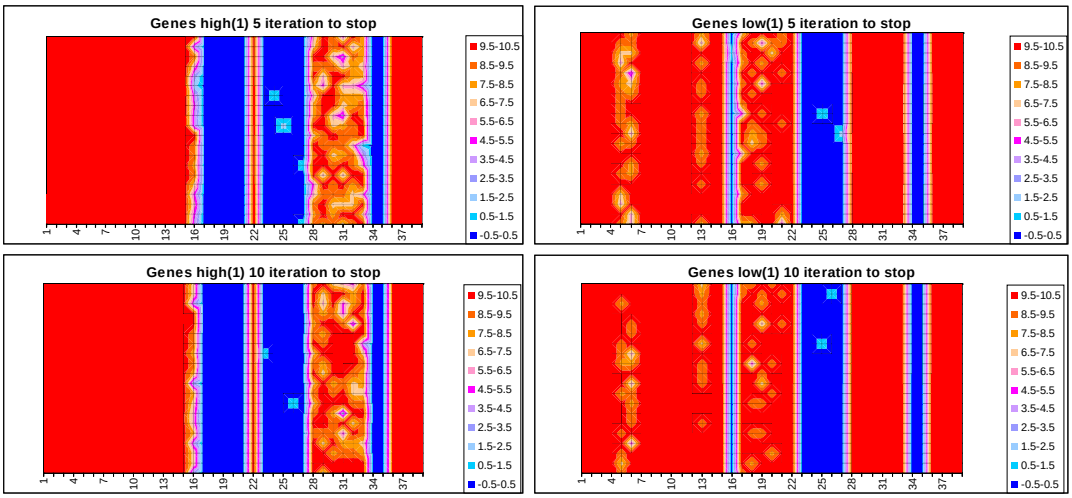

Figure H-18: Tabu Search 30 neighbours 3 elite

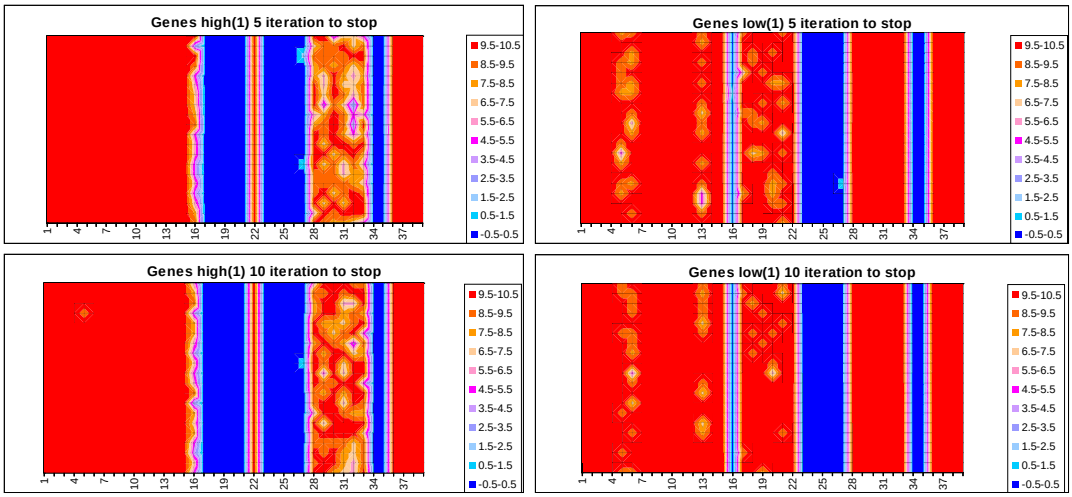

Figure H-19: Tabu Search 30 Neighbours 5 elite

H-6 Simulated Annealing– Feed conversion ratio

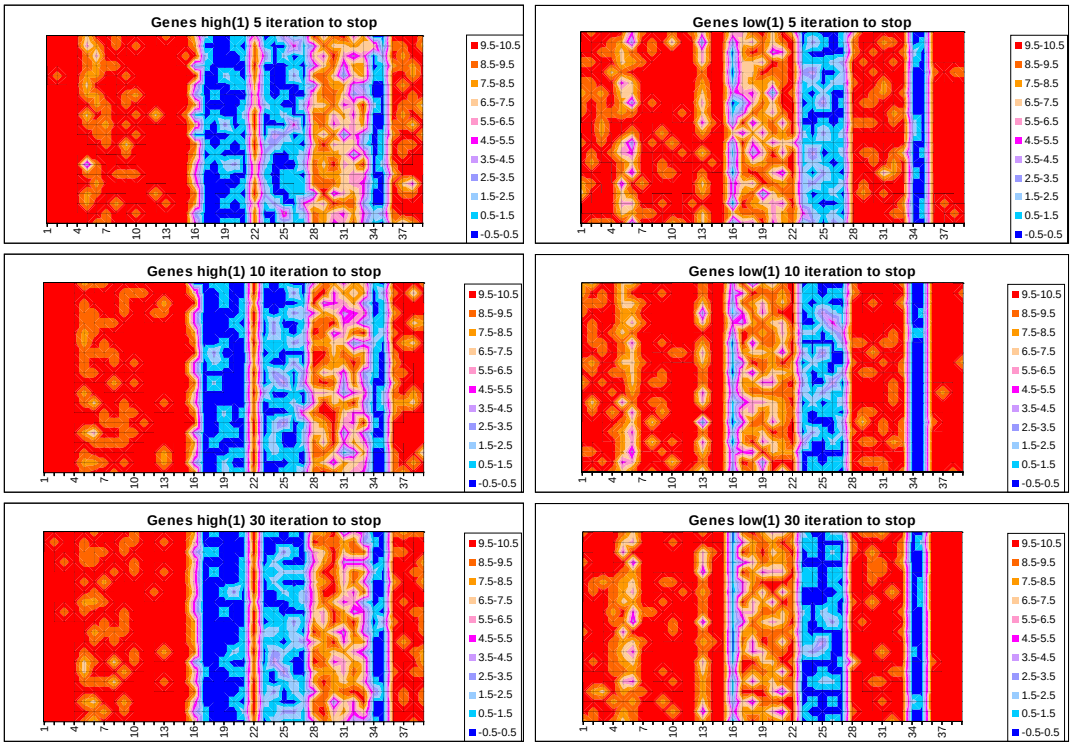

Figure H-20: Simulated Annealing 10 Neighbours 10 stop temp

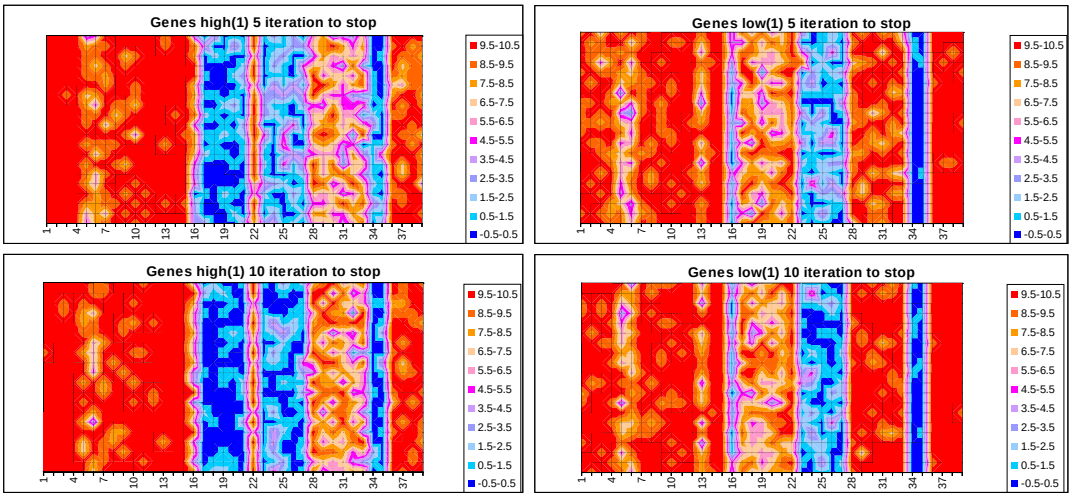

Figure H-21: Simulated Annealing 10 neighbours 50 stop temp

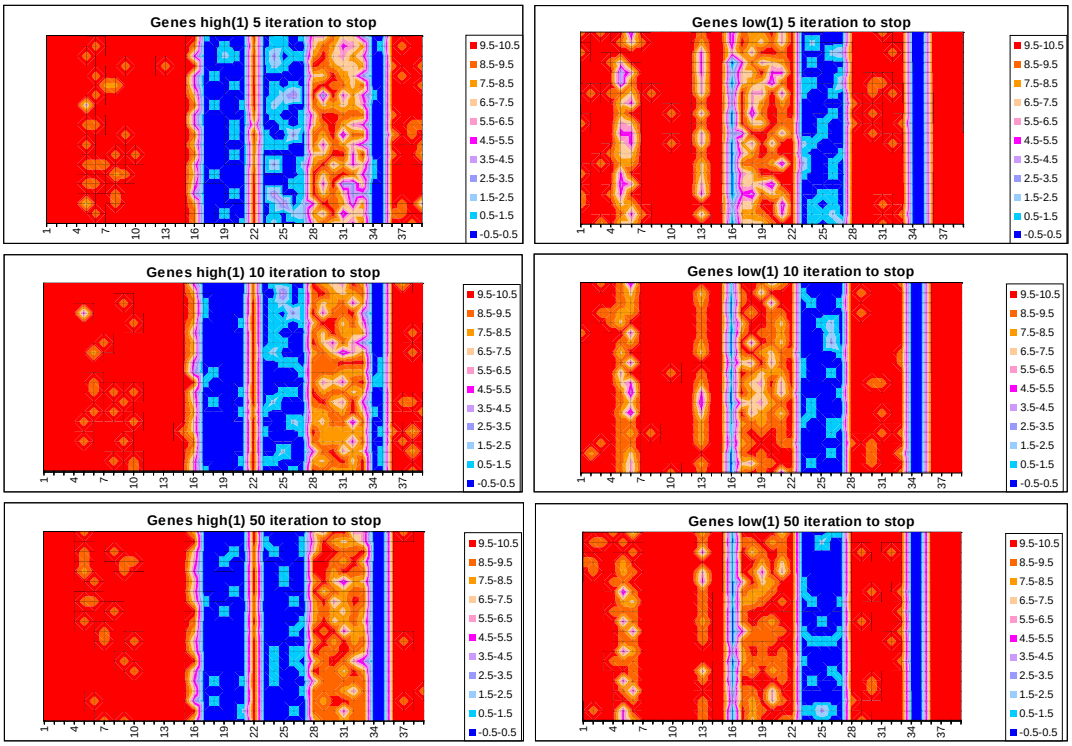

Figure H-22: Simulated Annealing 30 neighbours 10 stop temp

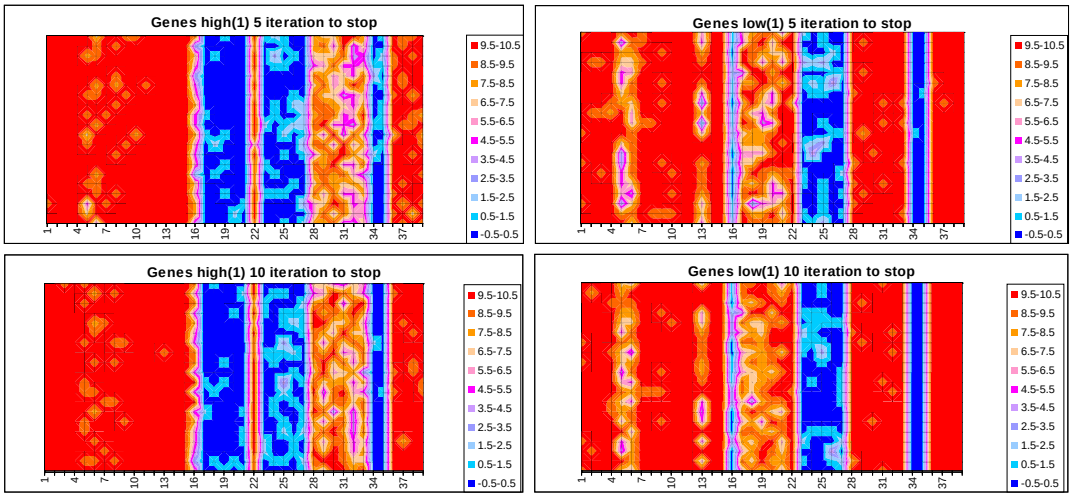

Figure H-23: Simulated Annealing 30 neighbours 50 stop

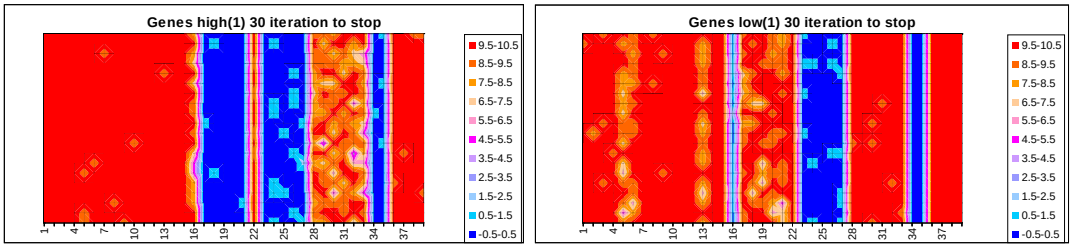

Figure H-24: Simulated Annealing 50 neighbours 10 stop temp

H-7 Genetic Algorithm – Back fat

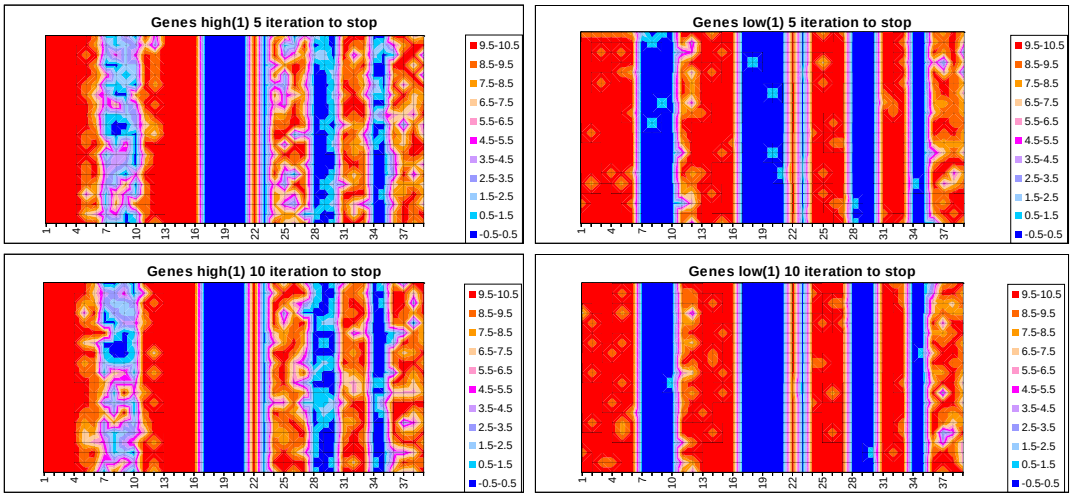

Figure H-25: Genetic Algorithm 10 Genomes

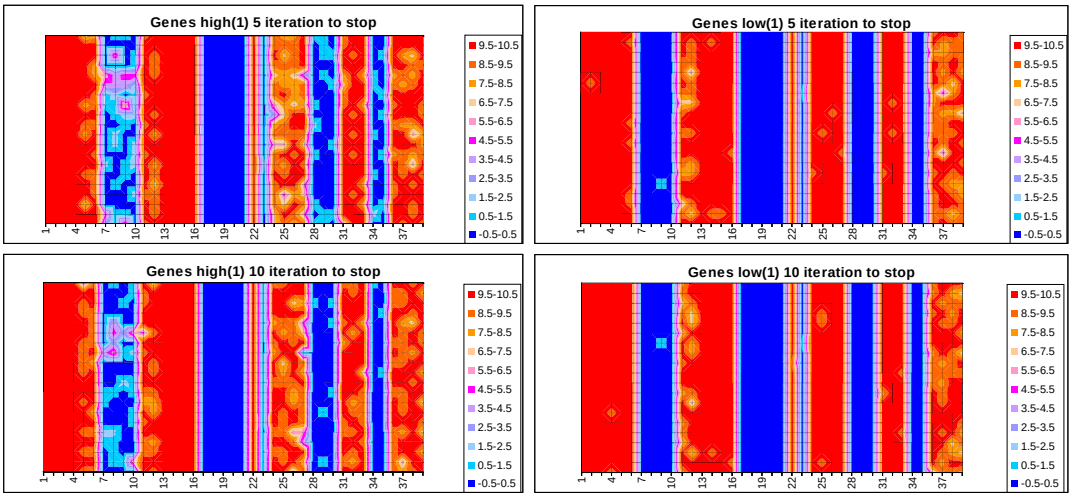

Figure H-26: Genetic Algorithm 30 Genomes

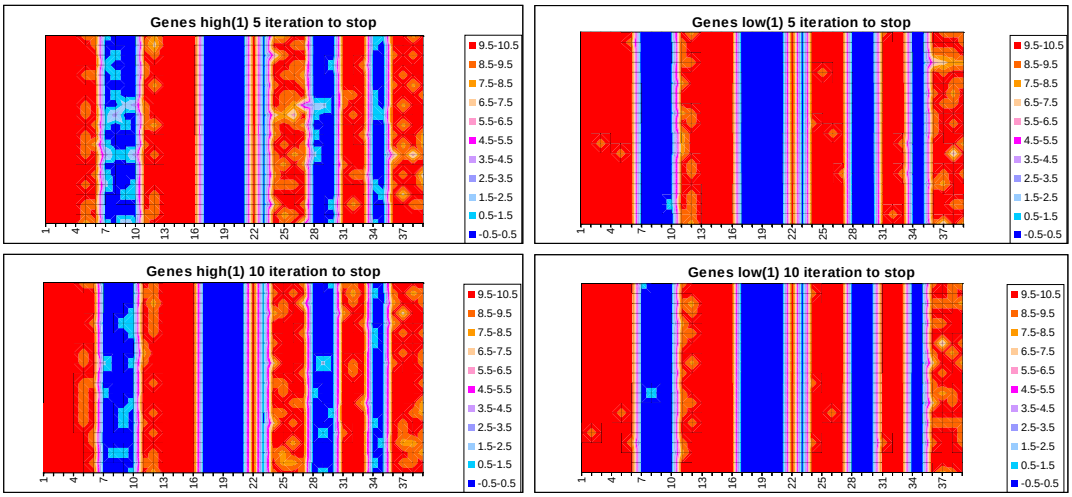

Figure H-27: Genetic Algorithm 50 Genomes

H-8 Tabu Search – Back fat

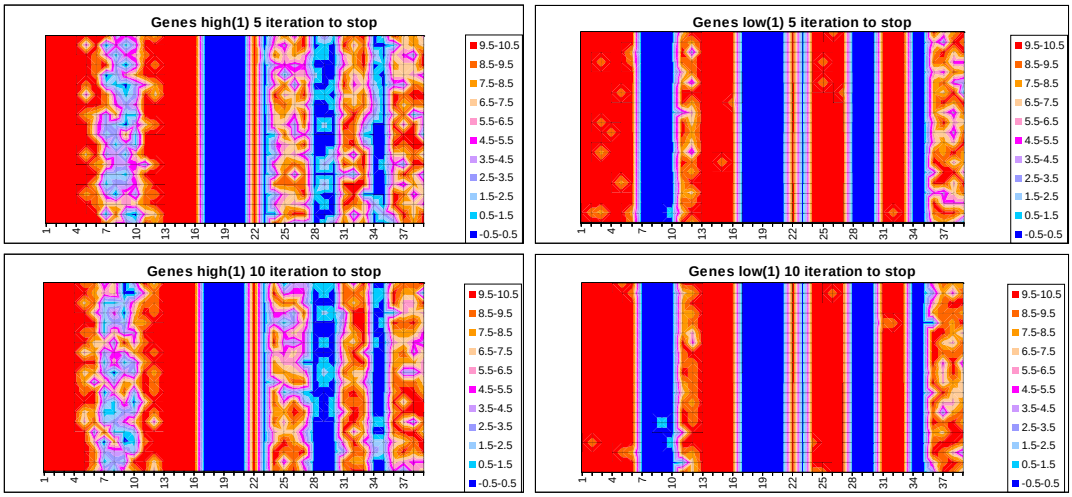

Figure H-28: Tabu Search 10 Neighbours 3 Elite

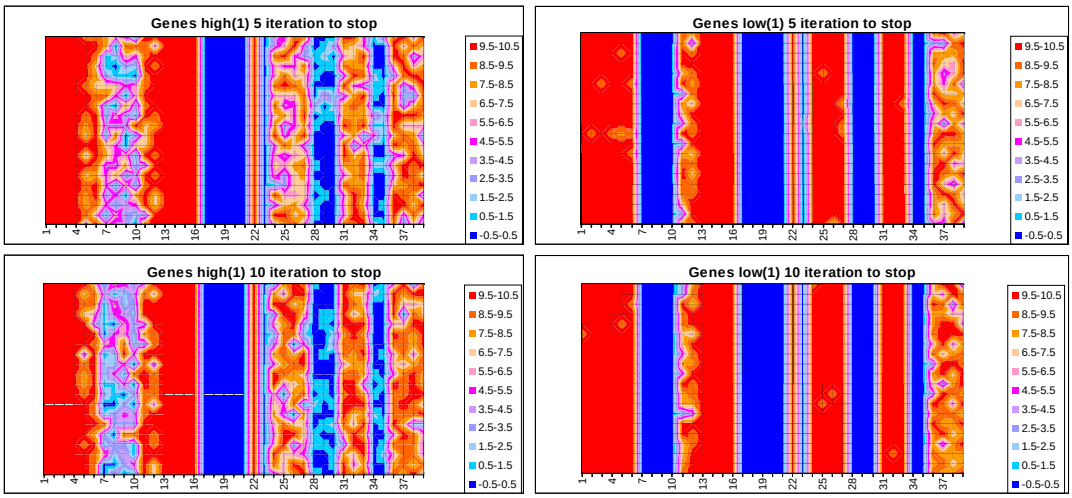

Figure H-29: Tabu Search 10 Neighbours 5 Elite

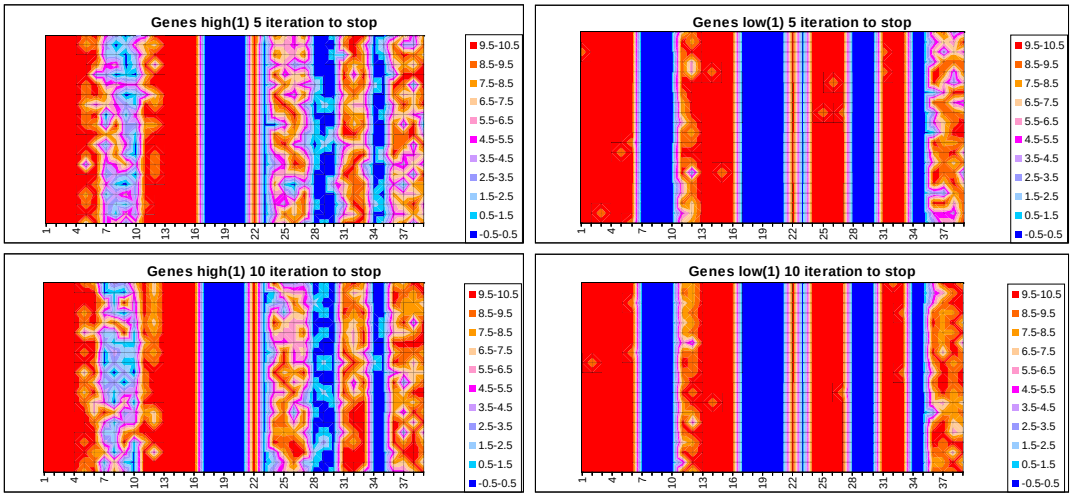

Figure H-30: Tabu Search 30 Neighbours 3 Elite

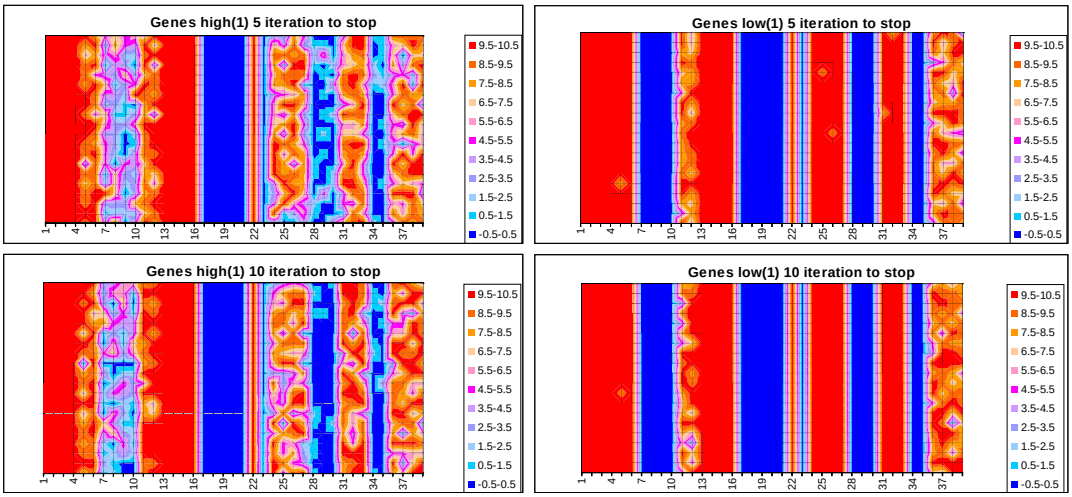

Figure H-31: Tabu Search 30 Neighbours 5 Elite

H-9 Simulated Annealing – Back fat

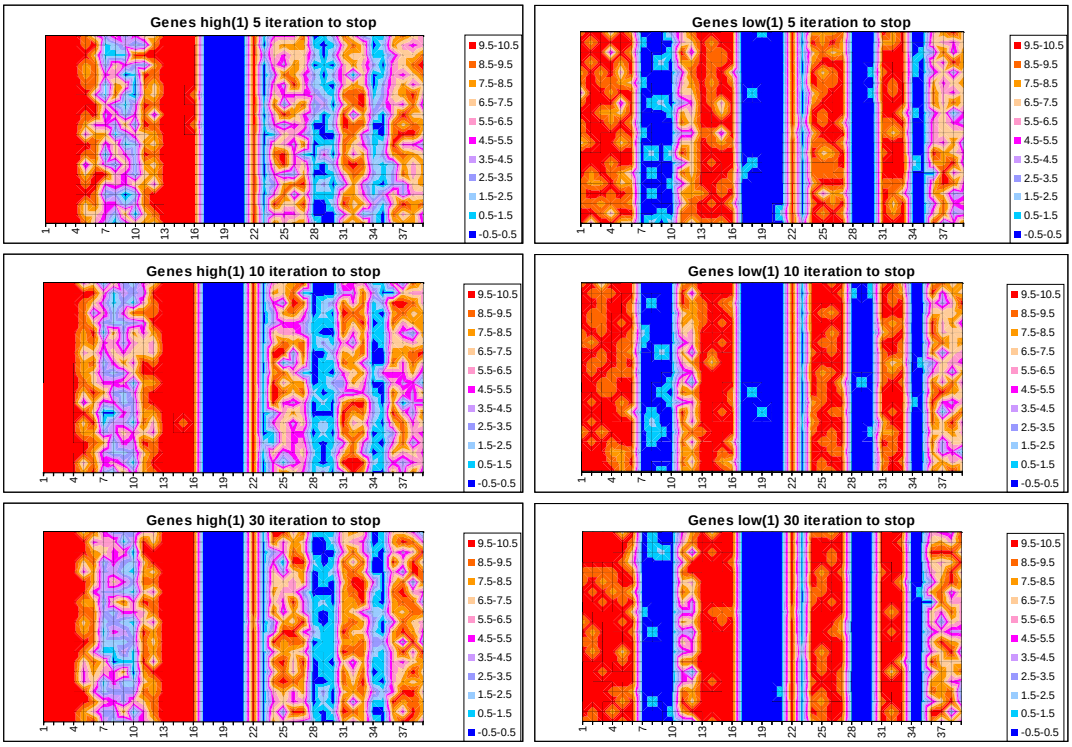

Figure H-32: Simulated Annealing 10 Neighbours 10 Stop temp

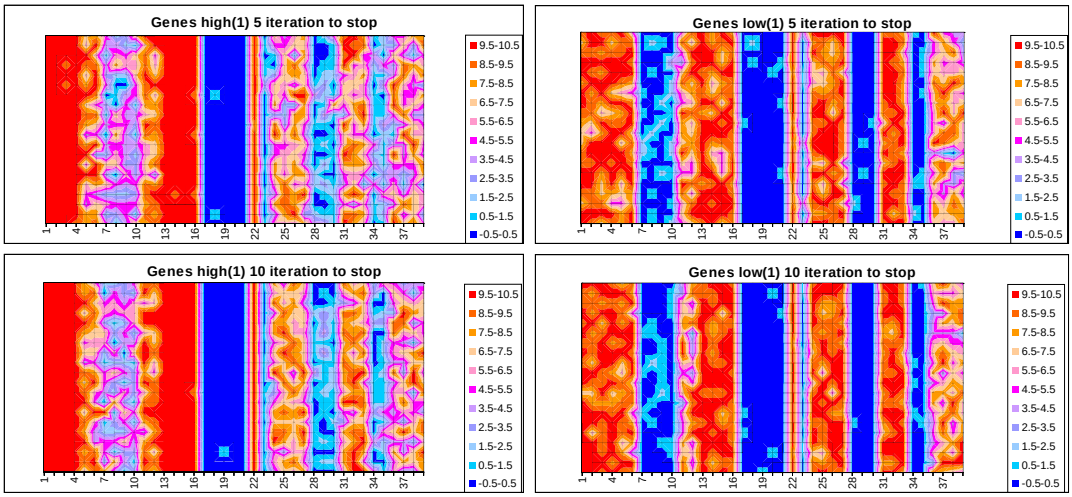

Figure H-33: Simulated Annealing 10 Neighbours 50 Stop temp

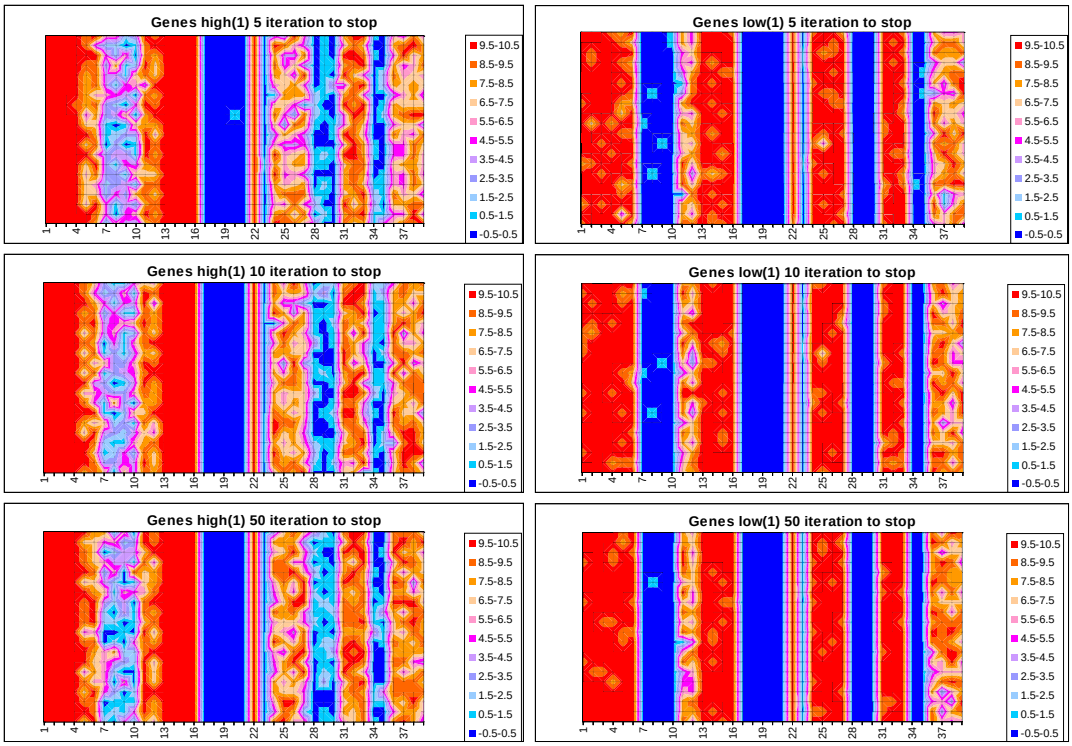

Figure H-34: Simulated Annealing 30 Neighbours 10 Stop Temp

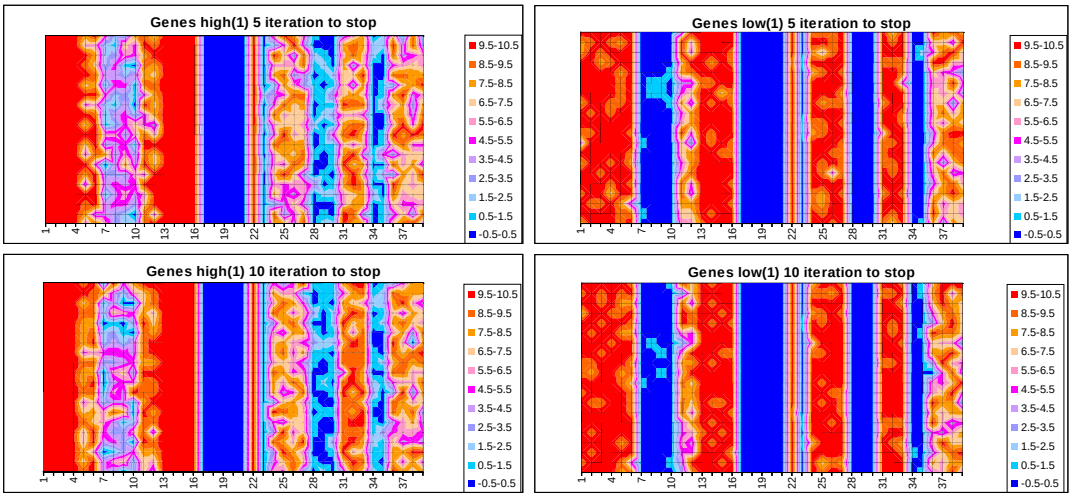

Figure H-35: Simulated Annealing 30 Neighbours 50 Stop Temp

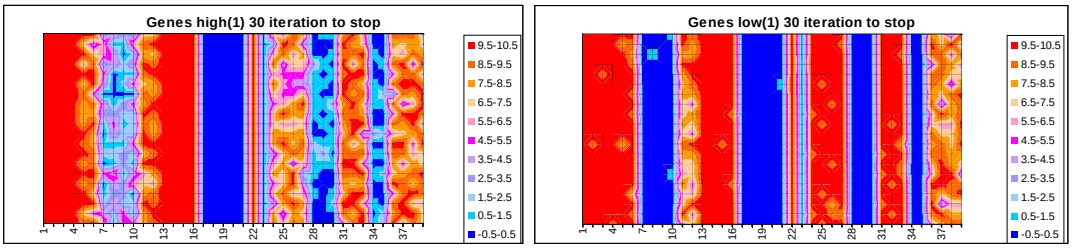

Figure H-36: Simulated Annealing 50 Neighbours 10 Stop Temp