Optimization of a Complex Simulation Model

D.S. Liu[a], E. Post[a], D. Kulasiri[a] and R.A. Sherlock[b]

[a]Centre for Advanced Computational Solutions (C-FACS), Applied Computing, Math & Statistics Group, Lincoln University, Christchurch, New Zealand.

[b]SmartWork Systems Ltd, Christchurch, New Zealand.

Abstract: In this paper we describe techniques utilised in the development of a scheme for identifying the regions in an 8-dimensional parameter space that gave optimal (or near-optimal) performance in a computational simulation of a real-world system. The system model, developed by Dexcel Ltd, attempts a detailed representation of pastoral dairying scenarios. It incorporates sub-models, themselves complex in many cases, of pasture growth, animal metabolism etc. Each evaluation of the objective function, a composite 'farm performance index', requires simulation of at least a one-year period of farm operation with a daily time-step and hence is computationally expensive. Since similar situations are likely to arise in other practical optimisation exercises, the results presented should have some quite general applicability.Two quite different methods of optimisation - Genetic Algorithm (GA) and Lipschitz Branch-and-Bound (LBB) algorithm are investigated and contrasted. Practical issues related to their efficient implementation in a Linux cluster parallel processing environment are discussed and their performance on the above problem is compared. The problem of visualisation of the objective function (response surface) in high-dimensional spaces is also considered in the context of the farm optimisation problem (where from a practical viewpoint knowledge of its behaviour in the region of optima is actually more important than the precise positions or values of the optima themselves). An adaption of the Parallel Coordinates visualization is described which helps visualise some important properties of the model's output topography.

Keywords: *black-box optimization; Lipschitz Branch-and-Bound; Genetic Algorithm; Parallel Coordinates.*

1. INTRODUCTION

The dairy industry is the largest industry of New Zealand and one of the largest dairy exporters in the world (The New Zealand Dairy Industry, 2001). Various research is being pursued to further improve the productivity of New Zealand's dairy farms. Among others research to improve the management of dairy farms is of one the most important fields. Traditionally, various management options are tested in field trials, which are costly in time and resources and also exposed to the perturbations of uncontrolled variables such as climate. As an alternantive in 1997 Dexcel Ltd developed a computer simulation model of New Zealand style dairy farms, namely the Dexcel Whole Farm Model (WFM) (Sherlock *et al*, 1997), and Post parallelized the model in 2002 (Post, 2002). One of the main objects for this computational simulation is to reduce the need for field trials and also make exploratory evaluations of management strategies faster, cheaper, automated and less biased.

However, due to the high complexity of the system studied and intensive computation expense of the Dexcel WFM evaluating all possible variations of New Zealand dairy farm's behavior is impossible within a reasonable time. In addition, the vast quantity and multidimensionality of the model's result data also increase the difficulty of understanding the result. These issues gave rise to our research aiming at identifying some solutions to these problems.

This paper describes an attempt to identify an appropriate optimization method to investigate the interesting region of the output data topography of the Dexcel WFM and also to implement the investigation and present the resultant data topography appropriately so as to facilitate the understanding of it.

2. THE DEXCEL WHOLE FARM MODEL (WFM)

The Dexcel WFM, the simulation model of New Zealand style pasture-based dairy farms, is a large open system based on the Farm System Simulation Framework (FSSF) developed by Sherlock *et al* (1997). The particular version of the WFM used in this study was based on Dexcel's 'Small Test Farm' specification in which just simplistic sub-models of the animal and pasture components are used. The model simulates a one-year period of farm operation,

and returns a scalar "farm performance index" (FPI) of the goodness of farm performance. Conceptually the model is regarded as an 8-variable "black-box" function, which takes 8 farm control variables (AGG, MPG, SR, CD, DOD, IS, IPC and IAL)as the inputs and yields FPI as the output.

Though the analytical expression of the Dexcel WFM is unavailable the following two properties have been found to apply to the model.

- The model can be evaluated on every point within its hyper-rectangular domain. Thus this is an unconstrained function.
- The model is deterministic, i.e. evaluations on the same point always give the same result.

On the basis of the first property, the second property ensures $\frac{|f(x_2)-f(x_1)|}{\|x_2-x_1\|}$ won't be infinitely large, since the $f(x)$ is a bounded finite function and $\|x_2-x_1\|$ won't be zero whilst $|f(x_2)-f(x_1)|$ is not zero. ($\|\ \|$ denotes the Euclidean norm). Therefore the model has a bounded slope. This makes the model a Lipschitz function.

Another fact about the model is that it is computationally expensive. It takes about 1.6 seconds for one evaluation (one year simulation of a small farm) on a Pentium 800Mhz processor. Considering the vast input variable domain of the model an exhaustive search on it is impractical with commonly available computing resources. As an illustration, even a modest 16^8 grid search, i.e. an enumerating search assuming that each of the 8 parameters takes only 16 values, could take more than 27 years on an 8-nodes parallel cluster with Pentium 800Mhz processors. Therefore a major objective in this research is an efficient optimization solution.

3. OPTIMIZATION TECHNIQUES

Among many black-box optimization techniques, Genetic Algorithm (GA) and Lipschitz Branch-and-Bound (LBB) Algorithm were chosen for further investigation in this work, as they are representative methods for stochastic and deterministic black-box optimizations respectively.

3.1. Genetic Algorithm

Genetic Algorithms form a significant branch of the Evolutionary Algorithms mainly developed by Holland (1975). They are stochastic search methods that emulate some processes of natural biological evolution.

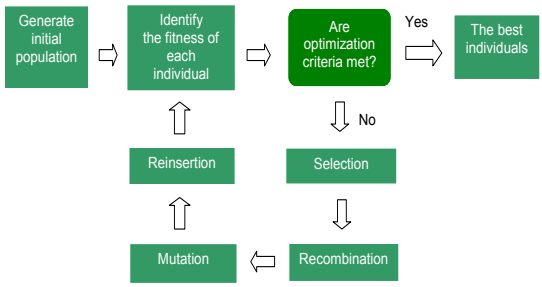


Figure 1: Structure of a single population Genetic Algorithm

As demonstrated in Figure 1 the algorithm starts with a number of individuals (a population) that are randomly initialized and where the genomes of individuals are candidate solutions of the problem. The objective function is then evaluated to identify the fitness of each individual and the initial population is produced. To make the population evolve towards better fitness the algorithm iteratively refreshes the population by a new generation of individuals. This refreshing procedure is achieved by selection, recombination, mutation and reinsertion. In detail, first the best individuals are selected according to their fitness for the production of offspring. The genomes of these parents are decomposed and recombined to produce offspring and the genomes of newly produced offspring mutate with a certain probability. Then the objective function is evaluated to identify the fitness of each offspring. The offspring are reinserted into the population, replacing the individuals with worse fitness. This refreshing cycle is performed until the optimization criteria are reached.

3.2. Lipschitz Branch-and-Bound Algorithm

The Lipschitz Branch-and-Bound algorithm is a family of deterministic optimization algorithms that exploits both a branch-and bound framework and the Lipschitz assumption.

To optimize black box functions by other than stochastic methods, some assumption must be made to make them less "black". Perhaps the most modest assumption we could have on black-box functions is that they have bounded slopes, in which case the black-box functions are said to be *Lipschitz*. More formally, a function $f(x)$ defined on $X \in R^n$ is Lipschitz if it satisfies the condition:

For arbitrary $x_1 \in X$ *and* $x_2 \in X$, *there holds* $|f(x_2)-f(x_1)| \le L\|x_2-x_1\|$

where L is the Lipschitz constant (upper bound of the function's "slope") and $\| \ \|$ denotes the Euclidean norm.

Estimation of the least Lipschitz constant (the tightest upper bound) for black-box functions remains an unsolved problem. Strongin (1973) suggested a method to obtain M as the largest slope in a large number of random samplings, and then obtain the estimation of the least Lipschitz constant by multiplying M by a factor of 2. There is no guarantee that the estimation $2M$ is greater than or equal to the least Lipschitz constant. However, Strongin's estimation is the only technique available for use with black-box functions up to now.

The idea behind Branch-and-Bound (B&B) is a straightforward "divide and conquer" strategy. As Horst, Pardalos and Thoai (1995) described, it partitions the problem into smaller sub-problems (branching) over which the upper bound of the objective function value can be determined (bounding). Some of the sub-problems may then be deleted from further consideration (pruning) if their upper bounds do not meet certain criteria. In Lipschitz Branch-and-Bound Algorithms problems are partitioned by dividing its domain into sub-domains. Thus a sub-problem is the original problem on one of its sub-domains.

Among the many variations of LBB algorithm, the Gourdin, Hansen and Jaumard's (GHJ) algorithm (Gourdin *et al*, 1994) was considered the most efficient one (Hansen and Jaumard, 1995). It partitions the original domain along the longest dimension, and decides the upper bound of a sub-problem according to an evaluation on the centre point of its corresponding sub-domain.

Hansen and Jaumard (1995) made a comparison of all these Lipschitz Branch-and-Bound algorithms by applying them on a wide range of test problems. The result shows that the GHJ algorithm needs significantly fewer function evaluations than other LBB algorithms to achieve an optimization with the same precision ε. The computational overhead other than function evaluation is also the smallest in general. The limitation of this comparison work is that it involves only 2-dimensional and 3-dimensional test problems. However, the advantage of GHJ algorithm on 3-dimensional problems over other algorithms is significantly larger than that on 2-dimensional ones, so it is reasonable to predict it will be even more superior on a still higher dimensional problem.

3.3. Empirical comparison of GA and LBB algorithm

To identify the most efficient optimization technique for use with the Dexcel WFM, we conducted a series of comparative experiments on two representative serial computational implementations of the GA and LBB algorithm.

The experiments were done on 10 sub-domains of Dexcel WFM. The optimal farm performance indexes (FPI) achieved within 500 evaluations on each sub-domain by the GA and LBB implementations respectively were recorded. The domains were all scaled into 8-dimensional hyper-cubical virtual domains with side width 8 in B&B implementation.

The tri-partitioning GHJ Branch-and-Bound algorithm is used in the experiments as the tri-partioning branching rule was considered the most efficient one for the GHJ algorithm. The parameter configuration of the GA for the experiments is as below:

- Genome represented by integer numbers
- Population is 10
- Single-point crossover with probability 1
- Each gene of the genome is mutated with probability 0.2
- Roulette selection based on individual's rank
- 30% of the population is replaced in each generation.

Table shows the results of the experiments on the Dexcel WFM.

The results show that GHJ algorithm, the LBB algorithm claimed to be most efficient, is still inferior to the GA in terms of finding optimal points for Dexcel WFM. On average, an optimum found by a 500-evaluation GA run is 70 higher in FPI than that found by a 500-evaluation GHJ run. Although the GA may risk premature convergence on local optima, the risk is much reduced if it is applied on a fine-cut sub-domain.

We also conducted comparative experiments on a series of test problems other than the Dexcel WFM. These experiments show that the higher the dimensionality of the problem, the more superior the GA is, and also show that the superiority of GA has some degree of generality.

The GA's efficiency on higher dimensionalities can be attributed to its *implicit parallelism*, i.e. the simultaneous allocation of search effort to many dimensions. Figure 2 demonstrates a trace of evaluations for the GA and the GHJ algorithm on minimization of a simple function $f = x + y$. It is seen the GHJ algorithm searches along only one

dimension in every single step, while the GA searches on both dimensions simultaneously. This parallelism of GA makes it more powerful in high-dimensional problems.

It is worth mentioning that the GA implementation in these experiments was not specially configured, otherwise the GA may have achieved even better performance.

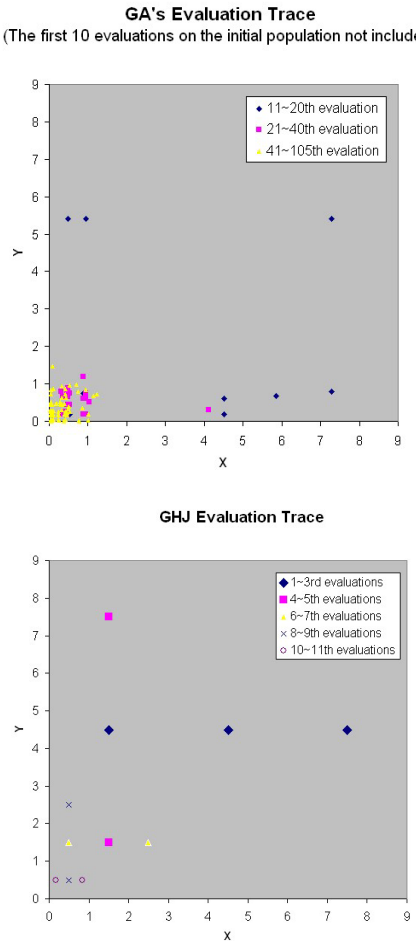


Figure 2: Evaluation trace for GA and GHJ algorithms on a simple 2D function.

Table 1: Optimal performance found by GHJ and GA within 500 evaluations

Zone	Optimal performance found (in FPI)	
	By GHJ	By GA
1	1782	1854
2	1540	1631
3	1578	1669
4	1321	1564
5	1127	1297
6	1644	1690
7	1362	1495
8	1267	1108
9	1101	944
10	1554	1722

3.4. Implementation of GA to investigate the Dexcel WFM

Scheme of Investigation

The first question in this research is how to detect computationally the most interesting prime topography of the Dexcel WFM by using optimization techniques. Since it is not practical to do an exhaustive investigation we instead decided to find the most representative points on the "landscape" that best outlines the topography. These are the extreme points, or local optimal points in our case, of the landscape. For this objective we need to diversify our optimality search effort evenly on the domain.

On this basis we divided the domain of the Dexcel WFM into 6561 (3^8) sub-domains by breaking the range of each of the 8 parameters into 3 sections and applied the Genetic Algorithm, the more efficient black-box optimization technique, on sub-domains in parallel to find their optimal points. We then obtained the overall topography of good farms by studying the optimal points found in these diverse localities. The result of this parallel optima-investigation consists of 6561 nine-component vectors (8 components for the values of the 8 optimising input variables, one for the corresponding optimal FPI) representing the 6561 optimal farm management strategies within respective sub-domains. We named this result data "The Result Data Set " and call it so in the succeeding parts of this paper.

The thoroughness of this investigation depends on not only the precision of those optimizations on sub-domains, but also more crucially on the division of the domain. A division that makes the Dexcel WFM unimodal on each sub-domain is the most favorable to the investigation, but either results in an extra fine-cut division, hence a heavy computation load, or requires advance knowledge of the topography. Thus practically we adopted a modest division on the domain of the Dexcel WFM, which proved adequate for detecting the prime topography in which we are interested.

Parallelization of GA on the Dexcel WFM

The parallelization of the GAs optimising Dexcel WFM on its sub-domains was achieved by using MPI (Message Passing Interface) (Gropp *et al*, 1994) functions. The program was designed for a parallel network consists of one single master node and several slave nodes. In this parallellization, the Master delivers a single job

(A job in the parallelization terms the optimization on a certain sub-domain) to a slave only after it has received a recent response from it. Although this causes more communication load in the network it is robust in case one or several slaves break(s) down, and flexible in job rearrangement after breakdowns happen. In addition it does an implicit dynamic load balancing in the possibly non-dedicated network as well, because the non-dedicated and hence slower processors consequently receive fewer jobs, which is proportional to their processing capacity. The master will possibly be overloaded and become a bottleneck in this parallelization when the number of slaves is very large, but this is not the case in our 9-node network.

4. VISUALIZATION OF THE INVESTIGATION RESULTS

The Result Data Set obtained from the optima-investigation contains information describing the Dexcel WFM's behavioural topography. However, due to the large amount and high dimensionality of the data it is difficult to understand the topography directly. Therefore, we visualized the data by using the Parallel Coordinate technique to improve the understandability of the data.

It is important to note that the conclusions drawn from the data apply to a very small "Test farm" with simplistic models of the animal and pasture components. They may not be applicable to more realistic whole farm models.

The Parallel Coordinate visualizations were produced by an OpenGL visualization application developed in this research. A common problem with Parallel Coordinate visualizations is that the "lines", which depict the vectors, will overlap each other. This problem handicaps observations of the data's topography. In our visualization, we drew the "lines" in a sequence according to the FPI values of the vectors, i.e. the "lines" with lower FPI were drawn earlier, and the "line" with the highest FPI was drawn at last. This ensures that the topography of the good farms, in which we are interested, is always "on the top", rather than covered. On the other hand, to maintain a reasonable comprehensiveness for the visualization, we made the "lines" slightly translucent. Thus the vectors with lower FPI will not be totally covered in some intensively overlapping areas.

Figure 3 shows a Parallel Coordinate visualization of local optimal vectors in the Result Data Set with $FPI \geq 2000$. It is seen from the visualizations that the good farms with FPI $FPI \geq 2000$ are quite diverse in their management strategies, especially the calving date shift (CD) and stocking rates (SR). e.g. while a farm with the earliest calving date (lowest CD) and highest stocking rate (highest SR) achieves a FPI above 2000, that with the latest calving date and lower stocking rate also does so. It is found from the Parallel Coordinate visualization that the farms with FPI above 2000 are also very diverse in their options on milk-production-genetics-scalar (MPG), dry-off date shift (DOD), initial silage (IS) and initial pasture cover (IPC). The only unalterable prerequisites for the farms with FPI above 2000 seem to be higher initial-animal-living-weight (IAL) value and lower animal-growth-genetics-scalar (AGG) value.

5. CONCLUSION

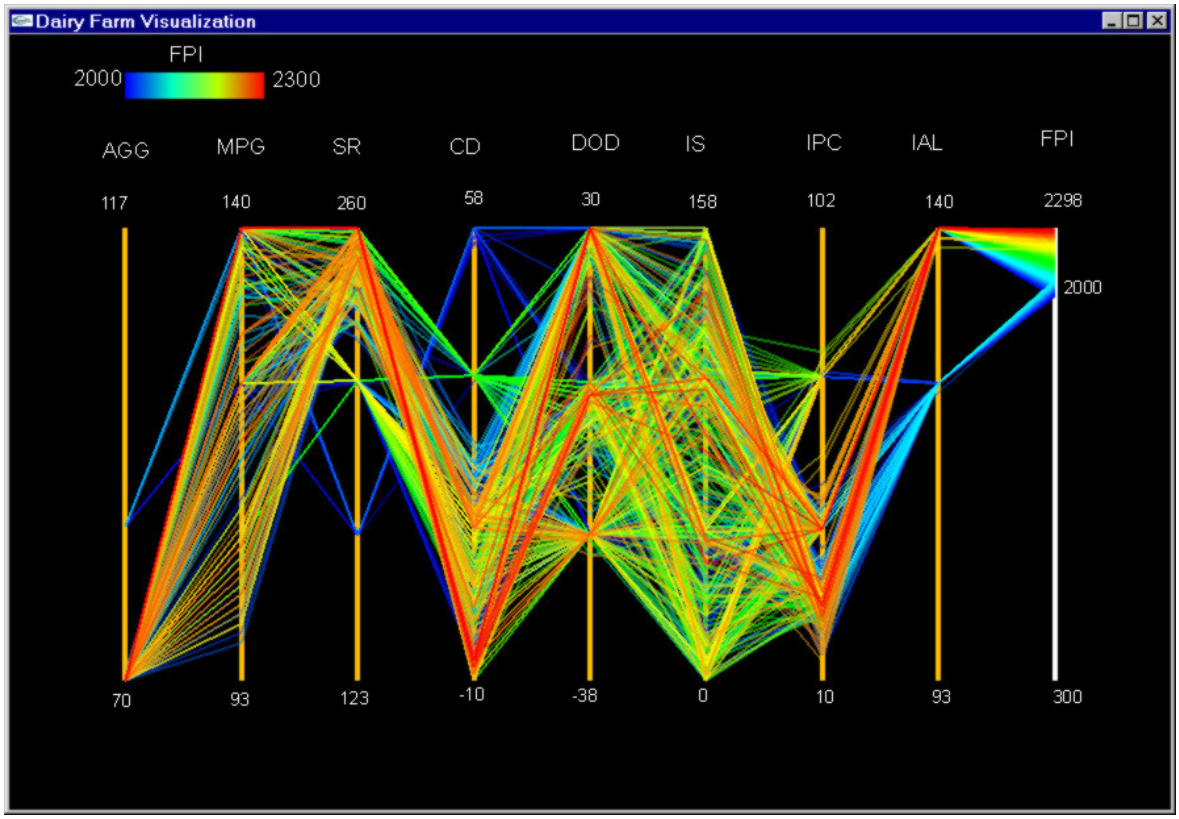
We compared the efficiency of representative GA and LBB implementations in a series of computational experiments. It was observed that the GA needs fewer function evaluations in optimization than the LBB algorithm does. It was also shown that the higher the dimensionality of the problem the more superior the GA is. However, it should not be ignored that the GA is a stochastic strategy that performs inconsistently from run to run and could possibly be trapped by local optima whereas the LBB algorithm can provide definite upper bounds.

In the study of the optimization implementation of Dexcel WFM we investigated the farm behaviour topography by running the parallel GA on sub-domains of the Dexcel WFM and visualized the result data by using Parallel Coordinate techniques. Thanks to the improved understandability of the data in the visualizations, it is observed that to achieve good farm performance with this simple model lower AGG value and higher IAL values are indispensable while values of other management options are more flexible and scenario-dependent.

6. ACKNOWLEDGEMENT

The authors are indebted to Dr M. E. Wastney and Dexcel Ltd for access to source code and permission to use the Dexcel WFM in this study.

Figure 3: Parallel coordinate visualization of the local optimal vectors with FPI ≥ 2000 *



* The gray-scale version of this coloured figure won't be equally informative.

7. REFERENCE

Gourdin, E., P.Hansen and B. Jaumard, *Global optimization of Multivariate Lipschitz Functions: Survey and Computational Comparison*, Les cahier du GERAD, 1994.

Gropp, W., E. Lusk, and A. Skjellum, Using MPI: Portable Parallel Programming with the Message Passing Interface. Massachusetts: The MIT Press, 1994.

Hansen, P. and B. Jaumard, *Lipschitz optimization* in R. Horst and P. M. Pardalos (eds.) *Handbook of Global Optimization* (pp 407-487). Dordrecht/Boston/London: Kluwer Academic Publishers, 1995.

Holland, J. H., Adaptation in Natural and Artificial Systems. Ann Arbor, Michigan: The University of Michigan Press, 1975.

Horst, R., P. M. Pardalos and N. V. Thoai. *Introduction to Global Optimization*, Amsterdam: Kluwer Academic Publishers, 1995.

Post, E., Adventures with portability. *Third LCI International Conference on Linux Clusters: The HPC Revolution 2002*, Florida, USA, 2002.

Sherlock, R.A., K.P. Bright and P.G. Neil, An Object-Oriented Simulation Model of a Complete Pastoral Dairy Farm, *Proceeding of the international conference on Modelling and Simulation, Modelling and Simulation Society of Australia,* Hobart, 1997.

Strongin, R.G., On the convergence of an algorithm for finding a global extremum, *Engineering Cybernetics* 11, 549-555, 1973.