

Optimization Approaches for a Complex Dairy Farm Simulation Model

Jagannath Aryal, Don Kulasiri and Dishu Liu

Abstract—This paper describes the optimization of a complex dairy farm simulation model using two quite different methods of optimization, the Genetic algorithm (GA) and the Lipschitz Branch-and-Bound (LBB) algorithm. These techniques have been used to improve an agricultural system model developed by Dexcel Limited, New Zealand, which describes a detailed representation of pastoral dairying scenarios and contains an 8-dimensional parameter space. The model incorporates the sub-models of pasture growth and animal metabolism, which are themselves complex in many cases. Each evaluation of the objective function, a composite 'Farm Performance Index (FPI)', requires simulation of at least a one-year period of farm operation with a daily time-step, and is therefore computationally expensive. The problem of visualization of the objective function (response surface) in high-dimensional spaces is also considered in the context of the farm optimization problem. Adaptations of the sammon mapping and parallel coordinates visualization are described which help visualize some important properties of the model's output topography. From this study, it is found that GA requires fewer function evaluations in optimization than the LBB algorithm.

Keywords—Genetic Algorithm, Linux Cluster, Lipschitz Branch-and-Bound, Optimization

I. INTRODUCTION

THIS paper presents the optimization of a complex dairy farm simulation model using the two optimization techniques Genetic algorithm (GA) and the Lipschitz Branch-and-Bound (LBB) algorithm. Optimization problems arise in many real-life applications and are intensively studied in different areas. Producing an optimized model of real-world problems is complex and time consuming [1]. As commercial enterprises and systems managers make every effort to become more competitive, numerical optimization techniques are increasingly being used to identify the best solution for modeled systems [2]. Further, computer simulation and optimization can facilitate the production of the best solution without the need to physically test massive numbers of various alternatives. Due to these reasons, optimization techniques are attracting the interests of entrepreneurs as well as researchers. Generally, optimization studies involve: (i) a model of the system of interest, (ii) an objective function, (iii) input

parameters, and (iv) an algorithm for the process [3].

Optimization techniques have been used to solve the agricultural problems in the context of New Zealand agriculture [1] [4]. New Zealand economy is highly dependent on the agricultural industries [5]. Among the various contributing industries, the dairy industry is generating the highest gross revenue (33.9%) equivalent to 5,709 million New Zealand dollars per annum [5]. Further, New Zealand is one of the largest dairy exporters in the world. In this context, there is a high significance of the research that can make even a small contribution to improve the dairy industry. Much research is being focused on improving the productivity of New Zealand's dairy farms. Traditionally, a variety of management options are tested in the field trials, which are costly in time and resources, and are also exposed to the disruptions of uncontrolled variables such as climate. As an alternative, in 1997, Dexcel Limited developed a computer simulation model of New Zealand style dairy farms, namely the Dexcel Whole Farm Model (WFM) [6]. This model was parallelized in 2002 [7]. One of the main objectives of this computational simulation is to reduce the need for field trials and also to explore the management strategies faster, cheaper, automated and less biased. However, due to the high complexity of the system studied and the difficulties in the evaluation of all possible variations of the behavior of New Zealand's dairy farms, there is a large computational expense using Dexcel WFM. In addition, the vast quantity and multidimensionality of the model's resulting data also increase the difficulty of understanding the output of the model.

To deal with the problems mentioned above, stochastic and deterministic optimization techniques have been tested on the model. (A review and comparisons of optimization studies on agricultural and other models can be found in [8]). The present study deals with the application of the GA, a branch of the stochastic method, and the LBB algorithm, a branch of the deterministic method.

GAs form a significant branch of the evolutionary algorithms [9]. They are stochastic search methods that follow some processes of natural biological evolution, and are probabilistic in nature, which is an alternative to traditional optimization methods [10]. GAs have been used to investigate optimization in various fields of study, for example; complex beef property optimization [8], fisheries [10], dairy tofu [11], and grazing systems [12]. As demonstrated in Fig. 1 the algorithm starts with a number of individuals (a population) that are randomly initialized, and where the genomes l of

Jagannath Aryal and Don Kulasiri are with the Centre for Advanced Computational Solutions (*C-fACS*), Lincoln University, Canterbury, P O Box 84, New Zealand (e-mail: {aryal, kulasird}@lincoln.ac.nz).

Dishu Liu is with the Institute of Scientific Computing, Technical University Braunschweig, Germany (e-mail: d.liu@tu-bs.de).

¹ A genome is a blend of gene and chromosome in GA.

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 with a new generation of individuals. This refreshing procedure is achieved by selection, recombination, mutation and reinsertion. More specifically, in the first step, 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 lower fitness. This refreshing cycle is performed until the optimization criteria are reached.

On the other hand, the LBB 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, an 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*.

Reference [13] 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, where M is the largest slope considered. There is no guarantee that the estimation $2M$ is greater than or equal to the least *Lipschitz* constant. However, [13]’s estimation is the only technique available for use with black-box functions up to now. A Branch-and-Bound method proposed in [14] does not use derivatives for solving the reduced problem. The proposed model either determines the infeasibility of the original problem or finds lower and upper bounds for the global solution. The idea behind Branch-and-Bound (B&B) is a straightforward “divide and conquer” strategy. As described in [15], 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 LBB algorithm, problems are partitioned by dividing its domain into subdomains. Thus a subproblem is the original problem on one of its subdomains.

Among the many variations of LBB algorithm, the Gourdin, Hansen and Jaumard’s (GHJ) algorithm [16] was

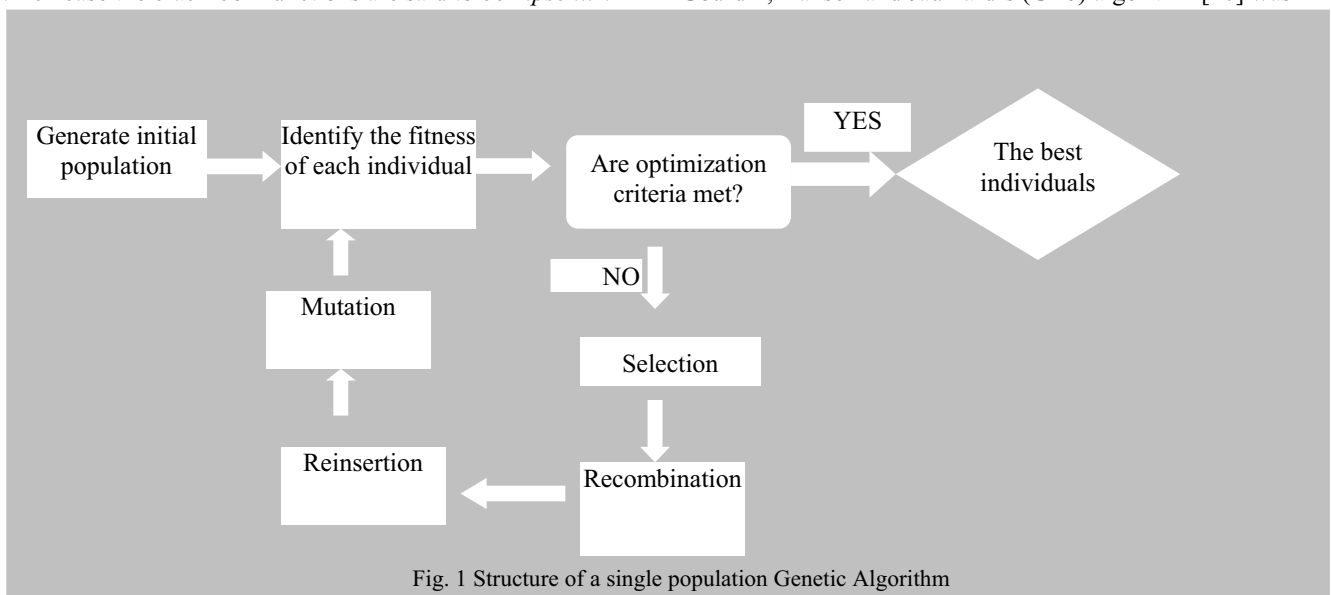


Fig. 1 Structure of a single population Genetic Algorithm

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,

$$|f(x_2) - f(x_1)| \leq L \|x_2 - x_1\| \quad (1)$$

where L is the *Lipschitz* constant (upper bound of the function’s “slope”) and $\| \cdot \|$ denotes the Euclidean norm (Refer to “(1)”).

Estimation of the least *Lipschitz* constant (the tightest upper bound) for black-box functions remains an unsolved problem.

considered the most efficient one [17]. It partitions the original domain along the longest dimension, and decides the upper bound of a sub-problem according to an evaluation on the center point of its corresponding subdomain. A comparison of all these LBB algorithms has made in [17] 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. 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. Therefore,

it is reasonable to predict it will be even more superior on still a higher dimensional problem

The objectives of the present study are: (i) to identify an appropriate optimization method to investigate the interesting region of the output data topography of the Dexcel WFM ; and (ii) to determine the best way of presenting the resultant data topography so as to minimize the difficulty in understanding the model output.

II. DEXCEL WHOLE FARM MODEL (WFM): AN OVERVIEW

The Dexcel WFM, a 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 [7]. The FSSF was designed to integrate component-models representing various real world bio-physical entities, as well as managerial and meteorological factors. The model was developed based on an object-oriented (OO) design of a framework implemented in a prototypical OO language, Visual Works Smalltalk.

The rationale for selecting the OO design included the reliability, reduction in maintenance costs of a large complex system, and the increased re-usability of the software system.

equations (DE's), provide the basic description of the time evolution of the state-variables. An approximation for the relevant processes is possible by rates of transfer of matter and energy between different 'lumped element' pools.

This process thus ignores local spatial dependencies and gives the important reduction to ordinary DE's that involve only time derivatives (Refer to "(2)"):

$$dx_i/dt = f_i(x_1, \dots, x_N, u_1, \dots, u_M) \quad (2)$$

In this farm model, most of the derivative functions $f_i(\cdot)$ have a direct dependence on only a relatively small subset of the SV's.

The particular version of the WFM used in this study was based on on Dexcel's 'Small Test Farm' specification in which simplistic sub-models of the animal and pasture components were used. The model simulates a one-year period of farm operation, and returns a scalar "farm performance index" (FPI²) of the goodness of farm performance. Conceptually the model is regarded as an 8-variable "black-box" function, which takes 8 farm control variables as the inputs and yields FPI as the output, which is shown as a conceptual structure in Fig. 2.

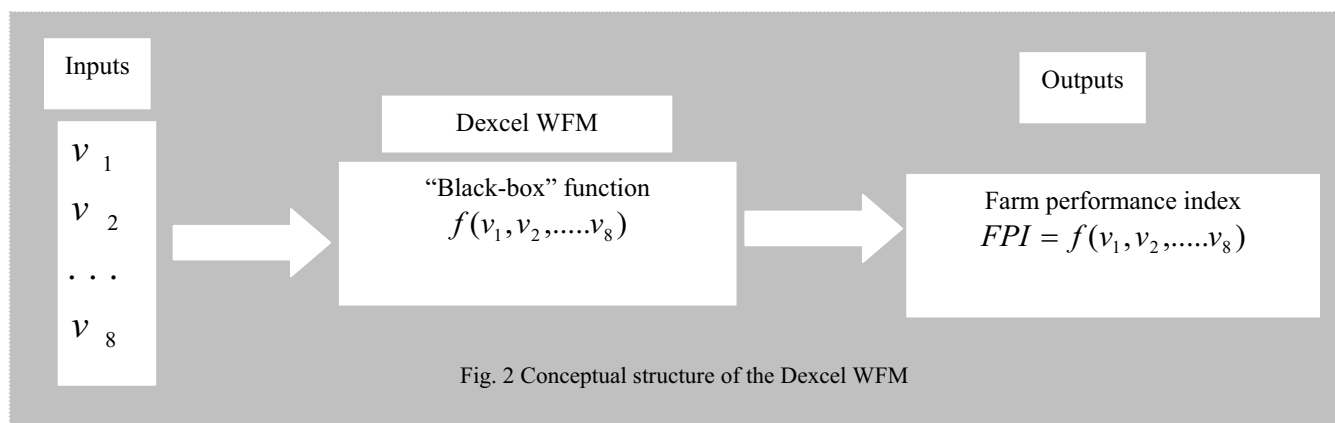


Fig. 2 Conceptual structure of the Dexcel WFM

In addition, an OO design provides mechanisms for managing the synchronization and interaction between component-models during the simulation process. The FSSF's open structure makes it feasible to work with alternative component-models with different properties and of differing levels of complexity, perhaps developed by multiple parties, in different programming languages. This far-sighted flexibility provides unlimited room for future development of the model, and makes it easy to tailor the model for specific research requirements.

More specifically, the farm is represented by a state-variable (SV) description of a continuous-time dynamical system (for eg; [18] [19]). State-variables - x_i - are always related with a storage mechanism (of matter or energy). In addition, they sum up the history of the system with values at a time, t_o , (along with knowledge of the system parameters) which provide all of the information that is needed to calculate future states (and outputs) on the basis of the inputs, u_k , at $t > t_o$ [6]. Differential

The eight variables and their domains are tabulated in Table I. They represent scaling (expressed as percentages) or time-shifts (expressed in days) of the eight most important parameters of the farm simulation. Although these parameters are by nature real number variables, in this study, they were entered as integers in consideration of both computational simplicity and the practical precision of actual dairy farm operations.

Though the analytical expression of the Dexcel WFM is unavailable, the following two properties were found to be applicable to the model;

- The model can be evaluated on every point within its hyper-rectangular domain, which is therefore an unconstrained function.

² FPI is the actual milk-solid production in kg/ha 'corrected' for any differences between the end state of the paddocks and cows and the starting date.

TABLE I. INPUT VARIABLES OF THE DEXCEL WFM

VARIABLES	DOMAIN
V_1 - AGG (Animal growth genetics %age scaling factor)	70 ~ 120
V_2 - MPG (Milk production genetics %age scaling factor)	80 ~ 140
V_3 - SR (Stocking rate %age scaling factor)	80 ~ 260
V_4 - CD (Calving date shift – day)	-10 ~60
V_5 - DOD (Dry-off date shift – day)	-40 ~ 30
V_6 - IS (Initial silage %age scaling factor)	0 ~ 200
V_7 - IPC (Initial pasture cover %age scaling factor)	10 ~ 140
V_8 - IAL (Initial animal living weight %age scaling factor)	80 ~ 140

- 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 $\frac{|f(x_2) - f(x_1)|}{\|x_2 - x_1\|}$ will not be infinitely large, since the $f(x)$ is a bounded finite function and $\|x_2 - x_1\|$ will not be zero whilst $|f(x_2) - f(x_1)|$ is not zero. Therefore, the model has a bounded slope. This makes the model a *Lipschitz* function [17].

Another fact about the model is that it is computationally expensive. Considering the vast input variable domain of the model, an exhaustive search on it is impractical with commonly available computing resources. Therefore, a major objective in this research is determining an efficient optimization solution.

III. METHODS

A. Empirical comparison of GA and LBB

To identify the most efficient optimization technique in the case of Dexcel WFM, we conducted a series of comparative experiments on two representative serial computational implementations of the GA and LBB algorithm. The experiments were conducted on 10 sub-domains of Dexcel WFM. The optimal farm performance indices (FPI) achieved within 500 evaluations on each sub-domain by the GA and LBB implementations respectively were reported. The domains were all scaled into 8-dimensional hyper-cubical virtual domains, with side width 8 in B&B implementation.

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

- 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 individuals
- 30% of the population is replaced in each generation.

B. GA implementation: investigation, parallelization and visualization

Scheme of *investigation* was applied keeping in mind that how to detect 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 needed to diversify our optimality search effort evenly on the domain. On this basis, we divided the domain of the Dexcel WFM into 6561 sub-domains by breaking the range of each of the 8 parameters into 3 sections, and applied the GA on the 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 optimizing input variables, and one for the corresponding optimal FPI), which represent all of the optimal farm management strategies within the respective sub-domains. This data set is referred to hereafter as - *The Result Data Set (TRDS)*.

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. However, it either results in an extra fine-cut division, and hence a heavy computation load, or requires advance knowledge of the topography. We therefore adopted a modest division on the domain of the Dexcel WFM, which proved adequate for detecting the prime topography of interest.

The *parallelization* of the GAs optimizing Dexcel WFM on its sub-domains was achieved by using Message Passing Interface (MPI) [20] functions. The program was designed for a parallel network consisting of one single master node and several slave nodes. Due to the large amount and high dimensionality of the data it is difficult to understand the topography directly hence the *visualization* was carried out using sammon mapping [21] and the parallel coordinate techniques.

Sammon mapping is a non-linear projection technique that projects high-dimensional vectors onto a 2-dimensional map while trying to preserve the high-dimensional topography of

the original data. Using sammon mapping functionality (SOM-PAK 3.1), we mapped the 8-dimensional vectors onto a 2-dimensional space. Further, the corresponding FPI were mapped in a color range, thus we obtained a set of points of various colors on an X-Y space. The domain of the Dexcel WFM was divided into 3^8 (6561) zones by breaking the range of each of the 8 input variables into 3 equal sections, i.e. sections with relatively low, median and high values. Considering the limitations of Sammon mapping such as treating every farm parameters with equal weight, we calculated the contribution of each farm parameters. The correlation between the FPI and each farm parameter was calculated that helped to observe the contribution of each farm parameters.

The parallel coordinate visualizations were produced using an OpenGL visualization application developed in this study. The advantages and disadvantages of both visualization methods are presented in the results and discussion section.

IV. RESULTS AND DISCUSSIONS

A. Empirical comparison of GA and LBB algorithm

The result of the experiments on the empirical comparison of GA and GHJ algorithm for Dexcel WFM is presented in Table II.

TABLE II. OPTIMAL PERFORMANCE FOUND BY GHJ AND GA WITHIN 500 VALUATIONS

ZONE	DOMAIN	
	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

The results show that GHJ algorithm appeared to be most efficient, but is still inferior to the GA in terms of finding optimal points for Dexcel WFM. On average, an optimum FPI derived from a 500-evaluation GA run is 70 units higher in FPI than that derived from a 500-evaluation GHJ run (Table II). Although the GA may risk premature convergence on local optima, the risk is significantly reduced if it is applied on a fine-cut sub-domain.

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. A result observed for the trace of evaluation for the GA and the GHJ

algorithm on minimization of a simple function (Refer to "(3)"),

$$f = x + y \quad (3)$$

This investigated result showed that 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.

B. Visualization of the investigated results

The *visualization* result for *TRDS* from sammon mapping as presented in Fig.3 shows a set of points of various colors on a 2-dimensional space, which is much more comprehensible than the data's original presentation. In Fig. 3, the colors of the points indicate the FPI values, and the distances between the points depict the degree of similarity of the values of the 8 farm parameters between them. The points are grouped mainly by four principal parameters, namely IAL, CD, AGG and SR (defined in Table 1), as labeled accordingly in the map. For example, a label "HLLH" means this group consists of vectors with high (H) IAL value, low (L) CD value, low AGG value and high SR value. However, it was learned that projecting 6561 (3^8) nine-dimensional vectors onto a single 2-D map is not practical, due to both the limitation of computing resources and also the projection error.

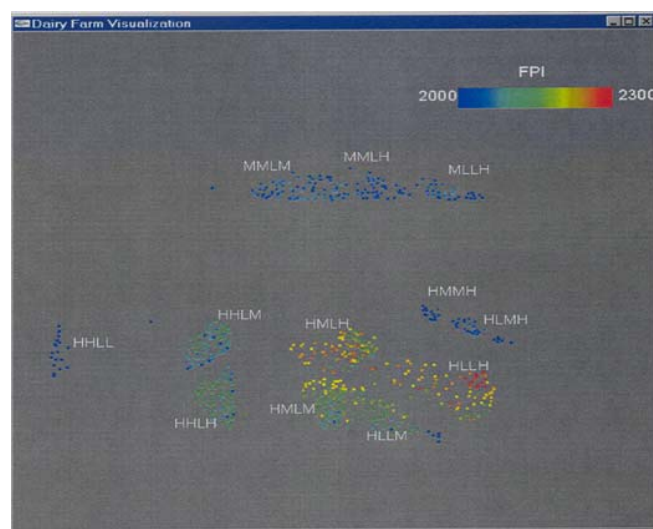


Fig. 3 Sammon map of the local optimal vectors with FPI>=2000

The results from the correlation analysis as presented in Table III for $FPI \geq 2000$ show that the parameters diverge greatly in their correlations with FPI. The parameters IAL, CD, AGG and SR have more effects than the others. This correlation result suggested that the above parameters which have high correlation also have high weight.

Parallel coordinate visualization revealed a well depicted vector map of *TRDS*. 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.

TABLE III. FARM PARAMETERS AND THEIR CORRELATION WITH FPI

VARIABLES	CORRELATION WITH FPI
AGG	-0.29
MPG	-0.02
SR	0.19
CD	-0.29
DOD	-0.02
IS	-0.05
IPC	-0.06
IAL	0.56

Fig. 4 shows a parallel coordinates visualization of the local optimal vectors in *TRDS* with FPI. It is seen from the visualizations that the good farms with $FPI \geq 2000$ are quite diverse in their management strategies, especially the calving date shift (CD) and stocking rates (SR). For *eg*; 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 (Table IV). 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.

V. CONCLUSION

In this paper, we compared the efficiency of representative GA and LBB implementations in a series of computational experiments. It was observed that the GA requires 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.

Further, we also investigated the farm behavior topography by running the parallel GA on sub-domains of the Dexcel WFM and visualized *TRDS* using both sammon mapping and parallel coordinate techniques. Due to the improved understandability of the data in the visualizations, it was observed that in order 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.

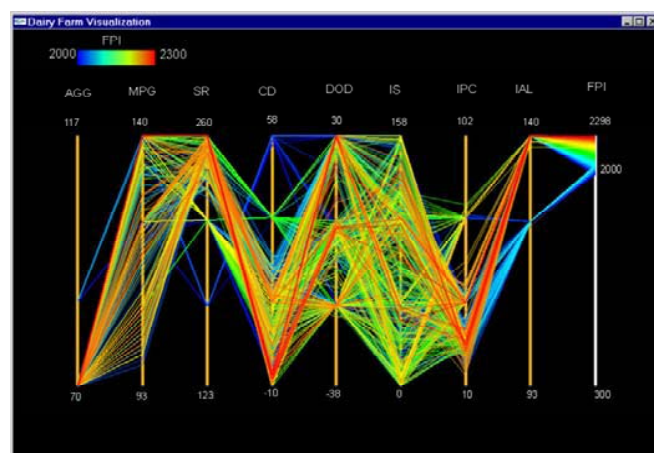


Fig. 4 Parallel coordinates visualization of the local optimal vectors with $FPI > 2000$

For the improvement of the optimization technique in the future, the Bayesian Evolution Algorithm (BEA) needs to be considered. Though BEA is generally more likely than GA and LBB to be trapped by local optima, it is not a significant problem if we use it in combination with a fine-cut domain division, in which case the Dexcel WFM on each sub-domain has less complexity and is more likely to be unimodal. A comprehensive visualization tool that can present 4D topography would be more informative and more helpful in understanding the FPI topography of the Dexcel WFM.

ACKNOWLEDGMENT

Authors would like to thank Dr. R. A. Sherlock and Ms E. Post for their valuable advice in this project.

REFERENCES

- [1] R.P.S. Hart, M.T. Larcombe, R.A. Sherlock, and L.A. Smith, Optimization techniques for a computer simulation of a pastoral dairy farm, *Computers and Electronics in Agriculture*, 19 (1998), 129-153.
- [2] M.C. Fu, Optimization via simulation: a review, *Annals of Operations Research*, 53 (1994), 199-247.
- [3] D.G. Mayer, B. P. Kinghorn, and A.A. Archer, Differential evolution – an easy and efficient evolutionary algorithm for model optimisation, *Agricultural Systems*, 83 (2005), 315–328.
- [4] A.E. Dooley, W.J. Parker, and H.T. Blair, Modelling of transport costs and logistics for on-farm milk segregation in New Zealand dairying, *Computers and Electronics in Agriculture*, 48 (2005), 75–91.
- [5] The Treasury. (2007, November 25). Available: http://www.treasury.govt.nz/economy/overview/2007/07.htm#_tocPrimary_Industries
- [6] R.A. Sherlock, 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 (1997), Hobart.
- [7] E. Post, Adventures with portability. Third LCI International Conference on Linux Clusters: The HPC Revolution 2002, Florida (2002), USA.
- [8] D.G. Mayer, J.A. Belward, and K. Burrage, Robust parameter settings of evolutionary algorithms for the optimisation of agricultural systems models, *Agricultural Systems*, 69 (2001), 199–213
- [9] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, Michigan: The University of Michigan Press, 1975.
- [10] S.J. Mardle, S. Pascoea and M. Tamizb, An investigation of genetic algorithms for the optimization of multi-objective fisheries bioeconomic models, *International Transaction in Operational Research*, 7.(2000), 33 - 49
- [11] M.J. Chen, K.N. Chen, and C.W. Lin, Optimization on response surface models for the optimal manufacturing conditions of dairy tofu, *Journal of Food Engineering*, 68 (2005), 471– 480.
- [12] L.G. Barioni, C.K.G. Dake, and W.J. Parker, Optimizing rotational grazing in sheep management systems, *Environment International*, 25, 617 (1999), 819-825.
- [13] R.G. Strongin, On the convergence of an algorithm for finding a global extremum, *Engineering Cybernetics* 11 (1973), 549-555.
- [14] Y.D. Sergeyev, D. Famularo, and P. Pugliese, Index branch-and-bound algorithm for Lipschitz univariate global optimization with multiextremal constraints, *Journal of Global Optimization*, 21, (2001) 317–341.
- [15] R. Horst, P.M. Pardalos, and Thoai N.V., *Introduction to Global Optimization*, Kluwer Academic Publishers, Amsterdam, 1995
- [16] E. Gourdin, P. Hansen and B. Jaumard, *Global optimization of Multivariate Lipschitz Functions: Survey and Computational Comparison*, Les cahier du GERAD, 1994.
- [17] P. Hansen 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
- [18] L. Padula and M.A. Arbib, *System Theory*, W.B. Saunders, Philadelphia, 1974.
- [19] S. J. R. Woodward, Dynamical systems models and their application to optimising grazing management, In R.M. Peart and R.B. Curry (eds.), *Agricultural Systems Modelling and Simulation*, Marcel Dekker, 1997.
- [20] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message Passing Interface*, The MIT Press, Massachusetts, 1994.
- [21] J.W. Sammon, A non-linear mapping for data analysis. *IEEE Transactions on computers*, 5 (1969), 401-409.

TABLE IV
OPTIMUM VECTORS OF THE GROUPS WITH $FPI \geq 2000$

GROUP	NUMBER OF VECTORS	THE OPTIMUM VECTOR								
		FPI	AGG	MPG	SR	CD	DOD	IS	IPC	IAL
HLLH	81	2298	70	140	260	-7	30	144	23	140
HMLH	81	2270	70	124	255	13	5	96	22	140
HMLM	81	2232	70	140	214	13	30	137	28	140
HLLM	74	2237	70	140	214	11	30	158	27	140
HHLM	79	2150	70	140	214	36	30	21	28	140
HHLH	81	2179	70	140	260	36	26	157	20	140
HLMH	27	2049	86	139	260	4	30	134	31	140
HMMH	15	2041	86	140	260	13	30	143	27	140
HHLL	20	2035	70	140	167	58	30	153	22	140
MMM	32	2052	70	140	214	20	30	158	27	124
MMLH	66	2073	70	140	236	23	30	108	43	124
MLLH	44	2069	70	140	260	-1	30	153	22	124

Jagannath Aryal received his Professional Masters Degree in Geo-informatics from International Institute for Geo-information and Earth Observation (ITC), The Netherlands (2003), a research M.Sc. Degree in image processing from the University of Otago, New Zealand (2006). Mr. Aryal became a student member (SM) of Society for Industrial and Applied Mathematics (SIAM) in 2006. His current research interests include application of mathematics and computational modelling.

D. Kulasiri received his Masters Degree (1988) and a Ph D (1990) from Virginia Tech, USA. He is currently a Professor of Computational Modelling and Simulation at the Lincoln University, New Zealand. Prof. Kulasiri is the head of the Centre for Advanced Computational Solutions (C-fACS) in Lincoln University, New Zealand. Prof. Kulasiri is the member of Modelling and Simulation Society of Australia and New Zealand (MSSAN), American Society for Agricultural Engineers (ASAE), International Society for

Computational Biology (ISCB), Society for Industrial and Applied Mathematics (SIAM).

Prof. Kulasiri is a visiting professor to the Mathematical Institute, Oxford University UK; a visiting Professor to Institute of Scientific Computing, Technical University of Braunschweig, Germany; Visiting Fellow, Princeton University, New Jersey, USA.

Dishi Liu received his Master of Applied Science degree from Lincoln University, New Zealand (2003). He is a Ph D student in the Institute of Scientific Computing, Technical University of Braunschweig, Germany. His current research interests include application of mathematics and computational modelling.