SIMBREED: Simulated Animal Breeding as a Computational Metaphor for Optimizing Agricultural Systems

Jaderick P. Pabico and Orville L. Bondoc

***Abstract*— We present a breeding-inspired computational method for solving optimization problems in agricultural systems. We discuss SIMBREED, a set of computer heuristics we developed, that simulates and uses animal breeding as a computational metaphor for finding solutions to combinatorial optimization problems such as the general diet formulation problem (DFP). We use DFP as a representative problem to demonstrate that SIMBREED algorithms can efficiently solve complex optimization problems. We present solutions to DFP by mapping morphogenetic traits to DFP variables and satisfy the objective function via natural selection or selective breeding, with small chance of mutation. We compare the DFP solution found by ADBASE, a deterministic algorithm, to those found by SIMBREED algorithms. We also compare the performance of two SIMBREED algorithms: simulated natural selection (EVOLVE) and simulated selective line breeding (BREED). Results show that SIMBREED can be used to find solutions to complex combinatorial optimization problems such as the DFP better than a deterministic algorithm and that the simulated selective line breeding performs better than the simulated natural selection.**

***Index Terms*— Breeding, combinatorial, computation, heuristics, SIMBREED, optimization.**

I. Introduction

Simulating a breeding program using a computer would be useful to find solutions to combinatorial optimization problems inherent in most agricultural systems such as the diet formulation problem (DFP). Usually, DFPs are simply formulated using linear programming and solved using the simplex or tableau pivoting method via long-hand computation or by a computer software [6]. However, when risks associated with the variability of nutrients and modeling of nutrient requirements in local (i.e. individual requirements) and global (i.e. average requirements of the population) levels are incorporated in the formulation, the problem grows in complexity and becomes mixed-integer nonlinear programming (MINLP) [11]. Although there are commercial computer software available for solving MINLP, most of them are constrained by the number of variables the MINLP is formulated with, while their runtimes become inefficient as the complexity of the MINLP increases. This scenario is also true for most problems in agricultural systems.

Up until the present time, several mathematical methodologies were used to accomplish the task of finding solutions to DFPs. Examples of these procedures are Pearson's square, solving simultaneous equations, matrix algebra, linear programming (LP), mixed-integer linear programming (MILP), stochastic nonlinear programming (SNLP), fuzzy programing, dynamic programing, goal programming, multiple criteria optimization, and weighted Tchebycheff [7], [14], [15], [11], [12], [3], [13], [1]. All of these procedures use deterministic algorithms to find solutions that would satisfy an user-defined objective function (some are even useful for multiple-objective functions). Because of the deterministic nature of these algorithms, the methods become inefficient in terms of computer time when the DFPs are formulated with increasing complexity (e.g. the number of variables is very large). The formulation becomes more complex when the variable types are a mix of integer- and real-valued numbers and at least one of the constraints is nonlinear. When this happens, the problem becomes a MINLP, which is a generalization for LP and NLP. Although some of the approaches mentioned above can be used to solve MINLP, the lexicographic nature of existing deterministic procedures may only reach local optima, not the true global optima. The process of finding the optimal values becomes inefficient and difficult even when using expensive propriety deterministic packages that can handle complex MINLP formulations [4], [14].

Our proposed solution is a computational procedure based on animal breeding metaphor. Because it uses a stock of solutions that solves the problem in implicitly parallel way [5], the methodology is efficient, robust, and able to handle complex MINLP formulations. The mapping of genetic traits to DFP variables, and the algorithm that will drive a stock of DFP solutions to breed better solutions should be capable of finding optimum or near-optimum solutions to any complex optimization problem in agricultural systems.

In the current effort, we developed a multi-agent-based stochastic computer heuristic, called SIMBREED, to solve combinatorial optimization problems such as the MINLP for DFP. In SIMBREED, we designed a pair of algorithms to simulate natural selection and selective line breeding, with a small chance of mutation. These algorithms were used to solve a representative DFP formulated under the MINLP. The performance of SIMBREED algorithms were then compared with each other.

In this paper, SIMBREED is presented as a novel procedure

To appear in the Philippine Journal of Veterinary and Animal Sciences.
Extended Abstract appeared in the Proceedings of the 40th Scientific Seminar and Annual Convention of the Philippine Society of Animal Science, October 23–24, 2003, Heritage Hotel, Manila, pp. 96–97.
Adjudged Best Scientific Poster during the same convention.

for solving a complex optimization problem in agricultural systems using animal breeding as a computation metaphor. DFP, formulated as MINLP, is used as a representative problem of a complex agricultural system. Construction details of the mapping between genetic traits and DFP variables are given. The algorithms that drive the selection, and the mating and mutating methodologies are also discussed. Results of the performance of a deterministic algorithm and the different SIMBREED algorithms are presented.

II. Materials and Methods

A. A Representative DFP

We selected a very large and complex diet formulation problem available in the Internet as a test bed [9]. The MINLP formulation is defined using the mathematical notations described in Table I. The objective is to minimize the linear function described in equation 1 subject to constraints defined by equations 2 through 9.

$$\text{Min} \sum_{i=1}^{||\text{Foods}||} (\text{cost}_i \times x_i) \tag{1}$$

$$\text{s.t.cost}_i > 0 \tag{2}$$

$$\text{food}_{i,\min} \geq 0 \tag{3}$$

$$\text{food}_{i,\max} \geq \text{food}_{i,\min} \tag{4}$$

$$\text{nutr}_{\min,j} \geq 0 \tag{5}$$

$$\text{nutr}_{\max,j} \geq \text{nutr}_{\min,j} \tag{6}$$

$$\text{nutr}_{i,j} \geq 0 \tag{7}$$

$$\text{food}_{i,\max} \geq x_i \geq \text{food}_{i,\min} \tag{8}$$

$$\text{nutr}_{\min,j} \leq \sum_{i=1}^{||\text{Foods}||} (\text{nutr}_{i,j} \times x_j) \leq \text{nutr}_{\max,j} \tag{9}$$

$$\forall \quad i = 1, 2, \ldots, ||\text{Foods}||$$

$$\forall \quad j = 1, 2, \ldots, ||\text{Nutrients}||$$

To model the variability of nutrient requirements in the local and global levels, we reformulated the problem as MINLP by changing some of the constraints (Equations 8 and 9) using the method discussed by [12]. We used 64 variables, 20 of which are integer-valued while the rest are real-valued (Table II). There were 11 nutritional values per variable namely calories, cholesterol (mg), total fat (g), Na (mg), carbohydrates (g), dietary fiber (g), protein (g), Vitamin A (IU), Vitamin C (IU), Ca (mg), and Fe (mg). The cost per serving for each variable was also given. All nutrient values were taken from the USDA National Nutrient Database for Standard Reference [17] while the recommended daily allowances for each nutrient were taken from the National Agricultural Library [8]. We solved the DFP using a conventional deterministic method called ADBASE [16] that was ported to Linux using a standard Fortran.

B. Algorithms in SIMBREED

The SIMBREED algorithms presented in this paper are the following:

1) Finding DFP solutions using natural selection or evolution (EVOLVE); and
2) Finding DFP solutions using selective line breeding (BREED).

Modeling natural selection or evolution, EVOLVE was taken from the simple genetic algorithm described by [5] and implemented to solve complex multi-dimensional, multi-modal, noisy, and discontinuous optimization and search problems in agriculture [10], [2]. We modified EVOLVE to give the algorithm more direction in searching the optimal solutions by introducing a selective line breeding program we called BREED. The computer programming language we used to implement the algorithms was Perl [18].

C. Mapping of Morphogenetic Traits to DFP Variables

Each individual in the SIMBREED stock is portrayed as a sequence of binary characters that encode a solution to DFP. The sequence, aptly called chromosome, is similar to the DNA structure in genetics such as the one shown in Figure 1. The length of the chromosome is represented by the total number of DNA molecules the chromosome has. The DNA molecules in our context are the binary numbers 0 and 1 as opposed to the real molecules in genetics (purines and pyramidines) which symbolically are A, C, T, G, and U (respectively adenine, cytosine, thymine, guanine, and uracil). Binary number is the natural number system used by computers. The number of encoded genes in the chromosome is equal to the number of variables in the DFP (i.e. 64 in this study). Each gene is composed of DNA molecules whose length is defined by mapping the values between the morphogenetic trait that the gene encodes and the DFP variable.

Given the vector of variables $X = (x_1, x_2, \ldots, x_n)$ for DFP, where some $x_i \in \mathcal{R}$ and some $x_j \in \mathcal{I}$, we mapped each $x_k \in X$, $\forall k = 1 \ldots n$, to a specific morphogenetic trait using the concatenated, multiparameter fixed-point discretized binary encoding described in the following algorithm:

1) For each variable $x_k \in X$, $\forall k = 1 \ldots b$, we got the interval $[U_{k,\min}, U_{k,\max}]$ and the desired resolution r_k from the DFP constraints (Equation 8). The number of discretized search point for x_k was computed as $(U_{k,\max} - U_{k,\min}) \times 10^{r_k}$. The length of the gene that represented x_k was
$$l_k = \left\lceil \log_2 \left((U_{k,\max} - U_{k,\min}) \times 10^r \right) \right\rceil . \tag{10}$$
We defined the resolution r_k as the number of decimal points that describes the precision of x_k when $x_k \in \mathcal{R}$. If $x_k \in \mathcal{I}$, then $r = 0$. $\mathcal{R}$ is the set of real-valued numbers while $\mathcal{I}$ is the set of integer-valued numbers.
2) The lower bound $U_{k,\min}$ corresponded to the binary string 0^{l_k} while the upper bound $U_{k,\max}$ corresponded to the binary string 1^{l_k}.
3) We linearly mapped the intermediate values $(U_{k,\min}, U_{k,\max})$ between the strings 0^{l_k} and 1^{l_k} to create a sequence of characters of the form $(0|1)^{l_k}$.
4) We computed the length L of the chromosome as the sum of the lengths of all genes. This is simply given as

TABLE I
THE MATHEMATICAL NOTATIONS USED IN FORMULATING THE DFP.

Notation	Description
$\text{Foods} = \{\text{Beef}, \ldots, \text{carrots}\}$	The set of foods
$\text{Nutrients} = \{\text{A}, \ldots, \text{C}\}$	The set of nutrients
cost_i	cost of the ith food, $1 < i < \|\|\text{Foods}\|\|$
x_i	unknown amounts of food to be included in the diet, $1 < i < \|\|\text{Foods}\|\|$
$\text{nutr}_{i,j}$	amount of the jth nutrient in the ith food, $1 < i < \|\|\text{Foods}\|\|$ and $1 < j < \|\|\text{Nutrients}\|\|$
$\text{nutr}_{\text{min},j}$	minimum amount of jth nutrient required to be taken per day, $1 < j < \|\|\text{Nutrients}\|\|$
$\text{nutr}_{\text{max},j}$	maximum amount of jth nutrient required to be taken per day, $1 < j < \|\|\text{Nutrients}\|\|$
$\text{food}_{i,\text{min}}$	minimum amount of ith food desired to be taken per day, $1 < i < \|\|\text{Foods}\|\|$
$\text{food}_{i,\text{max}}$	maximum amount of ith food desired to be taken per day, $1 < i < \|\|\text{Foods}\|\|$

TABLE II
THE 64 VARIABLES AND TYPES OF VARIABLE USED IN THE DFP.

Variable	Type	Variable	Type
Pack of broccoli	R	Cup of shredded carrots	R
Stalk of celery	I	Cup of frozen corn	R
Leaf of lettuce	I	Pc. of peppers	I
Block of tofu	R	Kg. of roasted chicken	R
Cup of spaghetti	R	Pc. of tomato	I
Pc. of apple	I	Pc. of banana	I
Pc. of grapes	I	Pc. of kiwi fruit	I
Pc. of orange	I	g. of bagels	R
Slice of wheat bread	R	Slice of white bread	R
Pc. of oatmeal cookie	I	g. of apple pie	R
Pc. of chocolate chip cookie	I	g. of butter	R
g. of cheddar cheese	R	L. of whole milk	R
L. of low fat milk	R	L. of skim milk	R
Pc. of poached egg	I	Pc. of scrambled egg	I
g. of bologna	R	g. of frankfurter	R
Slice of ham	I	Slice of kielbasa	I
g. of Cap'N Crunch	R	g. of Cheerios	R
g. of corn flakes	R	g. of raisin bran	R
g. of Rice Krispies	R	g. of Special K	R
g. of oatmeal	R	g. of Malt-O-Meal	R
slice of pizza	I	Pc. of taco	I
Pc. of hamburger	I	Pc. of hotdog	I
Cup of Couscous	R	Cup of rice	R
Cup of macaroni	R	Tbsp. of peanut butter	R
Kg. of pork	R	Pc. of sardines	I
g. of tuna	R	g. of popcorn	R
g. of potato chips	R	g. of pretzels	R
g. of tortilla chips	R	Cup of chicken noodle soup	R
Cup of ham soup	R	Cup of vegebeef soup	R
Cup of New England clam chowder	R	Cup of tomato soup	R
Cup of clam chowder with milk	R	Cup of cream of mushroom soup	R
Cup of bean and bacon soup	R	Cup of baked potato	R

R=Real valued; I=Integer-valued
Pc.=Piece; g.=gram; Kg.=Kilogram; L.=Liter; Tbsp.=Tablespoon

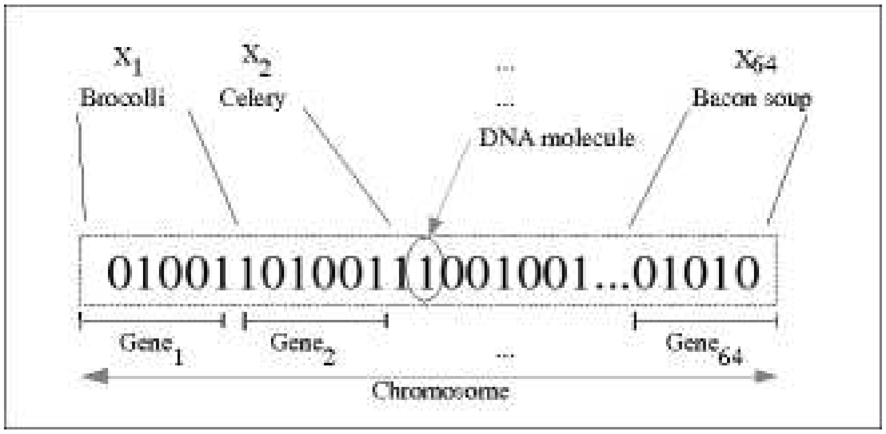

Fig. 1. Encoding of DFP variables into SIMBREED chromosomes.

$L = \sum_{k=1}^{n} l_k$. The total discretized search space was

$$\prod_{k=1}^{n}\left((U_{k,\max} - U_{k,\min}) \times 10^r\right). \tag{11}$$

We decoded the values in the chromosome by reversing the process of encoding. We transformed the binary digits into the corresponding integer value a_k between the interval $[0, 2^{l_k}]$ and then linearly mapped a_k back into the interval $[U_{k,\min}, U_{k,\max}]$ using

$$x_k = U_{k,\min} + \frac{U_{k,\max} - U_{k,\min}}{2^{l_k} - 1}\left(\sum_{j=0}^{l_k-1} g_{(l_k-j)\times 2^j}\right) \tag{12}$$

where g is a binary number.

This mechanism implies that in the hyperspace of DFP variables X, only a search on discretized grid points is performed [10], such that instead of the true global optimum point, SIMBREED searched for the grid point whose objective function value is nearest to the true global optimum. However, it is possible that SIMBREED can find the true global optimum when all x_k are integers (i.e. the problem is transformed as an integer nonlinear programming).

D. Reproduction of Solutions via Mating and Mutation

The mating procedure employed to produce new solutions out of parent solutions is outlined in the following algorithm:

1) The mating procedure we used is the 3-point crossover.
2) For all pairs of solutions in the SIMBREED stock, we generated three random integers o_1, o_2, and o_3 between 2 and $(L - 1)$ such that $o_1 << o_2 << o_3$.
3) For each parent pair A_1 and A_2 from the SIMBREED stock, we swapped the sequence of characters between molecules in position o_1 through o_2 and o_3 to L to create two offspring individuals B_1 and B_2. This step is summarized in Figure 2.

We implemented mutation using the following algorithm:

1) The chance that a particular molecule of the chromosome is mutated is set at a constant value of 0.001.
2) For each molecule a_k, $k = 1 \ldots L$, we generated a random number c_k between the inclusive range $[0, 1]$.
3) If $c_k < 0.001$, then the molecule was mutated. A mutated binary molecule of value 1 was flipped to 0 while a mutated 0 was flipped to 1.

E. Similarity Metric Between Two Solutions

We defined a metric that would measure the similarity of morphogenetic traits between two solutions. This metric is the Hamming distance $\mathcal{H}$ between two binary strings b_1 and b_2 defined mathematically as:

$$\mathcal{H} = \sum_{i=1}^{L}\left|b_{1,i} - b_{2,i}\right|, \tag{13}$$

where $b_{1,i}$ and $b_{2,i}$ are the ith molecule of respective solutions b_1 and b_2. The smaller the value of $\mathcal{H}$, the more similar the two solutions are. If $\mathcal{H} = 0$, then the two solutions are identical (i.e. one is the exact clone of the other).

F. EVOLVE: Finding Solutions Based on Evolution

We designed EVOLVE as the algorithm that will find solution using natural selection or evolution. This algorithm is based on a powerful multi-agent stochastic algorithm known as the genetic algorithm [5], [10]. The steps for evolving solutions are as follows:

1) We started with a stock of solutions. The number of solutions in the stock is set to 50.
2) For each solution A_1 in the stock, we assigned another solution A_2 as A_1's mate.
3) The offsprings B_1 and B_2 of the parent pair A_1 and A_2 were generated using the algorithm discussed in section II-D.
4) Each of the offspring B_k was subjected to mutation using the algorithm discussed in section II-D.
5) A new stock is selected based on a simulated roulette wheel with slots sized according to the DFP costs encoded in the solutions [5]. Those that were not selected were culled.
6) The process was repeated until at least 80% of the stock has similar morphogenetic traits of the current best solution. We used the metric defined in Equation 13 to measure the similarities among solutions in the stock.

G. BREED: Finding Solutions Based on Line Breeding

We modified EVOLVE by implementing a selective line breeding scheme we termed as BREED. The algorithm runs as follows:

1) The solution which has the lowest DFP cost in the initial stock was selected as the first champion (geneticaly superior breeding animal);
2) The mating pool was created by selecting solutions from the stock using a simulated roulette wheel with slots sized according to DFP cost;
3) The champion mated everyone in the mating pool to generate the next generation using the mating and mutation algorithms discussed in section II-D.
4) If the new champion of the next generation is better than the current champion, the current champion was replaced by the new champion.
5) The process was repeated until at least 80% of the stock has similar morphogenetic traits as the champion.

III. Results and Discussion

The deterministic method for solving the MINLP using ADBASE reported that the minimum cost is at 215.86. ADBASE did not report the existence of other combinations of variables X that can produce the same minimal cost within the constraints. The solution by ADBASE shows that the reformulated MINLP for DFP is a unimodal search space and that there is only one global optimum. However, results from EVOLVE and BREED suggest that the nature of the MINLP search space is multi-modal and therefore multiple global optima exist.

We ran EVOLVE with a stock size of 100 solutions, mutation rate of 0.001, and a Hamming distance equivalent

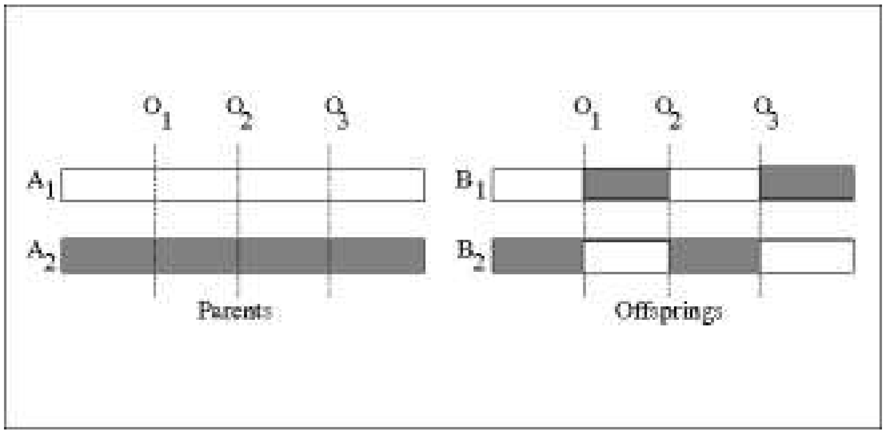

Fig. 2. Reproduction of solutions via mating.

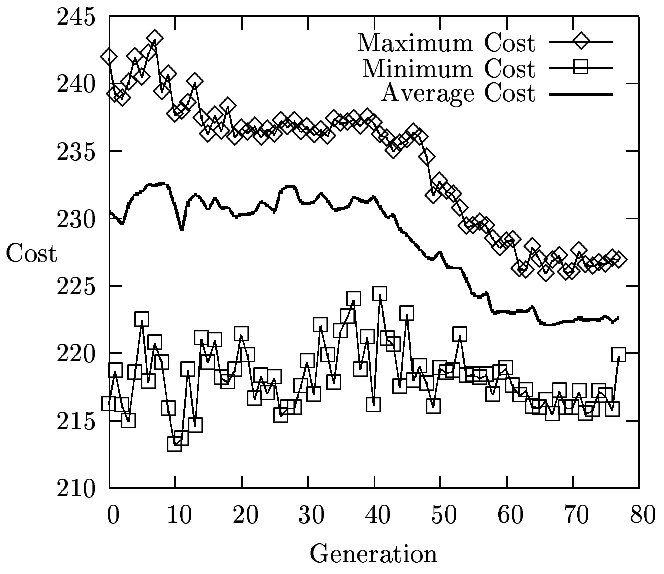

Fig. 3. Evolution of maximum, average, and minimum cost of DFP using EVOLVE.

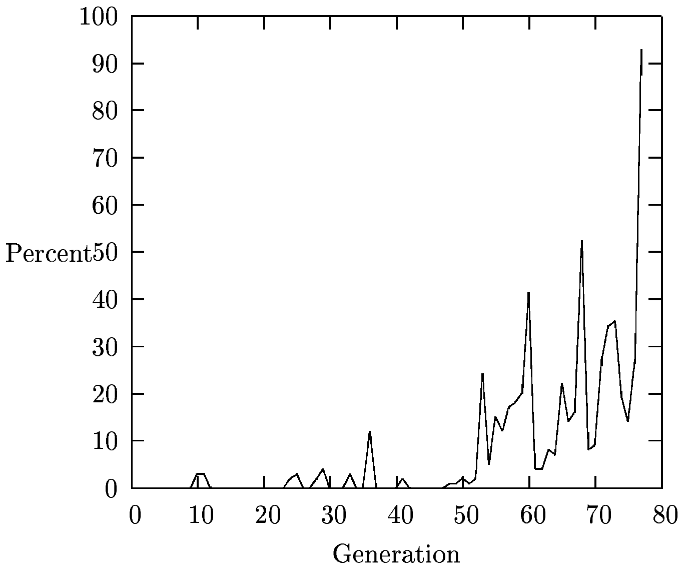

Fig. 4. The percentage of similarities of genetic meterials in the stock per generation based on Hamming distance of 10 percent of the chromosome length using EVOLVE.

to 10 percent of the chromosome length ($\mathcal{H} = 0.10 \times L$). The result of evolving the cost of the DFP is shown in Figure 3. Both the maximum cost and the average cost have evolved into lower costs over generations. It can be noted here that the minimum cost of 213.16 was achieved as early as the 10th generation and is lower than the minimum cost of 215.86 found by ADBASE. Figure 4 shows the similarities of the genetic code of the solutions in the stock per generation. The similarities suggest that there could be combinations of variables X that can produce the same minimum cost within the constraints.

We ran BREED with the same stock size (100 solutions), mutation rate (0.001) and similarity requirement (10 percent of the chromosome length) as EVOLVE. The result of selective line breeding the cost of the DFP is shown in Figure 5. The average cost has been bred into lower costs while the minimum cost of 188.46 was achieved right after the first generation. The minimum cost found by BREED is better than the minimum cost found by both ADBASE and EVOLVE while BREED found the minimum cost at a faster rate than that of EVOLVE. BREED was able to breed a better stock of solutions within fewer generations than that of EVOLVE. BREED was also able to generate similar genetic materials (champions) in the stock per generation within a shorter period of time (Figure 6).

The results show that BREED and EVOLVE are better optimization methods than that of the deterministic method in ADBASE for solving the DFP. Both BREED and EVOLVE were able to find other combinations of variables that can produce the same minimum cost for the DFP within the constraints. Comparison between the quality of the solutions found by heuristic methods shows that BREED is a much better optimization heuristic than EVOLVE. Thus, the algorithms in SIMBREED can be used to find solutions to complex combinatorial optimization problems including MINLP.

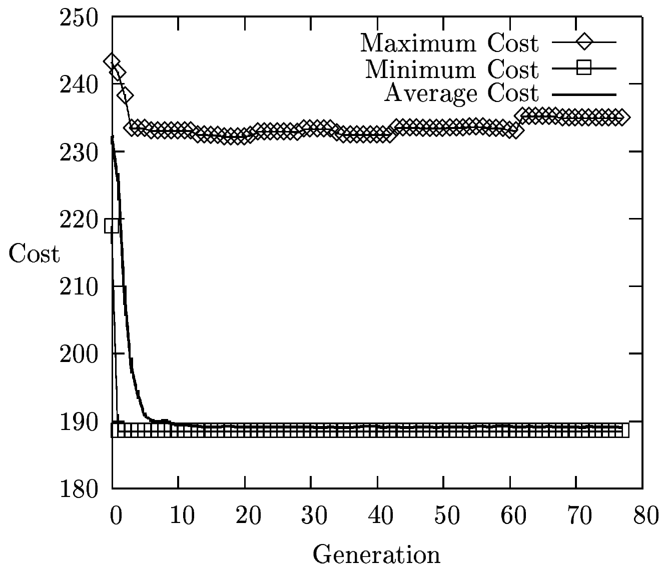

Fig. 5. Selective line-breeding of maximum, average, and minimum cost of DFP using BREED.

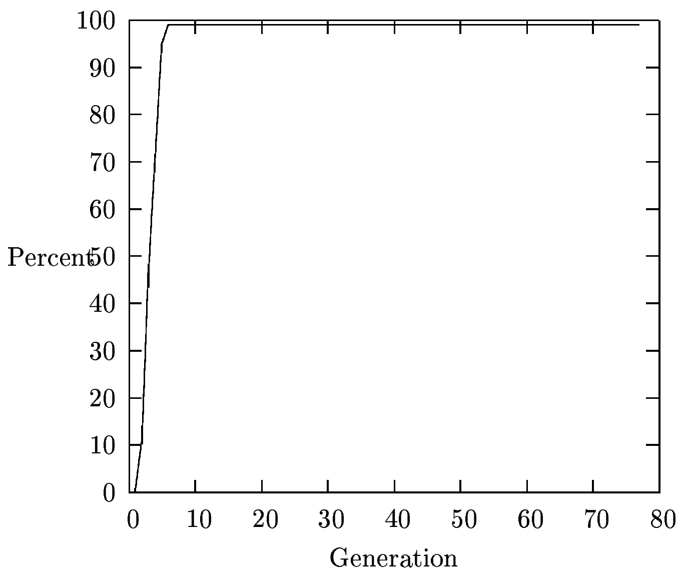

Fig. 6. The percentage of similarities of genetic meterials in the stock per generation based on Hamming distance of 10 percent of the chromosome length using BREED.

References

[1] J. M. Cadenas, D. A. Pelta, H. R. Pelta, and J. L. Verdegay. Application of fuzzy optimization to diet problems in Argentinean farms, 2002. (Submitted to European Journal of Operations Research).

[2] A. de Vries. Optimization of dairy heifer purchasing decisions under herd constraints with a genetic algorithm. *Journal of Dairy Science*, 85 (Suppl. 1):230, 2002. (Presented also as Abstract 917 during the ADSA/ASAS/PSA Annual Meeting, Quebec, Canada, July 22–28, 2002).

[3] G. Gallenti. The use of computer for the analysis of input demand in farm management: A multicriteria approach to the diet problem. In *First European Conference for Information Technology in Agriculture*, Copenhagen, Denmark, 1997.

[4] S. I. Gass. *Linear Programming: Methods and Applications*. McGraw–Hill, 1985.

[5] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison–Wesley, Boston, MA, 1989.

[6] R. Gum and G. Thompson. Least cost feed rations on your personal computer. *Ranch Business Management*, pages 69–72, 1994.

[7] C. R. Long. *Application of Mathematical Programming to the Evaluation of Systems of Beef Cattle Breeding*. PhD thesis, Texas A & M University, College Station, Texas, 1972.

[8] National Agricultural Library. Food and nutrition information center dietary reference intakes (DRI) and recommended dietary allowances (RDA). http://www.nal.usda.gov/fnic/index.html, 2001.

[9] Optimization Technology Center. http://www.ece.northwestern.edu/OTC/, 1994. Argonne National Laboratory and Northwestern University.

[10] J. P. Pabico, G. Hoogenboom, and R. W. McClendon. Determination of cultivar coeffi cients of crop models using a genetic algorithm: A conceptual framework. *Transactions of American Society of Agricultural Engineers*, 42(1):223–232, 1999. (Presented also as ASAE Paper no. 96–3101).

[11] W. B. Roush. Stochastic nonlinear programming: A new feed formulation. *Feedstuffs*, 65(50):14–18, 1993.

[12] W. B. Roush. Stochastic nonlinear programming: A new generation of feed formulation. In *55th Minnesota Nutrition Conference and Roche Technical Symposium*, Bloomington, MN, 1994.

[13] C. Rutcas. Pearson's square. http://www.ticalc.org/archives/fi les/fi leinfo/73/7330.html, 1998.

[14] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application*. John Wiley and Sons, 1986.

[15] R. E. Steuer. The Tchebycheff, aspiration criterion vector, and combined procedures of interactive multiple objective programming. *Journal of Automation and Information Sciences*, 26(6):27–37, 1993. (Originally published in *Avtomatika*, No. 6, 1993, pp. 44–63.).

[16] R. E. Steuer. *Manual for the ADBASE: Multiple Objective Mathematical Programing Package*, 1995.

[17] United States Department of Agriculture. USDA national nutrient database for standard reference, release 16. http://www.nal.usda.gov/fnic/foodcomp/, 2003.

[18] L. Wall, T. Christiansen, and R. L. Schwartz. *Programming Perl*. O'Reilly, 2nd edition, 1996.

Jaderick P. Pabico (Assistant Professor 4, Institute of Computer Science) received the B.S. and M.Sc. degrees in agricultural engineering from the University of the Philippines Los Baños in 1990 (minor in land and water resources engineering and technology) and the University of Georgia, Athens, GA in 1996 (specialization in applied computational intelligence), respectively. He earned his M.S. degree in computer science from University of the Philippines Los Baños in 1999 (minor in management science). He has developed several high–performance client/server–based information systems for the University of the Philippines Los Baños. His research interests include the application of biology– and nature–inspired heuristics for solving hard and complex optimization problems in agricultural, natural, and engineering sciences.

Orville L. Bondoc (Professor 1, Institute of Animal Science) received the B.S. and M.S. degrees in agriculture from the University of the Philippines Los Baños in 1983 and 1987, respectively. He obtained Ph.D. in animal breeding genetics (minor in quantitative genetics and statistics) from the University of Guelph, Ontario, Canada in 1991. His major research interests include the design of animal breeding strategies, livestock genetic evaluation, and conservation and use of animal genetic resources. He is also conferred the 2001 UPLB Outstanding Researcher Award, the NAST-DOST 1997 Outstanding Young Scientist and the 2002 Third World Academy of Sciences (TWAS) Prize for Young Scientists in the Philippines. Dr. Bondoc is currently the Associate Dean of the College of Agriculture.