Available online at www.isas.org.in/jisas
JOURNAL OF THE INDIAN SOCIETY OF AGRICULTURAL STATISTICS 67(1) 2013 13-22

Applications of Genetic Algorithms in Agricultural Problems — An Overview

Avnish Kumar Bhatia
National Bureau of Animal Genetic Resources, Karnal

Received 13 May 2011; Revised 4 September 2011; Accepted 15 December 2011

SUMMARY

Genetic algorithms (GA) are inspired by Darwin's theory of survival of the fittest in natural genetics. GA is an optimization technique that uses processes of evolutionary biology such as mutation, selection and crossover for artificial evolution towards global optimum in a number of iterations. Various problems in agriculture and livestock management are solved by formulation as optimization problems and hence are candidates for solving with genetic algorithm. Combination of machine learning techniques such as neural networks, fuzzy systems with genetic algorithms has wide applicability in precision farming and green house entailing accuracy of operations. This paper presents a survey of GA applications in solving agricultural problems.

Keywords : Genetic algorithm, Agricultural problems.

1. INTRODUCTION

Numerous problems in operations research and engineering design require optimization of an objective function (Deb 1995), which are modeled as mathematical program: Maximize/Minimize $F(X)$, $X = (x_1, x_2, ..., x_n)$ subject to $C_j(X)$, $j = 1, m$, where X is a vector of $n \geq 1$ model variables and $C_j(X)$ is a set of $m \geq 0$ constraints. A solution vector X^* that satisfies all the constraints of the problem is called a feasible solution. A feasible solution in the search space that maximizes or minimizes the objective function is the optimal solution.

Most of the real world optimization problems are NP-hard, indicating that there are no known polynomial time algorithms to solve these problems. Therefore, various heuristics have been designed for their solution, which may provide sub-optimal but acceptable solution in a reasonable computational time. Some of the problems don't have well defined objective function that needs to be devised. A number of meta-heuristics such as simulated annealing, evolutionary algorithms, and artificial neural networks derived from natural physical and biological phenomena have been evolved in a quest to reach near optimal solution in this type of problems, which do not have well defined search space.

Evolutionary Algorithms (EAs) are derived from the philosophy of survival of the fittest in natural selection. EAs perform artificial evolution in a computer to solve optimization problems by maintaining a population of individuals that are potential solutions of the problem under the environment of objective function. By manipulation of genetic structure of these individuals (the genotypes) through selection, crossover, mutation operators, EAs evolve progressively better phenotypes (the physical expression of a genotype). There are three major variants of EAs: evolution strategies, evolutionary programming and genetic algorithms. The most well known EA is the Genetic Algorithm (GA) (Mitchell 1996, Goldberg 1989), which is fairly representative of the other EAs.

GA's are designed to solve optimization problem. Alternately a problem can be solved with genetic algorithm if it can be formulated as an optimization

E-mail address : avnish@lycos.com

problem. Traditional algorithms such as direct-search and gradient-based methods (Deb 1995) to solve optimization problems get trapped in local optimum in non-linear and multimodal function landscapes. Genetic algorithms are directed random procedures that jump over peaks in difficult search spaces and march towards global optimum. GA is also used in combination with other learning algorithms to form hybrid algorithms, for example, to evolve weights in artificial neural networks. Agricultural problem solving involves optimization problem formulation in a number of cases. Some examples of this type of agricultural problems are animal feed formulation, optimization of parameters of crop growth models, automation of green houses, irrigation management, land allocation, farm layout, etc. GA's have performed better than traditional methods for solving agricultural problems in a number of cases (Hart *et al*. 1998, Iquebal *et al*. 2010).

There are numerous applications of genetic algorithms in agricultural problem solving. Recent reviews list and describe a few applications. Hashimoto (1997) explores potential of genetic algorithms application in ill-defined and complex agricultural systems. Huang *et al*. (2010) list applications of GA in the field of agricultural engineering. Bolboacã *et al*. (2010) list application of genetic algorithm in the fields of bioinformatics and agro-economic system.

This paper describes implementation steps for genetic algorithm and lists representative applications of genetic algorithm in agricultural problem solving in four categories: agriculture, agricultural engineering, livestock production, and fisheries.

2. GENETIC ALGORITHMS

Genetic algorithms (Goldberg 1989, Mitchell 1996) were invented by John Holland in the sixties and early seventies. Originally genetic algorithms were applied to optimize continuous function landscapes. Now these are being used to solve combinatorial and real world problems with integer and other types of variables.

2.1 Implementation of Genetic Algorithm

Implementation of genetic algorithm starts with encoding of variables followed by random creation of an initial population of potential solutions (termed as individuals or chromosomes or strings). Individuals in the population are evaluated using an objective function. This is followed by selection of individuals (parents) on the basis of fitness values, and reproduction among the selected parents using crossover and mutation operators to create offsprings for next generation. This completes one generation. Over a number of generations, GA marches towards global optimum. Fig. 1 displays pseudo-code of a genetic algorithm. A genetic algorithm is made up of the five major steps described below.

```
begin GA
  g:=0 {generation counter}
  Initialize population P(g)
  Evaluate population P(g)
  while not stopping-criteria
     g:=g+1
     Select P(g) from P(g-1)
     Crossover P(g)
     Mutate P(g)
     Evaluate P(g)
  end while
end GA
```

Fig. 1. Pseudo-code of a standard genetic algorithm.

Step 1: Randomly Create an Initial Population

Genetic algorithm starts with randomly created initial population of encoded strings of variables. The potential solution of the candidate problem is termed as chromosomes and variables are termed as genes in line with natural genetics. Traditionally, binary numbers have been used for coding of chromosomes/string representation. Other types of variables, for example real numbers, are mapped to binary strings of a given size. Binary coding has been found unsuitable for representing chromosomes according to the structure of several problems. Therefore, other coding schemes have also been utilized. Real number coding has been used for continuous function optimization. Order-based permutation coding has been used for scheduling problems. Tree based coding have been used for genetic programming and computer games. Chromosome representation according to problem structure improves performance of GA. Real-coded GA have been demonstrated to perform much better than binary-coded GA (Michalewicz 1994) because of required precision and non-formation of hamming-cliff where the binary coding of two successive function values differs in each bit.

Step 2: Calculation of Fitness Values

Fitness value of each individual in the population is calculated using a fitness function derived from objective function of the optimization problem. Many of the real world problems may not have a well defined objective function and require the user to define a fitness function.

Step 3: Selection of Individuals to Create Next Generation

Highly fit individuals in a population are selected using a selection method, which selects prospective parents from the population on the basis of their fitness values.

Selection scheme in a GA is implemented as in natural selection. To bias the selection toward more fit individuals, each individual is assigned a probability of selection $P(x)$ as the fitness of individual x relative to rest of the population. Fitness-proportionate selection is the most commonly used selection method. Given that f_i as the fitness of i^{th} individual, $P(x)$ in this method is calculated as $P(x) = \dfrac{f_x}{\sum f_i}$.

Fitness-proportionate selection ensures that the top performing individuals are given higher opportunity to spread their genes through the new population. After assignation of the expected values $P(x)$, the individuals are selected using roulette wheel sampling that works in the following steps.

- Let C be the sum of expected values of individuals in the population.
- Repeat the following two or more times to select the respective number of parents for mating.
  - (i) Obtain a uniform random integer r in the interval $[1, C]$.
  - (ii) Loop through the individuals in the population, summing the expected values until the sum is greater than or equal to r. The individual index where the sum crosses this limit is selected.

In stochastic remainder sampling, expected value is calculated as f_i/f where f is the average fitness of current population. It assigns parents deterministically from the integer part of each individual's expected value, and then uses roulette wheel selection on the remaining fractional part to complete the population size.

An individual may be selected several times. Some of the relatively unfit individuals may also be selected due to inherent randomness of this process. Fitness-proportionate selection exhibits high selection pressure in the beginning of GA execution because of high diversity in the population. Selection pressure in later generations reaches near zero as all the individuals possess similar fitness values and it slows the evolution in the GA. Therefore, other selection strategies such as tournament selection, rank selection are used to avoid this selection biasness (Mitchell 1996). Tournament selection compares two or more individuals and selects the better one with a pre-specified probability. Rank selection selects individuals on the basis of ranking according to increasing fitness values. Tournament and rank selection methods maintain uniform selection pressure throughout the GA execution and avoid premature convergence unlike fitness-proportionate selection. Details on selection methods in genetic algorithms can be found in a recent survey by Sivaraj and Ravichandran (2011).

Step 4: Apply Genetic Operators

Strings in the initial population and populations at subsequent generations are perturbed by use of genetic operators. Crossover and mutation are fundamental genetic operators in genetic algorithms. Role of genetic operators is to explore new areas in the search space.

Generally, two parents are selected at a time and are used to create two new children for the next generation using crossover operator with a pre-specified probability of crossover. Crossover is a recombination operator that interchanges part of participating chromosomes. It preserves good genes in the population. Single-point crossover is the most common, which works by marking a random crossover spot and exchanging the genetic material on the right of the spot as shown in Fig. 2.

```
0 1 0 1 0 1 0 1 | 0 1 0 0            0 1 0 1 0 1 0 1 1 1 0 1
                              ==>
0 1 1 1 0 1 0 1 | 1 1 0 1            0 1 1 1 0 1 0 1 0 1 0 0
```

Fig. 2. Implementation of crossover operator

There are other crossover operators such as double point crossover and uniform crossover. Specialized crossover operators have been devised for codings other than binary.

Crossover operator creates two offsprings that may be subjected to mutation by mutation operator, which changes the value of each gene in a chromosome with a pre-specified probability of mutation. Mutation operator is said to maintain diversity in the population by introducing new genes, which prevents the GA to converge towards a suboptimum solution. In an example of implementation of mutation operator in Fig. 3, the fifth bit (underlined) has been mutated in the individual.

0 1 1 1 0 1 0 1 0 1 1 1 ⟶ 0 1 1 1 1 1 0 1 0 1 1 1

Fig. 3. Implementation of mutation operator

Non-uniform mutation in real-coded GA (Michalewicz 1994), and specialized mutation in multi-chromosomal representation for solving bin-packing problem (Bhatia and Basu 2004) are examples of implementation of mutation operator in other codings.

Step 5: Test the Stopping Condition

Step numbers 2, 3 and 4 complete one generation in genetic algorithm. A GA is executed for a number of generations till a stopping criterion is satisfied that may be defined in many ways. Pre-fixed number of generations is the most used stopping criterion. Other stopping criteria used are the desired quality of solution, number of generations without any improvement in the result, etc.

A standard genetic algorithm utilizes three operators: selection, crossover and mutation. Many extensions such as elitism and niching have been added to genetic algorithms to avoid pre-mature convergence. Elitism ensures that the best individual is passed on unperturbed to the next generation. It saves high performing individuals from getting perturbed by genetic operators. Niching (Shir *et al*. 2010) performs a task similar to elitism by special treatment to high performing individuals in multi-modal optimization.

2.2 Genetic Parameters

Values of genetic parameters such as population size, crossover probability, mutation probability, total number of generations affect convergence properties of the genetic algorithms. Values of these parameters are generally decided before start of GA execution on the basis of previous experience. Experimental studies recommend the values of genetic parameters as: population size 20-30, crossover rate 0.75-0.95, and mutation rate 0.005-0.01.

Genetic parameters may also be fixed by tuning in trial runs before the actual run of GA. Deterministic control and adaptation of the parameter values to a particular application have also been used (Eiben *et al*. 1999). In deterministic control, value of a genetic parameter is altered by some deterministic rule during GA execution. Adaptation of parameters allows change in parameter values on the basis of previous performance. In self-adaptation, the operator settings are encoded into each individual.

2.3 Constraint Handling in GA

Chromosomes in the initial population are generated randomly. Also, the genetic operators such as crossover and mutation alter composition of chromosomes in the population. Initialized and altered chromosomes may violate one or more constraints in the constrained problem and thus represent infeasible solutions. Several methods have been used with GAs to treat infeasibility. Use of penalty functions is the most common method where a penalty term is added (subtracted) from the fitness function of minimization (maximization) problem. It avoids selection of infeasible chromosomes, as selection in genetic algorithm is fitness-biased. Other methods to handle constraints in continuous optimization problems include use of specialized operators for maintaining feasibility of solutions, behavioral memory emphasizing distinction between feasible and infeasible solutions and homomorphous mapping of the search space via decoders (Koziel and Michalewicz 1999).

2.4 Hybrid Genetic Algorithms

Genetic algorithm alone may not provide acceptable solution for many hard problems. There are general characteristics of GA that limit its effectiveness. The fundamental genetic operators may not provide optimum solution in all applications. GA requires extensive experimentation for finding appropriate values of genetic parameters that requires time and computing resources. So it is necessary to hybridize problem-specific domain knowledge into GA's to ward off randomness in the procedure to obtain acceptable solution in a reasonable time.

Hybridization of genetic algorithms with other problem-related heuristics is implemented in two ways. (i) Action of genetic operators involves a local optimizer in the form of hill climber applied to each individual before it is passed to the population at subsequent generation. This form of hybridization has been called memetic algorithms (Radcliffe and Surry 1994). (ii) Initial population generation and implementation of genetic operators involve problem related heuristics (Grefenstette 1987). Genetic algorithms augmented with problem related heuristics have often been found to perform better than the genetic algorithms alone. El-Mihoub *et al*. (2006) provide a detailed survey of hybrid genetic algorithms.

2.5 Multi-objective Genetic Algorithm

Multiple objectives in a problem have a set of optimal solutions known as Pareto-optimal solutions. One of these solutions cannot be said to be better than the other and demands to find many solutions. GA's are well suited to solve multi-objective optimization problems due to global search abilities (Deb *et al*. 2002, Konak *et al*. 2006). The ability of GA to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems with non-convex, discontinuous, and multi-modal solutions spaces. Crossover operator of GA may exploit structures of good solutions with respect to different objectives to create new non-dominated solutions in unexplored parts of the Pareto front. In addition, most multi-objective genetic algorithms do not require the user to prioritize, scale, or weigh objectives. Therefore, GA's have been the most popular heuristic approach to multi-objective design and optimization problems.

3. APPLICATIONS OF GENETIC ALGORITHMS

Genetic algorithms have been applied to solve hard problems in diverse fields. Some of example applications can be listed as Automatic programming: to evolve computer programs; Machine learning: classification and prediction tasks, feature selection; Economics: bidding strategies, emergence of economic markets; Ecology: host-parasite co-evolution, resource flow; Evolution and learning: how individual learning and species evolution affect each other; Social systems: the evolution of cooperation, communication in multi-agent systems; Bioinformatics: protein structure prediction, DNA sequencing, multiple sequence analysis, gene expression analysis. Applications of GA's in various fields are available in recent reviews. Kumar *et al*. (2010) list applications in image processing, sensor-based robot path planning, gaming, real-time systems, job shop scheduling. Wang and Zhang (2010) provide applications in bioinformatics problems such as sequence analysis, protein structure prediction, protein-protein recognition and docking.

There are numerous applications of genetic algorithms in agricultural problem solving including management of agricultural resources. Some of representative applications are described in the following subsections grouped as applications in agriculture, agricultural engineering, livestock production and fisheries.

3.1 Applications in Agriculture

This subsection explores applications of genetic algorithms in crop production, post-harvest processing and storage.

Morimoto *et al*. (1997) utilize GA along with neural networks in a technique for optimal control of fruit-storage process with objective function as reciprocal number of sum of average values in the parameters – water loss and development of legion by fungi. A three-layer neural network has been used for identifying a multi-input (relative humidity and days after storage), multi-output (water loss and fungal development) nonlinear system. Optimal set-points of relative humidity that maximize the objective function were found using binary-coded genetic algorithm among the numerous values of objective function obtained from simulation. Genetic algorithm was fast and successful in searching the optimal set-points.

Loonen *et al*. (2006) utilize genetic algorithm to determine optimal spatial distribution of agricultural ammonia emissions to minimize atmospheric nitrogen deposition in nature reserves. Results obtained with GA for the linear atmospheric emission–deposition process have been compared with results of linear programming (LP) by analyzing (a) the spatial distribution of optimized emissions and (b) the spatial distribution of resulting increase of the critical N-loads in nature cells. The performance of GA was similar to that of LP procedure.

Dai *et al*. (2009) apply genetic algorithm in parameter optimization to calibrate the growth model of greenhouse crop. An adaptive GA is proposed and evaluated that is composed of two GAs. The primary one is utilized to parameterize the growth model and the secondary GA determines genetic parameters of the primary GA. The procedure demonstrates superior performance when applied to three test functions and greenhouse optimization problems compared with other two existing genetic algorithms.

Sheikh and Lanjewar (2010) design a decision support system to optimize the cotton bales blending/ mixing so as to reduce the cost of overall cotton cost subject to quality constraint. Optimization in the decision support system is performed with genetic algorithm. The problem is formulated to minimize cost of cotton bales mixing subject to quality constraints. GA reduces the cost without affecting quality and its performance is better than existing mathematical techniques.

Iquebal *et al*. (2010) demonstrate superiority of genetic algorithm over search algorithm for fitting of self-exciting threshold autoregressive non-linear time-series model to India's lac export data. A real-coded genetic algorithm with simulated binary crossover operator is implemented on the multi-modal function. Genetic algorithm can find global optimum value of threshold while search algorithm determines final value of threshold only from discrete potential values.

3.2 Applications in Agricultural Engineering

Noguchi and Terao (1997) develop a GA based method to create a path of an agricultural mobile robot. They apply a control technique combining a neural network (NN) and a genetic algorithm (GA). NN is applied to describe motion of the robot as a nonlinear system. GA is used to optimize a path using a simulator described by NN. GA optimized time series of steer angles as control input and created an optimal work path of the mobile robot. The time series of steer angle changes of mobile robot was encoded as an integer in a range of variables.

Ines *et al*. (2006) present an approach to explore water management options in irrigated agriculture considering constraints of water availability and heterogeneity of irrigation system properties. They set up a soil–water–atmosphere–plant model in a deterministic–stochastic mode for regional modeling. The distributed data - sowing dates, irrigation practices, soil properties, depth to groundwater and water quality required as inputs for modeling were estimated by minimizing the residuals between distributions of field-scale evapo-transpiration simulated by regional application of the model, and by surface energy balance algorithm for land using two Landsat7 ETM+ images. Derived distributed data were used as inputs in exploring water management options. Genetic algorithm was used in data assimilation and water management optimizations. Objective function consisted of maximization of regional yield subject to constraints of water availability, water management practices, and crop management practices. Results showed that under limited water condition, regional wheat yield could improve further if water and crop management practices are considered simultaneously.

Zhu and Eisaka (2009) propose design methods to improve an existing sugar beet topper to improve efficiency of harvest. Two redesign approaches have been proposed. The first modifies relevant structural parameters of the existing machine by numerical optimization. GA optimizes the tunable topper unit parameters. The other approach is appending an actuator and a controller to a machine and then employing simultaneous optimization of both controller and machine parameters. GA optimizes the tunable extended active topper parameter unit. Both results satisfy the design specification.

Annepu *et al*. (2011) develop different agriculture strategies for land allocation to different crops. Objective functions are formulated as maximization of net profit, production of crops and minimization of fertilizer consumption with availability of cultivable land, agriculture labour, agriculture machinery and water as constraints. A case study of Visakhapatnam district, Andhra Pradesh, India has been solved through genetic algorithm. The multi-objective problem has been converted to a single-objective problem. A real-coded genetic algorithm has been used that implements tournament selection. Penalty term has been used to manage the constrained optimization. The model can help in reorganizing cultivated land to get maximum satisfaction of the stakeholders for sustainable development in agriculture.

3.3 Applications in Livestock Production

Tan *et al*. (1996) use genetic algorithm to solve the frequency domain system linearization problem to design a control system for nonlinear engineering system such as chemical or dairy plant. Non-differentiable and multi-input, multi-output systems are difficult to solve using the traditional calculus Taylor expansion around an equilibrium operating point. The problem has been formulated as minimization of linearization error and solved with a genetic algorithm. The method with GA utilizes plant input/output data directly and requires no derivatives. It allows linearization of an entire operating region and for the interested frequency range, the benefit of which cannot be matched by existing methods.

Hayes *et al*. (1997) assess efficiency of selecting mates using a genetic algorithm. Mate selection is of potential value in increasing progeny merit in animal breeding. The GA found the optimal solution in every case, and efficiency of the GA increased with increasing mating set sizes. GA is a valuable tool in solving mate selection problems which include issues such as connection between herds, parameter estimation and inbreeding in future generations.

Hart *et al*. (1998) optimize management variables in a simulation model of dairy farm economics based on factors such as feed, stocking rate and milk prices. Milk production being the main economic output of a farm, farm fitness has been defined as the weight of milk-fat produced over the entire milking season. Binary coding of input variables has been used. A chromosome in the GA that violates one or more constraints is penalized and assigned a very low fitness value. Ten farm optimizations are executed over a period of twelve month season. Results of genetic algorithm are comparable with other hill climbing algorithms.

Takaaki *et al*. (1999) develop a multi-objective design method of livestock feed formulations using a genetic algorithm with simultaneous optimum design of three kinds of feed - poultry, swine and cattle. Twelve different rations of feed ingredients such as corn, grain sorghum, defatted rice bran, and others were designed under constraint conditions of specified nutrient contents. Genetic algorithm was used to minimize the raw material cost of the three feed formulations. Simulations were carried out for different feed formulation using twelve raw materials under three conditions. The total volume of specified raw materials such as corn and/or corn gluten feed and their usable volumes for three types of feed were restricted. Also, the total volume of specified raw materials such as corn and/or corn gluten feed was restricted but their usable volumes for three types of feed were not restricted. The multi-objective genetic algorithm attains the simultaneous optimum design of three kinds of feed formulation under various conditions.

Kenji (2002) utilize genetic algorithm to optimize mating design. Genetic relationships among carcass traits and between reproductive or growth traits and carcass traits have been estimated for Japanese Black cattle. These estimates indicated that truncation selection by daily gain would remove superior young bulls in marbling from selection candidates and selection by marbling would lead calving interval longer. Hence, selection criteria, such as estimated or predicted breeding values, of beef marbling and calving interval should be introduced into appropriate selection stages of breeding programs. The mating designed by genetic algorithm succeeded to reduce 8.1% to 14.8% of inbreeding level for various heritabilities compared with random mating after nine generations of selection and mating.

Pérez *et al*. (2004) present a methodology based on genetic algorithm for the generation of optimal layouts in milk goats' units. Systematic Layout Planning methodology developed for the planning of industrial facilities has been used, and a computer program for layout generation using genetic algorithms and on slicing-tree techniques is employed. The procedure consists of building an initial population of solutions by means of a recursive process of location domain cuts; so that each solution obtained can be represented by a slicing tree and the initial population of solutions sequence can be coded by the application of genetic operators. Optimization involves an objective multi-criteria function that considers parameters of a qualitative, quantitative and geometric character. The methodology has been applied to two type of farms that present a semi-intensive, free housing production system in south of Spain, with 120 and 240 milk goats. The design results in good layouts that minimize cost of the flow of materials through the farm in terms of saving both in costs and in labour, and improvement in the welfare of the animals.

Bryant *et al*. (2007) utilize a genetic algorithm to a mechanistic model of the mammary gland to find parameter values that minimizes the difference between predicted and actual lactation curves of milk yields in New Zealand Jersey cattle managed at different feeding levels. The effect of feeding level, genetic merit, body condition score at parturition and age on total lactation yields of milk, fat and protein, days in milk, live weight and evolutionary algorithm derived mammary gland parameters was then determined using a multiple regression model. The mechanistic model of the mammary gland was able to fit lactation curves that corresponded to actual lactation curves with a high degree of accuracy.

3.4 Applications in Fisheries

Truong *et al*. (2005) develop a decision support system augmented with genetic algorithm for fisheries policy and management decisions, which is applied to the real situation in the Northeastern U.S. A simulation optimization model assists authorities in scheduling for a fleet of hundreds of vessels in terms of time and location of fishing, as well as amount and target species to be fished. Simulation-based optimization utilizes the simulation model in obtaining the objective function values of a particular fishing schedule. A genetic algorithm is used as the optimization routine to determine the optimal fishing schedule, subject to fleet capacity and conservation requirements.

Komeyama *et al*. (2008) utilize a genetic algorithm to analyse occurrence of one-tagged fish. They monitor occurrence of common carp, *Cyprinus carpio L*. and environmental conditions near a set-net using acoustic telemetry and a data logger to elucidate the conditions under which carp approaches the net. The conditions near the set-net were simulated, which included the current profile; wind system, water temperature and rainfall. GA assessed the pattern of factors that influenced the occurrence of tagged fish. It selected a significant portion of the truly important factors.

Sathianandan and Jayasankar (2009) develop a genetic algorithm for simulation of trawl net and ring seine fishery using surplus production model and spectral methods. Basic surplus production model is used for calculation of biomass, fishing mortality and yield in the simulation with parameters - the initial biomass, carrying capacity, intrinsic growth rate and catchability coefficient. A genetic algorithm was designed for estimation of these parameters using time series data on catch and effort of mechanized trawl net and outboard ring seine in Kerala, India during 1985-2004. Simulations were carried out for six different levels of exploitations and the average biomass and average yield were calculated and compared with the maximum sustainable yield.

4. CONCLUSION

Genetic algorithms are optimizers derived from the phenomenon of selection of the fittest in natural genetics. It performs artificial evolution within the search space of optimization problems to march towards the global optimum. Genetic algorithms are included in the category of black box methods, which can be applied to search spaces with unknown, complex landscapes exhibited by nonlinearity and multi-modality. Some of the agricultural problems such as livestock feed mixing are optimization problems with defined objective function. A large number of problems can be formulated as optimization problems with or without constraints. Many of the real world problems including agricultural problems have multi-modal objective functions with difficult and even unknown search spaces. These are generally hard problems and there is either no available method to solve them or available hill climbing methods may provide local optimum. Agricultural problems include optimum use of fertilizers, irrigation, land, labour, animal and poultry feed, etc, which are candidates for solution with genetic algorithm. Simulation modeling is natural procedure to study complexities of agricultural systems such as crop production, livestock management, fisheries management, irrigation scheduling, etc. Optimization of parameters in a simulation model has been commonly performed with genetic algorithms. Genetic algorithms have wide applications in optimizing parameters of other soft computing techniques such as artificial neural networks and fuzzy systems utilized to solve agricultural problems. There is wide scope of application of GA combined with these machine learning approaches in precision farming and green houses where real-time accuracy of agricultural operations is key to success due to non-stationery system environment.

REFERENCES

Annepu, G., Subbaiah, K.V. and Kandukuri, N.R. (2011). Land allocation strategies through genetic algorithm approach – a case study. *Global J. Res. Engg.*, **11(4)**, 7-13.

Bhatia, A.K. and Basu, S.K. (2004). Packing bins using multi-chromosomal genetic representation and better-fit heuristic. Lecture Notes in Computer Science No. 3316, 181-186, Springer-Verlag.

Bolboacã, S.D., Jäntschi, L., Bãlan, M.C., Diudea, M.V. and Sestras, R.E. (2010). State of art in genetic algorithms for agricultural systems. *Notulae Botanicae Horti Agrobotanici Cluj-Napoca*, **38(3)**, 51-63.

Bryant, J.R., Lopez-Villalobos, N., Holmes, C.W., Pryce, J.E., Pitman, G.D. and Davis, S.R. (2007). The effect of level of feeding, genetic merit, body condition score and age on biological parameters of a mammary gland model. *Animal*, **1**, 175-183.

Dai, C., Yao, M., Xie, Z., Chen, C. and Liu, J. (2009). Parameter optimization for growth model of greenhouse crop using genetic algorithms. *Appl. Soft Computing*, **9(1),** 13-19.

Deb, K. (1995). *Optimization for Engineering Design-Algorithms and Examples*. Prentice Hall of India Pvt. Ltd., New Delhi.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, **6(2)**, 182-197.

Eiben, A.E. Hinterding, R. and Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Trans. Evol. Comput.*, **3(2)**, 124-141.

El-Mihoub, T.A. Hopgood, A.A. Nolle, L. and Battersby, A. (2006). Hybrid genetic algorithms: a review. *Engineering Letters*, 13:2.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley.

Grefenstette, J.J. (1987). Incorporating problem specific knowledge into genetic algorithms. In: Davis, L., editor, *Genetic Algorithms and Simulated Annealing*. Morgan Kaufmann Publishers, California.

Hart, R.P.S., Larcombe, M.T., Sherlock, R.A. and Smith, L.A. (1998). Optimization techniques for a computer simulation of a pastoral dairy farm. *Comput. Elect. Agric.*, **19**, 129-153.

Hashimoto, Y. (1997). Applications of artificial neural networks and genetic algorithms to agricultural systems. *Comput. Elect. Agric.*, **18**, 71-72.

Hayes, B.J., Shepherd R.K. and Newman, S. (1997). Selecting mating pairs with genetic algorithms. In: *Proceedings of the Association for the Advancement of Animal Breeding and Genetics*, **12**, 108-112.

Huang, Y., Lan, Y., Thamson, S.J., Fang, A. Hoffmann, W.C. and Lacey, R.E. (2010). Development of soft computing and applications in agricultural and biological engineering. *Comput. Elect. Agric.*, **71**, 107-127.

Iquebal, M.A., Gosh, H. and Prajneshu (2010). Application of genetic algorithm for fitting self-exciting threshold autoregressive nonlinear time-series model. *J. Ind. Soc. Agril. Statist.*, **64(3)**, 391-398.

Ines, A.V.M., Honda, K., Gupta, A.D., Droogers, P. and Clemente, R.S. (2006). Combining remote sensing-simulation modeling and genetic algorithm optimization to explore water management options in irrigated agriculture. *Agric. Water Manage.*, **83**, 221-232.

Kenji, O. (2002). Genetic improvement of Japanese Black Cattle based on predicted breeding values. *Science Reports of Faculty of Agriculture,* Kobe University, **26**, 49-54.

Komeyama, Suzuki, K., Takagi, T., Mitsunaga, Y., Yamane, T. and Hiraishi, T. (2008). Analysis of fish movement using a genetic algorithm: factors influencing the occurrence of fish near a set-net. *Fish. Manage. Ecol.*, **15(5-6),** 347-355.

Konak, A., Coit, D.W. and Smith, A.E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engg. Sys. Safety*, **91,** 992-1007.

Koziel, S. and Michalewicz, Z. (1999). Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evol. Comput.*, **7(1)**, 19-44.

Kumar, M., Husian, M. Upreti, N. and Gupta, D. (2010). Genetic algorithm: review and application. International *J. Inf. Tech. Knowledge Manage.*, **2(2)**, 451-454.

Loonen, W., Heuberger, P.S.C., Bakema, A.H. and Schot, P. (2006). Application of a genetic algorithm to minimize agricultural nitrogen deposition in nature reserves. *Agric. Sys.*, **88**, 360-375.

Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, second edition.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms.* MIT Press, MA.

Morimoto, T., Baerdemaeker, J. De and Hashimoto, Y. (1997). An intelligent approach for optimal control of fruit-storage process using neural networks and genetic algorithms. *Comput. Elect. Agric.*, **18,** 205-224.

Mukai, O.K. (1998). Determination of the optimum mating design with constraints on inbreeding level and mating frequency of sires via a simple genetic algorithm. *Anim. Sci. Tech.*, **69(4)**, 333-340.

Noguchi, N. and Terao, H. (1997). Path planning of an agricultural mobile robot by neural network and genetic algorithm. *Comput. Elect. Agric.*, **18**, 187-204.

Pérez, J., Santamarina, M., Vallés, J., Peña, A., Valera, D. and Carreño, A. (2004). Optimal layout design for milk goats livestock farms using genetic algorithms. *Agric. Engg. Intern.*, **VI**, 1-17.

Radcliffe, N.J. and Surry, P.D. (1994). Formal Memetic Algorithms. *Lecture Notes in Computer Science,* **865**, 1-16, Springer-Verlag.

Sathianandan, T.V. and Jayasankar, J. (2009). Managing marine fishery in Kerala through simulation using surplus production model, genetic algorithm and spectral methods. *Ind. Jour. Fisheries*, **56(3)**, 161-168.

Sheikh, R.A. and Lanjewar, U.A. (2010). Decision support system for cotton bales blending using genetic algorithm. *Inter. J. Comp. Appl.*, **8(1)**, 28-34.

Shir, O.M., Emmerich, M. and ack, T.B. (2010). Adaptive niche radii and niche shapes approaches for niching with the CMA-ES. *Evol. Comput.*, **18(1)**, 97-126.

Sivaraj, R. and Ravichandran, T. (2011). A review of selection methods in genetic algorithm. *Inter. J. Engin. Sci. Tech.*, **3(5),** 3792-3797.

Tan, K.C., Gong, M.R. and Li, Y. (1996). Evolutionary linearisation in the frequency domain. *Electronics Letters,* **32(1),** 74-76.

Takaaki, S., Tatsumi, F. and Yoshiyuki, M. (1999). Multiobjective design of livestock feed formulation using a genetic algorithm. *J. Soc. Agril. Struc.,* Japan, **30(2),** 163-171.

Truong, T.H., Rothschild, B.J. and Azadivar, F. (2005). Decision support system for fisheries management. In: *Proceedings of IEEE Winter Simulation Conference,* 2107-2111.

Wang, B. and Zhang, X. (2010). Evolutionary computation applications in current bioinformatics. In: *New Achievements in Evolutionary Computation,* Peter Korosec (eds), INTECH, Croatia, 173-180.

Zhu, Z. and Eisaka, T. (2009). Practical approaches for the design of an agricultural machine. International *J. Mathe. Models Methods App. Sci.*, **3(3),** 291-298.