Agrotechnology
ISSN: 2168-9881

Adewumi et al., Agrotechnol 2015, 5:1
DOI: 10.4172/2168-9881.1000138

Research Article Open Access

Genetic Algorithm: A Veritable Tool for Solving Agricultural Extension Agents Travelling Problem

Adewumi IO*, Oluwatoyinbo FI, Omoyajowo AO, Ajisegiri GO and Akinsete AE

Department of Agricultural Engineering, Federal College of Agriculture, P.M.B. 5029, Moor Plantation, Ibadan, Oyo State Nigeria

Abstract

Genetic algorithm simulates the logic of Darwinian selection as observed in the biological evolutionary process (Cells' division, DNA, Mutation, etc.) to solve problems. They are based on one hand on a heuristic gradient ascension method (selection and crossover) and in another hand on a semi-random exploration method (Mutation). In this research work, application of genetic algorithms was explored for the optimization problem embodied in the transit problem of agricultural extension agents or workers in disseminating new innovation and technological advancement in agriculture. An order representation for the cost matrix for 10 cities and chromosomes was used. The result revealed that genetic algorithm can solve the routing problem of an agricultural extension agent in terms of time minimization in order to search for the shortest route, which will increase number of places that the extension agents can touch at reduced cost of transportation. This will help in achieving the nations' vision 2020 on food security.

Keywords: Genetic algorithm; Crossover; Mutation; Chromosome; Selection

Introduction

Sebusang [1] Odoemelam [2] that the crucial role of Information and Communication Technologies cannot be overemphasized in todays' world globalization, perceived as a major driver of the worlds' economies, is powered by an enhanced use of ICTs on which virtually every sector now relies. The current dispensation of competiveness within the global market space could not have been made possible without the interconnectedness of institutions and people. But many factors have continued to impinge on the increase drive towards the use of ICTs in developing economies particularly in Nigeria. Adewumi [3] has stated that the need to solve optimization problems is a dominant theme in the engineering world. A great number of analytic and numerical optimization techniques have been developed, and yet there are still large groups of functions that present significant difficulties for numerical techniques, and moreover, for analytical methods these are not continuous or differentiable everywhere; functions, which are non-convex, multi-modal, and functions which contain noise (Table 1). As a consequence, there is a continuing search for more robust optimization techniques that are able to overcome such problems. Oliver [4] has described Genetic Algorithm (GA) has one of the relatively new class of stochastic search algorithms. Stochastic algorithms are those that use probability to help their search. As the name implies, GA behaves much like biological genetics. It encodes information into strings just as living organisms encode characteristics into strands of DNA. Adewumi [3] stated that a string in a GA is analogous to a chromosome in biology. A population of strings competes and those strings that are the fittest procreate, the rest eventually die off childless. As with biological parents, two strings combine and contribute part of their characteristics to create their offspring, the new individual. This new string joins the population and fight to produce the next generation. If both parents contribute good building blocks (short section of the string) to the offspring, it will be fit and will procreate in its turn. If the building blocks are poor, then the offspring will die off without generating offspring (Table 2). A second-but-important process occurs in Gas: sometimes vary rarely. A mutation occurs and the offspring will incorporate a new building block that came from neither parent. The above cycle of death and birth repeats until an acceptable solution to the problem is found. Goldberg [5] as reported by Prasanna [6] defines genetic algorithm (GA) as search algorithm based on mechanics of natural selection and natural genetics. Frick [7] gave a similar definition stating that GA is a software procedure modeled after genetics and evolution. Genetic algorithm is an adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetics. The basic concept of GA is designed to stimulate processes in natural system necessary for evolution, specifically those that follow the principle laid down by Charles Darwin of survival of the fittest. As such, it represents an intelligent exploitation of a random search within a defined search space to solve a problem. First pioneered by John Holland in the 1960s, genetic algorithm has been widely studied, experimented and applied in many fields of engineering world. Not only does it provide alternative methods to solving problem, it consistently outperforms other traditional methods in most of the problem links. Many of the real world problems involved finding optimal parameters, which might prove difficult for traditional methods, but ideal for GA. GA exploits the idea of the survival of the fittest and an interbreeding population to create a novel and innovative search strategy (Table 3). A population of strings, representing solutions to a specified problem is maintained by the GA.

N	1	2	3	4	6	10	20	50
No. Of Solutions	1	2	6	24	720	3×10^6	2×10^{18}	3×10^{64}

Source: Adewumi, 2009.

Table 1: Number of distinct solutions which exist for agricultural extension agents as a function of n, the number of cities.

***Corresponding author:** Adewumi IO, Department of Agricultural Engineering, Federal College of Agriculture, P.M.B. 5029, Moor Plantation, Ibadan, Oyo State Nigeria, Tel: +2348023821869, +2347034205740; E-mail: adexio2010@gmail.com, adexio@yahoo.com

Received August 28, 2015; **Accepted** September 25, 2015; **Published** September 30, 2015

Citation: Adewumi IO, Oluwatoyinbo FI, Omoyajowo AO, Ajisegiri GO, Akinsete AE (2015) Genetic Algorithm: A Veritable Tool for Solving Agricultural Extension Agents Travelling Problem. Agrotechnol 5: 138. doi:10.4172/2168-9881.1000138

	1	2	3	4	5	.....J
1	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	.....C_{1j}
2	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	.....C_{2j}
3	C_{31}	C_{32}	C_{33}	C_{34}	C_{35}	.....C_{3j}
4	C_{41}	C_{42}	C_{43}	C_{44}	C_{45}	.....C_{4j}
5	C_{51}	C_{52}	C_{53}	C_{54}	C_{55}	.....C_{5j}
I	C_{i1}	C_{i2}	C_{i3}	C_{i4}	C_{i5}	.....C_{ij}

Generational Result

Table 2: The cost matrix.

Chromosomes						Raw Fitness	Relative Fitness
5	3	6	4	1	2	01.172	306.862
4	2	3	6	5	1	00.283	074.296
2	1	3	5	4	6	00.201	052.865
2	4	1	3	5	6	00.153	040.071
6	5	4	2	3	1	00.115	027.893
5	2	1	6	3	4	00.093	023.685
6	4	5	2	1	3	00.081	019.692
2	3	5	1	6	4	00.062	016.809
3	1	2	5	6	4	00.063	015.074
5	1	2	4	6	3	00.051	013.776
1	3	2	5	4	6	00.045	011.751
5	3	2	1	4	6	00.041	010.691
4	6	2	1	3	5	00.039	010.280
6	5	1	3	4	2	00.036	009.397
1	4	3	2	6	5	00.033	008.678
5	3	6	4	1	2	00.032	008.440
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
2	1	3	5	4	6	00.029	007.897

The Best Solution for This Generation
The best chromosome=5, 3, 6, 4, 1, 2.
Raw fitness=01.172.
Relative Fitness=306.862
The worst solution for this generation
The worst chromosome=2 1 3 5 4 5
Raw fitness=00.029
Relative fitness=007.455

Table 3: The population for generation number 7.

Representation of the problem domain

Genetic algorithm does not deal with the data contained in the problem (or its possible solutions directly, but uses a representation of these data. This representation is mostly an encoding of the original numerical (decimal) values into a binary string of '0' and '1'. In theory, the encoding can be done over any form of finite alphabets, but the binary representation is the most efficient as it is the most basic form of representation. Very often, the problem or its solution contains data not in numerical form. A behavior-oriented model for instance, will contain possible behavior rules the system is looking to optimize. A trading model for financial markets (such as Adaptive Portfolio Trading (APT) system) contains the behavior rules of 'buy' and 'sell' that the GA processes should learn to select once certain conditions are met. A GA-ready model will typically represent these rules through enumerative integer values (as simply as <enum> or <int> data types). When the GA engine returns the selected integer value, the application object will execute the appropriate rule normally using some kind of a <switch> or nested <if> statement.

Biological metaphors for genetic algorithm

Genetics: Within most cells in the human body (and in most living organisms) are rod-like structures called Chromosomes. These chromosomes dictate various hereditary aspects of the individual. Within the chromosomes are individual genes: a gene encodes a specific feature of the individual, for example, a person's eye color is dictated by a specific gene (Table 4). The actual value of the gene is called an Allele; so, the eye color gene may produce brown eyes. A hierarchical picture is built up with alleles being encoded as genes with sequences of genes being chained together as chromosomes, which makes up the DNA of an individual [8]. When two individuals mate, both parents pass their chromosomes onto their offspring. In humans, who have 46-paired chromosomes in total, both parents pass on 23 chromosomes each to their child. Each chromosome passed to the child is an amalgamation of two chromosomes from a parent. The two chromosomes come together and swap genetic material, and only one of the new chromosome strands is passed to the child. So, the chromosome strands undergo a crossover of genetic material, which leads to a unique new individual. As if this were not enough, genetic material can undergo mutations, resulting from imperfect crossovers or other external stimuli. Although mutation is rare, it does lead to an even greater diversification in the population. It must be noted

Chromosomes						Raw Fitness	Relative Fitness
5	3	6	4	1	2	01.172	306.862
4	2	3	6	5	1	00.283	074.296
2	1	3	5	4	6	00.202	052.865
2	4	1	3	5	6	00.153	040.071
6	5	4	2	3	1	00.107	027.893
5	2	1	6	3	4	00.091	023.685
6	4	5	2	1	3	00.075	019.692
2	3	5	1	6	4	00.064	016.809
3	1	2	5	6	4	00.058	015.074
5	1	2	4	6	3	00.053	013.776
1	3	2	5	4	6	00.045	011.751
5	3	2	1	4	6	00.041	010.691
4	6	2	1	3	5	00.039	010.280
6	5	1	3	4	2	00.036	009.397
1	4	3	2	6	5	00.033	008.678
5	3	6	4	1	2	00.032	008.440
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.029	007.897

The best solution for this generation
The Best Chromosome=5 3 6 4 1 2
Raw fitness=01.172
Relative fitness=306.862
The worst solution for this generation
The worst chromosome=4 2 3 6 5 1
Raw fitness=00.030
Relative fitness=007.76

Table 4: The population for generation number 8.

however, that a significant number of mutations are harmful and can destroy good genetic code, so the rate of mutation must be low in order to prevent severe degradation of the genetic code.

Benefits of genetic algorithm

Martello [9] as cited by Goldberg [5] discovered that one of the GA's most important qualities is its ability to evaluate many possible solutions simultaneously. This ability, called Implicit Parallelism (Keith Grant) is the cornerstone of the GA's power. Implicit parallelism results from simultaneous evaluation of the numerous building blocks that comprises the string. Each string may contain millions of these building blocks, and the GA assesses them all simultaneously each time it calculates the string's fitness (Table 5). In effect, the algorithm selects for patterns inside the string that exhibit high worth, and passes these building blocks onto the next generation. This selection process enables genetic algorithm to perform well where traditional algorithm flounder such as in problems with huge search spaces [10]. The objective of this study is to use genetic algorithm (GA) to go through a number of cities that will be visited by agricultural extension agents or workers when disseminating new innovation and technology in agriculture to local farmers in order that.

The tour is completed within minimum time (time management).

Reduction in total cost value incurred on the tour.

Find the routes that minimize the total distance.

Profit maximization.

Chromosomes						Raw Fitness	Relative Fitness
5	3	6	4	1	2	01.172	306.862
4	2	3	6	5	1	00.283	074.296
2	1	3	5	4	6	00.202	052.865
2	4	1	3	5	6	00.153	040.071
6	5	4	2	3	1	00.107	027.893
5	2	1	6	3	4	00.091	023.685
6	4	5	2	1	3	00.075	019.692
2	3	5	1	6	4	00.064	016.809
3	1	2	5	6	4	00.058	015.074
5	1	2	4	6	3	00.053	013.776
1	3	2	5	4	6	00.045	011.751
5	3	2	1	4	6	00.041	010.691
4	6	2	1	3	5	00.039	010.280
6	5	1	3	4	2	00.036	009.397
1	4	3	2	6	5	00.033	008.678
5	3	6	4	1	2	00.032	008.440
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	007.897
4	2	3	6	5	1	00.030	000.000

The best solution for this generation
The best chromosome=5 3 6 4 1 2
Raw fitness=01.172
Relative fitness=306.349
The worst solution for this generation
The worst chromosome=4 2 3 6 5 1
Raw fitness=00.030
Relative fitness=000.000

Table 5: The population for generation number 9.

Due to time constraint, the number of cities is restricted to ten (10) to give 2 x 106 initial solutions from which the fitter chromosomes can be selected for further generation. The data type of chromosomes is also restricted to only integer numbers, binary and floating point numbers are not provided for in this study.

Outline of the basic genetic algorithm

[**Start**] Generate random population of *n* chromosomes (suitable solutions for the problem).

[**Fitness**] Evaluate the fitness *f(x)* of each chromosome *x* in the population.

[**New population**] Create a new population by repeating following steps until the new population is complete.

[**Selection**] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected).

[**Crossover**] With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.

[**Mutation**] With a mutation probability mutate new offspring at each locus (position in chromosome).

[**Accepting**] Place new offspring in a new population.

[**Replace**] Use new generated population for a further run of algorithm.

[**Test**] If the end condition is satisfied, stop, and return the best solution in current population.

[**Loop**] Go to step 2.

Methodology

The ultimate goal of this research work is to determine a route through a set of n cities and back to starting point without repetition, which will maximize resources and save cost. N number of cities will be supplied from console for this project. The genetic algorithm search paradigm looks for the best permutation set, which will meet the objective of the study as stated earlier. In carrying out these work, n cities representing the number of cities was permuted, which will yield n! The fitter chromosomes among them are selected for future generation. The selected chromosomes are crossed over and mutated until an optimal solution is achieved.

Encoding the problem

The problem was coded using a sequence of integer values to represent the cities in a tour. For example: 1-2-3-4 represents a tour from city 1 through cities 2, 3 and 4 in that order.

The cost matrix

The cost matrix is represented by the distances between the two adjacent cities. For example C_{ij} represents the distance between city i and city j in cost table. The cost matrix is shown in the table below.

Analysis of genetic algorithm system for agricultural extension agent

The system of transit for agricultural extension agent can be analyzed using the following components:

Input: The input to the system is the number of cities to visit by

an agricultural extension agent. This number of cities will be captured from the keyboard or from a remote location provided that there is accessibility to the system from that location. The input data is expected to be a valid integer number. Also, a user is expected to choose stopping criteria, which will be provided. Mutation probability and crossover probability are to be supplied from the console. They should be fraction number n, where $0 < n < 1$.

Process: The main transformation that is expected to take place within the system is the determination of a set of solutions to solve the problems. Once a valid input data is supplied from the console, the first process that will take place is to randomly generate a set of valid tours, which will be N! where N is the number of cities supplied by the user. After the generation of all possible random tours, fitness value of each tour will be determined. Each tour in genetic algorithm paradigm is what we refer to as Chromosomes. Part of the fitter chromosomes will be selected for future mating among them. Initial population will be displayed. Thereafter reproduction operators (selection, crossover, mutation merge) will be supplied on each generation thereby leading to the production of subsequent generations. This trend continues until stopping criteria is true.

Output: At the initial stage, the result of initial generation will be displayed before, and thereafter, the result of subsequent generations will be displayed from where the user can select whatever option he can afford.

Feedback: The feedback can either be positive or negative. The feedback from this system will be whether the system is optimizing or not. If it optimizes well, the user may have to increase the value of some of these parameters or reduce the parameters depending on the parameter in question.

Control: This depends on the result of the feedback. If the result is favorable, the generation number may have to be increased so that more optimal solutions can be achieved. Control is demonstrated with series of input data and result obtained in appendices A and B.

Environment: The environment of this system that worth mentioning here is minimum hardware requirement that was used to carry out this research work. This will flourish in an environment where search space is very wide and fuzzy.

Boundary and interface: This system can be interfaced with other areas of artificial intelligence like fuzzy logic, simulated annealing, evolutionary algorithm.

System development tool

The system development tool chosen for this project is dependent on the features provided. Visual Basic Version 6.0 is used for the implementation of this design. This is due to the flexibility, simplicity of features that enhances easy movement of data from console to the program and ability to easily print results into an output file and monitor simultaneously.

System description

This system deals with how best n cities can be toured at the lowest distance and cost possible. Visual Basic 6.0 was used to implement this and the entire program is explained below.

The main program: The main menu gives the user the opportunity to either optimize the traveling salesman problem using typical method, which will use just a method from each of the operators as follows: generational, integer representation, two-point crossover, uniform mutation, and roulette selection and generation termination method. Customized optimization can also be done from this point by selecting from numbers of methods provided by each operator.

Generational algorithms: This module offers two options if customized is selected from the main menu. The generational option randomly generates the chromosomes automatically without using any city as a starting point or terminal point. On the other hand, steady state enables you to have options of selecting the city that will be origin or destination of a chromosome.

Data types: This customized option enables us to select what will be the format for representing cities in the chromosome. The data representations possible are binary representation, integer representation and floating-point representation. Once a data representation is chosen, it is this representation that will be used as object of manipulation for the subsequent period. The screenshot of this module is shown in Appendix B.

Crossover operators: This module allows the user to choose the type of crossover methods to be used in case we do not want the system to choose for us. The crossover methods available in this work are partially mapped crossover, order crossover and cycle crossover. The methods are actually expected to optimize the problem, but the solution provided by these methods are relatively the same. The screenshot of the module is shown in Appendix B. the results produced by these methods are shown in Appendix C.

Mutation operators: This module enables better result to be produced by carrying out some structural adjustment in the genes of the selected chromosome. The functionality of mutation depends on the data type selected. If the binary data type is only carried out by flipping over a bit from 0 to 1 or from 1 to 0, if the data type selected under data type module is integer representation, mutation is carried out by randomly selecting two mutation points. The genes in these two points are swapped to form other chromosomes. The mutation methods available are flip bit, boundary, uniform and Gaussian mutations.

Selection operation

Selection operation is the reproduction operator. This operator selects chromosomes that will form population for the subsequent generation(s). The methods under this module include roulette, tournament, top percent, best and random method. Whichever method selected here will copy chromosomes from both the old and the new populations to form population for the following generation.

Termination operator: This is the last module that determines when the genetic algorithm should stop. The methods available under this module include generation number, evolution time, fitness convergence, population convergence and gene convergence. The screenshot of this module is shown in Appendix B.

Parameter specification: This module takes all the necessary parameters that will be used by the operators. These parameters among other things include the following:

Population size: This parameter specifies the number of cities that will be involved in the problem. Due to time constraints, 10 cities were considered in this work. However, this can be improved upon for it to work on several numbers of cities. Whatever number of cities supplied will determine the number of solutions that will be provided. If population size is n, then, the population size will be n!

Termination parameter: This parameter determines the number of generations that genetic operation will pass through. This will

depend on the type of termination method selected; for example, in generation number, if the generation number is 10, it implies that the genetic operation will go for 10 times.

Crossover probability: This probability specifies whether a crossover can be carried out on any selected chromosomes or not. The default is 0.6. The essence is to ensure that fitter chromosomes are able to survive to the following generation.

Mutation probability: This probability specifies the possibility of changing the internal structure of a chromosome. It is normally kept very low so that there will be no unnecessary changes in the structure of genes. The default is 0.01.

Program listing

The program listing of this research work is displayed in Appendix C.

System specification

The basic systems requirements to run this study are as given below:

Windows XP Professional, Window Vista, Linux Operating System.

Pentium 233 MMX Processor.

32MB SD RAM.

VGA Monitor.

The above requirement is the minimum requirement whereby we can use to run this research.

Testing

The approach adopted in testing the functionality of this program is mainly black box testing. Series of input genetic parameters are supplied from the console to get what the result will be. The operation is repeated for series of cities between 4 and 10 to see how the efficiency of the program as the search space continues to increase. Under this approach, we are only interested in the output and to what is actually happening within the program. Therefore, it may not be able to detect any redundant code that we may have within the program. In view of this, Glass Box Testing is also carried out to detect errors within the program. The methods adopted under glass box testing technique include the following: Loop testing, Branch testing, Path coverage and Statement coverage. The results generated as a result of this testing are shown in Appendix B of this work.

Result Analysis

Four different parameters are supplied and their results are displayed in Appendices A and B: the first set of parameters taken so small to show the limitation of genetic algorithms over a narrower search space. Notwithstanding, the population size, if the generation number is very small with lower crossover probability and mutation probability, the GA may go through series of generation with optimizing anything at all; for example agricultural extension parameters. As the generation number continues to increase with corresponding increase in the crossover probability and decrease in mutation probability, a search space will continue to increase thereby unleashing the capability of GA approach to solving a traveling salesman problem. Figure 1 in Appendix B shows the parameters supplied and their corresponding results are shown in Appendix B. However, care should be taken to vary the parameters of the transit for agricultural extension agent, the reason being that while we are trying to exploit the benefit arising from increasing mutation probability, we may also loose the initial population entirely. Therefore, we should try as much as possible to keep the probability low. The default for mutation probability is 0.01 and crossover probability is 0.6. There is no default for population and stopping criteria; it is situational, that is, it is the problem at hand that determines their values.

Conclusion

Genetic algorithms balance exploitation with exploration. The crossover and mutation operators control exploration while the selection and fitness functions control exploitation. Mutation increases the ability to explore new areas of the search spaces but it also disrupts

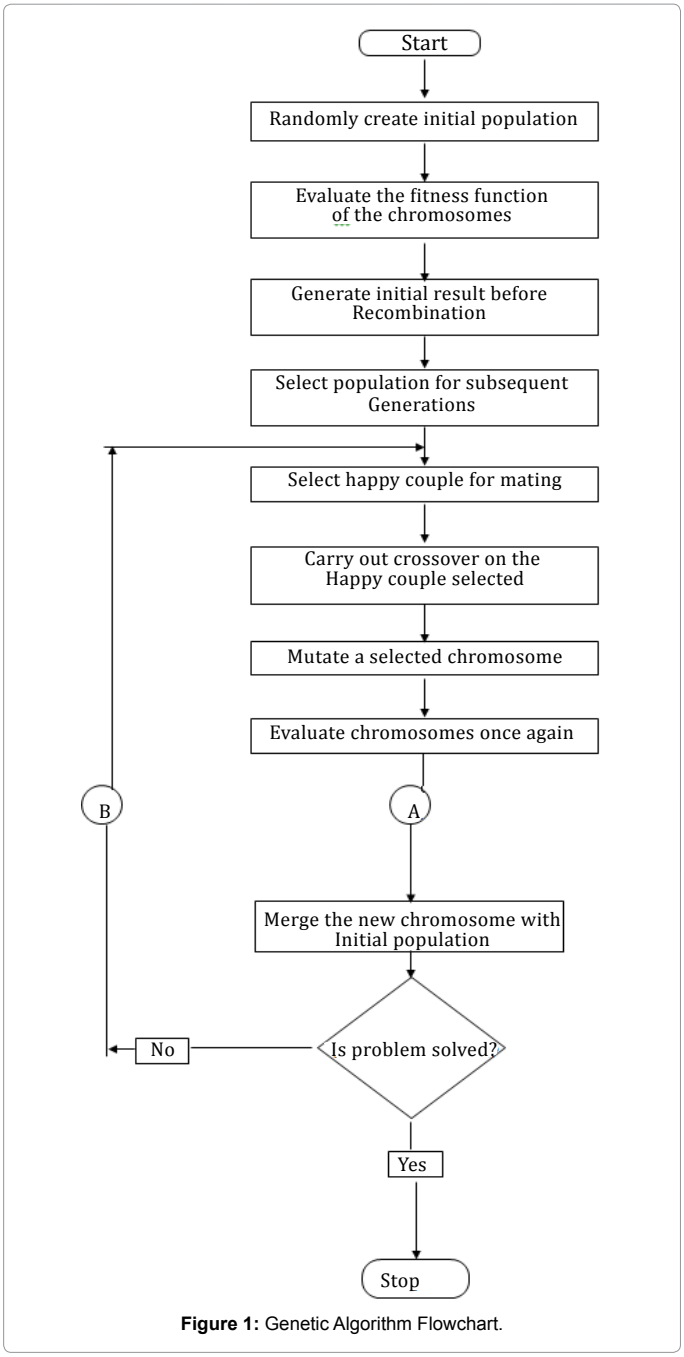

Figure 1: Genetic Algorithm Flowchart.

the exploitation of the previous generation. It has been clearly showed from the study so far that the objectives of the work has been achieved in terms of time management, shortest routes that minimize the total distance which will leads to reduction in total cost value incurred and profit maximization for the agricultural extension agents/firm during the course of disseminating new and improved technology to local farmers in order to increase the food security in the country. The ability to separate the search mechanism from the model representation makes genetic algorithms an ideal approach for the solution of very complex combinatorial optimization problems. Travelling problem of an agricultural extension agent is one of those complex problems with a very wide search space where genetic algorithm has proved very effective to find a lasting solution to the problem of determine the least possible cost of routing a number of cities.

Recommendation

For further work on genetic algorithm implementation in agriculture, the following must be taken into consideration:

Evaluation on a larger dataset, because in most time GA fails when it is applied to a very larger problems because it scales poorly in terms of complexity as the number of cities increases. So more research has to be carried out on the convergence of the chromosomes.

It also has to be evaluated for a longer time frame; this is because the solution quality degrades rapidly.

Proper fitness function has to be examined.

References

1. Sebusang SEM, Masupe (2003) ICT development in Botswana: connectivity for rural communities. Southern African Journal of Information and Communication (SAJIC) the Edge Institute/Research ICT Africa, Braamfortein ZA.
2. Odoemelam LE, Onuekwusi GC (2013) Effects of ICTS-Based Agricultural Extension Services in Enhancing the Well-Being of Farmers in Abia State, Nigeria. International Journal of Applied Research and Technology. 2: 15-22.
3. Adewumi IO (2009) Using Genetic Algorithm For Solving Travelling Salesman Problem. B.Sc. Project Submitted to Department of Industrial & Production Engineering, Faculty of Technology, University of Ibadan 2.
4. Oliver IM, Smith DJ, Holland JRC (1987) A study of permutation crossover operation on the Travelling Salesman Problem. Proceeding of the 2nd International Conference on Genetic Algorithms. 224-230.
5. Goldberg DE (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.
6. Prasanna J, Jung YS, Dirk VG (1991) Parallel genetic algorithms applied to the traveling salesman problem. SIAM Journal of Optimization 1: 515-529.
7. Frick A (1998) TSPGA-An evolution program for the symmetric traveling Salesman Problem. In HJ. Zimmermann, editor, EUFIT98 - 6th European Congress on Intelligent Techniques and Soft Computing 513-517.
8. Whitley D (1994) A Genetic Algorithm Tutorial. International Journal of Statistics and Computing 4: 65-85.
9. Martello S (1983) An enumerative algorithm for finding Hamiltonian circuits in a directed graph. ACM Transactions on Mathematical Software 9: 131-138.
10. Torimiro DO, Kolawole OD (2008) In: Akinyemiju AO, Torimiro DO (eds) "New Partnership for Africa"s Development and Agricultural Technology transfer in a globalised world, Agricultural ExtensionL a comprehensive tratise. Lagos: Ikeja ABC Agricultural systems Ltd., pp. 402-414.