

Genetic Algorithm based Cosmetic Product Forecasting

Khin Aye Mar
Faculty of Computing
University of Computer Studies (Mandalay)
Mandalay, Myanmar

Abstract: The resulting greedy GA favours objects with higher value densities when it builds random chromosomes, in crossover, and in mutation. The greedy heuristics do well, as does the naive GA, but the greedy GA exhibits the best performance. Genetic algorithms initiate the process of evolution on an optimization problem. This system is combined greedy idea and genetic algorithm to form the greedy genetic algorithm which incorporates the global exploring ability of the genetic algorithm and the local convergent ability of the greedy algorithm. In this system, population evolution utilizes quantity of sales for cosmetic goods as integer variables. And this system includes fitness function for requirement of profit amount and quantity of products can be calculated by the specific formula. Then, the selection is performed with method of roulette wheel selection. The final result is the forecasting of cosmetic goods for monthly. The greedy genetic algorithm (GGA) always chooses the best goods during the crossover and mutation process according to their fitness values.

Keywords: Genetic algorithm, crossover, mutation, fitness function, initial population

1. INTRODUCTION

Genetic algorithms are a type of optimization algorithm, meaning they are used to find the optimal solution(s) to a given computational problem that maximizes or minimizes a particular function. Greedy Genetic algorithms (GGA) are general-purpose, parallel search techniques for solving complex problems. GGA works by repeatedly modifying a population of artificial structures through the application of genetic operators. A GGA maintains a population of feasible solutions, also known as chromosomes on which the concept of the survival of the fittest among string structures is applied. This system is applied greedy genetic algorithm for forecasting of cosmetic products in the business application [5, 7].

This system represents an optimization approach to a real-world product forecasting problem and will forecast the products by using greedy genetic algorithm. This system is discovered near optimal solutions for business area within an acceptable time. This system also provides a statement for effectively sampling large search spaces and forecast products for the future in business application. The remaining of sections are described theoretical background of the system. Section 3 discusses greedy genetic algorithm for cosmetic product forecasting. Section 4 describes system design. Section 5 explains the implementation results of cosmetic goods forecasting using greedy genetic algorithm in step by step processing. Finally, Section 6 presents conclusion, limitations and future plan of the system.

2. GENETIC ALGORITHM

The simple GA is composed of population of strings, or chromosomes, and three evolutionary operators: selection, crossover, and mutation. The chromosomes may be binary-coded, or they may contain characters from a larger alphabet.

The initial population is typically generated randomly, but it may also be supplied by the user.

Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem. [4]

2.1 Greedy Genetic Algorithm

The greedy genetic algorithm (GGA) is similar to the classical genetic algorithm (CGA) except for the operation of choosing offsprings in the crossover and mutation. Genetic algorithms (GA's) imitate the process of evolution on an optimization problem. Each feasible solution of a problem is treated as an individual whose fitness is governed by the corresponding objective function value. A GA maintains a population of chromosomes on which the concept of the survival of the fittest (among string structures) is applied.

```
algorithm genetic;  
begin  
    obtain initial population;  
repeat  
    select two individuals I1 and I2 in the population;  
    apply the crossover operator on I1 and I2 to produce  
    a child I3;  
    choose two individuals out of I1, I2 and I3;  
    occasionally perform immigration;  
until the population converges;  
end;
```

Figure. 1 The Greedy Genetic Algorithm

There is a structured yet randomized information exchange between two individuals (*crossover* operator) to give rise to better individuals. Diversity is also added to the population by

randomly changing some genes (*mutation* operator) or bringing in new individuals (*immigration* operator). A GA repeatedly applies these processes until the population converges. The Greedy Genetic Algorithm is illustrated in figure 1 [1, 7].

3. GGA FOR COSMETIC GOODS FORECASTING

This system combines greedy idea into the genetic algorithm. To build string for its initial population, the greedy GA begins by computing the absolute value densities (2) of all the objects. To build each string, the GA considers the objects one at a time. The object to consider next is determined by a tournament with replacement of two objects randomly chosen from those not previously considered.

3.1 Initial Population as Integer

An individual is represented by a set of unsigned integer numbers. Values may be restricted to a finite set. This can be formulated as GAs operate on a number of potential solutions, called a population, consisting of some encoding of the parameter set simultaneously [1]. An example of integer representation of strings is shown in figure 2.

Strings v_1	23 45 55 30 38 58
Strings v_2	44 35 21 56 27 39

Figure.2 Integer representation of Initialization

3.2 Two-Site Crossover

This operator randomly selects two crossover points within a string, and then interchanges the two parent strings between these points to produce two new offsprings. Two site crossover is a variation of the one site crossover, except that two crossover sites are chosen and the bits between the sites are exchanged. [2]. An example of two site crossover is shown in figure 3.

String 1	011 011 00	String 1	011 110 00
String 2	110 110 01	String 2	011 011 01
	Before crossover		After crossover

Figure.3 Two site crossover operation

The GGA does not directly accept two offsprings v_1' and v_2' as CGA does. Compute the fitness of $\{v_1, v_2, v_1', v_2'\}$. Then choose two best chromosomes from these four as the “offsprings” according to their fitness values [7].

3.3 Random Resetting Mutation

In natural evolution, mutation is a random process where one allele of a gene is replaced by another to produce a new genetic structure. In GAs, mutation is randomly applied with low probability, typically in the range 0.001 and 0.01, and modifies elements in the string. An example of random resetting mutation is shown in figure 4.

Original Offspring v	23 45 <u>21</u> 56 27 58
Mutated Offspring v'	23 45 <u>33</u> 56 27 58

Figure. 4 Random resetting mutation

3.4 Stopping Criterion

The stopping criterion is used in the evaluation process to determine whether or not the current generation and the best solution found so far are close to the global optimum. The standard stopping criteria is used to stop the procedure after the given number of iterations. And, another is used to stop after the 'best' solution has not changed over a specified number of iterations. Another stopping criterion is used when the average fitness of the generation is the same or closes to the fitness of the 'best' solution. In this system, the stopping criterion is used to stop the procedure after a given number of iterations [3].

3.5 Fitness Function

In order to efficiently use the criterion function, it is necessary to define a fitness function which properly assesses the decision rules. The fitness function takes as an input a set of feature or attribute definitions, a set of decision rules. The fitness function or evaluation function plays an important role in a GGA.

The fitness of individuals was evaluated base upon their ability to correctly predict monthly sales data for the product and profit. The fitness of this system means the amount of product for next month that is the best individual. This system defines $f(v_i)$, the fitness of individual i , according to [2,6]

$$f(v_i) = \frac{1}{1 + \alpha}$$

where,

$$\alpha = \frac{1}{t} \sum_{j=1}^N (x_j - x_{jk})^2 + \sum_{i=1}^N (x_i NP_i - x_{ik} NP_i) + \wedge$$

$NP_i = GP_i - \text{Expenses}$
 $GP_i = \text{Sales} - \text{Cost of Sales}$

and where

- α = forecast amount of product with profit
- \wedge = $10^{-5} \times (\text{chromosome length})$, is a penalty term that favors shorter genetic programs over longer ones.
- t = the number of months in training set

- N = chromosome length
- x_i = the actual sales of products for each individual v_i
- $\hat{x}_{i,k}$ = the forecast of products for each individual v_i
- NP_i = net profit for each product
- GP_i = gross profit for each product

3.6 Roulette Wheel Selection

The commonly-used reproduction operator is the proportionate reproduction operator where a string is selected for the mating pool with a probability proportional to its fitness. Thus, the i th string in the population is selected with a probability proportional to F_i . Since the population size is usually kept fixed in a simple GA, the sum of the probability of each string being selected for the mating pools must be one. the probability for selecting the i th string is

$$p_i = \frac{F_i}{\sum_{i=1}^n F_i}$$

where n is the population size. One way to implement this selection scheme is to imagine a roulette-wheel with its circumference marked for each string proportionate to the string's fitness. The roulette-wheel is spun n times. each time selecting an instance of the string chosen by the roulette-wheel pointer. The average fitness of the population is calculated as

$$F_i = \sum_{i=1}^n f_i$$

4. SYSTEM FLOW DIAGRAM

This system uses 56 kinds of cosmetic goods. At the beginning, it retrieves actual sales data for 24 months (two years) as product strings from the database. For example, 5 kinds of cosmetic goods for 4 months as strings are shown in figure 5. In this figure, v_1, v_2, v_3 and v_4 are month 1, 2, 3, 4 and the short terms are mean as: S01 = lipstick, S02= Eyes harrow, S03= Body Lotion, S04=Makeup, S05= Toner.

Strings (Monthly)	S01	S02	S03	S04	S05
V1	35	40	45	39	51
V2	44	33	58	42	53
V3	28	54	31	53	48
V4	50	26	29	23	36

Figure. 5 Initial population generation of goods string

Then, the system calculates the fitness value for each goods string based on quantity of sales data, forecast values, profit, time interval and expenses to forecast the goods for the future. The system prompts the user to input number of iteration to be continued to get optimal solution of the forecasted goods.

After that, the system processes probability of the goods string, selecting products by using roulette-wheel selection method, exchanging between goods with two-point crossover operators, mutating the new goods string (offsprings) with random resetting mutation operators, and calculates fitness of the new goods chromosomes. At the stages of crossover and mutation, the system chooses the best chromosomes according to GGA. After completion above processes, the system checks whether it reaches final loop of iteration. If the system reaches final iteration, then it terminates the process and displays the optimal solution of the forecasted amount of goods for the future. The process flow of the system is shown in figure 6.

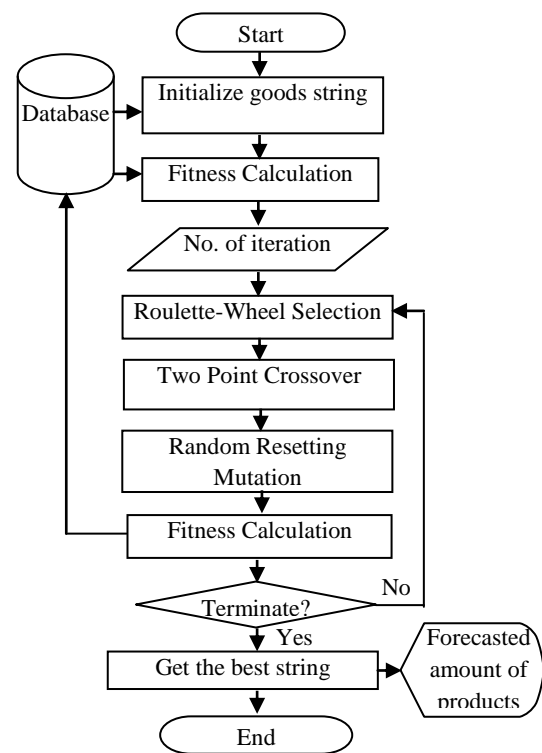


Figure. 6 System flow diagram

5. EXPERIMENTAL RESULT

Initially many individual solutions (actual monthly sales of products) are generated from the database to form an initial population. The control parameters are determined by a GA based on the fitness value calculated from the profit, time interval, actual and forecast sales of goods as shown in figure 7.

In roulette wheel selection, individuals are selected based on the probability that is directly proportional to their fitness value. The crossover is used to goods new solution. After crossover there are two parents and two new solutions. Then compare the fitness of these four solutions and choose the best

two solutions as offsprings. In mutation, the fitness of original offspring and mutated offspring are computed and choose the better string as the offspring.

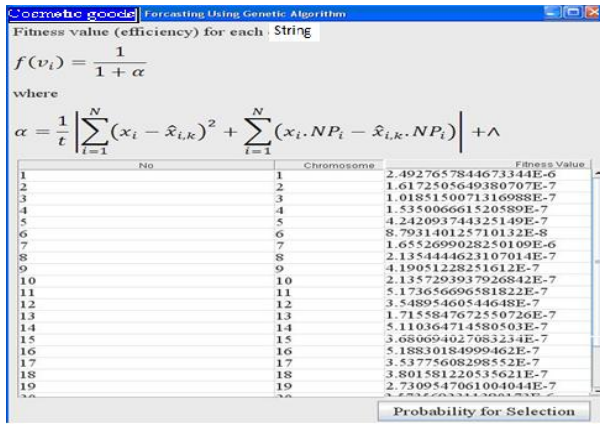


Figure. 7 Fitness calculation for goods

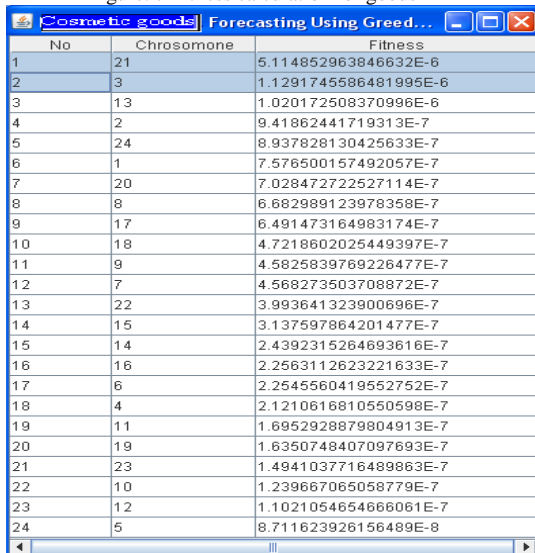


Figure. 8 The fitness values for best forecasted cosmetic goods

For the initial implementations, other control parameters were based on those suggested in popular GA: chromosome length is 56, population size is 24 months, and crossover probability is 0.67 and mutation probability is 0.003.

6. CONCLUSION, LIMITATIONS AND FUTURE PLAN

The power of GAs comes from the fact that the technique is robust, and can deal successfully with a wide range of problem areas, including those which are difficult for other methods to solve. GAs are not guaranteed to find the global optimum solution to a problem, but they are generally good at finding acceptably good solutions to problems acceptably quickly. In this system, integer representation for string are easy to understand and can represent a large number of populations for large search space. By defining meaningful fitness function, the objective of the system can be achieved effectively and quickly. In this system, GGA always chooses

the best string during the crossover and mutation process. This may give more satisfy results even the population is small.

In business application, there can have difficulty to forecast the goods manually if there are too large amount of datasets. So, this system can produce the result effectively and quickly by using GGA. Because cosmetic goods in business application are using as data sets in this system, user want to perform in practical that gets result within a short time.

The system can provide the user to forecast the goods for future in business application within acceptable time. By using this system, user can obtain the preferred solution for market such as profit, products and general expenses for future.

This system can extend to forecast large numbers of data sets such as entertainment products and telecommunication products for long-term. This system can only be solved to forecast cosmetic goods for monthly.

References

- [1] Bc.Marek Hrusovsky, "Genetic Algorithm Acceleration Using OpenCL" Department of Computer Systems, Brno University of Technology, 2010.
- [2] Craenen, B. C. W., Eiben, A. E. and Marchiori, E., How to Handle Constraint with Evolutionary Algorithms. In L. Chambers (Ed.) "The Practical Handbook of Genetic Algorithms" Applications, 2nd Edition, volume 1, Chapman & Hall/CRC, 2001, pp. 341 – 361.
- [3] Edwin N., Paul A., Marshall Mc B., Ron L. and Claudelte O., "Modeling and Simulation Optimization Using Evolutionary Computation", COLSA Corporation, 6726 Odyssey Drive Huntsville, Al 35806, 256-964-5295, 256-964-5355.
- [4] Mitchell Melanie, An Introduction to Genetic Algorithms, A Bradford Book The MIT Press Cambridge, Massachusetts, London, England Fifth printing, 1999, First MIT Press paperback edition, 1998
- [5] Ravindra k. Ahuja, James B.Orlin, Ashish Tiwari,"A Greedy Genetic Algorithm for the Quadratic Assigment Problem", University of Florida, Gainesville, Massacusetts Institute of Technology, Cambridge, State University of New York, Stony Brook.
- [6] Stephen D.Sloan, Raymond W.Saw, James J.Sluss, JR.,Monte P.Tull, and Joseph P.Havlicek, "Genetic Algorithm Forecasting for Telecommunications Products", School of Electrical and Computer Engineering, University of Oklahoma, Norman, Oklahoma.
- [7] ZHAO Xinchao, "A Greedy Genetic Algorithm for Unconstrained Global Optimization", Key laboratory of Mathematics, Institute of System Science, AMSS, Chinese Academy of Sciences, Beijing 100080, China.