

Optimization of a Mini-Golf Game using the Genetic Algorithm

A. Coskun

*Department of Computer Education, Gazi University Faculty of Industrial Arts Education,
Golbası Campus, Ankara, Turkey, phone: +90 (312) 4851124, e-mail: aysunc@gazi.edu.tr*

Introduction

A fundamental definition of optimization is finding the best solution under the given constraints. Solving the optimization problems is possible by using the algorithms based on mathematical expressions or rules that are providing defined limitations are met. There is not an effective method that can be commonly used for solving entire optimization problems and the methods used do not guarantee finding the exact solution of the problem. However, in general they are successful in finding a better solution than the acceptable measure at the acceptable speed.

The artificial intelligence optimization techniques aimed at solving the optimization problems are also known as meta-intuitional research techniques. These algorithms are the criteria or computer methods those are defined to decide which of the various alternative movements are effective to realize any one solution or to reach any specific target. For larger problems, researchers have been directed to study on the methods those are reaching a “good” solution in shorter times even though they may not be the best solution methods, that is because the number of problems are limited, which can be solved using definite solving methods and magnitudes of the required time for the calculations are so great. The application results of the artificial intelligence optimization algorithms are also important because they do constitute an upper or lower limit for the definite solution methods. Such solution algorithms do not guarantee the definite solution; however they may guarantee to find a solution that is convenient for the purpose.

Results of the application show that the artificial intelligence optimization algorithms are problem dependent algorithms. In other words, while they may be successful for a certain problem they may not be successful same way for another problem. The simulated annealing (SA) algorithms, genetic algorithms, tabu search algorithms may be given as the principal examples of the artificial intelligence optimization algorithms [1].

Genetic algorithms are more effective than the classical optimization methods and more flexible due to its progressiveness by making minor changes in the solution

approach. For this reason, genetic algorithm attracts the attention of the researchers. As already known, the main goal of optimization is to reach to an optimal point, or rather to approach it as close as possible. There are many known classical methods for realization of this goal. Successfulness of these methods is measured by whether they are able to reach the optimal point or how close they can get to it.

In this study, a mini-golf game has been simulated in a three dimensional virtual world by using genetic algorithm. There is a robot (red box) that is controlled by the algorithm, a ball that can be moved by the robot by applying a force on it, a target point (blue pole), where the ball has to be conveyed to by the robot, and two obstacles those may be located anywhere on the way of the ball towards the target point in the above mentioned virtual world. Goal of the application is to find the proper velocity and direction components of the ball to reach the target point by adapting the genetic algorithm to this problem.

General information in relation with the genetic algorithm

It is known that at present various studies are carried out in the area of developing new solution methods for solving the complicated and difficult problems faster and easier. In this context one of the most significant methods developed so far is the genetic algorithms.

Genetic algorithm is an iterative and stochastic process. To initialize the starting population randomly and to continue the algorithm until the best solution reached are the fundamental properties of the genetic algorithm. Famous mathematician J. Hadmard had said “Both in mathematics and in the other areas discoveries and inventions are realized by merging the ideas from different areas” [2]. This saying briefly explains how the Genetic Algorithms have come into existence while also explaining its basic operational principle. Structure of the genetic algorithm is given in the following paragraphs in basic steps.

Step 1. Initialize a starting population of the solutions.

Step 2. Calculate the suitability value of each solution in your population.

Step 3. Stop the research in case the stopping criterion is met.

Otherwise perform the following steps.

3.1. Apply the natural selection procedure.

3.2. Apply the genetic algorithm operators in order to change the starting population.

Step 4. Go to Step 2.

Flow diagram of GA that is used for solving a problem is shown in Figure 1

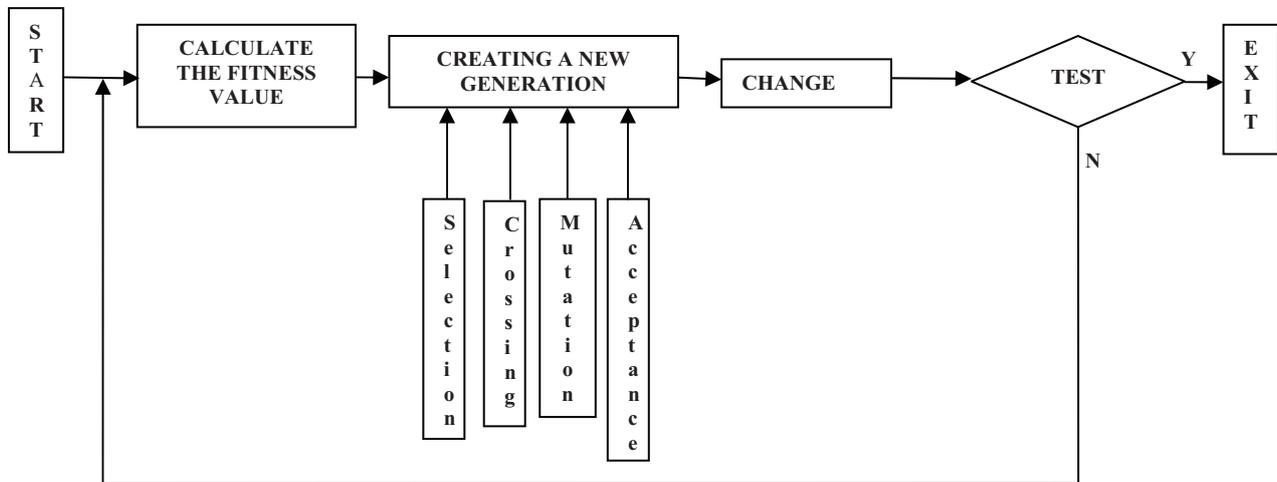


Fig. 1. Flow diagram of genetic algorithm

Steps those will be used in solving a problem must be as listed in the following paragraphs.

Representing the chromosomes (Coding). The first step in the genetic algorithm is coding the variables those consisting of the problem in every gene those consisting of the chromosomes. The created chromosome carries the entire information about a definite solving method. The coding mechanism is dependent on the variables of the problem and its method that is used has an importance for its performance. There are various ways of coding. Most commonly used coding is the binary coding. A simple genetic algorithm is operated on a population consisting of binary sets. The chromosomes here are the sets consisting of 0's and 1's. "n" bits are used for every parameter. "n" may differ for every parameter. [3].

Initializing of the starting population. The first step in application of the genetic algorithm is initializing of a starting population of the chromosomes. A solutions group is created where probable solutions are coded. The solutions group is called "population" for the sake of its resemblance in biology, and the codings of the solutions are called "chromosomes". This step is started by defining the number of individuals existing in the population. There is not a standard for this number. In general a magnitude in the 100-300 range is suggested. Complicatedness of the operations executed and the depth of the research are important for choosing the magnitude. Following this procedure population is initialized randomly.

Fitness value. The technique that is used in evaluation of the solutions is rather effective on the performance of the genetic algorithm. In case the genetic algorithm will be used for solving a problem a fitness function or a goal function is needed. In genetic algorithm basically every chromosome becomes subject to a qualitative assessment to see how good it is. The function, which qualitatively

assesses how good any chromosome is, is called fitness function. The procedure of operating this function and assessing the fitness of the chromosomes is called calculation (evaluation) [4, 5].

Selection. Selection is an operation, where the individuals are selected for reproduction. It is a process that is used to select the fitter solutions within the initialized population. It must be defined that how the individuals will be selected to breed a new generation to form the next generation and how many children each individual will have. An intermediate population is created with the individuals those will breed to form the next generation called the breeding pool. Then random individuals are selected for breeding from the pool. While there are many methods developed for selection, the most commonly used ones are roulette wheel selection (Ranking Scheme), tournament selection (Tournament Scheme) and elitist selection. Following the initialization population has been formed to generate the next new population by reproduction the selection method must be decided upon; the selection probabilities of the individuals with higher fitting values will also be higher. Here, the goal is to copy the individual having a higher fitting value (best fit) into the next generation.

Genetic operators. Selection procedure may provide information in relation with the solution of the problem. However, do not lead to a definite solution. Selection procedure continues until the magnitude of the population has been reached, and every individual that has been selected is taken into the breeding pool. Selected individuals are the parents of the new generation. The new individuals those will ensure to reach the solution are obtained at this stage by means of genetic operators. There are two important operators that take part in generating new solutions: crossing and mutation.

Genetic algorithm (GA) applications

GA literature encompasses many successful applications. However, there are also many cases where, GA's are not successful. How can it be decided whether the GA is a good method to be used, when an application has to be made? Even though many researchers are commonly sharing the opinions those are listed in the following paragraphs, there is not a serious explanation yet. In case the space is large, where the search will be made within, and it is known that it is rather regular and not having a single peak, if not firmly comprehended, or if the fitness function is noisy (i.e. assigns different values to the same chromosome in every other assessments) then GA may have a chance to be more successful than the weak methods or to compete with them.

GA applications may be gathered in three main groups. The first one of these groups is the experimental applications; these are the studies where the genetic algorithm is used in solving certain problems to prove the superiority of it against other existing optimization algorithms. Traveling Salesman Problem and Graphical Division Problem may be given as examples of such. The second group is practical applications; these are the applications where the genetic algorithm is used for the industrial solutions and other real problems. Timetable Problems, Numerical Optimization Problems, Image Processing Applications are the examples for this group. Last group is consisting of applications of classifying systems. This class is consisting of the studies where the genetic algorithms are used for data retrieval purposes. The third group applications are the applications, where the genetic algorithm is used in order to obtain the rules those are constitute the information base of an expert system.

Optimization of a mini-golf game using the genetic algorithm

An individual, a gene, and a fitness function have to be defined, suitable for the problem, in order to use the genetic algorithm for solving that problem. These definitions are made as shown in the following paragraphs:

- *Individual (Chromosome)*. Each application of a hit to the ball of the robot constitutes an individual
- *Gene*. Each component of the power vector that the robot applies to the ball in every hit constitutes a gene. Therefore, every individual is consisting of two genes: one is the x component of the power vector and the other is the y component of the power vector;
- *Fitness Function*. The ball will come to a halt after a certain time elapses, following each hit of the robot applied to the ball. A fitness function has been defined that is inversely proportional with the distance of the ball's halt position to the target point.

Contents of the functions those are necessary for the algorithm is given in the following paragraphs:

- *Fitness Function* (100 / Squared distance of the ball to the target point);
- *Crossing Function* (Function receives 2 individuals as input: individual1 and individual2. Each

individual has x and y components and these are shown as individual1.x , and individual2.y):

- o A random number is selected in the [0,1] interval;
- o In case the selected number is greater than the crossing ratio crossing is not realized and the original individuals are added to the new generation;
- o In case the selected number is smaller than the crossing ratio 2 random numbers are selected in the [0,1] interval. (random1 , random2);
- o $newIndividual1.x = individual1.x * random1 + individual2.x * (1 - random1)$;
- o $newIndividual1.y = individual1.y * random2 + individual2.y * (1 - random2)$;
- o $newIndividual2.x = individual2.x * random1 + individual1.x * (1 - random1)$;
- o $newIndividual2.y = individual2.y * random1 + individual1.y * (1 - random2)$;
- o *newIndividual1 and newIndividual2 are added to the generated new generation.*

- Mutation function:

- o Function receives one individual as input, named individual1;
- o A random number is selected in the [0,1] interval;
- o In case the selected number is greater than the mutation probability ratio, mutation is not realized and the original individuals are added to the new generation;
- o In case the selected number is smaller than the mutation ratio 2 random numbers are selected in the [0,1] interval. (random1 , random2);
- o $newIndividual1.x = individual1.x * maximumMutationRatio * random1$;
- o $newIndividual1.y = individual1.y * maximumMutationRatio * random2$;
- o *newIndividual1 is added to the new generation.*

Tests

In the table complicatedness of the problem has been evaluated according to how the obstacles are located.

Table1. Some test results obtained in the solution

Magnitude of population	Is it Elitist?	Complicatedness of the problem	Total of Generation Necessary for Solution	Duration of solution (sec)
20	Yes	Medium	43	397
20	Yes	Medium	24	206
6	Yes	Simple	45	121
6	Yes	Simple	6	17
6	Yes	Medium	137	348
6	No	Simple	24	55
6	No	Simple	36	74

As can be seen in the test results, when the magnitude of the population is chosen as a small number and the elitist property is used solution can be reached within very different times from each other. Main reason of this is: when the randomly initialized first population is far beyond the solution, very few new chromosomes can be generated in order to the chromosomes in the population get closer to the solution. The elitist property also constitutes a factor reducing the generation chances of new chromosomes since the elitist property itself occupies 2 chromosomes. Whenever, the initialized first population is consisting of values closer to the solution, due to the small magnitude of the population its good chromosomes can reach the solution rapidly.

Conclusions

In this study, the simple genetic algorithm and applications of the genetic algorithm to the general problems and the problems in the area of business administration has been investigated by carrying out a literature research. Even though, there are also applications of the genetic algorithms in many different areas such as chemistry, and physics, these areas have not been included in the scope of this study.

Other search methods are also used for solving the problem types those are investigated in the study. However, genetic algorithms are accepted as more convenient solving methods due to their properties such as being independent from the initial solution, parallel search for the solution, and rapid operation. Difficulties also exist in using of the genetic algorithms for the above mentioned problems. One

of these is defining the problem parameters. Since the parameters of each problem type are different, in general, these parameters are defined by using the experiment design method. In addition, the genetic algorithms may face with difficulties in solving the optimization problems those consisting of constraints.

The mini-golf game that is simulated by using the genetic algorithm has been designed as being three dimensional. Required velocity and direction components of the ball to reach the target point are the factors aiding for adapting the problem to the genetic algorithm.

References

1. **Ulusoy G.** Resource Constraint Charting in Project Planning // Sabancı University Engineering and Natural Sciences Faculty Magazine, 2002. – Vol. 8. – P. 23–29.
2. **Alexander J. T.** An Indexed Bibliography of Genetic Algorithms // Report Series. – Dept. Of Information Technical and Production Economics, Univ. of Vaasa, 1997. – P. 94–101.
3. **Bridges C. L., Goldberg, D. E.** An Analysis of Reproduction and in A Binary Coded Genetic Algorithm// Proceedings of the Second International Conference on Genetic Algorithms. –Cambridge, MA, 1987. – P. 9–13.
4. **Painton L., Campbell J.** Genetic Algorithms in Optimization of System Reliability // IEEE Transactions on Reliability, 2000. – Vol. 44. – P. 172–178.
5. **NirmalaDevi M., Mohankumar N., Arumugam S.** Modeling and analysis of Neuro–Genetic Hybrid System on FPGA // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 8(96). – P. 69–74.

Received 2010 08 27

A. Coskun. Optimization of a Mini-Golf Game using the Genetic Algorithm // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 3(109). – P. 97–100.

While playing golf, it is important that the ball must be conveyed to the defined target point by making the minimum number of shots. The chosen algorithm and the applying techniques for conveying the ball to the targeted point are significant in optimizing this game. A research has been carried out on the basic concepts of the genetic algorithm in this study, and has been concentrated on how this algorithm can be applied on a physics problem. Properties those have learnt as a result of the carried out research have been applied to the problem by making various arrangements as required by the nature of the problem for conveying the ball to the target point in optimizing of a mini-golf game. Effects of the presented genetic algorithm elements over the calculation time of the solution of the problem have been evaluated by various tests during the application process. It has been observed that the genetic algorithm is effective in determining the required velocity and direction components for conveyance of the ball to the target point. Ill. 1, bibl. 5, tabl. 1 (in English; abstracts in English and Lithuanian).

A. Coskun. Mini golfo žaidimo optimizavimas taikant genetinį algoritimą // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2011. – Nr. 3(109). – P. 97–100.

Žaidžiant mini golfa, svarbu, kad kamuoliukas pasiektų reikiamą tašką kuo mažesniu skaičiumi mušimų. Pasirinktas algoritmas ir taikoma kamuoliuko mušimo technika gali šį žaidimą optimizuoti. Genetinis algoritmas buvo pritaikytas fizikinėms problemoms spręsti. Atlikta analizė parodė, kaip galima optimizuoti greitį ir kryptį. Il. 1, bibl. 5, lent. 1 (anglų kalba; santraukos anglų ir lietuvių k.).