ISSN 2250-3153

Optimization Drilling Sequence by Genetic Algorithm

Abdhesh Kumar and Prof. Praveen Pachauri

Mechanical Engineering Department, Noida Institute of Engineering & Technology, Greater Noida, India

Abstract- In this paper, a Genetic Algorithm (GA) is used for the travelling salesman problem (TSP) to reduced total time and distance of tool travel for drilling sequence. By using Traveling salesman problem, it become a little bit simple and less time consuming. But there are many possible sequences in TSP. So there is need to find out the best possible solution so that the process becomes more efficient and less time consumer. By using GENETIC ALGORITHM in MTLAB software, I got best possible sequence in TSP.

Index Terms- Optimization, Genetic algorithm, MATLAB, TSP (Travelling Salesmen Problem)

I. Introduction

Drilling is a one of most common machining process. There are a lot of areas where drilling is used. A sugar mills consists of hundreds of different sized holes distributed over a large area. CNC machines are used to today perform the drilling process. These machines are capital intensive and their maximum utilization is the to their economic viability. An issue that affects the utilization of these machines is the drilling sequence because usually there is 'n' number of holes that has to be visited. Determination of drilling sequence is similar to a Traveling Salesman Problem (TSP) and exhibits characteristics of an NP-hard problem.

Drilling is defined as "An implement with cutting edges or a pointed end for boring hole in hard material, usually by a rotating abrasion or repeated blows".

The most prominent member of the rich set of combinatorial optimization problems is undoubtly the traveling salesman problem (TSP), the task of finding a route through a given set of cities with shortest possible length. It is one of the few mathematical problems that frequently appear in the popular scientific press (Cipra (1993)) or even in newspapers (Kolkata (1991)). It has a long history, dating back to the 19th century (Hoffman & Wolfe (1985)).

II. Review of Literature

Bryan Irina (1) The travelling salesman problem (TSP) seeks to minimize the cost of the route for a salesman to visit all the cities exactly once and return home. TSP is easy to describe, but difficult to solve because it requires finding a Hamiltonian cycle of minimum cost. There are several sufficient conditions for Hamilton city, but no necessary conditions. Some sufficient conditions are demonstrated in Dirac's and Ore's Theorems. Two approaches to solving the TSP are presented. The approaches use trees but in different ways. Although the first approach is systematically considering all possible ways to build Hamilton cycles, it is also searching for the cheapest way to build the cycles. It uses a "branch and bound" method to limit the number of different vertices in the search tree that must be inspected in search of minimal solution. The second approach is a quicker approximate algorithm for obtaining good (near-minimal) tours. It is followed by a theorem stating that at worst the approximate algorithm's tour is always less than twice the cost of a true minimal tour. This algorithm uses a successive nearest-neighbor strategy.

Milena, N., Karova, Vassil J. Smarkov, Stoyan Penev, (2) This paper introduces a flexible method for finding a solution to the travelling salesman problem using a genetic algorithm. The travelling salesman problem comes up in different situations in out world. It is a special kind of optimization problem. There had been many attempts to address this problem using classical methods, such as integer programming and graph theory algorithms with different success. The solution, which this paper offers, includes a genetic algorithm implementation in order to give a maximal approximation of the problem, modifying a generated solution with genetic operators.

Otman Abdoun, Jaafar Abouchabaka (3) Genetic algorithm includes some parameters that should be adjusting so that the algorithm can provide positive results. Crossover operators play very important role by constructing competitive Genetic Algorithms (GAs). In this paper, the basic conceptual features and specific characteristics of various crossover operators in the context of the Traveling Salesman Problem (TSP) are discussed. The results of experimental comparison of more than six different crossover operators for the TSP are presented. The experiment results show that OX operator enables to achieve a better solutions than other operators tested.

Moreno Angel Goñi (4) In this paper it is explained how to solve a fully connected N-City travelling salesman problem (TSP) using a genetic algorithm. A crossover operator to use in the simulation of a genetic algorithm (GA) with DNA is presented. The aim of the paper is to follow the path of creating a new computational model based on DNA molecules and genetic operations. This paper solves the problem of exponentially size algorithms in DNA computing by using biological methods and techniques. After individual encoding and fitness evaluation, a protocol of the next step in a GA, crossover, is needed. This paper also shows how to make the GA faster via different populations of possible solutions.

Liliana Ursache (5) Given a collection of cities and the cost of travel between each pair of them, the traveling salesman problem, or TSP for short, is to find the cheapest way to visit all the cities and to return to the starting point. The importance of the TSP is that it is representative of a larger class of problems known as combinatorial optimization problems. The TSP problem belongs in the class of combinatorial optimization problems known as NP-complete. It arises in numerous applications, such as circuit board drilling to insure time signals remain constant, cutting raw materials to minimize waste,

ISSN 2250-3153

clustering data arrays, analyzing crystal structures, scheduling and much more. Despite an intensive study by mathematicians, computer scientists, and others, it remains an open question whether or not an efficient general solution method exists. Although the complexity of the problem is still unknown, for over 50 years its study has led the way to improve solution methods in many areas of mathematical optimization. The TSP became a target for the GA community; several genetic-based algorithms were developed. This paper has the purpose to compare some of the most important approaches, paying particular attention to the representation and genetic operators .

III. RESEARCH METHODOLOGY

Genetic Algorithm

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

The GA works by updating a set P of originally randomly generated solution. The set P is canonically called a population. In this approach, the candidate solutions of the problem are represented by a vector of elements resembling the genetic structure of a chromosome. A new solution, an offspring, can be obtained by recombining a subset of two elements in P. recombination is done by using genetic operator, the most common ones being crossover and mutation. The crossover between two parents of which the first one is usually selected on a fitness basis and the other one randomly, is done by choosing a crossover point X at random and swapping the right- hand segments of the parent vector. The process is illustrated in heading Mutation. The mutation operator simply alters one or more elements of the candidate solution in order to increase the variability of the population. GA uses crossover as a primary operator and mutation as a secondary operator. The genetic heuristic framework of GA can be presented as follows:

1. Initialization a set P of solution
2.Construct a set Pc of solution by recombining pairs of randomly chosen elements in P
3.Construct a set Pm of solution by randomly modifying elements in Pc
4.Construct a new set P by extracting elements from Pm by following a Monte Carlo strategy with repetition. This means that a candidate solution in Pm can be selected to P more than once
5.If not end condition than go to step 2

The construction of the initial population and of the new groups for crossover and mutation operators is done by an intelligent encoder.

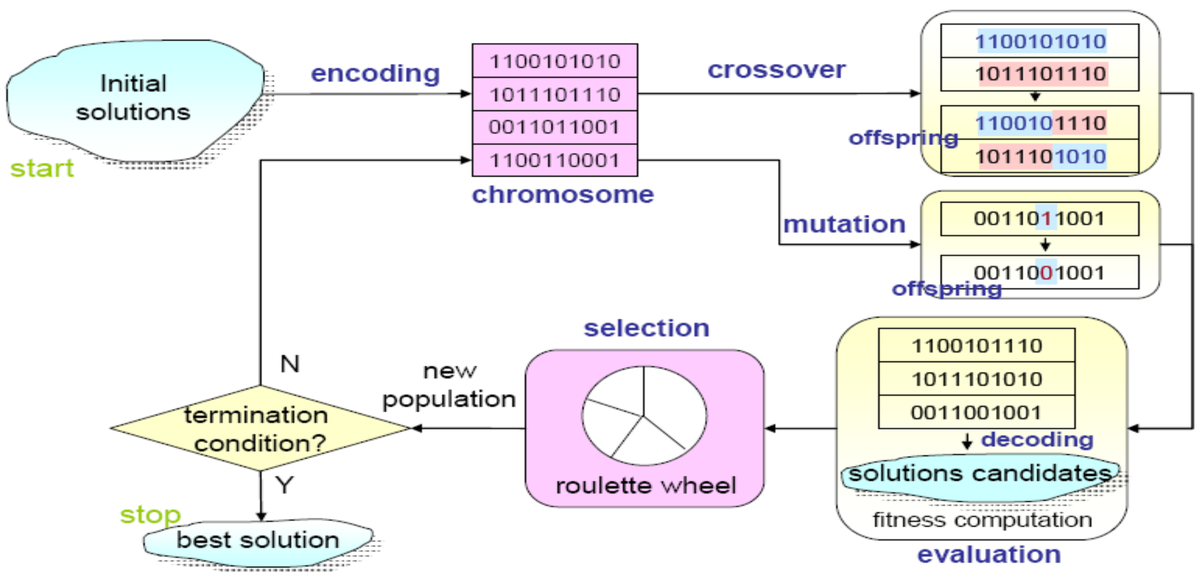


FIG. 1 (General structure of GA)

The construction of the initial population and of the new groups for crossover and mutation operators is done by an intelligent encoder.

Working of GA

Generally GA is used to solve the maximization problem but GA can also handle minimization problems. This can be done by choosing a fitness function suitably. The working of GA to solve

a problem consists number of steps, which are discussed in the following pages of this chapter.

Coding

In order to use GA to solve a particular problem, the variant, which are present in the given problem, are first coded in some string structure. Coding of the variables is not absolutely necessary. There exist some studies where GA is directly used on the variables themselves, but in simple GA coding is done most of the time. Binary coded strings having 1's and 0's are mostly used. The length of the string is usually determined according to the desired accuracy. The accuracy that can be obtained with a four bit coding is only approximately 1/16th of the search space. But as the string length is increased by one, the obtainable accuracy increases exponentially to 1/32th of the search space. It is not necessary to code all variables in equal substring length. The length of a substring representing a variable depends on the desired accuracy in the variable. Once the coding of the variable has been done, the corresponding point can be found by using a mapping rule. Thereafter, the function value at the point can be calculated by substituting that very point in the given objective function.

GA operator

The GA operators are used to perform certain function, which help to produce and select good offspring from, a set of candidate solutions. The various GA operators that are used generally for solving a given problem are given below.

Reproduction

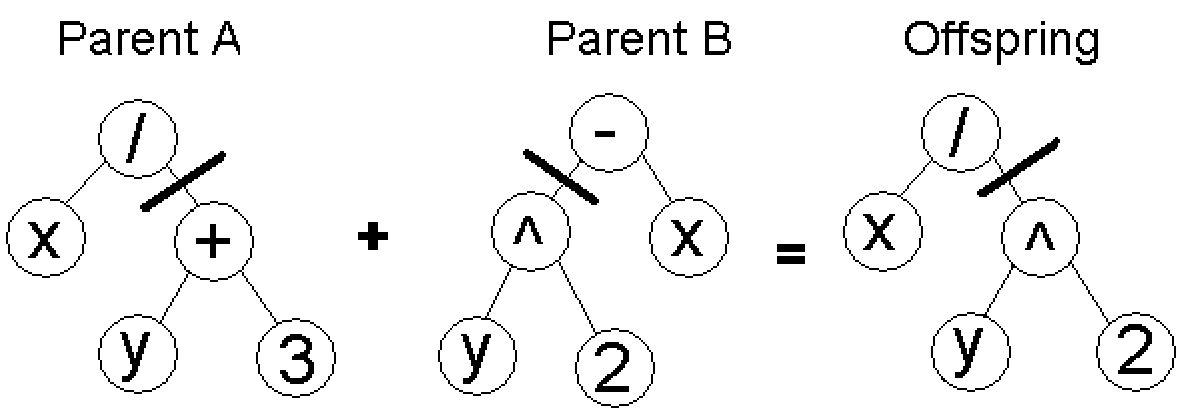
It is usually the first operator applied on a population. Reproduction select good string in a population and form a mating pool. This is why the reproduction operator is sometime known as the selection operator. **Selection** is the stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding (recombination or crossover). There are several generic selection algorithms, such as tournament selection and fitness proportionate selection (also known as roulette-wheel selection). But the essential idea in all of them is that above average string are picked from the current population and their multiple copies are inserted in the mating pool in a probabilistic population manner. The latter may be implemented as follows:

- The fitness function is evaluated for each individual, providing fitness values, which are then normalized. Normalization means multiplying the fitness value of each individual by a fixed number, so that the sum of all fitness values equals 1.
- The population is sorted by descending fitness values.
- Accumulated normalized fitness values are computed (the accumulated fitness value of an individual is the sum of its own fitness value plus the fitness values of all the previous individuals). The accumulated fitness of the last individual should of course be 1 (otherwise something went wrong in the normalization step).
- A random number *R* between 0 and 1 is chosen.
- The selected individual is the first one whose accumulated normalized value is greater than *R*.

There are other selection algorithms that do not consider all individuals for selection, but only those with a fitness value that is higher than a given (arbitrary) constant. Other algorithms select from a restricted pool where only a certain percentage of the individuals are allowed, based on fitness value.

Crossover

In the crossover operator, new string are created by exchanging information among strings of the mating pool. In most crossover operators, two strings are picked from the mating pool at random and some portions of the strings are exchanged between the strings. A single point crossover operator is performed by randomly choosing a crossing site along the string and by exchanging all bits on the right side of the crossing site. Multipoint crossover can also be used depending on the nature problem the two crossover sites, are exchanged. The crossing site is still chosen randomly.

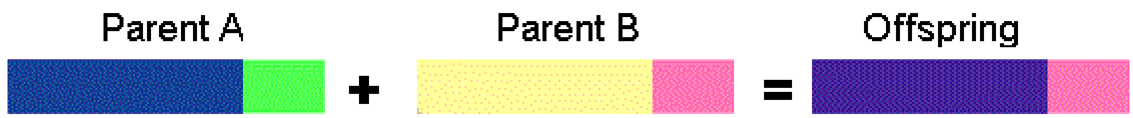


SIMPLE CROSSOVER OF TWO PARENTS

The two strings participating in the crossover operation are known as parent string and the resulting are known as children strings. It is intuitive from this construction that good substrings from parent strings can be combined to form a better child string, if an appropriate site is chose. Since the knowledge of an appropriate site is usually not known beforehand, a random site is often chosen. With a random site, the children strings produced may or may not have good substrings from parent strings, depending on whether or not the crossing site fall in the appropriate place.

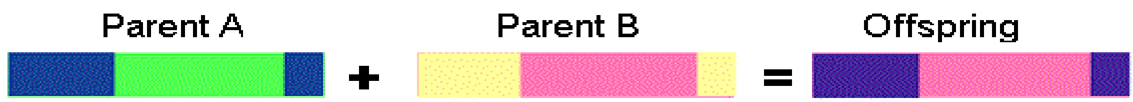
Single point crossover - one crossover point is selected, binary string from beginning of chromosome to the crossover point is copied from one parent, and the rest is copied from the second parent.

ISSN 2250-3153



11001011+11011111 = 11001111

- **Two point crossover** - two crossover point are selected, binary string from beginning of chromosome to the first crossover point is copied from one parent, the part from the first to the second crossover point is copied from the second parent and the rest is copied from the first parent



11001011 + 11011111 = 11011111

- **Uniform crossover** - bits are randomly copied from the first or from the second parent.

11001011 + 11011101 = 11011111

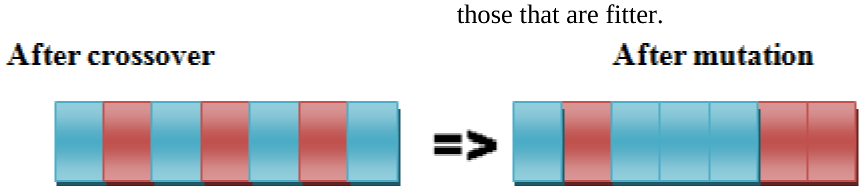
- **Arithmetic crossover** - some arithmetic operation is performed to make a new offspring

11001011 + 11011111 = 11001001 (AND)

Mutation

In genetic algorithms, mutation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next. It is analogous to biological mutation.

The classic example of a mutation operator involves a probability that an arbitrary bit in a genetic sequence will be changed from its original state. A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable tells whether or not a particular bit will be modified. The purpose of mutation in GAs is to allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. This reasoning also explains the fact that most GA systems avoid only taking the fittest of the population in generating the next but rather a random (or semi-random) selection with a weighting toward those that are fitter.



1010101 => 1011100

In Permutation Encoding:

(1 2 3 4 5 6 8 9 7) => (1 8 3 4 5 6 2 9 7)

IV. WORK PROCEDURE

Objectives of the Present Investigation

- To collect the data from the company
- Write a programme using MATLAB to optimize the drilling sequence problem by using the Genetic Algorithm (GA)
- Find out the least distance travel by tool.

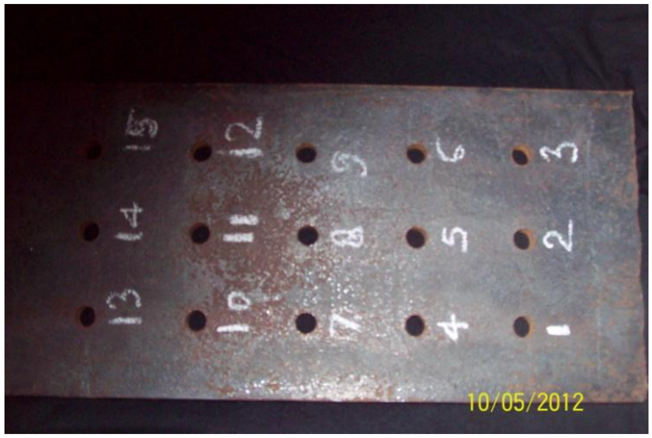


Fig.2 (Wok Piece Model)

Introduction of Software

In this research work 'MATLAB 7.8.0 (R2009a)' are used. MATLAB is the sort form of Matrix Laboratory and 7.8 is the version. MATLAB take the input data in the Matrix form so due to this quality it got the name MATLAB.

Programming Procedure for Modal Using MATLAB

The procedures of programming of modal using MATLAB software are described in the following steps:

1. Determination of the Position of the Point
2. Specify Ga Parameters
3. Initialize the Population to Run the Ga by Evaluating Each Population Member
4. Coding to Finding the Best Rout In The Population
5. Coding for Ga Operator
6.Code for Display The Graph And Result

V. RESULT & DISCUSSION

After finishing the coding of problem, now run the program. After running the program, we got two results on the graph and the minimum distance travel by the tool after the 64th combination. And the problem is solved. The minimum distance travel by the tool is 694.03 units. And the optimum path is 10-13-14-15-11-12-9-8-5-6-3-2-1-4-7 as shown in the fig.3.

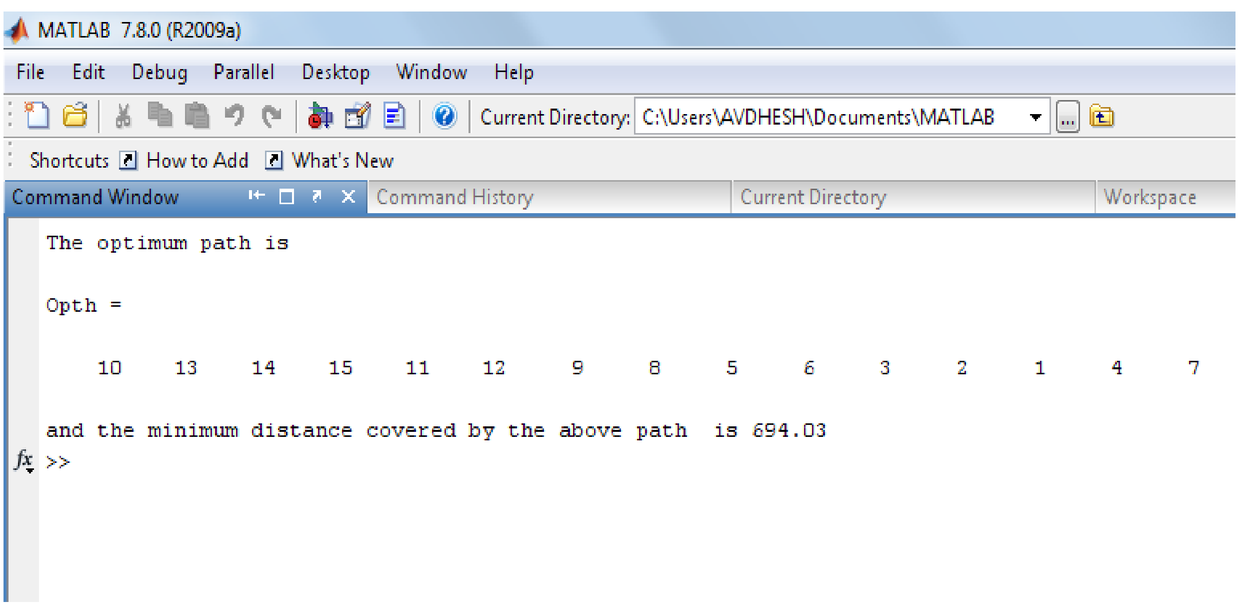


Fig.3: Result of program

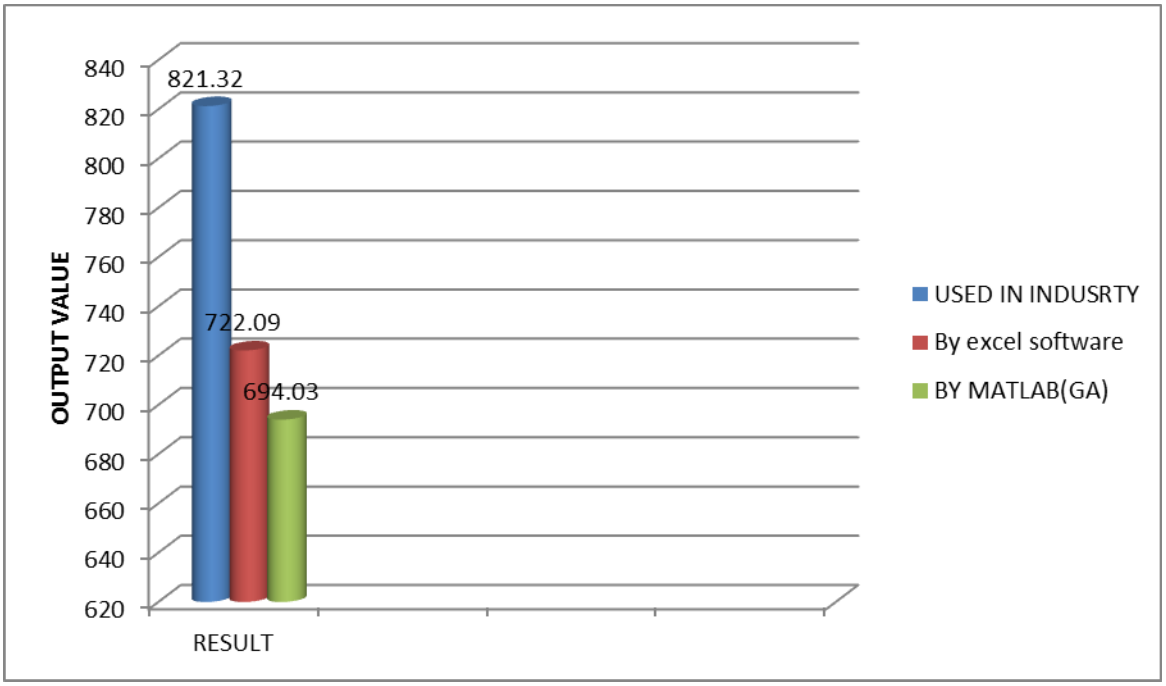


Fig.4

VI. CONCLUSION

The present research work deals with the study of TSP based 2D sequential problem followed by the development of model using MATLAB software. The company from where we take the data, is using the sequence 1-4-7-10-13-14-15-12-9-6-3-2-5-8-11-1 and finding the tool travel distance is 821.32 units and the excel sheet software, After running it gives that 722.09 units is the least distance travel by the tool and the path is 11-12-15-14-13-10-7-8-4-1-2-3-5-6-9-11 But in the present research work, MATLAB find out the best sequence is 10-13-1415-11-12-9-8-5-6-3-2-1-4-7. And the minimum tool travel distance is 694.03 units. This result is best among all 3 results described above So it is concluded that the minimum tool travel distance is 694.03 units followed by the sequence 10-13-14-15-11-12-9-8-5-6-3-2-1-4-7 By using this drilling sequence we can decrease the total work time and improve the efficiency.

VII. RECOMMENDATIONS FOR FURTHER WORK

In our research work, an attempt has been made towards the development of the model using MATLAB software to find out the minimum tool travel distance. Some of the areas which can be examined in future are described as follow:

1.The model of the process has been developed for the 2-D problems. So attempt can be made to develop the 3-D model.

2.The present work has been done for the minimization of tool travels without considering tool wear so attempt may be made with considering the tool wear.

3.The present work has been done for same size drill in diameter and depth. So attempt can be made for the varying diameter size and depth.

REFERENCES

[1] Annamalai, Murugappan, (2005).”Total completion time minimization in a drilling sequence problem considering tool wear: an ant algorithms approach”. Annamalai University.

[2] Arthur E.”, (2003). “Design and Application of Genetic Algorithms for the Multiple Traveling Salesperson Assignment Problems”, Doctor of Philosophy Thesis, Blacksburg, Virginia.

[3] Bryant, Kylie”, (2000). “Genetic Algorithms and the Traveling Salesman Problem”, Department of Mathematics, Harvey Mudd College.

[4] Dantzig G. B., Fulkerson R., and Johnson S. M., (1954). “Solution of a large-scale traveling salesman problem*”*, Operations Research 2 , 393-410.

[5] Deb, K. (1999). An introduction to genetic algorithms. 293-315.

[6] Deb, K., Reddy, A. R. and Singh, (2003). “Optimal scheduling of casting sequence using genetic algorithms” .Journal of Materials and Manufacturing Processes, 18(3). 409-432.

[7] E.H.L. Aarts & J. Korst (1989), “Boltzmann Machines for Traveling Salesman Problems”European Journal of Operations Research 39, 79–95. [155]

[8] Fang-Geng Zhao, Jiang-Sheng Sun, Su-Jian Li, Wei-Min Liu, (2009). “A Hybrid Genetic Algorithm for the Traveling Salesman Problem with Pickup and Delivery”. International Journal of Automation and Computing.

[9] Liliana Ursache, (2007),**“**Representation models for solving TSPwith genetic algorithms “Stud.Cercet.Ştiint., Ser.Mat., 17 (2007), SupplementProceedings of CNMI 2007, Bacău, Sept.2007, pp. 291-298.

[10] Goldberg, D. E., (1989)."Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley.

[11] Moreno Angel Goñi,(2008),**“**Solving Travelling Salesman Problem In A Simulationof Genetic Algorithms With Dna”International Journal "Information Theories & Applications" Vol.15 / 2008 – 357

[12] Holland, John H., (1992). “Adaption in Natural and artificial systems”, the Mit Press. Jean-Yves Potvin, “Genetic Algorithms for the Traveling Salesman Problem”, A research paper. University of Montreal, Canada.

[13] Leng, L. T. (1999)."Guided genetic algorithm," Doctoral Dissertation. University of Essex.

[14] Lin, S., Kernighan, B.W.(1973). “An effective heuristic for the traveling salesman problem”.Operation Research **21:** 498–516.

[15] Milena, N., Karova, Vassil J. Smarkov, Stoyan Penev, (2005). "Genetic operators crossover and mutation in solving the TSP problem", Technical University Varna, International Conference on Computer Systems and Technologies - CompSysTech'.

[16] Ray, Shubhra Sankar · Bandyopadhyay,Sanghamitra · Pal,Sankar K., (2007)."Genetic operatsors for combinatorial optimization in TSP and microarray gene ordering".

[17] Otman Abdoun, Jaafar Abouchabaka,(2011),"A Comparative Study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Travelling Salesman Problem",International Journal of Computer Applications (0975 – 8887) Volume 31–

[18] Taha, H.A. (1998), "Operation Research- An introduction", Prentice Hall, Upper Saddle River, New Jersey.

[19] www.thecodeproject.com

[20] www.knowledgerush.com

[21] Nagi, I.S., (2006), "Cutting Stock Optimization Using Genetic Algorithm", A dissertation of M.E., Computer Integrated Manufacturing, PEC Chandigarh**:** 32-34.

[22] Bryan Irina, (2009),"The Traveling Salesman Problem",1-17

[23] Price and Storn,(1995), "Differential Evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces", ICSI technical report.

[24] Price and Storn,(1997), "Differential Evolution: A simple evolution strategy for fast optimization". Dr. Dobb's Journal.

AUTHORS

First Author – Abdhesh Kumar, M.Tech Scholar , Dept. of Mechanical Engineering , NIET Greater Noida (U.P), India., Email id: akjha@rediffmail.com, Mob No: +91- 9990062215