ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

A Comparative Study of Genetic Algorithm with the KNN Evolutionary Optimization Algorithm in Data Mining

Dr. M. Subha

Asst. Professor, Dept. Information Technology, Dr.NGP Arts and Science College(Autonomous), Coimbatore, India

Abstract: Evolutionary optimization algorithms have been proved to be good solutions for many practical applications. They were mainly inspired by natural evolutions. However, they are still faced to some problems such as trapping in local minimums. This paper proposes the comparative study of inspired algorithms like Stem Cells Algorithm (SCA), Ant Colony Optimization (ACO) algorithm with the K-nearest neighbor algorithm (KNN) to reduce the local minima by using benchmark functions in data mining.

Keywords: Evolutionary inspired optimization algorithm, local minima, benchmark functions.

I. INTRODUCTION

The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated. The fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

A typical genetic algorithm requires:

1. a genetic representation of the solution domain,
2. a fitness function to evaluate the solution domain.

Genetic algorithms are simple to implement, but their behavior is difficult to understand. In particular it is difficult to understand why these algorithms frequently succeed at generating solutions of high fitness when applied to practical problems. The building block hypothesis (BBH) consists of:

1. A description of a heuristic that performs adaptation by identifying and recombining "building blocks", i.e. low order, low defining-length schemata with above average fitness.
2. A hypothesis that a genetic algorithm performs adaptation by implicitly and efficiently implementing this heuristic.

Although good results have been reported for some classes of problems, skepticism concerning the generality and/or practicality of the building-block hypothesis as an explanation for GAs efficiency still remains. Indeed, there is a reasonable amount of work that attempts to understand its limitations from the perspective of estimation of distribution algorithms.

There are limitations of the use of a genetic algorithm compared to alternative optimization algorithms: Repeated fitness function evaluation for complex problem is often the most prohibitive and limiting segment of artificial evolutionary algorithms. Finding the optimal solution to complex high-dimensional, multimodal problems often requires very expensive fitness function evaluations. Genetic algorithms do not scale well with complexity. That is, where the number of elements which are exposed to mutation is large there is often an exponential increase in search space size. This makes it extremely difficult to use the technique on problems such as designing an engine, a house or plane. In order to make such problems tractable to evolutionary search, they must be broken down into the simplest representation possible. Hence we typically see evolutionary algorithms encoding designs for fan blades instead of engines, building shapes instead of detailed construction plans, and airfoils instead of whole aircraft designs. The second problem of complexity is the issue of how to protect parts that have evolved to represent good solutions from further destructive mutation, particularly when their fitness assessment requires them to combine well with other parts. For specific optimization problems and problem instances, other optimization algorithms may be more efficient than genetic algorithms in terms of speed of convergence. The suitability of genetic algorithms is dependent on the amount of knowledge of the problem. Well known problems often have better, more specialized approaches.

II. STEM CELLS OPTIMIZATION ALGORITHM

Most naturally-based optimization algorithms are introduced because of their simplicity and because they have been shown to effectively solve complex optimization problems in nature. Stem cells are found in all multi-cells of body organs where they are able to

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

become a full organ. The stem cells optimization algorithm is like other optimization algorithms in that it is based on population and the idea of evolutionary process, but it is different in that it uses minimal constraints and has a simpler implementation than the others. This algorithm converges faster than other Optimization algorithms because of its simplicity and its ability to escape from local minima. Population is placed in a Range of members (stem cells forming initial population) in this algorithm and it starts with the minimum. Considering That each population member of each stem cell (e.g. in genetic algorithm, chromosomes and in swarm algorithm, etc.) indicates an optimal answer for all considered problems, increasing the population at each iteration is related to the problem space, but defining a large population in this type of algorithms results in abundant iterations to achieve optimal response which consequently raises many problems. Mean while defining the population in an interval and increasing the population according to the space of the considered problem are especially advantageous in implementation by resulting in few iterations in simple problems and increasing the speed of convergence. Considering the goal of all optimization algorithms, including stem cells algorithms, which is to obtain a response with respect to variables of the problem, a matrix of variables should be formed at the beginning of the process.

III. SWARM INTELLIGENCE (ANT COLONY OPTIMIZATION ALGORITHM

Swarm intelligence is a sub-field of evolutionary computing. Ant colony optimization (ACO) uses many ants (or agents) equipped with a pheromone model to traverse the solution space and find locally productive areas. Particle swarm optimization (PSO) is a computational method for multi-parameter optimization which also uses population-based approach. A population (swarm) of candidate solutions (particles) moves in the search space, and the movement of the particles is influenced both by their own best known position and swarm's global best known position. Like genetic algorithms, the PSO method depends on information sharing among population members. In some problems the PSO is often more computationally efficient than the GAs, especially in unconstrained problems with continuous variables.

IV. K-NEAREST NEIGHBOR ALGORITHM (KNN)

K-nearest neighbor algorithm (KNN) is part of supervised learning that has been used in many applications in the field of data mining, statistical pattern recognition and many others. KNN is a method for classifying objects based on closest training examples in the feature space.

An object is classified by a majority vote of its neighbors. K is always a positive integer. The neighbors are taken from a set of objects for which the correct classification is known. It is usual to use the Euclidean distance, though other distance measures such as the Manhattan distance could in principle be used instead.

The algorithm on how to compute the K-nearest neighbors is as follows:

Determine the parameter K = number of nearest neighbors beforehand. This value is all up to you. Calculate the distance between the query-instance and all the training samples. You can use any distance algorithm. Sort the distances for all the training samples and determine the nearest neighbor based on the Kth minimum distance.

Since this is supervised learning, get all the Categories of your training data for the sorted value which fall under K. Use the majority of nearest neighbors as the prediction value. The results obtained by applying the different clustering algorithms to different test data sets with the cost function calculated as follows:

$$\text{Cost function} = \sum_{j=1}^{N} \min\left(Y^{j} - Z^{i}\right) \text{ for } i = 1, 2, \ldots, K \quad (10)$$

Y = input data,
Z = cluster center,
N = number of input data,
K = number of cluster center.

Here, computed the Euclidean distances between each input data and all cluster centers and then determine the minimum of these distances and finally sum all minimums for all input data. The mean, minimum and maximum cost function values were computed for each algorithm over 100 different runs on vowel data set. As can be seen, the KNN algorithm demonstrates better results in obtaining lower mean value with minimum difference between min and max values for vowel datasets.

V. RESULTS

The results obtained are shown below. As can be seen, the KNN algorithm has better performance than the other algorithms. The running time for each algorithm is the time when the algorithm achieves its best result. The KNN typically took less time than the other algorithms to achieve its best result. It is mostly due to the fact that KNN has fewer constraints, fewer parameters to be computed and fewer loops, which causes it achieves the result in far fewer iterations than other algorithms.

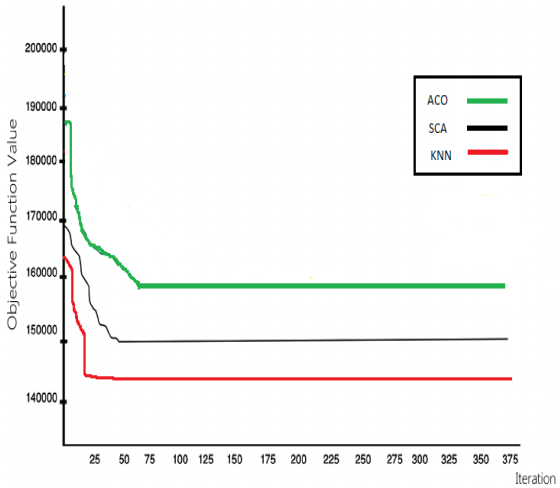

Fig.1 Comparative results of algorithms

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICITCSA 2017

Pioneer College of Arts and Science, Coimbatore

Vol. 6, Special Issue 1, January 2017

VI. CONCLUSION

A comparative study on optimization algorithms with genetic algorithm and KNN algorithm in data mining. Only one benchmark function based on Euclidian distance is used for comparative study. In future, instead of one, more benchmark functions can be used to show the evolutionary optimization algorithm for data mining.

REFERENCES

[1] Eiben, Agoston; Smith, James. "Introduction to Evolutionary Computing",Springer.
[2] D. Coomans; D.L. Massart "Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-Nearest neighbour classification by using alternative voting rules".
[3] Ramaswamy, S.; Rastogi, R.; Shim, K. , "Efficient algorithms for mining outliers from large data sets". Proceedings of the ACM SIGMOD international conference on Management of data.
[4] Taherdangkoo, Mohammad Paziresh, Mahsa, Yazdi, Mehran, Bagheri, Mohammad, "An efficient algorithm for function optimization: modified stem cells algorithm".
[5] Taherdangkoo, Mohammad Yazdi, Mehran Bagheri, Mohammad, "A powerful and efficient evolutionary optimization algorithm based on stem cells algorithm for data clustering".
[6] Whitley, Darrell, "A genetic algorithm tutorial", Satistics and computing

Datasets used in experiments and their characteristics

Dataset Name	No. of Objects	No. of features	No. of Classes
Vowel	871	3	6

Parameters used in the clustering algorithms

Algorithm	Parameter	Value
ACO	Number of ants	50
	Probability threshold for maximum trail	0.95
	Local search probability	0.01
	Evaporation rate	0.01
SCA	Number of stem cells	20
	Z_{max}	0.98
	Z_{min}	0.01
	Number of iterations	200
KNN	Number of iterations	100

Clustering results obtained by applying the algorithms for 100 various runs on vowel dataset

Algorithm	F_{mean}	F_{min} = best	F_{max} = worst	Standard deviation
ACO	159668.442	157996.333	160113.226	28100.3
SCA	150003.662	149988.333	150024.277	6.871
KNN	150004.243	140021.235	114005.43	4.645