Review of Evolutionary Algorithms Based on Parallel Computing Paradigm

Bhanu Prakash Lohani [1], Vimal Bibhu[2], Ajit Singh[3]
[1]*Research Scholar Department of CSE, UTU, Dehradun*
[2]*Assistant Professor, Department of CSE, Amity University Gr Noida*
[3] *Associate Professor, Department of CSE, BTKIT, Dwarahat, Uttarakhand*

Abstract

Evolutionary algorithms are used to find the optimal solution for the real world problem based upon the concept of biological evolution. When the data size is very large and fitness function is complex in nature at that time sequential evolutionary algorithms fails to give satisfactory result in the given time domain. Hence the parallelization is done to that can handle the complex real world optimization problems in the reasonable time. Parallelization is a process which is used to do speed up to the existing system to provide the result in the required time frame. In this paper we have presented the review of various evolutionary algorithms studied for the literature review for the research work. We had also summarized Why parallelization is needed for any evolutionary algorithm how we can implement the parallelization with the help of Hadoop Map reduce architecture. We have also presented a comparative result analysis of Particle swarm optimization algorithm. The parallel version of Particle swarm optimization is implemented with the help of Hadoop MR Architecture.

Keywords

Evolutionary algorithm, Swarm, Parallelization, optimization, Hadoop, biological evolution, mutation.

I. INTRODUCTION

Evolutionary algorithm is a subset of evolutionary computing uses the mechanism & style implemented by biological evolution, which include the concept of reproduction, mutation, recombination &selection. Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the quality of the solutions. Evolution of the population then takes place after the repeated application of the above operators. Artificial evolution (AE) describes a process involving individual evolutionary algorithms; EAs are individual components that participate in an AE. Evolutionary algorithms are in the categories of computational intelligence .In the below figure 1 & figure 2 the location of the different families of the evolutionary algorithm are shown and also the process steps for the execution of any evolutionary algorithm are shown [1]. The concept of evolutionary computing is shown in the form of flow chart where we can see how value is calculated and when we have to terminate the process of computing.

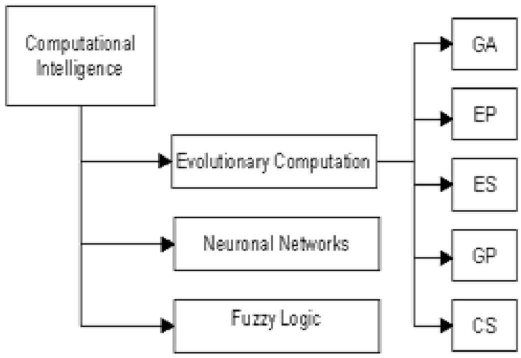


Fig.1 Location of different families of Evolutionay Algorithms

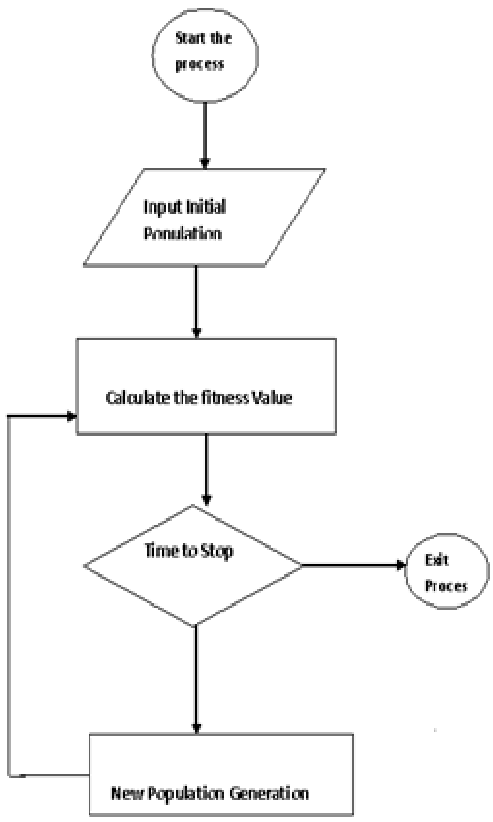


Fig.2 Process Steps for Evolutionay Algoritm

II. EVOLUTIONARY ALGORITHM – A REVIEW

In this paper we have summarized the various categories of evolutionary algorithms and their functionalities with respect to the population, behaviour and the applications. We had gone through with the following algorithms such as Particle swarm optimization algorithm, Genetic algorithm, Ant colony optimization algorithm, Gravitational based search algorithm, Artificial Bee Colony Algorithm, Cuckoo Search.

A. *Particle Swarm optimization:*

PSO is an optimization algorithm which is based on population and developed by Eberhart & Kennedy in 1995. In PSO the basic concept is driven from the general behaviour of bird flocking and fish schooling. The problems which can be transformed to optimization problem are solved by PSO, Fast coverage is the main property of PSO which compare the favourably with respect to the many other global optimization problem i.e. Genetic algorithm .In PSO the population is treated as swarm and the individuals are referred as particles, where each particle represents a potential solution of the problem. As soon as a particle find a correct direction to the move (stimulated result direction), other particles are notified and will be able to catch the same direction. The particles move in the space (world) and pass their position to others and fit their own position based upon the better position with respect to the correct result required [7].

B. Ant Colony Optimization

The Ant Colony Optimization is based upon the concept of Inherent Parallelization. This algorithm is used where we need crisp result. Ant colony optimization is more applicable when the source and destination for the problem is predefined [3].

C. Genetic Algorithm

Genetic algorithm are adaptive heuristic search algorithm based upon the concept of evolution, natural selection. In the implementation of genetic algorithm three operators are used selection, crossover and mutation [4].

D. Cuckoo Search

Cuckoo search algorithm is a metaheuristics based optimization algorithm. It is inspired by obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds. Cuckoo search algorithm is based upon three concepts:

1. Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest.
2. The best nests with the high quality of eggs will carry to the next generations.
3. The number of available host nest is fixed and if a host bird identifies the cuckoo egg with the probability of P(n)=0,1then the host bird can either throw them away or abandon them and build a new nest.[5],[6]

E. Artificial Bee Colony Algorithm

Artificial bee colony algorithm is a popular optimization algorithm based upon the concept of intelligent foraging behavior of honey bee swarm. In ABC algorithm three colonies are in concern employed, onlooker and scout bees. In this algorithm the basic operator and approach are of nature inspired algorithm [7]

III. NEED OF PARALLELIZATION / HYBRIDIZATION

In some situations the conventional evolutionary algorithm fails to obtain a convenient or optimal solution for the problem. At that time we need to design a hybrid evolutionary algorithm with other optimization algorithm and other machine learning algorithms or some heuristic techniques.. Some of the possible reasons for hybridization or parallelization of evolutionary algorithms are as follows:

1. To speed up the performance of the existing evolutionary algorithm.
2. The quality concern where we need to updated the result of evolutionary algorithms.
3. Parallel processing of data where the size of input data set is too larger.
4. To improve the execution time with respect to the conventional algorithm.

IV. HADOOP MAP REDUCE ARCHITECTURE

In this paper we have done some experimental work by the help of Hadoop Map Reduce architecture, so in here we are presenting the brief functionality of the Hadoop Map Reduce architecture and its functionality. Hadoop Map Reduce frame work is a software system to perform a large amount of data in parallel manner. Hadoop Map Reduce architecture also provides the analysis result in a reliable and fault tolerant manner for a large amount of data set.[8] The Map Reduce architecture job is to break the Input data into the independent chunks which are processed by the Mapper function in a parallel manner after that the software framework sorts the output of the mapper which is used as an Input for the Reducer function.

Then these Input and outputs are stored in the Hadoop file system. Distributed Data Processing is done the Map Reduce function of the architecture and the Distributed data storage is done by the HDFS file system. The MapReduce framework consists of a single master Job Tracker and one slave Task Tracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master.

The working model of Hadoop server side and map reduce architecture is shown in figure3,4 .[9]. In the following figure the role of Hadoop server is shown that how distributed data processing is done at server end and in the Map reduce working model the basic things are mapper , reducer and combiner.

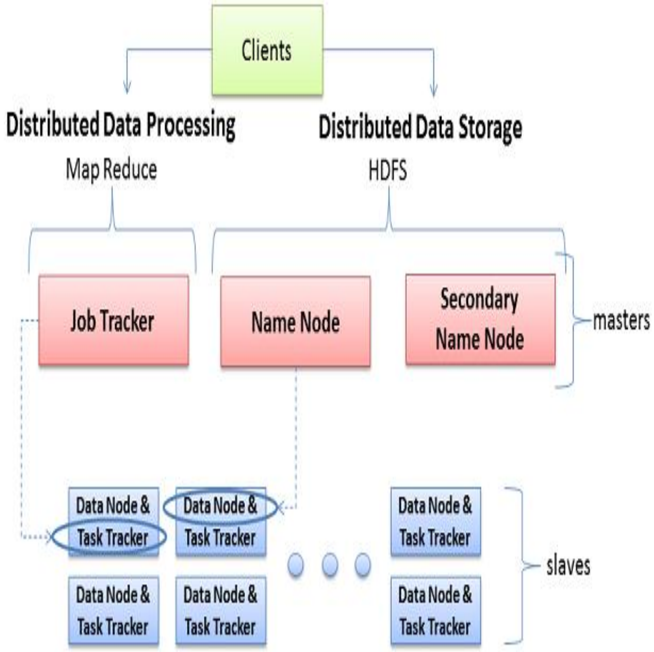


Fig.3. Hadoop Server Role.

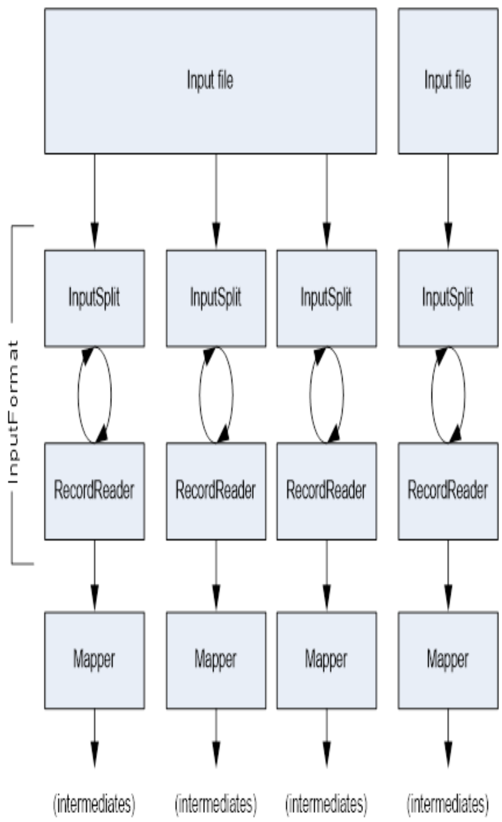


Fig.4.Map Reduce working model

V. EXPERIMENTAL RESULT

Comparison of the result of PSO and Parallel PSO using Hadoop MR Architecture: In a serial PSO, there are several particles that explore a search space, looking for the optimal parameters in terms of lowest associated "cost" in a user-specified objective function. This cost is just a scalar value that the user is trying to minimize. Each particle is represented by a single parameter set that evolves over iterations in response to two things:

1) the "cost" associated with those particular parameters when evaluated in the objective function.

2). the "cost" associated with the particle that yielded the lowest cost thus far across all particles in the swarm (global best).

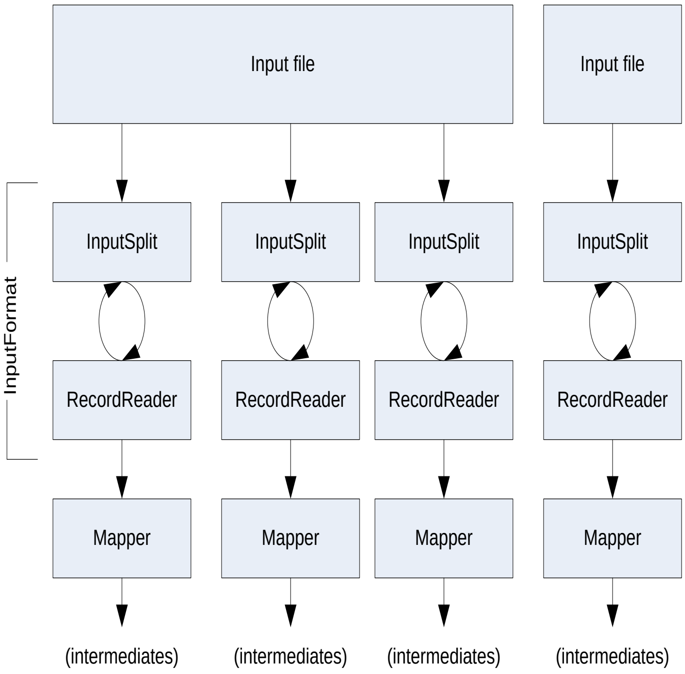
In the parallel implementation of Particle swarm optimization algorithm we have used the Map reduce architecture of Hadoop in which three functions are used Input splitter, Mapper and reducer. The PSO mapper finds the new position and velocity of the particle and evaluates the function at the new point. It then calls the update method of the particle with this information. In addition to modifying the particle's state to reflect the new position, velocity, and value, this method replaces the personal best if a more fit position has been found. The values list contains the newly updated particle and a message from each neighbour. The PSO reducer combines information from all of these messages to update the global best position and global best value of the particle.

Parallel PSO Implementation

```
1.  cost = calculate fitness(p.current_position);
                         if cost < global_best_cost:
                    global best =    p.current_position;
                                  global_best_cost = cost
2-                     if cost < p.personal_best_cost:
                                     p.personal_best = p;
                          p.personal_best_cost = cost;
  p.velocity = p.velocity + C1 * (global_best -
  p.current_position) + C2 * (p.personal_best
                              p.current_position);
 3 p.current_position= p.current_position +
                                      p.velocity;
```

Where C1 is a constant representing how heavily to weight the global best particle in the next position shift. C2 is a constant representing how heavily to weight the particle's own personal best in the next position shift.

Experimental Result: First we will take input of the entire population then output of first iteration is taken as input of second iteration the data set for the input is primarily taken from the NASA data. In the input part we need the information of each particle which is called the key value in numerical existence and the particle state which represented by a string . the common format is in the form of: The pair of key and value ie <k1,v1>. The state of a particle consists of its dependents list, position velocity, value, personal best position, personal best value, global best position and global best value .the state string is a semicolon- separated list of fields. In the following figure5 we can see the result in the serial PSO is constant but is the Parallel version of it the result is better with respect to the number of processor used in the following figure the result is compared with respect to the two dimensions one is execution time and another is number of processor used.



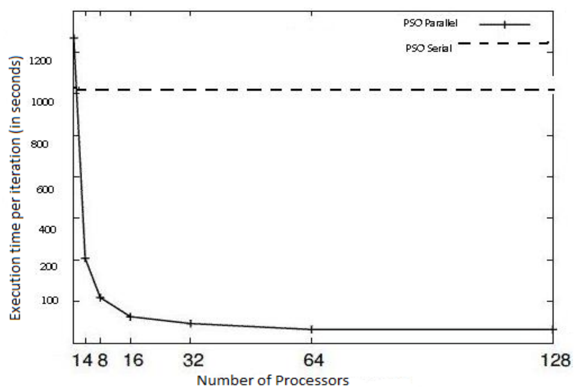


Fig.5 Comparison of the result of PSO and Parallel PSO wrt processor and execution time(in second)

VI. CONCLUSION

In this paper the review of various evolutionary algorithms are presented and also summarized when the sequential evolutionary algorithms are fails to obtain a optimum result and how we can enhance the capability of evolutionary algorithms so that the result can achieved in required time frame. The Hadoop MR architecture is also summarized which is used for parallelization process for a large amount of data. We have also presented the applications of Particle swarm optimization algorithm for the data set of NASA for calculating the values used in COCOMO model. The parallelization steps are also presented by which the PSO will give the result in better time and speed.

REFERENCES

[1] Yue-Jiao, Wei-Neng Chen, Zhi-Hui Zhan, Jun Zhang, Yun Li, and Qingfu Zhang. "Distributed evolutionary algorithms and their models: A survey of the state-of-the-art." Applied Soft Computing (2015).

[2] McNabb, Andrew W., Christopher K. Monson, and Kevin D. Seppi. "Parallel pso using mapreduce." In Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, pp. 7-14. IEEE, 2007.

[3] B. Bullnheimer, G. Kotsis, and C. Strauss. Parallelizati on Strategies for the Ant System. Technical Report POM 9/97, University of Vienna, 1997.

[4] K.S.Tang, K.F. Man,S.Kwong and P.J. Fleming, “GA Approach to Multiple Objective Optimization for Active Noise Control,”Algorithms and Architecturesfor Real-Time Control 95,pp. 13-19, Belgium,31 May-2 Jun 1995.

[5] Yang, Xin-She, and Suash Deb. "Engineering optimization by cuckoo search." International Journal of Mathematical Modeling and Numerical Optimization 1, no. 4 (2010): 330-343.

[6] Venkata Vijaya Geeta. Pentapalli,Ravi Kiran Varma P"Cuckoo Search Optimizationand its Applications: A Review"Vol. 5, Issue 11, November 2016

[7] Artificial Bee Colony (Abc), Harmony Search And Bees Algorithms On Numerical Optimization, D.Karaboga, B. Akay,Erciyes University, The Dept. Of ComputerEngineering, 38039, Melikgazi, Kayseri, Turkiye.

[8] del Río, Sara, Victoria López, José Manuel Benítez, and Francisco Herrera. "On the use of MapReduce for imbalanced big data using Random Forest." Information Sciences 285 (2014): 112-137.

[9] Alham, Nasullah Khalid, Maozhen Li, Yang Liu, and Suhel Hammoud. "A MapReduce-based distributed SVM algorithm for automatic image annotation." Computers & Mathematics with Applications 62, no. 7 (2011): 2801-2811

[10] Wu, Xindong, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. "Data mining with big data." Knowledge and Data Engineering, IEEE Transactions on 26, no. 1 (2014): 97-107.

[11] Chen, CL Philip, and Chun-Yang Zhang. "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data." Information Sciences 275 (2014): 314-347..

[12] Xu, Xingjian, Zhaohua Ji, Fangfang Yuan, and Xiaoqin Liu. "A Novel Parallel Approach of Cuckoo Search using MapReduce." In 2014 International Conference on Computer, Communications and Information Technology (CCIT 2014). Atlantis Press, 2014.

[13] Moraes, Antonio OS, et al. "A robust parallel algorithm of the particle swarm optimization method for large dimensional engineering problems." Applied Mathematical Modelling 39.14 (2015): 4223-4241.

[14] He, Yaobin, Haoyu Tan, Wuman Luo, Huajian Mao, Di Ma, Shengzhong Feng, and Jianping Fan. "Mr-dbscan: An efficient parallel density-based clustering algorithm using MapReduce." In Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on, pp. 473-480. IEEE, 2011.