Data Search Optimization Using Bio-Inspired Techniques

Dissertation

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY

IN

SOFTWARE TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SUBMITTED BY

AMIT KUMAR VERMA
ROLL NO: 2K13/SWT/03

MAJOR PROJECT REPORT II
(Paper Code: CO 821)

UNDER THE GUIDANCE OF

DR. KAPIL SHARMA

SHAHBAD DAULATPUR, MAIN BAWANA ROAD, NEW DELHI, DELHI

110042 INDIA

DELHI TECHNOLOGICAL UNIVERSITY
NEW DELHI

STUDENT DECLARATION

I hereby undertake and declare that this submission is my original work and to the best of my knowledge and believe, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of any Institute or other University of higher learning, except where due acknowledgement has been made in the text. Project work and published paper associated to the chapters are well discussed and improved under the guide supervision.

DATE:
SIGNATURE:

AMIT KUMAR VERMA

ROLL NO: 2K13/SWT/03

DELHI TECHNOLOGICAL UNIVERSITY
NEW DELHI

CERTIFICATE

This is to certify that the thesis entitled “**Data Search Optimization Using Bio-Inspired Techniques**”, is a bona fide work done by Mr. AMIT KUMAR VERMA in partial fulfilment of requirements for the award of Master of Technology Degree in software technology at Delhi Technological University (New Delhi) is an authentic work carried out by him under my supervision and guidance. The matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma to the best of my knowledge.

DATE:

SIGNATURE:

Dr. Kapil Sharma
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY

ACKNOWLEDGMENT

With the immense pleasure here I am presenting my work on "**Data Search Optimization Using Bio-Inspired Techniques**". I take this opportunity to thank my guide, Dr. Kapil Sharma, for guiding me and providing me with all the facilities, which paved way to the successful completion of this work. His scholarly guidance and invaluable suggestions motivated me to complete my thesis work successfully. I am thankful to my friends and colleagues who have been a source of encouragement and inspiration throughout the duration of this thesis. I am also thankful to the SAMSUNG who has provided me opportunity to enroll in the M.Tech Program and to gain knowledge through this program. This curriculum provided me knowledge and opportunity to grow in various domains of computer science. Last but not least, I am thankful to all the faculty members who visited the Samsung premises to guide and teach. Their knowledge and efforts helped me to grow and learn in the field of computer science. This project has provided me knowledge in the area of Big Data Analysis, Genetic Algorithms and Bio Inspired techniques also helped me in understanding the concept of clustering and MapReduce framework for development in data science domain. I have been given sufficient time and guidance to complete my project under timeline defined by the university.

AMIT KUMAR VERMA

ROLLNO: 2K13/SWT/03

ABSTRACT

All the information available these days are available in the form of digital data, whether it is network information, sensor information, healthcare information or from some other field. With the deluge of data every day the requirement to store and process the data is increasing. So technology to store, process and analyze the data should also improve with the same stride. Researchers are innovating new ideas to process the data faster with new algorithms. There has been many research in the area of Data Science which proposed the parallel processing architecture, some tools and framework has been introduced to implement the parallel processing. But apart from parallel processing, optimization of algorithm on each node of parallelism is also very important. There are many activities in the nature which are happening perfectly optimized, without any external control or without any centralized structure. And sometime these activities have been proven to solve the most difficult problems observed by the traditional computing method. In this thesis we have presented a solution inspired by nature to optimize the Data Search for shortest possible path among given locations. The proposed algorithm has been applied on existing parallel processing framework "MapReduce" with the slight modification in the MapReduce architecture to find the global best among local best produced by each Map node. So MapReduce, which was twostep process earlier has been modified to three step process Map-Reduce-Reduce. And for finding shortest path inspiration has been taken from Bottlenose dolphin. Bottlenose Dolphins are well known for their intelligence and communication; they use echolocation to identify the prey location and calculate the shortest path with the group effort of all the dolphins in the group. Utilizing this behavior of bottlenose dolphin a new heuristic approach has been developed and implemented in this thesis to enhance the searching process for the shortest path.

Table of Contents

Table of Figures

Introduction

1.1 Scope and Motivation

Today’s era can be considered as era of digitalization. Every moment trillion GBs of data is generated and stored, which is increasing day by day. It is expected that this data will grow by 40% every year [1]. It can be personal data (social media, voice recording, text messages, personal emails etc), corporate data (system back-up, daily transactions, camera recordings, sensor recordings etc), digital contents (photographs, music, movies etc). Storing such big data is not useful unless there is appropriate system to analyze the data at later stage, search some useful record or extract some useful information from stored data. Existing systems are not scalable enough to manage such big data and process it with in the acceptable time frame. So we need some new algorithms and infrastructure to manage the data.

1.2 Scope of Data Search and Analysis

Data explosion propelled by the appearance of online social websites, net, and international-scale communications has rendered data-driven statistical learning burgeoningly significant. Any moment around the world, large volumes of statistics are generated by means of nowadays ubiquitous conversation, imaging, and cellular gadgets together with cellular telephones, surveillance cameras and drones, scientific and e-trade structures, in addition to social networking sites. The term big data is devised to explain this statistic surge and, quoting a current press article, “their effect is being felt everywhere, from commercial enterprise to technological know-how, and from government to the humanities” [2]. large economic growth and development inside the best of life hinge upon harnessing the potential blessings of mining big data [2]. Mining extraordinary volumes of facts guarantees to restrict the spread of epidemics and maximize the percentages that online advertising campaigns move viral [35]; to discover trends in financial markets, visualize networks, recognize the dynamics of emergent social-computational systems, in addition to guard crucial infrastructure such as the internet's spine network, and the power-grid.

1.3 Big Data Challenges and Opportunities

At the same time as Big Data come with "huge advantages," there are formidable demanding situations in coping with big-scale data units. First, the sheer volume and dimensionality of statistics make it regularly not possible to run analytics and traditional inferential methods the use of standalone processors. Distributed learning with parallelized multicores is favored, whilst the records themselves are stored within the cloud or decentralized file system as in MapReduce/Hadoop. Consequently, there is a quick requirement to explicitly account for the memory load, query, and message overhead.

In some instances, confidentiality issues prevent disclosing the full data records, by which only preprocessed statistics has been allowed to be communicated via well designed channels. Generally, because of different origins, often considerable size of big data gets missed out or incomplete. huge-scale statistics necessarily contain corrupted measurements, communication defects, or even suffer from cyberattacks as the procurement and transportation cost per record is pushed to the lowest. Moreover, as many of the information sources constantly generate statistics in real time, analytics which has been done online, should be fast enough to cater the requirement in real time. To explain this, it can be elaborated as high excellence solution obtained slowly can be less useful than a medium-excellence answer that is obtained by agile process.

Even though previous research on databases and records retrieval is considered as having focused on storage, look-up, and search, the possibility now could be to sweep via huge statistics units, to find out new phenomena, and to "learn". Ample opportunities for data search has been offered on data driven statistical learning techniques, which has been envisioned to support distributed and real time analytics.

Significant emphasis on time/data adaptively, has been given by both classical and contemporary search strategies, e.g. compression and dimensionality reduction as well as robustness. Recent rediscovery of stochastic gradient algorithms and stochastic approximation for scalable online convex learning and optimization, most often neglect Robbins–Monro and Widrow's seminal works, which has been done half a century ago, that is the testament of the fact.

Certainly it cannot be denied that computer science has important role to play in big data research, but data science scope and nature is not limited only to computer science rather it is multidisciplinary. Data search expertise from all trades are welcome to contribute to the big data search. As an instance, internet-gathered information are often replete with lacking entries, which motivates modern search imputation techniques that supports timely (low-rank) matrix disintegration, or, appropriate kernel-based inserts.

In the backbone of all large scale networks, there is a data matrix which is gathering traffic values, can be considered responsible for the clean traffic. Because of network topology induced correlations, volume of traffic irregularities that occur periodically in time and space, such matrices are low rank. Such matrices are rendered sparse across column and row. Potential to improve statistical learning performance has been offered by both quantity and richness of high dimensional data set. It has also been required to exploit hidden low-dimensional structure to effectively separate the data by innovative models. Consequent need has been recognized for learning these model by advance online, scalable optimization algorithm for information processing over graphs (Network sources of dispersed data and multiprocessor abstractions and computing architecture of extreme performance)

To suggest the architecture of new system it's important to understand the definition and properties of the Big Data. As per "NIST Big Data public working group" Big Data refers to the inability of traditional data architectures to efficiently handle the new datasets [1]. NIST has also defined characteristics of big data as 4 Vs (Volume, Variety, Velocity and Variability).

1.4 Big Data Definitions

Major stakeholders among commercial, academic and government leaders recognized the potential of Big Data, which has tremendous scope to ignite innovation, support commerce and lead the path to progress. They have done a broad level agreement and defined the common terms to describe the surge of data in todays connected world, which has lots of sources of digital data. Because of vast data resources, today we are able to answer the questions which were out of reach previously, as follows:

- Is it possible to detect the pandemic early enough so that danger can be avoided?
- Is it possible to predict new material with advance properties, before actual synthesis of that?
- How to make cyber security defender having edge over attacker, who is guarding the cyber threats?

The broad agreement also included the ability of the Big Data to over weigh the approaches, followed traditionally. Scientific and technological advances in the field of data analytics, management, transportation and users, are inspired by the burgeoning rate of data volume, speed and complexity. Regardless of the widespread agreement on opportunities and limitation of Big Data some crucial questions remain unclear or couldn’t make general consensus in the agreement, which keep confuse the potential users and blocks the progress. Some of these questions are as follows:

- Attributes of the Big Data Solution?
- Difference of Big Data as compare to the traditional data and their applications?

- How to define essential environmental characteristics of Big Data?
- The procedure to integrate these environments with the architectures already deployed?
- Define the challenges to deploy the robust Big Data solution and address the central scientific, technological and standardizations problems

USA presidential office has intervened in this context and announced the initiative in the area of Big Data research and development on March 29th, 2012. This initiative goal was to improve utilization and extraction of the information from the large and complex digital data set to accelerate the pace of discovery in science, establishing national security, and transforming teaching and learning. Announcement of more than $200million in commitment spread across 80 projects has been done by six federal departments and their agencies. These political developments aimed to organize, learn and mine the knowledge from the huge volume of digital data by the significantly improving tools and techniques. The industries, research universities and nonprofit organizations have been challenged by this initiative to join hands with the federal government to capitalize the opportunity created by this new field of Big Data.

National Institute of Standards and Technology (NIST) has been motivated by the White House initiative to accept the challenge to inspire the partnership among industry experts to further secure and effective implementation of Big Data. In NIST's cloud and Big data forum held on Jan 15-17, 2013, public working group has been proposed to further develop the interoperability framework for Big Data. Member of the forum decided that Big Data requirements including portability, interoperability, portability, extensibility, reusability, analytics, data usage and technology infrastructure should be defined and prioritized by the framework. This way, the implementation of the most secure and effective Big Data techniques and technology would be accelerated by the framework.

1.5 Objective of the Defining Big Data

To identify the Big Data concepts, defining terms to describe standard, and defining reference architecture terms, a new subgroup NBD-PWG definitions and Taxonomy has been formed. Hierarchy of components of the reference architecture has been provided by the taxonomy. To meet the requirement of specific user group it has been designed as follows:

- For Managers – to understand the emerging field, the terms will differentiate the classification of required techniques.
- For Procurement officers – to understand the organizational requirements and differentiating among available approaches, a framework is expected by the taxonomy.
- For marketers – For promotion and innovation of Big Data solutions, taxonomy is expected to provide platform.
- For technical community – A common terminology is expected from the taxonomy to differentiate the Big Data's special features.

1.6 Big Data and Data Science Definitions

Upgradation of existing technologies has been driven by the requirement of efficient and cost-effective data analysis, which propelled the evolution of data systems. For instance, when other available methods to handle the structured data were considered not efficient enough for the large data sets, then data storage paradigm and relational algebra models have been introduced as relational data storage model. Essential shift in data handling is introduced this way. Now because of known limitations of the relational data base technologies, a new paradigm shift has been introduced as Big Data, which started emerging. As structured data is incapable of handling large unstructured datasets.

Not just because Big Data is bigger, it is getting larger with the steady growth over the decades. A fundamental shift in architecture was required which has been fulfilled by the Big Data, which can be considered as same revolution as in the time of relational data model. As it took decades for the relational data model to evolve to the best of its efficiency, same way Big Data technologies are expected to evolve over the period of time.

Big data conceptual background has been there for the many years, yet the last decade has been observed as sudden maturation and application to the technology. Many concepts have been defined by the term Big Data, it also incorporates several distinctive aspects which consistently interact with one another. Interaction of four major aspects are necessary to explain the revolution, which has been listed as follows:

- Data set characteristics
- Data set analysis
- Data handling performance of the system
- Cost efficiency and business value

1.6.1 Big Data

As per NSIT-Big Data Public working Group (NBD-PWG) Big Data has been referred as the incapability of the existing data architecture to handle the new datasets proficiently. Characteristics which inspired the big data as a technology domain, are as follows:

- Volume (volume in terms of data set size)
- Variety (multiple data repositories, multiple domains of data and various type of data)
- Velocity (the rate on which data is flowing)

- Variability (there could be multiple other characteristics depending on the domain of the data)

1.6.2 Data Science

So far only three paradigm of the science were known but Data science could be considered fourth in addition to the computational science, theory and experiments, which has been suggested by Dr. Jim Gray in 2007. This paradigm has been evolved with the experimental science of directly learning from the data, is referred as Data Science or precisely Data-intensive science. In this paradigm scientist collect the data without predetermined theory and generate the theory based on the trend of the data.

1.6.3 Big Data Analytics

There are three ways to characterize Data Analytic procedure:

- Discovery – Data discovery to formulate the theory
- Development – Working on specific theory, by establishing the analytic process
- Applied – Redefining the operational system by abstraction of the analysis

Even tough Big Data encompasses analytic process of all three types, but the major changes are in the area of applied and development analytics. Types of analytics which are possible has been changed by the new Big Data technologies, however complete new type of analytics are not resulted with the change. Still analysts have developed the way of interaction with the data in such a speedy way that was not possible earlier. The way traditional statistical analytics were working, was by downsizing, sampling or summarizing prior to the analysis of the data. Such process applied to make analysis on large datasets on available hardware with the limited capability in terms of memory. In Big Data entire dataset value has been given importance for computation, which increases the chances for the analyst to get better results as compare to chances in correlation. Still correlation cannot be completely ruled out where just trend or direction information is enough for next step or presentation. These developments helped the analysis of statistics and data mining to find the cause, or describing the reason of something already happening or existing.

1.7 Scope of Data Search optimization using Bio Inspired Algorithms

Although data science, data analytics and other technologies like parallel execution has enhanced the performance of the data analysis tremendously but there can be further scope of optimization of performance by introducing Genetic algorithm or other bio-inspired algorithms, which would complement the big data technologies to optimize the data search performance beyond the efficiency level proposed by the Big Data.

A new phenomena of computing has been emerging which is inspired by the biological principles is named as Bio-Inspired computing. Bio-inspired computing is proving as an important stepping stone to produce excellent results in the area of industry, medicine, the environment and other multiple fields, which were not explored earlier by the biological inspiration. Most of the bio-inspired optimizations methods are working on the concept of maximizing the desired factors and minimizing the undesired ones under certain constraints, to find the most cost effective or best performance solutions. Any domain can be benefited by this optimization technique. Finding the best possible solution can be defined as optimization. There could be numerous real life problems where optimization may be applied. The problems, which can be solved by this technique requires the interaction with the real world of organism or system. Such behavior could be named as self-organization, which is primary mechanism behind the development of biological systems. Systems following self-organization methods are controlled without central command, by increasingly attractive self-motivated process. This follows the bottoms up approach by interaction between the lower level elements to form the global order. Very common example of this approach is Ant colony. Ant can work in the coordination to make complex network trail, also perform coordinated attack without central element guiding each ant. Such system needs decision making and good coordination in elementary level elements. This approach has been proven to improve the working efficiency of other search techniques if merged well with the other technologies.

One of the major advantage of considering self-organization techniques is to avoid convergence of the result in premature stage, that means optimization problem may result the value without optimizing the results completely. This problem has been tackled by the parental solution and optimization process exit criteria has been defined when evolution of next generation is not giving better outcome as compare to the previous generations. Considering this fact, there could also be the case when genetic variation is minimum, in such case there is high possibility of premature convergence. Multiple operator of genetic algorithm has provided solution to this problem. In GA there can be one or more than one selection operator can be taken for self-organization. To choose the proper selection operator each problem, there are certain defined conditions, such as depending on total distance between the points of the chromosome generated in last cycle. Defined number of methods and average fitness (as per the fitness formula) of the resulting chromosome, governs the Crossover or Mutation operations. The rate of crossover and mutation converge towards smaller values (near optimal solution).

In this thesis chapter 2 is covering the related work of clustering in Big Data, some old and new technologies of clustering fundamental. Then detailed discussion about the Mapreduce fundamentals and some more detail theory about the Bio Inspired algorithms, their evolution during the time. Then next point is about bottlenose dolphin foraging behavior, which is the basis of the thesis. After that discussion about the merging of bio inspired technique to the Mapreduce. Then in chapter 3 there is discussion about the proposed work in this thesis. Chapter 4 is discussion about the result of this new algorithm in the area of data search.

Literature Survey

2.1 Clustering

While considering unsupervised learning then clustering is one of the most important problems, it deals with the set of unidentified records and tries to find a structure. In broader way clustering is a process where objects have been organized into groups, based on some similarity. For example, “Group of M-Tech students” here common feature of students is, they all are enrolled in M-Tech program, same way students enrolled to the B-Tech program will be separated with the M-Tech students and form another cluster. So a cluster is the set of objects with similar features and objects dissimilar to this set will be member of other cluster. Figure 1

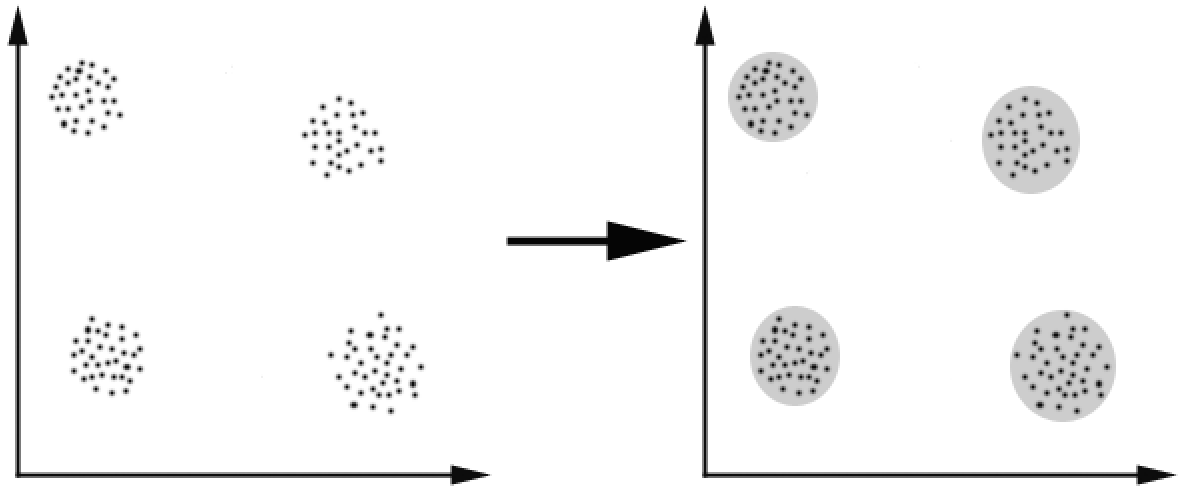


Figure 1

As shown in the figure, it is clear that considering similarity criteria distance among depicted points, all points can be easily divided in four clusters. Based on this explanation if objects are close enough (under threshold of being in same cluster) then they would be in same cluster. It can be named as spatial remoteness based clustering.

Clustering can also be done on any other parameter or concept. If two or more object defines a common concept, then these objects would be in same cluster. And concept of the cluster would be same as defined by these objects. So objects would not be grouped based on measurable similarity rather they would be grouped based on their concept.

It is one of the most suitable techniques to group the similar objects based on their characteristics [2] [3]. Further data can also be categorized in two forms, numerical data and categorical data. Categorical data clustering is difficult as compare to numerical data [4] [5]. Also such data sets are generally dynamic in nature so clustering becomes even more complicated. [6]

In this section various clustering algorithms based on categorical clustering and data labeling techniques, have been discussed [7] [8] [9] [10]. There is fundamental difference between categorical and numerical clustering analysis, that is in numerical clustering any element representing a specific cluster is used to summarize the clustering result, whereas same is not true for the categorical clustering [11] [12].

Clustering algorithm called BRICH is proposed by Zhang, Ramakrishnana and Livny, which was efficient to handle the noisy data [13]. The basic Idea proposed in BRICH is to first scan the entire data to form the initial clusters, then later re scanning the entire data set it would enhance the quality of the clusters. There is one more clustering algorithm, which has been proposed for large data sets, named CLARANS algorithm [14] but BRICH is considered superior based on efficiency. Later on one more algorithm CURE has been proposed which has addressed the issue of non-uniform sized or shaped clusters, based on hierarchical clustering algorithm [15]. This algorithm has given the basic idea about centroids, mentioning few fix number of elements, which are well scattered, are selected and brought closer to the centroids as a section. Selected elements are considered as representative of the cluster, which supports identifying the similar clusters and making less sensitive for outlier's elements. There is another algorithm K-mode, which says encompasses the clustering based on attribute. In this algorithm most frequent attribute in each attribute domain is considered for one cluster named "mode" [16]. It is not difficult to find the modes whereas to form a cluster using only one attribute value in each attribute domain, is not certain.

Numerical data clustering an algorithm named ROCK is proposed, which is considered efficient agglomerative hierarchical clustering algorithm [17]. The basis of the algorithm is not the distance rather link between the elements. This algorithm works on the concept of adjacency, considering some data point useful and other ignored. This concept has inspired the entropy based algorithm on every data element.

Another algorithm CACTUS is proposed by Ganti, Gehrke, and Ramakrishnan for clustering categorical data set, which works in three phases. In first phase entire data is summarized, in second phase the summary is used to identify the clusters and then finally in third phase clustering is accomplished [18]. Furthermore, algorithms for categorical data set, which cluster the elements as per their statistics are named as COOLCAL and LIMBO [19] [20]. Both of these algorithms depends on either minimizing statistical objective function or maximizing it. Another important categorical clustering algorithm proposed by Chen et.al (2007) based on data maximal similarity labeling method on N-node importance representative element. [21] [22]. Another algorithm on the basis of rough set data labeling for categorical data set has been proposed by Jiye Liang et.al [23]. In this thesis different approach of clustering has been applied on the MapReduce framework, which is inspired by the Biological behavior of bottle nose dolphin.

2.2 MapReduce

To address the requirements of Big Data paradigm and handling the data efficiently Google has developed a framework named MapReduce [24]. MapReduce divides the process for analyzing data in phases name Map phase and reduce phase. MapReduce framework is developed to utilize the capability of multiple nodes by parallel execution, which makes data analysis very efficient. Many powerful features have been added in MapReduce, like load balancing, fault tolerance and data distribution.

In this section working of MapReduce will be discussed. Basically MapReduce provides a framework to the developer to write their own logic to be executed in the distributed environment. Developer writes the Map function which takes an input pair and produces a set of Key/Value pair generated for the intermediate process. Now based on this intermediate key, MapReduce library groups all the associated values linked with the single key, and passes further to the next step that is Reduce method. Now in reduce method (written by developer) accepts intermediate key along with all the values associated with that particular key. All the values get merged here to make smaller set of values, if possible. Figure 2 Generally, per reduce invocation, it produces only zero or one value. Iterator has been used to provide intermediate values to the reduce method. With this approach values which are too large to fit in memory can also be handled.

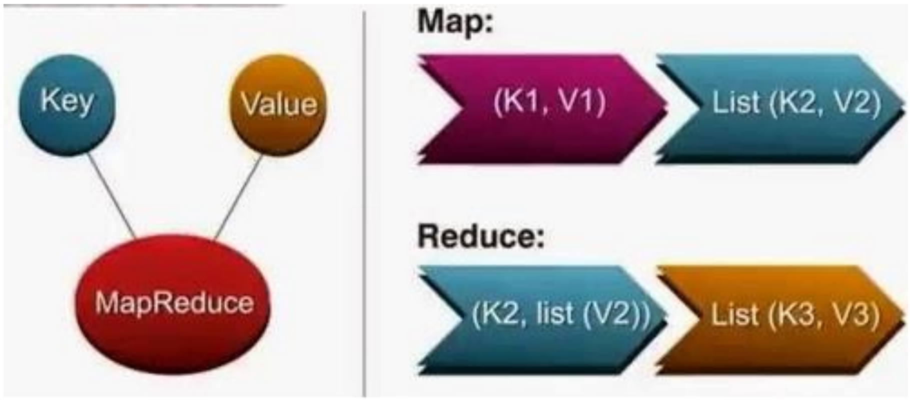


Figure 2

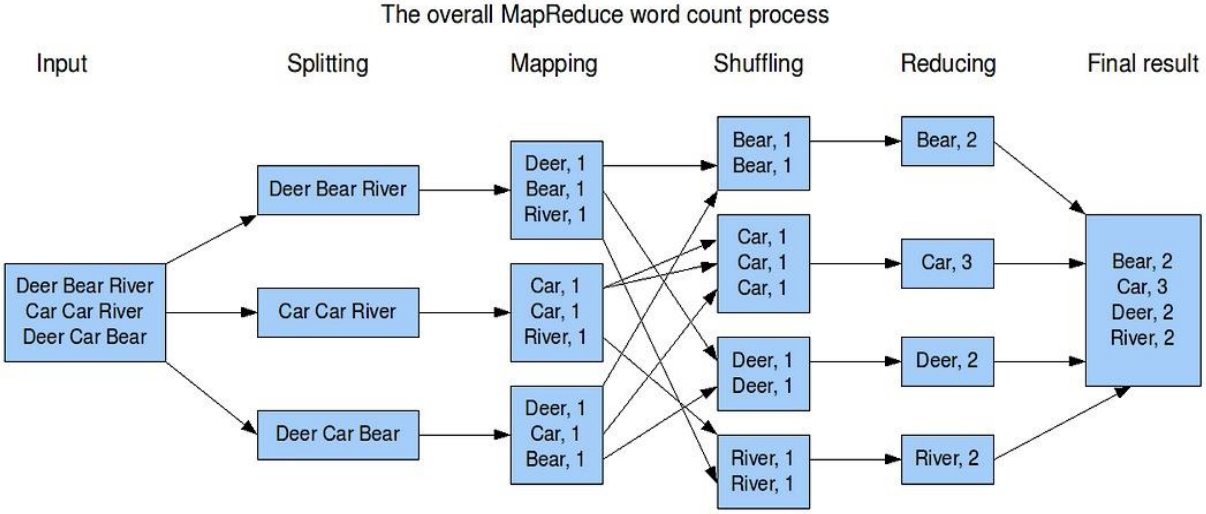


Figure 3

Overall concept of Mapreduce functioning is explained in Figure 3 with the help of one example.Further explainanation of system architecture for the mapreduce paradigm is depicted in the Figure 4. Splitter splits the input stream and sends the data (Key/Value) pair to the Map phase. There is master(user code) which assigns the set of instructions to the

Map and Reduce phase to operate as per the business logic.After Map phase intermediate files get saved on local systems to send further for the processing to Reduce phase. After reduce phase well organized table of data is expected, which can be used for meanigful data analyis.

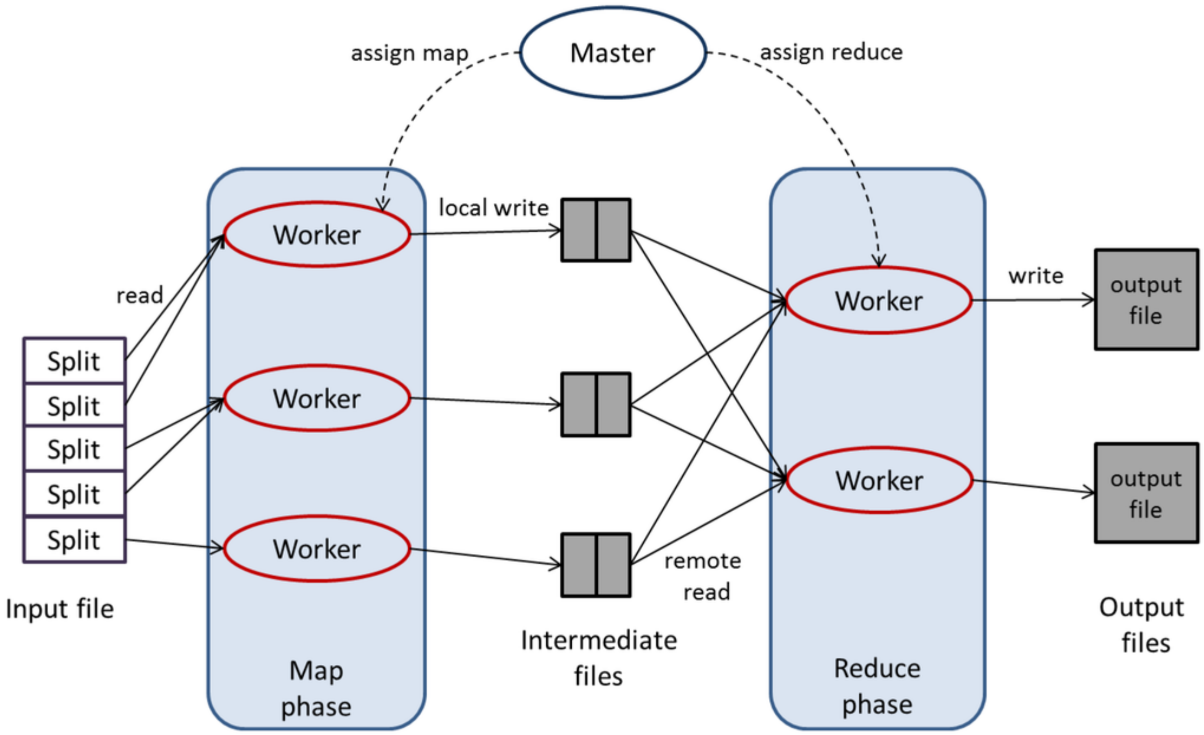


Figure 4

2.3 Bio-Inspired Algorithms

In cloud environment scheduling problem could be visualized same as Job-Shop scheduling problem, which is a well-known problem in the computer science, to allocate n jobs among the m available machines to maximize the performance. Given that all the n jobs have varying processing time and all the m machines have varying processing power. Now these types of problems could be easily optimized with the help of self-organizing Bio-inspired algorithms. In this section some well-known bio inspired algorithms will be discussed to understand the basic idea behind the algorithms. E.g. behavior of Ant groups inspired scientists to optimize many computational problems. Researchers have been applying such heuristic approaches to optimize well-known problems like graph coloring problem or traveling salesman problem. Natural behavior of ants, bees, swarm intelligence are inspiring many researchers to produce most efficient optimization algorithms. Marco Dorigo et al has introduced the ant system in 1996 [25]. Stochastic combinatorial optimization has been proposed and applied on TSP. Then researchers did many modifications and proposed different variants in various context. Dr Dervis Karaboga proposed “Artificial Bee Colony Algorithm” to optimize numerical problems in 2005 [26].

2.3.1 Ant Colony Optimization (ACO Algorithm)

“Ant Colony Optimization” (ACO) is one the most popular bio-inspired algorithms, which basically comprehends the basic behavior of the ants for foraging and returning to colonies. Many NP-Hard problems such as “travelling salesman problem”, assignment and scheduling problems etc., have been solved using the ACO algorithm.

Initially ants start searching for the food randomly, but they lay chemical name “pheromone” which helps them to navigate, while returning back to colony. They smell the pheromone and come back to the colony with the shorted path based on the intensity of the smell. Now it’s not necessary that one ant will find the shortest path while returning back, also the pheromone laid by her may evaporate after sometime. But the pheromone laid by the first ant also guides other ants to follow the same path or find their own best path. With this process whole group manages to find the best path and trail of ants start following this best path. During the random search process pheromone laid by first few ants will be evaporated itself and as a final solution the best path trail will be generated with rich pheromone.

2.3.2 Artificial Bee Colony Algorithm (ACB)

ACB is swarm based meta-heuristic algorithm [27] [28], which is inspired by the foraging behavior of Honey Bee. There are three four components in the Artificial Bee colony (ACB) algorithm: Scout, Unemployed Bees, Employed Bees and food source search. First scout randomly searches for the food source, if it gets the food source then it will turn to the employed bee, if it doesn't get the food source then it turns to the unemployed bee. Once bee get the food source (employed bee) shares the information to the other bees with their way of communication (waggle dance). This information sharing is actually broadcasting, which can be accepted by any number of bees at same time. This information includes the information about the direction, distance and quality of food. Unemployed bees wait for the employed bees to come to the colony and do waggle dance to share the information with them. If the food source exhausts the whole process of scout searching the food starts again.

This is good example of collective intelligence. This algorithm is useful in various problems like numerical optimization or combinatorial optimization etc. This model can be visualized in computer science domain: as node could be considered as food source and the efficiency of the node represents the quality of the food source.

2.4 Genetic Algorithm

Genetic algorithm is an approach to implement the artificial intelligence where self-organization produces the desired output. In 1956 John MacCarthy created a term artificial intelligence to propose "the science and engineering of making intelligent machine" and LISP has been developed [29]. Various known tools like Neural network, genetic algorithm and reinforcement learning have been known artificial intelligent methods which are biologically inspired. Adaptation in natural and Artificial systems book has been published by John Holland in 1975 [30], which was very influential on this topic. Although he started work on adaptive systems, much earlier 1962. Then in 1968 schema was developed by the publication of Schema processing [31]. After that schema theorem has been used to make building blocks for theory, work done by Goldberg [32]. Then in 1990 it has been discovered that stochastic and other noise distort the proportionate selection [33] [34] which was proposed in Schema Theorem.

Genetic algorithms can be classified as evolutionary algorithms [35] [36]. In genetic algorithm the given candidate for the possible solutions are considered to be the part of population and population evolves to get the best possible solution, with the nature inspired technique of parent selection, crossover, mutation and inheritance. Genetic algorithm practices the law of survival, which explains only fittest can survive. With this concept gradually the algorithm approaches towards the optimal solution because in computer science “fittest” is simulated with the optimal solution. GA algorithms progress as per the steps mentioned below:

1. Identified the candidate solution and then based on that Initial population is created.
2. Fitness function is defined to calculate the fitness value of each individual in the population.
3. Considering the fitness value parents are selected for the crossover
4. Reproduction operators are defined and applied on the parents to get next generation
5. Based on offspring fitness value, new population is created
6. Repeat the steps 3,4,5 to meet the termination condition.

There can be two ways of implementing genetic algorithm in the distributed computing environment: one way is to get the global optimal with some other algorithm and then execute the local GA steps based on the global optimal and other way is to find the local optimal and then find the global optimal based on all the local optimal outputs. The latter option is low efficient because they tend to exit locally on premature execution. [37]

2.4.1 Fractional Programming

There have been real life problems when optimization of one or several ratio of linear function is required [38]. Such problems have been addressed by the fractional programming methods. These methods are important in real life scenarios where decision making in involved (e.g. production planning, corporate and financial planning, hospital and healthcare planning etc.) By the end of 1980 researchers have been working only on single ratio problem. Later on various methods have been evolved like Linear programing problem to solve fractional function [39]. Then simplex method for solving fractional problems have been proposed by Gogia [40]. Then a researcher Horvath proposed a method using duality considerations, to define optimum in linear fractional programming [41]. Further Stahl [42] proposed solution to a simpler form of fractional programming by finite method. Then next step of linear functional programing, actually a class of that has been researched by many researchers was fractional interval programming. FIP using generalized inverses have been solved by Buhler [43].

2.5 BottleNose Dolphin Foraging Behavior

Bio Inspired optimization has been implemented in this thesis with the inspiration of Bottle Nose Dolphin foraging behavior [44]. They are one of the marine mammals out of seventy-six cetacean species. They have been starring many aquarium shows for quite long time and well recognize for their intelligence and charisma. They have capability to swim as much as 260 m beneath the surface of the sea. Bottlenose Dolphins are considered social beings, they do excellent communication and they are cooperative for each other for the food search. To achieve good hunting results, they work in team to optimize the attempt of searching. Bottlenose Dolphins have excellent technique to find meal and navigation that is called echolocation. Thru echolocation dolphins send ultrasound in the water, if this ultrasound waves falls on some object it reflects and returns to the dolphin again, which then detected by the dolphin Mellon (organ to decode sonar)

One of the incredible strength of the bottle nose dolphins are their communication. They send sound messages which varies in volume, wavelength, pattern and frequency. They have unique sound pattern for every important communication. Some of the observed sounds by bottlenose dolphins are buzzing squeaking creaking etc. Every individual dolphin has unique signature sound, to recognize each other by whistle. The range of frequency which can be produced by bottlenose dolphins are 0.2 to 150 KHz. For socializing they use lower frequency (0.2 to 50 kHz). Most of the dolphin's communication energy is spent on socializing frequencies (less than 40KHz). For echolocation mostly high frequencies are used, depending on requirement frequency may range between 40 to 150 KHz. This echolocation system is very efficient for the dolphins, they can easily identify the complete structure of the object including size, speed, shape, direction, distance and internal structure too.

As per researcher in biological field, mostly dolphin's daily activities are not very different than any other social animal, which includes feeding, socializing, traveling and napping. One of the most important aspect of dolphin's behavior is their social learning, which also improves their foraging strategies. As mentioned, in this thesis bottlenose dolphin foraging behavior has been simulated to search the best solution among the big data set. As per researchers' young calf of bottlenose dolphin acquire these foraging strategies by their connection with their mothers.

A calf of bottlenose dolphin is expected to master the techniques of foraging before being independent. Such complex communication used for foraging is not easy for calf to master so quickly, calf picks-up the strategies gradually, whereas the first year of education for calf includes nursing and foraging both. Calf generally remain dependent nutritiously on their mother for quite long time even after learning the foraging techniques, later they all form a group which bolsters the team effort to forage for good food. And also they can forage individually depending on the requirement.

Based on the information communicated by the other dolphins, individual dolphin knows the location for the food sources and it tries to optimize the path for the prey by finding the shortest path, which is possible by the bottlenose dolphins sonar system. When dolphin identifies the location it uses its beak to grill the bottom where prey is hiding.

During this whole process, team work is very important to share the locations of prey near to any individual, let's suppose if any information has been given to individual dolphin about the location of the prey which is not near, and it will take time to approach to that location. Then individual dolphin may pass on this location to the other dolphins who can approach there easily. With this approach overall performance of searching could be optimized.

2.6 Map Reduce for Parallel Genetic Algorithm

This section is required because MapReduce framework itself doesn't have feature to exploit parallel genetic algorithms. Here concept of genetic algorithm has been implemented to the MapReduce framework by upgrading the MapReduce as per requirement. This concept was proposed by the Chao Jin in [45]. Another proposed application of MapReduce for energy efficient architecture design on Field Programmable Gate Arrays is given by Zeke Wang [46]. Some other works which have been proposed for distributed data analysis on MapReduce [47] [48]

2.6.1 Overview of Parallel Genetic Algorithm

Genetic algorithms abstract the problem space as a population of individuals, and explore the optimum individual through a loop of operations. Usually the individual is represented by a string of symbols, and each step of the loop produces a new generation with reproduction, mutation, evaluation and selection operations. Given a generation of individuals as ancestors, the reproduction operation generates their offspring by combining several ancestors and the mutation operation performs simple stochastic variations on each offspring to generate a new version of it. The evaluation operation evaluates the offspring according to an objective function and the selection operation chooses the best one from the population for next generation. This process repeats until the optimum individual is found.

Among these operations, the evaluation and selection operation consumes most of the time and has been estimated to take more than 1 CPU year for the problems in complex domains [49].

```
function Distributed_GA()

        t=0                                 // index of the generation
        P[0] = $a_1$[0], ..., $a_u$[0]              // initialization
        Evaluation($a_1$[0], ..., $a_n$[0])

        While not T(P[t]) do
                P[t] = Mutation(Crossover(P[t]))
                Evaluation($a_1$[0], ..., $a_n$[0])
                <Communication>
                P[t+1] = Selection(P[t])
                t = t+1
        endwhile

return Optimum(P[t])
```

Figure 5

There are several models for PGAs. We choose the distributed model as a general presentation of the principle of PGA. With this model, there exist many elementary *worker* GA working on separate populations. Each worker performs same computations as the rest. Figure 5 illustrates the principle of PGAs with a distributed model. Essentially, after mutation and crossover operations, there exists a communication phase, where each worker communicates with neighbors for exchanging a set of individuals or statistics. After this communication phase, the offspring will be selected as the starting point to evolve next generation.

Each individual, P, consists of u elements, $a_{1}, a_{2}, \ldots a_{u}$. T is the function that determines whether the optimum value has been generated.

$$P_x > P_y \Leftrightarrow a_{xi} > a_{yi}, \; i \in (1\ldots u)$$

$$P_x < P_y \Leftrightarrow a_{xi} < a_{yi}, \; i \in (1\ldots u)$$

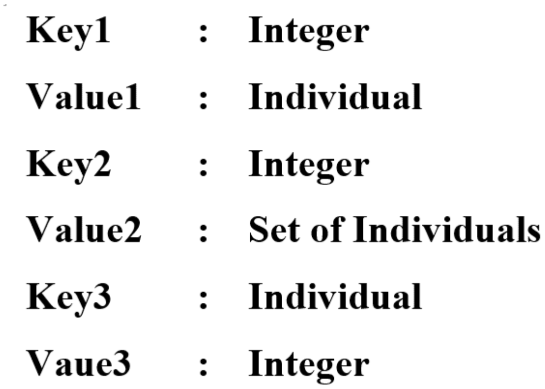
$$P_x = P_y \Leftrightarrow \begin{cases} (\ni i, a_{xi} > a_{yi}) \cap (\ni j, a_{xj} < a_{yj}) \quad i, j \in (1\ldots u) \\ OR \\ a_{xi} == a_{yi} \quad i \in (1\ldots u) \end{cases}$$

Figure 6

To perform the selection operation, the individual with the best value of evaluation are chosen. Each individual after evaluation still consists of an array of u elements: a_1, $\ldots a_u$. Given any two individuals, P_x and P_y, P_x is bigger than P_y only if every element of P_x is bigger than the corresponding element of P_y, as illustrated in Figure 6.

2.6.2 MapReduce for parallel GA

We deploy MapReduce to parallelize those parts of a PGA that are the most time-consuming. Essentially, a map operation can be used to express the phase of local evaluation. The communication phase can be achieved by collecting dependent inputs for the reduce operation through the runtime system. However, the execution of selection cannot be achieved by one reduce operation, because after the local selection, a global selection is required. Therefore, we have to express the selection phase through two phases of reduce operations. Thus the whole execution model consists of three phases: map, reduce and reduce. The types for input key/value pairs of MRPGA are illustrated in



Key1	:	Integer
Value1	:	Individual
Key2	:	Integer
Value2	:	Set of Individuals
Key3	:	Individual
Vaue3	:	Integer

Figure 7

At first, each individual is identified by a numerical key. After being evaluated in the map phase, each individual will be associated with a common key as the result. This intermediate result is kept on the local machine. In the standard MapReduce, with this common key, all mutated individuals will be associated together without any partition. However, MRPGA deploys a different policy. Essentially, the set of individuals associated with the common key is partitioned according to their locations. Each reduce function will be called for each partition, which is actually taken as the input list of value. As a result, a set of sub-optimal individuals is produced with the selection algorithm implemented by users in the 1st phase of reduce operation. In the final reduce phase, all sets of sub-optimal individuals are collected and then merged and sorted. Only the best individuals are selected by the system as the input of final reduce function.

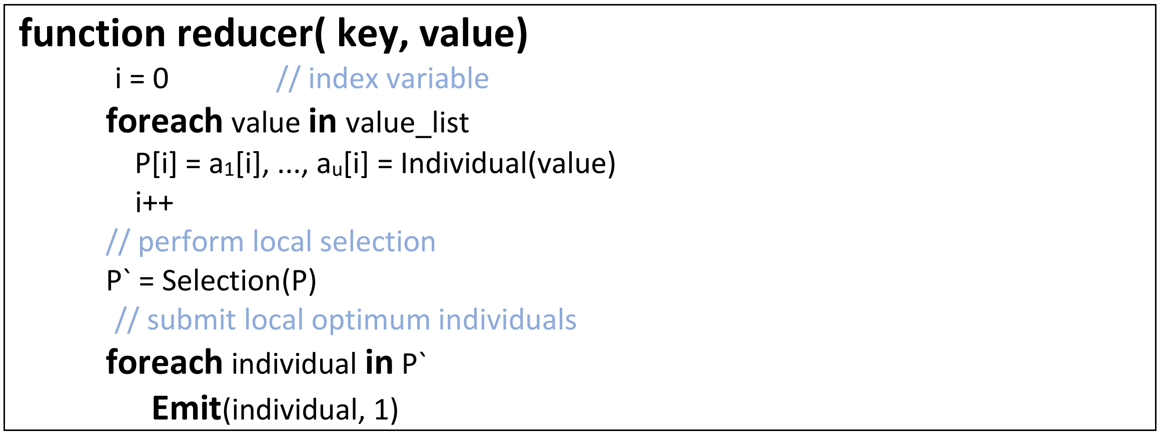
2.6.2.1 Map Phase

The map operation is for every individual and is called once for each of the individuals in each of the steps of the loop. As an input fed into the map function, *key* is the index of the individual, while *value* is the individual. The map operation extracts the individual from *value*, performs evaluation, and then submits the result as an intermediate output, as shown in Figure 8. In the figure, ***Emit*** is used to submit results.

```
function mapper( key, value)

        /*translation*/
        P[0] = $a_1[0], ..., a_u[0]$ = Individual (value)

        /*perform evaluation*/
        P` = Evaluation($a_1[0], ..., a_n[0]$)

        /*Submit intermediate results*/
        Emit (default_key, P`)
```

Figure 8

The results generated by the map phase are kept in a persistent database on the local machine. All the results are associated with same key, *default_key*. We adopt a partition policy different from the standard implementation of MapReduce. The intermediate results generated by Map functions are not partitioned by key. However, they are automatically split into pieces according to their locations. This partition policy allows each of the reduce tasks to collect dependent input just from the local machine without fetching data from a remote machine. Intermediate results produced by map operations on the same node will be merged by key as the input for the 1st phase of reduce operation.



```
function reducer( key, value)
        i = 0          // index variable
       foreach value in value_list
         P[i] = $a_1$[i], ..., $a_u$[i] = Individual(value)
         i++
       // perform local selection
       P` = Selection(P)
        // submit local optimum individuals
       foreach individual in P`
           Emit(individual, 1)
```

Figure 9

2.6.2.2 First Phase of Reduce

The 1st phase of reduce operation is for each of the partition groups generated by the map phase. As illustrated in Figure 9, the reduce operation extracts populations from *value_list*, and performs selection operation on those populations to choose local optimum individuals. Finally, it submits the selection result as input for *final_reducer*. The key of intermediate result is individual and the value is just a number. All the intermediate results generated by the 1st phase of reduce operations are collected as the input for the 2nd phase of reduce operation.

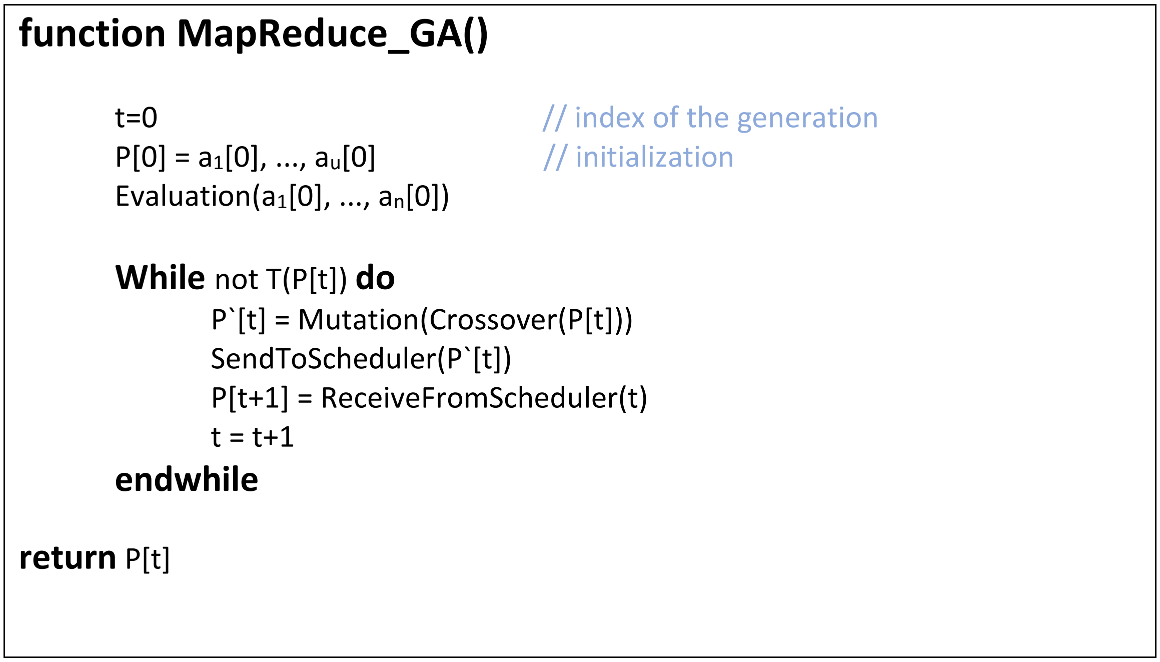
2.6.2.3 Second Phase of Reduce

The 2nd phase of reduce operation is for the global selection, called once at the end of each iteration of the loop. Essentially, there is only one operation in the 2nd reduce phase. The final reducer takes the intermediate result generated by the reducer in the first phase and produces the final selection results for the current generation. This result will be taken as the input for the next round of MRPGA operations.

```
function final_reducer (key, value)

        // translation
        $P = a_1[0], ..., a_u[0]$ = Individual(key)

         // Submit global optimum individuals
         Emit (P,1)
```

Figure 10

Local optimum individuals selected by the operation in the 1st phase of reduce are merged and sorted to select the global optimum individuals. The merging and sorting are performed by the runtime system. Only the best individuals are fed to the final reducer as input. Therefore, the final reducer just extracts each optimum individual from *key* and submits it as the final results, as illustrated in Figure 10



```
function MapReduce_GA()

        t=0                                  // index of the generation
        P[0] = $a_1$[0], ..., $a_u$[0]               // initialization
        Evaluation($a_1$[0], ..., $a_n$[0])

        While not T(P[t]) do
                P`[t] = Mutation(Crossover(P[t]))
                SendToScheduler(P`[t])
                P[t+1] = ReceiveFromScheduler(t)
                t = t+1
        endwhile

return P[t]
```

Figure 11

To achieve the iterations for the population evolution, a coordinator is adopted. As illustrated in Figure 11, the coordinator works on the reproduction, mutation, and submission of offspring to the scheduler of MRPGA and on the collection optimum individuals for each of the rounds of the evolution. Users do not have to face the difficulties of distributed computing. Instead, they only need to work on sequential programming for all the components, including one map function, two reduce functions and one coordinator. The runtime system coordinates the parallel execution of map and reduce tasks.

2.6.3 Architecture of Parallel GA on Map Reduce Set-up

The architecture of the runtime system that supports MRPGA is shown in Figure 12 The runtime system consists of one *master*, and multiple *mapper* and *reducer* workers. Mapper workers are responsible for executing the map function defined by users and reducer workers execute the reduce function, while the master schedules the execution of parallel tasks.

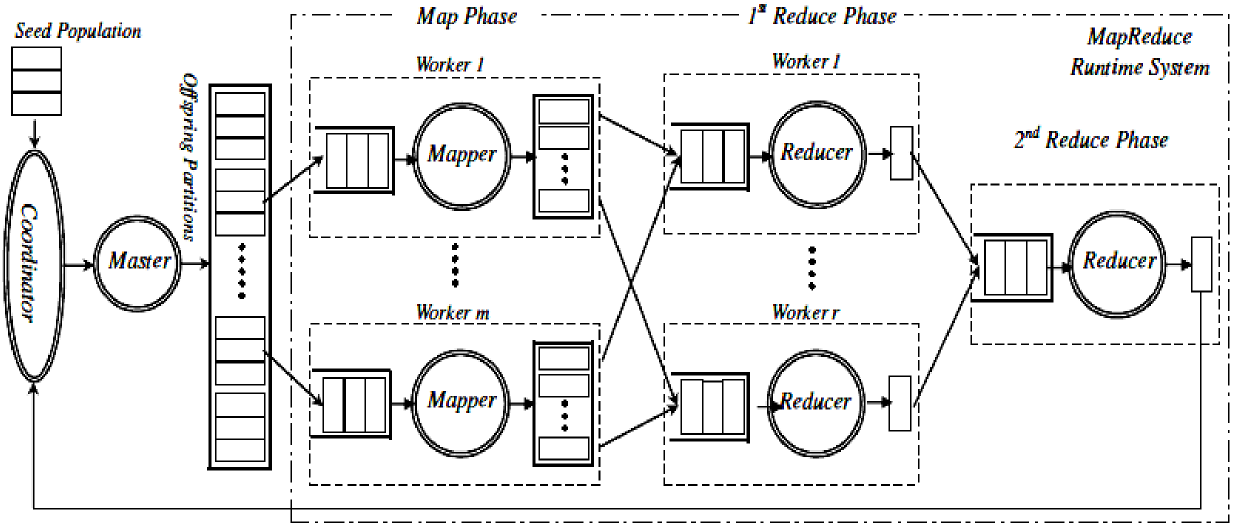


Figure 12

The control flow of execution consists of the following stages:

1) The coordinator generates offspring and performs mutation. Then, it sends the offspring to the master for evaluation and selection.
2) The master splits the offspring into *m* pieces respectively for *m* map tasks. The value of *m* is chosen so as to maximize parallelism for map tasks. Generally, this value is larger than the number of machines.
3) Each piece of offspring is sent to a machine with a mapper worker. The mapper worker iterates over the individuals in the piece of input to execute the map function

for each individual. Intermediate results generated by the map function are kept locally.

4) Each reducer worker is assigned with reduce tasks for the 1st phase of reduce operations. Normally the input is taken from the local machine. In case of heterogeneity, to make uniformly distribute loads over all workers, some reduce workers fetch intermediate results from neighboring machines.
5) The reduce function is invoked to select local optimum individuals that are then stored on the local machine.
6) A reducer worker is assigned to execute the final reduce function. This worker collects all the results generated in the 1st phase of reduce operation.
7) The final reduce function is invoked to produce the global optimum individuals as final results.
8) The final results are sent to the client for the next round of the evolutionary algorithm.

The above stages are repeated until the optimum individuals meet the specified requirements.

Different from the standard implementation of the MapReduce runtime system, an additional support is added for the 2nd reduce phase, including a special optimization for selecting the global optimum individuals. Since there is only one reduce task in the final phase, normally the master selects the most powerful machine from all the available resources to execute the *final_reduce* function.

Usually in the reduce phase, all inputs are collected, merged through sorting before feeding to the reduce function. MRPGA adds a policy support in the merging phase. That allows users to specify the order for the sorting and the number for the top elements which users want to process. For instance, to meet the requirements of the final reducer, users specify an ascending order, just to process the individuals with the biggest ranking value.

From Figure 6, we can know that the best offspring means a set of individuals, not just one individual. For those individuals with biggest rank value, they will be fed to the final reducer at any order.

To simplify handling faults during execution, the master replicates the optimum individuals selected by MRPGA for each round of evolvement. If some machines become un-available during the execution, we just restart the execution from the last round. This fault tolerance mechanism is different from standard MapReduce implementation and therefore we do not need a complex distributed file system for reliability purpose.

2.7 Literature Gap

As of now already lot of work has been presented or proposed in the area of data analytics and Big data search technologies, but still there is very less work done to reduce the infrastructure requirement to accomplish data search on Big Data. Even after so many years of Big data evolution, although technology has been improved many tools are available but still the real world data processing needs powerful computing, networking and storage hardware.

With the proposed method in this thesis there is attempt to optimize the data search with the help of innovative Bio-Inspired methods, utilizing the existing tools and framework. In this thesis same architecture of MPRGA (MapReduce parallel Genetic Algorithm) [45] has been used but the genetic algorithm has been optimized further by the bottlenose dolphin algorithm proposed by G.Kiruthiga [44].

In the proposed algorithm of bottlenose dolphin, it has been applied in single processing environment on limited set of data, whereas in this thesis the algorithm has been exploited on the large set of data with the help of dividing the complete data set in N (even number) equal data sets. Considering the communication behavior of the dolphin, assuming that N numbers represents the available dolphins, which interact among themselves to produce the best optimal result.

[redacted]Proposed Work

It has been observed there is still scope of optimization the way data search is performed in the area of Big Data. Very less work has been done to integrate the novel Bio-Inspired algorithms to integrate with the existing tools on the Big Data to exploit the search mechanism to its optimal level. There is an interesting algorithm proposed in 2015, which could be further customized to use the same for large data set, efficiently. In this work we are trying to report the shortest possible geo location path in the most time efficient manner by simulating the problem with the Bottlenose Dolphin foraging behavior.

3.1 Bottlenose Dolphin Communication

The bottlenose dolphin has excellent capability of echolocation which is used to communicate with other bottlenose dolphins during their foraging. In this paper we have utilized this behavior of bottlenose dolphin considering they work in team and share the information about the food location among each other. Let's suppose bottlenose dolphin A^1 has X locations which are near to her, but Y more locations which are not very much suitable for the dolphin A^1 to approach, so this bottlenose dolphin will share the information with the nearest A^2 dolphin. If A^2 could reach the location herself then it would be utilized otherwise she will pass on the information to the other group member in same way. With such kind of information sharing whole team of Z number of bottlenose dolphins search the entire area for the prey in most optimal way.

3.2 Variables used

L	Total number of locations $L = L_1$, L_2 …. L_n
i, j, k, l	Individual locations
d	Distance between the locations
t_d	Total distance
Indiv D_1	Number of locations allocated to Dolphin 1 { },

Indiv D_z	Number of locations allocated to Dolphin z { },
m	Highest distance
seqPath	covered by the individuals
MaxGen	Maximum generation
IndivCurrent	individual taken from generation
N	Maximum number of individuals in the generation
Gen	Current generation in the process
Z	Total number of bottlenose dolphin

Variable has been initialized as MaxGen = 100, Gen=0, N=10000 and Indiv=0, m=1, Z=4 Selection of initial value could be done with the most optimum level, by the help of fractional programming methods, considering the volume of data

3.3 Algorithm Description

Let us consider the problem of finding shortest path which connects the location L = {L_1, L_2, …. L_n} with d be the distance between location L_i to $L_{k.}$ With this problem statement, there is attempt to simulate the shortest geo location path between the given geo locations. The basic concept of this algorithm is the communication between bottlenose dolphins by echolocation and bottom grubbing technique used for searching the prey.

The locations L are related to the locations of prey. The dolphin D_1 has to estimate the distance of the prey location from its current location in order to signify whether this prey location is suitable for her or it should be suitable for any other dolphin. Based on this calculation dolphin D_1 communicates the prey location to other dolphin if required, else keep the location in allocation list of its own to optimize the shortest path till the prey location. In this algorithm all the location in specific order, has been considered as one individual out of N individual’s generation. And N individuals make a generation. So this process has to be executed for N times (Gen). The number of maximum generation (maxGen =100) has been taken for the demonstration, whereas this value should be selected based on the calculation of Fractional Programming method. Every iteration of next generation the sequence having shortest distance, selected from the current generation, as best individual (fittest individual), is carried forward to the next generation. The process continues until MaxGen is calculated.

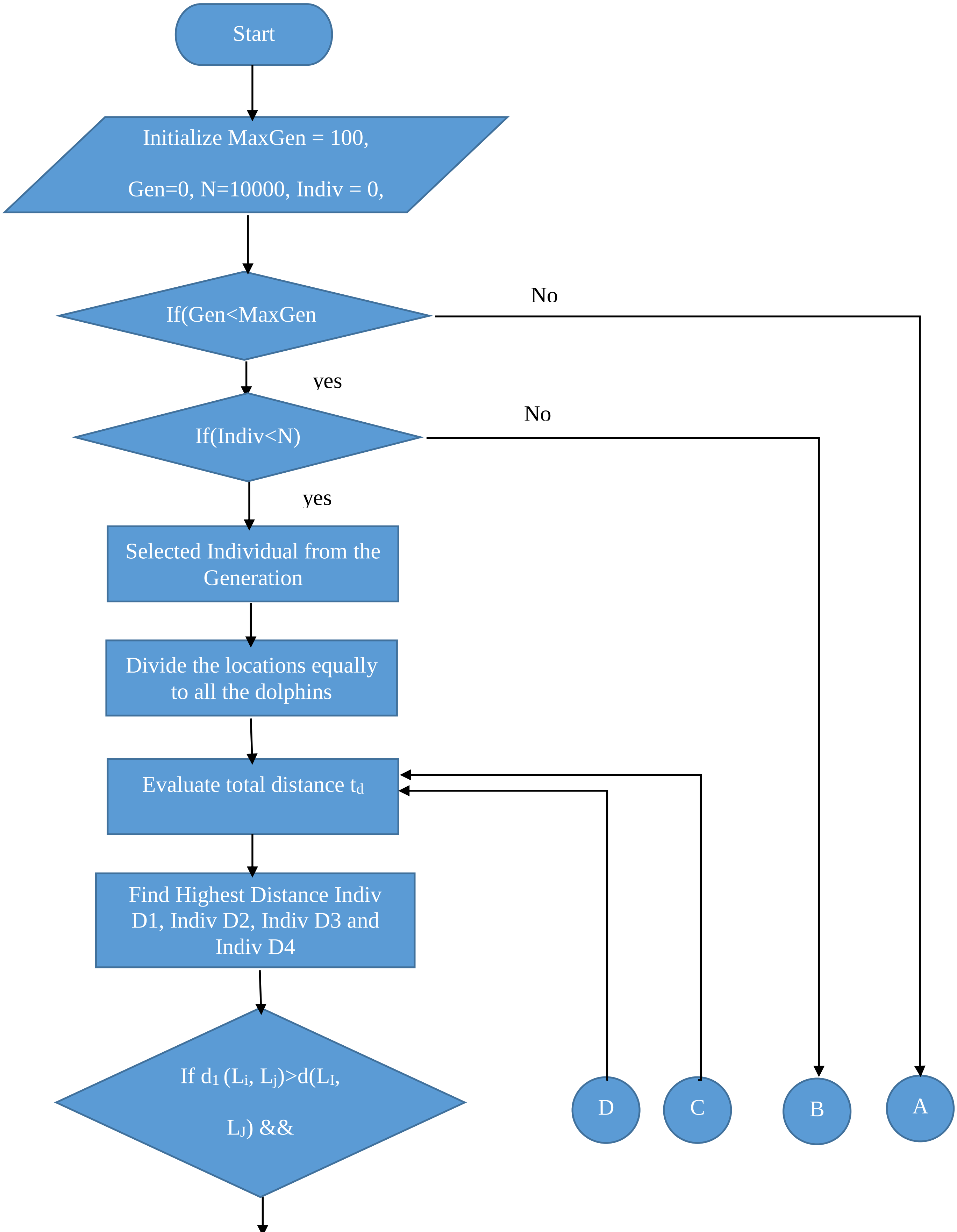
The algorithm starts from the 0^{th} generation. The locations are divided into N equal parts, considering locations for the N bottlenose dolphins. For simplicity let's imagine there are only 4 bottlenose dolphin. (N=4)

$$\begin{array}{ll}
\text{Indiv D1} & = \{\ L_1 , L_2 , \ldots\ L_{\frac{n}{4}}\ \} \\
\text{Indiv D2} & = \{\ L\frac{n}{4} + 1 , L_2 , \ldots\ L_{\frac{2n}{4}} \} \\
\text{Indiv D3} & = \{\ L\frac{2n}{4} + 1 , L_2 , \ldots\ L_{\frac{3n}{4}} \} \\
\text{Indiv D4} & = \{\ L\frac{3n}{4} + 1 , L_2 , \ldots\ L_{\frac{n}{4}} \}
\end{array}$$

Where Indiv D_1 implies the first quarter of the individual and Indi D_2 implies the second quarter of the individual and so on.

Once locations are allocated to all the bottlenose dolphins, let's start evaluating the total distance t_d between the locations. Calculate the highest distance between locations in D1, D2, D3 and D4. Swap the location positioned at the right side of the highest distance in Indiv D_1 and lndiv D_2. Follow the same process for the D_3 and D_4. Now again calculate the total distance t_d. If the value decreases after the step then repeat the step, otherwise find the first highest distance from the Indiv D_1 and second highest distance from the lndiv $D_{2,}$ third from D_3 and fourth from $D_{4.}$ If the t_d value increases further repeat the above process by taking the second highest value of Indiv D_1, third highest value from D_2, fourth from D_3 and fifth from D_4 and so on. If the value keeps decreasing, then follow the usual process. If the sequence start repeating then stop the process, that means off-spring is not generating better results, search has been completed.

Flow Chart


Start
Initialize MaxGen = 100,
Gen=0, N=10000, Indiv = 0,
If(Gen<MaxGen
No
yes
If(Indiv<N)
No
yes
Selected Individual from the Generation
Divide the locations equally to all the dolphins
Evaluate total distance t_d
Find Highest Distance Indiv D1, Indiv D2, Indiv D3 and Indiv D4
If d_1 (L_i, L_j)>d(L_I,
L_J) &&
D
C
B
A
]

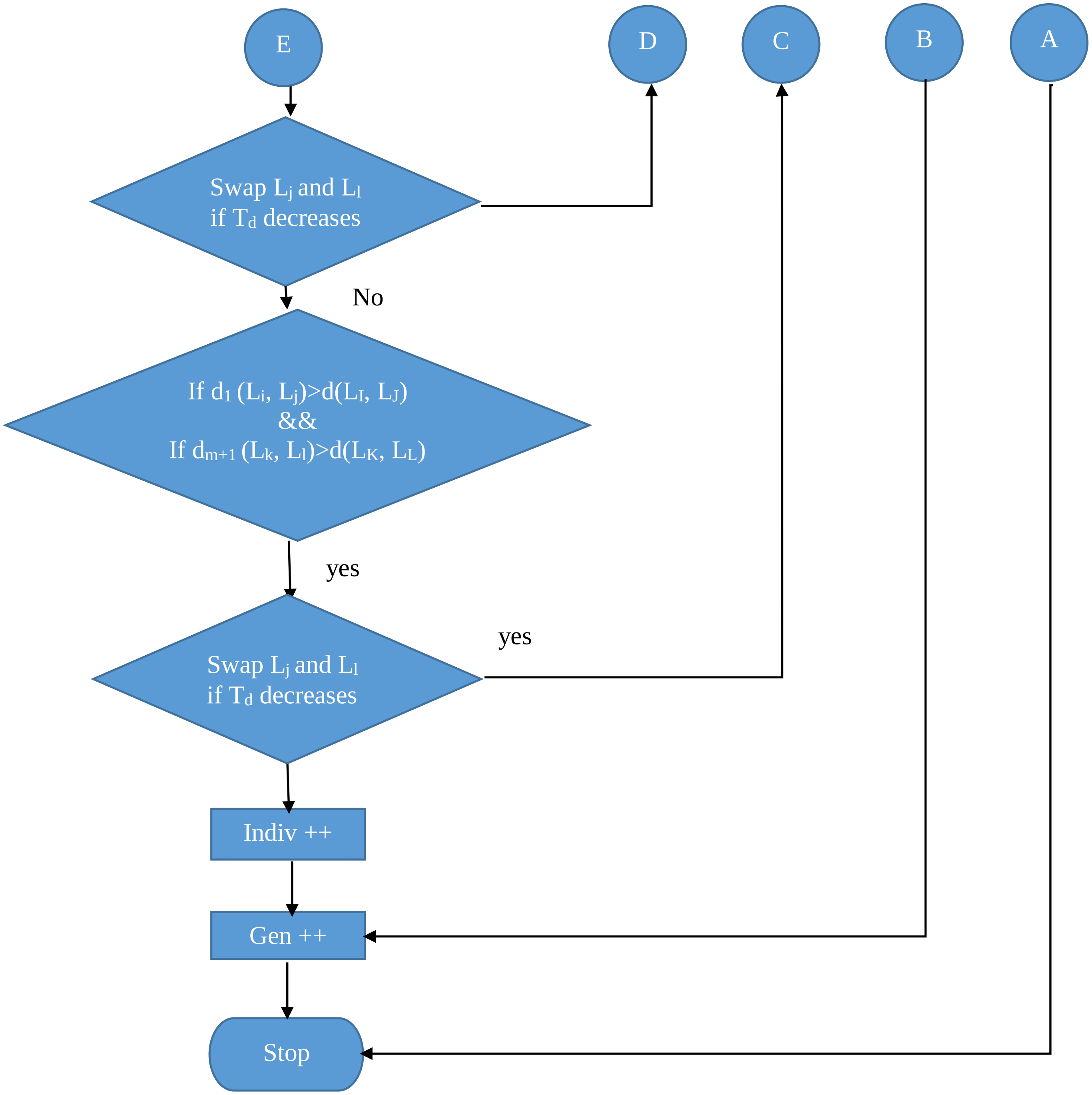
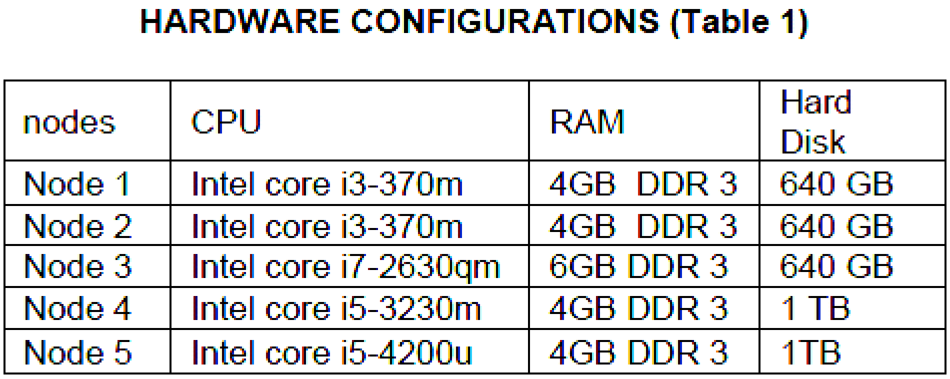

E
D
C
B
A
Swap L_j and L_l
if T_d decreases
No
If d_1 $(L_i, L_j)>d(L_I, L_J)$
&&
If d_{m+1} $(L_k, L_l)>d(L_K, L_L)$
yes
yes
Swap L_j and L_l
if T_d decreases
Indiv ++
Gen ++
Stop

Figure 13

Results and Analysis

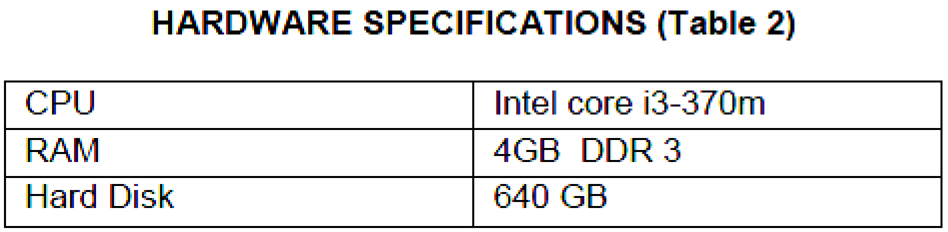
We compared the performance of the proposed algorithm with a sequential algorithm. Both the algorithms were executed for a total of 500 iterations with cross-over probability of 6% and mutation probability of 0.25%. The proposed algorithm was executed on a multi-node cluster with a total of 5 nodes each running hadoop v2.6.4 on an ubuntu 13.0 under vmware virtual machine with an allotted RAM of 4 GB, hard disk of 250 GB and two allotted processing cores. Hardware configuration of the cluster is shown in Figure 14. The sequential algorithm was executed on a single node with configuration shown in Figure 15. To evaluate performance, we measured the accuracy achieved and total execution time. Execution time was measured using system clock. The data set used for this experiment represents the differential coordinates of Europe map. It consists of 169308 instances and 2 dimensions.



HARDWARE CONFIGURATIONS (Table 1)

nodes	CPU	RAM	Hard Disk
Node 1	Intel core i3-370m	4GB DDR 3	640 GB
Node 2	Intel core i3-370m	4GB DDR 3	640 GB
Node 3	Intel core i7-2630qm	6GB DDR 3	640 GB
Node 4	Intel core i5-3230m	4GB DDR 3	1 TB
Node 5	Intel core i5-4200u	4GB DDR 3	1TB

Figure 14



HARDWARE SPECIFICATIONS (Table 2)

CPU	Intel core i3-370m
RAM	4GB DDR 3
Hard Disk	640 GB

Figure 15

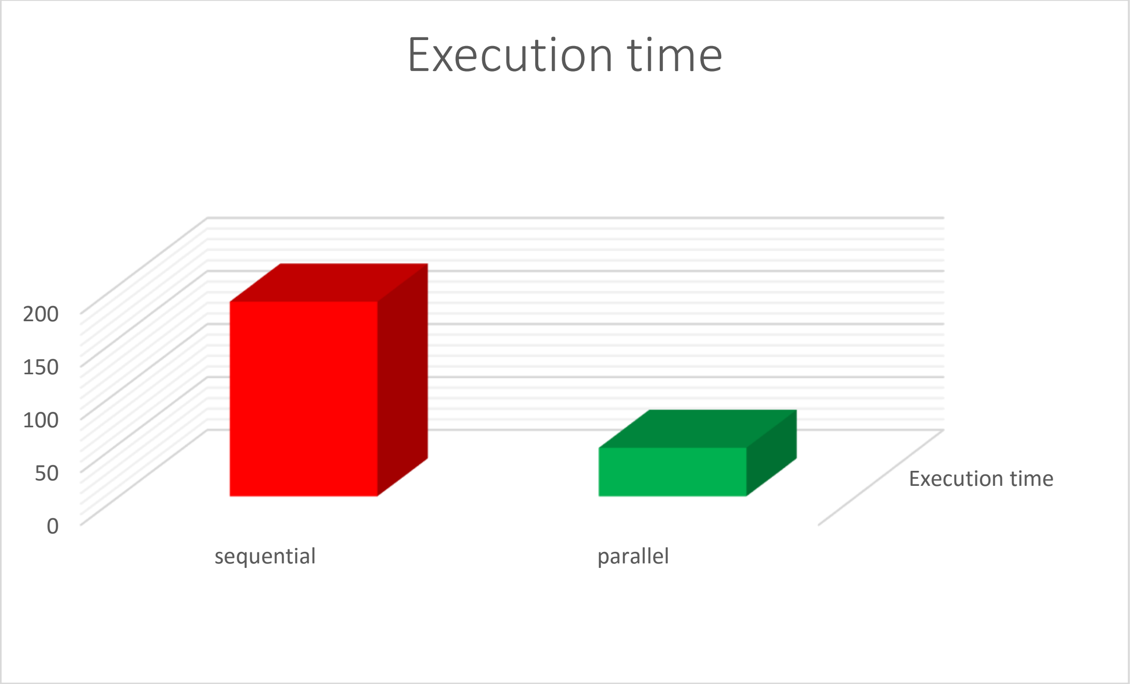


Figure 16

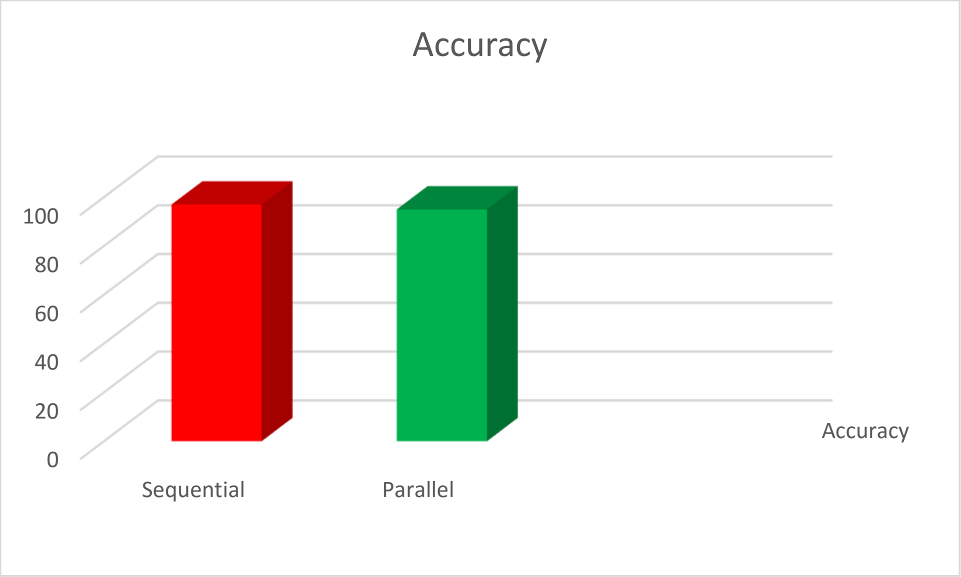


Figure 17

Conclusion and Future Scope

Mainly this paper has utilized the architecture of the MapReduce model to parallelize the Bio inspired heuristic algorithm. Considering the MapReduce programming model doesn’t save the state, we have updated the architecture for our purpose with adding additional reduce phase. This extension allowed us to execute BottleNose dolphin algorithm in parallel environment. Tremendous efficiency has been observed using bottlenose dolphin algorithm over MapReduce structure, in this particular case. With the help of this solution it’s possible to search the shortest possible time, which could be utilized in many real time problems. E.g. searching geo location path, searching best network path in the given topology. Considering the successful optimization with the algorithm, this algorithm could be used in other Big Data problems.

References

[1] NIST Big Data Public Working Group, "NIST Big Data Interoperability Framework," *NIST Special Publication 1500-1,* vol. 1, 2015.

[2] R. C. D. Anil K. Jain, "Algorithms for Clustering Data," 1998.

[3] P. J. F. Jain A K MN Murthy, "Data Clustering: A Review," 1999.

[4] J. K. P. R. David Gibson, "Clustering Categorical Data An Approach Based on Dynamical Systems," New York, USA, 1998.

[5] K. M. Han J, Data Mining Concepts and Techniques, Morgan Kaufmann, 2001.

[6] U. F. a. C. R. P.S. Bradley, "Scaling clustering algorithms to large databases," 1998.

[7] K.-T. C. M.-S. C. Hung-Leng Chen, "Labeling Un clustered Categorical Data into Clusters Based on the Important Attribute Values," in *IEEE International Conference. Data Mining (ICDM)*, 2005.

[8] R. Klinkenberg, "Using labeled and unlabeled data to learn drifting concepts," in *IJCAI-01 Workshop on Learning from Temporal and Spatial Data*, 2001.

[9] B. S. K. S. V. H. Venkateswara Reddy, "A Data Labeling Method for Categorical Data Clustering Using Cluster Entropies in Rough Sets," in *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference*, Bhopal, M.P. India, 2014.

[10] M.-S. C. S.-C. L. Hung-Leng Chen, "Catching the Trend: A Framework for Clustering Concept-Drifting Categorical Data," in *IEEE Transactions on Knowledge and Data Engineering*, 2009.

[11] J. W. ,. Y. Q. Jiye Liang, "A new measure of uncertainty based on knowledge granulation for rough sets," *Information Sciences,* vol. 179, no. 4, p. 458–470, 2009.

[12] C. E. SHANNON, "A Mathematical Theory of Communication," *The Bell System Technical Journal,* vol. 27, p. 379–423, 1948.

[13] R. R. M. L. Tian Zhang, "An Efficient Data Clustering Method for Very Large Databases," vol. 25, no. 2, pp. 103-114, June 1996.

[14] J. H. R. T. Ng, "CLARANS: a method for clustering objects for spatial data mining," *IEEE Transactions on Knowledge and Data Engineering,* vol. 14, no. 5, pp. 1003 - 1016, 2002.

[15] R. R. K. S. Sudipto Guha, "CURE: an efficient clustering algorithm for large databases," *ACM SIGMOD international conference on Management of data,* vol. 27, no. 2, pp. 73-84, 1998.

[16] M. K. N. Zhexue Huang, "A Fuzzy k-Modes Algorithm for Clustering Categorical Data," *IEEE Transactions on Fuzzy Systems,* vol. 7, no. 4, pp. 446 - 452, 1999.

[17] R. R. K. S. S. Guha, "ROCK: A Robust Clustering Algorithm for Categorical Attributes," in *International Conference on Data Engineering*, Sydney, NSW, 1999.

[18] J. G. R. R. Venkatesh Ganti, "CACTUS–Clustering Categorical Data Using Summaries," in *ACM SIGKDD*, 1999.

[19] Y. L. J. C. Daniel Barbará, "COOLCAT: An Entropy-Based Algorithm for Categorical Clustering," in *ACM international conference on Information and knowledge management (CIKM)*, New York, NY, USA, 2002.

[20] P. T. R. J. M. K. C. S. Periklis Andritsos, "LIMBO: Scalable Clustering of Categorical Data," *Extending Database Technology,* 2004.

[21] K.-T. C. M.-S. C. Hung-Leng Chen, "On Data Labeling for clustering Categorical data," *IEEE Transactions on Knowledge and Data Engineering,* vol. 20, no. 11, pp. 1458 - 1472, 2008.

[22] P. A. S. V. R. H. Venkateswara Reddy, "Data Labeling method based on Cluster Purity using Relative Rough Entropy for Categorical Data Clustering," in *IEEE International Conference on Advances in Computing*, Mysore, 2013.

[23] J. L. Fuyuan Cao, "A Data Labeling method for clustering categorical data," in *Expert systems with applications*, 2011.

[24] Google Inc, "MapReduce: Simplified Data Processing on Large Clusters," 2004.

[25] V. M. A. C. M. Dorigo, "The Ant System: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 26, no. 1, pp. 29 - 41, 1996.

[26] B. B. Dervis Karaboga, "Algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," 2007.

[27] D. Karaboga, "Artificial bee colony algorithm," 2010. [Online]. Available: http://www.scholarpedia.org/article/Artificial_bee_colony_algorithm.

[28] B. B. Dervis Karaboga, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm.," vol. 39, no. 3, 2007.

[29] V. Lifschitz, "Artificial Intelligence and Mathematical Theory of Computation," *San Diego: Academic Press,* 1991.

[30] John H. Holland, "Hierarchical descriptions, universal spaces and adaptive systems : technical report," 1968.

[31] "Schema Processing," *Handbook of Evolutionary Computation, Oxford University Press,* 1997.

[32] D. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning," 1989.

[33] J. J. Grefenstette, "Deception considered harmful," *Foundations of Genetic Algorithms,* 1992.

[34] A. G. D. B. Fogel, "Schema processing under proportional selection in the presence of random effects," *IEEE Transactions on Evolutionary Computation ,* vol. 1, no. 4, pp. 290 - 293, 1997.

[35] E. a. L. T. Zitzler, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," in *IEEE transactions on evolutionary computation*, 1999.

[36] E. a. L. T. Zitzler, "Multiobjective optimization using evolutionary algorithms—a comparative case study.," in *In Parallel problem solving from nature—PPSN*, Berlin Heidelberg, 1998.

[37] X. W. Y. H. Pengfei Guo, "The enhanced genetic algorithms for the optimization design," 2010.

[38] J. Cohon, "Multi criteria Programming, Brief review and application," *New York: Academic Press,* 1985.

[39] A. C. W. Charnes, "An explicit general solution in linear fractional programming," *Naval Res Logist,* vol. 20, pp. 449-467, 1973.

[40] N. Gogia, "Revised Simplex Algorithm for Linear Fractional Programming Problem," *Math. Student,* vol. 36, no. 1, pp. 55-57, 1969.

[41] W. B. C. Horvath, "B-Convexity," *Optimization,* vol. 53, pp. 103-127, 2004.

[42] J. Stahl, "Two new methods for solution of hyperbolic programming,," *Publications of the Mathematical Institute of Hungarian Science,* vol. 9, pp. 743-754, 1964.

[43] W. Bühler, "A note on fractional interval programming," *Zeitschrift für Operations Research,* vol. 19, no. 1, pp. 29-36, 1975.

[44] S. V. N. P. P. G.Kiruthiga, "A Novel Bio-inspired Algorithm based on the Foraging Behaviour of the Bottlenose Dolphin," in *INTERNATIONAL CONFERENCE ON COMPUTATION OF POWER, ENERGY, INFORMATION AND COMMUNICATION*, 2015.

[45] C. V. R. B. Chao Jin, "MRPGA: An Extension of MapReduce for Parallelizing Genetic Algorithms," in *IEEE Fourth International Conference on eScience*, 2008.

[46] S. Z. B. H. W. Z. Zeke Wang, "Melia: A MapReduce Framework on OpenCL-based FPGAs," in *IEEE Computer Society*, 2016.

[47] H. He, "D3-MapReduce: Towards MapReduce for Distributed and Dynamic Data Sets," in *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, Chengdu, 2015.

[48] R. B. C. W. L. Y. C. N. G. Yijun Ying, "Optimizing Energy, Locality and Priority in a MapReduce Cluster," in *IEEE International Conference on Autonomic Computing (ICAC)*, Grenoble, 2015.

[49] Sean Luke, "Genetic Programming Produced Competitive Soccer Softbot Teams for RoboCup97," in *Proc. of the 3rd Annual Genetic Programming Conference*, 1998.

[50] P. Radha Krishna, Big Data Search and Mining, Springer, 2015, p. 94.

[51] P. M. Yuri Demchenko, "Defining Architecture Components of the Big Data Ecosystem," 2014.

[52] NIST Big Data Public Working Group, "NIST Big Data Interoperability Framework," *NIST Special Publication 1500-2,* vol. 2, 2015.

[53] K. Cukier, "Data, data everywhere," The Economist, 2010. [Online]. Available: http://www.economist.com/node/15557443.

[54] S. M. S. O. Nivranshu Hans, "Big Data Clustering Using Genetic Algorithm On Hadoop Mapreduce," *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH,* vol. 4, no. 4, 2015.