Application of Distributed Optimization Algorithms in Health Sciences

Fazal Noor
Faculty of Computer Science and Information Systems
Islamic University
AlMadinah, Saudi Arabia

***Abstract*—In this paper we present optimization algorithms namely Genetics Algorithm and Runner-Roots Algorithm based on Distributed Optimization techniques to compute Reducts. Knowledge discovery is a very important area of research in Data mining. In the health industry there exists vast amounts of data and sometimes it is required to sift through it, removing redundancy yet retaining enough knowledge to base decisions upon. Removing redundant information is extremely time consuming and efficient distributed optimization methods are presented with results. It is seen the distributed optimization methods provide results faster than their ordinary methods.**

Keywords-Runners-root Algorithm, Data Mining, MPI, PC-Cluster

I. INTRODUCTION

In Medical Centres around the world there is vast amount of data being collected and stored in data centers around the world. The data in the health industry consists of undiscovered information which may contain valuable knowledge. The amount of data to be sifted through is so large and may need Super Computers to reduce the time to process it. Data may contain redundant information and is desirable to remove it. Pawlak and his colleagues introduced Rough Set Theory and is very useful in removing redundancy. There are numerous applications where it has been successfully used such as in medicine, drugs, diseases, image analysis, pattern recognition, and many others. In many optimization applications, the search space is so huge that it is impossible to perform the searching in reasonable time. It may take months or even years to search all the space. In this case, it becomes desirable to use optimization algorithms. Nature has always been inspiring humans in many facets of life. The literature is proliferated with nature-inspired algorithms. Many optimization algorithms have been inspired by nature. One of the earliest algorithm has been the Genetic Algorithm developed by Holland [3]. Since then there have been a proliferation of algorithms based on nature, some of which are Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Bats Algorithm, and many others. All these meta-heuristic optimization algorithms were devised to solve optimization problems where classical algorithms can not be used. All the algorithms require to search the solution space, therefore the algorithms generally consists of many iterations such the solution obtained is reasonable within the desired accuracy. The algorithms start with a set of a solution space, testing for fitness, modifying solution space using trial and error methods and with some random variables in order to perform a global and/or local search. The a fewer the number of tuning parameters the better the algorithm is considered to perform.

The main contribution of this paper is GA and Runner-roots algorithms are proposed for efficient search of the space and PC cluster is used to accelerate the computation process and time. The usage of the proposed algorithms can be in many fields of science.

The paper is organized as follows. Section 2 discusses some related work. Section 3 presents the methodologies namely the genetic algorithm, distributed genetic algorithm, and reduct computation using Runner-roots and distributed genetic algorithm on a PC Cluster with MPI methods. Section 4 contains the experimental methods and section 5 provides discussion. Last, Section 6 presents the conclusion and future work.

II. RELATED WORK

There are numerous optimization algorithms appearing in the literature based on mimicking nature. For example, Genetics algorithm mimicking production of superior off-springs from their parents, Ants optimization algorithm based on ants finding food with minimum shortest path. Bees algorithm flying in search of best flower to produce honey. Latest is Bat's algorithm based on their sonar system zoning on their prey and locking on flying insects to capture them for their food. There are hundreds of optimization algorithms based on nature and books have been written on them. All of the genetic algorithms have two phases, one is the coarse search phase and the other is local search phase. The rate of convergence is a desired property of an optimization search algorithm in that reduction of time needed in the computation phase is reduced. The most commonly used optimization algorithm is the Genetic Algorithm and is used as a benchmark to compare with other optimization algorithms. Genetic Algorithms have been used in search and optimization problems. GA has also been used in finding reducts [2]. Reducts basically is refined information where all redundancy

have been removed and reduct is then sufficient to differentiate the objects. The runner- roots algorithm is another optimization algorithm appearing in the literature and has been proposed by Merrikh-Bayat [9]. Little work has been done with the runner's roots algorithm compared to Genetics, Ants, and other optimization algorithms. In this work, we have proposed to use the runner-roots algorithm and we have extended it to parallel and distributed systems. An application of our extended runner's root algorithm is done Health sciences field.

III. METHODS

The methods namely, rough set theory, GA, Runners-root, and PC cluster used together to devise an efficient algorithm which might be very useful in the health sciences and other areas of research.

In 1982, Zdzislaw Pawlak developed rough set theory which deals with analyzing data tables (information system) [1],[4]. Each row of the table represents an object and each column represents an attribute. An information system is defined as *H=(D,S)*, where *D* is a set of objects which is non empty and finite. *S* is a set of attributes such that $s: D \to V_s$ *for every* $s \in S$. Decision Systems are information systems having a decision attribute and represent all the knowledge of the system. Decision system are defined as $H = (D, S \cup \{d\})$, where $d \notin S$ is the decision attribute and may take on several values. Elements of *S* are called conditional attributes. Next an equivalence relation on a set *Q* has the following 3 properties; reflexive, symmetric, and transitive. Equivalence class of an element $x \in X$ consists of all objects $y \in Y$ such that *xRy*. In other words, given an equivalence relation *R* on a set *Q*, we define the *equivalence class containing an element x* of *Q* by: $[x]_R = \{y \mid (x,y) \in R\} = \{y \mid x\ R\ y\}$.
B-indiscernibility relation is defined as,

$$IND(B) = \{(x, x') \in D^2 \mid \forall s \in B\ s(x) = s(x')\} \quad (1)$$

Objects *x* and *x'* are indiscernable from each other by attributes from *B*. Equivalence classes of the B-indiscernibility relation are represented as $[x]_B$. Given $B \subseteq S$ and $X \subseteq D$ of an *H=(D,S)* information system. Using the information contained in *B* an approximation to *X* can be made by defining B-lower and B-upper approximations of objects *X*. B-lower is defined as $BLX=\{x \mid [x]_B \in X\}$ and B-upper is defined as $BUX=\{x \mid [x]_B \cap X \neq \phi\}$. Using the knowledge in *B* the objects in BLX are assured to be members of *X*, whereas the objects in BUX are not assured but possible members of *X*. Next B-boundary region of *X* is defined as the set $BN_B(X)=B \cup X \Leftrightarrow$ BLX which consists of those objects that cannot firmly be classified into *X*. Last B-outside region of *X* is defined as the set $U \Leftrightarrow$ BUX : consisting of objects assurely classified as not belonging to *X*. A set is called "rough" if the boundary region is non-empty and is called "crisp" if the boundary region is empty. Reducts are defined to be a minimal set of attributes needed for classification. Computation of reducts is NP-hard. The discernibility matrix of H is symmetric matrix defined as:

$$c_{ij} = \{s \in S \mid s(x_i) \neq s(x_j)\} \qquad for \qquad i, j = 1,...,n \quad (2):$$

A. PC Cluster

A PC cluster consists of ordinary computers connected to a fast switch and forming a network. The PCs have a Red Hat Linux operating system installed on them and Message Passing Interface (MPI) installed for communications. A PC cluster is used to accelerate the search of reducts. In other words, information consists of redundancy and data has to be sifted out so reducts are found, which consist of minimum information sufficient enough to provide information contained in the original vast amount of data.

Computational performance is measured by a metric called speedup and is defined as

$$S_p = \frac{T_s}{T_p} = \frac{T_s}{T_{comp} + T_{comm}}, \quad (3)$$

where T_s is the sequential execution time on one node, and T_p is the execution time on *K* PCs consisting of computation and communication time. In parallel processing one would prefer the computation time to dominate communication time.

IV. OPTIMIZATION ALGORITHMS

Nature has its own ways of optimizations, the nature's algorithms. In optimization usually one is faced with the problem of finding the minimum or the maximum of a given function,

$$\textbf{min}\ f(x) \qquad x_{lower} < x < x_{upper} \quad (4)$$

where f: R^n -> R is the m-variable objective (cost) function to be minimized, x element of R^n is the solution vector to be searched for in the interval of x_{lower} and x_{upper}.

Genetic Algorithms (GAs) have their roots in biology and have been used in areas of search and optimization. In genetic algorithms the basic idea is to mimic nature, i.e. the following processes are performed; Selection process of selecting species to be parents, Crossover process of producing a child from the parents, Mutation process of producing child with symptons, Replacement process of

inserting the new produced species in the population and removal of the weak. Each individual in a population is given a fitness level and individuals are paired for mating (crossover). Genes consisting of chromosomes are passed on to the offspring and evolution occurs toward a better generation. Occasionally mutation occurs in the new generated offspring certain percentage of time. The new generation becomes part of the population replacing the weak individuals.

In relation to search and optimization the above processes are repeated till an optimized solution is reached and then GA stopped. The implementation of a genetic algorithm for parallel distributed computing maybe done in various ways. In our implementation we let each node process independently on an isolated population and transmit best individuals to master node PCs through migration. There are two migration models one is the stepping-stone model and the other is the island model. In stepping-stone model the migration is allowed only to neighboring subpopulations. In the island model, individuals are free to move to any other subpopulation (e.g. any-to-any). In communicating between PCs simple MPI send/receive or MPI collective functions, such as broadcast or scatter are used to send the whole population or subpopulation to a node and gather is used to collect newly produced *m*-best individuals from subpopulation.

A Distributed Genetic Algorithm (DGA) for reduct computation is given below.

Distributed Genetic Algorithm for Reduct Computation:

Master Node:

Initialization:

1. *Create Population matrix P and Discernibility matrix C.*
2. *Find number of length of P (i.e. number of 1s in each chromosome)*
3. *Find the number of combinations each chromosome in P can discern between.*

Loop for k-generations:

1. *Gather best m of population from node 1 to node N.*
2. *Broadcast best m times N equal to population size p to all in node group. (Note: in case m x N is > population size p chose the best p to send.)*

Gather the m-best after convergence from all PCs 1 to N.

Worker PCs:

Each worker node executes a copy of the genetic algorithm.

1. *Receive population.*
2. *Find number of length of P (i.e. number of 1s in each chromosome)*
3. *Find the number of combinations each chromosome in P can discern between.*
4. *While no convergence do*
5. *Select 2 individuals with fitness from population.*
6. *Perform crossover to create 2 children*
7. *Fill new population with individuals*
8. *Apply mutation to new population*
9. *Check termination condition*
10. *Send to Master Node top m-best individuals after j iterations.*

The communication pattern among the *N*+1 PCs is a simple send and receive. Each node 1 to *N* has a copy of the genetic algorithm running. Once every *j* iterations the PCs send to master node the top *m*-best individuals (e.g. 20 best) the master node after receiving the *m*-best from each node combines them into a new population and sends or broadcasts *m* times *I* individuals (e.g. 100) back to each node. The PCs again work on this new set of population and after a certain number of generations the PCs again send to master node a new set of *m*-best reducts and process is repeated till convergence or a limit on number of iterations is reached. At the end the master node would have the best from all the PCs and this information would be retained.

Runner-Root Algorithm for Reduct Computation:

First step:, a population space is randomly generated consisting of N points called mother plants.

Second step:, each mother plant generates two random points such, one is very near to itself and the other is very far from itself. This is analogous to local search or refined search and the far point is analogous to global search (jumping over local minimums).

Defining x_j(ith iteration) where j denotes the j-th mother plant at i-th iteration.

Next defining Xprop(i) matrix to consist of runners and roots to be constructed as follows,

Xprop(i) = [Xrunner(i) Xroot(i)]
where
Xrunner(i) = X(i) + dist_root x random r1
and
Xroot(i) = X(i) + dist_runner x random r2

X(i)= [x1(i) .... xN(i)]
Xprop(i)= [x1,prop(i) x2,prop(i) .... X2N,prop(i)]

where the matrix Xprop is concatenation of 2 matrices Xroot and Xrunner and therefore having 2N columns, r1 and r2 are random matrices having elements in the range of [-0.5, 0.5] and consisting of m-rows and N-cols. Droot and drunner are both scalars representing the distance of roots and runners of mother plant.

Third step is to calculate the fitness at each vector (potential solution) and

Fourth step to select the best vectors (possible solutions among the 2N) to be labeled as mother plants in the next iteration.

All these meta-heuristic optimization algorithms were devised to solve optimization problems.

V. EXPERIMENTAL RESULTS

Our test bed consists of a PC cluster which consists of 20 PCs with the following specifications: Intel Core™ 2 Duo CPU, 2.00 GHz, 1.00 GB of RAM. The PCs are connected to a Gigabit Ethernet switch. Each machine has RedHat Enterprise AS Linux 2.6.9-11 operating system installed and LAM 7.0.6/MPI 2 is used

. The following is an example of an information system, in which the objects are patients and attributes are as shown in the table.

TABLE I. INFORMATION SYSTEMS

Object /Attribute	Attribute 1 Runny	Attribute 2 Fatigue	Attribute 3 Fever	Attribute 4 Headache	Cold
Patient 1	No	No	No	slight	no
Patient 2	No	Yes	No	Normal	no
Patient 3	No	Yes	No	Normal	no
Patient 4	No	No	Yes	Slight	no
Patient 5	Yes	Yes	No	Slight	no
Patient 6	Yes	No	Yes	Slight	Yes
Patient 7	Yes	Yes	No	Slight	Yes
Patient 8	Yes	Yes	Yes	Heavy	yes

The objective is to decide whether a patient with a set of attributes will have a cold or not. This is a small example where in reality the list of patients may run into thousands and attributes may run into tens or hundreds.

Results of Sequential Genetic methods

For the sequential Genetic method, only one node is used to execute the algorithms. We used two algorithms, the ordinary GA and modified GA. An initial population size of 100 was chosen and one point crossover. In the Ordinary Genetic Algorithm the mutation rate is constant for the duration of all the iterations whereas in Modified Genetic Algorithm the mutation rate is variable. The following fitness function was used [2],

$$f(r)=\frac{q-k}{q} + \frac{c}{(p^2-p)/2} \quad . \tag{5}$$

where q is the number of attributes, k is the number of 1's in r, c is the number of object combinations r can discern between, and p is the number of objects. Table I shows the comparison of the two algorithms, the Ordinary GA with the Modified algorithm.

TABLE II. COMPARISON BETWEEN ORDINARY AND MODFIED GENETIC ALGORITHMS

Sequential Genetic Methods	Number of generations	Mutation Rate	Total time (secs)
Ordinary	100	1 bit constant	1876
Modified	100	Variable bits	1864

We noted that in Ordinary GA the search toward an optimum solution is slower in comparison to modified GA which uses variable number of bits for mutation.

Distributed Genetic Methods

In Parallel Distributed GA, the population matrix is sent to a group of PCs. Each node works on the population and sends the best b=20 solutions back to the master. The master then sends the new population to the PCs and the process is repeated till convergence to an optimum solution. We ran the DGA on a set of 3 groups consisting of 7 PCs, 11 PCs, and 21 PCs.

Each time the master sends the population to the PCs. The master specifies the mutation rate and the crossover rate. The plan was as follows: mutate 20% of the children, for the first 5 generations with 3 random bits, for the next 5 generations with

2 random bits, and for the rest with 1 random bit. Number of iterations for each generation were specified based on the following formula:

$$num_{parallel} = \frac{number\ of\ iterations\ in\ sequential}{number\ of\ nodes} \quad (10)$$

For PC clusters of 7, 11, and 21 PCs, the number of iterations on each was 30, 20, and 10, respectively. As the number of PCs are increased the time decreases.

The Runner-roots algorithm was run and the results tabulated in Table 2. It is also observed the distributed version has a faster rate of convergence to an optimal solution.

TABLE III. COMPARISON BETWEEN ORDINARY AND MODFIED RUNNERS-ROOT ALGORITHM

Runners-Root Algorithm	Number of mother plants	Total time (secs)
Ordinary	100	1978
Distributed	100	1963

Application to drugs

A drug can be defined as any substance that alters the human body's function either psychologically and/or physically. Some drugs are legal in certain countries such as alcohol, tobacco, and caffeine or illegal such as cocaine, heroin, cannabis, ectasy, etc. Psychoactive drugs are such that affect the brain's central nervous system and alters person's behaviour, mood, and decision making. Basically, there are four classes of psychoactive drugs, such as depressants, stimulants.

TABLE IV. TYPES OF DRUGS AND THEIR AFFECTS

	Types	Affect	Examples
1	**Depressants**	Drugs decreasing alertness by slowing down activity of central nervous system	Alcohol,analgesics heroin
2	**Stimulants**	Drugs which increase the body's state of arousal by increasing activity of the brain	Amphetamines, caffeine, nicotine
3	**Hallucinogens**	Drugs which alter perception and may cause hallucinations, such as hearing or seeing something that is not present	LSD and magic mushrooms
4	**Other**	Some drugs fall into the 'other' category, which may have properties of more than one of the above categories such as cannabis has depressive, hallucinogenic and some stimulant properties	

VI. DISCUSSION

The DGA algorithm and Runners-root algorithm are inherently parallel methods. Both depend on initial population and have faster convergence to an optimal solution or an approximation to it. In fact, population size, number of iterations, and number of generations, mutation rate, and number of PCs in a cluster all have affect on the performance of the algorithms. Increasing the population size increases the chances of finding the optimal solution. There is similiarity in the two algorithms, the mutation in GA and the size of Runner in the Runners-root algorithm. A larger number tends to move a search point further away from a local point and therefore search of the solution space. A smaller number tends to move search point only locally and tendency to get stuck there. Increasing the number of PCs participating gives a faster convergence rate to an optimal solution in comparison with a fewer number of PCs.

VII. CONCLUSION AND FUTURE WORK

It is observed that distributed optimization algorithms can be used to compute reducts efficiently on a PC cluster with MPI functions. We hope our application to finding reducts will find more applications in the Health Sciences which is proliferated with enormous amount of patient data, drugs and their side effects data, and many types of diseases with symptoms data, etc. Further research is to work with very large sets of data and provide solutions by obtaining reducts in each categories.

ACKNOWLEDGMENT

The author would like to thank Faculty of Computer Science and Information Systems Research Lab at the Islamic University in AlMadinah AlMunawarah for carrying out the experiments.

REFERENCES

[1] Z. Pawlak, “Rough Sets”, International Journal of Computer and Information Sciences, vol. 11, pp. 341-356, (1982).

[2] J. Komorowski, L. Polkowski, A. Skowron, Rough Sets: A Tutorial, http://citeseer.ist.psu.edu/komorowski98rough.html

[3] A. T. Bjorvand, and J. Komorowski, “Practical Applications of Genetic Algorithms for Efficient Reduct Computation,

[4] J. WrÖblewski, "Finding Minimal Reducts using Genetic Algorithm, Warsaw University of Technology, Institute of Computer Science, Reports, 16/95, (1995).

[5] M. M. Rahman, D. Slezak, J. Wroblewski, Parallel Island Model for Attribute Reduction, Proc. of the PReMI'05, Kolkata, India, Springer-Verlag (LNAI 3776), Berlin, Heidelberg, pp. 714 – 719, 2005.

[6] A. Leko, H. Sherburne, et al, "Practical Experiences with Modern Parallel Performance Analysis Tools : An Evaluation", Parallel and Distributed Processing, IPDPS 2008 IEEE Symposium 14-18 April 2008, Miami, Fl, pp. 1-8, 2008.

[7] J. P. Grbovic, et al, "Performance Analysis of MPI Collective Operations", Journal Cluster Computing, Vol 10, Issue 2, June 2007.

[8] C.F. Lacy, L. L. Armstrong M.P. Golman, L.L. Lance, Drug Information Handbook, 17th Edition, 2008.

[9] F. Merrikh-Bayat, The runner-root algorithm: A metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature

.

AUTHORS PROFILE

Dr. Fazal Noor received his PhD Degree from McGill University, Montreal, Canada, Masters and Bachelor Degrees from Concordia University, Montreal, Canada. Dr. Noor is an Associate Professor with the Faculty of Computer and Information Systems, Islamic University in Madinah, Saudi Arabia. Dr. Fazal Noor has published many papers in International Journals and Conferences. His current research interests include; Image recognition, Parallel and Distributed Computing, Fingerprint Verification thru Cloud Computing, Embedded Systems, Robotics and Computer Vision, Spread Spectrum Communications and Signal Processing.