Comparison of Parallel Distributed Metaheuristic Optimization Algorithms in Computing Reducts

Fazal Noor,
mfnoor@gmail.com
Faculty of Computer and Information Systems, Islamic University of Madina, Madinah USA

Abstract
Optimization algorithms have been applied in many fields. This paper presents performance of Genetics Algorithm and Runner-Roots Algorithm in computation of Reducts. An important area of research in Data Mining is knowledge discovery. Massive amounts of data exists in the health industry and problem is to sift through it, removing redundancy and at the same time retaining enough information to base decisions upon. As the amounts of data is huge, it is required use parallel distributed optimization methods for efficient search and PC Clusters for fast computations. The results indicate the huge benefits of parallel distributed systems to be utilized in such applications.

Key words:
Input here the part of 4-5 keywords.

1. 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 algorithms has been the Genetic Algorithm developed and introduced 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 cannot be used. All the algorithms require to search the solution space, therefore the algorithms generally require 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 contributions are the following, devising of Parallel GA and Parallel Runner-Roots for a PC cluster running MPI, comparing the performance of GA and Runner-Roots algorithms and the parallel distributed versions of, and applying these algorithms in an application. The GA and Runner-roots algorithms are proposed for efficient search of the space and PC cluster is used to accelerate the computation process and reduction of time. The usage of the proposed algorithms can be in many fields of science.

The paper is organized section wise with Section 2 introducing the methods with rough set theory, optimization algorithms namely the distributed genetic algorithm and the distributed Runner-roots algorithms. Section 3 presents the experimental results and section 4 provides the discussion. The last section 6 presents the conclusion with future work.this section, input the body of your manuscript according to the constitution that you had. For detailed information for authors, please refer to [1].

2. The Methods

The methods namely, rough set theory, GA, Runners-root, and PC cluster used together to devise an efficient algorithm, namely Parallel Distributed Metaheuristic Optimization (PDMO) algorithms which might be very useful in the health sciences and many other areas of research.

Manuscript received August 5, 2020
Manuscript revised August 20, 2020

2.1 Rough Set Theory

In 1982, Zdzislaw Pawlak developed rough set theory for analyzing data tables [1]. An object is placed in a row and attribute is placed in the column of a table. The information system can be represented as IS=(O, A), where O represents set of objects which is non-empty and finite. A represents set of attributes defined as $A:D \rightarrow V$ for every a member of A. Decision systems are information systems having a decision attribute and represent all the knowledge of the system. Decision systems are represented as $DS = (O, A \cup \{d\})$, where $d \notin A$ represents attribute of decision and can have various values. Elements of *A* are called conditional attributes.

B-indistinguishable relation is defined as,

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

The objects *x* and *x'* are indistinguishable from each other due to the attributes from *B*. 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:

$$c_{ij} = \{s \in S \mid s(x_i) \neq s(x_j)\}$$

$$for\, i, j = 1, \ldots, n \qquad (2)$$

2.2 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,

$$\text{Min } f(x)\ X\text{lower} < x < X\text{upper} \qquad (3)$$

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

Genetic Algorithms have been applied in many areas involving 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 Parallel Distributed Genetic Algorithm is presented below.

Parallel-Distributed Genetic Algorithm for Reduct Computation:

```
Master Node:

Initialization:
    1. Create Population matrix P and Discernibility matrix C.
    2. Find number of 1s in chromosome
    3. Check how many combinations in P can distinguish.

Repeat for G times:
    1. Receive x best solutions in Population from M Workers
    2. Send New population to Workers to work with.
    3. Receive the best from the Workers.

Worker PCs:

  Run genetic algorithm.
    1. Receive population.
    2. Find number of 1s in possible solution
    3. Calculate the number of combinations that distinguish.
    4. Repeat
    5.  Selection process of 2 individuals .
    6.  Crossover to produce 2 childs
    7.  Form the new solution space
    8. Perform mutation
    9. Is termination criteria satisfied
    10. Send to Master Node x best solutions.
```

Another metaheuristic algorithm named as Runner-roots algorithm was proposed by Merrikh-Bayat [9]. Only summary of the algorithm is presented here and the reader is referred to [9] for details. Note: for our application we have converted the Runner-roots algorithm for binary vectors and extended it for parallel-distributed systems.

The basic idea of the Runner-roots algorithm is as follows: There are Q mother plants, each producing plants near its position and plants far from its position therefore called the runner-roots. The plants growth which are far from it acts as a global search and the plants near it acts as local search. This in effect creates a good strategy to cover the search area and at the same time not getting stuck locally as some optimizations algorithms can.

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 produces two random positions such, one is very near to its position 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 denoting the distance of roots and runners of mother plant. Note: the vectors are converted to binary values, i.e. 0 or 1.

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

3. Experimental Results

3.1 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

$$S_p = \frac{T_S}{T_p} = \frac{T_S}{T_{comp}+T_{comm}} \qquad (4)$$

where T_s represents the serial time on a node, and T_p represents the time on *K* PCs consisting of computation and communication time, respectively. In parallel processing it is desirable the computation time to dominate the communication time.

The PC cluster is used as a master-slave that is one master and the rest of the PCs function as workers. The communication is using send and receive commands inserted in the C code implementing the PDMO algorithms. Every worker has a copy of PDMO C code running, every k iteration the worker nodes send their best x solutions to the master node. Once the master node receives x best solutions form all the workers, it further chooses the best of the best received and sends them to the worker nodes to work with. This scenario is repeated back and forth between master and slave workers till convergence to a solution is achieved based on satisfaction of a set criteria.

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

Object /Attribute	Attribute 1 Runny	Attribute 2 Fatigue	Attribute 3 Fever	Attribute 4 Headache	Attribute 5 Dry Cough	Cold
Patient 1	No	No	No	slight	no	no
Patient 2	No	Yes	No	Normal	no	no
Patie	No	Yes	No	Nor	no	no

nt 3				mal		
Patient 4	No	No	Yes	Slight	no	no
Patient 5	Yes	Yes	No	Slight	no	no
Patient 6	Yes	No	Yes	Slight	yes	Yes
Patient 7	Yes	Yes	No	Slight	yes	Yes
Patient 8	Yes	Yes	Yes	Heavy	yes	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. For our test data, the number of objects were 100 and the number of attributes were 14. Next, the results are presented for the two parallel distributed metaheuristic algorithms, namely the Genetic and the Runner-roots algorithm. The following fitness f(x) was used [2],

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

where q represents number of attributes, k indicates 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.

The PDMO algorithms were devised in C language and run on PC cluster of sizes 7, 11, and 17 PCs, respectively. Each method uses the master-slave principle, with the master performing the coordination of dispatching the solutions population to the slave workers and gathering Xbest solutions back from each slave. The slaves do the computation part by using the optimization algorithms to search and calculate the fitness value. The process of Master sending to Slaves and receiving best X solutions is repeated over and over again till convergence to the best solution is achieved. The fitness function acts as a guider to the optimum solutions. Note, by optimum it is meant best solution and not need to be exact solution. Number of iterations for each generation were specified based on the following formula:

$$num_{parallel} = \frac{number\ of\ iterations\ in\ sequential}{number\ of\ nodes} \qquad (6)$$

As the number of PCs in a cluster are increased the convergence time decreases. The PDMO algorithms were run and the results tabulated in Tables 1 and 2. It is also observed the distributed version has a much faster rate of convergence to an optimal solution.

Table 1. Comparison between Ordinary and Distributed Runners-root Algorithms for PC Custer size of 17.

Runners-Root Algorithm	Number of mother plants	Total time (secs)
Ordinary	250	4878
Distributed	250	452

Table 2. Comparison between Ordinary and Distributed GeneticAlgorithms for PC Custer size of 17.

Genetic Algorithm	Number of mother plants	Total time (secs)
Ordinary	250	4673
Distributed	250	463

4. Discussion

Parallel and distributed versions of Genetic algorithm and the Runners-root algorithm were devised and their performance in computation of reducts is studied. Both depend on initial population and have faster convergence to an optimal solution or an approximation to it. In fact, the size of the solution population, the number of generations, the number iterations, and the size of the PC cluster all affect the performance of the parallel distributed algorithms studied here.

There is similarity in the two parallel distributed 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 a search point only locally and the 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. Also, the PC cluster size is seen to be very useful in accelerating the rate to a solution. The combination of the parallel distributed algorithms and the PC cluster provide a very viable platform to carry out research problems involving massive amounts of data.

5. Conclusion and Future Directions

The parallel distributed optimization algorithms modified Genetic algorithm and Runner-roots algorithm were compared in terms of computing reducts for large data set. It was observed the reduction in time using parallel methods is drastic and further improves with PC cluster size from 7, 11, to 17. The proposed parallel distributed optimization algorithms presented will be extremely useful in massive data mining areas. One of the areas in which data is increasing at a tremendous rate is the Health science area where patient data, drug information, types of diseases, are sitting in databases. These databases need to be mined and with the parallel distributed algorithms presented here will be of great benefits. The combination of optimization methods and use of PC clusters together form a framework and system useful in many applications demanding efficient search and fast computation.

Acknowledgments

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

Fazal Noor received the B. Eng. and M. Eng. degrees in Electrical and Computer Engineering from Concordia University, Montreal, Canada in 1984 and 1986, respectively. He received his Ph.D. from McGill University, Montreal, Canada in 1993. Currently, he is with the Faculty of Computer and Information Systems at Islamic University of Madinah, Saudi Arabia. He has published numerous papers in IEEE, ACM, Springer, IEEE Transactions, Elsevier, Hindawi, and various international journals and conferences. He has been a reviewer for IEEE, Elsevier, Springer, and various other journals. He is Program Coordinator for Master of Computer Science program. He has received numerous awards. He has been a TPC member of many conferences. He has been QA evaluator. His research interests are in AI, FANETS, Neural Networks, Embedded Systems, Signal Processing, Network Security, Optimization Algorithms, and Parallel and Distributed computing.

https://www.khanacademy.org/computing/ap-computer-science-principles/algorithms-101/x2d2f703b37b450a3:parallel-and-distributed-computing/a/distributed-computing