OPTIMIZATION WITH MAP REDUCE AND CWW ALGORITHM

[1]MUHAMMED EMRE COLAK, [2]ASAF VAROL

[1,2]Department of Software Engineering, Department of Software Engineering,
Firat University 23119 ELAZIĞ/TURKEY
E-mail: [1]memrecolak@firat.edu.tr, [2]avarol@firat.edu.tr

Abstract—As world grow, parameters of jobs are increasing and optimization of this parameters has significant importance, in this regard; metaheuristic optimization algorithms has broad usage. However these algorithms can be more effective if they can work in distributed environment. It has been noticed that such a distributed optimization application exist but an application in MapReduce platform has yet to be done. In this paper we have been choose to work with MapReduce framework and Circular Water Wave (CWW) Algorithm. With MapReduce framework bigger problems may be solved with lesser cost and time.

Index Terms— Distributed working, MapReduce, Metaheuristic Algorithms, Optimization.

I. INTRODUCTION

In today's scientific world; nature based algorithms have many benefits in optimization and demanding more research. Some of these algorithms are Particle Swarm Optimization (PSO) [1], Artificial Bee Colony (ABC) [2], Genetic Algorithms [3], Firefly Algorithm [4], Bat Algorithm [5], Artificial Chemical Reaction Optimization Algorithm [6]. These algorithms are being improved by scientists and still have room for more improvements. Some of the new researches in this area are; A Modified Flower Pollination Algorithm for Global Optimization [7], A novel optimization method, Gravitational Search Algorithm (GSA), for PWR core optimization [8], GACE: A meta-heuristic based in the hybridization of Genetic Algorithms and Cross Entropy methods for continuous optimization [9], A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm [10], and A Novel Intelligent Optimization Algorithm Inspired from Circular Water Waves[11].

As world grows, parameters of jobs are increasing and optimization of this parameters has significant importance, in this regard; metaheuristic optimization algorithms has broad usage. However these algorithms can be more effective if they can work in distributed environment, on the other hand using supercomputers to solve optimization problems may prove costly. In this regard MapReduce platform gives low cost, simple and easy deployment for problems which can be divided map and reduce.

MapReduce framework has a broad usage in terms of big data processing which includes text mining [12], extreme learning [13], distributed learning [14],classification[15] ,image segmentation [16], clustering[17] etc. MapReduce platform means to provide a framework for processing large data sets. Bu it may be possible to implement it for large process needs also.

Main problem of optimization algorithms in terms of distributed work is heavy need of communication. In every step algorithms need to exchange data in order to search optima. However Circular Water Wave(CWW) algorithm doesn't have such a need and it is embarrassingly parallel which makes it good choice for parallel work. In this paper we proposed distributed optimization method using CWW algorithm.

This work has been organized as follows; at section II introduction to MapReduce platform is given, at section III CWW algorithm is explained , at section IV method for optimization in MapReduce platform is given, at section V experimental result for a optimization problem is given and finally at section VI conclusion of the work and possible future research topics are given.

II. INTRODUCTION TO MAPREDUCE PLATFORM

According to official Hadoop website "Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner." [18]MapReduce framework is a whole system that gives ability to connect and control commodity hardware to use as a distributed work platform. It has also necessary codes and tutorials to make your own program to solve your own problem. It is also possible to not make clusters to solve your problem instead just buying these as service from firms like Amazon. Amazon has an elastic MapReduce web service and according to amazon website "Amazon Elastic MapReduce (Amazon EMR) is a web service that makes it easy to quickly and cost-effectively process vast amounts of data.Amazon EMR simplifies big data processing, providing a managed Hadoop framework that makes it easy, fast, and cost-effective for you to distribute and process vast amounts of your data across dynamically scalable Amazon EC2 instances"[x]

It is important to say that not all problems can be solved using MapReduce framework. It is necessary that problem could be divided into key and value pairs and all key-value pairs must be processed by the same code.

"The MapReduce framework operates exclusively on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types".[18] Simply put after generating <key, value> pairs from the data we wish to process by using map and reduce methods MapReduce framework solves problem at hand. So input and output of the MapReduce are <key, value> pairs.

(input)<k1, v1>->map-><k2, 2>->reduce-><k3,v3>output)

MapReduce took <key, value> pairs as input and sends them to the map method. Map method process these values to create new intermediate <key, value> pairs then whose values are send to reduce method to reduced according to their keys.

Even though purpose of MapReduce is to process large amount of data it is possible to shape it to process vast amount of calculation because one of the good things about MapReduce is that it won't care the code you will write for the map or reduce functions only thing that matters is that your map method should take <key, value> pairs as input and should return <key, value>pairs as output.

MapReduce framework does one map and reduce operations for all data. For an optimization problem it was a problematic since same operation may need to work over and over. But with Apache Hadoop Main 2.6.2 APIchainMapper class has been introduced so map and reduce operation can be chained. It is up to the user to make chaining and how many operation will be chained.

For more information and tutorials of MapReduce visit http://hadoop.apache.org/.

III. CWW ALGORITHM

CWW algorithm is a , a novel nature inspired algorithm for continuous optimization of numerical functions Algorithm itself inspired from water drops and forming waves around it.

Forfunction f with n variables and take the variables of function f as x_i ($0 \leq i<n$), for searching maximum or minimum values of y, where;

$$f(x_0, x_1, \ldots\ldots x_{n-2}, x_{n-1}) = y \quad (1)$$

Calculate:

$$d_i = r_i \times w \times c_j \quad (2)$$

Where d_i is the difference from previous value of x_i, r_i is the radius of i^{th} domain, w is a random value generator (between 0 to 1) , and $c_j$is the j^{th} circle.

When the fitness is not improving $r_i$values updated as follows;

$$r_i = r_i / m \quad (3)$$

Pseudo code of CWW algorithm

Randomly choose starting points.

Do:

For each starting point

for each wave circle

-Calculate di values according to (2)

-Create new wave points in each direction and calculate fitness.

-Randomize best points

-Calculate fitness of new points.

If fitness is not improved

-Increment fail count and update ri values according to (3)

Use new points which are best b to create new starting points.

While(fail count < 10)

IV. OPTIMIZATION WITH MAPREDUCE

In this work we have used Apache Hadoop 2.7.2 framework to work with MapReduce. For trying our program we have used single nod cluster setup (pseudo-distributed) in a single computer which runs Ubuntu 12.4 operating system.

Since MapReduce framework won't care the code you will write for the map or reduce functions and only thing that matters is that your map method should take <key, value> pairs as input and should return<key, value> pairs as output, it is possible to use CWW algorithm inside the map function. In this work we have used some initial points, the function values of these points and necessary arguments as value and used an integer as key to provide input for map method.

In map function we have used CWW algorithm to provide new and better fitness points of the function as new <key,value> pairs as output of the map method.

By using chainMapper class we have chained map methods to provide better results.

Since all of the job has been done in map method we didn't use anything in the reduce method so our MapReduce looks like;

(input)<k1, v1>->map-><k2,2>->reduce-><k2,v2>output)

In order to give necessary data to CWW algorithm we have used text based formatting in value of<key,value> pair. x_i values, r_i values , fail count, fitness, number of waves, number of iteration, and completed iteration values inserted into text to provide value area of the <key,value> pair. Below shows a sample input fileformat :

x0/x1/x2/…./xn%r0/r1/…../rn%fail count %fitness%number of wave%number of iteration%complated iteration

It should be noted that this input format is for one line not multiple line, each input must consist of this areas and it is possible to give more than one input in an input file. For finding minima in a two dimensional Sphere function (x) if we were to use 4 starting point((1,1),(2,-1), (-1,-1),(-1,2)) with radius as 1, initial fail count would be 0, fitness would be a function value, number of waves 3, number of iteration 100, and completed iteration would be 0 and our input file should be like;

1/1%1/1%1/0/2/3/100/0
2/-1%1/1%1/0/5/3/100/0

-1/-1%1/1%1/0/2/3/100/0
-1/2%1/1%1/0/5/3/100/0

It is up to user to decide how many waves or iteration will be used. These values are illustrative. Format for text and values also illustrative it is possible to change values according to need, for example it is possible to add initial values to these values to know where was the initial point. MapReduce has ways to use complex parameters as value it is also possible to use these.

We did not used key values for any significant purpose. But it should be noted that key values may contain some information about optimization algorithm for reducing purposes. For example it is possible to use fitness values as key values to decide which of the new points should be used in the output and/or prevent more than one point with same fitness exist in the output.

It is also up to the user to decide how many new point will be generated through each map operation. It is possible to use best n points to create new output. It should also be noted that by using additional parameters it may also be possible to change algorithm to work as multi objective.

V. BENCHMARK FUNCTIONS

We have used some of the original papers test functions to test our application in MapReduce platform which includes:

F1 Ackley's Function;

$$f(x,y) = -20 \times \exp\left(-0.2 \times \sqrt{0.5 \times \left(x^2 + y^2\right)}\right) \exp(0.5 \times (\cos(2\pi x) + \cos(2\pi y))) + e + 20 \quad (4)$$

F2 Beale's Function;

$$f(x,y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2 \quad (5)$$

F3 Six Hump Camelback Function;

$$f(x,y) = (4 - 2.1x^2 + \frac{x^4}{3}) \times x^2 + xy + (-4 + 4y^2) \times y^2 \quad (6)$$

F4 Goldstein-Price Function;

$$f(x,y) = (1 + (x + y + 1)^2 \times (19 - 14x + 3x^2 - 14y + 6xy + 3y^2)) (30 + (2x - 3y)^2 (18 - 32x + 12x^2 + 48y - 36xy + 27y^2)) \quad (7)$$

F5 Lévi Function N.13;

$$f(x,y) = \sin^2(3\pi x) + (x-1)^2 (1 + \sin^2(3\pi y)) + (y-1)^2 (1 + \sin^2(2\pi y)) \quad (8)$$

F6Hölder Table Function;

$$f(x,y) = -\left|\sin(x)\cos(y)\exp\left(\left|1 - \frac{\sqrt{x^2+y^2}}{\pi}\right|\right)\right| \quad (9)$$

F7 Sphere Function;

$$f(x) = \sum_{i=0}^{n} x_i^2 \quad (10)$$

Solving these functions does not require a lot of process power, however because we have used a single node cluster setup; they were merely used to prove algorithm would work in a MapReduce framework.

Experimental results are showing that in MapReduce platform gives similar resultsto ordinary algorithm which proves for optimization problems that needs lots of calculation it is possible to use MapReduce platform to divide work load and find solutions in a distributed environment.

In Table I search domain and minimum point of each function is provided.

TABLE I. FUNCTIONS

Function	Search domain	Minimum
F1 (4)	-5≤x,y≤5	f(0,0)=0
F2 (5)	-4.5≤x,y≤4.5	f(3,0.5)=0
F3 (6)	-3<x<3 -2≤x,y≤2	f(0.0898,-0.7126) =1.0316 f(-0.0898,0.7126) =1.0316
F4 (7)	-2≤x,y≤2	f(0,-1) = 3
F5 (8)	-10≤x,y≤10	f(1,1) = 0
F6 (9)	-10≤x,y≤10	f(+8.05502, +9.66459) =-19.2085
F7 (10)	5.12 < xi < 5.12, i = 1, 2, . . . , n	x* = (0, . . . , 0), f(x*) = 0.

VI. EXPERIMENTAL RESULTS

We have used single line input file for each function which is provided in Table II.

TABLE II. FUNCTIONS

FUNCTION	INPUT
F1	3%-3/2%2/0/10000000/1000000/3/5/0
F2	0%0/1.5%1.5/0/1000/1000/3/5/0
F3	0%0/1%1/0/1000/1000/3/5/0
F4	0%0/0.66%0.66/0/10000000/100000000/3/5/0
F5	0%0/3.33%3.33/0/1000/1000/3/5/0
F6	0%0/3.3%3.3/0/100000000/1000000000/3/5/0
F/	3%-3/2%2/0/1000/1000/3/5/0

TABLE III. EXPERIMENTAL RESULTS

FUNCTION	OUTPUT
F1	-9.500041121023957E-16%1.0863446880863609E-15/5.14 7557589468029E-85%5.147557589468029E-85/243/3.552 713678800501E-15/3.55271367880050E-15/3/5/61
F2	2.9999991487773214%0.49999978506604986/1.43051147 4609375E-6%1.430511474609375E-6/1/1.1632042404531 215E-13/1.1632042404531215E-13/3/5/280
F3	-0.089842013607745 32%0.7126564029962014/4.8407398 93561648E-122%4.840739893561648E-122/387/-1.031628 4534898774/-1.0316284534898774/3/5/27
F4	-6.6128221219783375E-9%-1.0000000030251963/6.86097 0445325634E-113%6.860970445325634E-113/350/2.9999 99999999921/2.999999999999921/3/5/42
F5	1.0%1.0/2.8084415320766114E-80%2.8084415320766114 E-80/226/1.3497838043956716E-31/1.3497838043956716 E-31/3/5/65
F6	-8.055023465925194%-9.664590026085996/8.5762130566 57041E-113%8.576213056657041E-113/348/-19.20850256 7886754/-19.208502567886754/3/5/34
F7	-6.527455315601574E-32%1.085325270641747E-31/1.577 7218104420236E-30%1.5777218104420236E-30/1/1.6040 07672065334E-62/1.604007672065334E-62/3/5/123

For these experiments radius values are 1/3 of the original search range, initial fail count zero, fitness are

suitable big numbers, number of waves three, number of iteration five, and completed iterations are taken as zero. Experimental result for each function is provided in Table III. For this experiment we have chained 500 map functions.The output file has same format as input file.

The solutions which provided by output files are consistent with the real solutions of these functions.However it should be noted that algorithm is using more process power than it is necessary for finding these functions minima because it is working even after finding the minima. This is one downside of working with MapReduce because it is not possible to end the process after finding a solution. All map functions must be completed in order to end the process so finalizing with a condition is not possible. Still it is possible to use an output file as an input file in case solution couldn't be found in output of given map functions. In this regard it is possible to say that memory of the work is saved in the output file. Using appropriate number of map functions may be vital in working with MapReduce.

CONCLUSION

In this paper, a novel metaheuristic optimization algorithm entitled as CWW inspired from circular water waves have been used in MapReduce platform to provide distributed optimization solution.

We have tested CWW algorithm in MapReduce platform in pseudo-distributed form with seven different benchmark functions and seen that solutions provided with output files are consistent with the real solutions of these functions.

As results stands algorithm is capable of working in MapReduce platform. However benchmark functions which we used are not costly functions in terms of process power, for this reason experiment with more costly functions must be made for future work.

Using MapReduce as distributed optimization may provide fast and easy to implement solution for costly functions.

REFERENCES

[1] J. Kennedy, R. Eberhart, "Particle Swarm Optimization". *Proceedings of IEEE International Conference on Neural NetworksIV*. pp. 1942–1948, Nov. 1995.

[2] D. Karaboga, B. Basturk, "A powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", *Journal of Global Optimization*, Volume:39, Issue:3, pp:459-171, Nov. 2007, ISSN:0925-5001 , doi: 10.1007/s10898-007-9149-x

[3] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company Inc., 1989 ISBN 0-201-15767-5.

[4] Yang, X.-S. *Nature-Inspired Metaheuristic Algorithms, Second Edition*. Luniver Press, pp. 81-89 2008.

[5] Yang, X.-S. *Nature-Inspired Metaheuristic Algorithms, Second Edition*. Luniver Press, pp. 97-103 2008.

[6] B.Alatas "ACROA: Artificial Chemical Reaction Optimization Algorithm for global optimization" *Expert Systems with Applications* Vol. 38, Issue 10, 15 September 2011, pp 13170–13180

[7] E. Nabil "A Modified Flower Pollination Algorithm for Global Optimization" *Expert Systems with Applications* Volume 57, 15 September 2016, Pages 192–203

[8] S.M. Mahmoudi, M. Aghaie,M. Bahonar, N. Poursalehi"A novel optimization method, Gravitational Search Algorithm (GSA), for PWR core optimization"Annals of Nuclear EnergyVolume 95, September 2016, Pages 23–34

[9] P. Lopez-Garcia, E. Onieva, E. Osaba, , A.D. Masegosa, A. Perallos "GACE: A meta-heuristic based in the hybridization of Genetic Algorithms and Cross Entropy methods for continuous optimization" Expert Systems with ApplicationsVolume 55, 15 August 2016, Pages 508–519

[10] AlirezaAskarzadeh "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm" Computers & StructuresVolume 169, June 2016, Pages 1–12

[11] M. E. ÇOLAK, A. VAROL" and A Novel Intelligent Optimization Algorithm Inspired from Circular Water Waves" ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 21, NO. 5, 2015

[12] Yanqing Jia , , Yun Tian, , FangyangShenc , John Tran "Leveraging MapReduce to efficiently extract associations between biomedical concepts from large text data" Microprocessors and Microsystems Available online 11 March 2016In Press, Corrected Proof — Note to users

[13] Junchang Xin, Zhiqiong Wang, Luxuan Qu, Ge Yu, Yan Kang"A-ELM□: Adaptive Distributed Extreme Learning Machine with MapReduce" Neurocomputing Volume 174, Part A, 22 January 2016, Pages 368–374

[14] [Chun-Yang Zhan, C.L. Philip Chen, Dewang Chen, , Kin Tek NG, "MapReduce based distributed learning algorithm for Restricted Boltzmann Machine" Neurocomputing Available online 17 March 2016 In Press, Corrected Proof — Note to users

[15] AlessioBechini, Francesco Marcelloni, Armando Segatori"A MapReduce solution for associative classification of big data" Information Sciences Volume 332, 1 March 2016, Pages 33–55

[16] [e] Saeed Shahrivari, Saeed Jalili "Single-pass and linear-time k-means clustering based on MapReduce" Information Systems Volume 60, August–September 2016, Pages 1–12

[17] [f] Xiu Li, Jingdong Song, Fan Zhang, Xiaogang Ouyang, Samee U. Khan "MapReduce-based fast fuzzy c-means algorithm for large-scale underwater image segmentation" Future Generation Computer Systems Available online 28 March 2016 In Press, Accepted Manuscript

[18] **https://aws.amazon.com/elasticmapreduceaccesed date:3.3.2016**

★★★