2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)

Energy–aware Discrete Symbiotic Organism Search Optimization algorithm for task scheduling in a cloud environment

Megha Sharma Amandeep Verma UIET, Panjab University UIET, Panjab University Chandigarh, India Chandigarh, India E-mail: megha1892@gmail.com E-mail: amandeepverma@pu.ac.in

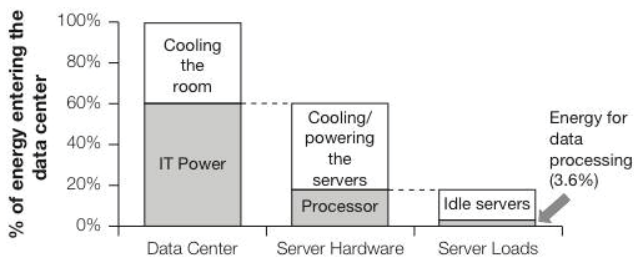
***Abstract*— With progress in technology and unceasing improvement in the standard of living, the practical scope of cloud computing is increasing day by day. Over the years, power consumption has become an important cost factor for computing resources. Because of this, management of energy has become critical for the implementation of networks and the pivotal target of all algorithms. A new metaheuristic task scheduling algorithm Discrete Symbiotic Organism Search (DSOS) algorithm was proposed recently, which focused on optimizing makespan but not from energy efficiency point of view. In this paper, a new energy aware optimization approach towards DSOS known as Energy aware Discrete Symbiotic Organism Search (E-DSOS) has been proposed for small-scale cloud networks. The detailed description and performance test results of both the original algorithm and its energy-efficient version are included. The experimental results prove that E-DSOS consumes lesser energy as the problem size increases.**

Keywords*— *Energy Efficiency, Cloud Computing, Neural Networks, Discrete Symbiotic Organisms Search.

I.INTRODUCTION

Cloud computing has been playing a vital role in our daily lives and is significantly changing the way individuals expend data. This is all becoming possible due to the fact that technology and gadgets are becoming more and more affordable day by day for the common folk. There exist around 1.5 billion Personal Computers [1] and 6 billion cell phones [2] in the world today, which tend to varying needs of people of all ages. In such a situation, datainferring, gathering and communication turn into a need, and Cloud computing is one viable method for offering all of these above mentioned features along with other additional features like enhanced security, and minimal cost [3]. Cloud computing services are provided by the service providers, which in turn depend on the data center industry, with more than 500,000 server farms set up around the globe [4]. The functioning of such broadly disseminated data centers, in any case, requires a lot of energy for processing and cooling purposes, which represents a substantial part of the

aggregate has assessed that energy dissipation estimates up to 10 percent ofthe present data centers' operational costs (OPEX), with this gauge conceivably ascending to 50 percentby 2020 [7]. Also, the cost of energy for running infrastructure like switches, servers, routers, *etc.* may as of now cost more than the cost of the equipment itself [8] [9]. Currently the running of data centers accounts to more than 50 million metric cube of CO2 outflows every year [10]. Fig. 1 [11] describes the energy usage in data centers worldwide-



Source: IBM

Fig. 1.*Energy usage in a data center*

Energy management has never been an objective in the Information and Communication Technologies (ICT) industry in the past. Since the time this industry has begun to boom, the holy grail has been to convey progressively and in a faster manner. Notably, this has been customarily accomplished by pressing more into smaller capacity and running processors at a higher frequency. This devours more power, which creates more heat, and afterward requires a cooling framework the expense of which ranges from $2 to $5 million every year for corporate server farms [7]. These cooling frameworks may even require more power than that consumed by the IT hardware itself [12], [13]. Therefore it is the need of the hour to have an energy efficient approach towards cloud computing. Green computing [14] is the future of cloud with tremendous research being done in this

aggregate	operational expenses	[5][6].	The	Gartner	field since now the empha-sis is on saving the environ-ment.

In this paper, we make a metaheuristic task-scheduling

978-1-5090-2797-2/17/$31.00 ©2017 IEEE 513

algorithm known as Discrete Symbiotic Organisms Search (DSOS) energy efficient *i.e.* E-DSOS. The results clearly show reduction in energy as we compare the results of

where *f1* and *f2* are mutual benefit factors selected randomly as either 1 or 2. *r'* and *r''* arerandomly generated numbers between 0 and 1. Also,*xi* and *xj* are updated after every

previous work to the proposed work. The organization of the iteration as per *xbest.*

remainder of the paper is as follows: section 2 describes the proposed methodology for E-DSOS while shedding light on its parent algorithm *i.e.* DSOS, section 3 provides us with experimental results and its discussion and section 4 gives the conclusion as well as future scope.

II.PROPOSED METHODOLOGY

The previous section discusses the need of energy efficiency in the field of cloud computing. Working for a similar goal, we have made DSOS, which was previously solely a task scheduling algorithm into E-DSOS algorithm which is energy-efficient. Section II is divided into two parts. The first section gives an overview of the Discrete Symbiotic Organisms Search (DSOS) algorithm. The second section provides the methodology used to make DSOS energy

Step 4: Commensalism phase
In this phase,*xi* benefits from the randomly selected variable *xj* but the case is not the same for*xj* as it may benefit from *xi* or remain neutral.

$x'i = xi + r'(xbest - xj)$(3)
Step 5: Parasitism phase
A parasite vector *xp*is created by mutating*xi*with a randomly generated variable within a certain range. Also,*xj* serves as host to *xp*. If $f\ (xp) > f(xj)$,*xp*replaces*xj*, else it is discarded. *Step 6: Stopping criterion.*

Stopping criteria is reached when all the organisms pass throughthe above mentioned phases.

efficient. *A.Discrete Symbiotic Organisms Search (DSOS) Algorithm*	The scheduling decisions are taken on the basis of ETC (Expected Time to Compute) of each task. ETC values of tasks are calculated by dividing Million Instructions Per

Discrete Symbiotic Organisms Search (DSOS) [15] algorithm is a discrete approach towards Symbiotic Organisms Search (SOS) [16] algorithm,proposed by Abdullahi *et al.*for the purpose of task scheduling. It is a nature-inspired population-based algorithm. The algorithm has a similar process as SOS along with some added

discretization	functions.Basically like	its	parent

algorithm,DSOS imitates the following symbiotic behaviors

second (MIPS) with the length of task [17]. Also ETC values are represented by a matrix such that the number of tasks to be scheduled is represented by rows of a matrix and number of available VMs is represented by its columns [18]. Each row of ETC matrix represents execution time of a given task for each VM, while each column represents execution time of each task on a given VM. Fitness value of each organism is calculated as:

of organisms belonging to the ecosystem- *Mutualism, Commensalism* and *Parasitism*. The details of symbiotic interaction are described

$$fitness = max\{Cij, \text{all tasks } j \text{ mapped to } VM\ i\}$$

$$i\{1,2,3,\ldots,n\} \& j\{1,2,3,\ldots,m\} (4),$$

below. In each iteration (till the maximum number of iterations is reached), every organism *xi* (i = 1,...,N) goes through the following three phases.

where, m is the number of tasks, n is the number of VMs and *Cij* is the execution time of executing *ith* task on *jth* VM.

Step 1: Ecosystem initialization B.System Model

In this step, population of organisms is created and initialized. Also, control variables such as the maximum number of iterations to be allowed, the size of ecosystem,

DSOS was designed with the objective of task scheduling. In this paper, we have worked on the objective of making this algorithm more energy efficientand we call this proposed

etc. are specified. algorithm as E-DSOS. The basic model for this algorithm is

*Step 2:*All the organisms are analysed according to their fitness functions. The organism which has the best fitted objective function is known as *xbest.*

Step 3: Mutualism phase

For the *ith* iteration, an organism *xi* is selected to interact with a randomly selected variable *xj* for the purpose of mutual benefit *(i j)*. This is shown in equations (1) and (2)

the same as other algorithms *i.e.* it consists of a number of tasks which are to be mapped onto a limited number of virtual machines in the best way possible. As mentioned already DSOS works in three phases. Schedules are fed to these three phases to generate the best schedule possible for executing set of tasks. After mutualism, commensalism and parasitism, we get the minimum time to complete all the tasks. From this time, we have calculated energy as follows:

respectively:	(5)

*x'i = xi + r'(xbest – (xi+xj)/2*f1)*(1) *x'j = xj + r''(xbest – (xi+xj)/2*f2)*(2)

514

2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)

parasitism depend upon *xbest*. The smaller the value of *xbest*, the smaller the time is taken to complete a schedule. We have achieved low value of *xbest* with the help of neural

3. For i=1:schedule_count, a random schedule j is selected for every *i* such that (i!=j) and following processes are applied:

networks as described below. - Mutualism

- Commensalism

The neural network technique: - Parasite

An Artificial Neural Network (ANN) comprises of a large 4.The best schedule has the minimum execution

number of distributed processing elements with parallel time.

information processing among the connected elements. Each processing element has multiple inputs and a single

output.Dynamic	weight	adjustments	give	learning

capabilities to the networks and information is passed as signals which is why neural network is suitable for intelligent task scheduling strategies.

The neural network approach used in this algorithm is Lavenberg-Marquardt method [19]. It works to minimize the sum of the squares of the deviations. It can also be termed as Mean Square Error (MSE) which is the average squared error between network operations and the target operations. Using neural networks gives improved schedules which in turn gives a minimized *xbest*.

The methodology is divided into three algorithms to provide clarity:

Algorithm 1: Generation of Cloud Environment

1.Population of organisms is created and initialized. This is done by:

• •	Every virtual machine is randomly provided with MIPS at every instance. Every job is independent and is randomly provided with task length.

2.Expected time to Compute (ETC) is calculated for every organism by taking the ratio of million instructions per second (MIPS) of a VM to the length of the task.

Algorithm 2: DSOS Algorithm

1.From training data, every possible schedule is generated.

2.The minimum valued schedule is considered as *xbest* and is used in the following steps to update other schedules.

3.For i=1:schedule_count, a random schedule j is selected for every i such that (i!=j) and following processes are applied:
- Mutualism
- Commensalism
-
Parasite 4.The best schedule has the minimum execution time.

5.Process is terminated as per stopping criteria. 6.Energy for the schedule is calculated.

Algorithm 3: Energy Efficient DSOS algorithm

1.In this algorithm, training value is used to train a three layer ANN and yield improved schedules. 2.Improved *xbest* is obtained.

5.Process is terminated as per stopping criteria.

6.Energy is calculated for this schedule.

III.RESULTS AND DISCUSSION

Both DSOS and E-DSOS are implemented in MATLAB since it has internal support for neural network. A common data center is created with 20 virtual machines. The system model is heterogeneous; meaning that at any instant processing speeds of virtual machines can vary. Number of jobs at any instance is fed manually to the system. Range of MIPS for VM is taken from 100 to 1000 MIPS. It is assumed that the pool of target machines has enough memory and storage and communication links between these machines is uniform. Task distribution is taken as left-skewed, right-skewed and normal distribution. In left-skewed distribution there are more large-sized tasks than small-sized ones while in right-skewed distribution it's the reverse. In uniform distribution there is equal number of small, medium and large sized tasks. For each distribution, tasks range from 100 to 1000 for scalability of performance. Table I specifies parameters of E-DSOS along with their size.

TABLE I. SPECIFICATION OF PARAMETERS

Parameter Size

Number of VMs 20 Number of jobs Variable Job type Random Mips (1000-10000)

In this experiment, the number of tasks is varied from 100 to 1000 and energy is obtained for both DSOS and E-DSOS over this range of tasks.Task size varies from (1-30) Million Instructions. Also three task distributions have been

performed	*i.e.*	left-skewed	distribution,	right-skewed

distribution and uniform distribution.

The results have been described using these three scenarios as follows:

(i)**Scenario 1**: *Left-skewed distribution (having small number of large weight tasks and large number of small weight tasks)*

Table II shows the energy dissipation by DSOS algorithm as well as E-DSOS algorithm for left-skewed distribution. Note that for every set of tasks energy of E-DSOS is much lower than that of DSOS. Energy improvement ranges from 6.1% to 24.43 % which is very significant. For comparison of the two, a graph is plotted which clearly shows proposed work consuming much less energy than the original algorithm.

515

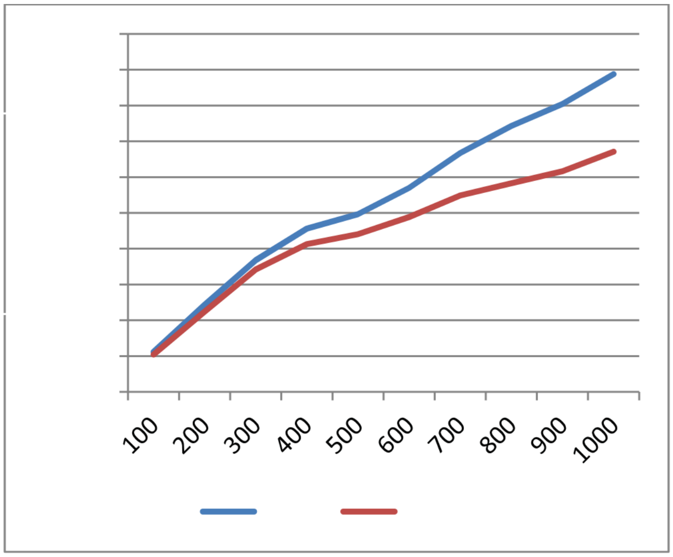


TABLE II. ENERGY OBTAINED IN CASE OF LEFT SKEWED DISTRIBUTION (IN JOULES)

400	44854	41023	8.54
500	49779	45712	8.17
600	57320	48899	14.69
700	66817	58050	13.12
800	73290	60420	17.56
		54	19.88
		26	21.95

100000
90000
80000
70000
60000
50000

Energy (in joules)

n
jo

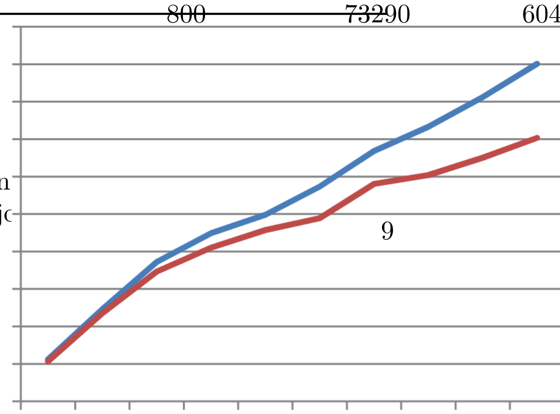


TABLE II. ENERGY OBTAINED IN CASE OF LEFT SKEWED DISTRIBUTION (IN JOULES)

400	44854	41023	8.54
500	49779	45712	8.17
600	57320	48899	14.69
700	66817	58050	13.12
800	73290	60420	17.56
900	81196	65054	19.88
1000	90104	70326	21.95

40000
30000
20000
10000
0

100000
90000
80000
70000
60000
50000
40000
30000
20000
10000
0

100 200 300 400 500 600 700 800 900 1000

DSOS

E-
D

Number of tasks

Energy (J

Fig. 3. *Comparison graph of energy for right-skewed distribution*

Table III demonstrates the energy efficiency graph for right-skewed distribution. Energy improvement ranges from 4.6% to 21.95 %. A steep curve is obtained for proposed work in

			the graph.
DSOS	th	E-DSOS	(iii)**Scenario 3**: Uniform distribution (having equal number of small, medium and large sized tasks)

Number of tasks

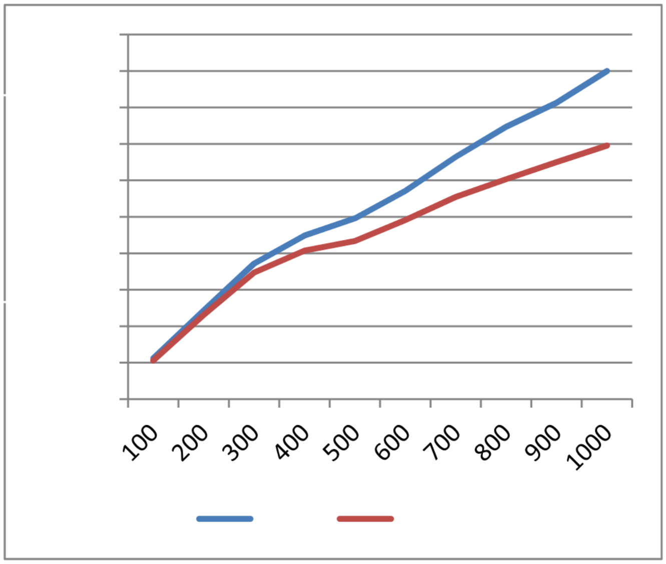
Fig. 2.*Comparison graph of energy for left-skewed distribution*

(ii)**Scenario 2**: *Right-skewed distribution (having large number of large weight tasks and few number of small weight tasks)*

TABLE III. ENERGY OBTAINED IN CASE OF RIGHT SKEWED DISTRIBUTION (IN JOULES)

Right-skewed distribution			
Number of jobs	***DSOS***	***E-DSOS***	***Improvement (%)***
100	11170	10656	4.6
200	24542	23521	4.2
300	37210	34638	6.91

516



2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)

900	81289	64991	20.04
1000	89982	69529	22.73

efficiency, it can prove to be an excellent metaheuristic algorithm in the future.

IV.CONCLUSION AND FUTURE SCOPE

	100000	Reduction in energy consumption is the most critical step towards green
	90000	computing. The proposed approach is to make the recently pioneer work in this paper discovered algorithm (DSOS)
Energy (in joules)	80000	energy-efficient. This is achieved with the help of a three
	70000	layered Artificial Neural Network. The enhanced version of
	60000	DSOS is called E-DSOS. The result clearly shows reduction
	50000	in energy dissipation by E-DSOS. This algorithm can be
	40000	applied in a heterogeneous environment to various parallel applications.
	30000	In future, this algorithm can be made more
	20000	suitable for the real time environment.
	10000	System reliability could be taken into account. It can also be helpful to adjust all the parameters to real time environment by various type of
	0	testing techniques.

Parameters like voltage or frequency switching overhead, communication overhead and other uncertain ones will be

DSOS E-DSOS

Number of tasks

Fig. 4. *Comparison graph of energy for uniform distribution*

Energy improvement ranges from 5.26 % to 22.73% for uniform distribution as shown in Table IV. Left-skewed distribution shows the best results out of all distributions.

The following observations are made from the above data:

1.Left-skewed distribution gives the maximum energy efficiency out of all the distributions. This concludes that our proposed technique is most suited for small sized tasks. Since right-skewed distribution gives slightly minimum energy-efficiency out of all three distributions, we can also conclude that this model of energy improvement is less suitable for large sized tasks.

2.As the number of tasks increases, energy consumption also increases significantly. This is due to the fact that with more number of tasks, more number of schedules have to be made and the maximum amount VMs in this experiment is 20.

3.With the increase in number of tasks, difference of energy between both algorithms increases significantly. Therefore E-DSOS becomes more efficient as the

considered for a real heterogeneous environment.

REFERENCES

[1] World Internet Usage and Population Statistics, *http://www.internetworldstats.com/.*

[2] "Forecast: Mobile Data Traffic and Revenue, Worldwide, 2010-2015", *Market report, Gartner Inc.*, 2011.

[3] Weiss, Aaron, "Computing in the clouds", *netWorker - Cloud computing: PC functions move onto the web*, vol. 11, no. 4, 2007, pp. 16–25.

[4] Power, Emerson Network. "State of the Data Center 2011.", 2011.

[5] Fan, Xiaobo, Wolf-Dietrich Weber, and Luiz Andre Barroso. "Power provisioning for a warehouse-sized computer." *In Proceedings of the 34th annual international symposium on Computer architecture*, pp. 13-23, 2007.

[6] Raghavendra, Ramya, Parthasarathy Ranganathan, Vanish Talwar, Zhikui Wang, and Xiaoyun Zhu. "No power struggles: Coordinated multi-level power management for the data center." *In Proceedings of the 13th international conference on Architectural support for programming languages and operating systems*, pp.

[7] Gartner Group, http://www.gartner.com/.

[8] Vasan, Arunchandar, Anand Sivasubramaniam, Vikrant Shimpi, T. Sivabalan, and Rajesh Subbiah. "Worth their watts?-an empirical

Sep.

DSOS E-DSOS

Number of tasks

Fig. 4. *Comparison graph of energy for uniform distribution*

Energy improvement ranges from 5.26 % to 22.73% for uniform distribution as shown in Table IV. Left-skewed distribution shows the best results out of all distributions.

The following observations are made from the above data:

1.Left-skewed distribution gives the maximum energy efficiency out of all the distributions. This concludes that our proposed technique is most suited for small sized tasks. Since right-skewed distribution gives slightly minimum energy-efficiency out of all three distributions, we can also conclude that this model of energy improvement is less suitable for large sized tasks.

2.As the number of tasks increases, energy consumption also increases significantly. This is due to the fact that with more number of tasks, more number of schedules have to be made and the maximum amount VMs in this experiment is 20.

3.With the increase in number of tasks, difference of energy between both algorithms increases significantly. Therefore E-DSOS becomes more efficient as the

considered for a real heterogeneous environment.

REFERENCES

[1] World Internet Usage and Population Statistics, *http://www.internetworldstats.com/.*

[2] "Forecast: Mobile Data Traffic and Revenue, Worldwide, 2010-2015", *Market report, Gartner Inc.*, 2011.

[3] Weiss, Aaron, "Computing in the clouds", *netWorker - Cloud computing: PC functions move onto the web*, vol. 11, no. 4, 2007, pp. 16–25.

[4] Power, Emerson Network. "State of the Data Center 2011.", 2011.

[5] Fan, Xiaobo, Wolf-Dietrich Weber, and Luiz Andre Barroso. "Power provisioning for a warehouse-sized computer." *In Proceedings of the 34th annual international symposium on Computer architecture*, pp. 13-23, 2007.

[6] Raghavendra, Ramya, Parthasarathy Ranganathan, Vanish Talwar, Zhikui Wang, and Xiaoyun Zhu. "No power struggles: Coordinated multi-level power management for the data center." *In Proceedings of the 13th international conference on Architectural support for programming languages and operating systems*, pp.

[7] Gartner Group, http://www.gartner.com/.

[8] Vasan, Arunchandar, Anand Sivasubramaniam, Vikrant Shimpi, T. Sivabalan, and Rajesh Subbiah. "Worth their watts?-an empirical

Clearly, the experiment shows significant energy reduction in E-DSOS as compared to the DSOS algorithm. The range of reduction for all three distribution ranges from 4.6 to 24.43 %. It shows that an improvement in the quality of schedules can significantly impact the time to compute tasks which directly impacts the energy consumption. DSOS has a few parameters and is easier to implement which is considered its advantage in addition to the explorative and exploitative ability. Therefore E-DSOS is also easy to implement and with added advantage of improved energy

2006-2010 forecast." *Market analysis, IDC Inc,* 2006.

[10]Koomey, Jonathan. "Growth in data center electricity use 2005 to 2010." *A report by Analytical Press, completed at the request of The New York Times* 9, 2011.

[11]Scott, Inara. "Antitrust and Socially Responsible Collaboration: A Chilling Combination?," *American Business Law Journal*,pp. 97-145, 2016.

[12]Atwood, Don, and J. G. Miner. "Reducing data center cost with an air economizer." *White Paper: Intel Corporation*, 2008.

[13]Rasmussen, Neil. "Calculating total cooling requirements for data centers." *White paper* 25, pp. 1-8, 2007.

[14]Li, Qilin, and Mingtian Zhou. "The survey and future evolution of green computing." In *Proceedings of the 2011 IEEE/ACM International Conference on Green Computing and Communications*, pp. 230-233, 2011.

[15]Abdullahi, Mohammed, and Md Asri Ngadi. "Symbiotic Organism Search optimization based task scheduling in cloud computing environment." in *Journal of Future Generation Computer Systems*, vol.56, 2016, pp. 640-650.

[16]Cheng, Min-Yuan, and Doddy Prayogo. "Symbiotic organisms search: a new metaheuristic optimization algorithm." *Journal of Computers & Structures,* vol.139, 2014, pp. 98-112.

[17]Loo, Sin Ming, and Earl Wells. "Task scheduling in a finite-

resource,	reconfigurable	hardware/software	codesign

environment." *INFORMS Journal on Computing*, vol. 18, no. 2, 2006, pp. 151-172.

[18]Demiroz, Betul, and Haluk Rahmi Topcuoglu. "Static task scheduling with a unified objective on time and resource domains." *The Computer Journal*, vol.49, no. 6, 2006, pp. 731-743.

[19]Mishra, Deepak, Abhishek Yadav, Sudipta Ray, and Prem Kalra.

"Levenberg-Marquardt learning algorithm for integrate-and-fire neuron model." *Neural Information Processing-Letters and Reviews*, vol.9, no. 2, 2005, pp. 41-51.

518