A Comparative Study of the Application of Glowworm Swarm Optimization Algorithm with other Nature-Inspired Algorithms in the Network Load Balancing Problem

Talha Akhtar
Department of Computer Science and Information Technology
NED University of Engineering & Technology
Karachi, Pakistan
talhaakhtar@nbp.com.pk

Najmi Ghani Haider
Department of Computer Science and Information Technology
NED University of Engineering & Technology
Karachi, Pakistan
najmi@neduet.edu.pk

Shariq Mahmood Khan
Department of Computer Science and Information Technology
NED University of Engineering & Technology
Karachi, Pakistan
shariq@neduet.edu.pk

Received: 20 April 20222 | Revised: 7 May 2022 | Accepted: 12 May 2022

Abstract-Vast amounts of data are transferred through communication networks resulting in node congestion, which varies according to peak usage times. The Glowworm Swarm Optimization (GSO) algorithm is inspired by the rummaging and courtship behavior of glowworms. The glow intensity of glowworms is a measure of fitness that attracts other glowworms in its neighborhood. This work applies the GSO algorithm to the computer network congestion problem in order to lessen the network burden by shifting loads to the fittest neighborhood nodes, thereby enhancing network performance during peak traffic times, when the response of systems on the network would go down. The proposed solution aims to alleviate the burdened nodes, thereby improving the flow of traffic throughout the network, improving the users' experience and productivity, and efficiency. In this paper, three swarm algorithms, namely Particle Swarm Optimization (PSO), Cuckoo Search (CK), and GSO have been employed to solve the network load balancing problem. The results produced by GSO show improvement of 71.17%, 74.14%, and 84.15% in networks consisting of 50, 100, and 200 nodes in peak hour load, while PSO shows 13.87%, 11.75%, and 23.72%, and CK 10.61%, 3.19%, and 6%. The results prove the superior performance of GSO.

Keywords-network congestion; load balancing; GSO; throughput; swarm intelligence

I. INTRODUCTION

The decentralized computer network system is a mixture of heavily and lightly loaded nodes, with the heavily loaded nodes becoming bottlenecks, thereby reducing traffic flow rate, causing system sluggish response on the networks. Load balancing addresses this issue to improve the network throughput by routing packets to adjacent edges having lesser traffic load. It enables the reasonable distribution of resources on computer networks and thereby a more efficient utilization of resources. The network burden at an edge depends on the queue of tasks that need to be processed through that node. The queue length is related to the response time of the jobs and is an important indicator of load burden on a computer network [1].

The effectiveness of conventional algorithms for load balancing in computer networks (such as the round Robin task scheduler and least connection [2]) will be adversely impacted due to the explosive increase of data coming from mobile devices or tablets. Data usage is expected to soon exceed 142 Exabyte a month [3]. Additionally, it is estimated that there will be 57.7 billion electronic and IoT devices linked to the Internet by 2023 [4]. All these devices will link to the network via technologies like 2G, 3G, 4G, LTE,5G and Machine-to-Machine (M2M) with wireless options like Bluetooth, Wi-Fi and Zig-Bee. Various approaches have been used to address this problem.

The meta-heuristic approach to optimization problems, such as the use of nature-inspired algorithms [5], has attracted the attention of researchers. The term Swarm Intelligence was introduced by Benni and Wang in 1989 in the global optimization framework of cellular robotics [6]. Swarm Intelligence follows the distributed methodology of governing

Corresponding author: Talha Akhtar

of work in chronological direction. This strategy is motivated from the behavior of ants searching for food in collaboration in appropriate order with nest mates. A swarm is an enormous group of homogeneous agents that are locally interacting among themselves. There is no central control and they are globally distributed in a decentralized manner. The idea of Ant Colony Optimization (ACO) algorithm comes from the movement of Ants foraging for food and its delivery into the nest [7-9]. The ACO is applied for finding the shortest route by emitting chemical substances (pheromones), updating the following path, therefore all ants move like a trail bridge. It is applicable in the travelling salesman problem for calculating the optimal path among different cities. A novel approach of Particle Swarm Optimization (PSO) was introduced in 1995 [10, 11] which is inspired from flocks of birds and schools of fish and their social behavior. The PSO is used for accessing a target with minimal duration by updating the particle's current velocity and position from its neighborhood. The Artificial Bee Colony (ABC) algorithm's [12] inspiration is the behavior of honeybees foraging foods to make honey in a nest. It transfers a message to its nest mate with the help of a wiggle dance. The Cuckoo Search (CK) algorithm is based on the cuckoo bird's life and breeding scheme [13]. Glowworm Swarm Optimization (GSO) is investigated in this paper with regard to network load balancing. The idea is to mitigate the traffic load burden during peak hours using a meta-heuristic approach about network optimization in IoT. Due to increasing massive amounts of data traffic on computer network the decentralized approach is attracting the researchers' attention. The main objective of load balancing method is to allocate the traffic load uniformly dispersed to achieve best performance on homogeneous and heterogeneous data networks. Load balancing is used to enhance user fulfillment, better utilization of resources, execution time, and waiting time of task coming from several locations/nodes. To do this, a load balancer needs to be able to predict the load on those edges having lesser amounts of traffic. It can be a static or sequential distribution of tasks among the machines in the neighborhood. One drawback is that all nodes can become overloaded. A dynamic technique in swarm intelligence based approach, such as ACO, is one in which data would be segregated on to those nodes that are having the highest amount of pheromone evaporation created as a path. The ants find the shortest route to achieve traffic in minimum amount of time. The Honey Bee in ABC uses another approach to optimize traffic on the optimal path. The main strategy for finding food location, to pass the message to other nest mates is by a waggle dance to find good food sources.

In both wired and wireless media, network packets are moved between nodes by routing. Routing techniques are of two types: static and dynamic. Routing can be based on centralized or decentralized techniques. In the centralized method all nodes are linked into a star configuration [14]. Compared to decentralized systems, centralized systems do not require replication. However, the disadvantage is that when the midpoint of a star network is down, the entire network becomes detached. The swarm intelligence algorithms follow the decentralized scheme of operation. The GSO algorithm is based on multimodal functions to find a set of local optimal solutions. Applying the analogy of glowworm behavior to communication networks, the network traffic can be shifted to neighborhood glowworm nodes having the highest luminance levels [15, 16].

Large quantities of data from diverse devices and applications traverse through a network. Network optimization has benefits such as faster information rate, information recovery, and elimination of redundant records enabling perceptible improvement in load balancing which plays a crucial role in network performance. The goals of this investigation are:

- To reduce the queue length and thereby divert network traffic to adjacent lighter loaded nodes which in turn would improve network performance.
- To increase the throughput and decrease response time and delay in network systems through the application of GSO.
- The comparison of load balancing via GSO and other nature-inspired algorithms.

Load balancing is a mechanism that distributes a fair allocation of resources among the network nodes. Load balancing methods can be implemented either in hardware or in software. In hardware balancing, there are four layers in a computer network. Switches offer server redundancy and load balancing. Software balancing is used to calculate usage and allocate resources. Furthermore, load balancing can be static and dynamic. In static load balancing, the predictable load is segregated among nodes which are connected directly or indirectly to the network. Dynamic load balancing is used to deal with unpredictable load which can be adjusted on various conditions of homogeneous and heterogeneous data passing through hybrid networks. Static load balancing utilizes the surveying strategy (the round Robin approach). The technique disperses data sequentially to each designated host, however it can be unbalanced among various virtual machines during data distribution [17]. The dynamic load balancing method is primarily based on the associated task, processing time and computational cost of each server that decide the distribution of information in order to acquire the processing time for the tasks. It will lead to uneven server distribution. It can be classified into several sub groups [18-22].

Data traversing through the computer network for choosing the shortest routes is highly beneficial when the data delivery is more important than the overall network lifetime [23]. For minimizing network traffic, uniform load balancing is used to maximize network throughput and reduce queue length [24]. For segregating network traffic, the sub-network load balancing approach, which is constructed on the greedy growing method is developed with strong emphasis on utilizing the on-board batteries of the sensor nodes effectively. It is the best approach for transferring traffic on neighboring nodes [25]. A biased energy and haphazard movement based technique [26] spreads the network traffic among multiple routes in the resource constraint environment. A tree-based load balancing method is introduced, where a leaf of a tree or child node finds one of its parent nodes [27]. The different parameters, i.e. packet delivery, packet loss, and energy consumption are considered when transferring packets from start to end node. A stochastic distributed approach based on node settlement and load optimization approach is proposed in [28] to balance the energy consumption through the network.

II. Load Balancing Performance Measures

The performance of load balancing is measured by the following parameters:

- Throughput: The number of packets transferred per unit time span. The performance of computer network is enhanced if throughput is maximized.

$$\text{Throughput} = \frac{\text{Data traffic}}{\text{Time}} \quad (1)$$

- Fault Tolerance: Recovery from state when a process is abnormally disconnected or in failure mode.
- Migration Time: it is the period required to transfer packets from a source node to the destination node.
- Traffic Reduce Rate: The traffic rate is calculated as the total reduction in traffic divided by the total data packets transferred from that node multiplied by 100.

$$\text{Traffic Reduce Rate} = \frac{\text{Traffic\_reduce}}{\text{Total\_traffic}} \text{ x } 100 \quad (2)$$

- Response Time: The total time-to-live when transferring a packet from one node to another. It is minimized with increase in throughput.

$$\text{Response time} = \frac{1}{\text{Throughput}} \quad (3)$$

- Queue Length: The unprocessed tasks are accumulated into a waiting stage. It is processed for decreasing accumulated task to improve response time and maximizing throughput.

$$Q(m) = Q(m) + L\,(m) - \frac{T\mu}{fs} \quad (4)$$

where $L(m)$ is the predictable amount of requests allocated to server m in T interval, $T\mu/fs$ is the mean of the new arrival request in the time interval that the server is continuously in busy state.

III. Glowworm Swarm Optimization Algorithm

A glowworm has the ability to spread randomly in a search space. It is attracted to the glowworm with the highest glow intensity in its neighborhood [29]. Each glowworm broadcasts its glow intensity and is attracted to its neighbors on the basis of a fitness function. The glowworm moves towards the neighboring glowworm having the most glow intensity. The fitness is proportional to the function being optimized.

IV. Steps of the Glowworm Swarm Optimization Algorithm

The steps of GSO algorithm are depicted in Figure 1. On initialization, all glowworms contain the same amount of Luciferin l_o. Every cycle of the algorithm passes through three stages: Luciferin modify stage, association stage, and community array update stage. The Luciferin modify stage denotes the function assessment at the glowworm position. Every glowworm updates its level from its prior Luciferin level:

$$l_i\,(t+1) = (1-\rho)\; l_i(t) + \gamma\, J\,(x_i\,(t+1)) \quad (5)$$

where $l_i(t)$ denotes the Luciferin level of the i^{th} glowworm at time t, ρ represents the Luciferin degeneration variable ($0 < \rho < 1$), γ is the Luciferin enhancement variable, and $J(x_i\,(t))$ denotes the cost of the objective function or the prior level of Luciferin of the i^{th} glowworm at time t.

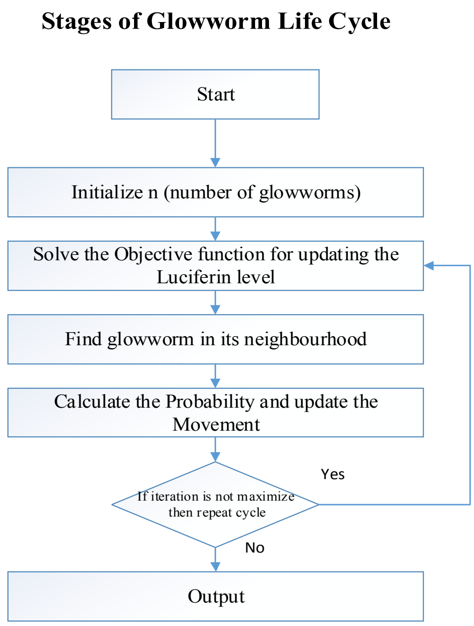

Fig. 1. Stages of GSO.

In the Movement period, each glowworm determines the probability to move in the direction where an adjacent glowworm has Luciferin quantity greater than its own, so the glowworm would move towards where the intensity of the radiance is the highest. The probability of moving towards a glowworm at the adjacent j^{th} position is given by:

$$pij\;(t) \;= \frac{lj(t) - li(t)}{\sum k\in Ni(t)\;\; lk(t) - li(t)} \quad (6)$$

where $j \in N_i(t)$, $N_i(t) = \{j\!: d_{ij}(t) < r^i{}_d(t)\;; li(t) < lj(t)\,(t)\}$ at neighborhood of glowworm i in time span t, $d_{ij}(t)$ the Euclidean distance between glowworms i and j in time span t, and $r^i{}_d(t)$ indicates the adjacent array related to glowworm i in time span t. So, the glowworm i selects a glowworm $j \;\in N_i(t)$ with $P_{ij}(t)$. The prototype for the glowworm arrangement [30] is specified as:

$$yi(t\;+\;1) = yi(t) + \left(\frac{yi(t) - yj(t)}{||yi(t) - yj(t)||}\right) \quad (7)$$

where $y_i(t) \in R^m$ is the position of glowworm i in time span t in the m-dimensional real space R^m, $\|.\|$ represents the Euclidean norm operator and s > 0) is the step size.

In the neighborhood range update stage, each glowworm i has a neighborhood in its radial range $r^i{}_d$ ($0 < r^i{}_d <= r_s$), where

r_s is the glowworm sensor range. Each glowworm's neighborhood range is updated as:

$$r^i_d(t)\ (t+1) = \min \{\ rs, \max \{\ 0, r^i_d(t)\ + \beta\ (n_t - |N_i(t)|)\ \}\quad (8)$$

where β is a constant and n_t is the number of neighbors.

V. Multi-modal Functions

The multi-modal function is a branch of optimization concerned with finding good solutions. It is defined as a function having multiple local optima. A local optimum is a solution which is better than all its neighboring solutions. These functions also have multiple global minima scattered throughout the search space. It is also defined as a statistical distribution of values with multiple peaks.

The GSO algorithm captures the multiple optima of the multi-modal optimization function [31]. In the context of load balancing on computer networks, the traffic is shifted towards either balanced or under loaded nodes. These problems are introduced for continuous data of equal (balanced nodes) or unequal (under loaded and over loaded nodes) peaks. The cases considered for multi-modal function during peak times of traffic are: unequal peaks, equal peaks, and peaks of concentric circles located at irregular intervals. It is much more helpful to optimize data in computer networks when the response time is increasing with increase in queue length. Using the GSO algorithm, the network traffic can be shifted towards the path to the fittest neighborhood with the lighter traffic [32]. The problem is to find the local optima having equal and unequal values. The multi-modal function contains maximum and minimum number of traffic during the passage of data [33-35].

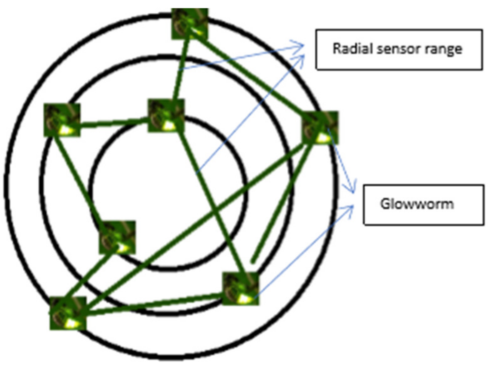

Fig. 2. GSO Circle function: each glowworm is attracted to its neighboring glowworm.

The multi-modal functions used in GSO regions are: (i) Rastrigin's function, (ii) Circle function, (iii) Equal-peaks-A function, (iv) Equal-Peaks-B-function, and (v) Equal-Peaks-C-function [36]. The Circle function is the most appropriate in application to load balancing. The Circle function contains multiple concentric circles as the region of local maxima is depicted in Figure 2. The concentric circles share the same center point, therefore circular lines of local peaks present an infinite-peak case unlike other multi-modal functions such as Rastrigin's function, two dimension exponential function, and Gaussian distribution functions where the peak is located on a single point. The circle function is given as:

$$J_4(y_1, y_2) = (y_1{}^2 + y_2{}^2)^{0.25}\ ((\sin^2(50(y_1{}^2 + y_2{}^2)^{0.1}))+1.0\quad (8)$$

VI. Experiment

This section presents the determination of load burden on computer network and the shifting of load towards neighboring nodes having lesser burden. It studies the performance of GSO algorithm and compares it with the performance of other nature-inspired algorithms such as PSO and CK. In the designed setup, all devices are connected from any location in a network and transfer data packets over it. The details of all the devices, such as device identification number, device name, and device geographical location is saved on the internet data repository [37]. The test model is depicted in Figure 3.

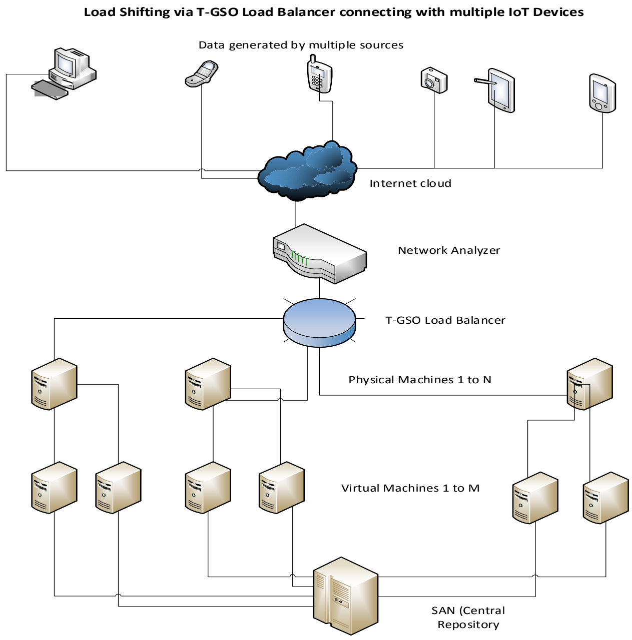

Fig. 3. Network topology of load balancing using T-GSO segregated on three layers.

Users' requests are coming from a number of IoT devices. The requests are segregated onto virtual machines via a load balancer (T-GSO) and are finally saved in a central repository. The load is mitigating queue tasks to be fetched on priority basis with increased throughput, thereby minimizing response and latency times. Data pass through the GSO to find the Luciferin level and compute the fitness function given in (4). The glowworms find the fittest neighbor on the basis of probability, so they move data packets towards fittest neighbor, having highest value of fitness function. It selects the network packet and transfers it to the least congested nodes. The parameters of GSO, CK, and PSO set for the experiment are depicted in Tables I-III. These parameters can be further adjusted for tuning the performance of the system. Experiments were conducted for network configurations of 50, 100, and 200 nodes. At the initial stage, data are extracted from a PRTG network monitoring tool configured on computer network for visualizing network traffic. The data are uploaded into the data repository. Then, the data are subjected to GSO, PSO, and CK algorithms and the performance is noted.

TABLE I. GSO VARIABLES

	Parameter	Fixed value
ρ	Rho	0.4
γ	Ghama	0.6
s	Sensor	0.03
n_t	Neighborhood nodes	5.00
l_o	Luciferin	5.00

TABLE II. CK PARAMETERS

	Parameter	Fixed value
npar	Quantity of enhancement	100
varLo	Minimum value of band	-5.0
varHi	Maximum value of band	5.0
numCuckooS	Initial quantity	5.0
numNewCuckooS	New quantity	0.0
minNumberOfEggs	Least quantity of eggs	2.0
MaxNumberOfEggs	Extreme quantity of eggs	4
maxIter	Highest repetitions	100
knnClusterNum	Quantity of clusters	1
motionCoeff	Lambda variable, default=2	9

TABLE III. PSO PARAMETERS

Parameter	Fixed value
Problem dimensions	2
Number of Particles	5
Inertia weight	0.729
Cognitive weight	1.49445
Social weight	1.49445

VII. SIMULATION RESULTS AND DISCUSSION

The performance results of GSO, PSO, and CK algorithms in terms of throughput and queue length are presented in this section. Figures 4-7 show the results of GSO. The result of the comparison of GSO with PSO and CK algorithms is depicted in Figure 8. The behavior of Fitness Function vs. the Number of Iterations is given in Figure 4. The less loaded node would have a high value of Luciferin level, therefore more data packets are delivered through this node. The load balancer routes data traffic towards neighboring nodes having higher Luciferin than its level. The Euclidean distance should be less than the radial sensor ranges of neighboring glowworm, e.g. from Table I, the radial sensor range is 0.22Mb/s therefore the highest probability of data packets shifted towards adjacent node (Node number 1). The number of iterations was computed for 358 time slots over 50, 100, and 200 nodes, and Luciferin level has stabilized at similar levels for all nodes (Figure 4) indicating load balanced over the nodes.

The comparison of traffic data packets and optimum load (GSO-X, GSO-Y) is depicted for 50, 100 and 200 nodes. The trend shows that the node capacity is maximized which would be segregating a large number of data packets on the virtual machine, therefore the load is mitigated on a neighboring node. The incoming and outgoing data packets are 20Mb/s and 13Mb/s. The load is optimized by T-GSO load balancer executing the iterations, thereby GSO-X and GSO-Y are 16Mb/s and 15Mb/s in node 1. The proposed T-GSO scheme shows improved network throughput during peak traffic for networks of 50, 100, and 200 nodes.

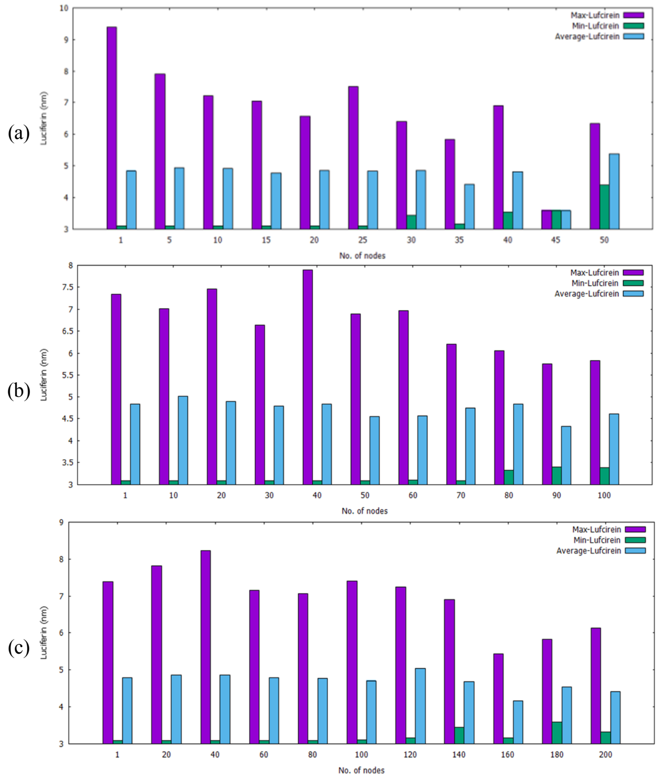

Fig. 4. Fitness Function vs. Iterations under (a) 50, (b)100, (c) 200 nodes

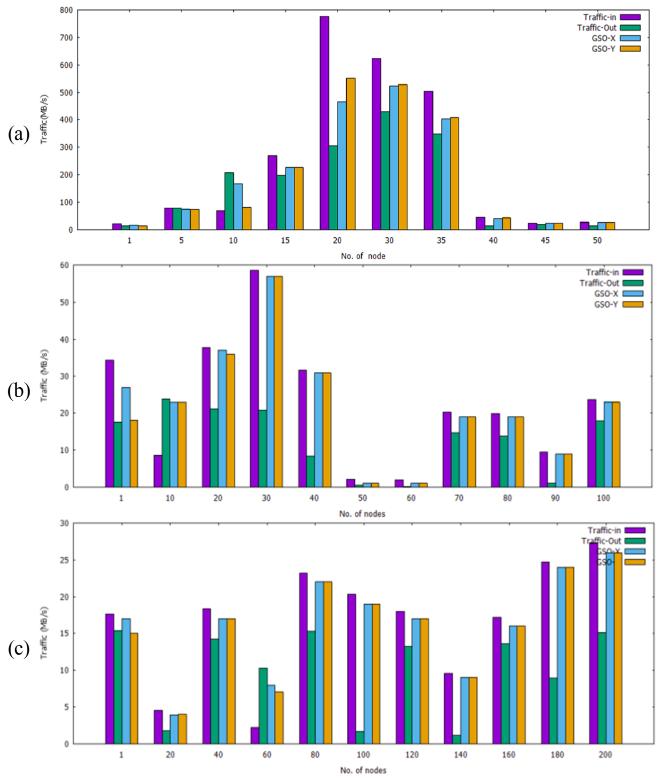

Fig. 5. Data traffic vs. optimum load under (a) 50, (b) 100, (c) 200 nodes.

Considering Traffic-In and Traffic-Out, Figure 5 shows the traffic at nodes indicated at the start of the GSO algorithm. GSO-X and GSO-Y give the resulting position on completion of iterations of GSO algorithm. It can be seen that the input traffic at nodes was reduced and the output traffic was increased in the completion of the GSO algorithm. Furthermore, the number of nodes increased, the resultant load GSO-X, GSO-Y appear more balanced.

Figure 6 shows queue length vs. iterations. The network performance depends upon important indicators of load balancing, such as the queued tasks that are to be executed on each node. If the queue processes are increased, the network throughput reduces, which may cause request timeout and time delay. Queue length is recorded on each node with respect to time. The queue length is reduced with the help of GSO algorithm for diverting traffic. The highest value of Luciferin level is corresponding with the minimum value of queue length on an adjacent node. The trend shows that the queue task is released when the number of nodes is increased, therefore network load was mitigated on adjacent nodes having maximum value of fitness function.

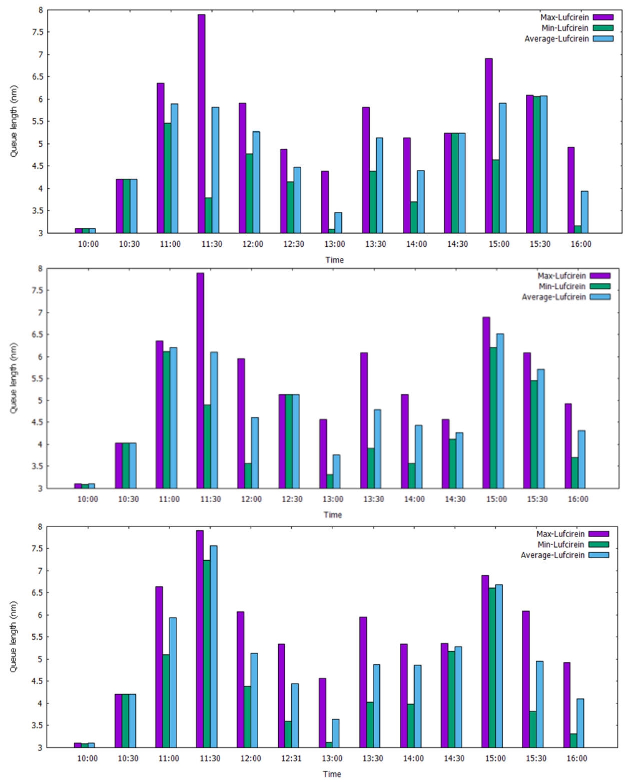

Fig. 6. Queue length vs. load for: (a) 50, (b) 100, (c) 200 nodes.

In GSO algorithm, each glowworm finds the optimal path where the glow intensity is maximum. The Luciferin level of that node is indirectly proportional to the traffic accumulated on the node. The highest Luciferin level on an adjacent node indicates the lowest number of data packets is delivered on every node. In Figure 7, at 10pm the Luciferin level is 3.1nm and Traffic-In and Traffic-Out are 1.94 and 0.2, which indicates inverse relation between fitness value and data bandwidth. The trend of the plot in Figure 7 shows that when the number of nodes increases, the Luciferin level is maximized, therefore the load is mitigated on that node.

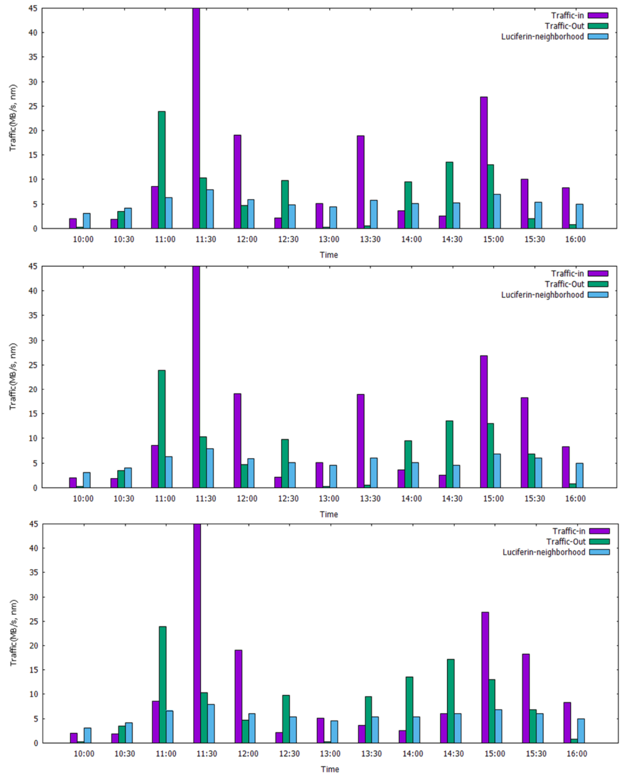

Fig. 7. Luciferin and data packets accumulated under: (a) 50, (b) 100, (c) 200 nodes.

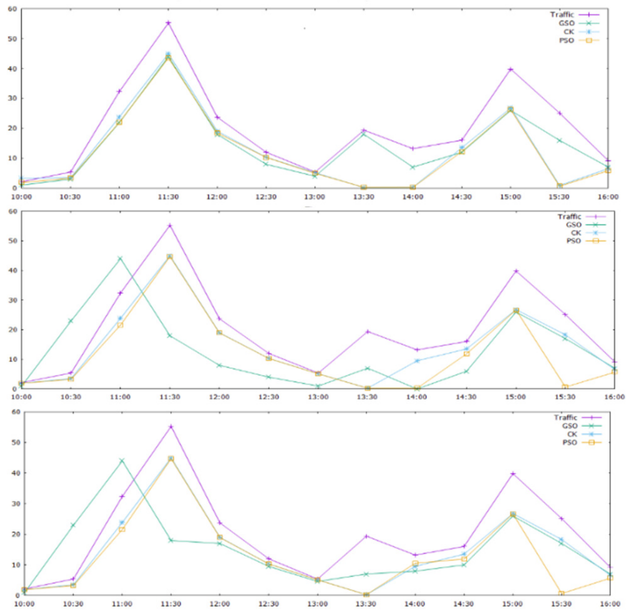

Fig. 8. Comparison of Luciferin and data packets accumulated under: (a) 50, (b) 100, (c) 200 nodes.

The performance comparison of GSO with PSO and CK algorithms for network load balancing is given in Figure 8. The results demonstrate the achievement of reducing node traffic in peak network usage periods. The GSO gave 71.79%, 61.17%, and 57.17% improved performance on 50 nodes, 74.14%, 70.95%, and 62.38% on 100 nodes, and 84.15%, 78.14%, and 60.42% on 200 nodes respectively, showing its superiority.

GSO picks the fit neighborhood where the traffic is lighter and forwards traffic to under loaded and normal loaded nodes. The GSO algorithm worked as a distributed mechanism by decreasing response time and overhead in addition by increasing throughput to improve the performance.

VIII. CONCLUSION

The explosive growth of data on the internet from the increasing numbers of IoT and M2M devices has the impact of diminishing the network performance, efficiency, and data delivery. The results from this work show that nature-inspired meta-heuristic approaches can be applied to the network load balancing problem and model network traffic. GSO is a more recent algorithm than PSO and CK. From the study of performance comparison of these three nature-inspired algorithms on the basis of node queue length, throughput, and response turnaround time, GSO showed better performance over PSO and CK. The comparison of GSO with PSO showed improvement in traffic reduce rate by 13.87%, 11.75%, and 23.72% for 50, 100, and 200 nodes respectively. Likewise, in comparison with CK, GSO improved the traffic reduce rate by 10.61%, 3.19%, and 6.00% respectively. Furthermore, it appears that performance improvement is scaling with increase in nodes. The GSO algorithm therefore seems more suited to mitigating the load balancing problem. For future work the following are proposed:

- Improvement of the GSO algorithm through study of different multi-modal functions depending on the basis of network topology.
- The multi-modal function parameters would be studied to find setting for optimal performance in different traffic scenarios.
- A hybrid model of GSO with another optimization algorithm such as the Genetic Algorithm.

ACKNOWLEDGEMENT

The funding agency of the current project is NED University of Engineering Technology and Karachi, Pakistan by Higher Education Commission of Pakistan.

REFERENCES

[1] H. Ben Alla, S. Ben Alla, A. Touhafi, and A. Ezzati, "A novel task scheduling approach based on dynamic queues and hybrid meta-heuristic algorithms for cloud computing environment," *Cluster Computing*, vol. 21, no. 4, pp. 1797–1820, Dec. 2018, https://doi.org/10.1007/s10586-018-2811-x.

[2] M. M. Abed and M. Younis, "Developing Load Balancing for IoT - Cloud Computing Based on Advanced Firefly and Weighted Round Robin Algorithms," *Baghdad Science Journal*, vol. 16, pp. 130–139, Mar. 2019, https://doi.org/10.21123/bsj.2019.16.1.0130.

[3] *Cisco Visual Networking Index: Forecast and Trends, 2017–2022*. CISCO, 2019.

[4] *Cisco Annual Internet Report (2018–2023) White Paper*. CISCO, 2020.

[5] D. A. Abduljabbar, S. Z. M. Hashim, and R. Sallehuddin, "Nature-inspired optimization algorithms for community detection in complex networks: a review and future trends," *Telecommunication Systems*, vol. 74, no. 2, pp. 225–252, Jun. 2020, https://doi.org/10.1007/s11235-019-00636-x.

[6] G. Beni and J. Wang, "Swarm Intelligence in Cellular Robotic Systems," in *Robots and Biological Systems: Towards a New Bionics?*, Toscana, Italy, Jun. 1989, pp. 703–712, https://doi.org/10.1007/978-3-642-58069-7_38.

[7] H. Yu, S. Wang, and Y. Bao, "Application of Ant Colony Algorithms in Path Planning of Ocean Survey," *Journal of Coastal Research*, vol. 94, no. SI, pp. 121–124, Jun. 2019, https://doi.org/10.2112/SI94-023.1.

[8] F. Ezazi, M. H. Mallah, J. Karimi Sabet, A. Norouzi, and A. Mahmoudian, "Investigation on the net cascade using Ant Colony optimization algorithm," *Progress in Nuclear Energy*, vol. 119, Jan. 2020, Art. no. 103169, https://doi.org/10.1016/j.pnucene.2019.103169.

[9] Md. R. Islam, C. M. K. Saifullah, and Md. R. Mahmud, "Chemical reaction optimization: survey on variants," *Evolutionary Intelligence*, vol. 12, no. 3, pp. 395–420, Sep. 2019, https://doi.org/10.1007/s12065-019-00246-1.

[10] M. A. Summakieh, C. K. Tan, A. A. El-Saleh, and T. C. Chuah, "Improved load balancing for LTE-A heterogeneous networks using particle swarm optimization," *International Journal of Technology*, vol. 10, no. 7, pp. 1407–1415, Nov. 2019, https://doi.org/10.14716/ijtech.v10i7.3253.

[11] T. M. Kumar and N. A. Singh, "Environmental Economic Dispatch with the use of Particle Swarm Optimization Technique based on Space Reduction Strategy," *Engineering, Technology & Applied Science Research*, vol. 9, no. 5, pp. 4605–4611, Oct. 2019, https://doi.org/10.48084/etasr.2969.

[12] S. Aslan, H. Badem, and D. Karaboga, "Improved quick artificial bee colony (iqABC) algorithm for global optimization," *Soft Computing*, vol. 23, no. 24, pp. 13161–13182, Dec. 2019, https://doi.org/10.1007/s00500-019-03858-y.

[13] S. Kumari, P. K. Mishra, and V. Anand, "Integrated Load Balancing and Void Healing Routing with Cuckoo Search Optimization Scheme for Underwater Wireless Sensor Networks," *Wireless Personal Communications*, vol. 111, no. 3, pp. 1787–1803, Apr. 2020, https://doi.org/10.1007/s11277-019-06957-z.

[14] F. Safaei, H. Yeganloo, and R. Akbar, "Robustness on topology reconfiguration of complex networks: An entropic approach," *Mathematics and Computers in Simulation*, vol. 170, pp. 379–409, Apr. 2020, https://doi.org/10.1016/j.matcom.2019.11.013.

[15] G. R. Asha and Gowrishankar, "A hybrid approach for cost effective routing for WSNs using PSO and GSO algorithms," in *International Conference on Big Data, IoT and Data Science*, Pune, India, Dec. 2017, pp. 1–7, https://doi.org/10.1109/BID.2017.8336564.

[16] D. Rewadkar and D. Doye, "FGWSO-TAR: Fractional glowworm swarm optimization for traffic aware routing in urban VANET," *International Journal of Communication Systems*, vol. 31, no. 1, 2018, Art. no. e3430, https://doi.org/10.1002/dac.3430.

[17] P. J and N. Venkataraman, "Threshold Based Multi-Objective Memetic Optimized Round Robin Scheduling for Resource Efficient Load Balancing in Cloud," *Mobile Networks and Applications*, vol. 24, no. 4, pp. 1214–1225, Aug. 2019, https://doi.org/10.1007/s11036-019-01259-x.

[18] K. Ono, T. Ono, K. Akimoto, S. Kameda, and N. Suematsu, "Traffic Navigation amp; Relay: System-Wide Load Balancing Method for Heterogeneous Network Using Route Direction and Packets Relay," in *URSI Asia-Pacific Radio Science Conference (AP-RASC)*, New Delhi, India, Mar. 2019, pp. 1–4, https://doi.org/10.23919/URSIAP-RASC.2019.8738647.

[19] S. Tahzeeb and S. Hasan, "A Neural Network-Based Multi-Label Classifier for Protein Function Prediction," *Engineering, Technology &*

Applied Science Research, vol. 12, no. 1, pp. 7974–7981, Feb. 2022, https://doi.org/10.48084/etasr.4597.

[20] Y. Bella and F. Z. Kebbab, "Application of Fminsearch Optimization to Minimize Total Maintenance Cost with the Aim of Reducing Environmental Degradation," *Engineering, Technology & Applied Science Research*, vol. 12, no. 3, pp. 8548–8554, Jun. 2022, https://doi.org/10.48084/etasr.4887.

[21] W. Lin, G. Peng, X. Bian, S. Xu, V. Chang, and Y. Li, "Scheduling Algorithms for Heterogeneous Cloud Environment: Main Resource Load Balancing Algorithm and Time Balancing Algorithm," *Journal of Grid Computing*, vol. 17, no. 4, pp. 699–726, Dec. 2019, https://doi.org/10.1007/s10723-019-09499-7.

[22] N. Yu, Z. Song, H. Du, H. Huang, and X. Jia, "Dynamic Resource Provisioning for Energy Efficient Cloud Radio Access Networks," *IEEE Transactions on Cloud Computing*, vol. 7, no. 4, pp. 964–974, Jul. 2019, https://doi.org/10.1109/TCC.2017.2715812.

[23] A. Sari and E. Caglar, "Load Balancing Algorithms and Protocols to Enhance Quality of Service and Performance in Data of WSN," in *Security and Resilience in Intelligent Data-Centric Systems and Communication Networks*, M. Ficco and F. Palmieri, Eds. Cambridge, MA, United States: Academic Press, 2018, pp. 143–178.

[24] R. Khan, "An efficient load balancing and performance optimization scheme for constraint oriented networks," *Simulation Modelling Practice and Theory*, vol. 96, Nov. 2019, Art. no. 101930, https://doi.org/10.1016/j.simpat.2019.101930.

[25] R. Sharma, N. Mittal, and B. S. Sohi, "Flower pollination algorithm-based energy-efficient stable clustering approach for WSNs," *International Journal of Communication Systems*, vol. 33, no. 7, 2020, Art. no. e4337, https://doi.org/10.1002/dac.4337.

[26] Y. Zhang and Y. Wang, "A novel energy-aware bio-inspired clustering scheme for IoT communication," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 10, pp. 4239–4248, Oct. 2020, https://doi.org/10.1007/s12652-020-01704-w.

[27] P. M. Daflapurkar, M. Gandhi, and B. Patil, "Tree based distributed clustering routing scheme for energy efficiency in wireless sensor networks," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, Chennai, India, Sep. 2017, pp. 2450–2456, https://doi.org/10.1109/ICPCSI.2017.8392157.

[28] Y. Liao, H. Qi, and W. Li, "Load-Balanced Clustering Algorithm With Distributed Self-Organization for Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1498–1506, Feb. 2013, https://doi.org/10.1109/JSEN.2012.2227704.

[29] A. Khan, F. Aftab, and Z. Zhang, "Self-organization based clustering scheme for FANETs using Glowworm Swarm Optimization," *Physical Communication*, vol. 36, Oct. 2019, Art. no. 100769, https://doi.org/10.1016/j.phycom.2019.100769.

[30] Y. Xiuwu, L. Qin, L. Yong, H. Mufang, Z. Ke, and X. Renrong, "Uneven clustering routing algorithm based on glowworm swarm optimization," *Ad Hoc Networks*, vol. 93, Oct. 2019, Art. no. 101923, https://doi.org/10.1016/j.adhoc.2019.101923.

[31] S. R. Salkuti and S.-C. Kim, "Congestion Management Using Multi-Objective Glowworm Swarm Optimization Algorithm," *Journal of Electrical Engineering & Technology*, vol. 14, no. 4, pp. 1565–1575, Jul. 2019, https://doi.org/10.1007/s42835-019-00206-w.

[32] A. N. Shinde, S. L. Lalbalwar, and A. B. Nandgaonkar, "Modified meta-heuristic-oriented compressed sensing reconstruction algorithm for bio-signals," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 17, no. 5, Sep. 2019, Art. no. 1950031, https://doi.org/10.1142/S0219691319500310.

[33] S. Barak, V. J. Kompella, K. N. Kaipa, and D. Ghose, "Push-pull glowworm swarm optimization algorithm for multimodal functions," in *Swarm Intelligence -Volume 2: Innovation, new algorithms and methods*, IET, 2018, pp. 185–219, https://doi.org/10.1049/PBCE119G_ch7.

[34] G. Miryala and S. A. Ludwig, "Comparing Spark with MapReduce: Glowworm Swarm Optimization Applied to Multimodal Functions," *International Journal of Swarm Intelligence Research*, vol. 9, no. 3, pp. 1–22, Jul. 2018, https://doi.org/10.4018/IJSIR.2018070101.

[35] P. Qiong, Y. Liao, P. Hao, X. He, and C. Hui, "A Self-Adaptive Step Glowworm Swarm Optimization Approach," *International Journal of Computational Intelligence and Applications*, vol. 18, no. 1, Mar. 2019, Art. no. 1950004, https://doi.org/10.1142/S1469026819500044.

[36] R. Isimeto, C. Yinka-Banjo, C. O. Uwadia, and D. C. Alienyi, "An enhanced clustering analysis based on glowworm swarm optimization," in *4th International Conference on Soft Computing & Machine Intelligence*, Mauritius, Nov. 2017, pp. 42–49, https://doi.org/10.1109/ISCMI.2017.8279595.

[37] Y. Verma, "Secure system simulation - Internet of Things," Ph.D. dissertation, California State University, California, USA, 2016.

AUTHORS PROFILE

Talha Akhtar received his B.S. and M.S. degrees in Computer Science & IT from NED University in 2002 and 2008 respectively. He is studying as a Ph.D. scholar at NED University of Engineering and Technology. Mr. Akhtar has 16 year industry experience and 5 years of teaching experience. He has successful completed many projects in the banking sector. The focus of his research lies in the area of swarm intelligence, graph theory and nature inspired algorithms.

Najmi Ghani Haider received his PhD. degree from Brunel University, U.K, in 1989. He has 24 years of experience in the industry and 14 years in academia. He has served as a Professor in the Department of Computer Science and Information Technology, NED University of Engineering and Technology, Karachi, Pakistan. His research interests include network and information security and artificial intelligence.

Shariq Mahmood Khan received his B.S. and M.S. degrees in Computer Science from NED University of Engineering and Technology, Pakistan, in 2005 and 2007 respectively. He earned his Ph.D. degree from the Brunel University London, UK in 2015. He is currently serving as a Professor of Computer Science at NED University. He has authored or coauthored multiple research articles in international journals and conferences. He is a regular reviewer for a number of international journals and serves as the TPC member and reviewer for various international conferences. He also served as the founding conference secretary of the International Conference on Innovations in Computer Science & Software Engineering (ICONICS). The focus of his research lies in the areas of routing protocol design in mobile and vehicular ad-hoc networks, IoT security, and wireless sensor networks.