

Grey wolf optimizer based fuzzy-PI active queue management design for network congestion avoidance

Sana Sabah Sabry¹, Nada Mahdi Kaattan²

¹College of Engineering, University of Information Technology and Communications, Iraq

²College of Business Informatics, University of Information Technology and Communications, Iraq

Article Info

Article history:

Received Jun 23, 2019

Revised Sep 24, 2019

Accepted Oct 15, 2019

Keywords:

AQM

Congestion control

Fuzzy-PI

Grey wolf

GWO

ABSTRACT

Congestion is one of the most important issues in communication networks which has attracted much research attention. To ensure a stable TCP network, we can use active queue management (AQM) for early congestion detection and router queue length regulation. In this study, it was proposed to use the Grey Wolf Optimizer (GWO) algorithm in designing a fuzzy proportional integral (fuzzy-PI) controller as a novel AQM for internet routers congestion control and for achieving a low steady-state error and fast response. The suggested Fuzzy logic-based network traffic control strategy permit us to deploy linguistic knowledge for depicting the dynamics of probability marking functions and ensures a more accurate use of multiple inputs to depict the the network's state. The possibility of incorporating human knowledge into such a control strategy using Fuzzy logic control methodology was demonstrated. The postulated controller was compared to proportion integral (PI) through several MATLAB simulation scenarios. The results indicated the stability of the postulated controller and its ability to attain a faster response in a dynamic network with varying network load and target queue length.

Copyright © 2020 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Sana Sabah Sabry,

College of Engineering,

University of Information Technology and Communications,

Baghdad, Iraq.

Email: Sana.sabah@uoitc.edu.iq

1. INTRODUCTION

The increase in the development of the internet makes network congestion more frequent, causing long delays in packets delivery, packets loss, and overflow in the buffer of the intermediate routers. Transmission Control Protocol (TCP) is the most commonly used protocol in managing internet traffic; the TCP senders reduce their sending rate if there is a packet loss detection. This detection depends on the acknowledged signal sent from the receivers as they successfully receive a packet [1]. A larger quantity of data might be lost to the process of router packet drop detection from its source. Due to packet drop, TCP also suffers from the problem global synchronization of a source [2].

However, active queue management (AQM) approaches have been initiated to overcome these challenges; such approaches are implemented in the router to avert congestion by generating packets dropping feedbacks through which congestion is reported to the source for necessary actions to be taken.

The targets of using AQM are to determine the incoming network traffic rate and to manage the queue length and use network resources with a lower rate of packet loss. In recent times, several techniques have been proposed for different AQM policies [3-7]. Random Early Detection (RED) [8] is considered the oldest common AQM technique which has been postulated to reduce link congestion through random packets dropping. It has been shown that the RED algorithm is unstable and more sensitive to the parameters of the network [9]. In the RED algorithm, the relationship between the system controller and the dynamic is poorly

defined. Therefore, to invent an effective controller for a stable and robust system, the TCP/AQM needs to understand the system’s dynamics. Several dynamic models have been postulated so far and one of the most widely used models is TCP’s nonlinear fluid-flow model [10].

In this model, the basics of control theory are used to study and develop new AQM controllers; these basics include proportional integral (PI) [11], proportional derivative (PD) [11], and LQ-Servo [12-13]. However, these AQM controllers have the setback of not being able to conserve their implementations. To overcome this difficulty, we can apply congestion control algorithm built on fuzzy logic control can be applied [14]. The fuzzy control tuning systems and systematic designs can ensure congestion avoidance in TCP routers and guarantee the desired performance specification [15-16]. The performance specifications can be assessed through many indices for examples: rise time, settling time, overshoot, and the objective functions in several optimization problems.

Many intelligent optimization algorithms are available for optimal tuning of the fuzzy controller, for example: Genetic Algorithms (GA) [17], and Particle Swarm Optimizer (PSO) [18]. The Grey Wolf Optimizer (GWO) [19] algorithm was created based on the observation of hunting habits and social hierarchy of Grey wolves [20]. Based on the inspiration from the social hierarchy of wolves, the search population in the GWO is grouped into four, which are tagged alpha, delta, omega, and beta depending on the wolves’ physical attributes. The search procedure is fabricated to imitate grey wolves’ hunting behavior which involves three basic steps - hunting, encompassing, and striking of the prey. The first two are used to explore while the latter is used to envelope the exploitation.

This finding is used in the proposed GWO algorithm for an optimal tuning of fuzzy controllers (like an AQM for TCP network) to stabilize queue lengths efficiently with speedy settling time and high link utilization. The second contribution is based on the proposed controllers’ performance analysis against network parameters, including various TCP connections and different target queue lengths.

This manuscript is sectioned as follows: The TCP model was presented in Section 2, while the GWO-based fuzzy controller design for AQM was presented in Section 3. The obtained results from the simulation studies on the efficiency of the postulated control system were presented in Section 4, while conclusions from the study were in Section 5.

2. DESCRIPTION OF THE SYSTEM

For network flow dynamics, the TCP behavior was modelled based on the previous study by utilizing a stochastic differential equation analysis [11]. The model deployed one router and N uniform TCP-controlled source networks as illustrated in Figure 1, where W, C, q, R, N, Tp, and p are the TCP window size, link capacity of a single bottleneck link, router queue length (in packets), round trip time (in second) of TCP connection, number of TCP connections (load factor), propagation delay, and probability of packet drop.

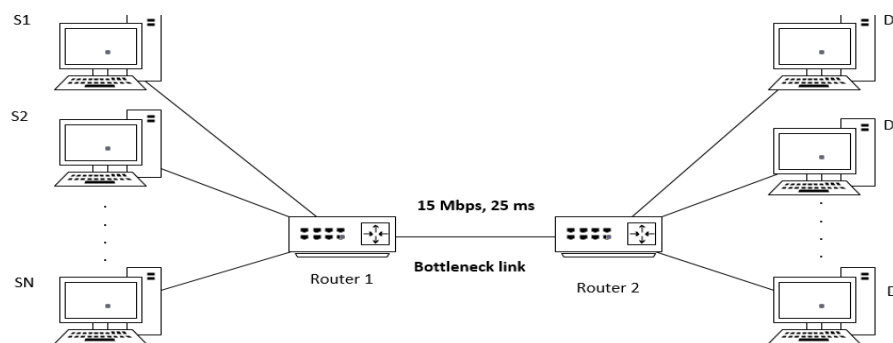


Figure 1. Bottleneck scenario

The linearized network dynamic model is expressed using (1),

$$\begin{aligned}
 \dot{W}(t) &= -\frac{2N}{R^2C}W(t) - \frac{RC^2}{2N^2}P(t - R) \\
 \dot{q}(t) &= \frac{N}{R}\delta W(t) - \frac{1}{R}q(t)
 \end{aligned}
 \tag{1}$$

The round-trip time (R) was determined using (2),

$$R = \frac{q}{c} + T_p \tag{2}$$

Figure 2, shows the linearized AQM control system. More linearization details are referred to a previous study reported by [13]. The transfer function for the linearized AQM model in (2), is obtained thus:

$$F(s) = \frac{q(s)}{p(s)} = \frac{\frac{C^2}{2N}e^{-sR}}{(s + \frac{2N}{R^2C})(s + \frac{1}{R})} \tag{3}$$

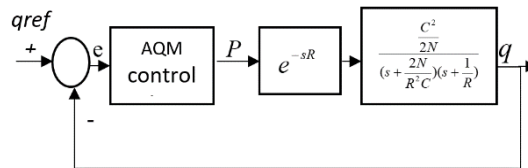


Figure 2. Block diagram for AQM as feedback control system

3. FUZZY-PI BASED GWO CONTROLLER

3.1. Controller Design

The Fuzzy PI controller formation [21] for TCP/AQM was illustrated in Figure 3. This model combines a PI controllers and a fuzzy logic. The input scaling factors were Kp and Ki while the output scaling factor was Ku. The proposed fuzzy-PI controller was designed to regulate TCP router queue by achieving a specific desired queue length to avoid congestion.

The fuzzy controller utilized two inputs, error (e) and the integration of errors (∫e). The fuzzy controller was schemed to run buffer queues for the router, as well as utilize linguistic rules for marking packets in TCP/AQM networks. Figure 3, presents how the fuzzy controller can dynamically determine the mark probability p(t) by utilizing the error and its derivative.

Due to the difficulty of formulating rules based on integral errors, it is possible to relocate the integration so that it follows a fuzzy controller rather than the integral error. In such way, it is the output that is integrated despite the error and its associated changes as shown in Figure 4 [21].

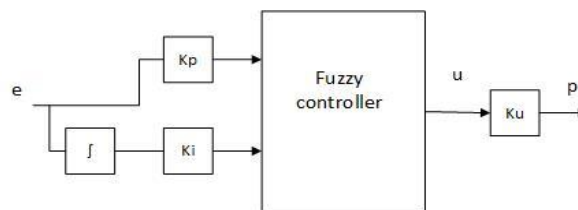


Figure 3. structured block diagram for Fuzzy PI controller

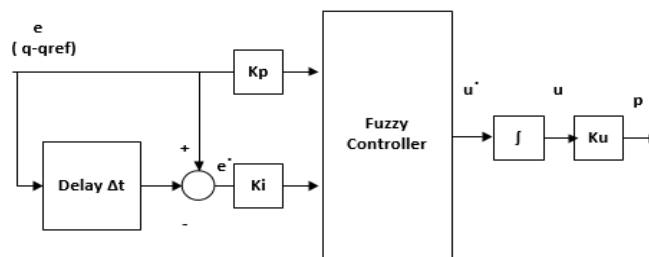


Figure 4. structured blockdiagram for Fuzzy PI controller (modified version)

linguistic values of input and output were presented in Table 1 and Table 2. Triangular membership functions were used for the input parameters (error, error derivative) while mark probability (P_{mark}) was the output, as presented in Figure 5, the membership functions that were considered for this finding are triangular and trapezoidal due to their simplicity and ability to give the best response and the fact that they consist of simple line portions that minimizes computational difficulty [22-23]. The formation of the fuzzy controller was presented in Figure 6.

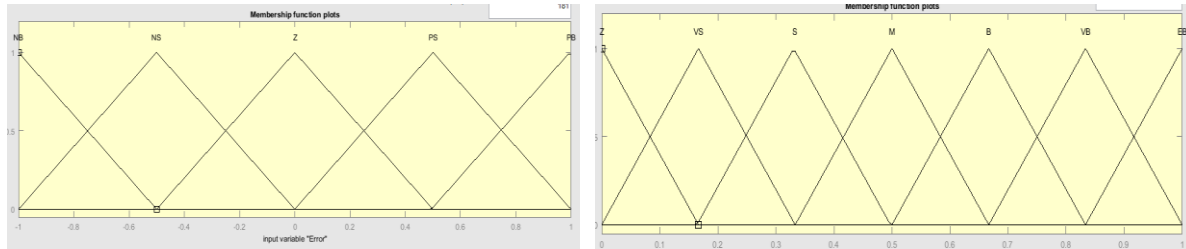


Figure 5. Triangular Membership function for input and output variable

Table 1. Input Linguistic Values

Linguistic values	Meaning in Error	Meaning in change of Error
NB: Negative big	The router queue much above q_{ref}	The router queue is moving away from q_{ref} in the upward direction
NS: Negative small	The router queue slightly above q_{ref}	The router queue is slightly moving away from q_{ref} in the upward direction
Z: Zero	The router queue is very closed q_{ref}	The router queue is very closed q_{ref}
PS: Positive small	The router queue is slightly below q_{ref}	The router queue is slightly moving away from q_{ref} in the downward direction
PB: Positive big	The router queue much below q_{ref}	The router queue is moving away from in the downward direction

Table 2. Output Linguistic Values (P_{mark})

Linguistic values	meaning
Z	Zero output control signal
VS	Very Small output control signal
S	Small output control signal
M	Medium output control signal
B	Big output control signal
VB	Very big output control signal
EB	Extra big output control signal

The controller output was evaluated using the center of gravity approach of defuzzification. The fuzzy-PI controller rule base is comprised of 25 (5 x 5) rules as presented in Table 3.

Table 3. Fuzzy Rule Base

		error derivative				
		\dot{e}				
error e	P_{mark}	NB	NS	Z	PS	PB
	NB	B	B	VB	VB	EB
	NS	S	M	M	B	VB
	Z	Z	Z	VS	S	M
	PS	Z	Z	Z	VS	VS
PB	Z	Z	Z	Z	Z	

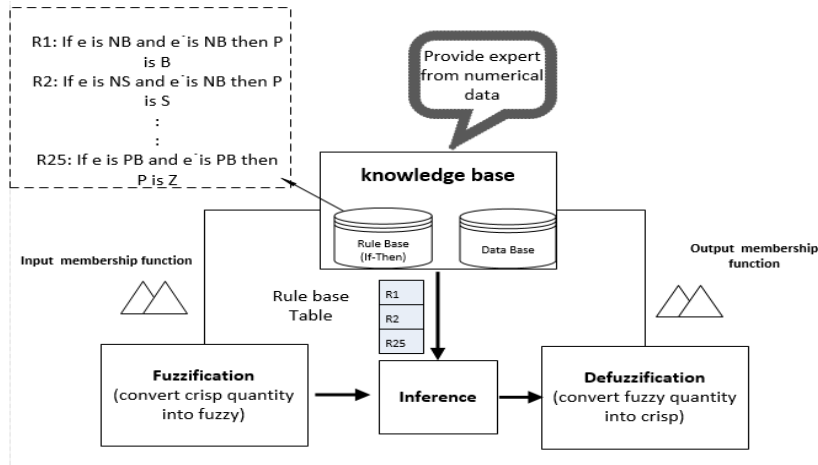


Figure 6. the fuzzy controller structure

3.2. Objective Function

A performance index (Objective function) is used in the quantification of a system performance. It is helpful in the effective designing of a control system with the desired specifications. In this study, integral time absolute error (ITAE) was utilized as the objective function to adequately design the postulated fuzzy PI controller and check its ability to reduce the settling time and peak overshoot. In (4) expressed the ITAE cost function.

$$ITAE = \int_0^{\infty} t|e|dt \tag{4}$$

3.3. Grey Wolf Optimization Algorithm

In this study, the GWO approach was utilized to tune the fuzzy PI controller parameters. Recently, GWO was postulated as a heuristic algorithm that uses the hunting technique of grey wolves and their social hierarchy as inspiration. Compared to other common evolutionary approaches, The GWO has the advantages of easy programming, no need of input variables for the implementation, simplicity, and reduced computational complexity. The hunting cycle in the GWO begins with inconstantly acquired wolves (solutions). while hunting, a wolf examines the prey’s optimal locations using a repetitive process. Alpha (α), omega (ω), delta (δ), and beta (β) are the four categories of the solution. α denotes the optimal solution while β represents the second-best solution and δ the third. [24]

Other solutions are considered as ω (less valuable). The procedure in the GWO approach is completed in four stages, which include hunting, encompassing the prey, striking and searching the prey (exploration and exploitation stages) [20]. These principal stages of the grey wolf searching process are briefly discussed below.

Encompassing prey: The process of encompassing the prey by the wolves is presented as follows:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|, \vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \tag{5}$$

- t : indicates the current iteration.
- \vec{C}, \vec{A} : coefficient vectors.
- X : the position vector of a wolf.
- X_p denotes the position vector of the prey. The vectors A and C are calculated as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \vec{C} = 2 \cdot \vec{r}_2, \tag{6}$$

where r1, r2 are random vectors in the range [0, 1] and vector a is linearly declined during iterations from 2 to 0.

Hunting process: Betas and deltas help alphas in guiding the hunting procedure. There are three excellent points associated with deltas, betas and alphas which results in an optimum point. The upgraded wolf points across the prey are evaluated using (7), (8) and (9),

$$\begin{aligned}\bar{D}_\alpha &= |\bar{C}_1 \cdot \bar{X}_\alpha - \bar{X}| \\ \bar{D}_\beta &= |\bar{C}_2 \cdot \bar{X}_\beta - \bar{X}| \\ \bar{D}_\delta &= |\bar{C}_3 \cdot \bar{X}_\delta - \bar{X}|\end{aligned}\quad (7)$$

$$\begin{aligned}\bar{X}_1 &= \bar{X}_\alpha - \bar{A}_1 \cdot \bar{D}_\alpha \\ \bar{X}_2 &= \bar{X}_\beta - \bar{A}_2 \cdot \bar{D}_\beta \\ \bar{X}_3 &= \bar{X}_\omega - \bar{A}_3 \cdot \bar{D}_\omega\end{aligned}\quad (8)$$

$$\bar{X}(t+1) = \frac{\bar{X}_1 + \bar{X}_2 + \bar{X}_3}{3}\quad (9)$$

Attacking prey (exploitation): In this stage, the exploitation of a search procedure is enabled for the GWO algorithm. A prey is attacked by the grey wolves and then stop searching. In a mathematical sense, this procedure is given by a decrease in \vec{a} which latter reduces changes in \vec{A} . At first, \vec{A} is a randomize data within the interval $[-a, a]$ and “a” is reduced from 2 to 0 with the iterative process. If $|A| < 1$, the wolves proceed in the direction of the prey for striking.

Search for prey (exploration): The exploration search is enabled in the algorithm; Wolves: alpha, delta, and beta are located in the right place in search of the prey. Then, the wolves proceed individually to find and strike the prey. The GWO algorithm integrates the exploration ability as \vec{A} values ranging from of -1 to 1 . The wolves stay far from the prey in searching of a good prey as $|A| > 1$; however, the \vec{C} component takes part in the process of exploration and ranging from 0 to 2 . The prey are assigned certain weight for defining the gaps. The hunting agents are permitted to modernize their positions depending on the locations of α, β, δ and their striking of the prey [25].

A flowchart of the GWO applied to a fuzzy PI was illustrated in Figure 7.

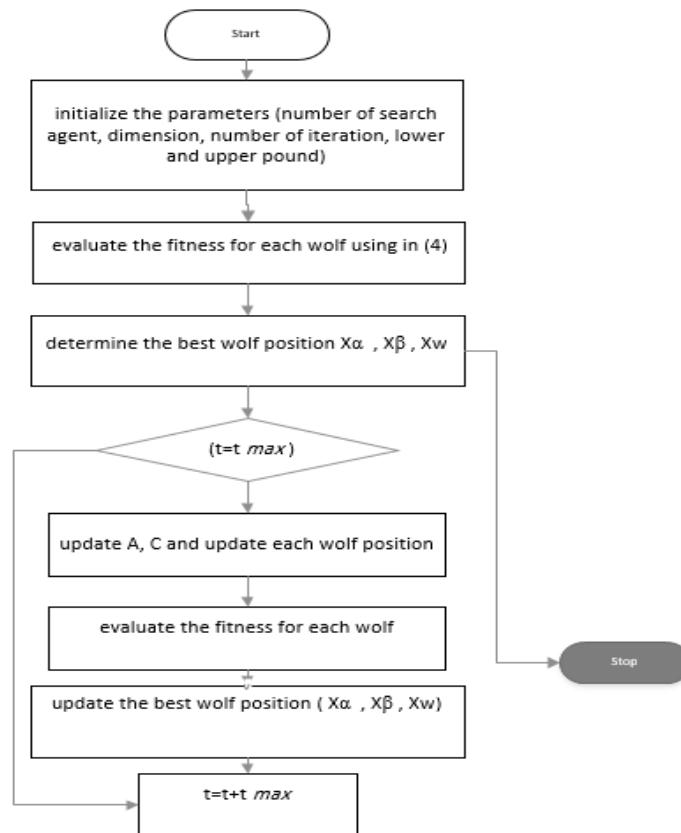


Figure 7. GWO flow chart

4. RESULTS AND ANALYSIS

In this part, the robustness and performance of the postulated fuzzy-PI based GWO for TCP network were examined, evaluated, and compared with PI in various environments. A series of MATLAB simulation scenarios were implemented to determine the efficiency of the postulated controller and examine its capability in avoiding network congestion.

4.1. Network Topology

A single bottleneck dumbbell topology modelled (1) was presented in Figure 1. It comprises two routers, a set of homogenous TCP sender, and a receiver used in simulating different scenarios. The network parameter values were defined as follows: N = 60 (TCP session number), C=15 Mbps (3750 packet/seconds link capacity) with packet size=500 byte, Tp =0.2 seconds (the propagation delay) and R= 0.253 seconds (the round-trip time) where $q_{des} = 300$ packets (the desired queue size) and $q^{max} = 700$ packets (maximum queue length in router 1).

For the simulation purpose, the transfer function for the continuous system given in (3) for the above values must be converted to a discrete transfer function using zero-order hold through the MATLAB command (c2d) with sampling time (0.125). The discrete transfer function is given as:

$$P(z) = \frac{q(z)}{p(z)} = \frac{742.8z + 619}{z^2 - 1.555z + 0.5783} \tag{10}$$

Figure 8, illustrated the systems' behavior without any AQM controller. it is noticeable that the queue highly congested and could not follow the desired queue length.

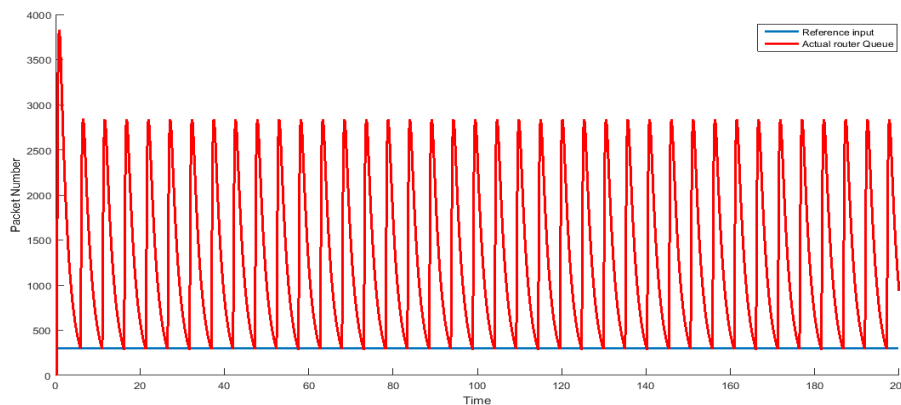


Figure 8. System response without controller

4.2. Simulations Scenarios

Scenario 1: First, a PI controller was analyzed as a comparative reference. The GWO parameters used for tuning the postulated controller were given in Table 4.

Table 4. Factors of GWO Algorithm

Factor	Value
Search agents	30
Dimension	3
Number of iteration	100

The controller gain by the GWO algorithm was: $Kp = 0.001, Ki = 0.0009$. Figure 9, showed the PI and fuzzy-PI-based GWO performance. From the figure, the tracking ability of the postulated controller was illustrated with lower overshoot and reduced settling time when compared to PI controller.

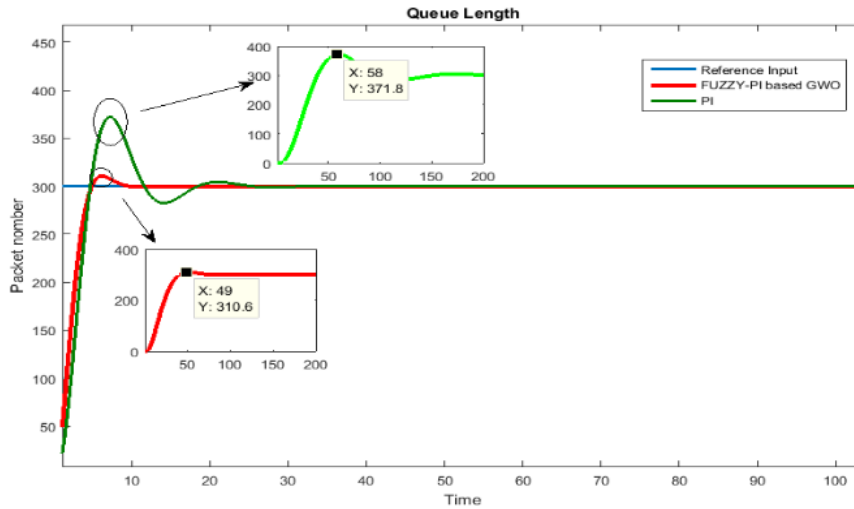


Figure 9. Queue length for fuzzy-PI based GWO

Scenario 2: In this scenario, the performance of the fuzzy-PI based GWO controller was determined in a dynamic network traffic. Initially, for t=0, the number of TCP connection (N) was 60, at t=30, 20 more TCP connections initiated the transmission. The discrete transfer function for this scenario is given as:

$$P(z) = \frac{553.2z + 457.8}{z^2 - 1.536z + 0.5666} \tag{11}$$

Figure 10, showed the corresponding queue length obtained from the PI and fuzzy-PI based GWO. It was observed that the PI was weakened by the load changes, while the postulated controller was strong based on the number of TCP connections.

Scenario 3: The toughness of the postulated method over changes in the desired queue length was determined in this scenario. Firstly, the target queue size was 200 packets and later increased to 350, 300, 400, 300, 350, and 200 packets every 25 seconds. Figure 11, depicted that the fuzzy-PI based GWO successfully controlled the queue length within the target and attained a quick response time and lower overshoot (loss rate of data packets) than the PI controller.

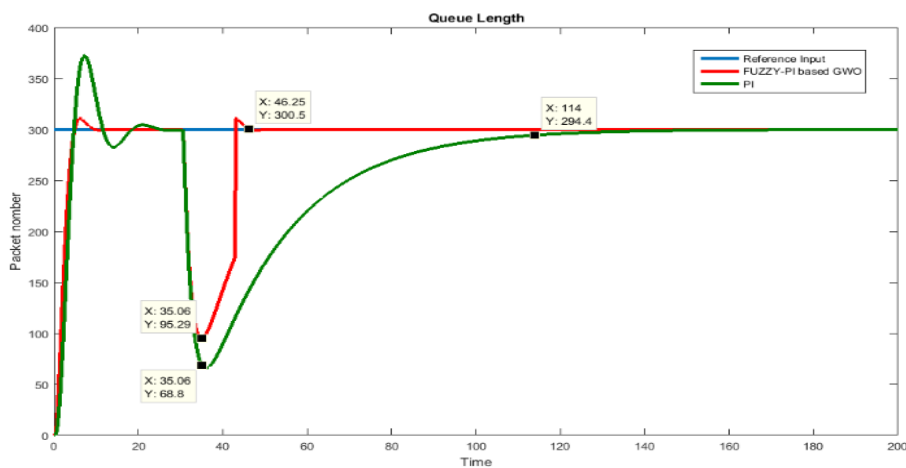


Figure 10. Queue length for fuzzy-PI based GWO with changing TCP load

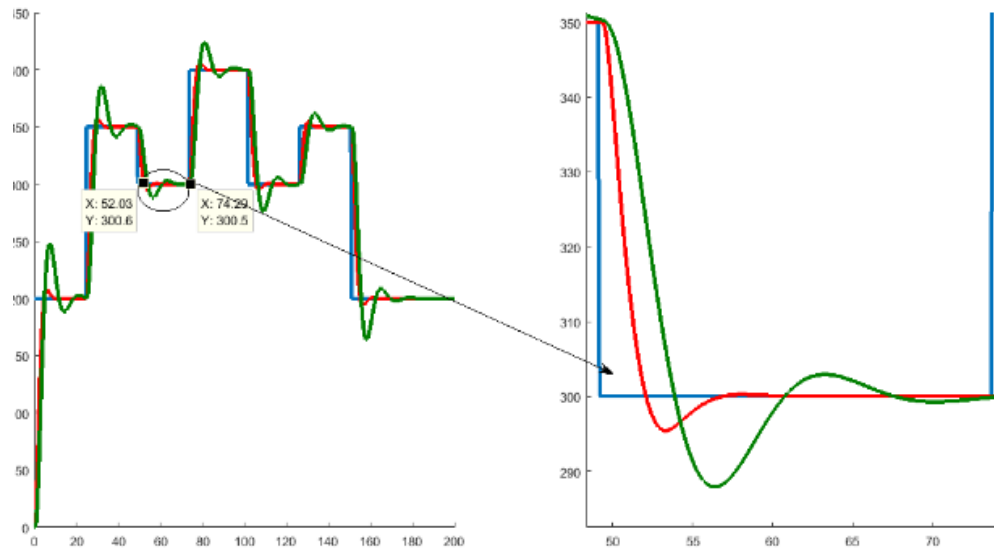


Figure 11. Queue length for fuzzy-PI based GWO with changing target queue

5. CONCLUSION

This study postulated a GWO algorithm as an AQM scheme for tuning fuzzy-PI controllers to avoid network congestion. The novelty of this approach lies in utilizing the GWO algorithm for TCP/AQM network. The fuzzy-PI based GWO was simulated using MATLAB in three different scenarios and the simulation results were compared.

The proposed algorithm was shown to effectively secure the queue length within the target based on the number of TCP connections, the target queue length, and the RTT. In addition, the postulated controller attained a faster settling time and less overshoot (Loss rate of data packets) than PI.

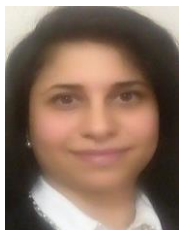
The simulation results were improved significantly which indicates the potential of applying GWO algorithm-tuned control models as AQMs in a computer network. Therefore, future studies can improve the results of this study by utilizing more complex network scenarios. Furthermore, it is worth studying the ability of the design and invention of this AQM scheme in reality.

REFERENCES

- [1] V. Jacobson, "Congestion avoidance and control," in *ACM SIGCOMM computer communication review*, 1988, pp. 314-329.
- [2] S. Floyd and V. Jacobson, "On traffic phase effects in packet-switched gateways," *Internetworking: Research and Experience*, vol. 3, pp. 115-156, 1992.
- [3] J. Aweya, M. Ouellette, D. Y. Montuno, and K. Felske, "Design of rate-based controllers for active queue management in TCP/IP networks," *Computer Communications*, vol. 31, pp. 3344-3359, 2008.
- [4] H. C. Cho, S. M. Fadali, and H. Lee, "Adaptive neural queue management for TCP networks," *Computers & Electrical Engineering*, vol. 34, pp. 447-469, 2008.
- [5] W. Zhang, L. Tan, and G. Peng, "Dynamic queue level control of TCP/RED systems in AQM routers," *Computers & Electrical Engineering*, vol. 35, pp. 59-70, 2009.
- [6] L. Yu, M. Ma, W. Hu, Z. Shi, and Y. Shu, "Design of parameter tunable robust controller for active queue management based on H_∞ control theory," *Journal of network and computer applications*, vol. 34, pp. 750-764, 2011.
- [7] V. Kushwaha, "Interaction of High Speed TCPs with Recent AQMs through Experimental Evaluation," *International Journal of Computer Network & Information Security*, vol. 8, 2016.
- [8] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on networking*, vol. 1, pp. 397-413, 1993.
- [9] T. Eguchi, H. Ohsaki, and M. Murata, "On control parameters tuning for active queue management mechanisms using multivariate analysis," in *Applications and the Internet, 2003. Proceedings. 2003 Symposium on*, 2003, pp. 120-127.
- [10] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *ACM SIGCOMM Computer Communication Review*, 2000, pp. 151-160.
- [11] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Transactions on automatic control*, vol. 47, pp. 945-959, 2002.

- [12] S. S. Sabry and T. M. Nayl, "Particle Swarm Optimization Based LQ-Servo Controller for Congestion Avoidance," *Iraqi Journal of Computers, Communication and Control & Systems Engineering*, vol. 19, pp. 63-70, 2019.
- [13] M. Z. Al-Faiz and S. S. Sabry, "Optimal linear quadratic controller based on genetic algorithm for TCP/AQM router," in *Future Communication Networks (ICFCN), 2012 International Conference on*, 2012, pp. 78-83.
- [14] G. Di Fatta, G. L. Re, and A. Urso, "A fuzzy approach for the network congestion problem," in *International Conference on Computational Science*, 2002, pp. 286-295.
- [15] M. Bernal, A. Sala, A. Jaadari, and T.-M. Guerra, "Stability analysis of polynomial fuzzy models via polynomial fuzzy Lyapunov functions," *Fuzzy Sets and Systems*, vol. 185, pp. 5-14, 2011.
- [16] E. Ruano, S. S. Ge, T. M. Guerra, F. L. Lewis, J. C. Principe, and M. Colnarič, "Computational intelligence in control," *Annual Reviews in Control*, vol. 38, pp. 233-242, 2014.
- [17] M. Mekhanet, L. Mokrani, A. Ameer, and Y. Attia, "Adaptive fuzzy gain of power system stabilizer to improve the global stability," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 5, pp. 421-429, 2016.
- [18] Z. S. Salam Waley Shneen, Qusay A. Jawad, Haider Shareef, "Advanced optimal by PSO-PI for DC motor," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. Vol. 16, No. 1 pp. 165-175, 2019.
- [19] R. Mohammedi, R. Zine, M. Mosbah, and S. Arif, "Optimum Network Reconfiguration using Grey Wolf Optimizer," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, pp. 2428-2435, 2018.
- [20] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46-61, 2014.
- [21] L. Reznik, "Fuzzy controllers handbook: how to design them, how they work", *Elsevier*, 1997.
- [22] H. Bevrani and P. R. Daneshmand, "Fuzzy logic-based load-frequency control concerning high penetration of wind turbines," *IEEE systems journal*, vol. 6, pp. 173-180, 2012.
- [23] J. Zhao and B. K. Bose, "Evaluation of membership functions for fuzzy logic controlled induction motor drive," in *IECON-Proceedings-*, 2002, pp. 229-234.
- [24] R. Mohammedi, R. Zine, M. Mosbah, and S. Arif, "Optimum Network Reconfiguration using Grey Wolf Optimizer," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, pp. 2428-2435, 2018.
- [25] R.-E. Precup, R.-C. David, and E. M. Petriu, "Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity," *IEEE Transactions on Industrial Electronics*, vol. 64, pp. 527-534, 2016.

BIOGRAPHIES OF AUTHORS



Sana Sabah Sabry was born in Basrah, Iraq in 1986. She received her B.S. degree in computer Engineering from Al-Nahrain University, Iraq in 2008 and her M.S. in Control Engineering from Al-Nahrain University, Iraq in 2011. She has 10 years' work experience at different educational organizations. She is currently a lecturer in the college of engineering at university of information technology and communications, Baghdad, Iraq. Her research topics include control theory, optimization algorithms and congestion control in computer networks, Internet of Things, Machine Learning. She can be contacted via email sana.sabah@uoitc.edu.iq.



Nada Mahdi Kaittan is a Lecturer at college of business informatics from University of Information Technology and Communications. She graduated with Bachelor in information engineering Engineering in 2008 from from Al-Nahrain University, Iraq and obtained Master in information engineering from from Al-Nahrain University, Iraq in 2013. Her research interests are on control theory, Image processing, optimization. She can be contacted via email nadait.2016@uoitc.edu.iq.