

Particle Swarm Optimization (PSO) Based Robust Active Queue Management Design for Congestion Control in TCP Network

Believe Bariledum Nwamae[1] Ledisi G. Kabari [2*]

1. School of Applied Sciences, Ignatius Ajuru University of Education, P. M. B. 5047, Rumulumeni, Port Harcourt, Nigeria
2. School of Applied Sciences, Ken Saro-Wiwa Polytechnic, P. M. B. 20, Bori, Nigeria

* E-mail of the corresponding author: ledisigiokkabari@yahoo.com

Abstract

Active queue management (AQM) is an effective solution for the congestion control problem. It can achieve high quality of service (QoS) by reducing the packet dropping probability and network utilization. Robust particle swarm optimization (PSO) algorithm is proposed in this paper in order to design a robust AQM schemes. Robust PSO controllers can achieve desirable time-response specifications with a simple design procedure and low-order controller in comparison to the conventional H∞ controller. Ranges of system parameters change and iterations are used to show the robustness of the designed controllers. The ability of the designed controllers to meet the specified performance is demonstrated in this paper by simulations using MATLAB, (R2016a). Finally, it was shown that the proposed robust PSO can achieve desirable performance.

Keywords: Active queue Management (AQM), Quality of Service (QoS) Particle Swarm Optimization(PSO), Transmission control protocol (TCP), MATLAB.

1. Introduction

As the use of the internet continues to grow, there is increase in competition for network resources (Ali and Khalid, 2016). Competition for network resources affects the performance of networks. Under light node, any network performs reasonably well, however, problems surface when they are extensively used. The most notable and common problem that networks are faced with is loss of data. There are a variety of reasons, to explain loss of data in a network, congestion is the most common one. Loosely speaking, congestion refers to the loss of network performance when a network is heavily loaded. This loss of performance can be data loss and large delays in data transmission, which are often unacceptable. Because of this, control and avoidance of congestion is a critical issue in network management and design (Peterson and Davie, 2012).

Transmission control protocol (TCP) congestion control mechanism, while necessary and powerful, is not sufficient to provide good service under all circumstances, especially with the rapid growth in size and the strong requirements to quality of service (QoS) support, because there is a limit to how much control can be accomplished at the end system (Ali and Khalid, 2016). Some measures need to be implemented in the intermediate nodes to complement the end-system congestion avoidance mechanisms. Active queue management (AQM), as one class of packet dropping/marking mechanism in the router queue, has been proposed to support the end-to-end congestion control in the Internet. Network congestion occurs when the amount of data injected in the network is larger than the amount of the data that is delivered to the destinations. TCP protocol adopts the end-to-end approach that represents the beginning of network congestion control, where the responsive data sources reduce their transmission rate when they infer the occurrence of congestion from packet losses(Kurose and Ross (2013).

Many AQM schemes have been proposed. The first AQM scheme, known as random early detection (RED) (Misra et al., 2010), was introduced to routers for solving the synchronization problem and keeping the average queue length low, but it is sensitive to traffic loads and parameter configurations. Therefore, some modified RED schemes have been proposed to minimize the problems of RED, such as adaptive random early dedication (ARED), loss ratio based red (LRED), dynamic-red (DRED), and fair random early drop (FRED).

Recently, control theory was applied to design AQM schemes. Specifically, the PI controller (Wang et al., 2006) and (Hollot et al.,2001), PID (proportional–integral–derivative) controller (Yanfie et al., 2003), and Fuzzy PID controller (Yan and Lei, 2011) were proposed based on the linear-time-invariant model of the TCP/AQM system for congestion control in computer networks with different design methods for obtaining suitable controller parameters. On the other hand, the wide change in parameters of TCP/AQM causes high packet losses and large delay; therefore, it is necessary to develop a high-performance closed-loop controller. It has been shown that quite a number of papers (Kuo et al., 2008) deal with small changes in network parameters. Furthermore, it was found that the robustness of the congestion control algorithm is still not well addressed under a dynamic network environment.

In this paper, Particle Swarm Optimization algorithm is proposed in order to design a robust AQM algorithm. Swarm intelligence optimization methods are used to arrive at the near-optimum solution to our optimization problem, for which one optimization method may fail or may not satisfy the more desirable performance specifications. The PSO method is suitable for solving optimization problems because of its fast convergence, small number of parameters, and simplicity of operations (Ali and Khalid, 2016).

2. TCP Control Model

Nonlinear dynamic model for TCP flow control has been developed based on fluid flow theory(Misra et al., 2010). This model can be stated as in equation1

$$\begin{cases} \frac{dW(t)}{dt} = \frac{1}{R(t)} - \frac{W(t)W\left(t-R(t)\right)}{2R(t)} p\left(t-R(t)\right) \\ \frac{dq(t)}{dt} = \frac{N(t)}{R(t)} W(t) - C(t) \end{cases} \tag{1}$$

The nonlinear and time-varying system above was approximated as a linear constant system by small-signal linearization about an operating point as in figure1. In the block diagram, C(s) and G(s) are the controller and the plant, respectively. The meaning of parameters presented in Figure1 is as following:

$$K(t) = \frac{[R(t)C(t)]^3}{2N(t)}, \qquad T_1(t) = R(t),$$
$$T_2(t) = \frac{R^2(t)C(t)}{2N(t)} \tag{2}$$

where
C(t) : Link capacity (packets/sec)
qo : Queue reference value
N(t) : Load factor, i.e., number of active sessions
R(t) : Round-trip time (RTT), ( ) $R(t) = 2\,q(t)\,/\,C(t) + T\,p$, $T\,p$ is the fixed propagation delay
p(t) : Dropping/marking probability
q(t) : Instantaneous queue

It was suggested that the AQM controller designed with the simplified and inaccurate linear constant model should not be optimal, because the actual network is very changeful; the state parameters are hardly kept at a constant value for a long time (Mahdi Jalili-Kharaajoo et al., 2013). Moreover, the equations (1) only take consideration into the fast retransmission and fast recovery, but ignore the timeout mechanism caused by lacking of enough duplicated ACK, which is very usual in burst and short-lived services. In addition to that, there are many non-receptive User Datagram Protocol (UDP) flows besides TCP connections in networks; they are also not included in equation (1). These mismatches in model will have negative impact on the performance of controller designed with the approach depending with the accurate model. For the changeable network, the robust control should be an appropriate choice to design controller for AQM. The variable structure sliding mode control action is one of the best that can help us.

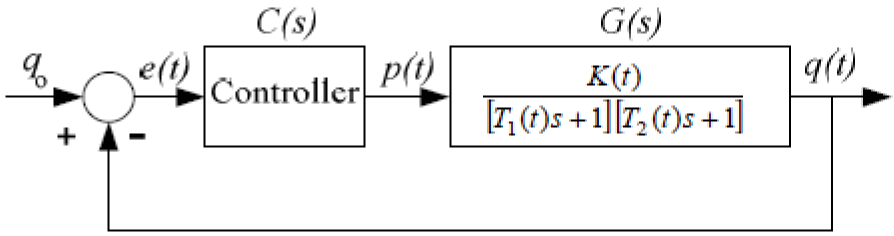

Figure 1- Blocked diagram f AQM control System

3. PSO Controller Design

The PID controller is the most commonly used controller. Much interest was shown to develop PID control in the early development of automatic control. However, there has been a resurgence of interest in PID control because of the possibility of making PID controllers with automatic tuning, automatic generation of gain schedules, and continued adaptation. The PID controller is expressed by equation (3).

$$K(s) = k_p + \frac{k_i}{S} + k_d s \tag{3}$$

where k_p, k_i, and k_d are the controller parameters.

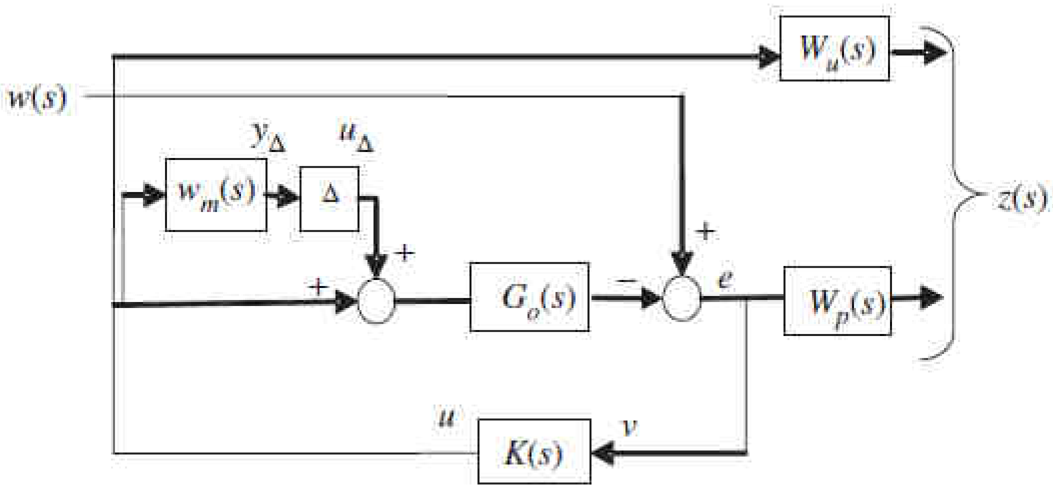

Figure 2 - System with weighting functions and uncertainty (Ali and Khalid, 2016)

PID controller is designed based on PSO algorithm and H∞ constraint, and it is called robust PSOPID controller. PSO is an evolutionary computation optimization technique (a search method based on a natural system) developed by Kennedy and Eberhart in 1995. The behavior of PSO can be envisioned by comparing it with bird swarms searching for optimal food sources, where the direction in which a bird moves is influenced by its current movement, and the best food source it ever experienced. In other words, birds are driven by their inertia, their personal knowledge, and the knowledge of the swarm. In terms of PSO, the movement of a particle is influenced by its inertia, its personal best position, and the global best position. The basic concept of the PSO technique lies in accelerating each particle toward its Pbest (personal best position) and Gbest (global best position) locations, with a random weighted acceleration at each time step. (Ali and Khalid, 2016)

The velocity and position of the particle are changed according to the following velocity and position equations, respectively as in equation (4):

$$\begin{aligned} V_{id} &= h \times V_{id} + C_1 \times \mathrm{rand}_1 \times (P_{id} - X_{id}) + C_2 \times \mathrm{rand}_2 \\ &\quad \times (P_{gd} - X_{id}) \\ X_{id} &= X_{id} + V_{id} \end{aligned} \tag{4}$$

where V_{id} and X_{id} represent the velocity and position of the ith particle with d dimensions, respectively; rand_1

and $rand_2$ are two uniform random functions between 0 and 1; h is the inertia weight; and P_{id} and P_{gd} are the best value and the global best value, respectively. The constants C1 and C2 are termed the cognition and social components (learning factors), respectively

The PSO method is used for tuning the parameters of performance weighting function and the PID controller parameters that ensure a controlled system with performance and robust stability. The optimal parameters have been obtained by minimizing the following proposed cost function in equation (5):

$$J_{\min} = \|W_{\mathrm{p}} S + W_{\mathrm{m}} T\|_{\infty} + \frac{1}{M}\sum_{i=1}^{M}(e_i - \mu)^2 \tag{5}$$

where M is the length of error vector, μ is the mean of the error, and e is the error of the system. The sensitivity function (S) multiplied by the performance weighting function (Wp) represents the condition of robust performance, and the complementary sensitivity function (T) multiplied by the performance weighting function represents the condition of robust stability.

The flowchart is shown in Figure 4, which describes the PSO algorithm for obtaining the optimal parameters of the robust controller and weighting function. The overall block diagram of the system with the PSO tuning algorithm is shown in Figure 3. It is worth highlighting the ability of method to swiftly converge to best solutions. It does not require any gradient information of the cost function to be optimized. The algorithm obtains the minimum value of the infinity norm of the performance criterion from the search space that minimizes the cost function. The following parameters have been used for carrying out the robust PSOPID controller:

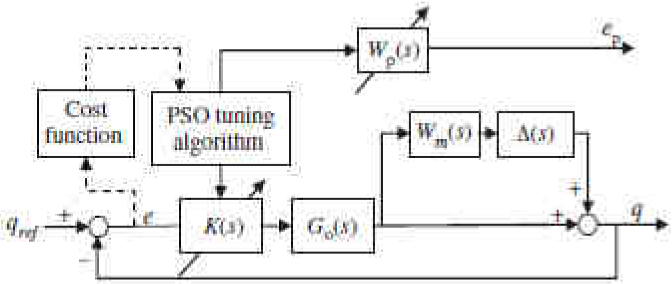

Figure 3 -Overall block diagram of the controlled system

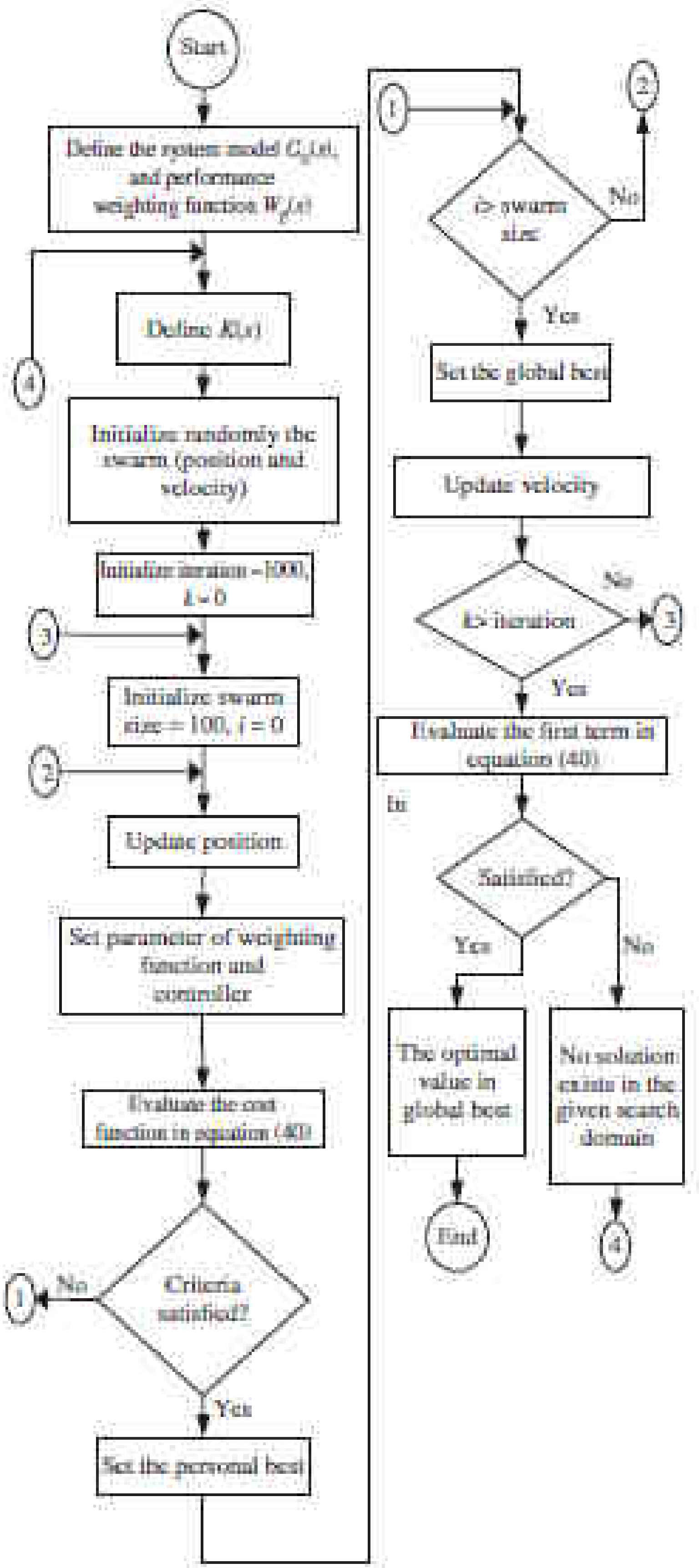

Figure 4-Flowchart for robust PSO controller design

4. Simulation Experiments and Results

4.1 Experiment 1

The members of each individual in the PSO algorithm:

*k*p, *ki* , *k*d, *M*s, *ω*b and *e*ss.

(ii) Swarm size: 50.

(iii) Inertia weight factor: *h* = 2.

(iv) *C*1 = 2 and *C*2 = 2.

(v) Maximum number of iterations: 1000.

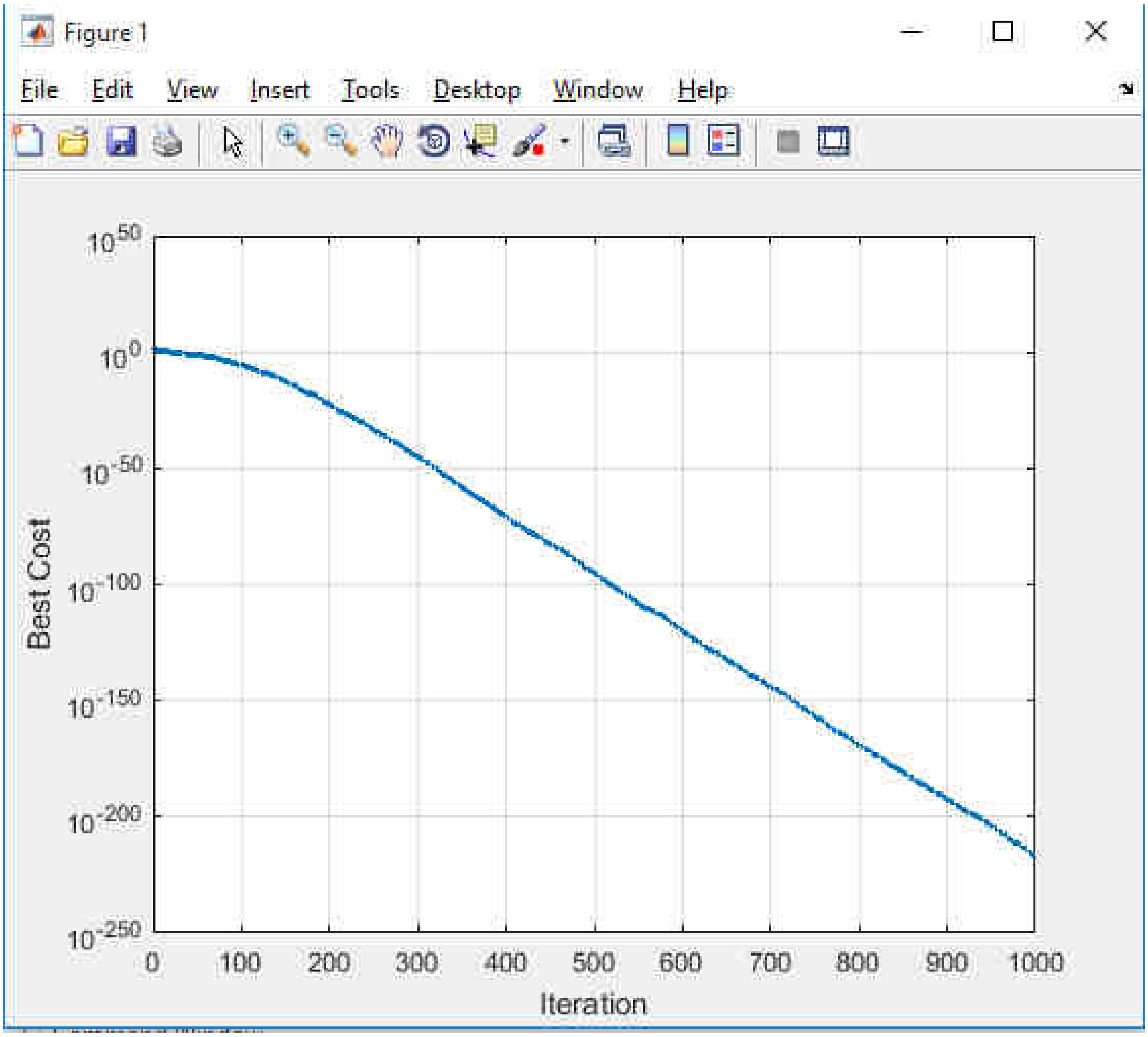

Figure 3 - 1st Scenario Diagram of Swarm Population at 50

4.2 Experiment 2

The members of each individual in the PSO algorithm:

*k*p, *ki* , *k*d, *M*s, *ω*b and *e*ss.

(ii) Swarm size: 100.

(iii) Inertia weight factor: *h* = 2.

(iv) *C*1 = 2 and *C*2 = 2.

(v) Maximum number of iterations: 1000.

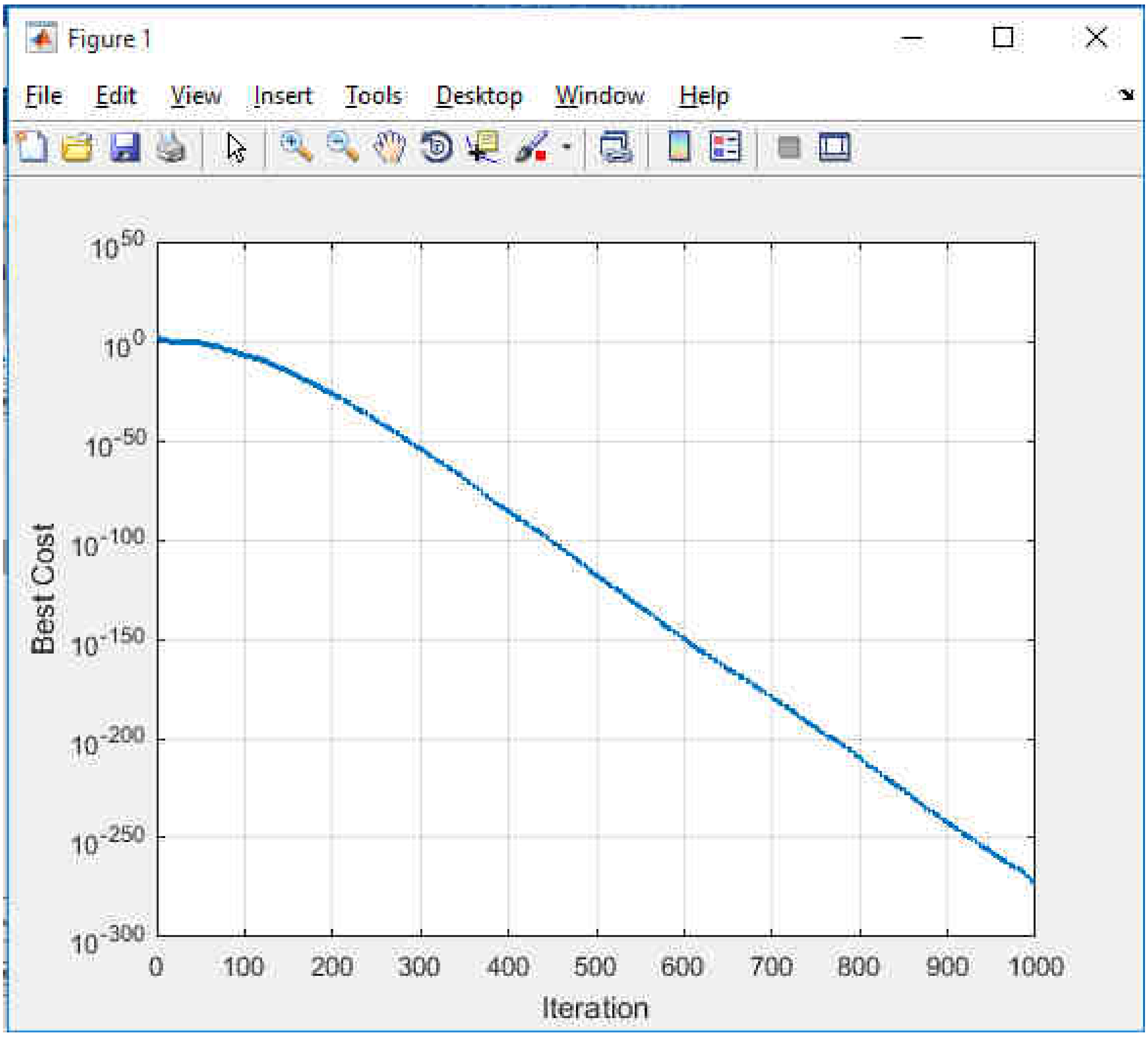

Figure 4 – 2nd Scenario Diagram of Swarm Population at 100

4.3 Experiment 3

The members of each individual in the PSO algorithm:

*k*p, *ki* , *k*d, *M*s, *ω*b and *e*ss.

(ii) Swarm size:200.

(iii) Inertia weight factor: $h = 2$.

(iv) $C1 = 2$ and $C2 = 2$.

(v) Maximum number of iterations: 1000.

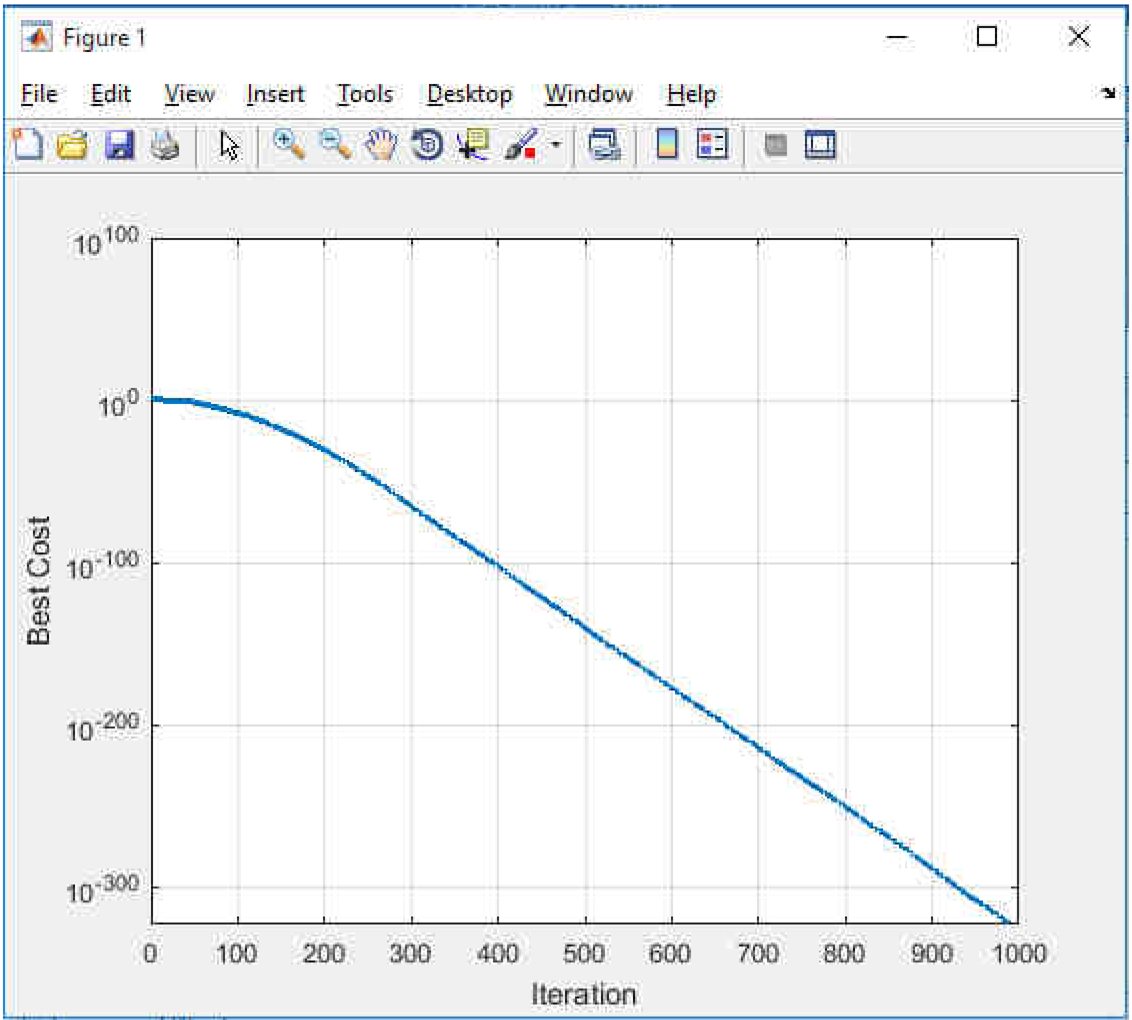

Figure 5- 3rd Scenario Diagram of Swarm Population at 200

4.4 Experiment 4

The members of each individual in the PSO algorithm:

*k*p, *ki* , *k*d, *M*s, ωb and *e*ss.

(ii) Swarm size: 400.

(iii) Inertia weight factor: *h* = 2.

(iv) *C*1 = 2 and *C*2 = 2.

(v) Maximum number of iterations: 1000.

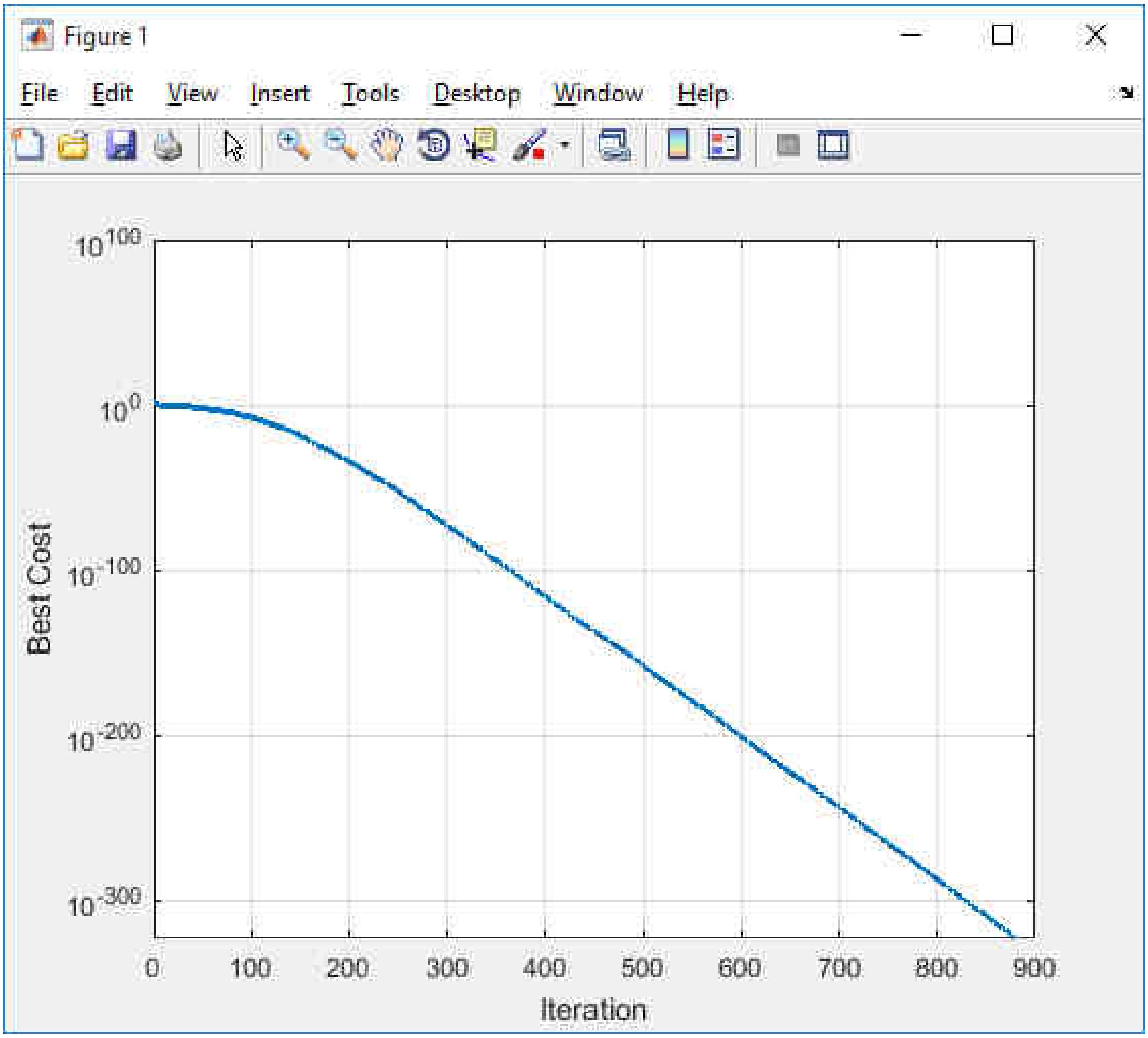

Figure 6 - 4th Scenario Diagram of Swarm Population at 400

Table 1, Summary of the PSO simulation with different Swarm

Scenario	**1**	**2**	**3**	**4**
Swarm Size	50	100	200	400
Achieve Convergence @	Does not achieve convergence after 1000 iterations	Does not achieve convergence after 1000 iterations	Achieves convergence just before 1000th iteration	Achieves convergence just before 900th iteration

From simulation tests in Table 1, it was found that the above settings for the PSO parameters were adequate for this application. Data from table1 suggests that significant increase in the swarm population affects the rate of convergence, since all other parameters were not adjusted.

5. Conclusion

This is study has investigated the implementation of PSO algorithm in improving AQM techniques so as to mitigate network congestion. MATLAB was used to simulate the efficiency of the PSO algorithm. The results shows that PSO algorithm is very reliable especially under very high stress conditions that computer network undergo.

References

Ali, H., & Khalid, K. (2016). Swarm Intelligence Based Robust Active Queue Management Design for Congestion Control in TCP Network. IEEJ Transactions On Electrical And Electronic Engineering, 11(3), 308-324. doi: 10.1002/tee.22220

Hollot, C., Misra, V., Towsley, D., & Wei-Bo Gong. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. Proceedings IEEE INFOCOM 2001. Conference On Computer Communications. Twentieth Annual Joint Conference Of The IEEE Computer And Communications Society (Cat. No.01CH37213). doi: 10.1109/infcom.2001.916670

Kuo, H., Chen, C., Yan, J., & Liao, T. (2008). A GA-based PID Active Queue Management Control Design for TCP/IP Networks. Journal of Physics: Conference Series, 96, 012101. doi: 10.1088/1742-6596/96/1/012101

Kurose, J., & Ross, K. (2013). Computer networking. Boston: Pearson. Mahdi Jalili-kharaajoo, Alireza Dehestani and Hassan Motallebpour(2013). Active Queue Management: A New Robust Fuzzy Second Order Sliding Mode Control Applied to. Iran Telecommunication Research Center. Retrieved from https://pdfs.semanticscholar.org/c870/70e03690ce79bcc6cb3328ae4648b69f1231.pdf

Misra, S., Oommen, B., Yanamandra, S., & Obaidat, M. (2010). Random Early Detection for Congestion Avoidance in Wired Networks: A Discretized Pursuit Learning-Automata-Like Solution. IEEE Transactions On Systems, Man, And Cybernetics, Part B (Cybernetics), 40(1), 66-76. doi: 10.1109/tsmcb.2009.2032363

Peterson, L., & Davie, B. (2012). Computer networks (4th ed.). Amsterdam: Morgan Kaufmann.

Wang, X., Wang, Y., Zhou, H., & Huai, X. (2006). PSO-PID: a novel controller for AQM routers. 2006 IFIP International Conference On Wireless And Optical Communications Networks. doi: 10.1109/wocn.2006.1666682

Yan, Q., & Lei, Q. (2011). A New Active Queue Management Algorithm Based on Self-Adaptive Fuzzy Neural-Network PID Controller. 2011 International Conference On Internet Technology And Applications. doi: 10.1109/itap.2011.6006116

Yanfie, F., Fengyuan, R., & Chuang, L. (2003). Design a PID controller for active queue management. Proceedings Of The Eighth IEEE Symposium On Computers And Communications. ISCC 2003. doi: 10.1109/iscc.2003.1214244