ITC 1/54
Information Technology and Control
Vol. 54 / No. 1/ 2025
pp. 307-328
DOI 10.5755/j01.itc.54.1.37862

Optimization of RED-PID Controller Using the Chaotic-Subpopulation Strategy-based Aquila and Math Algorithms

Received 2024/07/02 | Accepted after revision 2025/02/04

HOW TO CITE: Tang, J., Ma, R., Li, H., Liang, X. (2025). Optimization of RED-PID Controller Using the Chaotic-Subpopulation Strategy-based Aquila and Math Algorithms. *Information Technology and Control*, 54(1), 307-328. https://doi.org/10.5755/j01.itc.54.1.37862

Optimization of RED-PID Chaotic-Subpopulation Strategy-based Aquila and Math Algorithms

Junyong Tang, Ruilong Ma, Hui Li, Xiangyang Liang

Department of Computer Science and Engineering, Xi'an Technological University, Xi'an, China;
e-mails: jytang@xatu.edu.cn, maruilong@st.xatu.edu.cn, lihui@st.xatu.edu.cn, xyliang@xatu.edu.cn

Jiankang Zhang

Department of Computing and Informatics, Bournemouth University, Poole, U.K;
e-mail: jzhang3@bournemouth.ac.uk

Corresponding authors: xyliang@xatu.edu.cn

The Transmission Control Protocol (TCP) plays a crucial role in congestion control by adjusting packet sending rates, but it falls short of addressing the buffer bloat issue in critical routers. To mitigate this, Active Queue Management (AQM) mechanisms like Random Early Detection (RED) have been introduced to form a TCP/RED feedback system for congestion control. However, by analyzing the magnitude-frequency characteristic of TCP/RED, this paper finds it has sluggish response time and slowly stabilizes in congestion control. Therefore, this paper presents a novel AQM controller named RED-PID, which integrates a Proportional-Integral-Derivative (PID) adjustor into RED, enhancing the control structure. Furthermore, frequency domain analysis provides the stability criteria and parameters for TCP/RED-PID. Given the lack of a special optimization of control parameters for adapting to TCP/RED-PID effectively, this paper introduces a novel heuristic algorithm (AOMOA), which combines the global exploration strengths of the Aquila Optimizer (AO) with the local exploitation capabilities of the Math Optimizer (MO). Meanwhile, a chaotic-subpopulation strategy is proposed,

utilizing two subpopulations simultaneously to fasten the converging speed. Moreover, the dynamic k-worst shift is introduced to strike a balance between global exploration and local exploitation across both optimizers. The TCP/RED-PID model was analyzed and validated using the NS3 simulator. Comprehensive simulations demonstrate that RED-PID, optimized by AOMOA, significantly outperforms the standard RED controller, exhibiting superior congestion control performance.

KEYWORDS: AQM; TCP/RED; Congestion control; PID; Aquila Optimizer; Math Optimizer.

1. Introduction

With the integration of more applications in networks to meet the demands of a growing number of web users, network congestion has become a prevalent issue. Despite the significance of the Transmission Control Protocol (TCP) in congestion control by adjusting senders and receivers, TCP focuses on the peer-to-peer side and overlooks the buffer bloat problem in critical routers. Consequently, the effectiveness of TCP in resolving congestion is limited, as it only works from a single perspective [15, 32]. In response, Active Queue Management (AQM) [2, 36, 37] has arisen and combined with TCP to construct a feedback system (TCP/AQM) as another solution to diminish buffer queue sizes in vital routers, ensuring that the packet queue length remains within a specified range [16]. Traditional AQM controllers use fixed algorithms to drop the incoming packets. They are insensitive to time variations of the packet queue [14]. For this problem, an AQM controller, Random Early Detection (RED), is proposed to control network congestion based on variations of queue length. Up to now, RED has been a classical and widely used AQM controller and plays a significant role in modern computer networks, especially with high throughput and low latency networks emerging.

However, utilizing entirely RED to avoid congestion and improve transmission efficiency is very difficult. The main reason is that the burst microflows everywhere in networks aggravate fluctuations of the packet queue. In this case, RED is easily affected by instant variations of the packet queue and, therefore, drops packets inaccurately. Meanwhile, RED does not follow the past track of the variations to identify congestion. In fact, the past track of queue variations often reflects the trend of congestion evolution. The ignorance of considering the history experience of the packet queue degrades the RED's function. In addition, several parameters in RED, e.g., the maximum queue length (lmax), the minimum queue length (lmin), and the maximum dropping packet probability (pmax), play an essential role in making the most advantage of the RED performance. Setting available values of these parameters is blind as it depends on expertise.

According to the above analysis, the focus of taking advantage of RED has been to design an effective and new mechanism for RED to cope with constant variations of network communication. This paper proposes a high-performance RED controller incorporating the Proportional-Integral-Derivative (PID) adjustor, namely RED-PID, which controls network congestion effectively and steadily. Because the TCP peer sides and the RED-PID controller construct a feedback system (TCP/RED-PID), the parameters of RED-PID need to be optimized to balance the feedback system sensitivity and stability. To achieve the above aim, we design a new heuristic algorithm based on the Aquila optimization and Math optimization (MO) algorithms (AOMOA) for optimizing AQM/RED.

The main contributions of this paper can be summarized as follows.

- According to the AQM/RED fluid model and the feedback theory, this paper reveals an intrinsic drawback in RED - its lack of responsive control to variations in the packet queue degrades the effectiveness of congestion control.
- To address the flaw of RED's difficulty in coping with constant variations of the packet queue, a PID adjustor is introduced into RED to improve its intrinsic sluggish character. To the best of our knowledge, although PID control is popular in automatic and electrical fields, RED-PID is the first RED controller to incorporate the PID algorithm in network communication.
- We delineate the parameter optimization model, which maintains the TCP/RED-PID stability and sensitive responsiveness to congestion control.

– To optimize the constrained parameters of TCP/RED-PID, we propose a heuristic algorithm (AOMOA) that combines global exploration of AO with local exploitation of MO. This hybrid approach leverages the strengths of both algorithms to achieve comprehensive parameter optimization.

The rest of the paper is structured as follows. The related work is introduced in Section 2. Section 3 analyzes shortcomings of RED and, therefore, offers the RED-PID fluid model to improve the original TCP/RED structure. The introduction to AO is in Section 4. In Section 5, the motivation and mathematical model of AOMOA are given in detail. In Section 6, AOMOA, with other compared algorithms, is tested. In addition, the performance of AOMOA-optimized RED-PID is also verified. A comprehensive discussion about the research's purpose, methodology, future work, etc., is given in Section 7. Finally, this paper is summarized in Section 8.

2. Related Works

AQM/RED plays a crucial role in suppressing congestion evolution, effectively mitigating packet loss and alleviating network congestion. Meanwhile, AQM/RED would perform excellently only on the condition of complex parameter tuning aided by heuristic algorithms. This section provides a comprehensive overview of existing research related to AQM/RED and heuristic algorithms.

2.1. AQM/RED

The AQM mechanism is deployed on intermediate network devices, especially critical routers. AQM assists TCP in reducing packet loss and relieving network congestion. RED is one of the most classical and popular AQM mechanisms due to its implementation following the evolution of the packet queue. Firstly, we summarize the current work related to RED.

The drop packet policy of RED is characterized by adopting a linear policy of dropping packets. Hassan et al. [17] replace the linear policy with the amended method and the quadratic exponential approach, to combat network complex and frequent variations. Giménez et al. [13] present a modified Dynamic Beta RED (mDBetaRED), in which the parameters are dynamically adjusted so that the queue length remains stable around a predetermined reference value for fitting new network traffic conditions. Additionally, Lhamo et al. [24] combine the RED and CoDel with Static Priority (SP) scheduling for quality-of-service differentiation to make the AQM with essentially parameterless priorities. These studies focus on the vision that changes the rigidly linear packet discard of RED to adapt dynamically to the variations of network communication. However, adjusting linear dropping packets is only available to limited scenarios. This type of approach makes it difficult to cope with complex communication cases, such as heterogeneous networks and concurrent flows competing bottleneck-link bandwidth.

To get AQM strong adaptability, some studies have considered the dynamics of TCP transport and AQM as a closed-loop feedback system, referred to as TCP/AQM. With the assistance of the feedback control theorem, TCP/AQM achieved good results in congestion control. For instance, a robust AQM, PID-R [38], combines the PID adjustor with the Recursive Least Squares filter to resist the interference of random communication variations and, thus, maintain the high performance of the AQM controller. Khan et al. [23] use the nonlinear-loop control system to describe delay-based congestion behavior. Abd Mohammed et al. [1] utilize proportional-integral control to limit queue growth better. These researches all verify the AQM/RED controller requires appropriate control parameters [33] (e.g., link bandwidth, concurrent TCP flows, transmission rate, round-trip time, etc.) for optimal effectiveness. Therefore, the parameters of TCP/AQM must be optimized to meet actual communication scenarios for effective congestion control.

2.2. Meta-heuristic Algorithms

It can be seen from the analysis in Section 2.1 that adopting available optimization algorithms is of great significance. In recent years, newly proposed algorithms, which leverage CNN [28] or supervised learning methods, have been applied in image and vision processing. They have relied on plenty of training sets to achieve excellent performance. However, TCP/AQM, as a feedback system, must instantly adjust according to communication variations. Additionally, as far as we know, no available training set currently

applies to TCP/AQM due to network communication fluctuating. In view of this, meta-heuristic algorithms of animal-based or nature-based phenomena are considered when optimizing the parameters in TCP/AQM.

New heuristic algorithms combining classical algorithms or creating adaptable versions have been proposed recently to improve the development of relevant research [18, 24, 27]. For example, Akyol et al. [7] propose a new hybrid approach that uses the tangent search algorithm (TSA) instead of the limited exploration stage to improve the exploitation of Aquila optimizer (AO). Tang and Wang [37] propose a whale-based atom-like structure differential evolution (WOAAD). Han et al. [10] are inspired by the behaviors of walruses and then proposed the Walrus optimizer (WO). The reptile search algorithm (RSA) [4], emulating the hunting behavior of crocodiles, is proposed to solve optimization problems. However, these optimization algorithms are often tailored to specific engineering problems. Acquiring available and accurate parameters is still tedious and needs specific expertise [26]. Given the proposed RED-PID as the AQM controller in this paper, the PID component naturally has the merits of self-correction and history-experience tracking. So, we believe the heuristic algorithm suitable for RED-PID should prioritize responsiveness and execution efficiency. Additionally, since no specific algorithms have been designed based on the fluid model of the TCP/AQM feedback system, this paper designs a new heuristic algorithm, AOMOA, following the Aquila and Math optimization optimizers [3, 5] for optimizing AQM mechanisms in network communication.

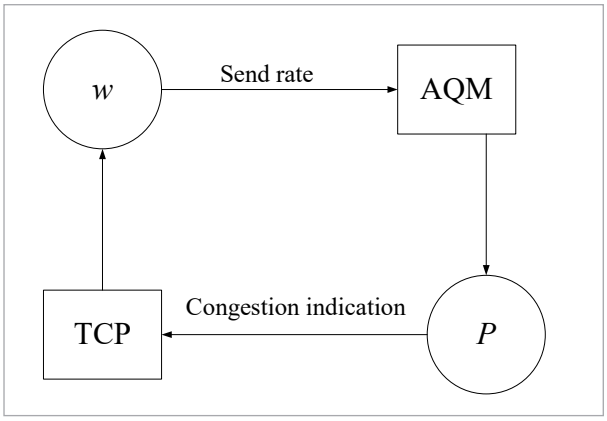
3. TCP/RED-PID

In this section, we determine the constrained correlation among the RED-PID parameters via stability and frequency analysis of the TCP/RED-PID fluid model. The parameters are subsequently characterized by both equality and inequality constraint conditions, forming an optimization model that AOMOA would address.

3.1. Fluid Model

In the TCP/AQM mechanism, AQM drops packets in the queue of the intermediate device with some probabilities. This means that an intermediate device signals congestion to the TCP receiver. The receiver, afterward, acknowledges the congestion to the TCP sender, providing feedback that adjusts the sender's congestion window (*cwnd*) to change the transmitting rate. The TCP/AQM mechanism is shown in Figure 1.

Figure 1
TCP/AQM working mechanism



The TCP/AQM dynamics involve various network parameters that affect the communication performance of TCP. The dynamics are represented as the differential equations [36], which are shown below.

$$\begin{cases} \dot{w}_s(t) = \dfrac{\gamma n}{r(t)} - \dfrac{w_s(t)}{2}\dfrac{w_s(t-r(t))}{nr(t-r(t))}p(t-r(t)) \\ \dot{d}(t) = \dfrac{1}{r(t)c}w_s(t) - 1 \end{cases} \quad (1)$$

where w_s(t) signifies the number of *cwnd* in all TCP senders at time *t*, *r* is the compensation factor for w_s(t), $r(t)$ denotes the round-trip time, *n* represents the number of TCP flows, and $p(t)$ represents the probability of the AQM controller dropping or marking the packets in the queue at time *t*. In Equation (1), $d(t)$ corresponds to the queuing delay at the time *t*, c is the capacity of the bottleneck link connecting directly to the intermediate device.

Suppose the operation point is (w_0, d_0, p_0) in which the TCP/AQM system is in convergence, i.e., the differentials of $w(t)$ and $d(t)$ are both equal to zero. Therefore, we can get

$$\begin{aligned} &\dot{w} = 0 \Rightarrow {w_0}^2 p_0 = 2\gamma n^2 \\ &\dot{d} = 0 \Rightarrow w_0 = r_0 c \end{aligned} . \quad (2)$$

The linearization and Laplace reformation of Equation (1) at the operation point can be deduced, as shown below.

$$\begin{cases} s\delta w(s) = \dfrac{\partial f}{\partial w}\delta w(s) + \dfrac{\partial f}{\partial w_r}e^{-sr_0}\delta w(s) \\ \quad + \dfrac{\partial f}{\partial p_r}e^{-sr_0}\delta p(s) + \dfrac{\partial f}{\partial d}\delta d(s) \\ \quad + \dfrac{\partial f}{\partial d_r}e^{-sr_0}\delta d(s) \\ \quad = (-\dfrac{\gamma n}{r_0^{\,2}c} - \dfrac{\gamma n}{r_0^{\,2}c}e^{-sr_0})\delta w(s) \\ \quad - \dfrac{r_0^{\,2}c}{2n}\delta p(s)e^{-sr_0} \\ \quad - \dfrac{\gamma n}{r_0^{\,2}}(1-e^{-sr_0})\delta d(s) \\ s\delta d(s) = \dfrac{1}{r_0 c}\delta w(s) - \dfrac{w_0}{r_0^{\,2}c}\delta d(s) \\ \quad = \dfrac{1}{r_0 c}\delta w(s) - \dfrac{1}{r_0}\delta d(s) \end{cases} \quad (3)$$

Let e^{-sr0}, which is the parameter of $\delta w(s)$ and $\delta d(s)$, be approximately equal to 1 because r_0 generally is very small and is located on the forward path where the output mainly consists of the linear parts. Finally, Equation (3), incorporating Equation (2) can be simplified as

$$\begin{cases} s\delta w(s) = -\dfrac{2\gamma n}{r_0^{\,2}c}\delta w(s) - \dfrac{r_0^{\,2}c}{2n}\delta p(s)e^{-sr_0} \\ s\delta d(s) = \dfrac{1}{r_0 c}\delta w(s) - \dfrac{1}{r_0}\delta d(s) \end{cases}. \quad (4)$$

Let $N(\cdot)= \delta q(s)/\delta p(s)$ denotes the transfer function of a specific AQM controller. In this paper, it represents the transfer function of the RED controller [18], which is represented as

$$N^{RED}(\cdot) = \frac{\delta q(s)}{\delta p(s)}\frac{p_{\max}}{l_{\max} - l_{\min}}\frac{k}{j\omega + k}. \quad (5)$$

Accordingly, the simplified TCP/RED block diagram is shown in Figure 2.

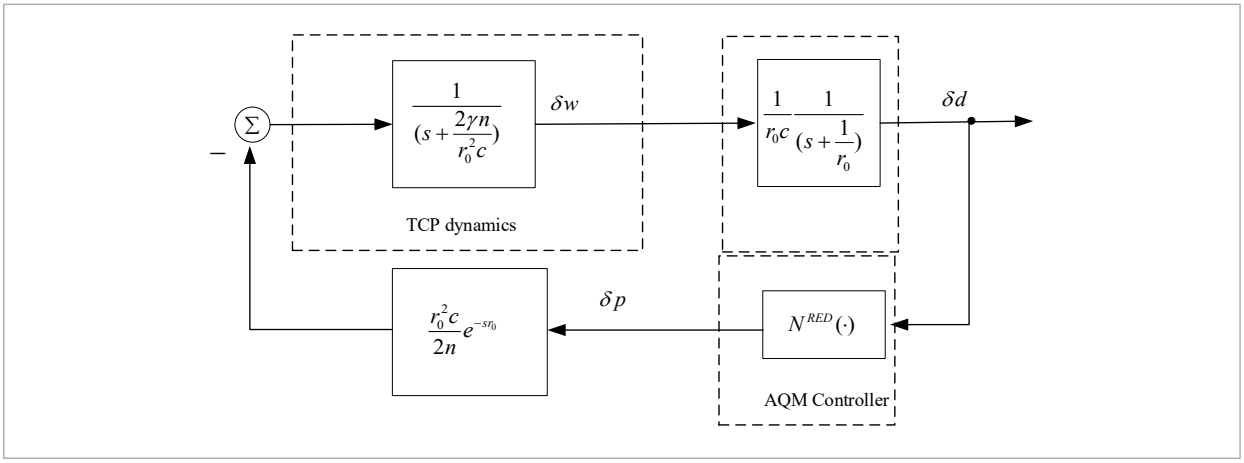
According to Figure 2, the TCP/RED dynamic model forms a negative-feedback system. The open-loop transfer function of TCP/RED is expressed in the tail-one-canonical form, which is

$$L(s) = \frac{r_0}{2n}\frac{e^{-sr_0}}{\left(s + \dfrac{2n\gamma}{r_0^{\,2}c}\right)\left(s + \dfrac{1}{r_0}\right)}N^{RED}(\cdot). \quad (6)$$

According to the Maclaurin expansion for e^{-sr0}, Equation (6) is expressed approximately by

$$L(s) = \frac{1}{2n}\frac{1}{\left(s + \dfrac{2n\gamma}{r_0^{\,2}c}\right)\left(s + \dfrac{1}{r_0}\right)^2}N^{RED}(\cdot). \quad (7)$$

Figure 2
TCP/RED dynamic block diagram



Therefore, the open-loop transfer function of the TCP/RED feedback system, in frequency domain, is

$$L(j\omega)=\frac{1}{2n}\frac{1}{\left(j\omega+\frac{2n\gamma}{r_0^2c}\right)\left(j\omega+\frac{1}{r_0}\right)^2}\cdot\frac{p_{\max}}{l_{\max}-l_{\min}}\frac{k}{j\omega+k}. \quad (8)$$

In Equation (8), the feedback system is of high order, and the three poles' absolute value ($2n\gamma/r_0^2c$, $1/r_0$, k) are typically large. Consequently, the numerator is significantly smaller than the denominator. This results in the very small magnitude-frequency characteristic, which in turn lags the response time and convergence.

To overcome the shortage inherited from the RED controller, we add the PID adjustor to the TCP/RED system to improve the system performance. Equation (9) depicts the transfer function and corresponding phase angle for the PID adjustor in frequency domain.

$$\begin{cases} N^{PID}(j\omega)=\dfrac{K_D(j\omega)^2+K_Pj\omega+K_I}{j\omega} \\ \angle N^{PID}(j\omega)=\arctan\dfrac{K_P}{K_I-K_D\omega^2}-90^\circ \end{cases}. \quad (9)$$

The transfer function of RED-PID is

$$N^{RED-PID}(j\omega)=N^{RED}\cdot N^{PID}=\frac{p_{\max}}{l_{\max}-l_{\min}}\frac{k}{j\omega+k}\times\frac{K_D(j\omega)^2+K_Pj\omega+K_I}{j\omega}. \quad (10)$$

At last, the open-loop transfer function of TCP/RED-PID derived from Equation (8) is

$$\begin{aligned} L^{TCP/RED-PID}(j\omega) &= \frac{1}{2n}\frac{1}{\left(j\omega+\frac{2n\gamma}{r_0^2c}\right)\left(j\omega+\frac{1}{r_0}\right)^2}N^{RED-PID}(j\omega) \\ &= \frac{1}{2n}\frac{p_{\max}kK_D((j\omega)^2+\frac{K_P}{K_D}j\omega+\frac{K_I}{K_D})}{\left(j\omega+\frac{2n\gamma}{r_0^2c}\right)\left(j\omega+\frac{1}{r_0}\right)^2} \end{aligned}. \quad (11)$$

Because PID has double zero points, these zero points can counteract the pole points in Equation (11), thereby reducing the order level of the denominator of the TCP/RED system. Meanwhile, PID introduces one pole point to enhance the system sensitivity. The two improvement approaches contribute to enhancing the system performance, thus establishing the RED-PID feedback system.

As seen from the above equations, various parameters constrain the performance and stability of the RED-PID system. The following section will give the optimization model for these parameters.

3.2. Optimization Model

According to the feedback theory, a high-performance feedback system must maintain stability while simultaneously ensuring a suitable margin of its phase angle. The approach counteracting the zero and pole points are adopted to simplify the RED-PID optimization. The two pole points ($-2n\gamma/r_0^2c$, and $-k$) and the two zero points in the numerator of $N^{PID}(j\omega)$ are counteracted, i.e., $(j\omega+2n\gamma/r_0^2c)(j\omega+k)$ is equal to $((j\omega)^2+K_P\cdot j\omega/K_D+K_I/K_D)$ in Equation (11). To meet the criteria of zero-pole cancelation, $K_P/K_D=2n\gamma/r_0^2c+k$ and $K_I/K_D=2n\gamma k/r_0^2c$.

As a result, after zero-pole point cancelation, Equation (11) is furtherly deduced to

$$L^{RED-PID}(j\omega)=\frac{kK_D}{2n}\frac{p_{\max}}{l_{\max}-l_{\min}}\frac{1}{j\omega\left(j\omega+\frac{1}{r_0}\right)^2}. \quad (12)$$

According to the Nyquist criterion, we set the phase angle margin to -40° when the amplitude of Equation (12) equals 1. The angle margin (-40°) is chosen to balance responsive time and vibration overshoot. Therefore, under the condition of specific values for c, n, and r_0, the optimization problem of RED-PID is defined as follows.

$$Assume\ X=[\omega_c,k,P_{\max},l_{\max},l_{\min},\gamma,K_P,K_D,K_I], \quad (13)$$

$$\begin{aligned} \min f(X) &= \left|angle(L^{RED-PID}(j\omega))+140^\circ\right| \\ &= \left|-\arctan\frac{\omega r_0^2c}{2n\gamma}-2\arctan\omega r_0-\arctan\frac{\omega}{k}-\arctan\frac{K_P}{K_I-K_D\omega^2}+50^\circ\right| \end{aligned} \quad (14)$$

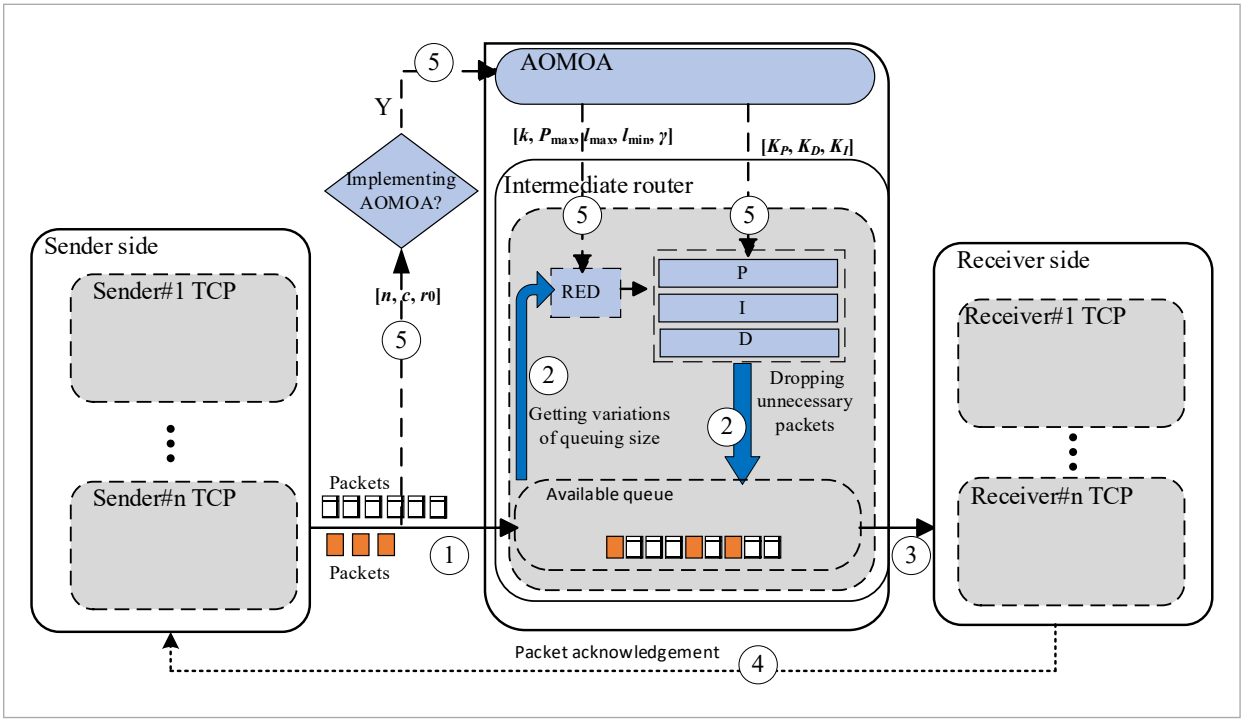
s.t.

$$
\begin{aligned}
&\left|L(j\omega_c)\right| = 1 \\
&0 < \omega \le \omega_c \\
&-180° < angle(L(j\omega_c)) < angle(L(j\omega)) < 0° \\
&k = c \cdot \ln(1-\delta) \quad \delta \in [0.05, 0.95] \\
&\frac{K_P}{K_D} = \frac{2n\gamma}{r_0^2 c} + k \\
&\frac{K_I}{K_D} = \frac{2n\gamma}{r_0^2 c} \\
&\left|L(j\omega_c)\right| = 1
\end{aligned}
\quad . \tag{15}
$$

Since the phase angle margin of TCP/RED-PID is -40°, $angle(L^{RED\text{-}PID}(j\omega))$ = -140°. Therefore, the optimization objective represented in (14) is most close to zero. In (15), ω_c is the cross-frequency, and the amplitude of $L^{RED\text{-}PID}(j\omega_c)$ is equal to 1 according to the control theory. Thus, $L^{RED\text{-}PID}(j\omega_c)$ is an important constraint condition. Similarly, based on the stable criterion of the feedback system, $angle(L^{RED\text{-}PID}(j\omega))$ is in the range of 0° and 180°. Because RED uses the Exponentially Weighted Moving Average (EWMA) filter to resist variations of the queue size, the parameter k represents the weight of EWMA in the Laplace form and is defined in [18]. In this paper, we use the AOMOA algorithm, which optimizes RED-PID and become one part of the TCP/RED-PID system. We give the workflow diagram of TCP/RED-PID with AOMOA, as shown in Figure 3.

The TCP sender side, the TCP receiver side, and the intermediate side construct the TCP/RED-PID feedback system. Firstly, nodes on the sender side transmit packets. The packets enter the router's receiving buffer and form a queue. Secondly, according to a particular policy, the RED-PID controller drops unnecessary packets. Thirdly, the packets from the queue arrive at nodes on the receiver side. Fourthly, the receivers acknowledge the received packet to the sender, which forms a feedback path. Finally, if the RED-PID fails to adjust the queue, and thus congestion occurs, AOMOA optimizes and updates the RED-PID parameters. The feedback system becomes stable and recovers good performance again because the new parameters adapt to the latest communication environment.

Figure 3
Workflow diagram of TCP/RED-PID and AOMOA



4. AO Optimizer

The Aquila Optimizer is a heuristic algorithm that simulates the hunting behavior of Aquila eagles capturing squirrels and is employed to solve optimization problems. It has been successfully applied to various engineering problems, including image classification [8, 12, 34], vehicle route planning [6, 19-20], risk prediction [30-31], and machine learning hyperparameter optimization [22]. Considering its universality, AO will be introduced briefly in this section.

4.1. Search Phases

The goal of AO is to quickly search for the optimal solution by simulating the hunting behavior of the Aquila eagles. Each individual in AO's population represents a possible solution.

According to hunting of Aquila eagles, AO can be represented in the following four phases. (1) The phase of expanded exploration. An eagle flies at high altitudes to ensure the hunting area. That is, one individual preliminarily searches the space of solutions. (2) The phase of narrowed exploration. Once one eagle finds the prey, it will circle the target to reduce the searching area. This phase, in essence, contracts the area of solutions, improving the convergence of AO. (3) The phase of expanded exploitation. In nature, the Aquila eagle circles the prey at a low altitude. This phase contributes to the eagle locating the center of the reduced area where the prey is spotted. (4) The phase of narrowed exploitation. After experiencing the above three phases, the hunting eagle has been much closer to the prey. Finally, it accurately attacks the target in this last phase.

4.2. Mathematical Model

AO algorithm emulates the hunting behaviors of Aquila eagles. The phase of expanded exploration is modeled and can be expressed as follow.

$$\boldsymbol{x}(t+1) = \boldsymbol{x}^{best}(t) \times \left(1 - \frac{t}{\mathrm{T}}\right) + \left(\boldsymbol{x}^{m}(t) - \boldsymbol{x}^{best}(t) \times rand\right), \tag{16}$$

in which $x(t+1)$ is the solution vector for the next iteration $(t+1)^{\text{th}}$. x_t^{best} and x_t^m are the optimal and mean solutions at the current iteration t^{th}, respectively. In addition, the symbol T denotes the total number of iterations, and the variable *rand* is a random value within the interval (0, 1). At the onset of AO running, x_t^{best} holds a larger proportion, leading individuals in the subsequent iterations to converge toward the current best value. This search way enhances the efficiency of pinpointing the optimal value in an unknown space.

During the phase of narrowed exploration, the mathematical model is presented as

$$\boldsymbol{x}(t+1) = \boldsymbol{x}^{best}(t) \times Levy(D) + \boldsymbol{x}^{r}(t) + (y - x) \times rand, \tag{17}$$

where $Levy(D)$ is the function of *Levy* Flight Distribution [19]. $x^r(t)$ is a random solution at the iteration tth. The algebraic expression (y - x) denotes the difference in (x, y) coordinates of a circle radius. The circle is the site of the narrowed exploration for finding a solution.

After determining the reduced searching site, AO is ready to exploit it and approach the optimal solution in the expanded exploitation. The equation of this phase is

$$\boldsymbol{x}(t+1) = (\boldsymbol{x}^{best}(t) - \boldsymbol{x}^{m}(t)) \times \alpha - rand + ((\boldsymbol{ub} - \boldsymbol{lb}) \times rand + \boldsymbol{lb}) \times \delta. \tag{18}$$

In Equation (18), the ***ub*** and ***lb*** are the upper and lower bounds of the problem under consideration, respectively. The symbols α and δ, denoting the adjustment parameters, fall within the range of (0, 1).

In the fourth phase, AO aggressively grabs the optimal solution in the final location while considering the stochastic movements of itself and the target. The mathematical model in this phase is

$$\boldsymbol{x}(t+1) = Q(t) \times \boldsymbol{x}^{best}(t) - g_1 \times \boldsymbol{x}(t) \times rand - g_2 \times Levy(D) + g_1 \times rand, \tag{19}$$

where Q(t) is the quality function to affect the transfer of the searching phases, the function that is denoted by

$$Q(t) = t^{\frac{2 \times rand - 1}{(1 - \mathrm{T})^2}}. \tag{20}$$

The g_1 is the parameter representing the random movements of the target or solution, and the g_2 denotes linear variation of each individual's start and end positions. The expressions of the two parameters are

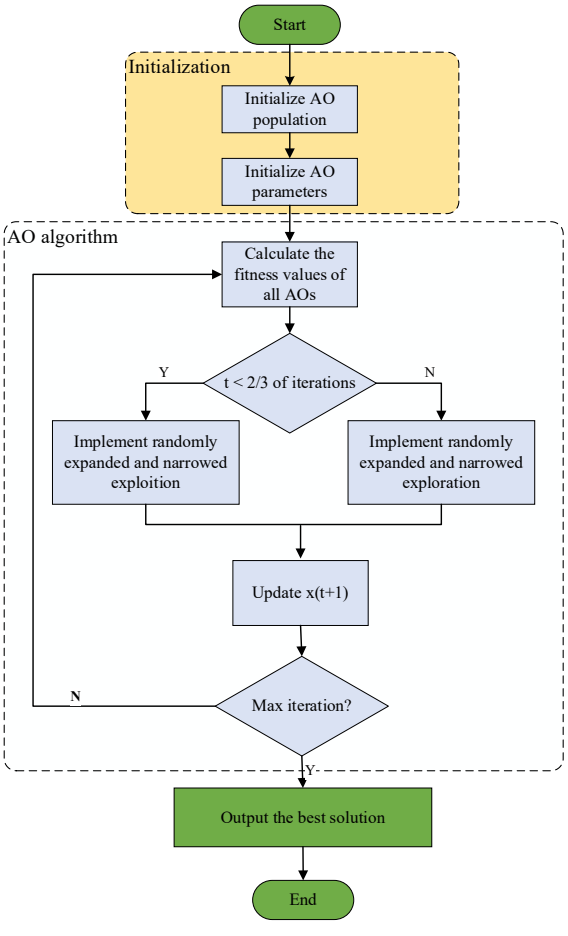
$$g_1 = 2 \times rand - 1, \tag{21}$$

$$g_2 = 2 \times (1 - t / \mathrm{T}). \tag{22}$$

To explicate the process of AO, the flowchart of AO is shown in Figure 4.

Given that the current optimal solution, $\boldsymbol{x}^{best}(t)$, directly engages with exploration phases, AO quickly searches for the next-iteration optimal solution and converges towards it. For the same reason, the individuals in AO easily fall into the local optimal points. That is, AO is insufficient to deal with the optimization of TCP/RED-PID. Therefore, we introduce a novel search algorithm in the following section.

Figure 4
Flowchart of AO



5. AOMOA Algorithm

Although AO exhibits rapid convergence, which enables the exploration of global optimal solutions, its individuals easily fall into suboptimal solutions. Considering that MO is more refined than AO in exploitation phases, this paper employs AOMOA, a hybrid algorithm of Aquila and Math optimizers, to struggle to evade local optima. Additionally, AOMOA incorporates the chaotic-subpopulation strategy and the *k*-worst shift of individuals between dual subpopulations to enhance its whole performance.

5.1. Chaotic-subpopulation Strategy

Prabakeran et al. [29] highlight the crucial role of the initial population's quality in influencing the efficiency and accuracy of meta-heuristic algorithms. In the case of AO, the reliance on Random Number Generation for initial population might result in uneven distribution and inadequate diversification of individuals. To address the problem, AOMOA uses the Tent chaotic map to generate individuals in the two subpopulations. This design considers that chaotic maps have better ergodicity and unpredictability than random numbers, and Tent, defined by Equation (23), stands out as a renowned and widely used chaos map.

$$x_{t+1} = \begin{cases} \dfrac{x_t}{\lambda} & 0 < x_t \le \lambda \\ \dfrac{1 - x_t}{1 - \lambda} & \lambda < x_t < 1 \end{cases}. \tag{23}$$

The strategy of dual subpopulations incorporates the exploration of AO and the hybrid exploitation of AO and MO in each subpopulation, forming the AO subpopulation and the hybrid subpopulation. This strategy leverages the advantages of the two optimizers. In detail, the strategy, on one hand, advantages the powerful convergence of the exploration searches of AO; on the other, it benefits from the enhanced precision of local searches in MO through the addition and subtraction searches. The exploitation of MO is represented as

$$\boldsymbol{x}_{t+1} = \begin{cases} \boldsymbol{x}_t^{best} - P_t^{mo} \\ \times((\boldsymbol{ub} - \boldsymbol{lb}) \times u + \boldsymbol{lb}), \quad rand < 0.5 \\ \boldsymbol{x}_t^{best} + P_t^{mo} \\ \times((\boldsymbol{ub} - \boldsymbol{lb}) \times u + \boldsymbol{lb}), \quad otherwise \end{cases}, \tag{24}$$

$$P_t^{mo} = (1 - \frac{t}{\mathrm{T}})^{\frac{1}{a}}, \quad (25)$$

where, *rand* is a random number, and u is the parameter that is a constant equal to 0.499 and controls the step length of exploitation phases. In Equation (25), P^{mo} is the probability of MO, and $\alpha = 5$.

In the hybrid subpopulation set, a chaotic acceleration function, $A^{mo}(t)$, is first defined to switch exploitation methods between MO and AO optimizers. The function is

$$A^{mo}(t) = 2A^{mo}(0)(1 - \frac{t}{\mathrm{T}}) + \omega_{\max} - (\omega_{\max} - \omega_{\min})\frac{t}{\mathrm{T}} x_t . \quad (26)$$

In Equation (26), $\omega_{\max}$ and $\omega_{\min}$ are equal to 0.9 and 0.5 respectively. x_t is a chaotic number in Equation (23).

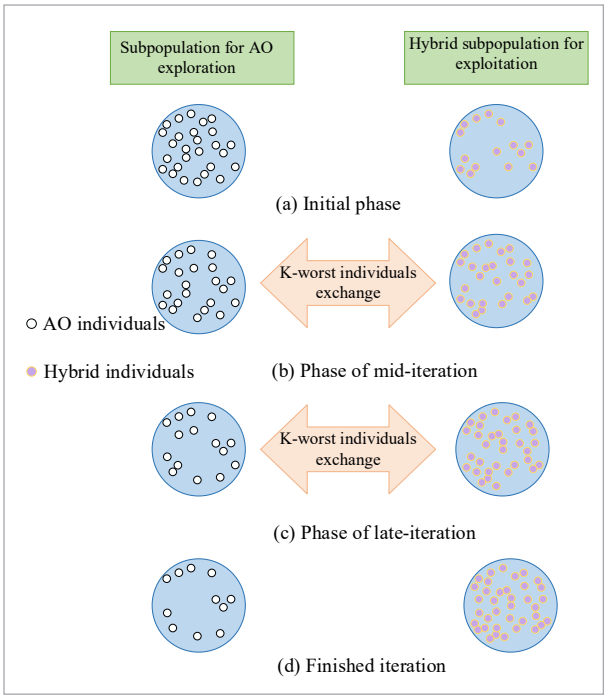
During the hybrid exploitation, each individual selects the exploitation searches of AO or MO according to Equation (26). If an individual opts for MO subtraction or addition searches for the current solution, the method could shorten the length of exploiting steps, thereby contributing to finding the optimal solution. On the other hand, if an individual implements the AO's exploitation, this method could responsibly navigate the space of solutions.

5.2. Dynamic k-worst Shift of Individuals

Since AO regularly shifts its exploration to the exploitation phase at 2/3 of the total iterations, this rigid setting deteriorates the performance of AO in exploiting solutions and makes AO fall into suboptimal conditions. Therefore, we propose a k-worst shift strategy between the dual subpopulations. The strategy is illustrated in Figure 5.

The *k*-worst shift strategy includes four phases. In the first initial phase, the number of individuals for AO exploration is more than that of the hybrid subpopulation for exploitation. This design focuses on contracting the resolution spaces. With AOMOA iteration, the focus should transfer to searching unknown resolution spaces. That is, the algorithm tends to find the resolution as quickly as possible. Guided by the design idea, the uneven number of individuals in the two subpopulations exchange places with each other in the mid- and late-iteration phases. More individuals enter the hybrid sub-population from the AO sub-population. However, only a few individuals

Figure 5
Schematic diagram of k-worst shift of individuals in dual subpopulations



in the hybrid population transfer to the AO subpopulation. The trend will continue until the iteration of AOMOA finishes its iteration.

In the shift strategy, AO conducts the exploration searches in one subpopulation, while the *k*-worst individuals shift to the hybrid subpopulation to engage in exploitation. The change in proportion emphasizes the role of exploitation in the later phase of iterations. To realize the proportion of individuals executing exploration to decrease as the iterations evolve, the decay rate of the individuals in the AO subpopulation is defined as

$$D(t) = \exp(-\frac{t}{\mathrm{T}}) . \quad (27)$$

Supposed the initial individuals of the AO subpopulation is termed as initial AO, the number of individuals of the AO subpopulation at current iteration, N(t), is

$$N(t) = D(t) \times initial\_AO . \quad (28)$$

From Equation (28), the individuals involved in AO exploitation are reduced in a negative-exponential manner. Meanwhile, the same number of worst-performing

individuals from the hybrid population migrate to the AO subpopulation to ensure that the number of AO individuals remains equal to N(t). Consequently, the AO subpopulation consists of smaller individuals, while the hybrid subpopulation comprises more individuals during the later stages of AOMOA execution.

The values of k-best are computed as Equation (29).

$$k\text{-}worst = final\_per \times \frac{t}{\mathrm{T}} \times N(t) + (1 - final\_per) \times \xi_t \times N(t) \quad (29)$$

In which, *final_per* is the percentage of the population to shift from the exploration phase towards the exploiting phase, and is set at 10%. The symbol ξ_t is the chaotic number of Logistic Mapping, which is defined as

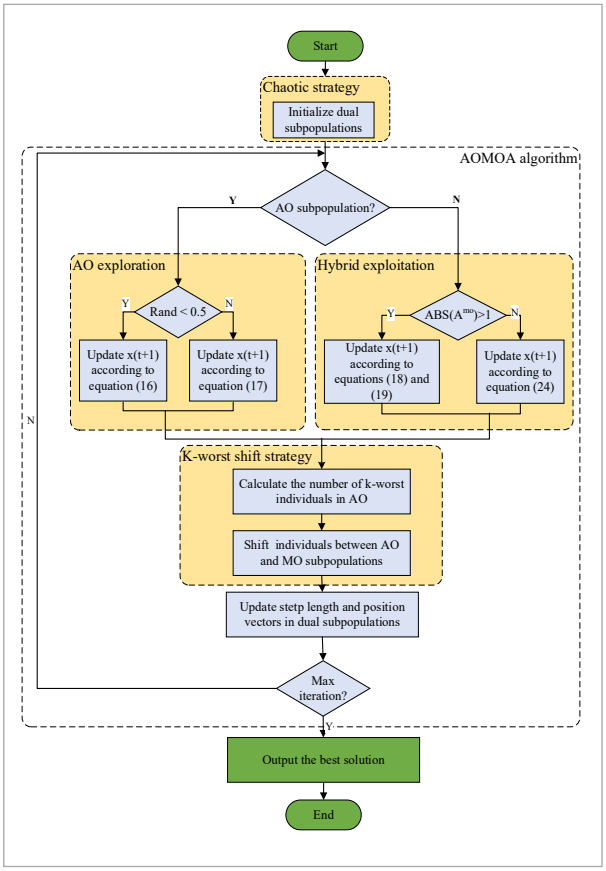
$$\xi_t = u \times \xi_{t-1} \times (1 - \xi_{t-1}), \quad (30)$$

where, u is termed as biotic potential constant being equal to 4.

In summary, the dynamic k-worst shift strategy shifts more individuals derived from the chaotic map to search for the best solution elaborately, consequently minimizing the risk of premature convergence. Figure 6 details the flowchart of AOMOA.

In addition, Algorithm 1 provides the pseudo-code of AOMOA.

Figure 6
Flowchart of AOMOA



ALGORITHM 1 AOMOA

Input:
AO subpopulation size N_1; Hybrid subpopulation size N_2; Maximum iterations MaxIter; Flow number n; Trunk link bandwidth c; Expected delay r_o

Output:
Optimized nine-dimension parameters [ω_c, k, P_{max}, l_{max}, l_{min}, γ, K_P, K_D, K_I]
/ refer to Equation (8)-(11) for the parameter meaning */*

1. Chaotically initialize AO subpopulation S_1 with N_1 individuals;
2. Chaotically initialize hybrid subpopulation S_2 with N_2 individuals;
3. Evaluate fitness of each individual in {S_1, S_2};
4. Set the best solution G_best from {S_1, S_2};
5. **FOR** *iter* = 1 to MaxIter **DO**
6. **FOR** each individual *i* in {S_1, S_2} **DO**
7. Update position using Aquila Optimizer (AO) strategy if *i* in S_1;
/ refer to Equation (16)-(17) */*
8. Update position using Aquila Optimizer (AO) strategy if *i* in S_2;
/ refer to Equation (18), (19) and (24) */*
9. Evaluate new fitness of individual *i*; */*refer to Equation (14)-(15)*/*
10. **END FOR**
11. Identify the *k* worst-performing individuals from S_1 and S_2;
12. Swap the positions of these *k* individuals between S1 and S2;
13. **FOR** each subpopulation S in {S1, S2} **DO**
14. **FOR** each individual *j* in S **DO**
15. Update position according to the step length;
16. **END FOR**
17. **END FOR**
18. Update G_best with the best solution found in current iteration;
19. **END FOR**
20. Return G_best as the optimized parameters;

According to the above pseudo codes, we now analyze the time complexity of AOMOA. The time complexity depends on various steps performed during each iteration and the number of iterations needed for convergence. In the subpopulation initializing step, the time complexity is $O((N_1 + N_2)\cdot D)$ due to the two subpopulations. For each individual, the evaluation cost is denoted as $O(E)$. Therefore, the fitness evaluation complexity is $O((N_1 + N_2)\cdot E)$. As shown in Figure 6, the dimension of search methods, denoted as k, equals 2. So, the complexity of selecting a specific search is $O(2\cdot(N_1 + N_2))$. As to updating the AO exploration and the Hybrid exploitation, the complexity is $O((N_1 + N_2)\cdot D)$. Because the k-worst shift strategy affects only k individuals, the complexity of this step is $O(k)$. To sum up, the total time complexity for each iteration is approximately $O((N_1 + N_2)\cdot E)+ O((N_1 + N_2)\cdot D)+ O(2\cdot(N_1 + N_2))$. Since the algorithm runs for T iterations, the overall time complexity of AOMOA is $O(T(N_1+N_2)\cdot(D+E+2))$ if we ignore the initial phase. On the condition of the same total individuals ($N=N_1+N_2$), the overall time complexity of AO and MO is equal to $O(T\cdot N\cdot(D+E))$ and $O(T\cdot N\cdot(D+E+2))$, respectively. As a result, the complexity of AOMOA is equal to that of MO and is a little more that of AO. The promotion in complexity is not significant.

6. Results and Evaluation

In this section, the performance of AOMOA is assessed using ten classic benchmark functions sourced from the CEC-2017 test suite [9, 11] because CEC-2017 is widely recognized and frequently employed for performance evaluations. Additionally, extensive simulation experiments are implemented to verify that RED-PID effectively improves congestion control after being optimized by AOMOA.

6.1. Performance Evaluation of AOMOA

The selected benchmark functions offer a comprehensive basis for evaluating the effectiveness of various heuristic algorithms. These benchmark functions include the simple function set (F1, F2, F5, and F8), the hybrid function set (F12, F15, F16, F17 and F19), and the composited function set (F21 and F23). In the performance test, AOMOA and the other nine state-of-the-art heuristic algorithms run 50 times to solve the benchmark functions, respectively. For each benchmark function, the best result is highlighted in boldface. The symbol "==" denotes the non-significant difference in the value compared to the best solution. On the contrary, the significant difference is indicated by "++".

Table 1 shows that AOMOA, compared to other algorithms, achieves competitive results for these benchmark functions with 30 dimensions. AOMOA gets the best performance in the test of the simple function set. Additionally, AOMOA approaches the best value in the hybrid function set. Although AOMOA obtains an inferior optimal solution than DevBBO and DevSMA, the significant difference is not apparent at a significance level of 0.05. The test results confirm the overall best performance of AOMOA.

Table 2 presents the outcomes of various algorithms for solving the optimal solution of the 50-dimension benchmark functions. Similarly to the results of 30-dimension tests, AOMOA continues to demonstrate the best performance in the simple function set. Notably, while none dominates over other algorithms as the dimensionality of the hybrid and composited function sets increases, AOMOA consistently performs better, followed by MFO and BaseGA.

From the performance evaluation, we can see that AOMOA simultaneously searches the optimal resolution in two different spaces where the exploration and exploitation modes of AO and MO are running, respectively. The hybrid modes help AOMOA converge the optimal states because they leverage the strength of the two kinds of search methods. On the other hand, individuals in the two subpopulations disproportionately swap each search method and space to help the proposed algorithm search unknown resolution spaces, which improves the accuracy. The superior performance of AOMOA shows that its chaotic subpopulations and the dynamic k-worst shift of individuals significantly improve its exploration and exploitation capabilities.

6.2. Simulation Topology

We utilize a testbed created by the NS3 simulator to construct a typical parking-lot topology, the topology that is popular in real network communication and contains various flows with different workloads. In this testbed, as shown in Figure 7, communication

Table 1
Test results of 10 benchmark functions with 30 dimensions

Functions	Index	DevBBO	OriginalDE	DevSMA	GWO	AOMOA
F1	Best	5.9340E-01	3.1699E+00	1.9025E-01	6.5416E-01	4.7559E-01
	Mean	6.3613E+00	1.8524E+01	6.3693E+00	1.2538E+01	3.6716E+00
	Std	2.4242E+00	1.6588E+01	2.8078E+00	9.3639E+00	1.3679E+00
	z value	1.9663E+00	1.0858E+01	1.9721E+00	6.4816E+00	--
	p value	2.4629E-02	0.0000E+00	2.4296E-02	4.5375E-11	--
	Significance	++	++	++	++	--
F2	Best	1.9917E+01	1.1985E+01	5.3449E+00	6.9525E+01	1.9899E+00
	Mean	3.8558E+01	1.1034E+02	1.9682E+01	1.0760E+02	8.9879E+00
	Std	1.1656E+01	1.3550E+02	8.0884E+00	1.6098E+01	4.8709E+00
	z value	6.0708E+00	2.0809E+01	2.1955E+00	2.0245E+01	--
	p value	6.3620E-10	0.0000E+00	1.4062E-02	0.0000E+00	--
	significance	++	++	++	++	--
F5	Best	4.4489E+01	1.0364E+02	1.8032E+01	5.2971E+02	1.3291E+01
	Mean	8.8403E+01	3.0021E+03	3.0219E+01	1.4425E+03	1.9777E+01
	Std	2.1480E+01	5.4648E+03	8.6494E+00	1.0684E+03	4.2828E+00
	z value	1.6024E+01	6.9635E+02	2.4380E+00	3.3219E+02	--
	p value	0.0000E+00	0.0000E+00	7.3843E-03	0.0000E+00	--
	significance	++	++	++	++	--
F8	Best	2.5393E-04	1.5180E-03	1.7528E-01	1.6241E-02	1.4998E-32
	Mean	6.7000E-02	1.7540E-01	4.4751E-01	1.6944E-01	3.1465E-32
	Std	9.0371E-02	2.1386E-01	3.1984E-01	3.0711E-01	5.1502E-32
	z value	1.3009E+30	3.4057E+30	8.6892E+30	3.2900E+30	--
	p value	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	--
	significance	++	++	++	++	--
F12	Best	-4.1600E+03	-4.2993E+03	-4.1323E+03	-2.8541E+03	-4.0864E+03
	Mean	-4.1600E+03	-4.2993E+03	-4.1323E+03	-2.8541E+03	-4.0864E+03
	Std	-3.5198E+03	-3.4930E+03	-3.1012E+03	-1.8250E+03	-3.4693E+03
	z value	3.4394E+02	4.0894E+02	5.1356E+02	4.8420E+02	4.1250E+02
	p value	--	7.7884E-02	1.2171E+00	4.9277E+00	1.4693E-01
	significance	--	4.6896E-01	1.1179E-01	4.1591E-07	4.4159E-01
F15	Best	7.6486E+01	1.5024E+02	5.1892E+00	1.0573E+02	1.0483E+00
	Mean	2.0496E+03	3.0076E+03	4.3516E+01	3.6008E+02	1.6956E+01
	Std	2.1179E+03	2.7222E+03	3.7926E+01	3.7866E+02	1.5658E+01
	z value	1.2982E+02	1.9100E+02	1.6962E+00	2.1914E+01	--
	p value	0.0000E+00	0.0000E+00	4.4919E-02	0.0000E+00	--
	significance	++	++	++	++	--

Table 1 (continuation)

Functions	Index	DevBBO	OriginalDE	DevSMA	GWO	AOMOA
F16	Best	1.3664E+04	7.9616E+02	3.3318E+03	3.6707E+05	1.2603E+03
	Mean	9.2287E+04	2.5812E+06	3.6455E+04	3.2712E+06	4.5342E+04
	Std	1.7606E+05	1.0610E+07	2.0160E+04	1.9898E+06	1.7573E+04
	z value	2.7694E+00	1.2622E+02	--	1.6045E+02	4.4080E-01
	p value	2.8078E-03	0.0000E+00	--	0.0000E+00	3.2968E-01
	significance	++	++	--	++	==
F17	Best	4.1371E+01	2.7107E+01	2.5220E+01	1.7042E+02	2.8830E+01
	Mean	1.0047E+04	1.2505E+03	6.2168E+02	7.3424E+02	5.5794E+02
	Std	8.1457E+03	2.3262E+03	1.4546E+03	1.0381E+03	5.1251E+02
	z value	1.8515E+01	1.3512E+00	1.2436E-01	3.4400E-01	--
	p value	0.0000E+00	8.8315E-02	4.5051E-01	3.6542E-01	--
	significance	++	==	==	==	--
F21	Best	1.9690E+02	1.8825E+03	7.8939E+02	7.7948E+03	1.2536E+03
	Mean	1.0522E+04	1.3629E+05	7.1947E+03	2.7229E+04	7.9089E+03
	Std	8.3130E+03	2.8320E+05	5.7917E+03	2.4994E+04	7.5197E+03
	z value	5.7453E-01	2.2290E+01	--	3.4591E+00	1.2332E-01
	p value	2.8280E-01	0.0000E+00	--	2.7096E-04	4.5093E-01
	significance	==	++	--	++	==
F23	Best	2.4488E+02	5.7171E+00	9.5283E+01	2.1608E+03	1.4516E+01
	Mean	1.7672E+04	6.7703E+03	5.8715E+03	9.6208E+03	2.4911E+02
	Std	1.7061E+04	6.1974E+03	5.9393E+03	6.9277E+03	2.9390E+02
	z value	6.9938E+01	2.6177E+01	2.2570E+01	3.7620E+01	--
	p value	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	--
	significance	++	++	++	++	--

Table 2
Test results of 10 benchmark functions with 50 dimensions

Functions	Index	BaseGA	GaussianSA	AO	MFO	AOMOA
F1	Best	4.9154E+07	8.3139E+10	5.1559E+10	2.7319E+02	6.2365E+00
	Mean	9.1318E+07	1.4974E+11	6.7068E+10	5.6059E+07	3.9434E+03
	Std	2.5697E+07	3.0767E+10	7.4123E+09	2.1490E+08	5.5183E+03
	z value	1.6548E+04	2.7135E+07	1.2154E+07	1.0158E+04	--
	p value	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	--
	Significance	++	++	++	++	--

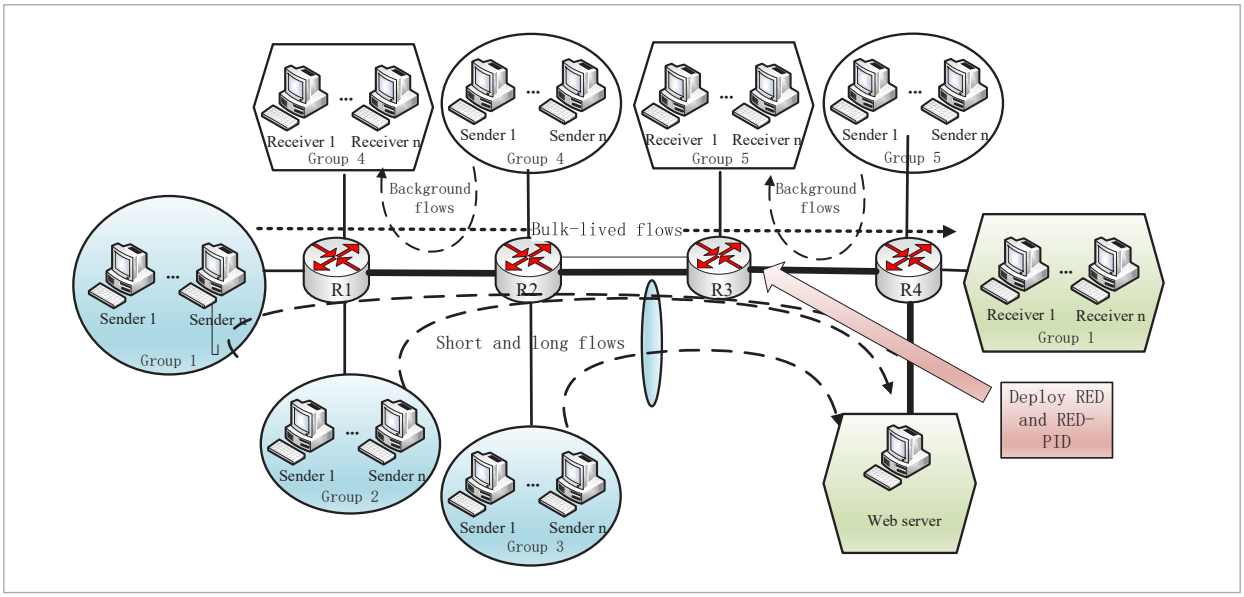
Table 2 (continuation)

Functions	Index	BaseGA	GaussianSA	AO	MFO	AOMOA
F2	Best	1.9348E+03	1.4417E+05	7.8900E+04	1.7467E-09	6.7225E-07
	Mean	7.4550E+03	9.0686E+08	1.1594E+05	1.2253E+03	1.1219E-01
	Std	3.8376E+03	4.5504E+09	2.0913E+04	1.2518E+03	6.0377E-01
	z value	1.2347E+04	1.5020E+09	1.9203E+05	2.0292E+03	--
	p value	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	--
	significance	++	++	++	++	--
F5	Best	3.4202E-05	7.4680E-02	3.4159E-02	2.7224E-06	4.9445E-07
	Mean	2.2123E-04	2.5553E-01	6.1533E-02	4.6078E-04	1.6838E-04
	Std	2.1163E-04	9.0514E-02	1.4592E-02	6.7429E-04	3.9988E-04
	z value	1.3216E-01	6.3860E+02	1.5346E+02	7.3122E-01	--
	p value	4.4743E-01	0.0000E+00	0.0000E+00	2.3232E-01	--
	significance	==	++	++	==	--
F8	Best	3.5875E-01	6.9415E+01	3.2073E+01	1.2668E+00	2.3579E+00
	Mean	2.1499E+00	1.1136E+02	5.1032E+01	4.3668E+00	9.2424E+00
	Std	2.0137E+00	2.8967E+01	7.4501E+00	2.2146E+00	3.9762E+00
	z value	--	5.4235E+01	2.4275E+01	1.1009E+00	3.5222E+00
	p value	--	0.0000E+00	0.0000E+00	1.3546E-01	2.1400E-04
	significance	--	++	++	==	++
F12	Best	5.7126E+04	1.9348E+10	3.1607E+09	1.0089E+04	3.6584E+03
	Mean	1.9594E+05	5.2701E+10	1.6203E+10	4.8125E+05	8.6571E+04
	Std	9.9742E+04	1.7338E+10	6.2515E+09	2.3793E+06	1.2315E+05
	z value	8.8806E-01	4.2794E+05	1.3157E+05	3.2048E+00	--
	p value	1.8725E-01	0.0000E+00	0.0000E+00	6.7568E-04	--
	significance	==	++	++	++	--
F15	Best	7.9654E+01	4.0228E+08	1.1454E+07	3.4126E+01	3.5616E+01
	Mean	5.9981E+05	8.6971E+10	2.3096E+09	3.8640E+03	6.2391E+03
	Std	9.5800E+05	1.2862E+11	2.1697E+09	5.8848E+03	1.7132E+03
	z value	1.0127E+02	1.4779E+07	3.9246E+05	--	4.0361E-01
	p value	0.0000E+00	0.0000E+00	0.0000E+00	--	3.4325E-01
	significance	++	++	++	--	==
F16	Best	5.0599E+02	2.3737E+14	1.0418E+09	9.4436E+02	5.1249E+02
	Mean	1.5850E+03	3.3073E+17	8.7237E+13	7.2520E+03	1.5985E+03
	Std	5.8796E+02	5.9568E+17	2.5859E+14	1.1968E+04	6.6070E+02
	z value	--	5.6251E+14	1.4837E+11	9.6383E+00	2.2964E-02
	p value	--	0.0000E+00	0.0000E+00	0.0000E+00	4.9084E-01
	significance	--	++	++	++	==

Table 2 (continuation)

Functions	Index	BaseGA	GaussianSA	AO	MFO	AOMOA
F17	Best	2.8258E+04	1.4408E+05	1.2807E+05	4.8088E+04	4.4203E+04
	Mean	9.3093E+05	1.5339E+09	4.9844E+06	1.1731E+05	9.4505E+04
	Std	9.6876E+05	1.7527E+09	4.3544E+06	6.2858E+04	3.4035E+04
	z value	2.4575E+01	4.5066E+04	1.4367E+02	6.6994E-01	--
	p value	0.0000E+00	0.0000E+00	0.0000E+00	2.5145E-01	--
	significance	++	++	++	==	--
F21	Best	-3.4219E+02	-1.3015E+02	-3.4220E+02	-3.4220E+02	-3.4220E+02
	Mean	-2.0823E+02	1.2760E+02	-2.7059E+02	-1.3584E+02	-3.2963E+02
	Std	2.0236E+02	1.1666E+02	9.1438E+01	2.2061E+02	6.7682E+01
	z value	1.7937E+00	6.7556E+00	8.7232E-01	2.8633E+00	--
	p value	3.6429E-02	7.1133E-12	1.9152E-01	2.0964E-03	--
	significance	++	++	==	++	--
F23	Best	3.6567E+00	1.0012E+02	9.9420E+00	4.4409E-15	4.4409E-15
	Mean	1.6550E+02	5.6271E+02	1.1060E+02	4.0000E+01	9.6667E+01
	Std	6.3666E+01	4.7299E+02	5.0302E+01	8.0000E+01	1.7951E+01
	z value	1.5688E+00	6.5339E+00	8.8256E-01	--	7.0833E-01
	p value	5.8348E-02	3.2036E-11	1.8874E-01	--	2.3937E-01
	significance	==	++	==	--	==

Figure 7
Parking lot topology



nodes are organized into groups 1 to 3. Each group delivers several TCP streams to the corresponding servers, emulating the widespread Incast communication. In detail, the nodes in Group 1 are divided into two parts: the sender group and the receiver group. Thus, the flows in Group 1 from senders to receivers compose the bulk-lived flows, emulating the workloads of edge computing and file download.

Additionally, Groups 1-3 send flows simultaneously to one web server. The sending mode results in a competition scenario composing multiple data flows on the R1-R4 trunk link. At the same time, Group 4 and Group 5 each consist of 20 nodes and send reverse flows within their respective groups as background flows. All nodes connect to leaf routers via 100 Mbps links. The trunk link's bandwidth is set to 100Mbps and 1Gbps, which corresponds to the slow and fast communication scenarios, respectively. In addition, five bulk-lived flows in Group 4 traverse the trunk link to saturate the available bandwidth. In the downstream of the R1-R4 trunk, Router3 will undertake all flows, and traffic congestion will most likely occur. So, the RED and RED-PID controllers are deployed in Router 3 to control its queue evolution, respectively.

6.3. Results and Analysis

The optimization aims at aligning the phase angle of Equation (12) with -140 degrees when its open-loop amplitude is 1, i.e., minimizes Equation (14). The optimization of RED-PID strikes a balance between congestion avoidance and transmission efficiency by controlling the appropriate queue size in the intermediate device.

In the slow communication scenario, we first set the trunk capacity to 100Mbps, the delay to 30ms, and the number of flows to 900. This configuration simulates the web search workload characterized by slow communication and small latency. Figure 8(a) illustrates the variation in the two queue sizes between RED and RED-PID. Facing concurrent flows, RED makes it challenging to control the increase in queue size, which is very close to the upper limit of 1000 packets. However, RED-PID maintains its queue size near 700 packets throughout the simulation. Compared to the higher queue size of RED, RED-PID limits the queue variation and, therefore, keeps the system stable. In Figure 8(b), the cumulated probability density function (CDF) for 500 short flows (each flow less than 100K) serves as the criteria for evaluating transmission efficiency. RED-PID accomplishes the transmission of 500 short flows in no more than 20s, compared to RED, which takes approximately 23s. The reason is that the stability of RED-PID avoids congestion and, thus, reduces packet losses and retransmissions, ensuring transmission efficiency. Figure 8(c) shows the average sojourn times of packets for RED and RED-PID. Due to the optimization of parameters to enhance the agility of RED-PID, the controller qualifies for low sojourn times. Next, the latency of the trunk link is set to 120ms, and the number of concurrent flows is set to 1200. This case emulates the congestion derived from microburst flows. Like the previous simulation, RED-PID still controls the queue size under 800 packets, as illustrated in Figure 9(a). RED, however, lacking stability in the face of increased latency, has reached the upper limit of the queue size. Figure 9(b) illustrates the compared CDFs for 800 short flows between the

Figure 8
Web search workload

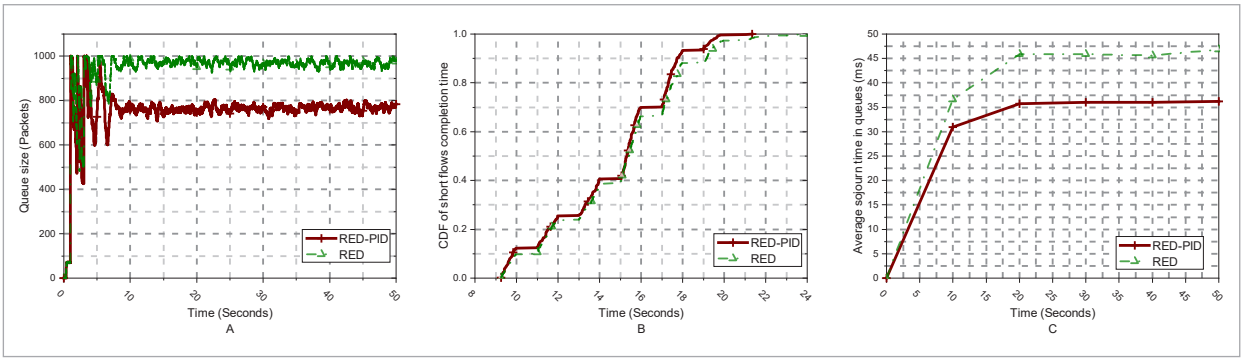


Figure 9
Microburst workload

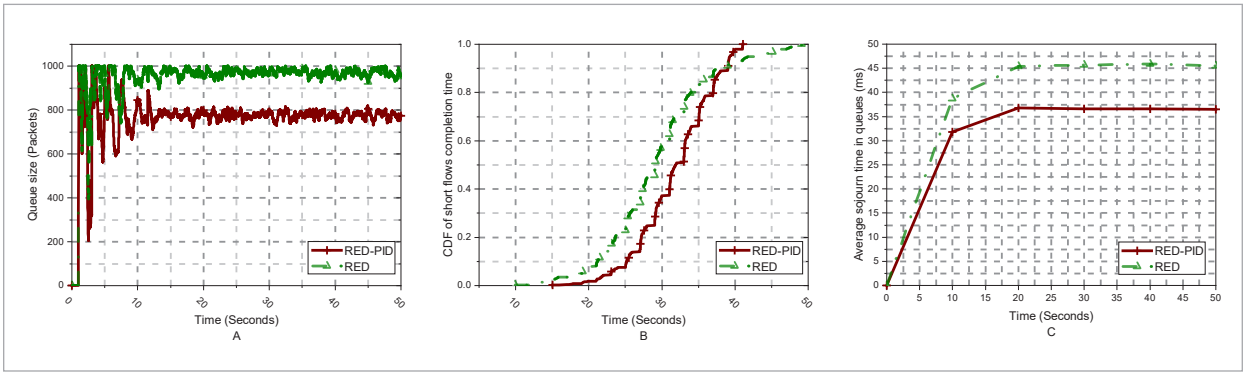
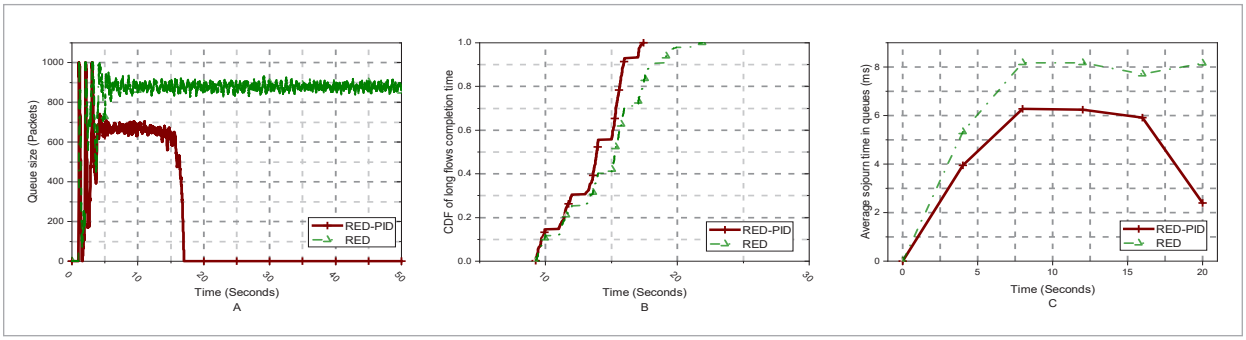


Figure 10
High-speed communication



two controllers. RED-PID outperforms RED in completing the number of microburst flows, although it lags slightly in the first 38s. RED-PID performs better as cumulative microburst flows increase over time. Figure 9(c) depicts the consistently good performance of RED-PID in the average sojourn time.

Finally, to assess the performance of RED-PID in the context of high-speed communication, high capacity of trunk links (1Gbps) and low latency (60ms) are configured, and the number of flows is set to 900. In this scenario, the queue of RED-PID only cumulates in just under 20s, while the queue of RED consistently stays at 800 packets, as shown in Figure 10(a). Considering the communication with high bandwidth, CDF for 400 long flows (each flow size exceeding 100K) evaluates the performance of the two controllers. The results are illustrated in Figure 10(b), wherein RED-PID finishes the long flows faster than RED. This outcome also elucidates that RED-PID can appropriately control the queue evolution and exhibit the shorter sojourn time of packets, as shown in Figure 10(c).

7. Discussion

RED-PID highlighted the RED controller design from the viewpoint that the TCP peer-to-peer sides and the AQM controller construct a feedback system. Previous studies about RED have only focused on adjusting the queue length in intermediated devices because the accumulation of the packet queue may result in buffer overflow and network congestion. However, solely suppressing the queue length overlooks the interaction between peer-to-peer nodes and flows, degrading the RED performance. In this paper, the experiments further confirm this conclusion. This study improves the TCP/RED dynamic model from a feedback system perspective by introducing the PID adjustor. In detail, the proportion component enhances the response of the entire feedback control system, and the integral-differential forms counteract the intrinsic poles that could lower performance in RED. The integral form, meanwhile, utilizes past experience to control

congestion. Ultimately, PID ensures the entire feedback control system stabilizes within an expected bandwidth. The experiment results have also proven the proposed method is correct.

The feedback system of TCP/RED-PID involves many parameters, which profoundly affect its congestion control function. Therefore, which optimization algorithm is used is crucial. In recent years, new heuristic optimizers from animal-based or nature-based phenomena, which combine current algorithms or create adaptable versions [24-25, 29], have been widely applied in various industry areas. Although these heuristic optimizers perform well in some areas, they must often be modified and tailored for specific engineering problems. The best optimizer for solving one problem would not necessarily perform excellently for another issue. As to the TCP/RED-PID feedback system, no existing heuristic algorithms can obtain the most appropriate parameters because analyzing TCP/RED-PID implementation requires relevant expertise. Given that the PID element has the functions of self-correction and experience utilization and that there are no practicable data sets, the optimizer available in this study does not require popular neural networks. The optimization target mainly focuses on improving real-time response and preventing the algorithm from falling into suboptimal solutions. While AO provides rapid convergence by simulating the hunting behavior of Aquila eagles, it is prone to trapping into suboptimal states due to its limited exploitation capabilities. To mitigate this, we incorporate MO, known for its refined exploitation processes, allowing for more accurate optimization during later stages of the algorithm's execution. Additionally, we introduce chaotic-subpopulation and dynamic k-worst shift strategies to maintain the balance between exploration and exploitation throughout the optimization process. This combination not only enhances AOMOA's performance but also makes it more robust in adapting to the dynamic and complex nature of TCP/AQM systems. Of course, AOMOA, in nature, is a heuristic algorithm. It can be utilized in other areas. i.e., inputs of the control system, if it sets some special criteria or control models. The evaluation in the CEC-2017 test suit has demonstrated AOMOA's efficacy compared to nine other popular algorithms. In addition, the network simulation also verifies that the algorithm can better optimize complex RED-PID parameters that improve the controller performance significantly.

In this study, RED-PID is implemented successfully in NS3 simulation, thanks to NS3 providing implementation codes for RED, Codel, and other popular AQM controllers. And these source codes are almost compatible with Linux platforms. We supplement the PID component in the RED source code. To implement the AOMOA algorithm, an open-source and cross-platform Python library for nature-inspired optimization, Mealpy [39], is used. The library provides many classical and state-of-the-art meta-heuristic algorithms, including the AO and MO optimizers. Based on the open-source platform, developers can conveniently design new algorithms.

In future research, we intend to deploy the RED-PID controller in soft-define devices based on the Linux platform, and will use the controller for areas of network congestion control. Since AOMOA is specially designed to optimize the TCP/AQM feedback system, this research could be extended to optimize other AQM controllers in emerging network technologies such as 5G and beyond. The integration of heuristic algorithms like AOMOA into network control systems could pave the way for more adaptive and intelligent network management solutions. Moreover, the insights gained from this study may contribute to advancements in autonomous network systems, where real-time adaptability and optimization are critical.

8. Conclusion

This study aimed to enhance the effectiveness of congestion control in the TCP/RED system by addressing the limitations of the traditional RED controller, particularly its sluggish response to dynamic network conditions. By integrating a Proportional-Integral-Derivative (PID) adjustor into the RED structure, we develop the RED-PID controller, which improves the system's responsiveness and stability. To optimize the performance of the RED-PID controller, we develop a novel heuristic algorithm, AOMOA, which combines the strengths of the Aquila Optimizer and Math Optimizer. Adding chaotic-subpopulation and dynamic k-worst shift strategies further enhances the algorithm's ability to avoid premature conver-

gence and balance global exploration with local exploitation. Through extensive simulations using the NS3 platform, we demonstrated that the RED-PID controller, optimized by AOMOA, significantly outperforms the standard RED controller, providing superior congestion control. This research contributes valuable insights into designing more responsive and stable AQM mechanisms, with implications for improving network performance and robustness.

Acknowledgement

We extend our sincere gratitude to the anonymous reviewers for their insightful comments and constructive suggestions, which have greatly enhanced the quality and clarity of this paper. We deeply appreciate their time and expertise in supporting the advancement of this research. Additionally, this work was supported by the China Scholarship Council [grant number 202108615044].

References

1. Abd Mohammed, Y., Abood, L. H., Kadhim, N. N. Design and Simulation an Optimal Enhanced PI Controller for Congestion Avoidance in TCP/AQM System. Telecommunication Computing Electronics and Control, 2023, 21(5), 997-1004. https://doi.org/10.12928/telkomnika.v21i5.24872
2. Abood, L. H., Oleiwi, B. K., Humaidi, A. J., Al-Qassar, A. A., Al-Obaidi, A. S. M. Design a Robust Controller for Congestion Avoidance in TCP/AQM System. Advances in Engineering Software, 2023, 176, 103395. https://doi.org/10.1016/j.advengsoft.2022.103395
3. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., Gandomi, A. H. The Arithmetic Optimization Algorithm. Computer Methods in Applied Mechanics and Engineering, 2021, 376, 113609. https://doi.org/10.1016/j.cma.2020.113609
4. Abualigah, L., Elaziz, M. A., Sumari, P., Geem, Z. W., Gandomi, A. H. Reptile Search Algorithm (RSA): A Nature-Inspired Meta-Heuristic Optimizer. Expert Systems with Applications, 2022, 191, 116158. https://doi.org/10.1016/j.eswa.2021.116158
5. Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A. A., Gandomi, A. H. Aquila Optimizer: A Novel Meta-Heuristic Optimization Algorithm. Computers and Industrial Engineering, 2021, 157, 107250. https://doi.org/10.1016/j.cie.2021.107250
6. Ait-Saadi, A., Meraihi, Y., Soukane, A., Ramdane-Cherif, A., Gabis, A. B. A Novel Hybrid Chaotic Aquila Optimization Algorithm with Simulated Annealing for Unmanned Aerial Vehicles Path Planning. Computers and Electrical Engineering, 2022, 104, 108461. https://doi.org/10.1016/j.compeleceng.2022.108461
7. Akyol, S. A New Hybrid Method Based on Aquila Optimizer and Tangent Search Algorithm for Global Optimization. Journal of Ambient Intelligence and Humanized Computing, 2023, 14(6), 8045-8065. https://doi.org/10.1007/s12652-022-04347-1
8. Alahmari, S., Yonbawi, S., Racharla, S., Lydia, E. L., Ishak, M. K., Alkahtani, H. K., Aljarbouh, A., Mostafa, S. M. Hybrid Multi-Strategy Aquila Optimization with Deep Learning-Driven Crop Type Classification on Hyperspectral Images. Computer Systems Science and Engineering, 2023, 47(1), 375-391. https://doi.org/10.32604/csse.2023.036362
9. Awad, N. H., Ali, M. Z., Liang, J. J., Qu, B. Y., Suganthan, P. N. Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single-Objective Bound-Constrained Real-Parameter Numerical Optimization. In Technical Report, 2016, 1-34. https://doi.org/10.13140/RG.2.2.32803.84007
10. Bousmaha, R. A Novel Hybrid Aquila Optimizer with Whale Optimization Algorithm for Global Optimization, Feature Selection, and Optimizing SVM Parameters. 4th International Conference on Pattern Analysis and Intelligent Systems, PAIS 2022 - Proceedings. https://doi.org/10.1109/PAIS56586.2022.9946891
11. Cheng, R., Li, M., Tian, Y., Zhang, X., Yang, S., Jin, Y., Yao, X. Benchmark Functions for CEC'2017 Competition on Evolutionary Many-Objective Optimization. Proceedings of IEEE Congress on Evolutionary Computation, 2017, 1-20.
12. Elaziz, M. A., Dahou, A., El-Sappagh, S., Mabrouk, A., Gaber, M. M. AHA-AO: Artificial Hummingbird Algorithm with Aquila Optimization for Efficient Feature Selection in Medical Image Classification. Applied Sciences (Switzerland), 2022, 12(19), 9710. https://doi.org/10.3390/app12199710
13. Giménez, Á., Bonastre, Ó. M., Valero, J., Amigó, J. M. Poster: Modified Dynamic Beta RED-A New AQM Algorithm for Internet Congestion Control. Proceedings of the 2023 ACM on Internet Measurement Conference, 2023, 718-719. https://doi.org/10.1145/3618257.3624996

14. Gomez, J., Kfoury, E. F., Crichigno, J., Srivastava, G. Evaluating TCP BBRv3 Performance in Wired Broadband Networks. Computer Communications, 2024, 222, 198-208. https://doi.org/10.1016/j.comcom.2024.04.037

15. Haile, H., Grinnemo, K.-J., Ferlin, S., Hurtig, P., Brunstrom, A. End-to-End Congestion Control Approaches for High Throughput and Low Delay in 4G/5G Cellular Networks. Computer Networks, 2021, 186, 107692. https://doi.org/10.1016/j.comnet.2020.107692

16. Hamdi, M. M., Mahdi, H. F., Abood, M. S., Mohammed, R. Q., Abbas, A. D., Mohammed, A. H. A Review on Queue Management Algorithms in Large Networks. IOP Conference Series: Materials Science and Engineering, 2021, 1076(1), 12034. https://doi.org/10.1088/1757-899X/1076/1/012034

17. Hassan, S. O., Solanke, O. O., Odule, T. J., Adesina, A. O., Usman, S. A., Ayinde, S. A. AmRED and RED-QE: Redesigning Random Early Detection Algorithm. Telecommunication Systems, 2024, 85(2), 263-275. https://doi.org/10.1007/s11235-023-01082-6

18. Hollot, C. V., Misra, V., Towsley, D., Gong, W. B. On Designing Improved Controllers for AQM Routers Supporting TCP Flows. Proceedings - IEEE INFOCOM, 2001, 3, 1726-1734. https://doi.org/10.1109/INFCOM.2001.916670

19. Houssein, E. H., Saad, M. R., Hashim, F. A., Shaban, H., Hassaballah, M. Lévy Flight Distribution: A New Metaheuristic Algorithm for Solving Engineering Optimization Problems. Engineering Applications of Artificial Intelligence, 2020, 94, 103731. https://doi.org/10.1016/j.engappai.2020.103731

20. Huang, H., Li, H., Wang, M., Wu, Y., He, X. Multi-UAV Cooperative Path Planning Based on Aquila Optimizer. International Conference on Autonomous Unmanned Systems, 2022, 2005-2014. https://doi.org/10.1007/978-981-99-0479-2_186

21. Jiao, L., Peng, Z., Xi, L., Guo, M., Ding, S., Wei, Y. A Multi-Stage Heuristic Algorithm Based on Task Grouping for Vehicle Routing Problem with Energy Constraint in Disasters. Expert Systems with Applications, 2023, 212, 118740. https://doi.org/10.1016/j.eswa.2022.118740

22. Jothi Lakshmi, S., Karishma, M. A Modified DSR Protocol Using Deep Reinforced Learning for MANETs. IETE Journal of Research, August 2023. https://doi.org/10.1080/03772063.2023.2223168

23. Khan, M., Zaki, Y., Iyer, S., Ahamd, T., Poetsch, T., Chen, J., Sivaraman, A., Subramanian, L. The Case for Model-Driven Interpretability of Delay-Based Congestion Control Protocols. ACM SIGCOMM Computer Communication Review, 2021, 51(1), 18-25. https://doi.org/10.1145/3457175.3457179

24. Lhamo, O., Ma, M., Doan, T. V., Scheinert, T., Nguyen, G. T., Reisslein, M., Fitzek, F. H. P. RED-SP-CoDel: Random Early Detection with Static Priority Scheduling and Controlled Delay AQM in Programmable Data Planes. Computer Communications, 2024, 214, 149-166. https://doi.org/10.1016/j.comcom.2023.11.026

25. Lu, K. Di, Wu, Z. G., Huang, T. Differential Evolution-Based Three-Stage Dynamic Cyber-Attack of Cyber-Physical Power Systems. IEEE/ASME Transactions on Mechatronics, 2023, 28(2), 1137-1148. https://doi.org/10.1109/TMECH.2022.3214314

26. Lu, K. Di, Zeng, G. Q., Luo, X., Weng, J., Luo, W., Wu, Y. Evolutionary Deep Belief Network for Cyber-Attack Detection in Industrial Automation and Control Systems. IEEE Transactions on Industrial Informatics, 2021, 17(11), 7618-7627. https://doi.org/10.1109/TII.2021.3053304

27. Pazhooheshy, P., Abbasloo, S., Ganjali, Y. Harnessing ML for Network Protocol Assessment: A Congestion Control Use Case. Proceedings of the 22nd ACM Workshop on Hot Topics in Networks, 2023, 213-219. https://doi.org/10.1145/3626111.3628182

28. Połap, D., Jaszcz, A. Decentralized Medical Image Classification System Using Dual-Input CNN Enhanced by Spatial Attention and Heuristic Support. Expert Systems with Applications, 2024, 253, 124343. https://doi.org/10.1016/j.eswa.2024.124343

29. Prabakeran, S., Sethukarasi, T. Optimal Solution for Malicious Node Detection and Prevention Using Hybrid Chaotic Particle Dragonfly Swarm Algorithm in VANETs. Wireless Networks, 2020, 26(8), 5897-5917. https://doi.org/10.1007/s11276-020-02413-0

30. Prokop, K., Połap, D. Heuristic-Based Image Stitching Algorithm with Automation of Parameters for Smart Solutions. Expert Systems with Applications, 2024, 241, 122792. https://doi.org/10.1016/j.eswa.2023.122792

31. Ramesh, B., Lakshmanna, K. A Novel Early Detection and Prevention of Coronary Heart Disease Framework Using Hybrid Deep Learning Model and Neural Fuzzy Inference System. IEEE Access, 2024, 12, 26683-26695. https://doi.org/10.1109/ACCESS.2024.3366537

32. Singh, D. S.. A Modified Stochastic Model for Rainfall Prediction Using Fuzzy Aquila Optimization. International Journal of Computational Intelligence Applications, 2023, 22(4). https://doi.org/10.1142/S1469026823500268

33. Sood, R., Kang, S. S. Hybrid Congestion Control Mechanism in Software-Defined Networks. International Journal of Intelligent Systems and Applications in Engineering, 2024, 12(6s), 676-686.

34. Soussi, N. Optimization of Columnar NoSQL Data Warehouse Model with Clarans Clustering Algorithm. Computing and Informatics, 2023, 42(3), 762-780. https://doi.org/10.31577/cai_2023_3_762

35. Subba Reddy, T., Harikiran, J., Enduri, M. K., Hajarathaiah, K., Almakdi, S., Alshehri, M., Naveed, Q. N., Rahman, M. H. Hyperspectral Image Classification with Optimized Compressed Synergic Deep Convolution Neural Network with Aquila Optimization. Computational Intelligence and Neuroscience, 2022. https://doi.org/10.1155/2022/6781740

36. Tan, L., Huang, K., Peng, G., Chen, G. Stability of TCP/AQM Networks Under DDoS Attacks with Design. IEEE Transactions on Network Science and Engineering, 2020, 7(4), 3042-3056. https://doi.org/10.1109/TNSE.2020.3012002

37. Tang, Junjie, Wang, L. A Whale Optimization Algorithm Based on Atom-Like Structure Differential Evolution for Solving Engineering Design Problems. Scientific Reports, 2024, 14(1), 1-27. https://doi.org/10.1038/s41598-023-51135-8

38. Tang, Junyong, Li, H., Zhang, J., Guan, K., Shan, Q., Liang, X. A Robust PID and RLS Controller for TCP/AQM System. Journal of Network and Computer Applications, 2024, 229, 103947. https://doi.org/10.1016/j.jnca.2024.103947

39. Van Thieu, N., Mirjalili, S. MEALPY: An Open-Source Library for Latest Meta-Heuristic Algorithms in Python. Journal of Systems Architecture, 2023, 139, 102871. https://doi.org/10.1016/j.sysarc.2023.102871

40. Wang, S., Kua, J., Jin, J., But, J., Jayaraman, P. P., Palanisamy, S. An SDN-Driven Active Queue Management Architecture for Assisting Mission-Critical Traffic Flows in Industrial Automation Systems. Proceedings of the ACM SIGCOMM 2024 Conference: Posters and Demos, 2024, 119-121. https://doi.org/10.1145/3672202.3673741

41. Xiong, W., Qiu, X., Wang, Z., Li, S. Robust Congestion Control for TCP/AQM Networks via a Predictive GPIO-Based Approach. IECON 2023-49th Annual Conference of the IEEE Industrial Electronics Society, 2023, 1-6. https://doi.org/10.1109/IECON51785.2023.10311975

42. Yen, C.-Y., Abbasloo, S., Chao, H. J. Computers Can Learn from the Heuristic Designs and Master Internet Congestion Control. Proceedings of the ACM SIGCOMM 2023 Conference, 2023, 255-274. https://doi.org/10.1145/3603269.3604838