

SDN 中基于蚁群系统优化的最优路径选择算法

董瑞芳¹

DONG Ruifang

摘要

蚁群算法是以不均等的概率进行多次迭代并收敛到最优路径的算法。结合软件定义网络 (software-defined networking, SDN) 统一管理、全局视图的特点, 蚁群算法适合解决 SDN 网络中的最优路径选择问题。但传统的蚁群算法在路由选择中缺乏针对性, 难以做到全局搜索和收敛速度的兼顾, 导致了算法结果的不可靠。针对以上问题, 文章提出了一种动态自适应蚁群系统算法 (dynamic adaptive ant colony system, DA-ACS)。DA-ACS 算法在蚁群系统算法 (ant colony system, ACS) 基础上, 重新设计了启发函数、信息素浓度、信息素挥发因子。实验结果表明, 该算法在 SDN 网络中的性能优于 ECMP 和 ACS 算法。

关键词

软件定义网络; 蚁群系统算法; 动态自适应蚁群系统算法; 启发函数; 信息素浓度

doi: 10.3969/j.issn.1672-9528.2025.10.012

0 引言

SDN^[1-2] (software-defined networking) 是一种可编程的网络架构技术, 其最大的特点是可以实现对底层网络设备的统一管理, 并获取全局的网络视图, 实现了底层设备对应用的透明。基于此特性, SDN 控制器具备灵活控制端到端流量的能力, 可实现整体链路的负载均衡。

近几年, 针对 SDN 网络的最优路径选择问题, 引起了较多学者的关注。文献 [3] 提出了一种基于 Dijkstra 算法, 计算最优路径的路由策略, 该路由策略为数据流选取的路径是当前的源和目的节点间负载最轻的路径, 该算法一定程度上提高了网络链路带宽的使用率, 但是评价指标单一,

且不太适用于规模较大的 SDN 网络场景。文献 [4] 将传统蚁群算法用于 SDN 网络的链路负载均衡, 实验结果表明, 采用蚁群算法的负载均衡效果比轮询算法的效果要好, 但是传统的蚁群算法的信息素更新缺乏对 SDN 网络的针对性, 而在蚁群算法中, 路径的选择又极其依赖信息素的更新, 因此传统蚁群算法不适合直接用于 SDN 网络的路径选择。文献 [5] 提出一种 K 路由算法, 通过对链路带宽等指标进行加权, 并计算 K 条可用路径, 一定程度上实现了链路负载均衡, 但是该算法在计算加权的未全面考虑链路的综合性能从而导致了链路的选择可能不是最优路径, 降低了整体的负载均衡效果。

基于当前的研究现状, 本文提出了基于 ACS^[6] (ant colony system) 改进的适用于 SDN 网络最优路径选择的蚁群算法

1. 南京财经大学红山学院 江苏南京 211300

[3] KINGMA D P, BA J. Adam: a method for stochastic optimization[EB/OL].(2017-01-30)[2025-06-26].<https://doi.org/10.48550/arXiv.1412.6980>.

[4] WOO S, PARK J, LEE J-Y, et al. CBAM: convolutional block attention module[C]//Computer Vision-ECCV 2018.Berlin:Springer,2018: 3-19.

[5] 茆震, 任玉蒙, 陈晓艳, 等. 一种改进 YOLOv5s 的多尺度目标检测算法 [J]. 传感技术学报, 2023, 36(2): 267-274.

[6] 潘锦业, 王苗苗, 阚威, 等. 基于 Adam 优化算法和长短期记忆神经网络的锂离子电池荷电状态估计方法 [J]. 电气技术, 2022, 23(4): 25-30.

【作者简介】

张浪浪 (1997—), 男, 江西九江人, 硕士, 助教, 研究方向: 知识图谱。

叶孟财 (1997—), 男, 浙江绍兴人, 硕士, 助教, 研究方向: 区块链、图像识别、职业教育。

祝武标 (1984—), 男, 浙江绍兴人, 硕士, 讲师, 研究方向: 职业教育。

游林伟 (1996—), 男, 四川广安人, 硕士, 助教, 研究方向: 软件开发、云计算。

(收稿日期: 2025-04-15 修回日期: 2025-10-09)

(DA-ACS)。该算法结合 SDN 网络的特性,重新定义了启发函数,并优化了信息素浓度的范围和信息素挥发系数,实验结果表明,对比 ACS 算法、ECMP^[7] (equal-cost multipath routing),该算法提高了网络带宽利用率,降低了端到端延时,有效地提高了网络的整体性能。

1 体系结构

在 SDN 的研究发展过程中,开放网络基金会^[8] (open networking foundation, ONF) 标准化组织是 SDN 的先驱,提出了基于 OpenFlow 协议的网络架构技术,为 SDN 的发展奠定了重要的基础。ONF 提出的 SDN 架构从上到下依次是应用层、控制层、基础设施层,其架构如图 1 所示。

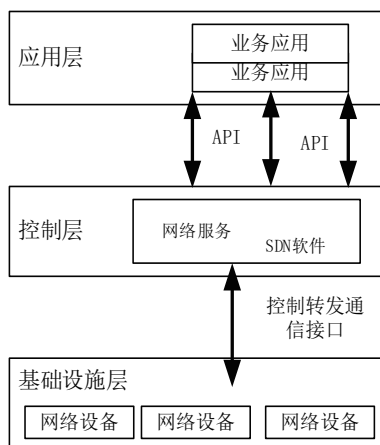


图 1 SDN 架构

目前开源的 SDN 控制器软件有 Floodlight、RYU、OpenDaylight、NOX 等,均都支持主流的南向接口协议 OpenFlow。其中基于 Python 编程语言的 RYU 控制器相比较其他类型的控制器,在扩展,编写和运行都较为灵活和方便,且其源代码清晰、文档较多,因此本文选择 RYU 控制器作为实验平台。

本文将 DA-ACS 算法作为 SDN 网络中最优路径的选择算法。通过 SDN 控制器实时监测链路的质量,并结合 DA-ACS 算法,计算出最优的路径,并添加到流表中,同时将更新过的流表下发给 OF 交换机,OF 交换机对数据流进行对应的动作处理。整体架构如图 2 所示。

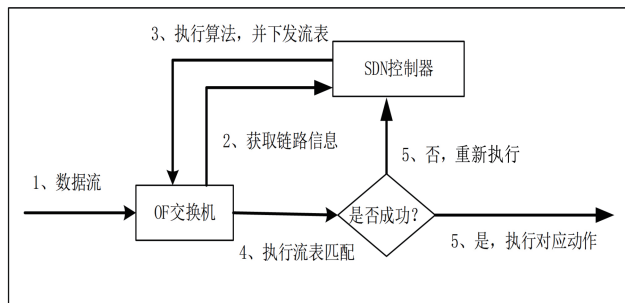


图 2 方案架构

2 基于改进蚁群算法的负载均衡方案

2.1 蚁群算法数学模型

蚂蚁 k 在运动过程中,转移的方向由路径上的信息素浓度决定, tabu_k ($k=1, 2, \dots, m$) 记录 k 只蚂蚁已经走过的节点,并且该表随着蚂蚁的运动状态调整,在算法的搜索过程中,蚂蚁会智能地选择下一步要走的路径。

设 m 表示蚂蚁总数量,用 d_{ij} ($i, j=0, 1, 2, \dots, n$) 表示节点 i 到节点 j 的距离; $\tau_{ij}(t)$ 表示在 t 时刻, i 节点和 j 节点所在路径上的信息素浓度。

在初始时刻蚂蚁是随机放置且各节点路径上的信息素浓度是相等的。在 t 时刻,蚂蚁 k 从 i 节点到 j 节点的概率公式为:

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{j \in \text{table}_k} \tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)} & j \in \text{tabu}_k \\ 0 & \text{other} \end{cases} \quad (1)$$

式中: $j \in \text{tabu}_k$ 表示蚂蚁 k 未走过的节点的集合; α 表示信息素的启发因子,该值反映信息素对蚂蚁选路的作用,值越大,信息素对路径选中的影响越大; β 表示探索新路径的启发因子,值越大表示蚂蚁对新路径越感兴趣; $\tau_{ij}(t)$ 表示经过 t 时刻路径的信息素浓度; $\eta_{ij}(t)$ 表示启发函数。

为避免残留过多的信息素而使启发信息被淹没,在每只蚂蚁遍历完成后,对残留的信息素进行更新,具体的更新规则用公式分别表示为:

$$\tau_{ij}(t+n) = (1-\rho) \times \tau_{ij}(t) + \tau_{ij} \quad (2)$$

$$\tau_{ij}(t) = \sum_{k=1}^M \tau_{ij}^k \quad (3)$$

式中: $1-\rho$ 表示残留的信息素挥发因子; $\tau_{ij}(t)$ 表示 t 时刻信息素 i 节点到 j 节点的增量。

在信息素的更新方式上, Dorigo^[9] 提出 3 种方式,分别为蚁周系统 (Ant-Cycle)、蚁量系统 (Ant-Quantity) 和蚁密系统 (Ant-Density)。

其中 Ant-Cycle 属于全局更新方式,后两种属于局部更新方式,采用 Ant-Cycle 更新方式,在求解诸多标准问题时,性能优于后两种方式。因此一般情况下,采用 Ant-cycle 作为信息素的更新方式。

2.2 典型蚁群算法分析

仿真表明,在求解小规模 TSP 问题时,基础的蚁群算法性能稳定,但是在大规模场景下,其性能下降严重,易出现搜索停滞、收敛速度慢等问题,因此众多学者提出了一些改进算法,典型的包括精英蚂蚁系统^[10]、最大最小蚁群系统^[11]、蚁群系统^[9]、基于排序的蚁群系统^[10]。

2.2.1 精英蚂蚁系统

精英蚂蚁系统由 Dorigo 等人提出, 其算法的核心思想是提高最优解在搜索路径中的分量, 每次迭代完成后, 对最优路径都有额外的信息素加成, 其信息素的更新方式为:

$$\tau_{ij}(t+n) = (1-\rho) \times \tau_{ij}(t) + \tau_{ij}^* \quad (4)$$

$$\tau_{ij}^* = \begin{cases} \frac{Q}{L^{bs}} & \text{if } (i,j) \in T^{best} \\ 0 & \text{other} \end{cases} \quad (5)$$

式中: T^{best} 表示最优路径; L^{bs} 表示路径的长度; τ_{ij}^* 表示额外加成的信息素增量; Q 表示信息素总量, 其他参数指标定义参照前文。

通过实验表明, 采用精英蚂蚁系统提高了算法的全局性, 同时增加了算法的收敛速度, 是一种有效的改进措施。

2.2.2 最大最小蚂蚁系统

该算法是目前解决 TSP 问题最好的蚁群算法, 其最主要的改进点是:

(1) 将路径上的信息素浓度限定在一个指定的范围内, 即 $[\tau_{min}, \tau_{max}]$, 若某条路径的信息素浓度超过了限定的范围, 会被重新设置为 τ_{min} 或者 τ_{max} , 避免了算法陷入局部最优解。同时当算法出现搜索停滞, 或者未找到解, 同样对路径上的信息素进行重置。

(2) 增强最优解的利用, 只有当前最优解或者是至今最优解的路径才能进行信息素的更新。

2.2.3 ACS 蚁群系统

相比于蚂蚁系统的算法, 其改进的主要点包括如下:

(1) 全局信息素的更新只在至今最优的路径上执行, 具体公式为:

$$\tau_{ij}(t+n) = (1-\rho) \times \tau_{ij}(t) + \rho \tau_{ij}^{bs} \quad (6)$$

$$\tau_{ij}^{bs} = \frac{Q}{L^{bs}} \quad (7)$$

式中: τ_{ij}^{bs} 表示最优路径上的信息素增量。

(2) 增加局部更新策略, 增加其他路径被选中的概率, 其具体的更新规则用公式表示为:

$$\tau_{ij}(t+n) = (1-\rho) \times \tau_{ij}(t) + \varepsilon \times \tau_0 \quad (8)$$

式中: ε 是常数, 范围在 0 到 1 之间; τ_0 是信息素的初始值。

当蚂蚁经过某条边时, 以一定的概率挥发掉该条路径上的信息素, 增加其他路径被搜索到的概率, 有效避免算法过早陷入停滞状态。

(3) 除信息素的更新之外, 蚁群系统 ACS 增加状态转移规则, 采用伪随机比例选择下一个节点, 其具体公式为:

$$P = \begin{cases} \arg\max \left\{ \tau_{ij}(t)^\alpha \cdot \eta_{ij}(t)^\beta \right\} & q \leq q_0 \\ P_{ij}^k(t) & q > q_0 \end{cases} \quad (9)$$

式中: q_0 是给定的参数, 在 0 到 1 之间; q 是均匀分布的随机变量。

当随机变量 q 的值小于等于给定 q_0 值时, 则将信息素和启发信息乘积最大的值的节点作为下一步的选择。反之, 则采用 $P_{ij}^k(t)$ 规则中的随机概率选择下一个节点。

2.2.4 排序的蚂蚁系统

核心的改进思想是当蚂蚁完成一次迭代后, 对所有的蚂蚁进行排序, 有且仅允许最优路径和前 k 只蚂蚁可以释放信息素, 其他路径信息素以一定的概率挥发。其信息素更新公式为:

$$\tau_{ij}(t+n) = \tau_{ij}(t) + \sum_{a=1}^k \tau_{ij}^a + \tau_{ij}^{bs} \quad (10)$$

式中: a 表示前 k 只蚂蚁; τ_{ij}^a 表示第 a 只蚂蚁经过边 (i, j) 时信息素的增加量。

2.3 改进的 DA-ACS 算法

蚁群算法在路径的选择中太过依赖于信息素机制, 所以算法易陷入局部最优解, 导致算法未找到最优路径就收敛, 影响选路的结果。同时信息素更新过于复杂容易导致算法收敛慢, 增加迭代次数。本论文在蚁群系统 ACS 算法的基础上, 结合 SDN 网络全局视图的特性, 提出 DA-ACS 算法。该算法的改进点主要在 3 个方面, 启发函数的定义、信息素浓度挥发系数的优化、信息素浓度范围的限制。

2.3.1 启发函数的定义

启发函数的定义必须能够真实反映网络路径的优劣情况, 由于 SDN 控制器可以获取全局的链路信息, 因此将带宽利用率和时延作为路径优劣的重要评价标准, 以此来定义启发函数。

(1) 带宽利用率

依据 SDN 控制器链路信息收集功能, 周期性地收集交换机接口信息, 计算出接口的已用带宽, 用公式分别表示为:

$$B_{ij}(T) = \frac{\text{Bytes}(T) - \text{Bytes}(T+t)}{t} \quad (11)$$

$$\sigma_{ij}(t) = \frac{B_{ij}(T)}{B_{ij}(\max)} \quad (12)$$

式中: $B_{ij}(T)$ 表示采样时间 t 间隔的接口已用带宽; $\text{Bytes}(T)$ 表示 T 时刻传输的字节数; $B_{ij}(\max)$ 表示最大带宽; $\sigma_{ij}(T)$ 表示带宽的利用率。

(2) 时延

链路时延探测通过 LLDP^[12] (link layer discovery protocol) 数据发送时的时间戳, 然后和收到的时间戳相减, 得到了 LLDP 数据包从控制器下发到 Switch A, 然后从 Switch A 到 Switch B, 再上报给控制器的时延 $t(i)$, 示例如图 3 所示的长划线箭头。同理反向的时延 $t(j)$ 由划线点的箭头组成。

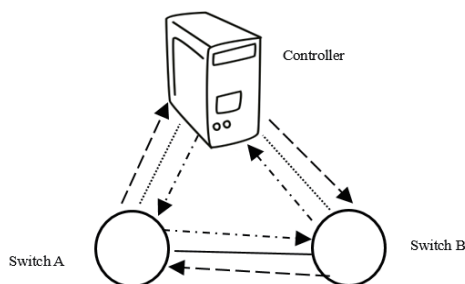


图3 时延获取原理简介

此外，控制器到交换机的往返时延由一个长划线箭头和一个划线的箭头组成，此部分时延由 echo 报文测试，分别为 $t(a)$ 和 $t(b)$ 。

最后链路的前向后向平均时延计算公式为：

$$t_{ij} = (t(i) + t(j)) - t(a) - t(b) \div 2 \quad (13)$$

(3) 指标归一化处理

带宽和时延两个指标参数，因其存在量纲上的差异，不能进行简单的加权评价，因此需要对这两个指标进行无量纲化处理^[12]，使得处理后的数值满足 $[0,1]$ 之间。具体的计算公式为：

对于正向指标：

$$x_{ij}^+(t) = u \times \frac{x_{ij}(t) - x_{\min}}{x_{\max} - x_{\min}} + (1 - u) \times \frac{x_{ij} - \bar{x}}{\Delta s} \quad (14)$$

对于负向指标：

$$x_{ij}^-(t) = u \times \frac{x_{\max} - x_{ij}(t)}{x_{\max} - x_{\min}} + (1 - u) \times \frac{x_{ij} - \bar{x}}{\Delta s} \quad (15)$$

式中： $x_{ij}^+(t)$ 、 $x_{ij}^-(t)$ 分别代表正负指标值，即带宽和时延两个参数； u 代表固定值，取值范围是 $(0,1)$ ，本文取值是 0.5； $x_{ij}(t)$ 代表 t 时刻的指标值； $x_{\max}$ 代表指标的最大值； $x_{\min}$ 代表指标最小值； $\bar{x}$ 代表指标的平均值； s 代表指标的标准差。

基于以上两个参数，带宽利用率和时延两个指标的权重取值为 0.5，其路径的综合评分计算公式为：

$$\text{Score}_{ij}(t) = 0.5 \times x_{ij}^+(t) + 0.5 \times x_{ij}^-(t) \quad (16)$$

式中： $\text{Score}_{ij}(t)$ 值越大，其代表的路径越优，因此可以将其作为启发函。

2.3.2 信息素挥发系数 ρ 动态优化

从现有的研究来看，挥发系数 ρ 和迭代次数之间存在一种非线性的负向关系，基于此为保证算法输出的可靠性和收敛阶段的速度，本文引入动态自适应的信息素挥发系数 ρ ，并限定范围 $\rho \in [\rho_{\min}, \rho_{\max}]$ ，其计算方式为：

$$\rho = (\rho_{\min} - \rho_{\max}) \times \sqrt[3]{\frac{X}{\text{NC}}} + \rho_{\max} \quad (17)$$

式中： $\rho_{\min}$ 、 $\rho_{\max}$ 表示信息素挥发系数的最小值和最大值；NC 表示总迭代次数； X 表示当前迭代次数。

算法初期限制了信息素的过度积累，增加了了解的多样性，后期随着迭代次数的增加，挥发次数的减少，信息素的浓度

快速增加，加快了后期算法的收敛速度。

2.3.3 信息素浓度的范围限制

引入最大最小蚂蚁系统的信息素处理方式，将路径上的信息素浓度限定在一个指定的范围内，即 $[\tau_{\min}, \tau_{\max}]$ 。当某条路径的信息素浓度超过此范围，会被重新设置为 $\tau_{\min}$ 或者 $\tau_{\max}$ ，避免了算法陷入局部最优解。同时当算法出现搜索停滞，或没有找到解，将对路径上的信息素进行重置。本论文信息素重置规则是：当路径信息素浓度低于 $\tau_{\min}$ ，则重置为 $\tau_{\min}$ ，高于 $\tau_{\max}$ 重置为 $\tau_{\max}$ 。

2.3.4 DA-ACS 算法步骤

DA-ACS 算法的具体步骤如下：

(1) 初始化网络及参数，网络拓扑，链路状态，蚂蚁数量 m 、最大迭代次数 NC、 α 、 β 、信息素浓度区间、信息素挥发系数 ρ 的区间、 q_0 伪随机比例因子。

(2) SDN 控制器获取链路信息，计算启发函数 $\eta_j(t)$ ，根据此前公式计算下一节点的选择概率，直到达到目的节点或者最大的迭代次数。

(3) 根据此前更新信息素浓度，如信息素浓度超出其限制的范围，则重置信息素，重复 (2)，并达到最大迭代次数。

(4) 输出最优解，并由 SDN 控制器下发到对应的 SDN 交换机，执行流表策略。

3 实验与结果分析

3.1 实验环境设定

Mininet2.0 是一款轻量级的仿真软件，可自定义需要的网络拓扑，并可配置交换机和网络的相关参数。本文采用此软件搭建实验环境，用 Iperf 软件生成数据流。SDN 控制器选用开源 RYU 平台。

蚁群算法的参数设置对其性能有决定性的作用，因此本论文中的参数选择参照了大量前人研究的结果，最终设定的参数如表 1 所示。

表 1 算法参数

参数	数值
蚂蚁数量 M	12
信息启发因子 α	1
期望启发因子 β	2
信息素挥发因子 ρ	$[0.3, 0.7]$
给定参数 Q	1
伪随机比例因子 q_0	0.5
τ_0	0.5
信息素浓度区间	$[0.3, 0.9]$
ε	0.3
算法迭代次数 NC	100

网络拓扑选用数据中心经典的 Fat-Tree 架构，并在 Mininet2.0 构建 8 个 Pod，其相关的命令是：topo = FatTree-

Topo ($k=4$); net=Mininet(topo); net.start()。带宽最大值 100 Mbit/s。

3.2 实验结果分析

为验证 DA-ACS 算法的性能,将通过时延、平均带宽利用率两个指标作为评价标准,并与 ACS 算法、ECMP 算法进行对比分析。

图 4 是 3 种算法的时延对比。在负载较低的情况下,其时延接近。但是随着负载的逐渐增大,ECMP 算法将数据流分配到多条路径上,时延较大。ACS 和 DA-ACS 算法能够相对准确地计算最优路径,但是 ACS 算法并未考虑时延等指标参数,而 DA-ACS 算法改进了 ACS 算法在 SDS 中的不足,因此其性能优于 ACS。

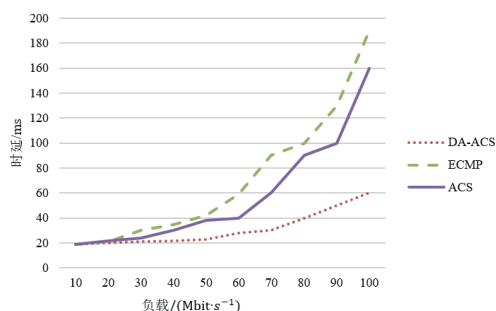


图 4 时延对比

图 5 是 3 种算法的平均带宽利用率对比,在负载较低的情况下,带宽利用率接近。随着负载的增大,网络出现拥塞,ECMP 算法带宽利用率出现明显下降,主要是因为该算法的负载方式相对简单,且自身没有拥塞的机制,在链路负载较大时,依然有概率将后续的数据流分配到负载较大的路径上。ACS 和 DA-ACS 算法能够相对准确地计算最优路径,其性能优于 ECMP 算法,且 DA-ACS 算法改进了 ACS 算法在 SDN 网络中的不足,因此其性能优于 ACS 算法。

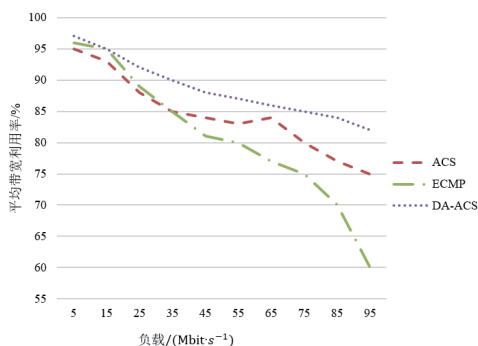


图 5 平均带宽利用率

4 总结

论文基于 SDN 网络和经典的 ACS 算法,提出了一种新的适用于 SDN 网络最优路径选择的算法,即 DA-ACS 算法。该算法从启发函数、信息素挥发因子和信息素浓度三个方

面进行优化,实验表明,该算法在 SDN 网络中,性能优于 ACS 算法、ECMP 算法,并且可以提高带宽的利用率,降低平均时延,提升了网络的整体负载。下一步将算法应用于校园的 SDN 网络中,验证算法的真实效果。

参考文献:

- [1] 崔峻玮,翟亚红.基于自适应遗传蚁群算法的 SDN 路由规划研究[J].天津理工大学学报,2023,39(3):39-48.
- [2] 张啸.基于流表和改进蚁群算法的 SDN 路由策略研究[D].南京:南京邮电大学,2023.
- [3] 刘懿.软件定义网络架构下的负载均衡路由策略研究[D].重庆:重庆邮电大学,2016.
- [4] 魏凯.基于蚁群算法 SDN 负载均衡的研究[D].长春:吉林大学,2015.
- [5] 吴宇文.基于 OpenFlow 的网络负载均衡算法的研究与设计[D].上海:华东师范大学,2014.
- [6] LIANG D, ZHAN Z H, ZHANG J. An adaptive ant colony system for public bicycle scheduling problem[J].Neural information processing, 2018, 11302: 417-429.
- [7] 张洁.数据中心网络 TCP 优化技术研究[D].长沙:湖南大学,2016.
- [8] Open networking foundation(ONF)[EB/OL].2011[2025-02-25].
<https://baike.baidu.com/item/ONF/823477>.
- [9] GAMBARDELLA L M, TAILLARD E D, DORIGO M. Ant colonies for the quadratic assignment problem[J]. Journal of the operational research society, 199,50:167-176.
- [10] 刘好斌.蚁群优化算法的改进及其在 TSP 中的应用[D].重庆:重庆大学,2010.
- [11] STUTZLE T,HOOS H. MAX-MIN ant System and local search for the traveling salesman problem[C/OL]// PProceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97).Piscataway:IEEE,1997[2025-02-23].
<https://ieeexplore.ieee.org/document/592327>.DOI:10.1109/ICEC.1997.592327.
- [12] 龙昭华,叶二伟,董瑞芳.SDN 中基于多指标的链路负载均衡模型[J].计算机工程与设计,2019,40(4): 948-952.

【作者简介】

董瑞芳(1992—),女,河南开封人,硕士研究生,助教,研究方向:SDN。

(收稿日期:2025-05-11 修回日期:2025-09-22)