Received: 20 August 2025 | Revised: 18 November 2025 | Accepted: 12 December 2025

DOI: 10.4218/etrij.2025-0344

ORIGINAL ARTICLE

ETRI Journal WILEY

Metaheuristic optimization of end-to-end service routing in multidomain network architectures

Doyoung Lee | **Tae Yeon Kim**

Network Research Division, Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea

Correspondence
Doyoung Lee, Network Research Division, Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea
Email: dylee90@etri.re.kr

Present address
Doyoung Lee, Division of Information and Communication Engineering, Kongju National University, Cheonan, Republic of Korea.

Funding information
Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2024-00392332, Development of 6G Network Integrated Intelligence Plane Technologies.

Abstract

As we move toward the era of 6G networks, the emergence of numerous end-to-end services with diverse user demands is anticipated. To support these services under varying network conditions, traffic must be routed through optimal paths that satisfy quality of service (QoS) requirements while minimizing transmission costs. Furthermore, climate change concerns are increasing pressure to reduce both energy consumption and carbon emissions resulting from service operations. Given that these complex factors affect multiple domains, it is essential to develop an effective method for routing optimization. To address this issue, we propose a metaheuristic optimization method for end-to-end service routing that considers dynamic network metrics and computing site information to reduce energy consumption and carbon emissions. Evaluation results show that our approach selects near-optimal paths by accounting for various factors, including QoS, energy consumption, and carbon emissions. Compared with benchmark schemes, our model reduces the joint energy and carbon objective by up to 63%, average service latency by up to 75%, and maintains the highest availability across all scenarios.

KEYWORDS

6G, genetic algorithm, metaheuristic approach, routing optimization, sustainable network

1 | INTRODUCTION

Considering the diverse and stringent requirements of services in dynamic network environments, operating end-to-end services in modern communication systems has become a challenging task, involving numerous considerations such as performance, cost, and efficiency [1]. Such challenges have made it difficult for network operators to manage networks manually while optimizing resources and service performance. To address these challenges, AI technologies have emerged as a key enabler, supporting automated network and service operations. Looking ahead to the era of 6G networks, it is anticipated that networks will evolve in an AI-native manner. Therefore, standards development organizations (SDOs), including 3GPP, IETF, and ETSI, are working to integrate AI into their system architectures [2].

Despite the benefits of AI for network management, its implementation poses significant energy-related challenges. Large-scale AI models require extensive data collection and computational resources for training and operation, which significantly increase energy consumption [3,4]. Furthermore, ongoing climate change, characterized by global warming, is pressuring network

operators to reduce both the energy consumption and carbon emissions associated with service operations [5,6].

Satisfying end-to-end service requirements necessitates the consideration of multiple heterogeneous domains [7–9]. For example, traffic exchanged between users and application servers, and delivered over mobile networks, traverses several domains, including access, transport, and core networks [10]. This traffic is then forwarded within data networks to appropriate computing nodes (for example, application servers) for processing, which consumes additional energy and emits carbon. Therefore, beyond network infrastructure itself, the state of computing sites must be considered to enable energy-efficient routing and support service operation. To meet service requirements while minimizing energy consumption and carbon emissions, it is essential to determine optimal traffic routes that account for both the current network status and condition of computing sites. A representative scenario in which such multidomain coordination becomes essential is emerging in 6G industrial IoT environments. In these settings, sensing, control, and analytics services must be delivered through access, transport, and data domains while relying on distributed computing nodes, making both quality of service (QoS) assurance and energy–carbon efficiency crucial [11].

To address these challenges, this paper proposes a novel metaheuristic routing method based on a genetic algorithm. The proposed method identifies energy-efficient traffic routes that guarantee QoS. Upon receiving a user request, our model monitors network metrics related to service performance, including latency, throughput, and packet loss rates, and computes site information, including available resources and energy status. This information is then used to make intelligent routing decisions, which are deployed on network elements to deliver the traffic required for the requested services.

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

- We construct a system model that integrates multiple heterogeneous network domains, distributed computing resources, service-level QoS constraints, and detailed cost functions. This model reflects the requirements of emerging large-scale service deployments, including applications such as 6G-enabled industrial IoT.
- We develop the enhanced genetic algorithm for routing optimization (EGARO), which introduces a variable-length chromosome structure, k-shortest-path-based population initialization, and specialized crossover and mutation operators designed to preserve path feasibility. This algorithm effectively addresses the combinatorial complexity of end-to-end service routing optimization problems.
- Comprehensive evaluations on real network topologies with diverse traffic scenarios are conducted to evaluate the effectiveness of the proposed method. Specifically, we demonstrate that EGARO consistently reduces total energy consumption and carbon emissions while satisfying QoS requirements. Compared with representative baselines, EGARO achieves significant improvements in terms of the multi-objective cost function and end-to-end performance metrics.

The remainder of this paper is organized as follows. Section 2 reviews related studies. Section 3 presents the system model for the proposed energy-efficient routing methodology. Section 4 details the proposed metaheuristic approach. Performance analysis is presented in Section 5, and Section 6 concludes this paper and discusses directions for future research.

2 | RELATED WORK

Over the past few decades, numerous studies have been conducted on traffic routing, aiming to improve network optimization and reliability [12-15]. Most of these studies have focused on routing decisions to establish optimal paths for service traffic while enhancing resource utilization and QoS. Because AI technologies have recently emerged as a promising solution for handling dynamic network conditions [16], machine learning (ML)-based approaches have been employed to automate path selection under varying network conditions [17,18]. In this context, Petro and others [19] proposed an AI/ML-based framework that predicts network conditions and makes routing decisions accordingly. Additionally, minimizing the cost and energy required for traffic forwarding has become a key research topic [20,21].

To achieve cost and energy savings while providing end-to-end services, both the communication and computing domains must be considered during traffic routing. Beshley and others [22] proposed a service-oriented, software-defined network architecture with advanced methods for service prioritization, server selection, and data transmission routing. Zhao and others [23] introduced a method for selecting computing nodes in a computing power network (CPN), where the computing power information from each site is combined with network metrics to determine routing paths. Additionally, several studies have explored approaches that jointly address routing optimization and broader network performance objectives. Ding and others [24] proposed a joint routing and resource allocation model aimed at minimizing overall network operating costs. Similarly, Chen and others [25] presented a method that simultaneously

optimizes user plane function (UPF) placement and traffic routing to enhance network efficiency.

Metaheuristic algorithm-based methods are also key enablers for making routing decisions in diverse network types. In particular, genetic algorithms (GAs) have been widely studied and applied to routing optimization problems. Riveros-Rojas and others [26] developed a GA-based solution to optimize the routing of traffic flows and assignment of software-defined networking (SDN) devices to these flows. Additionally, Akhter and Near [27] and Patel and H. E.-Ocla [28] proposed the energy-efficient routing methods for IoT networks and wireless sensor networks (WSNs), respectively. Moreover, some studies have combined different metaheuristic optimization techniques such as GAs and ant colony optimization (ACO) to optimize network performance while accounting for both network failures and traffic routing [29].

In addition, SDOs have made efforts to design system architectures that enable not only energy savings but also reductions in carbon emissions during end-to-end service operation. The 3GPP outlines use cases and requirements for energy efficiency in 5G networks [30], followed by the design of functions, interfaces, and procedures that are compliant with a service-based architecture [31]. In ETSI, the energy status and greenhouse gas emissions of virtual network functions have been studied to develop methods for estimating energy efficiency [32,33].

Although the work outlined above has contributed valuable insights into achieving energy savings and QoS for end-to-end service operations, studies specifically focusing on minimizing carbon emissions remain limited. Additionally, it is challenging to develop a metaheuristic-based method applicable to complex network environments. Therefore, further research is required to develop traffic-routing methods that jointly consider QoS, energy consumption, and carbon emissions, in line with the use cases and requirements defined by SDOs. Based on the reviewed studies, a summary of representative works is presented in Table 1.

TABLE 1 Summary of representative studies on routing optimization.

Ref.	Network type	Key contribution/approach
[19]	IP/SDN	AI/ML-based framework for predicting network conditions and enabling adaptive routing.
[22]	SDN	SDN architecture integrating service prioritization, server selection, and routing decisions.
[23]	CPN	Joint node selection and routing using computing power information and network metrics.
[24]	Heterogeneous	Joint routing and resource allocation model minimizing network operating costs.
[25]	5G Core	Joint UPF placement and traffic routing to improve core network efficiency.
[26]	SDN	Integration of GA and MILP to jointlyoptimize flow routing and device assignment.
[27]	IoT	GA-based routing method for reducing energy consumption in IoT traffic delivery.
[28]	WSN	GA-driven energy-efficient routing protocol for WSNs.
[29]	5G/6G	Hybrid GA-ACO algorithm enabling adaptive routing and network self-healing.

3 | SYSTEM DESIGN

3.1 | System model

In modern communication networks, users generally access a variety of end-to-end services, which generate and transmit diverse traffic types across multiple domains, spanning different network segments and computing sites. In the SDN paradigm, each network domain is managed by a controller that configures the data plane. Additionally, computing sites, which process service traffic, are managed by a management and orchestration platform based on network function virtualization technology. Subsequently, to establish service traffic paths, an end-to-end service orchestrator may be deployed at the top layer of the system, which interacts with the control planes of each domain. This system architecture is illustrated in Figure 1. Additionally, the notation used for system design is summarized in Table 2.

3.1.1 | Network model

We model an end-to-end network as an undirected graph $G=(N,L)$, where N and L represent the sets of nodes and links, respectively. The network consists of multiple domains, including access, core, transport, and data networks, which are denoted as $D\in\{\text{an, cn, tn, dn}\}$. The node set N is partitioned into domain-specific subsets N_d and a subset of computing nodes N_c such that $N=\left(\bigcup_{d\in D}N_d\right)\cup N_c$ and $N_c\cap N_d=\emptyset$. Each network node $i\in N_d$ has a forwarding capacity B_i, while each computing node $c\in N_c$ has a resource capacity $B_{c,r}$ for each

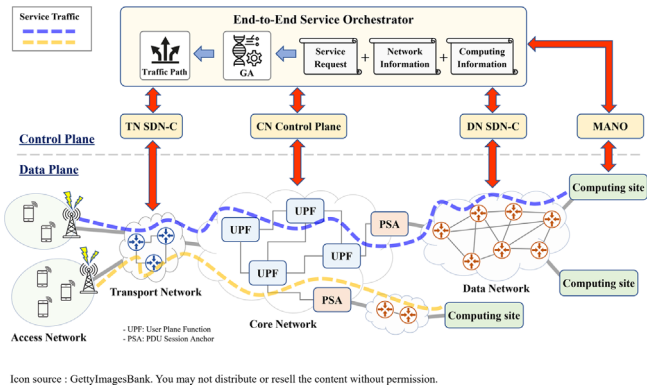

FIGURE 1 Overall architecture of traffic routing.

TABLE 2 Summary of notation.

Symbol	Description
$G=(N,L)$	Multidomain network graph
D	Set of domains (access, core, transport, data)
N_d, N_c	Nodes in domain d and computing nodes
L_d, L_{inter}	Intradomain and interdomain links
B_i	Forwarding capacity of node i
$B_{c,r}$	Capacity of resource r at compute node c
$\mathcal{S}$	Service demand set
$t \in \mathcal{T}$	Time slot index
$[t_s^{\text{start}}, t_s^{\text{end}}]$	Active interval of service s
s^{src}	Source node of service s
$l_s^{\max}$	Max latency of service s
$b_s^{\min}$	Min bandwidth of service s
$\theta_s^{\min}$	Min availability of service s
$u_{s,r}^{\min}$	Min required resource r for service s
a_{ij}, b_{ij}, d_{ij}	Avail., BW, delay of link between node i and j
$d_{s,c}^{p,t}$	Delay for processing service s in node c at time t
ρ_i^t	Traffic rate of node i at time t
$u_{c,r}^t$	Util. rate of resource r in node c at time t
$P_{\text{base},i}, P_{\text{base},c}$	Base power consumed at $i \in N_d, c \in N_c$
$\alpha_i, \beta_{c,r}$	Power consumed per $\rho_i^t, u_{c,r}^t$
E_i^t, E_c^t	Energy consumed in $i \in N_d, c \in N_c$ at time t
λ_i^t, λ_c^t	Carbon intensity in $i \in N_d, c \in N_c$ at time t
C_i^t, C_c^t	Carbon emissions in $i \in N_d, c \in N_c$ at time t
x_i^t	1 if node i is active at time t
y_c^t	1 if compute node c is active at time t
$z_{ij}^{s,t}$	1 if link (i,j) carries service s at time t
$y_{s,c}^t$	1 if service s uses compute node c at time t
$E_{\text{total}}^t, C_{\text{total}}^t$	Total energy and carbon emissions at time t
π_s^t	Path of service s at time t

resource type $r \in \mathcal{R}$, where $\mathcal{R} = \{\text{cpu}, \text{ram}, \text{gpu}, \text{disk}\}$. The links in L are categorized into intradomain and interdomain links (that is, $L = \left(\bigcup_{d \in D} L_d\right) \cup L_{\text{inter}}$ and $L_d \cap L_{\text{inter}} = \emptyset$). A link between nodes i and j is an intradomain link if both nodes belong to the same domain; otherwise, it is an interdomain link. Each link is characterized by a bandwidth b_{ij}, delay d_{ij}, and an availability a_{ij}.

3.1.2 | Service demand model

Let $\mathcal{S}$ denote the set of service demands in the network. Each demand $s \in \mathcal{S}$ is active over a time interval $[t_s^{\text{start}}, t_s^{\text{end}}]$ composed of discrete time slots, during which it generates traffic. A source node $s^{\text{src}} \in N_{\text{an}}$, which is typically located in the access domain and represents the entry point for service consumers, serves as the starting point of the service. Each service demand s is characterized by QoS and computing requirements, including a maximum end-to-end latency $l_s^{\max}$, minimum bandwidth requirement $b_s^{\min}$, minimum availability requirement $\theta_s^{\min}$, and required amount of computing resources $u_{s,r}^{\min}$ for each resource type $r \in \mathcal{R}$. These parameters must be satisfied for all time slots within $[t_s^{\text{start}}, t_s^{\text{end}}]$.

Different categories of services may exhibit diverse parameter requirements. For example, delay-sensitive applications typically impose small $l_s^{\max}$ and high $\theta_s^{\min}$, whereas data-intensive or computing-heavy services may tolerate higher latency but require greater bandwidth or = computational resources. Therefore, the proposed model accommodates heterogeneous service requirements while enabling the service orchestrator to determine end-to-end paths and computing locations that satisfy these requirements jointly.

3.1.3 | Energy consumption model

We model system energy consumption to quantify the energy required to provide services in the network by considering the power consumed during both traffic forwarding and service processing. Specifically, the total energy consumption is defined as the sum of the power consumed by network nodes and computing nodes within a given time interval.

The energy consumed by a network node i in time slot $t \in \mathcal{T}$ is defined as

$$E_i^t = (P_{\text{base},i} + \alpha_i \cdot \rho_i^t) \cdot \Delta t,\ i \in N_{d \in D}. \quad (1)$$

Here, E_i^t denotes the total energy consumed by node i during time slot t, which has a duration of Δt. This value

is computed as the sum of the base power consumption $P_{\text{base},i}$, which is the power required for node operation, and the additional power $\alpha_i \cdot \rho_i^t$, which depends on the traffic rate ρ_i^t and parameter α_i. The parameter α_i represents the amount of power consumed per unit of traffic rate. Multiplying the total power by the time slot duration Δt yields the total energy consumed.

The energy consumed by a computing node at a specific time slot t is denoted as

$$E_c^t = (P_{\text{base},c} + \sum_{r \in \mathcal{R}} \beta_{c,r} \cdot u_{c,r}^t) \cdot \Delta t,\ c \in N_c, \tag{2}$$

where E_c^t represents the energy consumption of computing node c at time t. This consumption is calculated as the sum of the base power $P_{\text{base},c}$ and the power used for processing service traffic, which is represented by the term $\beta_{c,r} \cdot u_{c,r}^t$. Here, $\beta_{c,r}$ denotes the power consumed per unit of resource usage for resource type r, while $u_{c,r}^t$ represents the utilization rate of that resource at time t. Just as in the network nodes, the total power is multiplied by the time slot duration Δt to determine the total energy consumed by the computing node.

$$C_i^t = E_i^t \cdot \lambda_i^t, \tag{3}$$

$$C_c^t = E_c^t \cdot \lambda_c^t. \tag{4}$$

In addition to energy consumption, the carbon emissions produced by network nodes and computing nodes are calculated using (3) and (4), respectively. The carbon emissions for each node type are derived from their respective carbon intensities, which represent the amount of carbon emitted per unit of energy consumed. Specifically, C_i^t and C_c^t denote the carbon emissions of a network node and computing node at time slot t, respectively. These values are computed by multiplying each node's energy consumption by its corresponding carbon intensity, which is denoted as λ_i^t for network nodes and λ_c^t for computing nodes.

3.2 | Problem formulation

When providing end-to-end services in a network, it is essential to minimize both energy consumption and carbon emissions while satisfying QoS requirements. To achieve this objective, service paths must be optimized. This task can be formulated as a routing problem that accounts for key factors affecting energy consumption and carbon emissions, considering both network and computing nodes.

3.2.1 | Optimization model

As the first key factor in addressing the aforementioned objective, it is essential to quantify the energy consumed while providing end-to-end services. The total energy consumption associated with a service path is defined as

$$E_{\text{total}}^t = \sum_{i \in N_d} x_i^t \cdot E_i^t + \sum_{c \in N_c} y_c^t \cdot E_c^t, \tag{5}$$

where x_i^t and y_c^t are binary variables indicating whether network node i and computing node c are assigned to forward and process service traffic, respectively. The value of each variable is determined as follows:

$$x_i^t = \begin{cases} 1, & \text{if } \exists s \in \mathcal{S}, z_{ij}^{s,t} = 1 \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

$$y_c^t = \begin{cases} 1, & \text{if } \exists s \in \mathcal{S}, y_{s,c}^t = 1 \\ 0, & \text{otherwise} \end{cases}. \tag{7}$$

Here, $z_{ij}^{s,t}$ is a binary variable indicating whether the link connecting nodes i and j transmits traffic for service s at time slot t. This variable takes on a value of one if the link is included in the service path and a value of zero otherwise. Similarly, y_c^t is defined as a binary variable, as shown in (7), where $y_{s,c}^t$ indicates that service s is processed at computing node c at time slot t. Therefore, the total energy consumption corresponds to the sum of the energy consumed by all nodes included in the selected service path. Likewise, the total carbon emitted at time t is the sum of the emissions from all network and computing nodes involved in the service path, as defined in (8).

$$C_{\text{total}}^t = \sum_{i \in N_d} x_i^t \cdot C_i^t + \sum_{c \in N_c} y_c^t \cdot C_c^t. \tag{8}$$

The goal of the routing problem is to minimize the total energy consumption and carbon emissions associated with all nodes involved in the service path, as defined in 9. The associated constraints are relevant to path selection while satisfying QoS requirements. Constraint (10) restricts the traffic rate in network node ρ_i^t to its maximum capacity, which is denoted by B_i. Constraint (11) ensures that the total delay across the selected links and the computing node does not exceed the service's maximum allowable delay, which is denoted by $l_s^{\max}$. Additionally, the availability of the links along the service path must be greater than or equal to the required service availability $\theta_s^{\min}$, as expressed in constraint (12). Here, π_s^t denotes the service path for

service s at time slot t. Constraint (13) guarantees that sufficient bandwidth is provisioned for service traffic along the selected path.

$$\min \sum_{t \in \mathcal{T}} (E^t_{\text{total}} + C^t_{\text{total}}), \tag{9}$$

$$s.t.\ \rho^t_i \le B_i,\ \forall i \in N_d, \forall t \in \mathcal{T}, \tag{10}$$

$$\sum_{(i,j) \in L} d_{ij} \cdot z^{s,t}_{ij} + d^{p,t}_{s,c} \cdot y^t_{s,c} \le l^{\max}_s,\ \forall s \in \mathcal{S}, \forall t \in \mathcal{T}, \tag{11}$$

$$\prod_{(i,j) \in \pi^t_s} a_{ij} \ge \theta^{min}_s,\ \forall s \in \mathcal{S}, \forall t \in \mathcal{T}, \tag{12}$$

$$\sum_{s \in \mathcal{S}} b^{min}_s \cdot z^{s,t}_{ij} \le b_{ij},\ \forall (i,j) \in L, \forall t \in \mathcal{T}, \tag{13}$$

$$\sum_{s \in \mathcal{S}} u^{min}_{s,r} \cdot y^t_{s,c} \le B_{c,r},\ \forall c \in N_c, \forall r \in \mathcal{R}, \forall t \in \mathcal{T}, \tag{14}$$

$$\sum_{c \in N_c} y^t_{s,c} = 1,\ \forall s \in \mathcal{S}, \forall t \in \mathcal{T}. \tag{15}$$

Constraints (10) to (13) apply to the network domain, whereas constraints (14) and (15) are specific to the computing domain. Constraint (14) ensures that the total resource usage across all services does not exceed the maximum capacity of each computing node, as represented by $B_{c,r}$. Finally, constraint (15) ensures that each service is assigned to exactly one computing node, thereby preventing service distribution across multiple sites.

3.2.2 | Computational complexity

The routing problem belongs to the class of constrained shortest path problems and is formulated as a mixed-integer nonlinear programming (MINLP) model with two sets of decision variables: x^t_i, which represent the network node selected for each service path, and y^t_c, which specify the computing nodes assigned to process the corresponding traffic. The resulting energy consumption E^t_{total} and carbon emissions C^t_{total} depend on the combination of routing and assignment decisions. Furthermore, these metrics correspond to coupled yet partially conflicting objectives, because minimizing energy does not necessarily minimize carbon emissions as a result of spatial and temporal variations in carbon intensity. Therefore, joint optimization over link-selection variables, resulting in network node selection, and computing node assignments yields a large-scale MINLP with $O(|\mathcal{S}| \cdot |\mathcal{T}| \cdot (|L| + |N_c|))$ variables. Because such MINLP problems are NP-hard and cannot be solved exactly in real time for realistic multidomain network sizes, we adopt a metaheuristic approach.

4 | PROPOSED APPROACH

4.1 | Metaheuristic approach for routing

Our system aims to find an optimal path that minimizes both energy consumption and carbon emissions while satisfying QoS requirements within given time slots. Based on the MINLP nature of this problem, which is known to be NP-hard, we propose a metaheuristic approach based on a GA. This approach is referred to as EGARO.

A GA [34] is a type of evolutionary method designed to find near-optimal solutions to NP-hard problems such as routing optimization, which is the subject of our analysis. The GA proceeds through several steps to obtain candidate solutions, each of which is also referred to as an individual or chromosome. To apply a GA to routing optimization, it is essential to define a suitable structure for chromosomes in advance. The GA attempts to derive optimal solutions over multiple generations by applying operators such as crossover and mutation. Therefore, it is also essential to design operators that not only guarantee the diversity of solutions but also enhance the efficiency of convergence to solutions.

4.2 | EGARO

In EGARO, a chromosome represents a service path, and each gene within the chromosome corresponds to a node ID, which is a unique value used to identify a specific node. The first and last nodes in a chromosome must be an access node and a computing node, respectively, as service traffic originates from users and is processed at computing sites. The intermediate nodes in a chromosome are the network nodes responsible for forwarding traffic within one of the network domains. Figure 2 illustrates the definition of a chromosome for routing optimization.

To derive diverse paths for routing optimization, the fixed-length chromosomes used in classical GAs are inappropriate because service paths can have variable lengths. Furthermore, service paths must not be excessively long, as they would incur increased costs and make it difficult to satisfy QoS requirements. Therefore, we define a chromosome $h \in \mathbb{N}^k$, where k is the maximum allowable path

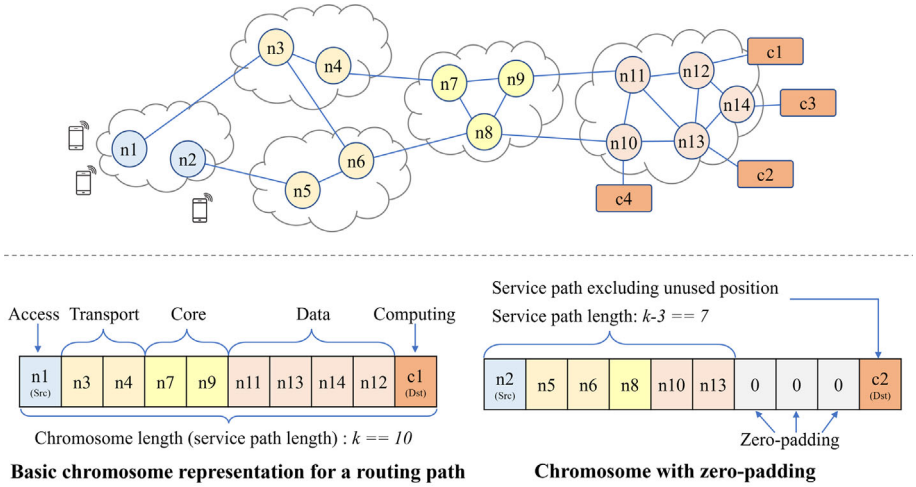

FIGURE 2 Chromosome representation.

Icon source : GettyImagesBank. You may not distribute or resell the content without permission.

length (for example, the maximum number of hops is k-1). If the actual path length is shorter than k, then zero-padding is applied to fill the unused positions in the chromosome with zero values. Specifically, zeros are inserted between the last network node in the path and the computing node.

Based on the defined solution representation, the chromosomes representing service paths must be generated to form an initial solution population. Because each service demand has a predetermined source, one of the computing nodes must be selected as its destination. When our system receives a set of service demands, the corresponding sources are extracted, and for each source node, m computing nodes must be selected as candidate destinations for the corresponding service traffic. Therefore, EGARO not only determines service traffic paths but also considers specific destinations, thereby preventing the exploration of impractical search spaces during optimization.

In general, initial chromosomes are generated randomly. However, in the case of multiconstrained routing problems in complex networks, random searching methods such as graph traversal algorithms incur significant computational overhead. Furthermore, using the shortest path algorithm to generate an initial population is detrimental to exploring the search space effectively. To address this issue, EGARO employs the k-shortest paths algorithm based on a randomly selected cost metric. Specifically, one of the following metrics is randomly selected and used as the cost: latency, bandwidth, or carbon intensity. Then, for each destination in the set of candidate destinations, the k-shortest paths are calculated to form the initial population. Subsequently, EGARO proceeds with the path determination process (for example, routing) using tournament selection. To evaluate the fitness of a solution h, (16) is applied to compute the corresponding fitness value.

$$f_h = \left(w_1 \cdot \hat{E}_h + w_2 \cdot \hat{C}_h + w_3 \cdot \delta_h + \varepsilon\right)^{-1}, \tag{16}$$

$$\hat{E}_h = \frac{E_h - E_{\min}}{E_{\max} - E_{\min} + \varepsilon}, \; \hat{C}_h = \frac{C_h - C_{\min}}{C_{\max} - C_{\min} + \varepsilon}. \tag{17}$$

The fitness function for chromosome h is calculated as the inverse of the weighted sum of its normalized energy consumption $\widehat{E_h}$, normalized carbon emissions $\widehat{C_h}$, and a penalty term δ_h, indicating whether the solution violates any QoS constraints. Each weight per component is denoted by w_1, w_2, w_3, and the sum of all weights is one. The normalized values are computed using (17), where the estimated static minimum and maximum values are used, and ε is included as a small constant to avoid division by zero. This fitness function ensures that our system prioritizes solutions that are both energy and carbon efficient while still satisfying QoS requirements.

To find an optimal path using EGARO, operators such as crossover and mutation are applied to two parent solutions associated with the same source node. Figure 3 illustrates the concept of genetic operations applied by EGARO. During the crossover process, EGARO employs single-point crossover, which first requires identifying a valid crossover position such that the crossover does not violate path connectivity. To determine this position, the system identifies common nodes, which are candidates for the crossover position that exist in both parent chromosomes but are not a source or destination, because choosing these points could result in unsuitable paths with no connections between intermediate nodes in offspring. Subsequently, one node is randomly selected among the candidates to traverse the unexplored search space effectively. The EGARO produces a wide variety of solutions by performing crossover operations on chromosomes that have the same source and different destinations, rather than being limited to chromosomes that have both the same source and destination.

In addition to the crossover process, a mutation operator is used to modify genes (for example, nodes in a service path) to preserve randomness and diversity within the population. During the mutation process, EGARO chooses a random number of consecutive nodes, excluding source nodes, rather than applying the single-node mutation used in typical GA-based methods [27,28], thereby enhancing exploration of the solution space. However, the mutation process may generate unqualified paths, where the modified nodes are not connected to the remainder of the original path, thereby disconnecting the path. To address this issue, EGARO applies the mutation process while checking the neighboring nodes of the modified nodes so as to guarantee that the mutated solution is still qualified to forward service traffic. In this manner, EGARO's mutation process prevents the waste of resources to generate unqualified solutions, thereby resulting in efficient exploration to derive optimal solutions. The overall procedure of EGARO is summarized in Algorithm 1.

Algorithm 1. EGARO

1: **procedure** EGARO
2: ▷ *Initialization Phase*
3: **for all** $s \in$ ServiceDemands **do**
4: $D \leftarrow$ RandSelect(s, m) // *random m dst.*
5: **for all** $d \in D$ **do** // d == *destination*
6: $cm \leftarrow$ RandCostMetric()
7: $paths \leftarrow$ KShortestPaths (d, k, cm)
8: Population $\leftarrow$ Encode $(paths)$
9: **end for**
10: **end for**
11: ▷ *Evolution Process*
12: **for** $gen \leftarrow 1$ to $G_{\max}$ **do**
13: $P \leftarrow$ TournamentSelection (Population)
14: $O \leftarrow$ Crossover(P)
15: $O \leftarrow$ Mutate(O)
16: $O \leftarrow$ EnsureFeasibility(O)
17: EvaluateFitness(O)
18: Population $\leftarrow$ Elitism $(O, \text{Population})$
19: **end for**
20: **return** BestChromosomes(Population)
21: **end procedure**

In Algorithm 1, the initialization phase (Lines 3–10) constructs the initial population by selecting candidate computing nodes for each service demand and generating corresponding paths based on the k-shortest paths, while filtering out infeasible solutions that violate basic connectivity. Subsequently, in the evolutionary phase (Lines 11–24), tournament selection is employed to select parent chromosomes, after which the crossover operator recombines feasible path segments at common intermediate nodes to preserve path connectivity. The mutation operator then rearranges selected genes by replacing nodes with their neighbors while ensuring that the resulting route remains a valid end-to-end path. Subsequently, each chromosome is evaluated using the fitness function, which jointly considers energy consumption, carbon emissions, and QoS violation penalties. Finally, an elitist replacement strategy preserves the best individuals from the current generation, and this process is repeated until the maximum number of generations is reached or convergence occurs.

5 | EVALUATION

5.1 | Simulation environment

To demonstrate the effectiveness of the proposed method, we implemented a Python-based network simulation environment consisting of different types of network nodes, computing nodes, and links. To construct this environment, network topologies with different numbers of nodes and links provided by SNDlib [35] were used: (i) Abilene with 12 nodes and 15 links, (ii) Nobel-EU with 28 nodes and 41 links, (iii) Germany50 with 50 nodes and 88 links, and (iv) TA2 with 65 nodes and 108 links. Next, the network nodes of the SNDlib topologies were categorized according to their betweenness centrality [36], which represents node importance by measuring how frequently nodes appear in shortest paths. In our multidomain scenarios, each network domain role is assigned according to descending betweenness centrality values, meaning the nodes with the highest centrality correspond to the data network, followed by core, transport, and access networks. Figure 4 presents examples of each of these network topologies.

Based on the simulation environment, we implemented not only the proposed method but also several conventional routing methods for performance comparisons: (i) Dijkstra, (ii) path assignment (PA), (iii) energy efficient GA (EEGA), (iv) GA-based ad-hoc on-demand multipath distance vector (GA-AOMDV), and (v) the proposed EGARO. Among these, the Dijkstra method determines a routing path using the number of hops, selecting the path

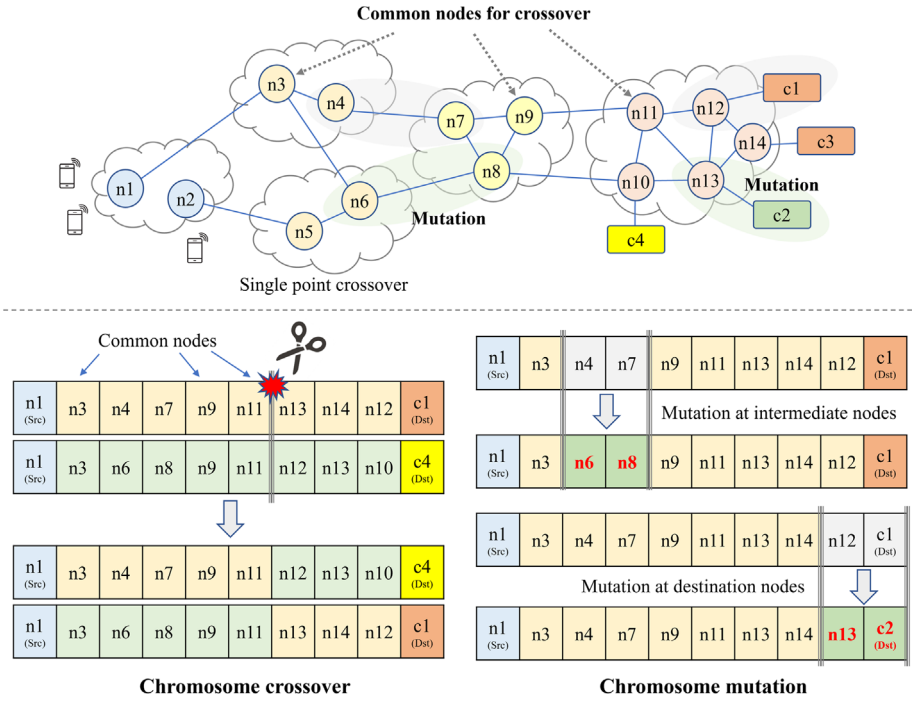

FIGURE 3 Chromosome operators: crossover and mutation.

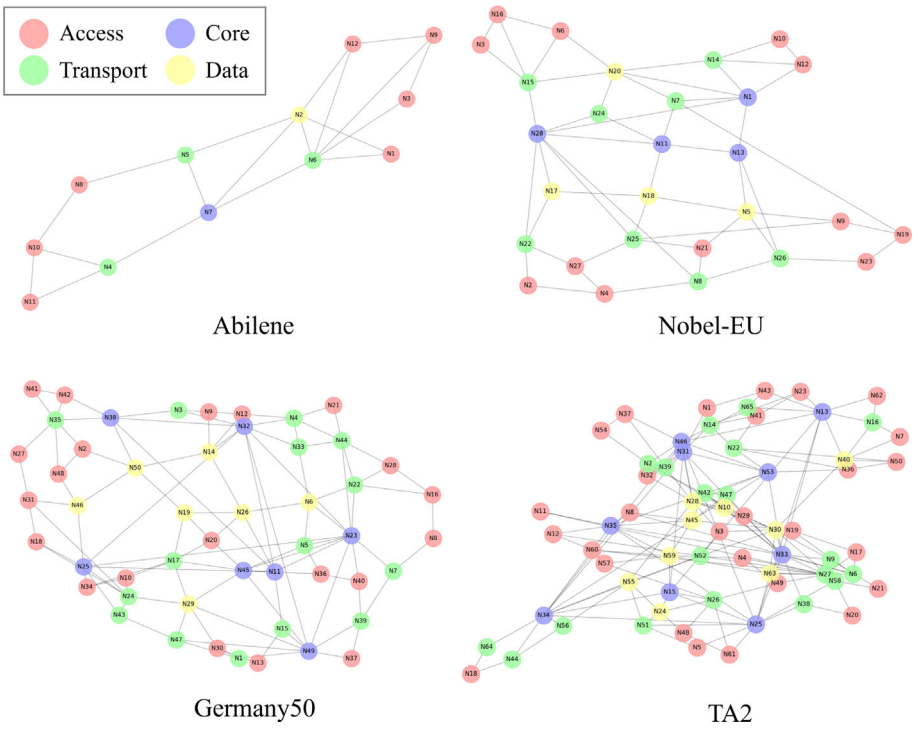

FIGURE 4 Network topologies for evaluation.

with the minimum number of hops from a source to a destination. The PA is a baseline approach that is widely used in tunneling mechanisms. It simply selects one of the predetermined paths to maximize link bandwidth [24]. The EEGA [27] is a GA-based approach that aims to minimize energy consumption for traffic transmission using a single-point crossover and single-gene mutation as operators. Similarly, GA-AOMDV [28] is a GA-based method that aims to find additional paths with less energy consumption based on the paths identified by a basic AOMDV protocol, which uses a single-node swapping operator.

We generated 100 service demands with random requirements for each topology and method in every time

slot with a population size of 50. In each time slot, random values were assigned to the attributes of links and nodes within predefined ranges for each domain to vary the network conditions. The remaining evaluation parameter details are listed in Table 3.

5.2 | Performance analysis

The first performance metric is the success rate, which is compared across different methods and network topologies in Figure 5. This metric indicates how many of the selected routing paths for the service demands meet the corresponding requirements. One can see that the proposed EGARO method consistently outperforms all other approaches, achieving the highest success rates across all four network topologies. Notably, while traditional methods such as Dijkstra and PA exhibit relatively high variability and poor average success rates, excluding the simple network topology, Abilene, EGARO demonstrates enhanced robustness and reliability. Additionally, GA-based methods such as EEGA and GA-AOMDV perform competitively but still fall short of EGARO, highlighting the effectiveness of the proposed optimization method.

Next, the objective function values defined in (9) for various methods and network topologies are presented in Figure 6. According to the results, EGARO yields the lowest objective function values in all cases, indicating more efficient resource allocation and routing decisions in terms of sustainability metrics such as energy consumption and carbon emissions. In contrast, the PA method yields the highest values, reflecting suboptimal path selection and increased overall cost. The EEGA and GA-AOMDV perform moderately, demonstrating the advantage of evolutionary optimization but without the full benefits of EGARO's joint consideration of multiple objectives.

To demonstrate the effectiveness of EGARO in terms of various performance metrics, including latency, bandwidth, and availability, we analyzed the average values of these metrics for all network topologies. The latency analysis in Figure 7 reveals that EGARO consistently achieves the lowest average latency across all cases. In addition, Dijkstra offers low latency because it aims to derive the path with the minimum number of hops; however, its lower success rate and higher variability

TABLE 3 Simulation parameters.

Attribute	Values
GA population size	50
GA generations	100
GA mutation and crossover rate	0.2, 0.7
GA fitness weights (w_1, w_2, w_3)	0.3, 0.6, 0.1
GA tournament size	3
Link latency (ms)	[0.1, 100]
Link bandwidth (Mbps)	[10, 10^5]
Link availability	[0.959, 0.95999]
Node power per traffic (W/Mbps)	[0.2, 0.6]
Node carbon intensity (gCO2e/kWh)	[100, 1000]
Node power per CPU (W/core)	[5, 15]
Node power per RAM (W/GB)	[0.1, 0.5]
Node power per GPU (W/GPU unit)	[50, 200]
Node power per Disk (W/GB)	[0.05, 0.2]

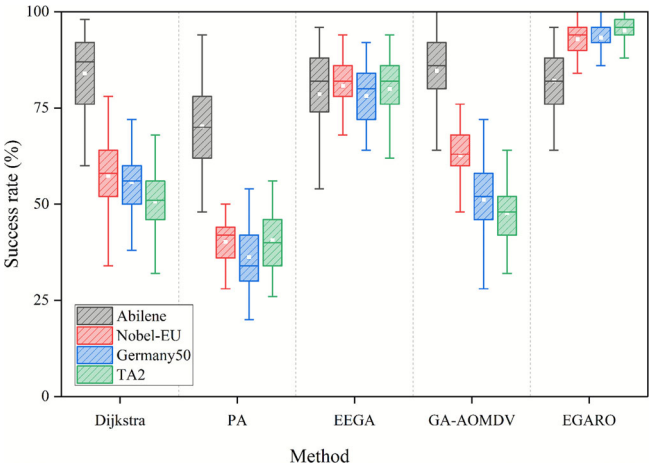

FIGURE 5 Comparison of service success rates.

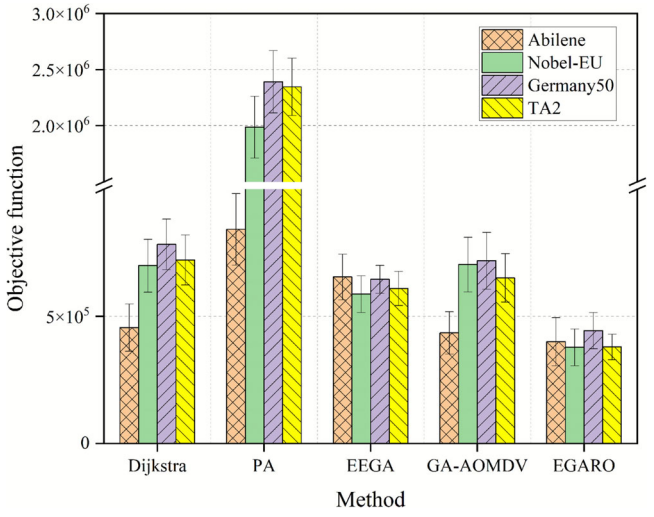

FIGURE 6 Comparison of objective function values.

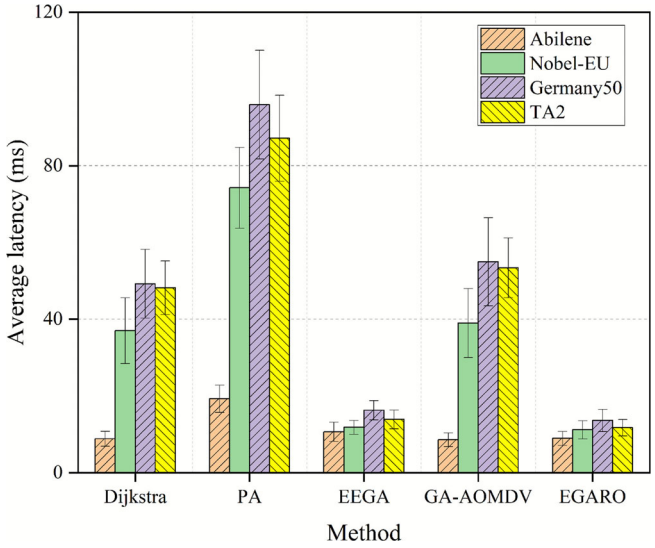

FIGURE 7 Comparison of average service latency.

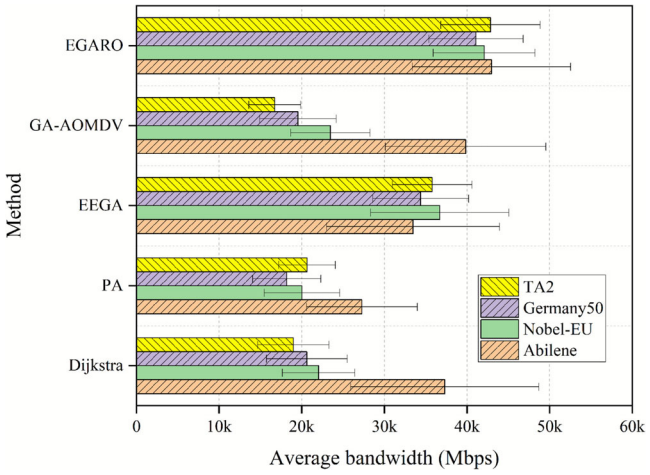

FIGURE 8 Comparison of average service bandwidth.

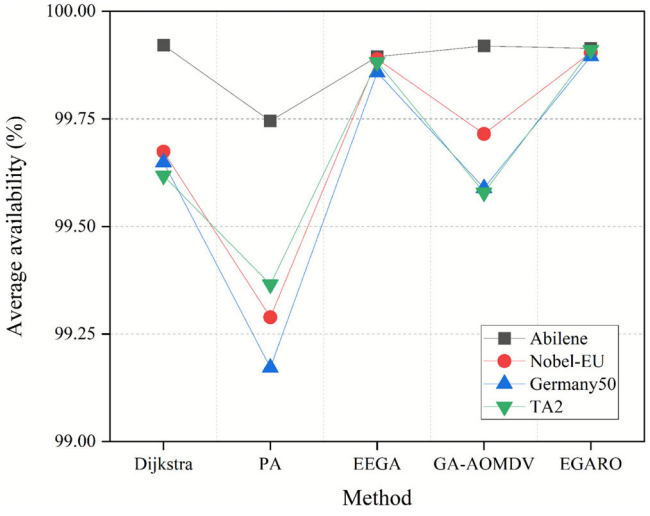

FIGURE 9 Comparison of average service availability.

diminish its overall reliability. Conversely, PA yields the highest latency, highlighting its inefficiency. The GA-based methods exhibit moderate latency, suggesting a partial compromise between optimality and delay.

As shown in Figure 8, EGARO achieves the highest average bandwidth across all network topologies. This result is particularly significant because bandwidth is directly correlated with throughput and QoS. While EEGA and GA-AOMDV follow closely behind, traditional methods such as PA and Dijkstra utilize significantly less bandwidth, reflecting their limitations in dynamic or high-demand network environments.

Figure 9 compares the average availability of each method. The EGARO maintains the highest availability levels, exhibiting values close to 100% across all network scenarios. This advantage is critical because it reflects EGARO's capacity to maintain service continuity under varying network conditions. In contrast, PA exhibits the lowest availability, demonstrating its poor suitability for scenarios requiring high reliability.

To further examine the scalability and performance of EGARO, we evaluated its behavior under different configurations (C1 to C4) for each service demand and conducted an ablation study. C1 refers to a case that uses all EGARO functionalities, including enhanced population initialization and genetic operations. In contrast, the other cases (C2 to C4) employ these functionalities selectively. C2 and C3 only adopt enhanced population initialization and genetic operations, respectively. C4 does not leverage any EGARO features, using only the composite fitness function. In the models not using EGARO features, the conventional approaches used in EEGA and GA-AOMDV were applied.

As shown in Figure 10, the average objective function value increases with each configuration that does not use the enhanced functionalities of EGARO, demonstrating reduced effectiveness when not using the proposed method. Furthermore, when EGARO's enhancements are applied, no significant increase in execution time is observed. These results indicate that although more complex configurations improve optimization outcomes, the associated computational cost remains relatively low, indicating EGARO's practicality for real-time or large-scale applications.

Figure 11 explores the impact of varying weight configurations in EGARO's fitness function on energy consumption, carbon emissions, and the penalty for QoS violation. As the weight for carbon emissions increases (from left to right), the total objective function value and carbon intensity both decreases, indicating the successful

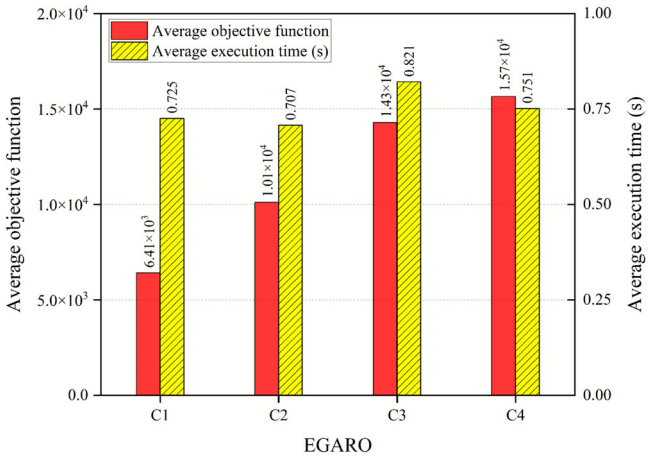

FIGURE 10 Effects of different EGARO configurations.

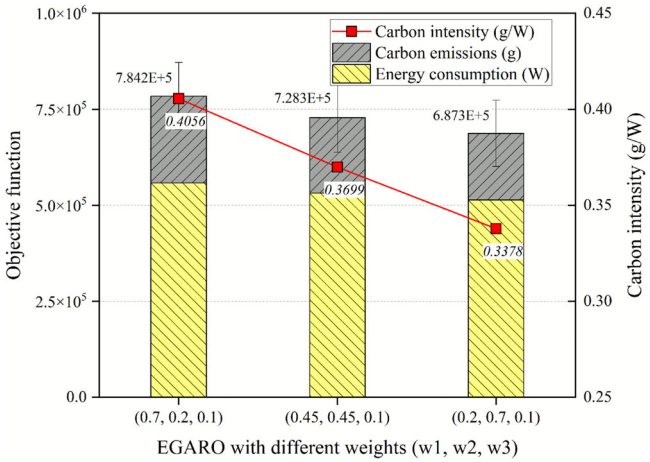

FIGURE 11 Effect of fitness function weights.

tuning of the algorithm to prioritize environmental sustainability. These results validate the flexibility of EGARO in terms of accommodating different policy or design priorities by adjusting its optimization weights.

5.3 | Discussion

While our evaluations demonstrate the effectiveness of EGARO, we did not explicitly quantify the energy consumption or computational cost associated with running the algorithm itself. Our analysis primarily focused on the energy and carbon footprint of the network and computing infrastructure, implicitly assuming that the orchestration overhead will remain relatively small. A more rigorous assessment of overhead, including run time and energy usage across different topology sizes and traffic loads, would strengthen our overall understanding of EGARO's operational efficiency. Incorporating such measurements into the optimization framework represents a valuable direction for future work and will help clarify the trade-off between optimization complexity and the achievable gains in energy and carbon reduction.

6 | CONCLUSION

To support end-to-end services while meeting stringent QoS requirements and reducing both energy consumption and carbon emissions, it is necessary to establish optimal traffic paths. To achieve such optimization, heterogeneous domains in which various networks transmit traffic to computing sites for processing must be considered. Because the metrics affecting QoS, energy consumption, and carbon emissions are complex and dynamically variable, we proposed a metaheuristic routing method employing a GA that accounts for these factors. Numerical experiments using real telecommunication topologies from SNDlib demonstrated that the proposed method consistently outperforms classical shortest-path routing and existing GA-based schemes. Compared with previous schemes, our method achieved up to 63% reduction in the joint energy–carbon objective and up to 75% lower average latency, while maintaining the highest availability.

In the future, we will investigate an ML-based routing strategy utilizing a federated learning mechanism. Additionally, our evaluation scenarios will be extended by adopting use cases envisioned for 6G networks and by implementing additional metaheuristic approaches to serve as baselines for comparison. Finally, a detailed experimental study of the run time and energy consumption of the proposed method will be conducted to account for the execution cost of the optimization algorithm.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflicts of interest.

ORCID

Doyoung Lee https://orcid.org/0000-0001-6814-1668

REFERENCES

1. M. Banafaa, I. Shayea, J. Din, M. H. Azmi, A. Alashbi, Y. I. Daradkeh, and A. Alhammadi, *6G mobile communication technology: requirements, targets, applications, challenges, advantages, and opportunities*, Alex. Eng. J. (2023), 245–274.
2. Y. Ouyang, L. Wang, A. Yang, M. Shah, D. Belanger, T. Gao, L. Wei, and Y. Zhang, The next decade of telecommunications artificial intelligence, arXiv preprint (2021). DOI 10.48550/arXiv.2101.09163

3. L. Lu and B. Lyu, *Reducing energy consumption of neural architecture search: an inference latency prediction framework*, Sustain. Cities Soc. **67** (2021), 102747.
4. E. Strubell, A. Ganesh, and A. McCallum, *Energy and policy considerations for modern deep learning research*, (Proceedings of the Aaai Conference on Artificial Intelligence), 2020, pp. 13693–13696.
5. T. Li, L. Yu, Y. Ma, T. Duan, W. Huang, Y. Zhou, D. Jin, Y. Li, and T. Jiang, *Carbon emissions of 5G mobile networks in China*, Nat. Sustain. **6** (2023), no. 12, 1620–1631.
6. P. Zhang, Y. Xiao, Y. Li, X. Ge, G. Shi, and Y. Yang, *Towards net-zero carbon emissions in network AI for 6G and beyond*, IEEE Commun. Mag. **62** (2023), no. 4, 58–64.
7. P. Ahluwalia and U. Varshney, *Managing end-to-end quality of service in multiple heterogeneous wireless networks*, Int. J. Netw. Manag. **17** (2007), no. 3, 243–260.
8. V. Lopez, J. M. G. ran Josa, V. Uceda, F. Slyne, M. Ruffini, R. Vilalta, A. Mayoral, R. Muñoz, R. Casellas, and R. Martínez, *End-to-end service orchestration from access to backbone*, J. Opt. Commun. Netw. **9** (2017), no. 6, B137–B147.
9. Q. Ye, J. Li, K. Qu, W. Zhuang, X. S. Shen, and X. Li, *End-to-end quality of service in 5G networks: examining the effectiveness of a network slicing framework*, IEEE Veh. Technol. Mag. **13** (2018), no. 2, 65–74.
10. F. Tang, B. Mao, Y. Kawamoto, and N. Kato, *Survey on machine learning for intelligent end-to-end communication toward 6G: from network access, routing to traffic control and streaming adaption*, IEEE Commun. Surv. Tutor. **23** (2021), no. 3, 1578–1598.
11. N. H. Mahmood, G. Berardinelli, E. J. Khatib, R. Hashemi, C. De Lima, and M. Latva-aho, *A functional architecture for 6G special-purpose industrial IoT networks*, IEEE Trans. Indust. Inform. **19** (2022), no. 3, 2530–2540.
12. R. Amin, E. Rojas, A. Aqdus, S. Ramzan, D. Casillas-Perez, and J. M. Arco, *A survey on machine learning techniques for routing optimization in SDN*, IEEE Access **9** (2021), 104582–104611.
13. B. Isyaku, K. B. A. Bakar, F. A. Ghaleb, and A. Al-Nahari, *Dynamic routing and failure recovery approaches for efficient resource utilization in openflow-SDN: a survey*, IEEE Access **10** (2022), 121791–121815.
14. J. Lee and H. Ko, *Reliability-guaranteed multipath allocation algorithm in mobile network*, ETRI J. **44** (2022), no. 6, 936–944.
15. N. Wang, K. H. Ho, G. Pavlou, and M. Howarth, *An overview of routing optimization for internet traffic engineering*, IEEE Commun. Surv. Tutor. **10** (2008), no. 1, 36–56.
16. O. Bouchmal, B. Cimoli, R. Stabile, J. J. Vegas Olmos, and I. Tafur Monroy, *From classical to quantum machine learning: survey on routing optimization in 6G software defined networking*, Front. Commun. Netw. **4** (2023), 1220227.
17. Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, *State-of-the-art deep learning: evolving machine intelligence toward tomorrow's intelligent network traffic control systems*, IEEE Commun. Surv. Tutor. **19** (2017), no. 4, 2432–2455.
18. Y. Xiao, J. Liu, J. Wu, and N. Ansari, *Leveraging deep reinforcement learning for traffic engineering: a survey*, IEEE Commun. Surv. Tutor. **23** (2021), no. 4, 2064–2097.
19. P. M. Tshakwanda, S. T. Arzo, and M. Devetsikiotis, *Advancing 6G network performance: AI/ML framework for proactive management and dynamic optimal routing*, IEEE Open J. Comput. Soc. **5** (2024), 303–314.
20. M. Andrews, A. F. Anta, L. Zhang, and W. Zhao, *Routing for energy minimization in the speed scaling model*, (2010 Proceedings Ieee Infocom), 2010, pp. 1–9.
21. X. Ge, S. Tu, G. Mao, V. K. N. Lau, and L. Pan, *Cost efficiency optimization of 5G wireless backhaul networks*, IEEE Trans. Mobile Comput. **18** (2018), no. 12, 2796–2810.
22. M. Beshley, N. Kryvinska, H. Beshley, O. Panchenko, and M. Medvetskyi, *Traffic engineering and QOS/QOE supporting techniques for emerging service-oriented software-defined network*, J. Commun. Netw. **26** (2024), no. 1, 99–114.
23. Q. Zhao, B. Lei, and M. Wei, *A method for selecting computing power nodes in a computing power network scenario*, (2024 Ieee/Cic International Conference on Communications in China (iccc)), 2024, pp. 1373–1378.
24. H. Ding, F. He, P. Zhang, L. Zhang, X. Zhang, and M. Qi, *Optimal routing and heterogeneous resource allocation for computing-aware networks*, ICT Express, **10** (2024), no. 3, 614–619.
25. S. Chen, J. Chen, and H. Li, *Joint optimization of UPF placement and traffic routing for 5G core network user plane*, Comput. Commun. **216** (2024), 86–94.
26. G. J. Riveros-Rojas, P. P. Cespedes-Sanchez, D. P. Pinto-Roa, and H. Legal-Ayala, *Energy-and-blocking-aware routing and device assignment in software-defined networking—a MILP and genetic algorithm approach*, Math. Comput. Appl. **29** (2024), no. 2, 18.
27. F. Akhter and J. P. Near, *Energy efficient data routing for IoT networks using genetic algorithm*, (2024 International Conference on Smart Applications, Communications and Networking (Smartnets)), 2024, pp. 1–8.
28. J. Patel and H. El-Ocla, *Energy efficient routing protocol in sensor networks using genetic algorithm*, Sensors **21** (2021), no. 21, 7060.
29. A. Agrawal and A. K. Pal, *Adaptive hybrid genetic-ant colony optimization for dynamic self-healing and network performance optimization in 5G/6G networks*, Proc. Comput. Sci. **252** (2025), 404–413.
30. 3GPP, Service requirements for the 5G system: stage 1, TS 22.261, 2024.
31. 3GPP, Study on energy efficiency and energy saving, TR 23.700-66, 2024.
32. ETRI TS, Part 2: energy efficiency—dynamic measurement method, 102 706–2, 2024.
33. ETSI ES, Enhanced interface for power management in network functions virtualisation (NFV) environments, 203 682, 2024.
34. S. Mirjalili, *Genetic algorithm*, Evol. Algo. Neural Netw. Theory Appl. (2019), 43–55.
35. S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, *SNDLIB 1.0-survivable network design library*, Netw. Int. J. **55** (2010), no. 3, 276–286.
36. M. Barthelemy, *Betweenness centrality in large complex networks*, Europ. Phys. J. B **38** (2004), no. 2, 163–168.

AUTHOR BIOGRAPHIES

Doyoung Lee received his BS degree in Computer Science and Engineering from Konkuk University, Seoul, Republic of Korea, in 2015, and his Ph.D. in the same field from POSTECH, Pohang, Republic of Korea, in 2021. From 2021 to 2023, he worked as a staff engineer at Samsung Electronics, Suwon, Republic of Korea. He is currently a Senior Researcher in the Intelligent Network Section of the Electronics and Telecommunications Research Institute (ETRI), Deajeon, Republic of Korea. His research interests include network intelligence, network optimization, and 5G/6G networks.

Taeyeon Kim received his BS and MS degrees from Chung-Ang University, Seoul, Rep. of Korea, in 1990 and 1992, respectively. He received his Ph.D. in Computer Science from Chungbuk National University, Cheongju, Republic of Korea, in 2007. He joined ETRI, Daejeon, Republic of Korea, in 1992. His current research focuses on 6G intelligence for autonomous networks and communication network infrastructure for the upcoming AGI era.

How to cite this article: D. Lee and T. Y. Kim, *Metaheuristic optimization of end-to-end service routing in multidomain network architectures*, ETRI Journal (2026), 1–14, DOI 10.4218/etrij.2025-0344