

Article

GRM-SFCOD: Dynamic Pricing-Based Online Deployment for Reliable Multi-Replica Service Function Chains

Haitong Gu ¹, Bin Guo ¹, Jun Dong ¹, Xingxing Feng ¹, Xiaoqiang Wu ¹, Chaoheng Liang ¹, Jingbo Lin ¹, Zhuohang Yu ¹, Wanjuan Xie ^{2,*}  and Quansheng Guan ³ 

¹ Guangzhou Power Supply Bureau, Guangdong Power Grid Co., Ltd., Guangzhou 510620, China; 13822100718@139.com (H.G.); guobin735000@163.com (B.G.); dojundong@163.com (J.D.); xingxfing@163.com (X.F.); wuxq999@126.com (X.W.); lch199091@126.com (C.L.); jingblin@163.com (J.L.); zhuohyu@163.com (Z.Y.)

² Information Network Engineering and Research Center, South China University of Technology, Guangzhou 510640, China

³ School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China; eeqshguan@scut.edu.cn

* Correspondence: wanjuanxie@scut.edu.cn

Abstract

Virtual Network Functions (VNFs) deployed on commodity servers exhibit failure rates two orders of magnitude higher than carrier-grade hardware, rendering service function chain (SFC) reliability a critical concern. Existing solutions are confined to offline static scenarios and single-instance serial models, which create single-point bottlenecks and constrain network service provider (NSP) revenue. This paper proposes GRM-SFCOD, which is a General Reliability-aware Multi-replica SFC Online Deployment algorithm. We introduce a primary-backup replica pool that parallelizes VNFs into lightweight instances while provisioning redundant backups for fault tolerance. The online deployment problem is formulated as a Mixed-Integer Nonlinear Program (MINLP) that maximizes NSP revenue; its NP-hardness is proved via reduction from the Multidimensional Knapsack Problem. To enable efficient online decisions, we design a dynamic pricing mechanism with exponentially scaling resource prices, decomposing the problem into (i) replica allocation based on marginal reliability gain and (ii) VNF placement via an improved Genetic Algorithm. Simulations demonstrate a $\sim 9.68\%$ lower deployment cost compared to uniform-allocation baselines, $\sim 17\%$ higher resource utilization than Best-Fit heuristics, and an attainment of $\sim 81\%$ of offline optimal revenue with orders-of-magnitude lower computational overhead.

Keywords: service function chain; online deployment; multi-replica mechanism; reliability guarantee; dynamic pricing; mixed-integer nonlinear programming; resource heterogeneity



Academic Editor: Juan-Carlos Cano

Received: 21 May 2026

Revised: 2 July 2026

Accepted: 7 July 2026

Published: 9 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

With the explosive growth of mobile traffic, IoT proliferation, and big data analytics, global network traffic has surged dramatically. Traditional rigid network architectures—relying on dedicated hardware with tightly coupled functions—suffer from prolonged deployment cycles, complex operations, and poor elasticity, rendering them inadequate for dynamic service demands. The convergence of Software-Defined Networking (SDN) [1] and Network Functions Virtualization (NFV) [2], which decouples the control plane from the data plane [3,4], and virtualized network functions such as Virtual Network Functions (VNFs) [5], has catalyzed the emergence of the service function chain

(SFC). The SFC enables an on-demand composition of VNFs—such as firewalls and load balancers—into customized service chains, breaking the hardware-function binding and significantly reducing deployment and operational costs [6]. As a key enabler for end-to-end flexible services in 5G, smart cities, and Industry 4.0 [7], the SFC has garnered substantial industrial and academic attention.

Under this paradigm, network service providers (NSPs) face a critical decision-making challenge: dynamically admitting user requests and deploying service chains over resource-constrained physical infrastructure to maximize long-term profits. However, SFC reliability is severely compromised in this context. Unlike rigorously tested dedicated hardware, VNFs deployed on commodity servers are susceptible to software defects, resource contention, and configuration errors, resulting in markedly higher failure rates [8]. Owing to the cascading dependency inherent in SFC architectures, a single VNF failure can cascade into a complete service chain interruption, violating Service Level Agreements (SLAs) and directly impairing the reputation and economic viability of NSPs [9]. In mission-critical domains such as finance, healthcare, and industrial control, service chain reliability is paramount—it underpins data integrity and, in extremis, human safety. Consequently, devising an effective redundancy mechanism that ensures high service chain reliability under dynamic online request arrivals while optimizing resource allocation for enhanced long-term NSP profitability is of urgent practical significance.

To address these challenges, this paper proposes GRM-SFCOD, which is a reliability-aware online service chain deployment algorithm based on a multi-replica mechanism. The contributions are summarized as follows: (i) a novel reliability model leveraging collaborative primary-backup replicas to transcend single-instance serial limitations; (ii) a MINLP formulation of the online deployment problem with proved NP-hardness as well as a dynamic pricing-based heuristic decomposing the decision into cost minimization and admission control; and (iii) comprehensive validation demonstrating significant reductions in resource consumption and improvements in long-term NSP profitability while guaranteeing high reliability.

The remainder of this paper is organized as follows. Section 2 surveys related work. Section 3 presents the system model and problem formulation. Section 4 develops the primal-dual framework and dynamic pricing mechanism. Section 5 details the GRM-SFCOD algorithm. Section 6 reports experimental results. Section 7 concludes the paper and discusses future directions.

2. Related Work

Existing research on SFC deployment primarily addresses two dimensions: resource-performance optimization and reliability assurance.

2.1. Resource and Performance Optimization

Researchers typically formulate SFC deployment as Mixed-Integer Linear Programming (MILP) or Integer Linear Programming (ILP) problems, aiming to maximize operator profit or resource utilization subject to delay and bandwidth constraints [10–12]. Given that this problem has been proven NP-hard [13] and exact algorithms incur prohibitive computational complexity in large-scale networks [14], numerous studies have turned to heuristic approaches. For instance, Jin et al. [15] proposed a two-stage approximation algorithm to optimize resource reuse in edge environments; Varasteh et al. [16] decomposed subproblems to minimize power consumption; while Guo et al. [17], Hawilo et al. [18], Wang et al. [19], and Thanh et al. [20] developed optimization schemes based on Steiner trees, Markov decision processes, graph matching, and energy-aware strategies, respectively. However, most of these studies operate under offline assumptions, presuming

complete a priori knowledge of all requests for global optimization [21,22], which renders them ill suited for online environments characterized by dynamic, stochastic request arrivals. Although some online studies have employed primal-dual analysis [23] or deep reinforcement learning (DRL) [24,25] for dynamic decision making, they often suffer from critical limitations: unidimensional resource consideration, neglect of heterogeneous request revenues, or poor decision interpretability. Furthermore, these DRL-based approaches require extensive offline training with large replay buffers and computationally expensive forward passes during inference, making them unsuitable for online scenarios demanding millisecond-level per-request decisions without prior knowledge of future requests.

2.2. Reliability Assurance

The mainstream approach to mitigating VNF software failures involves redundancy backup mechanisms. Shared redundancy schemes allow multiple VNFs to pool backup resources, thereby reducing costs. Representative studies include the shared protection and criticality-prioritized backup strategies proposed by Wu et al. [22], Ding et al. [26], Kita et al. [27], and Dinh et al. [28]. Proprietary redundancy schemes, conversely, provision dedicated backups for each primary instance, achieving faster recovery and higher reliability at increased resource expenditure. Relevant contributions encompass Li et al.'s [29] hybrid local-remote backup, Liang et al.'s [30,31] multi-hop redundancy deployment, Chen et al.'s [32] primary-backup routing optimization, and Alleg et al.'s [33] cost-reliability trade-off scheme. Beyond software failure protection, some scholars have investigated probabilistic protection against infrastructure failures [34,35] and proactive recovery strategies based on failure prediction [36–39].

Alomari et al. [40] propose a Survivability Management Framework for SFCs in NFV environments that achieves cost-efficient full survivability against node failures through shared backup provisioning. However, their approach remains confined to offline static scenarios where all requests are known a priori. In contrast, our GRM-SFCOD addresses the online dynamic setting with stochastic arrivals, and it employs a primary-backup replica pool that parallelizes VNFs into lightweight instances rather than relying solely on shared backup provisioning.

Despite these advances, significant deficiencies persist: (i) most reliability studies remain confined to offline static assumptions [21,22,40], lacking an in-depth exploration of online dynamic admission control; (ii) prevailing models predominantly adhere to single-instance serial architectures with scant attention to multi-replica parallel mechanisms capable of improving throughput and resource efficiency [41]; and (iii) resource modeling is frequently simplified to a single dimension [23], disregarding the heterogeneity of multidimensional resources (e.g., computing, storage) prevalent in practical deployments.

Multi-replica mechanisms have proven effective in enhancing reliability beyond the SFC domain particularly in cloud and IoT environments where replica consistency and resource efficiency are critical. For instance, cross-cloud multi-replica auditing schemes [42] and TEE-based multi-replica data auditing frameworks [43] employ collaborative replica management to balance reliability assurance with resource overhead. While these works focus on static data storage with predetermined replica placements, their core insight—leveraging lightweight replica coordination to achieve fault tolerance without excessive redundancy—directly motivates our primary-backup replica pool design. The key distinction lies in the dynamic nature of SFC deployment: unlike static cloud storage scenarios, our GRM-SFCOD must adapt replica placement to stochastic request arrivals and real-time resource constraints, necessitating a dynamic pricing mechanism for online admission control and marginal reliability-aware replica allocation.

To bridge these gaps—specifically, the restriction to offline static scenarios, the adherence to single-instance serial architectures, and the oversimplification of resource-revenue modeling—this paper advances the state of the art along three fronts. First, focusing on online environments with dynamic, stochastic arrivals, a refined model is constructed capturing multidimensional resource heterogeneity and request revenue diversity. A dynamic shadow price mechanism is introduced to decompose the decision process into deployment cost minimization and request admission control, enabling an explicit optimization of long-term NSP profits. Second, transcending single-instance limitations, a novel collaborative mechanism between primary and backup replicas is pioneered. By decomposing heavyweight VNFs into lightweight parallel replicas, high fault tolerance is guaranteed while single-point bottlenecks are eliminated and throughput is enhanced. Third, the GRM-SFCOD algorithm is proposed, which tightly integrates an improved Genetic Algorithm with the dynamic pricing strategy to achieve a balanced trade-off between reliability assurance and profit maximization under multidimensional resource constraints.

3. System Model and Problem Formulation

This section establishes the mathematical foundation for the reliability-aware online SFC deployment problem. Section 3.1 models the physical network infrastructure. Section 3.2 formalizes the service function chain request. Section 3.3 introduces the primary-backup multi-replica reliability model. Section 3.4 presents the MINLP formulation and proves its NP-hardness.

3.1. Physical Network Model

The underlying physical network is modeled as an undirected graph $G(V, E)$, where V denotes the set of physical nodes (servers) and E represents the set of physical links. A link $(v_1, v_2) \in E$ indicates a direct connection between nodes $v_1, v_2 \in V$. Consistent with common assumptions in data center networking, we assume that intra-data-center links provide effectively infinite bandwidth and negligible propagation latency [29,30], which is well justified in single data center environments where high-capacity switches render link constraints secondary to computational resource bottlenecks [29,30,32]. We note that link bandwidth constraints can be incorporated into the MINLP as additional constraints on traffic traversing each physical link if the model is extended to inter-data-center or edge computing scenarios.

Each server node $v \in V$ is characterized by a vector of available resources—namely CPU, memory, and disk storage. The capacity of resource type res on node v is denoted by C_v^{res} , where $res \in \mathcal{R}$. For simplicity, the set of resource types is defined as $\mathcal{R} = \{\text{cpu}, \text{mem}, \text{disk}\}$. Accordingly, the capacities for CPU, memory, and disk storage are represented as C_v^{cpu} , C_v^{mem} , and C_v^{disk} , respectively.

3.2. Service Function Chain Model

The set of SFC requests is denoted by $\mathcal{U}$. Each request $u \in \mathcal{U}$ is modeled as a six-tuple: $\langle s_u, t_u, B_u, R_u, h_u, F_u \rangle$, where s_u and t_u represent the source and destination nodes, respectively; B_u denotes the bandwidth requirement; and R_u specifies the reliability constraint. The parameter h_u represents the value of the SFC, corresponding to the price the user is willing to pay for this request. $F_u = (f_{u,1}, f_{u,2}, \dots, f_{u,|F_u|})$ denotes the ordered sequence of VNFs constituting the service chain, where $|F_u|$ denotes the total number of network functions in the chain. The traffic flow for request u originates at s_u , traverses each VNF in F_u sequentially, and finally terminates at t_u .

Let $\mathcal{F}$ denote the set of available VNF types in the network. We assume that all instances of the same VNF type share identical specifications. For a specific VNF type

$f \in \mathcal{F}$, the maximum throughput of a single instance is denoted by b_f , and its resource demand for resource type res is denoted by $\pi_{res}(f)$. Furthermore, the failures of different replicas are assumed to be independent events. The reliability of a single replica for network function $f \in \mathcal{F}$ is denoted by r_f .

3.3. Multi-Replica Reliability Model

To satisfy both the bandwidth and reliability requirements of SFC requests, this paper replaces the conventional single-instance deployment with a replica pool architecture. As illustrated in Figure 1, each VNF is instantiated as a pool comprising m primary replicas and n backup replicas.

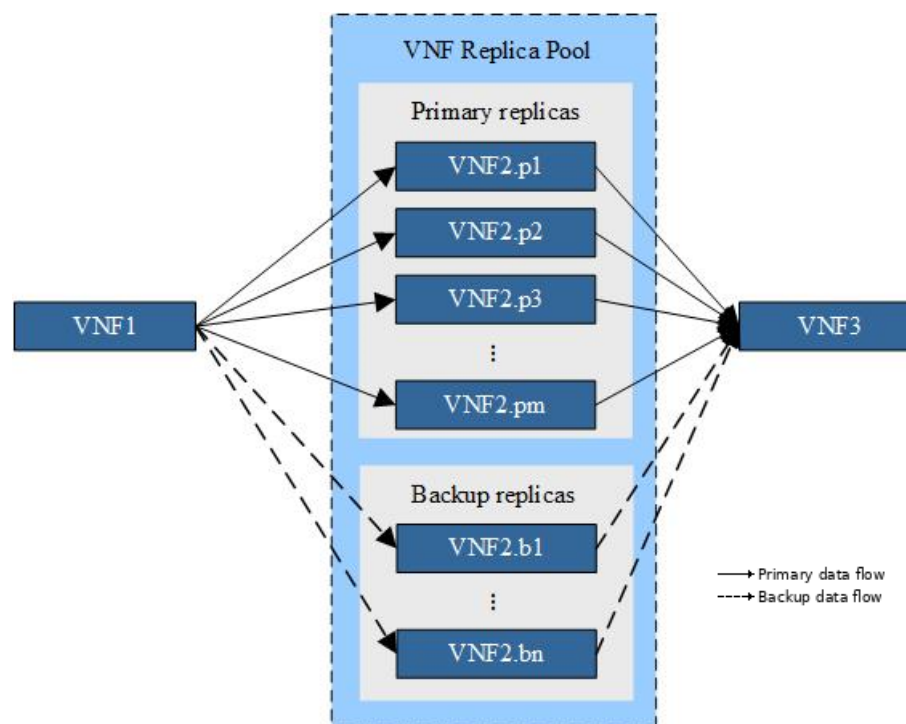


Figure 1. VNF replica pool.

The number of primary replicas, m , is determined by the minimum quantity required to meet the service chain's bandwidth demand. It is calculated as follows:

$$m = \left\lceil \frac{B_u}{b_{f_{u,i}}} \right\rceil \quad (1)$$

where $\lceil \cdot \rceil$ denotes the ceiling function, B_u represents the total bandwidth requirement of request u , and $b_{f_{u,i}}$ denotes the bandwidth capacity provided by a single replica instance of VNF $f_{u,i}$.

In this paper, we adopt a strict reliability model that mandates guaranteed bandwidth provisioning. For a VNF configured with m primary and n backup replicas, the system is considered operational if and only if at least m replicas within the pool remain available to handle the traffic load. Consequently, the reliability of a specific VNF instance, denoted by $R(f_{u,i})$, is formulated using the cumulative distribution function (CDF) of the binomial distribution:

$$R(f_{u,i}) = \sum_{k=m}^{m+n} \binom{m+n}{k} r_{f_{u,i}}^k (1 - r_{f_{u,i}})^{(m+n)-k} \quad (2)$$

where $r_{f_{u,i}}$ is the reliability of a single replica instance. This formula calculates the probability that at least m out of the total $m + n$ replicas are functional.

Since an SFC operates correctly only when all its constituent VNFs are functional, the end-to-end reliability of the service chain, $R(u)$, is modeled as the product of the reliabilities of all individual VNFs in the chain F_u :

$$R(u) = \prod_{f_{u,i} \in F_u} R(f_{u,i}) \quad (3)$$

It should be noted that the multi-replica architecture introduces synchronization overhead among primary replicas and failover latency from primary to backup instances. Since all replicas are deployed within the same data center where intra-DC links provide effectively infinite bandwidth and negligible propagation latency, these overheads are minimal in our target scenario. However, we acknowledge that in geo-distributed deployments or edge computing scenarios where replicas may span multiple data centers, inter-DC link latency and cross-node synchronization could become non-negligible bottlenecks. Furthermore, while this paper models replica failures as independent events, practical deployments may exhibit correlated failures when replicas share physical infrastructure (e.g., same rack, power supply, or hypervisor), software components, or management platforms. Addressing latency and synchronization challenges in geo-distributed multi-DC deployments and incorporating correlated failure models to capture infrastructure-level dependencies represent important directions for future work.

3.4. Problem Formulation

Based on the models described above, this subsection formalizes the online multi-replica SFC deployment problem.

Let δ_u be a binary decision variable indicating whether service chain request u is accepted. Specifically, $\delta_u = 1$ if the request is accepted, and $\delta_u = 0$ otherwise. Based on the proposed reliability-aware multi-replica VNF model, it is necessary to determine the optimal number of primary and backup replicas for each network function within the service chain. Let $M_{u,i}$ and $N_{u,i}$ denote the number of primary and backup replicas for the i -th VNF in service chain request u , respectively. To satisfy the bandwidth requirements of the service chain, the following condition must be met:

$$M_{u,i} \cdot b_{f_{u,i}} \geq \delta_u \cdot B_u, \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|] \quad (4)$$

where $b(f_{u,i})$ represents the bandwidth capacity provided by a single replica of VNF $f_{u,i}$.

Furthermore, to ensure the reliability requirements of the service chain, based on Equations (2) and (3), the end-to-end reliability must satisfy the following constraint:

$$\prod_{f_{u,i} \in F_u} \left[\sum_{k=M_{u,i}}^{M_{u,i}+N_{u,i}} \binom{M_{u,i}+N_{u,i}}{k} (r_{f_{u,i}})^k (1-r_{f_{u,i}})^{(M_{u,i}+N_{u,i})-k} \right] \geq \delta_u \cdot R_u \quad (5)$$

where $r_{f_{u,i}}$ is the reliability of a single replica instance of VNF $f_{u,i}$, and R_u is the minimum required reliability for the service chain. Note that the term inside the brackets represents the probability that at least $M_{u,i}$ out of $M_{u,i} + N_{u,i}$ total replicas are operational.

We define the binary placement variable $\eta_v^{u,i,j,B}$ to indicate whether the j -th backup replica of VNF $f_{u,i}$ is deployed on physical node v . Similarly, $\eta_v^{u,i,j,P}$ indicates whether the j -th primary replica is placed on node v . These variables take the value 1 if the replica is placed on the node and 0 otherwise.

The computing resources occupied by the replica instances allocated to a physical node must not exceed the node's total resource capacity. This resource constraint is expressed as follows:

$$\sum_{u \in \mathcal{U}} \sum_{i=1}^{|F_u|} \sum_{j=1}^{M_{u,i}} \delta_u \eta_v^{u,i,j,P} \pi_{res}(f_{u,i}) + \sum_{u \in \mathcal{U}} \sum_{i=1}^{|F_u|} \sum_{j=1}^{N_{u,i}} \delta_u \eta_v^{u,i,j,B} \pi_{res}(f_{u,i}) \leq C_v^{res}, \quad \forall v \in V, \forall res \in \mathcal{R} \quad (6)$$

where $\pi_{res}(f_{u,i})$ denotes the resource demand of VNF $f_{u,i}$ for resource type res , and C_v^{res} is the total capacity of resource res on node v .

For each accepted request, each replica instance must be deployed on exactly one unique node. This placement uniqueness constraint is expressed as follows:

$$\sum_{v \in V} \eta_v^{u,i,j,P} = \delta_u, \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|], \forall j \in [1, M_{u,i}] \quad (7)$$

$$\sum_{v \in V} \eta_v^{u,i,j,B} = \delta_u, \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|], \forall j \in [1, N_{u,i}] \quad (8)$$

The network service provider must make online decisions upon the arrival of service chain requests, determining whether to accept and deploy them. While satisfying bandwidth and reliability constraints, the accepted requests should be mapped onto the resource-limited underlying network. The objective is to maximize the overall benefit of network service chain deployments. The offline formulation of this optimization problem (P1) is defined as follows:

$$\text{P1: } \max_{\delta, M, N, \eta} \sum_{u \in \mathcal{U}} \delta_u h_u \quad (9)$$

s.t. (4)–(8)

$$\delta_u \in \{0, 1\}, \quad \forall u \in \mathcal{U} \quad (10)$$

$$M_{u,i}, N_{u,i} \in \mathbb{Z}_{\geq 0}, \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|] \quad (11)$$

$$\eta_v^{u,i,j,B} \in \{0, 1\}, \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|], \forall j \in [1, N_{u,i}] \quad (12)$$

$$\eta_v^{u,i,j,P} \in \{0, 1\}, \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|], \forall j \in [1, M_{u,i}] \quad (13)$$

4. Problem Analysis and Model Transformation

4.1. Complexity Analysis

By analyzing the objective function and constraints of problem P1, we conclude that P1 belongs to the class of Mixed-Integer Nonlinear Programming (MINLP) problems. Consequently, solving P1 is substantially more challenging than solving MILP problems. Even when employing classical methods such as branch-and-bound, the nonlinearity introduced by Constraint (5) imposes a significant computational burden.

To further analyze the computational complexity of P1, we consider a simplified instance. Suppose the underlying network consists of a single node v (i.e., $V = \{v\}$), and each service chain request u contains only a single VNF f_u . We set the bandwidth requirement such that $B_u \leq b_{f_u}$, which results in the number of primary replicas being $M_{u,1} = 1$. Additionally, we set the reliability requirement of the service chain to a minimal value, resulting in the number of backup replicas being $N_{u,1} = 1$. Under these conditions, the original problem is equivalent to the Multidimensional Knapsack Problem [44]. Since the Multidimensional Knapsack Problem is a classic NP-hard problem [45], Problem P1 is also NP-hard.

Given the NP-hardness of P1, directly obtaining its optimal solution is impractical for real-world applications. Therefore, we aim to develop an efficient algorithm that yields an approximately optimal solution within polynomial time. To this end, we first linearize

P1, and then we design an efficient online deployment algorithm for SFCs based on the primal-dual framework and economic theory.

4.2. Problem Linearization via McCormick Envelopes

In P1, the number of primary replicas $M_{u,i}$ and backup replicas $N_{u,i}$ for the i -th VNF of request u are governed by the bandwidth constraint (1) and the reliability constraint (5). Assuming that $M_{u,i}$ and $N_{u,i}$ are obtained by an approximation algorithm, these integer variables can be treated as fixed parameters in the original problem. Let $\tilde{M}_{u,i}$ and $\tilde{N}_{u,i}$ denote the fixed values of $M_{u,i}$ and $N_{u,i}$, respectively:

$$M_{u,i} = \tilde{M}_{u,i} \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|], \quad (14)$$

$$N_{u,i} = \tilde{N}_{u,i} \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|]. \quad (15)$$

In addition to the reliability constraint, the resource constraint (6) contains the product term $\delta_u \eta_v^{u,i,j}$ of two binary variables, which is nonlinear. We linearize this product using the classical McCormick envelope method by introducing auxiliary variables $z_v^{u,i,j,P}$ and $z_v^{u,i,j,B}$:

$$z_v^{u,i,j,P} = \delta_u \cdot \eta_v^{u,i,j,P}, \quad (16)$$

$$z_v^{u,i,j,B} = \delta_u \cdot \eta_v^{u,i,j,B}. \quad (17)$$

The following McCormick inequalities are then imposed:

$$z_v^{u,i,j,P} \leq \delta_u, \quad (18)$$

$$z_v^{u,i,j,P} \leq \eta_v^{u,i,j,P}, \quad (19)$$

$$z_v^{u,i,j,P} \geq \delta_u + \eta_v^{u,i,j,P} - 1, \quad (20)$$

$$z_v^{u,i,j,B} \leq \delta_u, \quad (21)$$

$$z_v^{u,i,j,B} \leq \eta_v^{u,i,j,B}, \quad (22)$$

$$z_v^{u,i,j,B} \geq \delta_u + \eta_v^{u,i,j,B} - 1. \quad (23)$$

Intuitively, $z_v^{u,i,j,P}$ (resp. $z_v^{u,i,j,B}$) indicates whether request u is accepted and the j -th primary (resp. backup) replica of the i -th VNF is deployed on node v . Consequently, the resource constraint (6) can be rewritten as

$$\sum_{u \in \mathcal{U}} \sum_{i=1}^{|F_u|} \sum_{j=1}^{\tilde{M}_{u,i}} z_v^{u,i,j,P} \pi_{res}(f_{u,i}) + \sum_{u \in \mathcal{U}} \sum_{i=1}^{|F_u|} \sum_{j=1}^{\tilde{N}_{u,i}} z_v^{u,i,j,B} \pi_{res}(f_{u,i}) \leq C_v^{res}, \quad \forall v \in V, \forall res \in \mathcal{R}. \quad (24)$$

The placement uniqueness constraints remain linear:

$$\sum_{v \in V} \eta_v^{u,i,j,P} = \delta_u, \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|], \forall j \in [1, \tilde{M}_{u,i}], \quad (25)$$

$$\sum_{v \in V} \eta_v^{u,i,j,B} = \delta_u, \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|], \forall j \in [1, \tilde{N}_{u,i}]. \quad (26)$$

After the above linearization, P1 can be reformulated as the following ILP problem, which is denoted by P2:

$$\text{P2: } \max_{\delta, \eta, z} \sum_{u \in \mathcal{U}} \delta_u h_u \quad (27)$$

s.t. (14)–(26)

$$\delta_u \in \{0, 1\}, \quad \forall u \in \mathcal{U} \quad (28)$$

$$z_v^{u,i,j,P} \in \{0, 1\}, \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|], \forall j \in [1, \tilde{M}_{u,i}] \quad (29)$$

$$z_v^{u,i,j,B} \in \{0, 1\}, \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|], \forall j \in [1, \tilde{N}_{u,i}] \quad (30)$$

$$\eta_v^{u,i,j,P}, \eta_v^{u,i,j,B} \in \{0, 1\}, \quad \forall u \in \mathcal{U}, \forall i \in [1, |F_u|], \forall j, v. \quad (31)$$

To facilitate subsequent analysis, let $\mathcal{L}_u$ denote the set of all deployment schemes for request u that satisfy constraints (24)–(26). Each scheme $l \in \mathcal{L}_u$ corresponds to a specific assignment of the variables η and z , representing a complete SFC deployment configuration. Let $g_{v,res}^{u,l}$ denote the consumption of resource res on node v under scheme l , which is defined as

$$g_{v,res}^{u,l} = \sum_{i=1}^{|F_u|} \left(\sum_{j=1}^{\tilde{M}_{u,i}} z_v^{u,i,j,P} + \sum_{j=1}^{\tilde{N}_{u,i}} z_v^{u,i,j,B} \right) \pi_{res}(f_{u,i}) \quad (32)$$

Define a binary variable $x_{u,l}$ indicating whether request u selects deployment scheme l . If $\sum_{l \in \mathcal{L}_u} x_{u,l} = 0$, request u is rejected. Then, P2 can be further reformulated as

$$\text{P3: } \max_x \sum_{u \in \mathcal{U}} \sum_{l \in \mathcal{L}_u} x_{u,l} h_u \quad (33)$$

$$\text{s.t. } \sum_{u \in \mathcal{U}} \sum_{l \in \mathcal{L}_u} g_{v,res}^{u,l} x_{u,l} \leq C_v^{res}, \quad \forall v \in V, \forall res \in \mathcal{R} \quad (34)$$

$$\sum_{l \in \mathcal{L}_u} x_{u,l} \leq 1, \quad \forall u \in \mathcal{U} \quad (35)$$

$$x_{u,l} \in \{0, 1\}, \quad \forall u \in \mathcal{U}, \forall l \in \mathcal{L}_u. \quad (36)$$

4.3. Primal-Dual Transformation

By relaxing the integrality constraint (36) to $x_{u,l} \in [0, 1]$, P3 becomes a linear programming (LP) problem. We associate dual variables $\phi_v^{res} \geq 0$ and $\mu_u \geq 0$ with constraints (34) and (35), respectively. The resulting Lagrangian dual problem, denoted by P4, is formulated as

$$\text{P4: } \min_{\phi, \mu} \sum_{v \in V} \sum_{res \in \mathcal{R}} \phi_v^{res} C_v^{res} + \sum_{u \in \mathcal{U}} \mu_u \quad (37)$$

$$\text{s.t. } \mu_u \geq h_u - \sum_{v \in V} \sum_{res \in \mathcal{R}} g_{v,res}^{u,l} \phi_v^{res}, \quad \forall u \in \mathcal{U}, \forall l \in \mathcal{L}_u \quad (38)$$

$$\phi_v^{res} \geq 0, \mu_u \geq 0, \quad \forall v \in V, \forall res \in \mathcal{R}, \forall u \in \mathcal{U}. \quad (39)$$

In the structure of P4, the dual variable ϕ_v^{res} can be interpreted as the unit marginal price (or shadow price) of resource res on node v . Consequently, the term $\sum_{v \in V} \sum_{res \in \mathcal{R}} g_{v,res}^{u,l} \phi_v^{res}$ in (38) represents the total resource cost of adopting scheme l for request u . Thus, constraint (38) ensures that μ_u is at least the net profit of the best deployment scheme for request u (i.e., revenue h_u minus the resource cost).

According to the complementary slackness condition of linear programming, if $x_{u,l} > 0$ in P3 (indicating that request u is accepted and scheme l is selected), the corresponding dual constraint (38) must be tight (i.e., equality holds). For each request u , the maximum attainable net profit is given by

$$\mu_u = \max \left\{ 0, \max_{l \in \mathcal{L}_u} \left\{ h_u - \sum_{v \in V} \sum_{res \in \mathcal{R}} g_{v,res}^{u,l} \phi_v^{res} \right\} \right\} \quad (40)$$

If $\mu_u > 0$, there exists at least one feasible deployment scheme yielding a positive net benefit for request u . In this case, the request is accepted, and the optimal scheme l^* with the maximum net benefit is selected:

$$l^* = \arg \max_{l \in \mathcal{L}_u} \left\{ h_u - \sum_{v \in V} \sum_{res \in \mathcal{R}} g_{v,res}^{u,l} \phi_v^{res} \right\} \quad (41)$$

If $\mu_u = 0$, the net profit is non-positive for all feasible deployment schemes, and request u is rejected.

In the online service chain deployment scenario, when a request arrives, the service provider must immediately decide whether to accept it and deploy it onto the physical infrastructure. Based on the above analysis, a deployment scheme that minimizes the total deployment cost should be sought to maximize the net profit. If the resulting net profit is non-negative, the request is accepted and deployed according to the optimal scheme. Otherwise, the request is rejected.

After a deployment decision is made, the resource utilization of the underlying physical infrastructure increases. According to market mechanisms, resource prices should adjust dynamically based on supply and demand. Within the primal-dual framework, this is realized by updating the dual variable ϕ_v^{res} .

The above analysis establishes a comprehensive framework for solving P1. Within this framework, the following key issues remain to be addressed:

- Price function design: The resource price critically determines the admission control threshold. If set too low, the system may prematurely accept low-value requests, depleting resources. Conversely, excessively high prices may reject high-value requests, leading to resource underutilization.
- Replica allocation decision: To satisfy reliability requirements, each VNF is implemented as a replica pool. An efficient approximation algorithm is required to determine the minimum number of primary and backup replicas that satisfy reliability constraints while minimizing resource consumption.
- Minimum-cost service chain deployment: Once the replica allocation is determined, an algorithm is needed to compute the minimum-cost deployment of the service chain onto the physical infrastructure, serving as the basis for admission decisions.

This section elucidates the complementary relationship between resource shadow prices and the net income of service chains, establishing that admission decisions are equivalent to an online threshold-based pricing mechanism. This dual perspective yields two key insights:

- Admission decisions can be made instantly upon request arrival provided that reasonable real-time prices for multidimensional resources are available.
- Identifying the lowest-cost deployment scheme within the feasible solution space is essential for maximizing the net profit predicted by the dual framework.

Consequently, the subsequent algorithm design is structured in two layers: the upper layer performs dynamic pricing and admission control, while the lower layer handles minimum-cost replica configuration and VNF mapping. The next section presents the approximation algorithm based on this theoretical foundation.

5. Algorithm Design

In the previous section, the overall framework for problem solving was established through linearization methods and primal-dual analysis. This section further proposes an online deployment algorithm for multi-replica SFCs with guaranteed reliability, which is termed GRM-SFCOD (Guaranteed Reliability Multi-replicas SFC Online Deployment). The GRM-SFCOD algorithm comprises a series of subalgorithms, including the service chain request admission control algorithm, the replica allocation decision algorithm, and the VNF node mapping algorithm based on a genetic algorithm.

5.1. Cost Function Design

In economics, pricing mechanisms serve as a vital tool for regulating the relationship between supply and demand. The price of a commodity is typically determined by production costs, market demand, and the competitive environment. When supply falls short of demand, prices rise, thereby curbing demand and incentivizing producers to increase supply; conversely, when supply exceeds demand, prices fall, stimulating demand while reducing supply [46]. This dynamic adjustment mechanism effectively balances market resources, preventing excessive concentration or waste. Building on the primal-dual analysis from the previous section, this subsection introduces the concept of deployment cost and designs a corresponding resource cost function [47]. This function guides service providers in screening incoming requests and proactively discarding certain SFC requests that are resource-intensive but yield limited profit, thereby enhancing long-term SFC deployment revenue. The function is required to satisfy three criteria:

- Additivity of resources: Resources from different dimensions can be linearly superimposed to form the total cost.
- Scarcity sensitivity: The closer a resource is to its bottleneck, the higher its unit price should be.
- Online Updatability: It only relies on the current utilization rate ρ_v^{res} and does not require prior knowledge of future loads.

Under these principles, the price function is designed as follows. First, we define the resource weight value α_v^{res} , which reflects the relative scarcity of the resource res :

$$\alpha_{res} = \frac{C^{cpu} + C^{mem} + C^{disk}}{C^{res}}, \quad \forall res \in \mathcal{R}. \quad (42)$$

Here, $C^{res} = \sum_{v \in V} C_v^{res}$ represents the total capacity of resource res across all nodes. If the capacity of a certain resource type is large, then its α_{res} value is smaller, indicating that the unit cost of this resource is lower and its impact on resource allocation is weaker. Conversely, if the capacity of a certain resource type is small, then α_{res} is larger, indicating that this resource is scarcer and may become the main bottleneck of the system. Therefore, the use of such resources should be restricted first. By assigning higher weights to globally scarcer resources, this scheme ensures their prices increase more aggressively as utilization grows, preventing a premature depletion of bottleneck dimensions.

Define γ_u as the value density of service chain request u , which is the ratio of the service chain's value to the total weighted resource consumption:

$$\gamma_u = \frac{h_u}{\sum_{res \in \mathcal{R}} \alpha_{res} \sum_{i=1}^{|F_u|} (M_{u,i} + N_{u,i}) \pi_{res}(f_{u,i})} \quad (43)$$

Based on this, define θ as the fluctuation range of the value density: This metric enables a comparison of heterogeneous requests with different resource demands and revenues,

allowing the pricing mechanism to discriminate between high-value and low-value requests during admission control.

$$\theta = \frac{\max_{u_1 \in \mathcal{U}} \gamma_{u_1}}{\min_{u_2 \in \mathcal{U}} \gamma_{u_2}}, \quad \forall u_1, u_2 \in \mathcal{U}. \quad (44)$$

The parameter θ directly influences admission control stringency by reflecting the heterogeneity of request revenues across the population.

Let ρ_v^{res} denote the current utilization rate of resource res at node v . Based on this, we define the normalized utilization rate $\bar{\rho}_v^{res}$:

$$\bar{\rho}_v^{res} = \lfloor \rho_v^{res} \cdot \log_2(\alpha_{res}\theta) \rfloor. \quad (45)$$

Based on the normalized utilization rate, the unit price of the resource res on node v is defined as an exponential function:

$$\phi_v^{res} = 2^{\bar{\rho}_v^{res}} - 1 \quad (46)$$

The exponential form is chosen for its self-adaptive property: it eliminates the need for manually tuned thresholds while ensuring prices remain low at normal utilization yet rise sharply near capacity. The use of such a cost function is based on the following considerations: When the resource utilization is low, the price increases slowly, encouraging the reception of more SFCs to improve resources utilization. When the resource utilization approaches the capacity limit, the price increases more rapidly, thereby avoiding the excessive use of scarce resources. The dynamic adjustment factor θ is used to adapt to the heterogeneity of task requirements. Specifically, a smaller θ indicates more homogeneous request values, resulting in lower resource prices and more lenient admission; conversely, a larger θ reflects higher revenue heterogeneity, triggering elevated prices that reserve scarce resources for high-value requests, thereby improving long-term revenue but potentially lowering the admission ratio in early stages. When the value density of the arriving service chain requests fluctuates greatly, the price is raised, thereby reserving more resources for higher-value requests; when the value density fluctuates less, the price is lowered, thereby improving system resource utilization. In addition, prices are set separately for different resources dimensions. Scarce resources will be assigned higher prices, guiding VNF instances to be deployed to nodes with a more matching resource structure. Compared to the commonly used First-Come-First-Serve (FCFS) strategy, this price mechanism can dynamically adjust the configuration priority based on the scarcity degree of each resource dimension, preventing a single dimension from being depleted prematurely, thereby improving overall system resource utilization.

5.2. Online Admission Control Policy

Based on the pricing function obtained in the previous subsection, the resource cost of any candidate deployment scheme $l \in \mathcal{L}_u$ can be calculated. According to the primal-dual analysis, the admission decision for service chain requests can be reduced to a threshold comparison. This subsection details the complete online admission control algorithm, outlined in Algorithm 1.

Firstly, the initial fluctuation range of resource density θ is initialized to 1. For each arriving service chain deployment request u , an attempt is made to allocate it to the underlying physical network. For a deployment scheme $\eta_v^{u,i,j,P}$ (indicating whether the j -th primary replica of $f_{u,i}$ is deployed on node v) and $\eta_v^{u,i,j,B}$ (indicating whether the j -th backup

replica of $f_{u,i}$ is deployed on node v), the resource deployment cost for each dimension of service chain u is calculated as follows:

$$cost_u^{res} = \sum_{v \in V} \sum_{f_{u,i} \in F_u} \left(\sum_{j=1}^{M_{u,i}} \eta_v^{u,i,j,P} + \sum_{j=1}^{N_{u,i}} \eta_v^{u,i,j,B} \right) \alpha_{res} \cdot \pi_{res}(f_{u,i}) \cdot \phi_v^{res} \quad (47)$$

The cost-minimization algorithm for service chain deployment is used to solve the minimum cost $\widetilde{cost}_u = \sum_{res \in \mathcal{R}} \widetilde{cost}_u^{res}$ required for deploying this service chain. When the service chain's benefit is greater than or equal to the cost, i.e., $h_u \geq \widetilde{cost}_u$, the task is accepted and the service chain is deployed to the underlying physical network according to the pre-assigned scheme; otherwise, the request is rejected.

After the admission decision is made, the value density γ is calculated based on the latest arriving request, and θ is updated to reflect the value fluctuations of the currently processed tasks.

Algorithm 1: Service Chain Request Admission Control Algorithm

Input : Request u , resource utilization ρ_v^{res} ($\forall v \in V, \forall res \in \mathcal{R}$), threshold θ

Output: Admission decision δ_u

```

1 Initialization:  $\delta_u \leftarrow 1$ ;
2 foreach  $res \in \mathcal{R}$  do
3    $\alpha_v^{res} \leftarrow \frac{C^{cpu} + C^{mem} + C^{disk}}{C^{res}}$ ;
4    $\bar{\rho}_v^{res} \leftarrow \bar{\rho}_v^{res} = \lfloor \rho_v^{res} \cdot \log_2(\alpha_{res}\theta) \rfloor$ ;
5    $\phi_v^{res} \leftarrow 2^{\bar{\rho}_v^{res}} - 1$ ;
6 end foreach
7 Invoke the cost-minimization service chain deployment algorithm to obtain
    $M_{u,i}, N_{u,i}, \eta_v^{u,i,j,B}, \eta_v^{u,i,j,P}$ ;
8 foreach  $res \in \mathcal{R}$  do
9    $cost_u^{res} \leftarrow \sum_{v \in V} \sum_{f_{u,i} \in F_u} \left( \sum_{j=1}^{M_{u,i}} \eta_v^{u,i,j,P} + \sum_{j=1}^{N_{u,i}} \eta_v^{u,i,j,B} \right) \alpha_{res} \cdot \pi_{res}(f_{u,i}) \cdot \phi_v^{res}$ ;
10 end foreach
11 if  $h_u \geq cost_u^{cpu} + cost_u^{mem} + cost_u^{disk}$  then
12    $\delta_u \leftarrow 1$ ;
13 else
14    $\delta_u \leftarrow 0$ ;
15 end if
16 Update the resource utilization rate  $\rho_v^{res}$  on all nodes and the current task value
   density  $\theta$ 
17 return  $\delta_u$ ;
```

5.3. Service Chain Deployment Cost Minimization Algorithm

The previous section presented a server node resource cost model that dynamically adjusts based on utilization rates and, accordingly, established a threshold-based service chain request admission algorithm. Upon the arrival of a service chain request, the system first attempts a pre-allocation within the network to determine the minimum resource cost required for deployment. Based on this calculated cost, an admission threshold is set to decide whether to accept the request. This subsection further designs the service chain cost minimization algorithm, which consists of two components: the replica allocation decision algorithm and the VNF node mapping algorithm.

5.3.1. Replica Allocation Decision Algorithm

Each network function in the service chain is replaced by a replica pool consisting of primary and backup replicas, which achieves load balancing within the network and mitigates resource fragmentation. When a primary replica fails, the backup replica can be rapidly activated to take over the service, thereby avoiding service interruption and enhancing the fault tolerance of the SFC.

Adding redundant VNF replicas to an existing service chain can improve the overall reliability of the service chain. However, since network resources are limited, blindly increasing the number of replicas leads to resource waste. Moreover, as indicated by Equation (3), the overall reliability of an SFC is a function of the reliability of all its constituent VNFs. Therefore, the key challenge lies in identifying which VNFs are most suitable for adding backup replicas to maximize resource efficiency. Accordingly, one of the core issues in SFC deployment is how to precisely determine the number of primary and backup replicas for each VNF based on specific reliability requirements, thereby achieving an effective balance between reliability assurance and resource consumption optimization.

The number of primary replicas for each network function in the service chain is determined by the bandwidth requirements of the service chain. Specifically, the number of primary replicas is the minimum number required to satisfy the bandwidth requirements of the service chain, as given in Equation (1).

The number of backup replicas is determined by the reliability requirements of the service chain. According to Equations (2) and (3), the following inequality can be obtained:

$$\prod_{f_{u,i} \in F_u} R(f_{u,i}) \geq R_u \quad (48)$$

To simplify the computation, taking the logarithm on both sides of the inequality yields

$$\sum_{f_{u,i} \in F_u} \log_2 R(f_{u,i}) \geq \log_2 R_u \quad (49)$$

Since the marginal contribution of each backup replica to the overall reliability of the service chain is not the same, high-efficiency backup replicas should be prioritized for deployment when resources are limited to optimize resource allocation. To this end, we define the utility gain of the service chain backup replicas to measure the marginal contribution of each backup replica to the overall reliability of the service chain:

$$G(f_{u,i}, m, n) = \begin{cases} \log_2 R(f_{u,i}, m, n) - \log_2 R(f_{u,i}, m, n-1), & n \geq 1 \\ \log_2 R(f_{u,i}, m, 0), & n = 0 \end{cases} \quad (50)$$

In this definition, $R(f_{u,i}, m, n)$ denotes the reliability of network function $f_{u,i}$ consisting of m primary replicas and n backup replicas. For $n \geq 1$, $G(f_{u,i}, m, n)$ represents the utility gain of the n -th backup replica for $f_{u,i}$ with m primary replicas, which measures the marginal benefit of adding this replica to the overall reliability of the SFC. For $n = 0$, $G(f_{u,i}, m, n)$ indicates the benefit of all primary replicas of the network function to service chain reliability.

Evidently, the sum of the utility gains contributed by all backup replicas of all network functions within the service chain equals the logarithmic value of the overall reliability of the service chain.

The following analysis examines the monotonically decreasing property of the reliability gain function of backup replicas defined in Equation (50). To establish this property, we proceed in two steps. First, we prove that the probability mass function of the binomial

distribution is log-concave (Lemma 1). Second, we leverage the fact that the cumulative distribution function of a log-concave probability mass function is also log-concave [48], and we show that this inherited log-concavity implies the monotonic decrease of the marginal reliability gain (Theorem 1).

Lemma 1. *The probability mass function of the binomial distribution $P(k) = \binom{n}{k}p^k(1-p)^{n-k}$ is log-concave with respect to k . For all $k \geq 1$, it satisfies $P^2(k) \geq P(k-1)P(k+1)$.*

Proof of Lemma 1. To prove that for all values of k , the following holds:

$$P^2(k) \geq P(k-1)P(k+1) \quad (51)$$

Substituting the expression of the binomial distribution yields

$$\left(\binom{n}{k}p^k(1-p)^{n-k}\right)^2 \geq \binom{n}{k-1}p^{k-1}(1-p)^{n-(k-1)} \cdot \binom{n}{k+1}p^{k+1}(1-p)^{n-(k+1)} \quad (52)$$

After eliminating the common factors, the combination relationship can be obtained:

$$\binom{n}{k}^2 \geq \binom{n}{k-1} \binom{n}{k+1} \quad (53)$$

By using the properties of combinations, the following relationship exists:

$$\binom{n}{k+1} = \binom{n}{k} \frac{n-k}{k+1}, \quad \binom{n}{k-1} = \binom{n}{k} \frac{k}{n-k+1} \quad (54)$$

After substituting into the original inequality, it can be transformed into

$$\binom{n}{k}^2 \geq \binom{n}{k}^2 \cdot \frac{k(n-k)}{(k+1)(n-k+1)} \quad (55)$$

Further simplified to

$$\frac{k(n-k) + n + 1}{k(n-k)} \geq 1 \quad (56)$$

In this inequality, both the denominator and the numerator are greater than 0, and the numerator is greater than the denominator. Therefore, this inequality holds true. The proof is complete. $\square$

Theorem 1. *For all $n \geq 0$, $G(f_{u,i}, m, n)$ decreases monotonically as n increases.*

Proof of Theorem 1. To simplify the expression, the utility gain $G(f_{u,i}, m, n)$ is abbreviated as $G(n)$, and the reliability $R(f_{u,i}, m, n)$ of a single network function consisting of m primary replicas and n backup replicas is abbreviated as R_n . This is equivalent to proving

$$G(n+1) - G(n) = (\log_2 R_{n+1} - \log_2 R_n) - (\log_2 R_n - \log_2 R_{n-1}) = \log_2 \left(\frac{R_{n+1}R_{n-1}}{R_n^2} \right) \leq 0 \quad (57)$$

That is to say,

$$R_{n+1}R_{n-1} \leq R_n^2 \quad (58)$$

Therefore, it is necessary to prove that R_n is log-concave with respect to n .

According to Lemma 1, the probability mass function of the binomial distribution is log-concave. Furthermore, according to [48], if the probability mass function of an integer random variable is log-concave on $\mathbb{Z}$, then its cumulative distribution function on $\mathbb{Z}$ is also log-concave. Therefore, the cumulative distribution function $F(k) = \sum_{i=0}^k \binom{n}{i}p^i(1-p)^{n-i}$

of the binomial distribution is log-concave at k . Then, for a fixed m and $p = 1 - r_{u,i}$, the sequence R_n is log-concave on n . In conclusion, the original theorem holds true and the proof is complete. $\square$

Physically, Theorem 1 indicates that as the number of backup replicas of the same network function gradually increases, the overall reliability gain of the service chain provided by the newly added replicas decreases. The reliability R_n measures the probability that at least m out of $m + n$ replicas are functional. When n is small, adding one backup replica significantly increases this probability. However, as n grows larger, the binomial distribution becomes increasingly concentrated around its mean, and the incremental gain from each additional replica diminishes. Consequently, the marginal utility gain $G(n)$ decreases monotonically, ensuring that the greedy allocation strategy—which always selects the replica with the highest marginal gain—yields an optimal resource allocation.

Guided by this principle, our algorithm proceeds as follows: When adding VNF backup replicas, for each subsequent backup replica of each VNF, we calculate the ratio of its utility gain to its resource consumption to determine the deployment priority:

$$\frac{G(f_{u,i}, m, n)}{\sum_{res \in \mathcal{R}} \alpha_{res} \pi_{res}(f_{u,i})} \quad (59)$$

In this formula, α_{res} represents the resource weight value defined in the previous section, and this value reflects the differences among resources in different dimensions.

Select the backup replica with the highest ratio of utility gain to resource consumption and add it to the function until the overall reliability of the service chain meets the predetermined requirements. This will determine the number of backup replicas for each network function. Since the gain function has a decreasing nature, the optimality of the gain-to-resource ratio (i.e., the ratio of utility gain to resource consumption) of each added backup replica can be maintained throughout the subsequent process, as summarized in Algorithm 2.

By sequentially adding backup replicas with the highest gain-to-resource ratio, it is possible to ensure the reliability required by the service chain requests while minimizing the consumption of underlying physical resources, thereby achieving a balance between reliability and resource efficiency.

5.3.2. Genetic Algorithm-Based Virtual Network Function Mapping Algorithm

After obtaining the replica allocation scheme for each network function in the service chain, it is necessary to map each replica instance onto the server nodes of the underlying physical network while minimizing deployment costs. This subproblem can be formulated as

$$P2: \min_{\delta, \eta} \text{cost}_u^{cpu} + \text{cost}_u^{mem} + \text{cost}_u^{disk} \quad (60)$$

s.t. (42)–(46)

$$\sum_{i=1}^{|F_u|} \left(\sum_{j=1}^{M_{u,i}} \eta_v^{u,i,j,P} + \sum_{j=1}^{N_{u,i}} \eta_v^{u,i,j,B} \right) \pi_{res}(f_{u,i}) \leq C_v^{res} - C_v^{res,used}, \quad \forall v \in V, \forall res \in \mathcal{R} \quad (61)$$

$$\sum_{v \in V} \eta_v^{u,i,j,P} = 1, \forall f_{u,i} \in F_u, \forall j \in [1, M_{u,i}] \quad (62)$$

$$\sum_{v \in V} \eta_v^{u,i,j,B} = 1, \forall f_{u,i} \in F_u, \forall j \in [1, N_{u,i}] \quad (63)$$

$$\eta_v^{u,i,j,P} \in \{0, 1\}, \forall i \in [1, |F_u|], \forall j \in \mathbb{Z}^+ \quad (64)$$

$$\eta_v^{u,i,j,B} \in \{0, 1\}, \forall i \in [1, |F_u|], \forall j \in \mathbb{Z}^+ \quad (65)$$

In these constraints, (42)–(46) define the node resource cost functions; (61) represents the node resource capacity constraints, where $C_v^{res,used}$ denotes the amount of used resource res on node v ; (62) and (63) ensure that each primary and backup replica is deployed on exactly one node; (64) and (65) are the binary indicator variable constraints.

Algorithm 2: Backup Replica Allocation Decision Algorithm

Input : Service chain request u : It includes bandwidth requirement B_u , reliability requirement R_u , and network function sequence $F_u = [f_{u,1}, f_{u,2}, \dots, f_{u,i}, \dots]$.
 Network Function Set $\mathcal{F}$: For each network function $f \in \mathcal{F}$, the bandwidth $b(f)$ and reliability $r(f)$.

Output: The number of primary replicas $m[i]$ and the number of backup replicas $n[i]$ for each network function $f_{u,i} \in F_u$.

```

1 Initialization:  $m[\cdot], n[\cdot], G[\cdot] \leftarrow$  empty arrays
2 for  $f_{u,i} \in F_u$  do
3    $m[i] \leftarrow \left\lceil \frac{B_u}{b(f_{u,i})} \right\rceil$ ;
4    $n[i] \leftarrow 0$ ;
5   if  $n[i] > 0$  then
6      $G[i] \leftarrow \log_2 R(f_{u,i}, m[i], n[i]) - \log_2 R(f_{u,i}, m[i], n[i] - 1)$ ;
7   else
8      $G[i] \leftarrow \log_2 R(f_{u,i}, m[i], 0)$ ;
9   end if
10 end for
11  $R(SC_u) \leftarrow \prod_{f_{u,i} \in F_u} R(f_{u,i}, m[i], n[i])$ ;
12 while  $R(SC_u) < R_u$  do
13    $i^* \leftarrow \arg \max_i \frac{G[i]}{\sum_{res \in \mathcal{R}} \alpha_{res} \pi_{res}(f_{u,i})}$ ; // Find the backup replica with the
      highest gain-to-resource-demand ratio.
14    $n[i^*] \leftarrow n[i^*] + 1$ ; // Increase the number of backup replicas.
15    $G[i^*] \leftarrow \log_2 R(f_{u,i^*}, m[i^*], n[i^*]) - \log_2 R(f_{u,i^*}, m[i^*], n[i^*] - 1)$ ; // Update
      gain
16   Update  $R(SC_u)$ ;
17 end while
18 return  $(m[\cdot], n[\cdot])$ ;
```

By observing the optimization problem, it can be identified as an ILP problem. Therefore, optimization solvers such as CPLEX or Gurobi can be employed to obtain the optimal solution. Upon further analysis, if the multi-resource constraint problem is simplified into a single-resource constraint problem, it can be reduced to the Multiple Knapsack Problem. Since the Multiple Knapsack Problem has been proven to be NP-hard, problem P2 is also NP-hard. To find an approximate solution within a reasonable time, this subsection designs a Genetic Algorithm-based Virtual Network Function Mapping Algorithm.

The core idea of the Genetic Algorithm is to maintain a set of candidate solutions (a population of chromosomes). In each generation, new solutions are generated through operations such as natural selection, crossover, and mutation, and the quality of solutions is progressively improved via the principle of ‘survival of the fittest’. Specifically, the design of a Genetic Algorithm typically encompasses the following steps: chromosome encoding scheme design, population initialization, fitness function design, selection operation, crossover between superior individuals, mutation, and termination condition. The specific design of each component is described below.

- **Chromosome Encoding Scheme**
Each chromosome represents a complete VNF mapping scheme. A chromosome is encoded as an integer vector whose length equals the total number of VNF instances required by service chain request u . The genes in the chromosome are arranged in order of VNF type i (from 1 to $|F_u|$). For each type i , the genes corresponding to its $M_{u,i}$ primary replicas are arranged first, which are followed by the genes corresponding to its $N_{u,i}$ backup replicas. The value of the k -th gene is a positive integer representing the ID of the physical server node to which the k -th VNF instance is mapped with a value range of $[1, |V|]$.
- **Population Initialization**
After determining the encoding scheme, an initial population of solutions should be initialized. When generating the initial solutions, it is crucial to ensure population diversity as much as possible, enabling subsequent iterations to fully explore the solution space. To this end, this subsection adopts a random initialization strategy: for each replica, traverse all nodes to identify feasible nodes that can host the replica without exceeding resource limits and include them in a set of candidate nodes. After completing the traversal, a node is randomly selected from the candidate set for allocation.
- **Fitness and Population Selection Strategy**
The setting of the fitness function aligns with the current optimization objective, reflecting the “degree of fitness” between the current solution and the optimization goal. The fitness function for the algorithm is defined as follows:

$$FF(P_r^m) = -(k+1)(cost_u^{cpu} + cost_u^{mem} + cost_u^{disk}) \quad (66)$$

Here, P_r^m denotes the m -th individual in the population of the r -th generation, and $0 \leq k \leq |V|$ represents the number of overloaded servers under this placement scheme. The fitness function takes negative values to evaluate the quality of a VNF mapping solution. When the deployment cost is high, the calculated fitness is lower, thereby guiding the algorithm to search for deployment schemes with lower costs. Meanwhile, if resource overloading occurs on certain server nodes in a deployment scheme, the original deployment cost is multiplied by the number of overloaded nodes, k . This significantly reduces the fitness value of that scheme. Through this mechanism, the algorithm severely penalizes excessive demand, progressively eliminates infeasible solutions during the evolutionary process, and ensures that the final iterative result converges to a deployment scheme with no resource overloading. The Genetic Algorithm evolves the population through a mechanism of survival of the fittest. In each generation, the algorithm selects individuals from the original population into the offspring based on the tournament scheme [49]. Each selection process constitutes a tournament: a specific number of individuals are chosen from the original population with replacement, and the one with the highest fitness is declared the winner to participate in subsequent evolutionary operations.

- **Crossover and Mutation**
After obtaining two parents through the selection operation, the algorithm combines the genetic information of the parents via crossover and introduces randomness into the solution space through mutation, thereby expanding the search scope and ultimately generating new offspring. This algorithm employs the uniform crossover method. For each gene of the offspring, a number $\mu \in [0, 1]$ is uniformly gener-

ated, and the parent from which the gene is inherited is determined based on the following condition:

$$child[i] \leftarrow \begin{cases} parent1[i], & \mu \geq cr_rate \\ parent2[i], & \text{otherwise} \end{cases} \quad (67)$$

where $parent1[i]$ and $parent2[i]$ denote the i -th gene values of the parent 1 and parent 2 chromosomes, respectively, i.e., the IDs of the server nodes where the i -th instance is placed. $child[i]$ represents the gene value of the offspring chromosome at this position. $cr_rate \in [0, 1]$ is the crossover rate, which is set to 0.5 in the specific implementation of this algorithm, thereby enabling the offspring chromosome to inherit genes from the parents with uniform probability.

Following the crossover operation to generate new offspring, a mutation operation is performed on each offspring to thoroughly explore the solution space. During the mutation process, a VNF replica is randomly selected from the deployment scheme. Then, a node is randomly chosen from the set of nodes—excluding the one to which the replica was originally assigned—that possesses sufficient remaining resource capacity to meet the replica’s requirements. The replica is then remapped to this newly selected node. This mutation mechanism enhances population diversity during each iteration and prevents the convergence results from getting trapped in local optima. Assuming that an SFC containing two VNFs needs to be deployed, Figure 2 illustrates a process of genetic evolution and mutation.

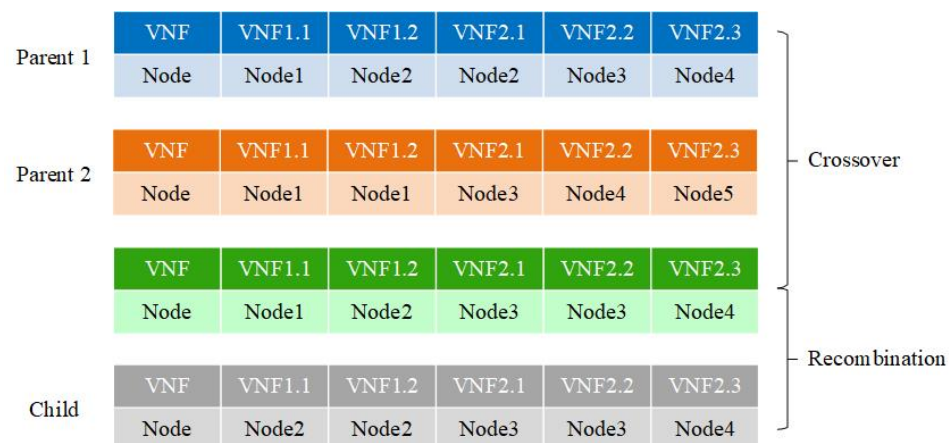


Figure 2. Example of genetic and mutation process.

- Elitism Strategy**
 To prevent the loss of the optimal individual during the iteration process, after generating a new population in each iteration, the two individuals with the lowest fitness in the new population are replaced by the two individuals with the highest fitness from the previous generation. This ensures that the algorithm can converge to the optimum more rapidly [50].
- Termination Criteria**
 If the termination condition is set too loosely, it may lead to “premature convergence” and poor solution quality. Conversely, if the condition is too strict, it will result in excessive running time. Therefore, this algorithm employs two termination criteria: the algorithm terminates when it reaches the maximum number of generations, $max_generations$, or when the improvement in the best fitness value over $no_improve_limit$ consecutive generations is lower than the convergence threshold $\delta_{fitness}$.

Algorithm 3 presents the complete pseudocode of the GA-based VNF mapping procedure. Figure 3 illustrates a representative convergence curve under the parameter settings detailed in Section 6.1. As shown in the figure, the best fitness value improves rapidly in early generations, progressively refines during the middle phase, and stabilizes at approximately generation 175—well before the $max_generations = 300$ limit. This behavior validates the effectiveness of the dual termination criteria: the $no_improve_limit$ mechanism acts as the dominant terminating condition in practice, ensuring that the algorithm does not run unnecessarily once the solution has converged.

Algorithm 3: Genetic Algorithm-based VNF Mapping Algorithm

Input : Physical Network: $G = (V, E)$; Resource utilization rate of server node: ρ_v^{res} ; SFC request: u ; Number of primary and backup replicas: $M_{u,i}, N_{u,i}$.

Output: VNF Mapping Scheme

```

1 Randomly generate the initial population  $P_0$ ;
2 Calculate the fitness of the initial population  $fitness\_list$ ;
3 Initialize the best fitness  $best\_fitness \leftarrow -\infty$ , the best solution  $best\_solution \leftarrow \emptyset$ ,
  and the no-improvement counter  $no\_improve\_count \leftarrow 0$ ;
4 while  $iteration\_count < max\_generations$  AND
   $no\_improve\_count < no\_improve\_limit$  do
5   Record the best individual and fitness of the current population:
      $best\_current \leftarrow \max(fitness\_list)$ ;
6   if  $best\_current > best\_fitness + \delta_{fitness}$  then
7      $best\_fitness \leftarrow best\_current$ ,  $best\_solution \leftarrow$  corresponding individual,
      $no\_improve\_count \leftarrow 0$ ;
8   else
9      $no\_improve\_count \leftarrow no\_improve\_count + 1$ ;
10  end if
11  Initialize new population:  $new\_population \leftarrow \emptyset$ ;
12  for  $i \leftarrow 1$  to  $pop\_size/2$  do
13    Select two parents  $parent1, parent2$  from  $P$  via tournament selection;
14    Generate two children  $child1, child2$  via uniform crossover;
15    Perform mutation on  $child1, child2$  by randomly reassigning a selected
      gene;
16    Add  $child1, child2$  to  $new\_population$ ;
17  end for
18  Replace the two individuals with the lowest fitness in  $new\_population$  with the
    two individuals with the highest fitness from  $P$ ;
19  Calculate  $new\_fitness\_list$ , update
      $P \leftarrow new\_population$ ,  $fitness\_list \leftarrow new\_fitness\_list$ ;
20 end while
21 if  $number\_of\_overloaded\_servers > 0$  then
22    $best\_solution \leftarrow \emptyset$ , deployment failed;
23 else
24   return  $best\_solution$ ;
25 end if

```

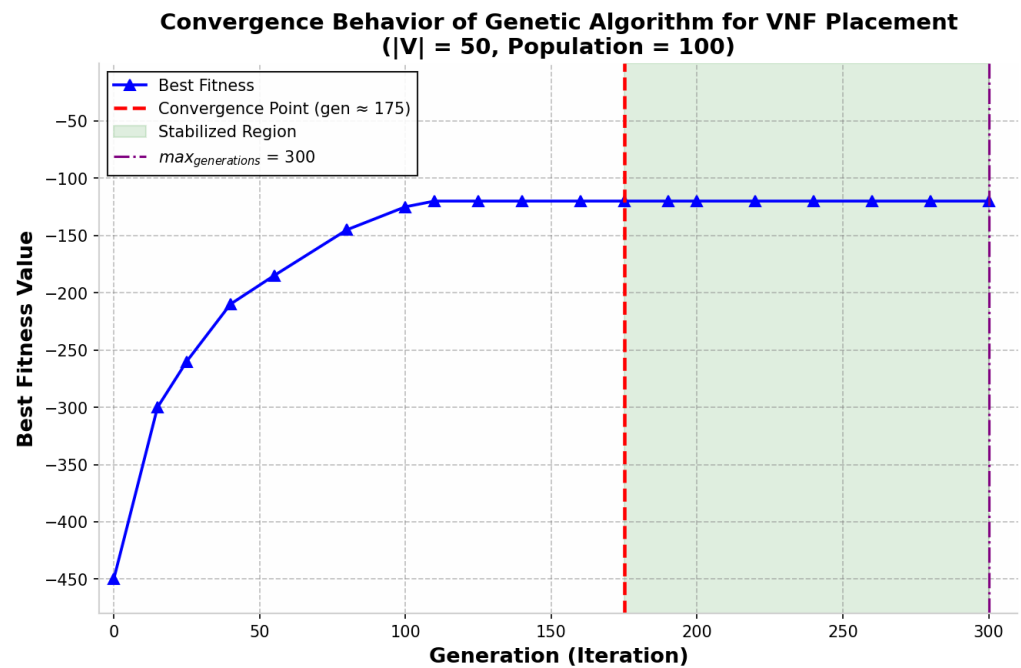


Figure 3. Convergence behavior of genetic algorithm for VNF placement.

5.4. Summary of the Algorithm Procedure

Based on the subalgorithm designs described above, this subsection summarizes the overall workflow of the GRM-SFCOD algorithm, as illustrated in Figure 4. The specific steps are as follows.

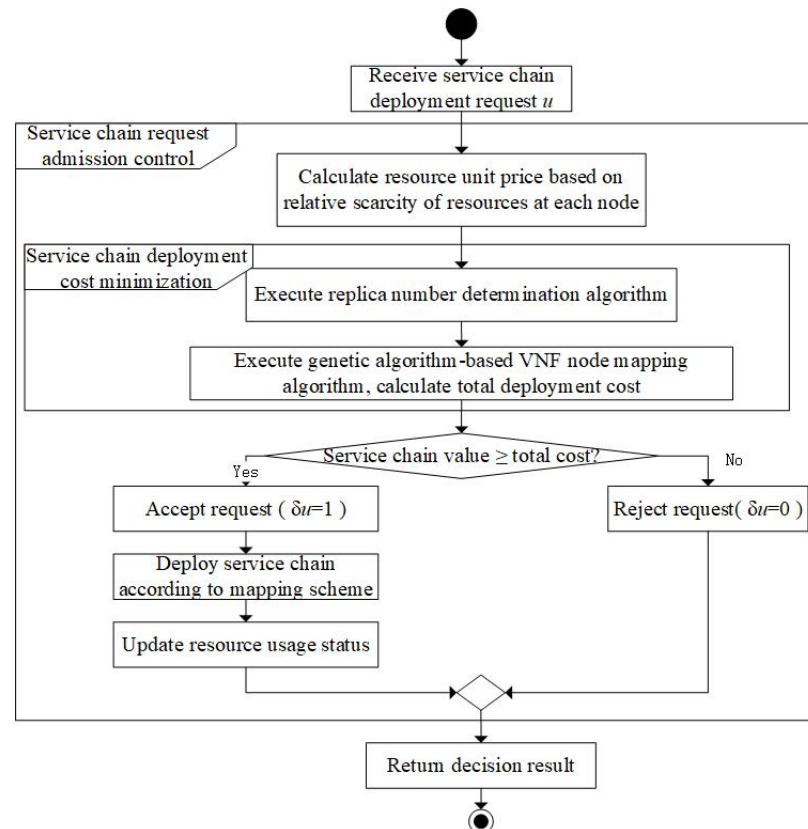


Figure 4. Workflow of the GRM-SFCOD algorithm.

- Request Arrival and Pricing: Receive the service chain requests arriving at the current time slot and calculate the unit resource price based on the relative scarcity of resources on each node.
- Replica Determination: Execute the replica number determination algorithm to decide the number of primary and backup replicas for each VNF in the service chain to be deployed, ensuring that reliability requirements are met.
- VNF Placement: Based on the calculated number of replicas, execute the GA-based VNF node mapping algorithm to determine the actual deployment location for each VNF replica instance and calculate the total deployment cost.
- Admission Control: Compare the value of the current service chain, h_u , with the total deployment cost calculated in the previous step. If the service chain value is greater than or equal to the deployment cost, accept the request. Otherwise, reject the request. If accepted, deploy the service chain according to the mapping scheme and update the current network resource usage.
- Result Feedback: Return the service chain deployment decision result to the user.

6. Simulation Experiments

To evaluate the performance of the GRM-SFCOD algorithm proposed in this section, a series of numerical simulation experiments were conducted in this subsection.

6.1. Experimental Parameters

All simulation algorithms in this section were implemented in Python 3.11 with results analyzed using MATLAB R2025b. Experiments were conducted on a host machine equipped with an Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz and 32 GB of RAM. The network topology $G = (V, E)$ was generated using the GT-ITM random topology generator [51].

Regarding resource configuration, the CPU and memory capacities for each server node follow a uniform distribution over [50, 100]. This range is selected to mirror the standard high-performance physical servers or aggregated resource pools typically found in data centers (e.g., nodes with approximately 64–128 cores or equivalent processing units). Similarly, disk storage capacity is uniformly distributed between [500, 1000] GB, aligning with common enterprise-grade solid-state drive (SSD) specifications in modern cloud infrastructure. For service chains, bandwidth requirements follow a uniform distribution over [5, 20] Mbps; this interval reflects the traffic characteristics of typical microservices and lightweight chains, which generally demand moderate throughput compared to bandwidth-intensive applications like video streaming.

Although real-world resource availability often exhibits skewed patterns (e.g., normal or Pareto distributions), this paper adopts a uniform distribution to establish a neutral baseline. This approach avoids introducing bias toward specific resource hotspots, thereby facilitating a generalized evaluation of the algorithm's robustness across diverse simulated network conditions.

The reliability requirement is randomly selected with equal probability from the set {0.99, 0.999, 0.9999}. The source and destination nodes of a request are randomly selected from all nodes in the network. The number of VNFs contained in each SFC request is uniformly distributed within [1, 5], and the value of an SFC is randomly generated within [1, 60]. It is assumed that there are 10 different types of VNFs in the network. For each VNF, the CPU demand follows a uniform distribution within [1, 5], the memory demand follows a uniform distribution within [1, 5], and the disk space demand follows a uniform distribution within [1, 50]. The reliability of a VNF instance is set to follow a uniform distributed within [0.9, 0.99] [52]. The VNF sequence contained in each SFC

request is composed of VNFs with distinct types, which are randomly drawn from the 10 available types.

Furthermore, in the GRM-SFCOD algorithm, the population size of the Genetic Algorithm is set to 100, and the maximum number of iterations is set to 300. The threshold for consecutive generations without fitness improvement, denoted as *no_improve_limit*, is set to 20, while the convergence threshold $\delta_{fitness}$ is set to 10^{-6} . The crossover rate is set to 0.5, and the mutation rate is set to 0.1. These settings are consistent with those used for the illustrative convergence analysis presented in Figure 3. The pricing parameter θ , which captures the fluctuation range of request value densities (Equation (44)), directly influences the admission control stringency: a smaller θ indicates more homogeneous request values, resulting in lower resource prices and more lenient admission, which may lead to a premature acceptance of low-value requests and reduced long-term revenue; conversely, a larger θ reflects higher revenue heterogeneity, triggering elevated prices that reserve scarce resources for high-value requests, thereby improving revenue but potentially lowering the admission ratio in early stages. The exponential pricing function (Equation (46)) inherently adapts to these variations by dynamically scaling prices based on real-time resource utilization, mitigating excessive sensitivity to the exact value of θ . The dual termination criteria ensure the algorithm stops upon solution stabilization, and the chosen population size and crossover rate represent standard settings in Genetic Algorithm design, striking a balance between solution quality and computational efficiency.

6.2. Analysis of Experimental Results

6.2.1. Performance Evaluation of the Replica Number Determination Algorithm

To validate the effectiveness of the proposed replica number determination mechanism based on reliability marginal gain (detailed in Section 5.3.1), we compare the GRM-SFCOD algorithm against two heuristic baseline algorithms. Since the reliability constraint in (5) is nonlinear and computationally expensive for traditional solvers, these baselines serve as practical benchmarks:

- Heu.1: Decompose the overall service chain reliability requirement among individual VNFs. For each VNF, first determine the number of primary replicas based on bandwidth, and then sequentially increase the number of backup replicas until Equation (68) is satisfied.

$$R(f_{u,i}) \geq R_u \frac{1}{|F_u|} \quad (68)$$

- Heu.2: First, determine the number of primary replicas for each VNF based on bandwidth. Then, iteratively traverse each VNF in the service chain and incrementally increase backup replicas until the global reliability target is met without prioritizing specific nodes based on marginal gain.

By replacing the core decision module of the GRM-SFCOD algorithm with Heu.1 and Heu.2, respectively, we establish two comparative schemes, which are denoted as Baseline-1 and Baseline-2. The simulation environment consists of 200 network nodes. To isolate the impact of redundancy strategies and prevent request rejection due to resource exhaustion, the total number of service function chain (SFC) requests is capped at 200.

Figure 5 illustrates the total deployment cost of SFCs using different redundancy backup strategies. As expected, the deployment cost for all three algorithms exhibits a monotonic increase with the request volume, which is driven by higher cumulative resource consumption. However, the proposed GRM-SFCOD algorithm consistently outperforms both baselines, maintaining the lowest cost curve across the entire range. Specifically, under heavy load (200 requests), GRM-SFCOD reduces the total deployment cost by 9.68% and 7.31% compared to Baseline-1 and Baseline-2, respectively.

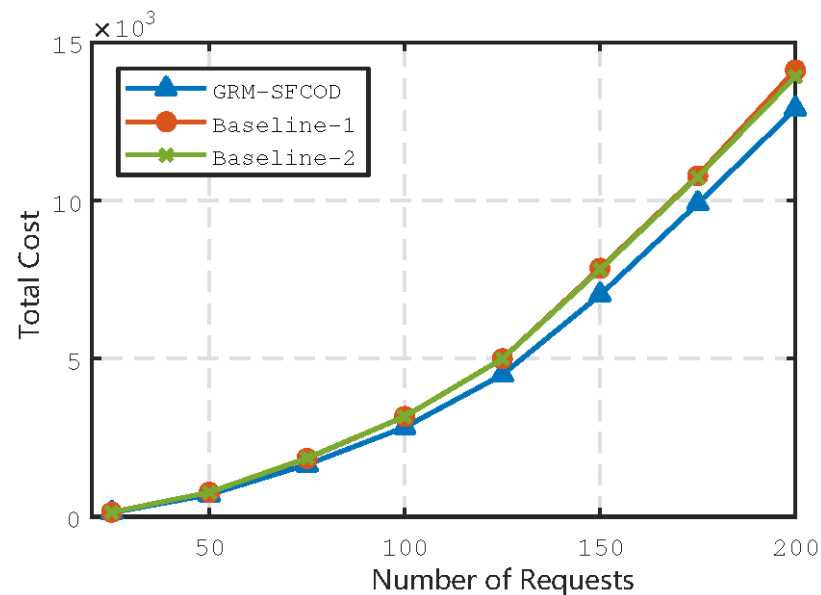


Figure 5. Deployment cost comparison of different VNF replica number determination algorithms.

This cost efficiency stems from the distinct replica allocation strategies. GRM-SFCOD leverages the marginal reliability gain metric to prioritize backups for VNFs that contribute most significantly to the system’s bottleneck reliability. This targeted approach satisfies global constraints with minimal redundancy. In contrast, Baseline-1 adopts a “uniform reliability” strategy, which often leads to over-provisioning for robust VNFs while neglecting weaker links, resulting in resource wastage. Similarly, Baseline-2 lacks a greedy optimization criterion, failing to distinguish the varying contributions of different VNF replicas to the overall reliability. Consequently, the experimental results corroborate the theoretical analysis in Section 5.3.1, demonstrating that the proposed method achieves superior resource utilization. Therefore, GRM-SFCOD is adopted as the default configuration for subsequent experiments.

6.2.2. Performance Analysis of Resource Allocation

The GRM-SFCOD algorithm employs a dynamic pricing function that adjusts according to the utilization of each resource dimension on physical nodes. This mechanism discourages the excessive occupation of scarce resources and incentivizes VNF placement on nodes with superior resource availability and lower costs. Consequently, it effectively mitigates the “resource fragmentation” issue where one dimension is exhausted while others remain idle. To evaluate this load-balancing capability, we analyze the static deployment performance with 1000 SFC requests as the node count varies from 20 to 100.

We compare GRM-SFCOD against three representative algorithms:

- NRC-VNM [53]: utilizes the Pearson correlation coefficient to map VNFs to nodes with the strongest resource matching degree, aiming to prevent unbalanced allocation.
- WRAP [54]: models node resource utilization via a weight table. Nodes are prioritized based on comprehensive correction weights derived from current utilization levels.
- Greedy: adopts a best-fit strategy, deploying replicas to the node with the maximum remaining resources. If capacity is insufficient, it iteratively selects the next most available node.

To isolate the impact of the pricing strategy on mapping efficiency, admission control is disabled in these experiments; requests are accepted as long as a feasible deployment plan exists. Thus, the pricing mechanism solely influences the node selection process.

- Average Resource Utilization

Average resource utilization is calculated as the mean of CPU, memory, and disk usage across all active nodes. This metric reflects the network's ability to accommodate requests without generating unusable resource fragments. Higher utilization implies more compact and efficient packing.

Figure 6 compares the average resource utilization of the four algorithms. As the network scales from 20 to 100 nodes, utilization remains relatively stable, indicating consistent efficiency across different topologies. Notably, GRM-SFCOD consistently outperforms the baselines. At 100 nodes, it achieves ~95% average utilization, surpassing NRC-VNM (~88%), WRAP (~93%), and Greedy (~82%).

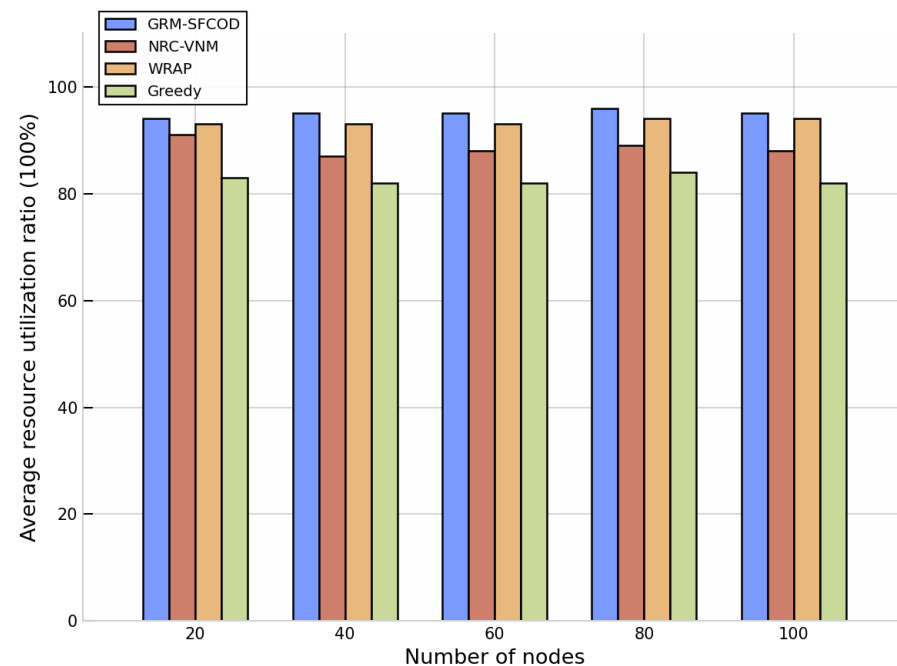


Figure 6. Comparison of average resource utilization across different algorithms and node counts.

This superiority stems from the multidimensional pricing mechanism. By dynamically adjusting prices based on real-time utilization, GRM-SFCOD guides VNFs toward nodes with balanced residual resources, thereby reducing the fragmentation caused by single-resource bottlenecks. In contrast, the Greedy algorithm ignores multidimensional correlations, leading to severe imbalance and wasted fragments. While NRC-VNM and WRAP consider resource capacities, they lack a global view of the deployment sequence's impact, resulting in suboptimal packing compared to our proposed approach.

- Service Acceptance Ratio

The Service Acceptance Ratio (SAR) is defined as the proportion of successfully deployed SFCs relative to total requests. In the absence of explicit admission control, a higher SAR directly indicates a more effective exploitation of limited physical resources to support a larger service volume.

Figure 7 illustrates the SAR trends across varying node counts. As expected, acceptance rates rise with network scale due to increased aggregate capacity. However, the performance gap widens significantly as the topology expands. At 100 nodes, GRM-SFCOD achieves the highest acceptance ratio of ~82% compared to ~78% for NRC-VNM, ~80% for WRAP, and ~72% for Greedy.

These results corroborate the findings in Figure 6. The Greedy algorithm's myopic strategy causes premature fragmentation, severely limiting its scalability. WRAP performs moderately well due to its weight-based selection but is hindered by its lack of sequence awareness. GRM-SFCOD, by maintaining balanced multi-dimensional utilization, minimizes resource wastage and secures a superior acceptance ratio, validating the effectiveness of the proposed pricing model.

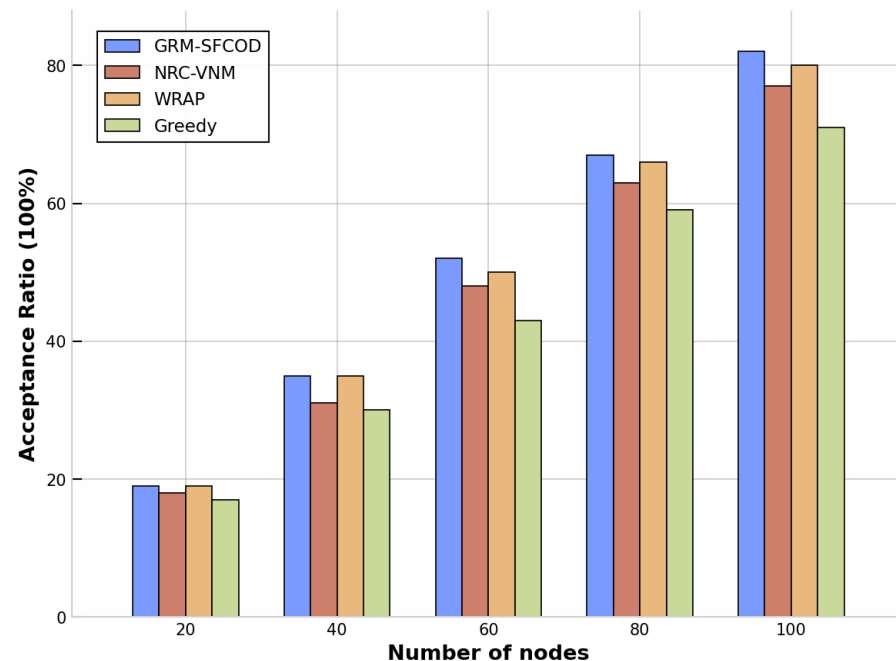


Figure 7. Comparison of SFC request acceptance ratio across different algorithms and node counts.

6.2.3. Performance Analysis of Online Deployment

As an online algorithm, GRM-SFCOD must make irrevocable admission and placement decisions immediately upon request arrival, lacking a priori knowledge of future demands. In contrast, offline algorithms can optimize globally based on full request information. To evaluate the competitiveness of GRM-SFCOD, we first compare it against the offline optimal solution (OPT) in a small-scale network scenario ($|V| = 25$). The OPT is obtained by solving problem P1 using the Gurobi solver. Given the high complexity of the original MINLP formulation, we employ the subalgorithm proposed in Section 5.3.1 to determine replicas counts, transforming problem P1 into a solvable MILP problem. To ensure tractability, the relative MIP gap for Gurobi is set to 0.1%, yielding a near-optimal benchmark.

Figure 8 illustrates the comparison of the total value of SFC deployment between GRM-SFCOD and the OPT as the number of requests increases. Both algorithms exhibit a monotonically increasing trend. When the number of requests is fewer than 60, physical resources are abundant, and both algorithms tend to deploy all arriving requests directly, resulting in negligible differences in total value. However, as the request count exceeds 60, resource contention intensifies. The offline optimal algorithm, possessing global knowledge, can evict low-value SFCs to accommodate higher-value ones. In contrast, GRM-SFCOD relies on its dynamic pricing mechanism to approximate this behavior online. Specifically, it raises deployment costs exponentially based on utilization, effectively rejecting low-value requests to preserve resources for future high-value opportunities. Consequently, when the request count reaches 300, the total value achieved by GRM-SFCOD is $\sim 81.05\%$ of the offline optimal solution. Furthermore, as indicated in Table 1, the computation time for the offline optimal solution increases drastically with the number of requests,

reaching intractable levels (e.g., ∞ at 400 requests), whereas GRM-SFCOD maintains a low computational overhead, validating its efficiency for online scenarios.

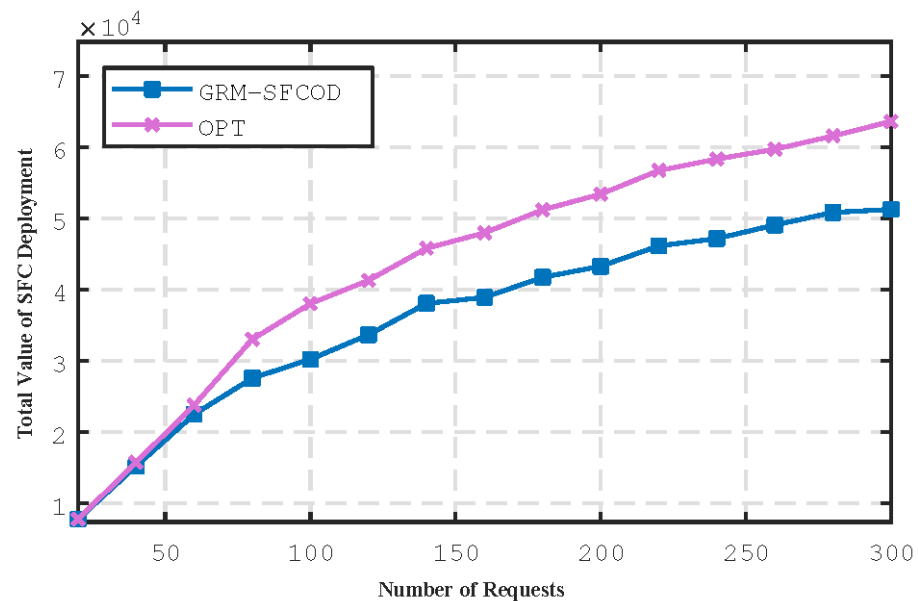


Figure 8. Comparison between GRM-SFCOD and offline optimal solution.

Table 1. Execution time (s) of Gurobi (OPT) and GRM-SFCOD algorithms under different request counts.

Number of Requests	GRM-SFCOD	Gurobi (OPT)
100	0.575	24.265
200	1.098	54.560
300	1.403	16,762.091
400	1.407	∞

The aforementioned results demonstrate that the GRM-SFCOD algorithm achieves near-optimal performance with low computational overhead in small-scale networks. To further evaluate performance in large-scale networks where exact optimization is infeasible, we compare GRM-SFCOD with two benchmarks: Best-Fit (a heuristic online algorithm prioritizing nodes based on WRAP weights [54]) and PBP (an offline probability-based heuristic [22] serving as a performance upper bound). Experiments are conducted with 50 server nodes and 800 SFC requests.

Figure 9 illustrates the variation in SFC request acceptance ratios. All algorithms show a downward trend due to resource saturation. Notably, in the initial phase (requests < 250), GRM-SFCOD exhibits a lower acceptance ratio compared to Best-Fit. This observation validates our algorithmic assumption regarding the exponential pricing mechanism: in the early stages, the pricing function imposes a stricter admission threshold to prevent the “myopic” consumption of resources by low-value requests—a common pitfall for greedy algorithms like Best-Fit. Although this results in a temporarily lower acceptance rate, it prevents premature resource exhaustion. As the load increases (requests > 250), Best-Fit’s acceptance ratio drops rapidly due to severe fragmentation, while GRM-SFCOD maintains a more gradual decline, eventually outperforming Best-Fit. The PBP algorithm consistently achieves the highest ratio due to its offline global optimization capability.

Figure 10 compares the average system resource utilization. The results corroborate the effectiveness of the VNF placement strategy via the improved Genetic Algorithm guided by marginal reliability gain. The Best-Fit algorithm fails to account for resource heterogeneity,

leading to imbalance where certain dimensions are exhausted while others remain idle; consequently, its steady-state utilization plateaus at $\sim 80\%$. In contrast, the decomposition strategy of GRM-SFCOD—specifically the placement via an improved Genetic Algorithm guided by marginal reliability gain—proves effective here. By dynamically adjusting prices based on multidimensional scarcity, GRM-SFCOD guides VNFs toward nodes with balanced resource availability. This significantly mitigates fragmentation, allowing the algorithm to achieve a utilization level close to the offline upper bound (PBP). When the request count exceeds 600, the average resource utilization of GRM-SFCOD is $\sim 17\%$ higher than that of Best-Fit. This improved utilization directly contributes to the superior long-term cumulative value shown in Figure 11.

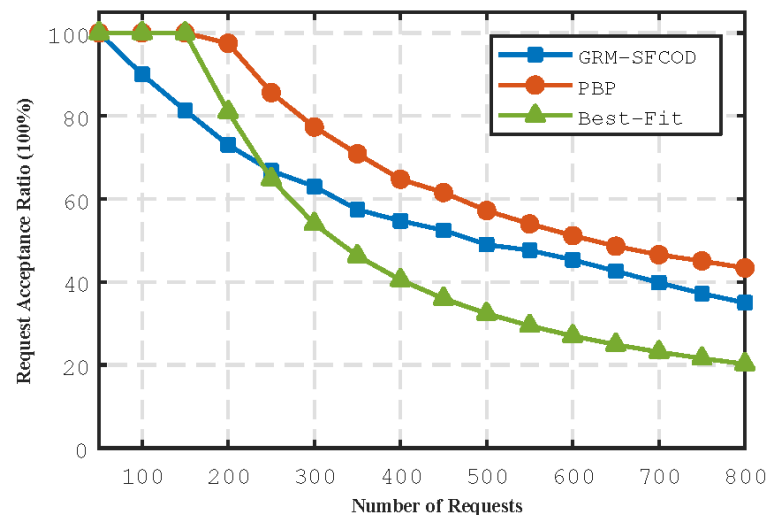


Figure 9. Comparison of SFC request acceptance ratio.

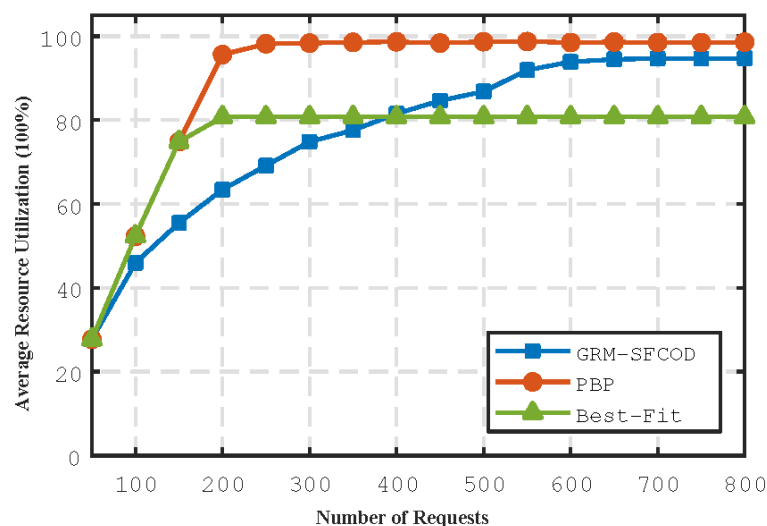


Figure 10. Comparison of average resource utilization.

Figure 11 illustrates the cumulative total value of accepted service chains. The results here further corroborate the accuracy of our value-aware assumptions and the synergy between the two decomposition components. In the initial stage (requests 0–200), despite the lower acceptance ratio shown in Figure 8, the cumulative total value of GRM-SFCOD remains comparable to Best-Fit. This confirms that the requests rejected by GRM-SFCOD in the early phase were indeed low-value services consuming disproportionate resources—precisely the scenario the exponential pricing mechanism is designed to filter out. As the system scales, the advantage of this strategy becomes pronounced: at

700 requests, the cumulative total value of GRM-SFCOD is $\sim 127\%$ higher than that of Best-Fit. Moreover, GRM-SFCOD achieves over $\sim 86\%$ of the offline PBP's total value when requests do not exceed 600, and it maintains approximately $\sim 81\%$ even at 800 requests. These results demonstrate that the combination of exponential pricing (for replica allocation) and reliability-aware decomposition via improved Genetic Algorithm (for VNF placement) enables GRM-SFCOD to approach offline optimality with significantly lower computational complexity.

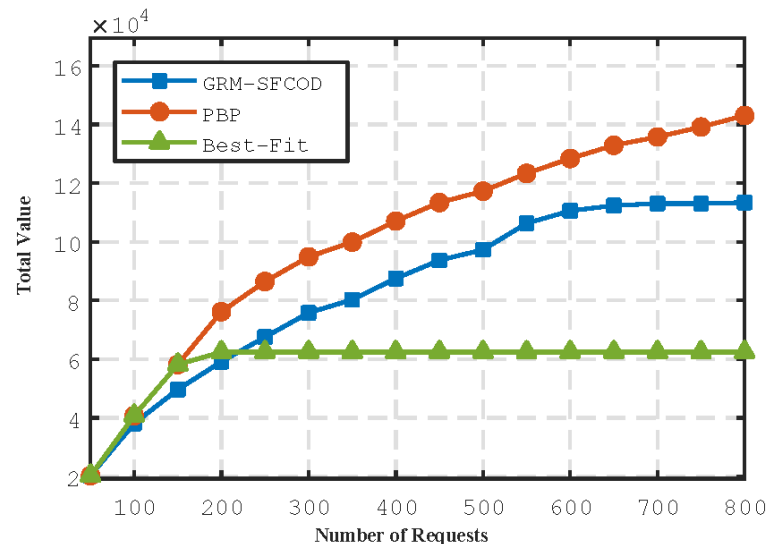


Figure 11. Comparison of cumulative total value.

Finally, we evaluate the scalability of GRM-SFCOD relative to Best-Fit by varying the node count from 20 to 200 with a fixed load of 4000 requests (averaged over 20 topologies). As shown in Figure 12, the total value achieved by GRM-SFCOD is significantly superior across all network scales, and the performance gap widens as the network grows. In a small-scale network (20 nodes), GRM-SFCOD yields $\sim 61\%$ higher total value than Best-Fit; in a large-scale network (200 nodes), this improvement reaches $\sim 97.6\%$, highlighting the algorithm's robustness and scalability.

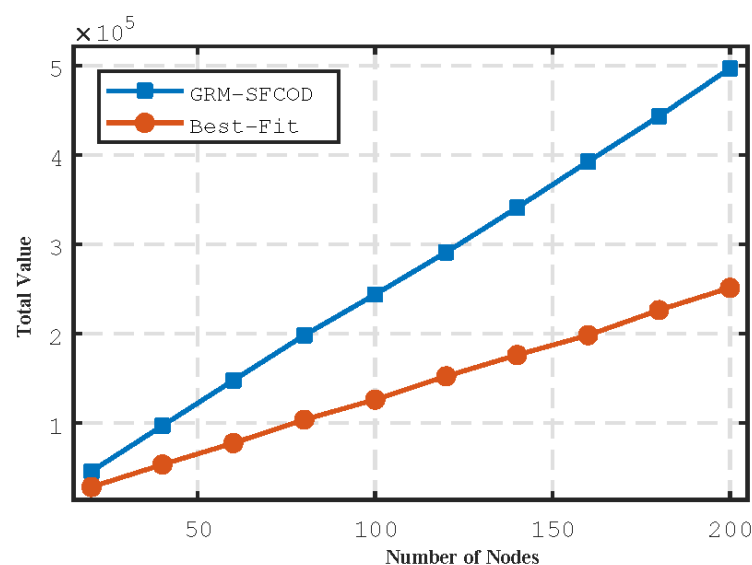


Figure 12. Comparison of algorithm cumulative total value under different network scales.

7. Conclusions

This paper has addressed the online deployment problem of reliable service function chaining in the resource-heterogeneous network. A reliability-aware service chain model supporting multi-replica parallelism and backup fault tolerance was established. The online deployment problem was formulated as an MINLP maximizing network service provider long-term revenue, and its NP-hardness was formally proved. Through linearization via McCormick envelopes, the problem was transformed into an Integer Linear Program. Leveraging Lagrangian duality, the concept of multidimensional resource shadow prices was introduced, establishing the theoretical foundation for dynamic pricing. Based on this framework, GRM-SFCOD was proposed, comprising three core components: (i) a dynamic pricing model that adjusts resource prices exponentially with utilization; (ii) an online admission control strategy based on net profit threshold comparison; and (iii) a replica allocation algorithm using marginal reliability gain and a Genetic Algorithm-based VNF placement heuristic for minimum-cost deployment. Simulation results demonstrate that GRM-SFCOD can effectively reduce resource consumption while ensuring the overall reliability of the service chain, improving the long-term revenue of the network service provider.

Several directions remain for future research. First, the competitive ratio of the online algorithm warrants rigorous theoretical analysis to provide performance guarantees under adversarial request sequences. Second, while deep learning has become a mainstream approach in related areas, its direct application to this problem is hindered by three key factors: (i) the stringent millisecond-level per-request decision latency precludes DRL's training–inference cycles; (ii) the lack of offline training data due to stochastic request arrivals; and (iii) the primal-dual framework already provides compact subproblems with efficient closed-form solutions, leaving limited room for neural networks to improve upon. Nevertheless, machine learning techniques may still complement this framework in predictive tasks such as refining pricing parameters or forecasting arrival patterns when historical data are available. Third, extending the multi-replica model to inter-data-center scenarios by incorporating bandwidth constraints and propagation delay (including cross-node synchronization overhead) into the MINLP formulation would broaden its applicability to bandwidth-limited edge and 5G/6G networks, where the current assumption of negligible intra-DC latency no longer holds. Fourth, incorporating correlated failure models to capture infrastructure-level dependencies—such as shared power supply, hypervisor, or management platforms among co-located replicas—would enhance the practical fidelity of the reliability model beyond the independent failure assumption adopted in this paper. Finally, investigating energy-aware reliability mechanisms to minimize power consumption while maintaining fault tolerance presents a promising avenue for green NFV deployment.

Author Contributions: Conceptualization, H.G. and B.G.; methodology, B.G. and J.D.; investigation, J.D.; resources, X.F. and X.W.; data curation, X.W. and C.L.; writing—original draft preparation, H.G. and W.X.; writing—review and editing, Q.G. and W.X.; visualization, H.G. and J.L.; supervision, Z.Y.; project administration, H.G. All authors have read and agreed to the published version of the manuscript.

Funding: [Science and Technology Project of China Southern Power Grid Co., Ltd.] (Grant No. 030100KC23110063 (GDKJXM20231279)).

Data Availability Statement: The original contributions presented in this paper are included in the article.

Acknowledgments: The authors would like to thank the editor and all reviewers for their valuable comments and efforts on this article.

Conflicts of Interest: Authors Haitong Gu, Bin Guo, Jun Dong, Xingxing Feng, Xiaoqiang Wu, Chaoheng Liang, Jingbo Lin and Zhuohang Yu were employed by the company Guangzhou Power Supply Bureau, Guangdong Power Grid Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

- McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 69–74. [\[CrossRef\]](#)
- Han, B.; Gopalakrishnan, V.; Ji, L.; Lee, S. Network Function Virtualization: Challenges and Opportunities for Innovations. *IEEE Commun. Mag.* **2015**, *53*, 90–97. [\[CrossRef\]](#)
- Chiosi, M.; Clarke, D.; Willis, P.; Reid, A.; Feger, J.; Bugenhagen, M.; Khan, W.; Fargano, M.; Cui, C.; Deng, H. Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges and Call for Action. In Proceedings of the SDN OpenFlow World Congress, Darmstadt, Germany, 22–24 October 2012; Volume 48, pp. 1–16.
- Hu, F.; Hao, Q.; Bao, K. A Survey on Software-Defined Network and Openflow: From Concept to Implementation. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 2181–2206. [\[CrossRef\]](#)
- Martins, J.; Ahmed, M.; Raiciu, C.; Olteanu, V.; Honda, M.; Bifulco, R.; Huici, F. ClickOS and the Art of Network Function Virtualization. In *Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*; USENIX Association: Berkeley, CA, USA, 2014; pp. 459–473.
- Xu, Z.; Zhang, Z.; Liang, W.; Xia, Q.; Rana, O.; Wu, G. QoS-Aware VNF Placement and Service Chaining for IoT Applications in Multi-Tier Mobile Edge Networks. *ACM Trans. Sens. Netw.* **2020**, *16*, 1–27. [\[CrossRef\]](#)
- Bhamare, D.; Jain, R.; Samaka, M.; Erbad, A. A Survey on Service Function Chaining. *J. Netw. Comput. Appl.* **2016**, *75*, 138–155. [\[CrossRef\]](#)
- Azadiabad, S.; Khendek, F. Dependability of Network Services in the Context of NFV: A Taxonomy and State of the Art Classification. *J. Netw. Syst. Manag.* **2024**, *32*, 37. [\[CrossRef\]](#)
- Kibalya, G.; Serrat, J.; Gorricho, J.; Serugunda, J.; Zhang, P. A Multi-Stage Graph Based Algorithm for Survivable Service Function Chain Orchestration with Backup Resource Sharing. *Comput. Commun.* **2021**, *174*, 42–60. [\[CrossRef\]](#)
- Promwongsa, N.; Ebrahimzadeh, A.; Glitho, R.; Crespi, N. Joint VNF Placement and Scheduling for Latency-Sensitive Services. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 2432–2449. [\[CrossRef\]](#)
- Farkiani, B.; Bakhshi, B.; MirHassani, S.; Wauters, T.; Volckaert, B.; De Turck, F. Prioritized Deployment of Dynamic Service Function Chains. *IEEE/ACM Trans. Netw.* **2021**, *29*, 979–993. [\[CrossRef\]](#)
- Wang, Z.; Zhang, J.; Huang, T.; Liu, Y. Service Function Chain Composition, Placement, and Assignment in Data Centers. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 1638–1650. [\[CrossRef\]](#)
- Amaldi, E.; Coniglio, S.; Koster, A.; Tieves, M. On the Computational Complexity of the Virtual Network Embedding Problem. *Electron. Notes Discret. Math.* **2016**, *52*, 213–220. [\[CrossRef\]](#)
- Li, D.; Hong, P.; Xue, K.; Pei, J. Virtual Network Function Placement and Resource Optimization in NFV and Edge Computing Enabled Networks. *Comput. Netw.* **2019**, *152*, 12–24. [\[CrossRef\]](#)
- Jin, P.; Fei, X.; Zhang, Q.; Liu, F.; Li, B. Latency-Aware VNF Chain Deployment with Efficient Resource Reuse at Network Edge. In *Proceedings of the IEEE INFOCOM*; IEEE: New York, NY, USA, 2020; pp. 267–276. [\[CrossRef\]](#)
- Varasteh, A.; Madiwalar, B.; Van Bemten, A.; Kellerer, W.; Mas-Machuca, C. Holu: Power-Aware and Delay-Constrained VNF Placement and Chaining. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 1524–1539. [\[CrossRef\]](#)
- Guo, H.; Wang, Y.; Li, Z.; Qiu, X.; An, H.; Yu, P.; Yuan, N. Cost-Aware Placement and Chaining of Service Function Chain with VNF Instance Sharing. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*; IEEE: New York, NY, USA, 2020; pp. 1–8. [\[CrossRef\]](#)
- Hawilo, H.; Jammal, M.; Shami, A. Network Function Virtualization-Aware Orchestrator for Service Function Chaining Placement in the Cloud. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 643–655. [\[CrossRef\]](#)
- Wang, M.; Cheng, B.; Li, B.; Chen, J. Service Function Chain Composition and Mapping in NFV-Enabled Networks. In *Proceedings of the IEEE World Congress on Services (SERVICES)*; IEEE: New York, NY, USA, 2019; pp. 331–334. [\[CrossRef\]](#)
- Thanh, N.; Trung Kien, N.; Hoa, N.; Huong, T.; Wamser, F.; Hossfeld, T. Energy-Aware Service Function Chain Embedding in Edge-Cloud Environments for IoT Applications. *IEEE Internet Things J.* **2021**, *8*, 13465–13486. [\[CrossRef\]](#)
- Yaghoubpour, F.; Bakhshi, B.; Seifi, F. End-to-End Delay Guaranteed Service Function Chain Deployment: A Multi-Level Mapping Approach. *Comput. Commun.* **2022**, *194*, 433–445. [\[CrossRef\]](#)
- Wu, Y.; Zheng, W.; Zhang, Y.; Li, J. Reliability-Aware VNF Placement Using a Probability-Based Approach. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 2478–2491. [\[CrossRef\]](#)

23. Li, J.; Guo, S.; Liang, W.; Chen, Q.; Xu, Z.; Xu, W.; Zomaya, A. Digital Twin-Assisted, SFC-Enabled Service Provisioning in Mobile Edge Computing. *IEEE Trans. Mob. Comput.* **2024**, *23*, 393–408. [\[CrossRef\]](#)
24. Fan, Q.; Pan, P.; Li, X.; Wang, S.; Li, J.; Wen, J. DRL-D: Revenue-Aware Online Service Function Chain Deployment via Deep Reinforcement Learning. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 4531–4545. [\[CrossRef\]](#)
25. Li, Y.; Liang, W.; Li, J. Profit Driven Service Provisioning in Edge Computing via Deep Reinforcement Learning. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 3006–3019. [\[CrossRef\]](#)
26. Ding, W.; Yu, H.; Luo, S. Enhancing the Reliability of Services in NFV with the Cost-Efficient Redundancy Scheme. In *Proceedings of the IEEE International Conference on Communications (ICC)*; IEEE: New York, NY, USA, 2017; pp. 1–6. [\[CrossRef\]](#)
27. Kita, N.; He, F.; Oki, E. Unavailability-Aware Backup Allocation Model Based on Two-Stage Shared Protection for Middleboxes. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 70–87. [\[CrossRef\]](#)
28. Dinh, N.; Kim, Y. An Efficient Availability Guaranteed Deployment Scheme for IoT Service Chains over Fog-Core Cloud Networks. *Sensors* **2018**, *18*, 3970. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Li, J.; Liang, W.; Huang, M.; Jia, X. Reliability-Aware Network Service Provisioning in Mobile Edge-Cloud Networks. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 1545–1558. [\[CrossRef\]](#)
30. Liang, W.; Ma, Y.; Xu, W.; Jia, X.; Chau, S. Reliability Augmentation of Requests with Service Function Chain Requirements in Mobile Edge-Cloud Networks. In *Proceedings of the 49th International Conference on Parallel Processing (ICPP)*; ACM: New York, NY, USA, 2020; pp. 1–11. [\[CrossRef\]](#)
31. Liang, W.; Ma, Y.; Xu, W.; Xu, Z.; Jia, X.; Zhou, W. Request Reliability Augmentation With Service Function Chain Requirements in Mobile Edge Computing. *IEEE Trans. Mob. Comput.* **2022**, *21*, 4541–4554. [\[CrossRef\]](#)
32. Chen, Y.; Wu, J. Latency-Efficient VNF Deployment and Path Routing for Reliable Service Chain. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 651–661. [\[CrossRef\]](#)
33. Alleg, A.; Ahmed, T.; Mosbah, M.; Boutaba, R. Joint Diversity and Redundancy for Resilient Service Chain Provisioning. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1490–1504. [\[CrossRef\]](#)
34. He, F.; Oki, E. Backup Allocation Model with Probabilistic Protection for Virtual Networks Against Multiple Facility Node Failures. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 2943–2959. [\[CrossRef\]](#)
35. Mandal, P. Comparison of Placement Variants of Virtual Network Functions from Availability and Reliability Perspective. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 860–874. [\[CrossRef\]](#)
36. Zhu, M.; He, F.; Oki, E. Resource Allocation Model Against Multiple Failures with Workload-Dependent Failure Probability. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 1098–1116. [\[CrossRef\]](#)
37. Ishigaki, G.; Devic, S.; Gour, R.; Jue, J. DeepPR: Progressive Recovery for Interdependent VNFs with Deep Reinforcement Learning. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 2386–2399. [\[CrossRef\]](#)
38. Natalino, C.; Coelho, F.; Lacerda, G.; Braga, A.; Wosinska, L.; Monti, P. A Proactive Restoration Strategy for Optical Cloud Networks Based on Failure Predictions. In *Proceedings of the 20th International Conference on Transparent Optical Networks (ICTON)*; IEEE: New York, NY, USA, 2018; pp. 1–5. [\[CrossRef\]](#)
39. Zhu, M.; He, F.; Oki, E. Optimal Primary and Backup Resource Allocation with Workload-Dependent Failure Probability. In *Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC)*; IEEE: New York, NY, USA, 2020; pp. 606–611. [\[CrossRef\]](#)
40. Alomari, Z.; Zhani, M.; Aloqaily, M.; Bouachir, O. On Ensuring Full Yet Cost-Efficient Survivability of Service Function Chains in NFV Environments. *J. Netw. Syst. Manag.* **2023**, *31*, 45. [\[CrossRef\]](#)
41. Ghaznavi, M.; Shahriar, N.; Kamali, S.; Ahmed, R.; Boutaba, R. Distributed Service Function Chaining. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 2479–2489. [\[CrossRef\]](#)
42. Hou, G.; Liu, Z.; Guo, J.; Miao, Y.; Ma, J.; Gao, D.; Liao, G. Cross-Cloud Associated Multi-Replica Auditing for Lightweight Devices in the IoT. *IEEE Internet Things J.* **2025**, *12*, 23495–23509. [\[CrossRef\]](#)
43. Tian, H.; Wang, M.; Quan, H.; Chang, C.; Vasilakos, A. TEEMRDA: Leveraging Trusted Execution Environments for Multi-Replica Data Auditing in Cloud Storage. *Comput. Secur.* **2025**, *150*, 104250. [\[CrossRef\]](#)
44. Chu, P.; Beasley, J. A Genetic Algorithm for the Multidimensional Knapsack Problem. *J. Heuristics* **1998**, *4*, 63–86. [\[CrossRef\]](#)
45. Garey, M.R.; Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*; W.H. Freeman: San Francisco, CA, USA, 1979.
46. Zhang, Z.; Li, Z.; Wu, C. Optimal Posted Prices for Online Cloud Resource Allocation. *Proc. ACM Meas. Anal. Comput. Syst.* **2017**, *1*, 1–26. [\[CrossRef\]](#)
47. Yang, L.; Zeynali, A.; Hajiesmaili, M.; Sitaraman, R.; Towsley, D. Competitive Algorithms for Online Multidimensional Knapsack Problems. *Proc. ACM Meas. Anal. Comput. Syst.* **2021**, *5*, 1–30. [\[CrossRef\]](#)
48. Hill, T.; Houdre, C. (Eds.) *Advances in Stochastic Inequalities: AMS Special Session on Stochastic Inequalities and Their Applications, 17–19 October 1997, Georgia Institute of Technology*; American Mathematical Society: Providence, RI, USA, 1999.

49. Back, T. Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. In *Proceedings of the IEEE Conference on Evolutionary Computation*; IEEE: New York, NY, USA, 1994; pp. 57–62. [\[CrossRef\]](#)
50. Back, T.; Schwefel, H. An Overview of Evolutionary Algorithms for Parameter Optimization. *Evol. Comput.* **1993**, *1*, 1–23. [\[CrossRef\]](#)
51. Zegura, E.; Calvert, K.; Bhattacharjee, S. How to Model an internetwork. *IEEE/ACM Trans. Netw.* **1997**, *5*, 770–783. [\[CrossRef\]](#)
52. Zhang, J.; Wang, Z.; Peng, C.; Zhang, L.; Huang, T.; Liu, Y. RABA: Resource-Aware Backup Allocation For A Chain of Virtual Network Functions. In *Proceedings of the IEEE INFOCOM*; IEEE: New York, NY, USA, 2019; pp. 1918–1926. [\[CrossRef\]](#)
53. Sun, S.; Peng, L. Network Resource Correlation-Aware Virtual Network Mapping Algorithm. *Appl. Res. Comput.* **2023**, *40*, 885–889.
54. Kim, S.; Kim, H. A VNF Placement Method Based on VNF Characteristics. In *Proceedings of the International Conference on Information Networking (ICOIN)*; IEEE: New York, NY, USA, 2021; pp. 864–869. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.