sensors

MDPI

Article

Resource Mapping Allocation Scheme in 6G Satellite Twin Network

Zhongliang Deng * and Xiaoyi Yu

School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
* Correspondence: dengzhl@bupt.edu.cn

Abstract: The sixth generation (6G) satellite twin network is an important solution to achieve seamless global coverage of 6G. The deterministic geometric topology and the randomness of the communication behaviors of 6G networks limit the realism and transparency of cross-platform and cross-object communication, twin, and computing co-simulation networks. Meanwhile, the parallel-based serverless architecture has a high redundancy of computational resource allocation. Therefore, for the first time, we present a new hypergraph hierarchical nested kriging model, which provides theoretical analysis and modeling of integrated relationships for communication, twin, and computing. We explore the hierarchical unified characterization method which joins heterogeneous topologies. A basis function matrix for local flexible connectivity of the global network is designed for the connection of huge heterogeneous systems to decouple the resource mapping among heterogeneous networks. To improve the efficiency of resource allocation in communication, twin, and computing integrated network, a multi-constraint multi-objective genetic algorithm (MMGA) based on the common requirements of operations, storage, interaction, and multi-layer optimal solution conflict is proposed for the first time. The effectiveness of the algorithm and architecture is verified through simulation and testing.

Keywords: 6G; satellite communication; twin network; integrated network framework; multi-objective optimization

check for updates

Citation: Deng, Z.; Yu, X. Resource Mapping Allocation Scheme in 6G Satellite Twin Network. *Sensors* **2022**, *22*, 5816. https://doi.org/10.3390/s22155816

Academic Editor: Jose F. Monserrat

Received: 27 June 2022
Accepted: 29 July 2022
Published: 4 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 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) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The giant satellite Internet constellation has emerged as one of the key visions for 6G communication research. It is significantly different from other 6G communication visions and existing satellite networks in that the giant satellite Internet must be capable of reliable and efficient access on demand in all space and time because of the nonuniform distribution of global users over time and space. It will face unprecedented challenges worldwide, such as on-demand reconfiguration of network node functions under highly dynamic variation in the topology of the giant satellite network, the end-to-end transmission under high-speed links, etc. To address these challenges, research institutions [1–4] have proposed techniques such as satellite software-defined network (SDN) and large-scale end-to-end slicing, which provide good reference ideas in theoretical exploration. To validate the related techniques, researchers have formed a virtual network based on computational resources by abstracting key functional features of the network and establishing mapping relationships among the features to analyze the network performance under the given network architecture. In fact, they have constructed an integrated model of communication, twin, and computing in a preliminary dimension [5–10]. With the research and exploration of the metaverse, it is the key to realizing the intelligent management of 6G networks and studying the innovation of 6G communication protocols, in that a twin network that can simulate and reflect the network operation in high fidelity and detail is constructed.

As shown in Figure 1, building a 6G satellite twin network faces several difficulties:

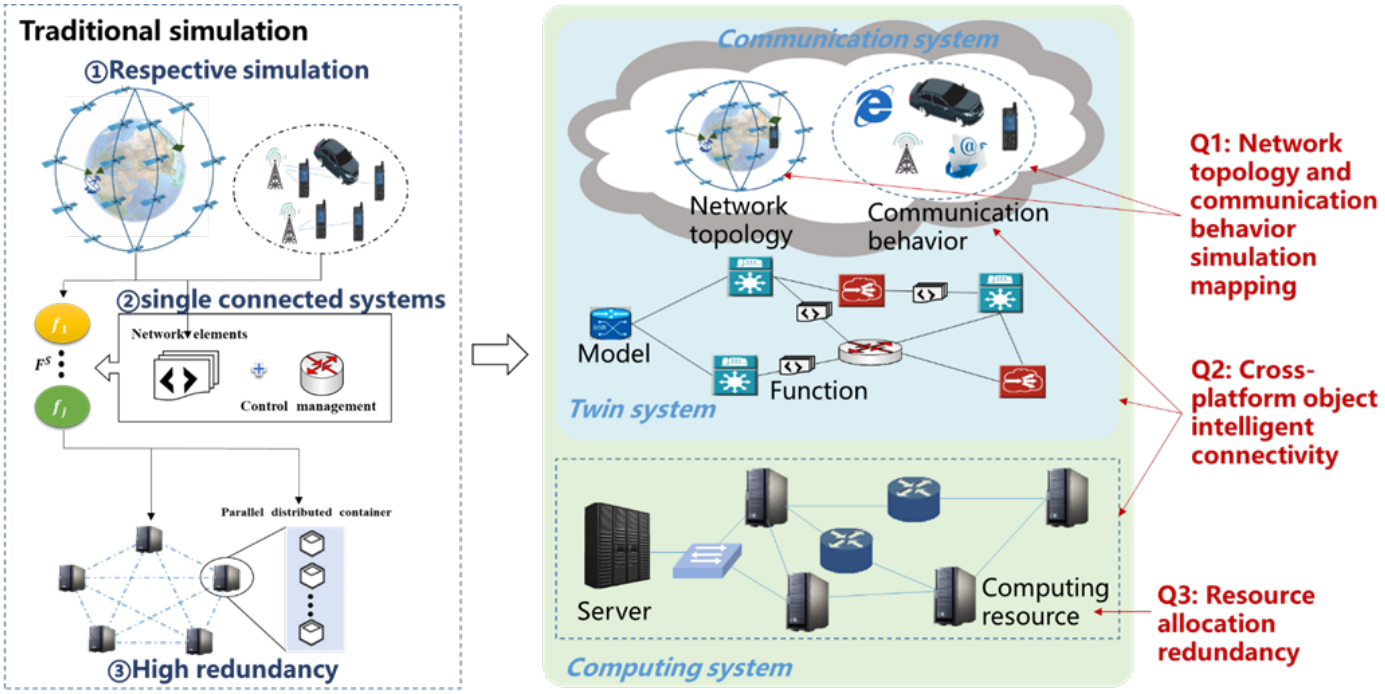


Figure 1. The challenges of mapping and resource allocation in 6G satellite twin network.

1. **Deterministic network geometric topology and fuzzy random communication behavior are difficult to map synergistically**: 6G satellite network simulation needs to realistically portray the network topology and node communication behavior under the rapid change of space and time of massive dynamic network nodes. The network topology is deterministic and discrete. The communication behavior is random and continuous. It is a great challenge for collaborative high-fidelity simulation of network topology and communication behaviors.
2. **The communication network simulation based on computational functions has the problem that cross-platform and cross-object connectivity is isolated**: The communication system, the twin system, and the computing system are completely different research objects in terms of function and logic. The intelligent management of the 6G network needs to realize the penetration of communication, twin, and computing. However, it is difficult to transparently sense the network state.
3. **The redundancy of resource allocation of the communication–twin–computing integration network is high**: The communication–twin–computing integration network needs to use computing, interaction, storage, and other functions. The implementations of communication, twin, and computing have a large redundancy. There is a major challenge to extract the common functional requirements of communication, twin, and computing in computing, interaction, and storage.

To realize the joint simulation of network geometric topology and communication behavior, the static transmission of network topology position [11], the static transmission of network geometric topology characterization [12–14], and synchronous transmission of network topology compression characterization [15] can be generally studied in existing studies. The static transmission of network topology position is to calculate the relative position relationship of the ground nodes and the satellite constellation according to the periodic operational position coordinate information of the satellite constellation. The relative position relationship is used as a premise for the mathematical simulation of random communication behaviors. However, this method cannot transmit the change of satellite position in real time. It is difficult to characterize the real-time network geometric topology range of satellite and ground nodes. The static transmission of network geometric topology characterization is that the connection between the satellite constellation and the

ground nodes is used as the input for the random simulation of communication behaviors. Although this method can simulate the discrete connectivity of the network geometric topology of the satellite constellation and the ground nodes, it also does not realize real-time interaction between the network geometric topology and communication behaviors. The synchronous transmission of network topology compression characterization is that the network topology of the satellite constellation and ground nodes is split into a series of event slices over the system cycle, in which each slice describes a fixed network topology over a period of time. This method can synchronously transfer the network topology by dividing the time slices, but the too-small slice division imposes higher requirements on the computer performance and the fidelity of real-time interaction between network geometric topology and communication behaviors.

For the connectivity problem of the communication–twin–computing integration model, object standardization as middleware [16,17], object unification as multi-agent [18,19], service-oriented integration of objects [20,21], microservice-independent object [22,23], and object normalization as function [24] are generally studied. The object of normalization as middleware is to shield the differences between different hardware and software environments and provide a standard, unified transport interface to accomplish transparent access between the communication system, the twin system, and the computing system. However, this approach requires specialized transmission protocols to re-encapsulate the data of systems. The complexity and massive data in the 6G satellite twin network present challenges. Object unification as multi-agent is to view heterogeneous system data as different agents and to unify the data processing commands to act accordingly during data processing. This approach has good scalability, but a large number of agents indicate that the ground nodes can cause a slower processing speed and increase system complexity. The service-oriented integration of objects is to encapsulate the tasks into different services, providing multiple different ways to access the services for heterogeneous systems and reusing the services. However, the service-oriented architecture requires the design of a complex service framework, and it is difficult to be transparently aware of the network state. Microservices-independent object is based on service-oriented with finer granularity and lightweight framework for transmission. However, for the 6G satellite twin network with massive data, too-fine service division leads to too many services and requires efficient computer performance for support. Object normalization as a function normalizes all functions in a heterogeneous system to functions. Functions are scheduled to connect the communication system, the twin system, and the computing system. However, feature extraction and normalization of cross-platform objects are huge challenges.

Facing the problem of high redundancy of resource allocation among heterogeneous systems, to meet the common demands of heterogeneous systems in computing, interaction, and storage, existing studies generally use efficient resource allocation methods such as demand resource adaptive allocation [25–28], multi-objective dynamic on-demand allocation [29–32], and artificial intelligence allocation [33,34]. The demand resource adaptive allocation method schedules computational tasks, segments tasks, and manages resources to place tasks by considering parameter size, historical configuration, and running time. Although this method can reduce the transmission time, it is difficult to distinguish the resource for different functions such as computing, interaction, and storage. The multi-objective dynamic on-demand allocation approach can provide a shared resource pool for multiple services simultaneously. Although this method can balance multiple optimization objectives, the multi-objective optimization method only finds the approximate optimal solution. This equilibrium comes at the cost of increased energy consumption. The artificial intelligence allocation method can select the bias of the resource allocation strategy by setting factors to complete the optimization search for the multi-objective problem. However, the multi-level, jointly nested, and diverse computing resources of the communication–twin–computing integration network need to achieve globally optimal cross-system scheduling. Existing artificial intelligence algorithms can only learn and evolve in existing systems. It is difficult to realize the optimization of other systems through its learning.

In order to simulate a 6G satellite twin network, the main contributions of this paper are as follows.

1. For the first time, the integrated relationships among the communication system, the twin system, and the computing system are theoretically analyzed and modeled. For the factor association analysis problem across systems, across dimensions, and across topologies, a hypergraph hierarchical nested kriging model that enables heterogeneous topological connection is introduced for the first time. A hierarchical unified feature description method is realized.
2. In the communication–twin–computing integration network model, the basis function matrix of the local flexible connection of the global network is established. The connection optimization of huge heterogeneous systems is realized, which provides a supportive way to realize the cross-dimensional collaboration of heterogeneous systems.
3. For the first time, a theoretical analysis method that can achieve joint objective optimization of complex heterogeneous systems is proposed. The common requirements of computing, storage, and interaction are also considered to achieve a multi-objective coordinate between function utilization, load balancing, and cost for heterogeneous systems. A cross-platform network simulation architecture with function isolation is proposed for the first time and the effectiveness of the algorithm is verified through experiments.

The rest of this paper is organized as follows. The related work is described in Section 2. The hypergraph hierarchical nested kriging model and the flexible coefficient autoregressive based on the basis function matrix model are given in Section 3. Section 4 analyzes a multi-constraint multi-objective genetic resource allocation optimization scheme. A large-scale network simulation architecture according to the functions is presented in Section 5. The simulation results are illustrated in Section 6. Finally, Section 7 concludes the paper.

2. Related Work

With the development of 6G, there are a large number of complex and diverse application scenarios for future 6G mobile communication systems to provide users with an extreme performance experience. For the complex 6G satellite twin network, accurate extrapolation of network performance requires system-level simulation, as well as carefully designed simulation scenarios to better understand the impact of complex relationships between system configuration parameters and corresponding network operations [35]. A semi-physical simulation system is constructed in [36], which uses a server to simulate a satellite and relies on cables and servers to model the channel of the satellite network. The coordinate model to reduce the communication latency of low Earth orbit (LEO) satellites is proposed. However, a large number of servers are required to simulate large-scale constellations under this architecture, which is costly to build. A fifth generation (5G) system-level simulation platform that meets the requirements of IMT-2020 is produced in [1], and its simulation results are verified by the 3rd Generation Partnership Project (3GPP) calibration and published. The modeling of the MAC layer in satellite communications is described in [37,38]. An event-driven satellite performance simulation framework is introduced to simulate the dynamic behavior of the network in the Brazilian region. The user's operation scenarios are customized in combination with collision detection and different coverage scenarios, but this framework is difficult to scale. The challenges of building a 5G simulation platform based on 4G evaluation methods are described in [39], where a cloud-based framework for a two-level system-level simulation platform framework is proposed and novel techniques in 5G are validated in this platform. A 5G terrestrial system-level simulation platform is introduced in [2], which tests the performance of large-scale multiple-in multiple-out (MIMO) in 5G by varying the number of antenna ports and the number of spatial multiplexing layers in different experimental environments, focusing on the effect of vertical beamforming on full-dimensional MIMO interference, demonstrating that the modular and flexible platform structure is easy to reconfigure and simulate. However, the above simulation platforms are all separate satellite systems or ground systems. The 6G

network has flexible and variable protocol stack functions and network architecture, which requires a scalable, multi-compatible, and highly flexible system-level platform for network simulation. In addition, to guarantee transmission stability for the 6G satellite network, low-density parity-check (LDPC) codes and polar codes are selected as the error-correction codes in 5G mobile communications and a lot of standards [40–42].

For large-scale simulation systems, some ideas are proposed in current research. A fast, scalable, and parallel simulator to accelerate the simulation of multicore co-processors on a graphics processing unit (GPU) platform is proposed in [43], but this platform depends on hardware conditions. A docker container-based software-as-a-service system is described in [44]. The study demonstrates the feasibility of emulating complete packet-level transmission and protocols on a serverless platform with built-in docker containers and proposes a generic queueing network approach to model the deployment behavior of application-layer software-as-a-service. A serverless cloud computing framework in which applications are modeled as functions of different granularity is proposed in [45]. These functions interact with each other to implement complex task processes. Serverless platforms are mainly used for stateless, data-parallel tasks, and the study verifies the advantages of serverless platforms in solving large-scale optimization problems [46].

For resource allocation of large-scale simulation systems, several studies have been proposed. The tenant demand model and tenant provider model are used to represent the tenant demand, provider demand, and their different attributes and behaviors. A resource deficiency model is designed in [47], and a resource allocation algorithm for resource rejection rate, average resource waiting period, and resource utilization is proposed. The relationship between the resources consumed by task requests and the resources occupied by the system in a discrete computing network is analogized to the relationship between healthy cells and infected cells in epidemiology, and a task-resource model in a discrete computing network based on the classical virus dynamics model is built [48]. I. Al-Azzoni et al. used genetic algorithms and ant colony algorithms to solve the problem of mapping a set of software components to the available computational units in a heterogeneous computing system, but the authors only considered the single-objective problem of component allocation [49]. A model-driven approach was proposed to specify the allocation problem and automatically compute feasible allocations in [50]. The author formulated the allocation problem as a 0–1 integer linear programming; however, the approach can only find feasible allocations (not optimal allocation) and cannot be extended to large-scale scenarios. A platform for implementing architecture optimization algorithms was provided in [51–53]. The component deployment problem was modeled as a dual objective optimization problem, but its optimization process stalled after a certain number of iterations. A dynamic allocation algorithm for docker container resources that adapts to application requirements considering node resources, node network consumption, and the energy consumption was proposed in [54]. The resource allocation algorithms when different services are available in docker containers are described in [55–58]. A genetic algorithm-based resource allocation and task scheduling for minimum completion time and maximum customer satisfaction was proposed in [59] in multi-cloud computing architecture. A hybrid genetic algorithm based on the case base for resource allocation was proposed in [60]. This algorithm finds Pareto solutions in a two-stage algorithm structure by computing crossover operators.

Virtual network mapping is the mapping of a virtual network into a physical network. The deployment of network functions in the virtual network is performed in physical hardware devices through software, which is essentially the allocation of physical resources. The solution to the virtual network mapping problem is mainly to map virtual nodes and virtual links by considering the availability of hardware resources such as computing resources and memory resources of physical nodes in the physical network and the bandwidth resources of physical links, which is the optimal allocation of physical networks for virtual network deployment.

Early virtual network mapping algorithms perform one-to-one mapping for simple networks in a small area, that is, a virtual node in a virtual network request can be mapped to

only one physical node. In solving such irreducible mapping problems, the problem is usually modeled as a linear programming problem or a mixed integer programming problem for solution [61,62,62,63]. Although this precise algorithm can traverse each virtual node request, it is only suitable for small virtual network mapping scenarios, and with the expansion of the network scale, the solution time and complexity of the exact algorithm grow exponentially. In addition, some of the early virtual network mapping algorithms [64–68] only consider part of the network requirements, consider the requirements of virtual network nodes in a simple normalized way, or only consider computational resources without considering the impact of storage resources on network mapping. The fidelity and practicality of this type of algorithm are lacking.

With the increase in network scale, the increase in network energy consumption, and the growth of demand for network troubleshooting, virtual network mapping algorithms propose a variety of different solutions for different mapping scenario requirements. For large-scale virtual networks, distributed virtual network mapping algorithms are proposed [69–72]. The large-scale virtual network is split into multiple small-scale virtual networks, which are deployed as multiple small-scale computational tasks to multiple distributed physical nodes for processing. Finally, the processing information in the distributed physical nodes is collected and aggregated for comprehensive processing of information. An energy-aware virtual network mapping algorithm based on energy consumption was proposed for the increase in network energy consumption [73–78]. The number of mapped physical nodes and physical links is taken as the goal of optimization, and the energy consumption of network mapping is controlled by solving for the minimum number of physical nodes and physical links. The survivability virtual network mapping algorithm is proposed for network fault handling requirements [79–82]. Survivable virtual network mapping algorithms refer to the ability of the network to still provide communication services through the reserved backup resources in case of problems such as natural disasters, network attacks, and equipment failures. The virtual networks and physical networks need to be considered in survivability virtual network mapping algorithms. The main solution considered in survivability virtual network mapping algorithms nowadays is to use the redundancy idea to provide backup resources for the network to resist network failures.

Network mapping for network simulation is studied [83], but communication systems, twin systems, and computing systems are not considered together in a comprehensive manner in the study of network mapping for network simulation. In the large-scale multi-dimensional multi-layer network mapping for network simulation studied in this paper, the mapping problem of the multi-layer heterogeneous network from the communication network to the twin network to the computing network needs to be solved. Among them, the communication network is a large-scale satellite-ground network that consists of nearly 10,000 communication nodes such as ground terminals, high and low orbit satellites, and ground stations distributed around the world. The twin network needs to complete the simulation of nearly 10,000 communication nodes, communication behaviors, and communication processes. The computing network hosts the twin network and provides physical resources such as CPU, memory, bandwidth, and throughput. In the network mapping problem, the traditional virtual network link connection is fixed, but in this paper, the connection between the nodes in the communication network is connected according to the communication behavior and satellite trajectory. The direct connection of different communication nodes changes over time. The network mapping strategy in this paper needs to consider the timeliness of the link connection while satisfying the link resource mapping. In this paper, the communication nodes in the communication network have the attributes of latitude and longitude. The constraint relationship between the geographical locations of the communication nodes needs to be considered when mapping. In view of the different requirements for multi-dimensional multi-layer mapping from the communication network to the twin network to the computing network, the mapping strategies adopted by existing virtualized simulation systems are not applicable to the network mapping in this paper.

In the simulation system studied in this paper, the mapping problem from the communication network to the twin network to the computing network can be summarized as the mapping problem from communication node to twin node to physical node, and communication link to twin link to physical link. After the mapping is completed, the following effects can be achieved: the communication node function can be modeled and reproduced by the twin node, and the physical resources required by the twin node can be carried by the physical node. The communication link function can be modeled and reproduced by the twin link, and the physical resources required and information transmitted by the twin link can be transmitted by the physical link. The amount of resources consumed by the functions carried and transmitted by each physical node and the physical link is basically the same, and the cost of power and energy consumed is minimal.

3. System Model of 6G Satellite Twin Network

3.1. A Hierarchical Unified Feature Description Method

As shown in Figure 2, the communication–twin–computing integration network is modeled as a space, which consists of three layers: one layer is the communication domain, one layer is the twin domain, and one layer is the computing domain. The communication behaviors in the communication domain are carried by the hardware resources to realize the performance indexes. The communication behaviors and resource load of the communication domain are regarded as the tasks of the twin domain to realize the performance indexes. The twin domain is regarded as the task of the computing domain, which is carried by the parallel distributed docker containers. It is difficult to uniformly model heterogeneous systems that are both independently composed within the system and interconnected, interacting, and nested with each other. We hope that a unified network model can be constructed by combining system characteristics and resource characteristics.

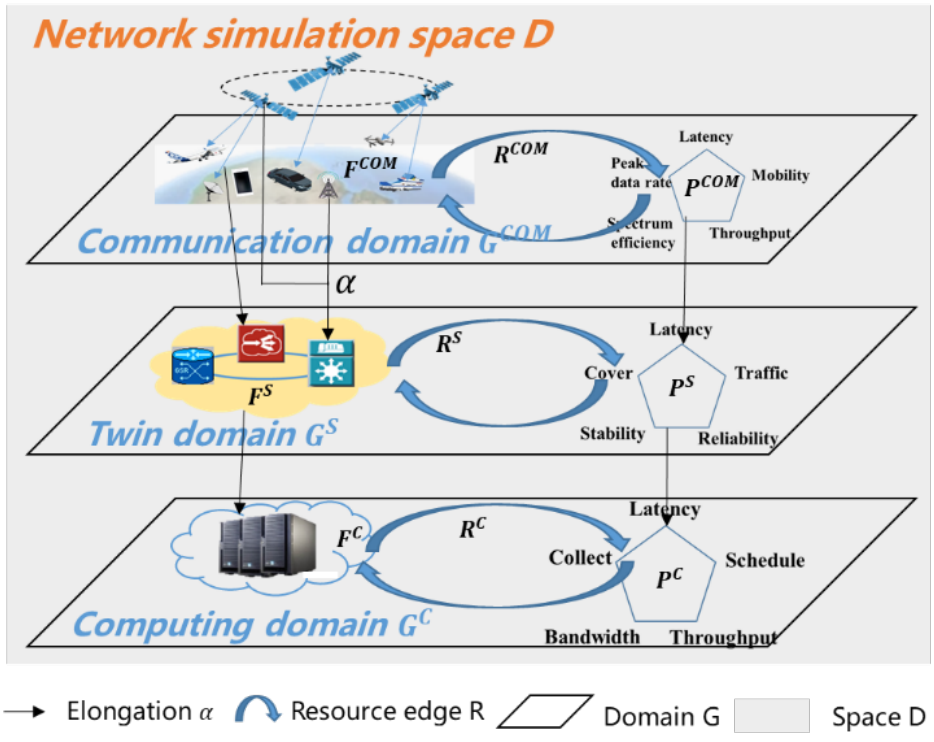


Figure 2. System model.

In this paper, as the performance indexes of the communication domain change over time, the twin domain needs to select different functions for simulation implementation. The computation resources required by the communication domain can change accord-

ingly. Therefore, we propose a hypergraph hierarchical nested kriging model based on the characteristics of full-modal and heterogeneous interconnections.

According to complex networks and graph theory, the three-dimensional network can model the content of interest as nodes and edges. The domain is a heuristic abstraction of compositions. The communication domain consists of nodes such as terminals and base stations of real communication networks, hardware resources for building nodes, communication behaviors, and communication performance indexes. The twin domain refers to the simulation of the real communication network functions, resources, environment, behavior, and simulation performance indexes. The computing domain refers to the service resources that constitute the parallel architecture, the computational tasks, and computational performance indexes. In the constructed domain, functional nodes and performance nodes have independent behaviors among themselves, while constituting synergistic behaviors through resource edges. The edges can not only represent connections established based on communication but also can be used to describe diverse associative relationships.

Definition 1. *A hypergraph hierarchical nested kriging space $D = \{G, P, Bound, \alpha\}$: the space D consists of the domain G, the performance index $P = \{p_1, \cdots, p_K\}$, the upper and lower limits of the performance index $Bound = \begin{pmatrix} p_{1,max}, \cdots, p_{K,max} \\ p_{1,min}, \cdots, p_{K,min} \end{pmatrix}$, and the extension factor α of the domain G. The range of the space D can be expressed as $\begin{pmatrix} p_{1,min} \le p_1 \le p_{1,max} \\ \vdots \\ p_{K,min} \le p_K \le p_{K,max} \end{pmatrix}$. The extension factor of the domain G can be expressed as*

$$\alpha(G) = [\alpha_1(G), \cdots, \alpha_L(G)]^T = 0.5T[|p_d v_1(R)|, \cdots, |p_d v_L(R)|], \tag{1}$$

$$p_d = p_{max} - p_{min}, \tag{2}$$

$$p_{max} = max\{p_{1,max}, \cdots, p_{K,max}\}, \tag{3}$$

$$p_{min} = max\{p_{1,min}, \cdots, p_{K,min}\}, \tag{4}$$

where T is the time interval and $v_l(R)$ is denoted as the orthogonal vector of the vector R in the domain (see Appendix B). p_d is the maximum difference of the performance index. p_{max} and p_{min} are the maximum and minimum of performance index, respectively.

Definition 2. *The domain $G = \{F, R, P\}$ is defined as a set of hypergraphs. The domain elements $F = \{f_1, \cdots, f_J\}$ represents the combinations of the nodes. $R = \{r_1, \cdots, r_P\}$ represents the combinations of the edges. $P = \{p_1, \cdots, p_Z\}$ represents the performance indexes that divide the domain.*

Theorem 1. *According to Definition 1 and Definition 2, consider a simulation space $D = \{G, P, Bound, \alpha\}$. The domain G consists of performance indexes $P^G = \left\{p_1^G, \cdots p_J^G\right\}$. The lower bound of each performance index is $l^G = \left[l_1^G, \cdots l_J^G\right]^T$ and the upper bound is $u^G = \left[u_1^G, \cdots u_J^G\right]^T$. Considering a set of node variables $d = [d_1, \cdots d_N]^T$ in the space D, the node d_n exists in the performance index $p_k^{d_n}$:*

(1) $p_k^{d_n} \subseteq P$ *and* $p_k^{d_n} \subseteq P^G$;

(2) $p_k^{d_n} \in [p_{k,min}, p_{k,max}]$;

(3) *There are* $p^G_{jmin} = min(\hat{\mu} + Ccvs^{-1^T} Ccvs(l^G - 1\hat{\mu}))$, $p^G_{jmax} = max(\hat{\mu} + Ccvs^{-1^T} Ccvs(u^G - 1\hat{\mu}))$ *(see Appendix A), and* $p^{d_n}_k \in \left[p^G_{jmin}, p^G_{jmax}\right]$. *Then,* $d_n \in G = \begin{pmatrix} p^G_{1min} \le p_1 \le p^G_{1max} \\ \vdots \\ p^G_{Jmin} \le p_J \le p^G_{Jmax} \end{pmatrix}$.

According to Definition 1, we build a simulation space $\mathrm{D} = \{\mathrm{G}, \mathrm{P}, \mathrm{Bound}, \alpha\}$. $\mathrm{G} = \left\{\mathrm{G^{COM}}, \mathrm{G^S}, \mathrm{G^C}\right\}$ denotes the communication system, the twin system, and the computing system. $\mathrm{P^G} = \left\{\mathrm{p^G_1}, \cdots \mathrm{p^G_J}\right\}$ denotes the performance index in this space: delay, rate, throughput, delay jitter, bandwidth, packet loss rate, etc. The upper and lower bounds of the performance indexes are $\mathrm{Bound^G} = \begin{pmatrix} \mathrm{p^G_{1min}}, \cdots, \mathrm{p^G_{1max}} \\ \mathrm{p^G_{Jmin}}, \cdots, \mathrm{p^G_{Jmax}} \end{pmatrix}$, and the range of space D can be expressed as $\begin{pmatrix} \mathrm{p^G_{1min}} \le \mathrm{p_1} \le \mathrm{p^G_{1max}} \\ \vdots \\ \mathrm{p^G_{Jmin}} \le \mathrm{p_J} \le \mathrm{p^G_{Jmax}} \end{pmatrix}$. There is a set of nodes $d = \{d_1, \cdots d_N\}$ in the simulation space.

According to Definition 2, the communication system can be represented as $G^{COM} = \{F^{COM}, R^{COM}, P^{COM}\}$. $F^{COM} = \left\{f^{COM}_1, \cdots, f^{COM}_J\right\}$ denotes the set of functions that make up the communication system. $R^{COM} = \{r^{COM}_1, \cdots, r^{COM}_W\}$ denotes the resources required to support the operation of the communication system. $P^{COM} = \{p^{COM}_1, \cdots, p^{COM}_Z\}$ denotes the performance indexes of the communication system. According to Theorem 1, when the performance index of the node $p^{COM}_{d_n}$ meets the requirements of the communication system, the node d_n can be divided into the communication domain.

The twin system can be represented as $G^S = \{F^S, R^S, P^S\}$. $F^S = \left\{f^S_1, \cdots, f^S_J\right\}$ denotes the set of functions that make up the twin system. $R^S = \{r^S_1, \cdots, r^S_W\}$ denotes the resources required to run each set of functions, $r^S_j = \left\{\mathrm{cap}f^S_j, \mathrm{mem}f^S_j, \mathrm{ban}f^S_j\right\}$. $P^S = \{p^S_1, \cdots, p^S_Z\}$ denotes the performance index of the twin system. According to Theorem 1, when the performance index of the node $p^S_{d_n}$ meets the requirements of the communication system, the node d_n can be divided into the twin domain. The attributes of resources can be expressed as follows:

CPU capf^S_j: indicates the processor resource consumed when processing the function f^S_j.

Memory consumption memf^S_j: indicates the memory consumed when processing the function f^S_j.

Bandwidth banf^S_j: indicates the data rate when transferring the function f^s_j.

Function level levelf^S_j: determines the function level according to the function utilization, which has been defined before using it.

Function cost costf^S_j: When the execution time of a function exceeds the specified acceptable time, the cost of processing the function is increased.

The computing system can be represented as $G^C = \left\{F^C, R^C, P^C\right\}$. $F^C = \left\{f^C_0, f^C_1, \cdots, f^C_J\right\}$ denotes the set of functions running on this computing system. $R^C = \{ \begin{matrix} s_{11}, s_{12}, \ldots, s_{1q}, \ldots, s_{1Q} \\ \ldots \\ s_{P1}, s_{P2}, \ldots, s_{Pq}, \ldots, \ldots, s_{PQ} \end{matrix} \}$ denotes the set of resources available in this computing system. The computing system consists of P servers, each of which is extended with Q docker containers. $P^C = \left\{p^C_1, \cdots, p^C_Z\right\}$ denotes the performance indexes. According to Theorem 1, when the performance index of the node $p^C_{d_n}$ meets the requirements of the communication system, the node d_n can be divided into the computing domain. $s_{pq} = \left\{\mathrm{cap}s_{pq}, \mathrm{mem}s_{pq}, \mathrm{ban}s_{pq}\right\}$

denotes the available resource in the docker container q. The attributes of s_{pq} can be expressed as follows:

CPU caps_{pq}: indicates the available processor resource of the resource s_{pq} in the docker container q.

Memory mems_{pq}: indicates the available memory of the resource s_{pq} in the docker container q.

Bandwidth bans_{pq}: indicates the data rate of the resource s_{pq} in the docker container q.

Resource cost costs_{pq}: indicates the cost of power, software, and equipment spent when processing.

3.2. Correlation between Domains within the Network Simulation Space

According to the parameters of the space and domain, each domain in the simulation space is functionally independent of the other. The independence coefficients of the space D can be expressed as

$$Acvs(D) = \begin{cases} Ccvs(G^{COM}, G^S, G^C), \alpha(G^{COM}) + \alpha(G^S) + \alpha(G^C) > 1 \\ SIM(G^{COM}, G^S, G^C),\ others \end{cases}, \tag{5}$$

where the $Ccvs(G^{COM}, G^S, G^C)$ can be expressed separately as

$$\begin{aligned} Ccvs(G^{COM}, G^S, G^C) &= \sum_{j=1}^{J} \alpha(G^{COM}) \times \log(|r_j^s|) / \log\left\{ -\frac{\max\{R^S\}}{\sum_{p=1}^{p}\sum_{q=1}^{Q}|S_{pq}|} \right\} \\ &= \sum_{j=1}^{J} 0.5T|p_d v_1(R^{COM})| \\ &\times \log(\sqrt{(\mathrm{cap}f_j^S)^2 + (\mathrm{mem}f_j^S)^2 + (\mathrm{ban}f_j^S)^2}) \\ &/ \log\left\{ -\frac{\max\{R^S\}}{\sum_{p=1}^{p}\sum_{q=1}^{Q}\sqrt{(\mathrm{cap}S_{pq})^2 + (\mathrm{mem}S_{pq})^2 + (\mathrm{ban}S_{pq})^2}} \right\}, \end{aligned} \tag{6}$$

The correlation coefficient between domains is related to the extension factor α of the domain and the resource R of the domain. The correlation coefficient of the domain is obtained based on the CPU capf_j, memory memf_j, bandwidth banf_j, and the extension factor α. The SIM(G^{COM}, G^S, G^C) can be expressed as

$$\mathrm{SIM}(G^{COM}, G^S, G^C) = \frac{V_{G^{COM}/G^S}}{V_{G^S/G^C}} = \frac{\int_{p_{min}}^{p_{max}} \alpha(G^{COM}) \cdot \alpha(G^S) \cdot v_{G^{COM}/G^S}(\mathrm{R}) dp}{\int_{p_{min}}^{p_{max}} \alpha(G^S) \cdot \alpha(G^C) \cdot v_{G^S/G^C}(\mathrm{R}) dp}, \tag{7}$$

where $v_{G^{COM}/G^S}(\mathrm{R}) = max\{R^{COM} - sum(R^{COM}/R^S), R^S - sum(R^S/R^{COM})\}$ and $v_{G^S/G^C}(\mathrm{R}) = max\{R^S - sum(R^S/R^C), R^C - sum(R^C/R^S)\}$. We calculate the maximum value of the residual of the ratio of resources between domains and the resources in the domain to obtain the relationship of domains in one space.

3.3. Function–Resource Mapping Model

The essence of the communication–twin–computing integration network mapping is to map the communication network topology to the twin network topology and then to the computation network topology. At the same time, the resources are used to abstractly describe the functions. The function F deployed to the resource R can be expressed as $\mathrm{H(F)} \rightarrow R$ in the domain. The twin network is taken as an example for analysis.

The twin function F^S deployed to the twin resource R^S can be expressed as $\mathrm{H}(F^S) \rightarrow R^S$. We design a flexible coefficient autoregressive model based on the basis function matrix to describe the mapping relationship among the networks.

$$R^S \leftarrow \mathrm{H}(F^S) = B^S \times (A \circ F^S), \tag{8}$$

where A $= (a_1, \cdots a_J)$ is the matrix containing all linear parameters, F^S indicates a function, and $B^S = (b^S_{\text{cap}f}, b^S_{\text{memf}}, b^S_{\text{banf}})$ is denoted as the parameter of the Gaussian basis function.

$$A \circ F^S = (a_1, \cdots a_J) \circ (f_1^S, \cdots, f_J^S) = \sum_{j=1}^{J} a_j \cdot f_1^S +, \cdots, + \sum_{j=1}^{J} a_j \cdot f_J^S, \tag{9}$$

$$b_f^S = \exp\left(-\frac{\|NUM - c_f\|^2}{2\sigma_f^2 \cdot \sum_{z=1}^{Z} p_z^S}\right), \tag{10}$$

In Equation (14), NUM is the number of events required by the twin function, c_f is denoted as the number of resources required to complete a unit event, and σ_f is denoted as the variance of resources required to complete a unit event. Then, the parameter of the Gaussian basis function can be expressed as follows:

$$b^S_{\text{capf}} = \exp\left(-\frac{\|NUM - c_{capf}\|^2}{2\sigma_{capf}^2 \cdot \sum_{z=1}^{Z} p_z^S}\right), \tag{11}$$

$$b^S_{\text{memf}} = \exp\left(-\frac{\|NUM - c_{memf}\|^2}{2\sigma_{memf}^2 \cdot \sum_{z=1}^{Z} p_z^S}\right), \tag{12}$$

$$b^S_{\text{banf}} = \exp\left(-\frac{\|NUM - c_{banf}\|^2}{2\sigma_{banf}^2 \cdot \sum_{z=1}^{Z} p_z^S}\right), \tag{13}$$

The mapping relationship of the twin resources and functions can be denoted as

$$\begin{aligned} R^s = B^s \times (A \circ F^s) = & \begin{pmatrix} b^S_{\text{capf}} \\ b^S_{\text{memf}} \\ b^S_{\text{banf}} \end{pmatrix} \times \left((a_1, \cdots a_J) \cdot f_1^S, \cdots, (a_1, \cdots a_J) \cdot f_J^S \right) \\ = & \begin{pmatrix} \sum_{j=1}^{J} a_j \cdot f_1^S \cdot b^S_{\text{capf}} & \cdots & \sum_{j=1}^{J} a_j \cdot f_J^S \cdot b^S_{\text{capf}} \\ \sum_{j=1}^{J} a_j \cdot f_1^S \cdot b^S_{\text{memf}} & \ddots & \sum_{j=1}^{J} a_j \cdot f_J^S \cdot b^S_{\text{memf}} \\ \sum_{j=1}^{J} a_j \cdot f_1^S \cdot b^S_{\text{banf}} & \cdots & \sum_{j=1}^{J} a_j \cdot f_J^S \cdot b^S_{\text{banf}} \end{pmatrix} = (r_1^S, \cdots, r_j^S) \\ = & \begin{pmatrix} \text{cap}f_1^S & \cdots & \text{cap}f_j^S \\ \text{mem}f_1^S & \ddots & \text{mem}f_j^S \\ banf_1^S & \cdots & banf_j^S \end{pmatrix}, \end{aligned} \tag{14}$$

where the row is the CPU, memory, and bandwidth required by the function f_j^S, and a_j is the mapping factor of the function f_j^S, with values in [0, 1].

4. System Optimization Scheme Based on A Multi-Constraint Multi-Objective Genetic Algorithm

In this section, the three optimization objectives of maximizing function utilization, maximizing load balancing, and minimizing total cost are first introduced, and then a multi-objective constrained genetic algorithm (MMGA) is proposed to optimize the system objectives to achieve a dynamic balance of resources and task demands under the constraints of system resources.

4.1. Problem Formulation

In the communication–twin–computing integration network, the network function is mapped to the docker containers according to the simulation requirements and the computing resources. The CPU, memory, and bandwidth of the docker containers can be viewed as fixed values before mapping. As shown in Figure 3, the resource in a docker

container can be viewed as a box. Thus, we can view the problem of mapping the twin functions to the computing resources as a function–resource three-dimensional packing problem. However, unlike the ordinary three-dimensional packing problem, the function–resource three-dimensional packing problem can only be packed along the diagonal. When the resources of a certain dimension are consumed, it is not possible to continue restarting a new layer of packing. In order to find the optimal solution, we focus on three main objectives: maximizing function utilization, maximizing load balancing, and minimizing the total cost in the 6G satellite twin network.

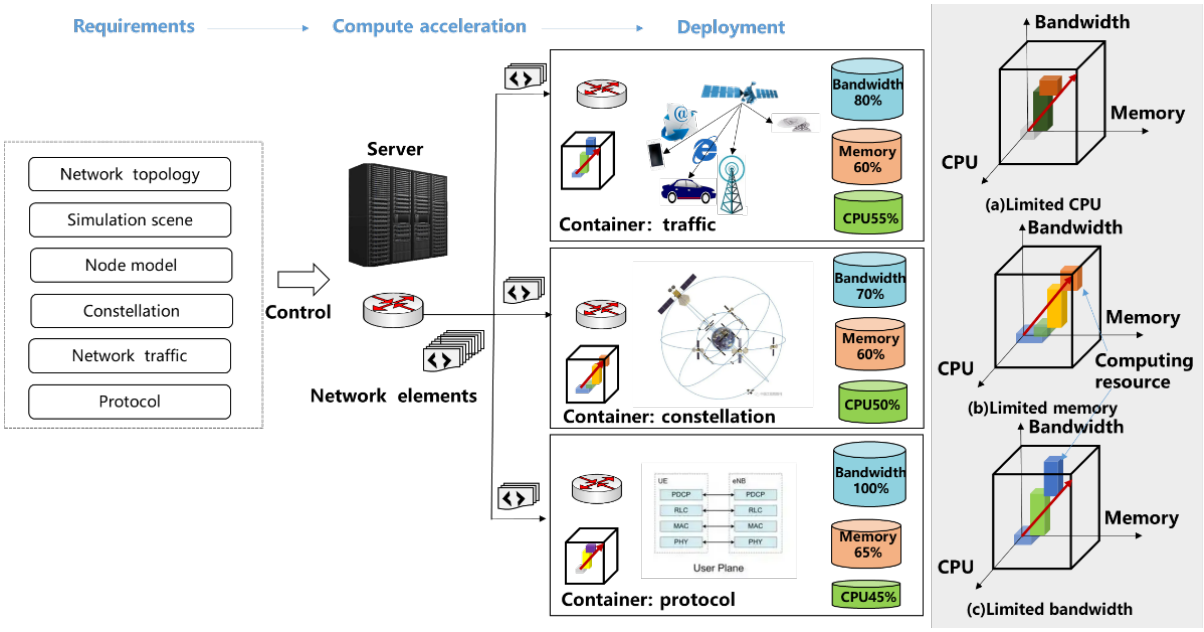


Figure 3. Function–resource three-dimensional packing model.

(1) **Function utilization**

There exists a set of functions that satisfy the event requirements $F' = \left\{f_1, f_2, \ldots, f_j^S, \ldots f_J^S\right\}$ in the domain G, where each function corresponds to the set of function utilization values $\{futl_1, futl_2, \ldots, futl_j, \ldots futl_J\}$. Then, the system function utilization can be expressed as

$$\text{utl}F' = \sum_{j=1}^{J} \frac{1}{J} \cdot \text{level} f_j^S \cdot futl_j, \tag{15}$$

Since each QoS has a different effect on the function utilization, each function has a weight of w_{ij}. Then, the function utilization $futl_j$ can be expressed as

$$futl_j = \sum_{i=1}^{3} w_{ij} \cdot \frac{QoS_i - QoS_j}{QoS_i}, \tag{16}$$

where $QoS_j = \left\{\text{cap} f_j, \text{mem} f_j, \text{ban} f_j, \text{cos t} f_j^S\right\}$. The QoS_i denotes the resources and the cost assigned to the function f_j, and the QoS_j denotes the resources and the cost consumed by the function f_j.

(2) **Load balancing**

There are two main components in the 6G satellite twin network, namely, distribution balancing of load among individual servers and distribution balancing of load among different docker containers within the same server. The former is used to prevent overloading

of individual servers, and the latter is used to prevent overloading of docker containers to ensure efficient utilization of resources. Moreover, when the CPU or memory of a server or docker container is exhausted, the remaining resources, no matter how much is left, can no longer provide services.

The load balancing of all servers can be expressed as

$$loadS = (\psi(cap) + \psi(mem) + \psi(ban)) \times e^{Acvs(D)}, \tag{17}$$

$$\psi(cap) = \sqrt{\frac{1}{P} \cdot \sum_{p=1}^{P} (\mathrm{cap}F' / \sum_{q=1}^{Q} \mathrm{cap}s_{pq} - \overline{capF'/\mathrm{cap}S})^2}, \tag{18}$$

$$\psi(mem) = \sqrt{\frac{1}{P} \cdot \sum_{p=1}^{P} (memF' / \sum_{q=1}^{Q} mems_{pq} - \overline{memF'/mem}S)^2}, \tag{19}$$

$$\psi(ban) = \sqrt{\frac{1}{P} \cdot \sum_{p=1}^{P} (banF' / \sum_{q=1}^{Q} bans_{pq} - \overline{banF'/ban}S)^2}, \tag{20}$$

where $\mathrm{cap}F' / \sum_{q=1}^{Q} \mathrm{cap}s_{pq}$, $memF' / \sum_{q=1}^{Q} mems_{pq}$, $banF' / \sum_{q=1}^{Q} bans_{pq}$ denote the utilization of CPU, the utilization of memory, and the utilization of bandwidth in the server p, respectively. $\overline{capF'/\mathrm{cap}S}$, $\overline{memF'/mem}S$, $\overline{banF'/ban}S$ denote the average utilization of CPU, the average utilization of memory, and the average utilization of bandwidth, respectively, which can be obtained by calculating the average resource in all servers and the average requirements of all functions.

Load balancing of the docker containers in server q can be expressed as

$$loadC = \frac{1}{Q} \cdot \sum_{q=1}^{Q} (\max\left(\frac{capF'}{caps_{pq}}, \frac{memF'}{mems_{pq}}, \frac{banF'}{bans_{pq}}\right) - \min(\frac{capF'}{caps_{pq}}, \frac{memF'}{mems_{pq}}, \frac{banF'}{bans_{pq}})), \tag{21}$$

where $\frac{capF'}{caps_{pq}}$, $\frac{memF'}{mems_{pq}}$, $\frac{banF'}{bans_{pq}}$ denote the utilization of CPU, the utilization of memory, and the utilization of bandwidth in the docker container p of the server q, respectively

(3) **Total cost**

In this paper, the cost of the 6G satellite twin network is divided into two main parts: one part is the cost of power, software, and equipment, and the other part is the increased cost due to the execution time of the function beyond the specified acceptable time. The total cost required for this network can be expressed as

$$\mathrm{cos\,t} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \mathrm{cos\,t}s_{pq} + \sum_{j=1}^{J} \mathrm{cos\,t}f_j^S, \tag{22}$$

The three optimization objectives in this paper can be formally expressed as follows:

$$\max(\mathrm{utl}F') = max \sum_{j=1}^{J} \frac{1}{J} \cdot \mathrm{level}f_j^S \cdot (\sum_{i=1}^{4} w_{ij} \cdot \frac{QoS_i - QoS_j}{QoS_i}), \tag{23}$$

$$
\begin{aligned}
max(loadS + loadC) & \\
&= max((\sqrt{\frac{1}{P} \cdot \sum_{p=1}^{P} \left(\mathrm{cap}F' / \sum_{q=1}^{Q} \mathrm{cap}s_{pq} - \overline{capF'/\mathrm{cap}S} \right)^2} \\
&+ \sqrt{\frac{1}{P} \cdot \sum_{p=1}^{P} \left(memF' / \sum_{q=1}^{Q} mems_{pq} - \overline{memF'/memS} \right)^2} \\
&+ \sqrt{\frac{1}{P} \cdot \sum_{p=1}^{P} \left(banF' / \sum_{q=1}^{Q} bans_{pq} - \overline{banF'/banS} \right)^2}) \times e^{Acvs(D)} \\
&+ \frac{1}{Q} \cdot \sum_{q=1}^{Q} (\max\left(\frac{capF'}{caps_{pq}}, \frac{memF'}{mems_{pq}}, \frac{banF'}{bans_{pq}} \right) \\
&- \min(\frac{capF'}{caps_{pq}}, \frac{memF'}{mems_{pq}}, \frac{banF'}{bans_{pq}}))),
\end{aligned}
\tag{24}
$$

$$
\min(\cos \mathrm{t}) = \min(\sum_{p=1}^{P} \sum_{q=1}^{Q} \cos \mathrm{t}s_{pq} + \sum_{j=1}^{J} \cos \mathrm{t}f_j), \tag{25}
$$

s.t.

$$
\mathrm{cap}S \geq \mathrm{cap}F', \tag{26}
$$

$$
\mathrm{mem}S \geq \mathrm{mem}F', \tag{27}
$$

$$
\mathrm{ban}S \geq \mathrm{ban}F', \tag{28}
$$

4.2. The Multi-Constraint Multi-Objective Genetic Algorithm

To efficiently allocate resources in the 6G satellite twin network, a multi-objective optimization model that considers function utilization, load balancing, and the cost is developed. Multi-objective optimization problems are basically discussed based on Pareto optimal solutions, where multiple-objective components are composed into a vector and the minimization of the vector is solved. Based on the above idea, the objective problem is modeled as follows:

$$
\begin{aligned}
\min f(F, S) \quad & = \min(f_1, f_2, f_3) \\
& = \min((1 - \max(\mathrm{utl}F')),\ \min(1 - max(loadS + loadC)), (\min(\cos \mathrm{t}))),
\end{aligned}
\tag{29}
$$

In contrast to single-objective optimization problems, multi-objective optimization problems have difficulty finding a solution that makes all the objective functions optimal at the same time due to the conflict between the objectives. Therefore, multi-objective optimization problems usually have a solution set. Each solution in the solution set is the non-dominated solution of all the objective functions. It is impossible to compare the advantages and disadvantages of each solution for the whole objective function.

A multi-constraint multi-objective genetic algorithm is proposed to find a non-dominated solution to the problem. We use a non-dominated hierarchical approach, which allows more suitable non-dominated solutions in each layer to have a greater chance of being inherited to the next selection during the solution process. At the same time, the fitness sharing strategy is used to make the non-dominated solutions in each layer uniformly distributed to maintain population diversity, avoid overpopulation of super individuals, and prevent premature convergence.

In the ath-level domination layer there exists $\mathrm{b_a}$, the non-dominated solution; the distance between the two solutions, $(F'_m, \mathrm{S_m})$ and $(F'_n, \mathrm{S_n})$, which represents the deployment of the function in the resource, can be expressed as

$$
\mathrm{d(m,n)} = \sqrt{(\frac{(F'_m, \mathrm{S_m}) - (F'_n, \mathrm{S_n})}{(F'_m, \mathrm{S_m})^u - (F'_m, \mathrm{S_m})^d})^2}, \mathrm{m, n} = 1, 2, \ldots, \mathrm{b_a}, \tag{30}
$$

where $(F'_m, \mathrm{S_m})^u$ and $(F'_m, \mathrm{S_m})^d$ are the upper limit and the lower limit of the $(F'_m, \mathrm{S_m})$, respectively.

The shared function SH represents the relationship between the non-dominated solution $(F'_m, \mathrm{S_m})$ and other non-dominated solutions within this layer, denoted as

$$\mathrm{SH(d(m,n))} = \begin{cases} 1 - \left(\frac{d(m,n)}{r_{ave}}\right)^{\beta} d(m,n) < r_{ave} \\ 0 \quad other \end{cases}, \tag{31}$$

where the r_{ave} is the average of the non-dominated solution $(F'_m, \mathrm{S_m})$ and the radii of other non-dominated solutions within this layer, and β is a constant.

The number of non-dominated solutions in this dominated layer can be expressed as

$$c_m = \sum_{m=1}^{\mathrm{b_a}} \mathrm{s(d(m,n))}, \tag{32}$$

The shared fitness of the non-dominated solution $(F'_m, \mathrm{S_m})$ can be expressed as

$$h_m = \frac{\mathrm{s(d(m,n))}}{c_m}, \tag{33}$$

A selection strategy based on the k-means algorithm and proportional distribution is used in each level of the dominance layers to select the optimal solution from the elite pool so that individuals are distributed along the Pareto front. The results in this dominance layer can be expressed as

$$KNum_m = sgn(h_m - \sqrt{\sum_{k=1}^{3} \left[\frac{f_k((F'_m, \mathrm{S_m})) - f_k((F'_n, \mathrm{S_n}))}{f_k^{max}(F,S) - f_k^{min}(F,S)} \right]^2}), \tag{34}$$

where $f_k((F'_m, \mathrm{S_m}))$ is the solution of the kth objective function. $f_k^{max}(F,S)$ and $f_k^{min}(F,S)$ are the maximum and minimum values of the kth objective function, respectively. sgn is the sign function, which means that when the unit ball contains more dominated solutions, $KNum_m$ will be larger.

5. Serverless-Based Decentralized Simulation Development Model

In this section, we describe the design of the three-dimensional hierarchical simulation framework and the decentralized adaptive development model. The component architecture and application process of the 6G satellite twin network are introduced.

5.1. The Three-Dimensional Hierarchical Simulation Framework

The 6G satellite twin network includes simulation network initialization, simulation task segmentation, simulation task assignment mapping, function combination, resource balance coordination, and simulation task process. In this paper, a three-dimensional hierarchical simulation framework is proposed, as shown in Figure 4. In this framework, CPU, memory, and bandwidth are collectively referred to as resources. The resources are gridded according to the distribution of the docker containers of the servers and modeled as $s_{pq} = \{\mathrm{cap}s_{pq}, \mathrm{mem}s_{pq}, \mathrm{ban}s_{pq}\}$. All tasks of the docker containers are performed by the function $F_{pq}^C = \left\{f_0^C, f_1^C, \cdots, f_J^C\right\}$. The resources support the functions to be executed. The docker containers interact according to the contact of the function.

In the 6G satellite twin network, we decouple resources, transport, control, and task. The resources of each docker container are logically layered according to function into the control management layer, service resource layer, and network ontology layer.

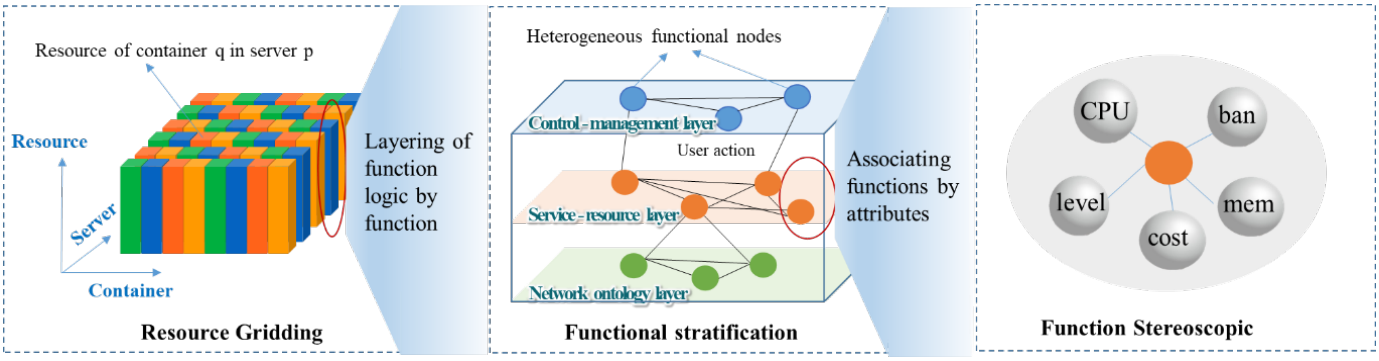


Figure 4. The three-dimensional hierarchical simulation framework.

Control-management layer: This layer includes the functions for network architecture. The simulation network initialization, the connection between different docker containers, and the transmission of communication data and control information between docker containers are carried by this layer.

Service-resource layer: This layer consists of task requirement discovery functions and resource orchestration mapping functions. The hardware resources are allocated to the functions. This layer completes the load balancing between different servers and docker containers.

Network ontology layer: This layer includes the ontology functions of the 6G satellite twin network. The network topology, simulation scenarios, node models, constellations, network traffic, and protocols are simulated.

A 6G satellite twin network is decomposed into multiple sub-simulation task sets distributed into different docker containers. Different simulation tasks are run in different docker containers. The differences in the functions that make up the simulation tasks lead to task differences between the docker containers. The consistency of the functions in the docker containers varies.

5.2. The Decentralized Adaptive Development Model

The development of the 6G satellite twin network needs to consider the support and connection between the communication system, the twin system, and the computing system. The global control and management of the communication–twin–computing integrated network are achieved through the grid-based management of resources and the logical hierarchy of functions. Thus, various requirements in the integrated network can be further developed and described by abstracting them into virtualized functions. Based on the analysis, we propose a new decentralized adaptive development model, as shown in Figure 5.

As shown in Figure 5, the proposed model includes an ontology layer, a twin layer, and a computing layer. The ontology of the simulation can be abstracted into behaviors and functions. The ontology layer of the 6G satellite twin network in this paper can be abstracted into communication behaviors such as access, handover, and routing. The communication behaviors are implemented logically through communication functions such as topology, constellation, and protocol to simulate the communication capability. The twin layer is the control and mapping of the twin network. The control and management functions such as configuration, interface, and cache in the twin network are represented by functions. The optimal allocation of functions and resources is calculated according to the resource requirements of all functions to achieve the optimal utilization of resources in the 6G satellite twin network, that is, on the basis that the resources consumed by the functions do not exceed the resource capacity provided by all servers, as many simulation tasks as possible are completed, and the tasks allocated in each docker container are balanced more effectively to achieve the optimal simulation time or optimal utilization of resources. In the computing layer, functions are deployed according to the optimal allocation of resources and functions to meet the execution of simulation tasks.

The development idea of the model is to establish the mapping relationship among functions and resources by sensing the correlation among ontology function models. Based on the mapping relationships, the function–resource adaptation relationships are calculated and the functions are deployed into the corresponding docker containers.

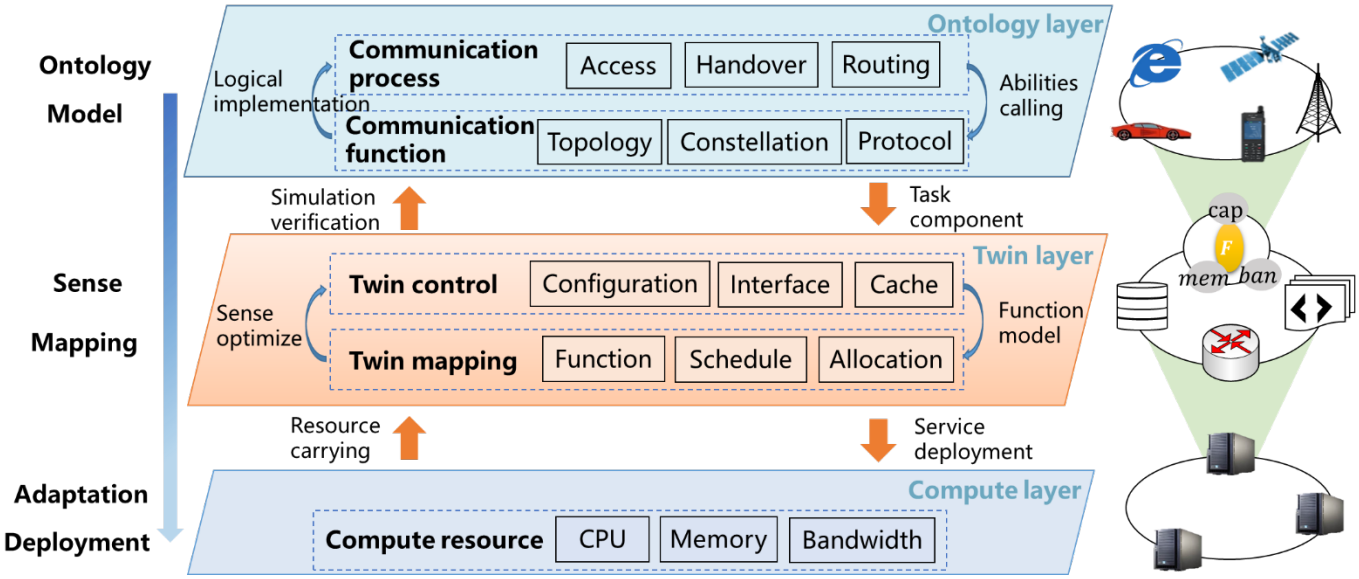


Figure 5. The decentralized adaptive development model.

6. Results and Discussions

As shown in Figure 6, in order to test the performance of the 6G satellite twin network, we built a 6G satellite twin network simulation platform consisting of servers, switches, control nodes, constellation, and protocol. As shown in Table 1, we used the servers with a core i7 4.6 GHz CPU, 8 GB RAM, and 512 GB SSD. We considered 1000 users with a data rate of 100 kbps, mainly distributed in China, Europe, and America. A Walker constellation of 64 satellites deploying 1200 km altitude was built, distributed in eight planes on average. RRC/SDAP/PDCP/RLC/MAC/PHY protocol layers are running in the satellites and users. A parallel computing platform with serverless architecture is used for simulation.

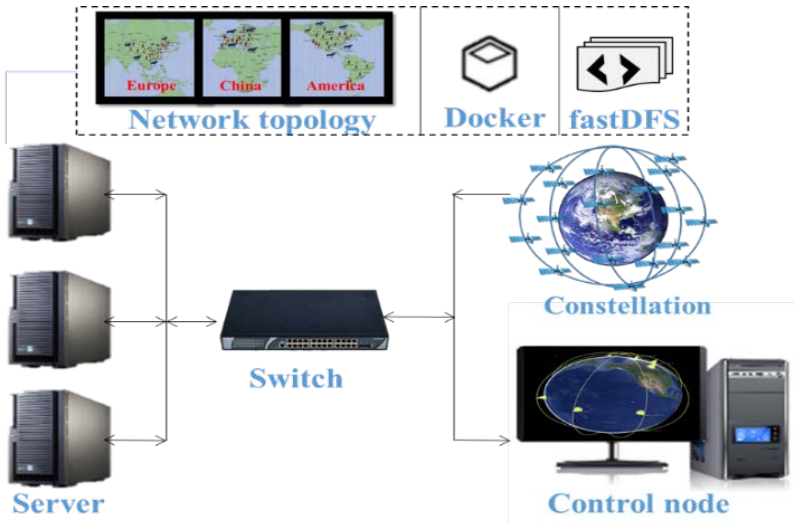


Figure 6. Simulation platform.

Table 1. Simulation parameters configuration.

Classification	Configuration	Parameter
Hardware	CPU	i7 4.6 GHz
	Memory	8 GB
	SSD	512 GB
Simulation	Constellation	Walker
	Satellite altitude	1200 km
	Number of satellites	64
	Number of satellite planes	8
	Distribution of users	Europe, China, America Poisson distribution
	Number of users	1000
	Protocol	RRC/SDAP/PDCP/RLC/MAC
	Code	LDPC/Polar
	Modulation	BPSK/QPSK

In this paper, we chose two traditional methods for solving optimization problems, namely, the simulation annealing algorithm (SA) and the particle swarm optimization algorithm (PSO), to compare with the MMGA proposed in this paper. The SA [84–86] and the PSO [87–89] are also frequently used in network mapping problems for objective optimization. In addition, we chose the virtual network mapping algorithm based on hierarchical relationship (NFA-LR) [90] and virtual network embedding based on attributes and resources for single-stage (VNE-ARS) [91] for comparison with the MMGA. We compare the normalized utilization of terrestrial networks and non-terrestrial networks in terms of access, handover, and routing behaviors. The normalized utilization of CPU, memory, and bandwidth with different algorithms is shown in Figure 7. It can be seen from Figure 7 that the MMGA, which is used to find the optimal utilization, can effectively improve resource utilization compared with other algorithms. At the same time, the normalized utilization of non-terrestrial networks is higher than that of terrestrial networks in terms of CPU, memory, and bandwidth. This is because unlike the terrestrial network, constellations are added to the non-terrestrial network. Satellites in a non-terrestrial network can perform different functions. When the satellite has the transparent forwarding function, the satellite is responsible for data forwarding between the terminal and the base station, which requires less processing capability for the satellite. When the satellite is used as the central unit processing function of the onboard base station, the satellite is mainly responsible for the access function of the user terminal, but it also needs to forward the access information to the ground base station to complete the data return. When a satellite acts as an onboard base station with all base station functions, the satellite is responsible for user terminal access and user data backhaul services. In the latter two functions, the processing capability of the satellite is required to be higher. In addition, the bandwidth utilization of the handover behaviors is significantly higher than that of the access and handover behaviors, which is due to that more information is exchanged between different network nodes in the handover behavior than in the access and routing behaviors. The memory utilization in the routing scenario is higher than that in the access and handover behaviors due to the need to store node states for routing.

The time consumed by using SA, PSO, MMGA, VNE-ARS, and NFA-LR algorithms in different scenarios is shown in Figure 8. The time running the non-terrestrial network is significantly higher than that running the terrestrial network. The MMGA algorithm proposed in this paper takes less time than the rest of the algorithms. This is because the time is taken as cost as the optimization objective in this paper. The optimal resource allocation scheme is obtained in the least amount of time.

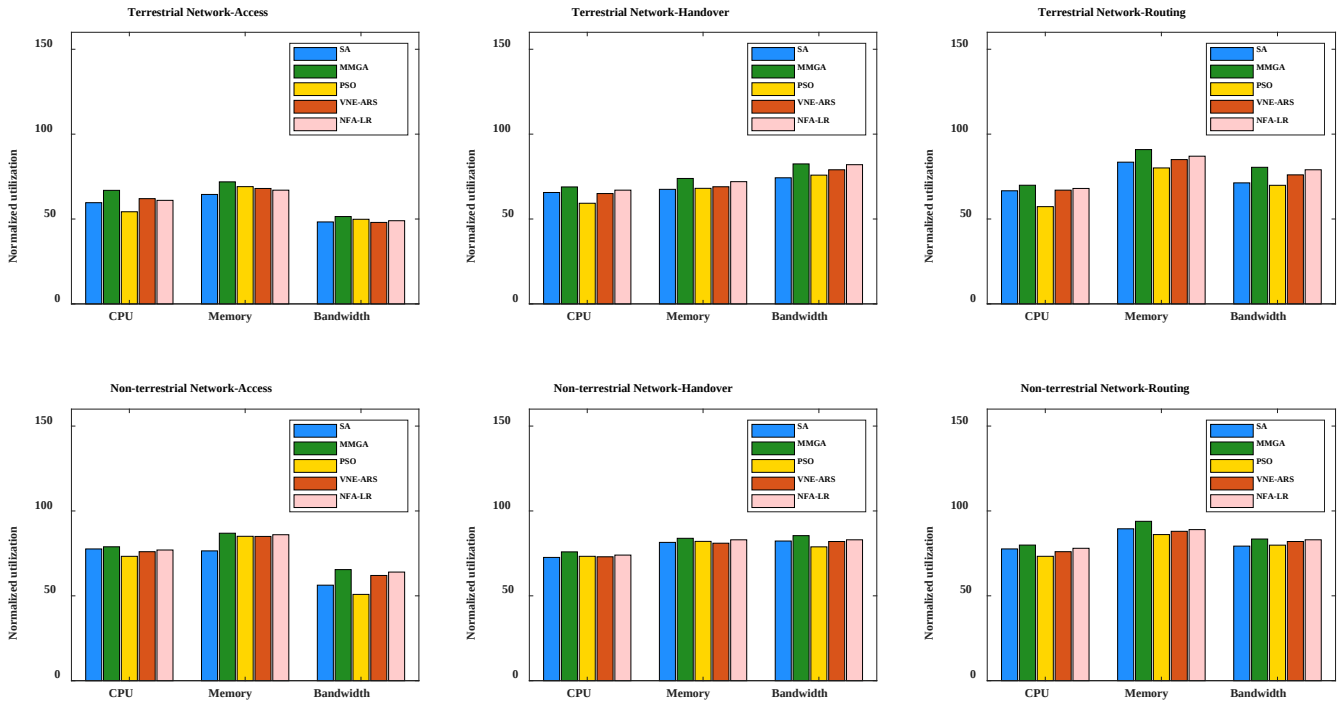


Figure 7. The CPU, memory, and bandwidth normalized utilization of different algorithms under access, handover, and routing behaviors in the terrestrial network and the non-terrestrial network.

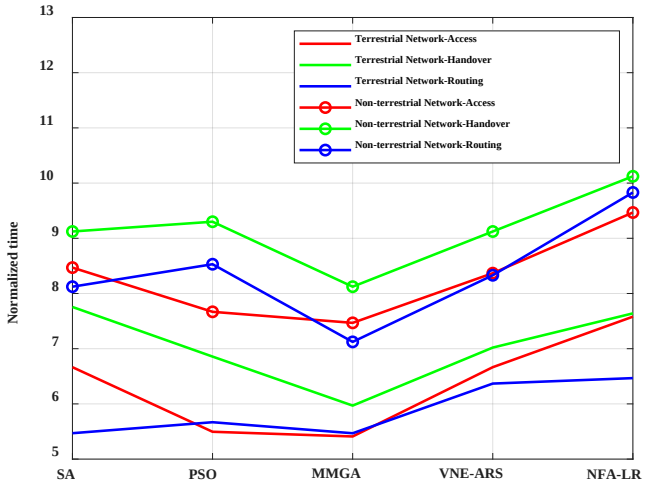


Figure 8. Normalization time consumed for different algorithms.

In Figure 9, we verify the effect of different algorithms on the normalized utilization of CPU, memory, and bandwidth for the different number of iterations. We can observe that as the number of iterations increases, the CPU normalized utilization of the PSO, the VNE-ARS, and the MMGA increases to a smooth maximum. However, as the number of iterations increases, the CPU normalized utilization of the SA increases and then decreases. At the same time, as the number of iterations increases, the memory normalized utilization of the SA and the PSO increases and then decreases, and the memory normalized utilization of the VNE-ARS and the MMGA increases to a smooth maximum. In addition, the bandwidth normalized utilization of the SA, the PSO, the VNE-ARS, and the MMGA increases to a smooth maximum. This is due to the overconvergence of the SA as the number of iterations increases. Since we assign the maximum weight to the optimal solution of each layer in the solution of the MMGA, it has a better chance of being inherited to the next layer, avoiding the overproduction of super-individuals and preventing premature

convergence. Meanwhile, in the MMGA, the solutions with the lowest 10% weight values are eliminated at each iteration. In this way, as the number of iterations increases, the number of individuals to be searched for the best gradually decreases and the resources consumed do not show a linear increase. However, due to the increase in the number of computations, resources are still consumed. In addition, no iteration is performed in the NFA-LR, so the normalized utilization of CPU, memory, and bandwidth does not change as the number of iterations increases.

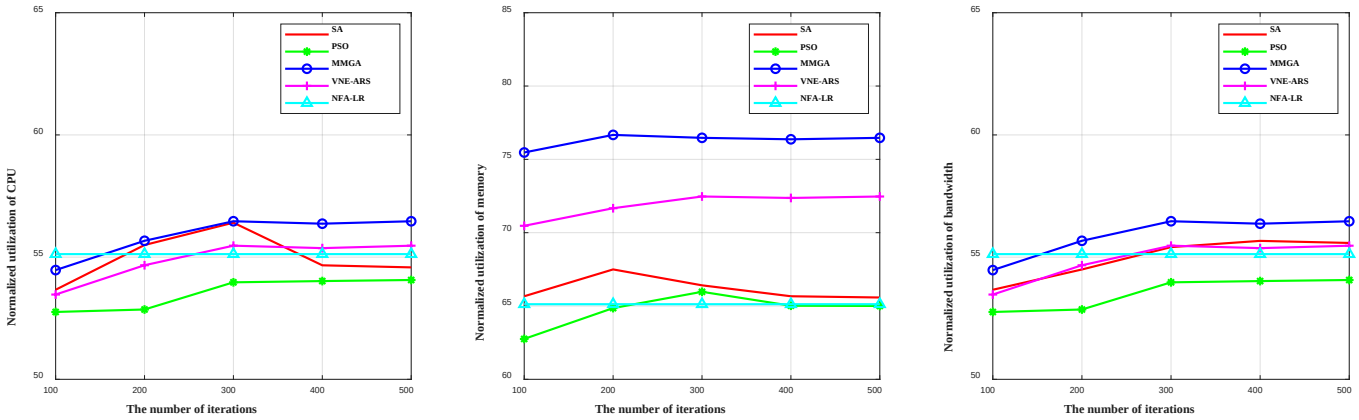


Figure 9. The CPU, memory, and bandwidth normalized utilization of different algorithms under the different number of iterations.

From Figure 10, we can observe that the time consumed by different algorithms varies as the number of iterations increases. This is due to the increase in the number of iterations leading to an increase in the optimal result of each algorithm. In addition, the SA, the PSO, the VNE-ARS, and the NFA-LR are highly affected by the iteration time. The MMGA is less affected by the iteration time. This is because the cost of consumption is used as a limiting condition for finding the optimal result in the MMGA. Minimization time is considered in the search for the optimal allocation result.

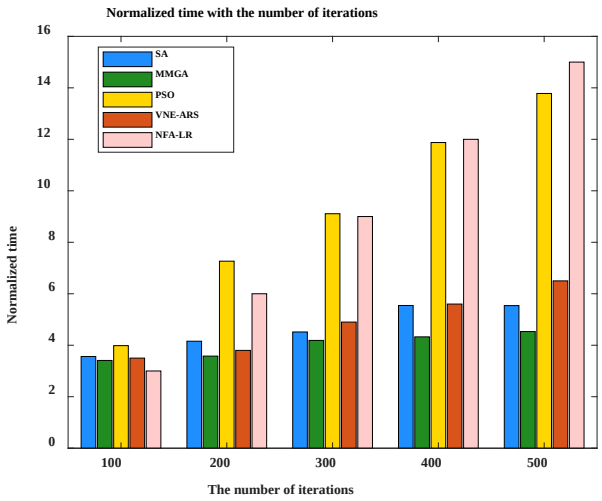


Figure 10. Normalization time consumed for different numbers of iterations.

In this paper, we simulate the communication system using standalone architecture, parallel architecture, and serverless architecture. The performance is shown in Figure 11. As shown in Figure 11, the normalized utilization using standalone architecture is much smaller than that using the parallel and serverless architectures among CPU, memory, and bandwidth. The normalized utilization of the parallel architecture and serverless architecture is the same, because the servers of these architectures can process the tasks

synchronously, but the tasks of standalone architecture can only be processed in one server at the same time. In addition, the processing time of the standalone architecture is the largest. The transmission of control information increases the processing time due to a master control node needing to control all servers in the parallel architecture. The control information transmission time of serverless architecture is less than that of parallel architecture due to the means of decentralized control, so the serverless architecture consumes the least simulation time.

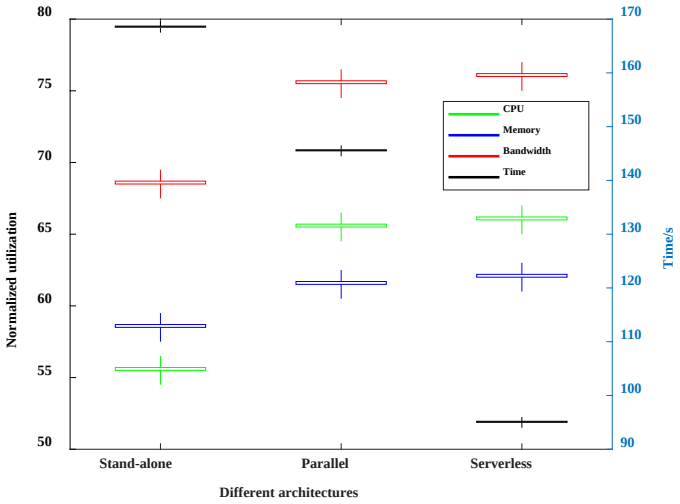


Figure 11. Computing performance under different architectures.

7. Conclusions

In this paper, we discuss the theoretical analysis and modeling of the 6G satellite twin network, in which the key issues of the communication–twin–computing integration network model are investigated. A hypergraph hierarchical nested kriging model and a hierarchical unified feature description method are proposed to describe the communication network and twin network and computing network. Meanwhile, the basis function matrix of the local flexible connection of the global network is introduced to map the functions to the resource. In addition, the optimal mapping problem is described as a three-dimensional packing problem, and the multi-constraint multi-objective genetic algorithm is proposed to find the optimal computational resource allocation. Finally, the three-dimensional hierarchical simulation framework and the decentralized adaptive development model are proposed.

Author Contributions: Conceptualization, X.Y. and Z.D.; methodology, X.Y.; software, X.Y.; validation, X.Y. and Z.D.; formal analysis, Z.D.; investigation, Z.D.; resources, Z.D.; data curation, Z.D.; writing—original draft preparation, X.Y.; writing—review and editing, X.Y.; visualization, Z.D.; supervision, X.Y.; project administration, X.Y.; funding acquisition, Z.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the China National Key Research and Development Plan (2020YFB1808005 and 2019YFB1803105).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

There is a set of performance lower bounds $\boldsymbol{l^G} = \left[l_1^G, \cdots l_J^G\right]^T$ for the domain G with the mean μ and the variance σ^2. The correlation of the domain G can be expressed as

$$\begin{aligned} Ccvs(G^1, G^2) &= \sum_{j=1}^{J} \alpha(G^1) \times \log\left(|\boldsymbol{r}_j^1|\right) / \log\left\{-\frac{\max\left\{R^2\right\}}{|\boldsymbol{r}_j^1|}\right\} \\ &= \sum_{j=1}^{J} \alpha(G^1) \times \log(\sqrt{(\mathrm{cap} f_j^1)^2 + (\mathrm{mem} f_j^1)^2 + (\mathrm{ban} f_j^1)^2}) \\ &\quad / \log\left\{-\frac{\max\left\{R^2\right\}}{\sqrt{(\mathrm{cap} f_j^1)^2 + (\mathrm{mem} f_j^1)^2 + (\mathrm{ban} f_j^1)^2}}\right\} \end{aligned} \tag{A1}$$

The maximum likelihood value is

$$\frac{1}{(2\pi)^{\frac{J}{2}}(\sigma^2)^{\frac{J}{2}}|Ccvs|^{\frac{1}{2}}} \exp\left[\frac{-(\boldsymbol{l^G} - \mathbf{1}\mu)^T Ccvs^{-1}(\boldsymbol{l^G} - \mathbf{1}\mu)}{2\sigma^2}\right], \tag{A2}$$

Taking the logarithm of the above equation,

$$-\frac{J}{2}\log\left(\sigma^2\right) - \frac{1}{2}\log\left(|Ccvs|\right) - \frac{-(\boldsymbol{l^G} - \mathbf{1}\mu)^T Ccvs^{-1}(\boldsymbol{l^G} - \mathbf{1}\mu)}{2\sigma^2}, \tag{A3}$$

The mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ of the maximum likelihood function can be calculated by the partial derivatives of the mean μ and variance $\hat{\sigma}^2$ of the above equation:

$$\hat{\mu} = \frac{\mathbf{1}^T Ccvs^{-1}\boldsymbol{l^G}}{\mathbf{1}^T Ccvs^{-1}\mathbf{1}}, \tag{A4}$$

$$\hat{\sigma}^2 = \frac{(\boldsymbol{l^G} - \mathbf{1}\hat{\mu})^T Ccvs^{-1}(\boldsymbol{l^G} - \mathbf{1}\hat{\mu})}{n}, \tag{A5}$$

The logarithmic maximum likelihood value is

$$-\frac{J}{2}\log\left(\hat{\sigma}^2\right) - \frac{1}{2}\log\left(|Ccvs|\right), \tag{A6}$$

The covariance matrix of the domain G can be expressed as

$$\widetilde{Ccvs} = \begin{pmatrix} Ccvs & Ccvs^{-1} \\ Ccvs^{-1} & \mathbf{1} \end{pmatrix}, \tag{A7}$$

$\boldsymbol{l^{G^*}}$ is denoted as the target value. The logarithmic maximum likelihood value is

$$-\frac{J}{2}\log\left(\sigma^2\right) - \frac{1}{2}\log\left(|Ccvs|\right) - \frac{-(\boldsymbol{l^{G^*}} - 1\hat{\mu})^T \widetilde{Ccvs}^{-1}(\boldsymbol{l^G} - \mathbf{1}\mu)}{2\hat{\sigma}^2}, \tag{A8}$$

Make C = $\left(1 - Ccvs^{-1T} CcvsCcvs^{-1}\right)^T$. Then, the inverse of the covariance matrix can be expressed as

$$\widetilde{Ccvs}^{-1} = \begin{pmatrix} Ccvs + CCcvs^{-1} & -CcvsC \\ -CCcvs^{-1} & C \end{pmatrix}, \tag{A9}$$

The expanded likelihood function is

$$\left[\frac{-1}{2\hat{\sigma}^2(1 - Ccvs^{-1T} CcvsCcvs^{-1})}\right](l^{G^*} - \hat{\mu})^2 + \left[\frac{Ccvs^{-1}Ccvs(\boldsymbol{l^G} - \boldsymbol{1}\hat{\mu})}{\hat{\sigma}^2(1 - Ccvs^{-1T} CcvsCcvs^{-1})}\right](l^{G^*} - \hat{\mu}), \tag{A10}$$

The above equation is the quadratic function of l^{G^*}. The derivative of the above equation, and equal to 0, can be obtained:

$$\left[\frac{-1}{2\hat{\sigma}^2(1-Ccvs^{-1T}CcvsCcvs^{-1})}\right](l^{G^*}-\hat{\mu})+\left[\frac{Ccvs^{-1}Ccvs(\boldsymbol{l^G}-\mathbf{1}\hat{\mu})}{\hat{\sigma}^2(1-Ccvs^{-1T}CcvsCcvs^{-1})}\right]=0, \quad \text{(A11)}$$

The solution is

$$l^{G^*}=\hat{\mu}+Ccvs^{-1T}Ccvs(\boldsymbol{l^G}-\mathbf{1}\hat{\mu}), \quad \text{(A12)}$$

Appendix B

There is a set of edges connecting nodes R = $\{\boldsymbol{r_1},\cdots,\boldsymbol{r_P}\}$. The vector sets $\boldsymbol{r_1},\cdots,\boldsymbol{r_P}$ are linearly independent.

Make $\boldsymbol{b_1}=\boldsymbol{r_1}$, then there exists

$$\|\boldsymbol{r_2'}\|=\|\boldsymbol{r_2}\|\boldsymbol{\cos\theta}=\|\boldsymbol{r_2}\|\frac{[\boldsymbol{b_1},\boldsymbol{r_2}]}{\|\boldsymbol{b_1}\|\cdot\|\boldsymbol{r_2}\|}=\frac{[\boldsymbol{b_1},\boldsymbol{r_2}]}{\|\boldsymbol{b_1}\|}, \quad \text{(A13)}$$

then

$$\boldsymbol{b_2}=\boldsymbol{r_2}-\boldsymbol{r_2'}=\boldsymbol{r_2}-\|\boldsymbol{r_2'}\|\frac{\boldsymbol{r_2'}}{\|\boldsymbol{r_2'}\|}=\boldsymbol{r_2}-\frac{[\boldsymbol{b_1},\boldsymbol{r_2}]}{\|\boldsymbol{b_1}\|}\frac{\boldsymbol{b_1}}{\|\boldsymbol{b_1}\|}=\boldsymbol{r_2}-\frac{[\boldsymbol{b_1},\boldsymbol{r_2}]}{[\boldsymbol{b_1},\boldsymbol{b_1}]}\boldsymbol{b_1}, \quad \text{(A14)}$$

$$\boldsymbol{b_p}=\boldsymbol{r_p}-\frac{[\boldsymbol{b_1},\boldsymbol{r_p}]}{[\boldsymbol{b_1},\boldsymbol{b_1}]}\boldsymbol{b_1}-\frac{[\boldsymbol{b_2},\boldsymbol{r_p}]}{[\boldsymbol{b_2},\boldsymbol{b_2}]}\boldsymbol{b_2}-\cdots-\frac{[\boldsymbol{b_{r-1}},\boldsymbol{r_p}]}{[\boldsymbol{b_{r-1}},\boldsymbol{b_{r-1}}]}\boldsymbol{b_{r-1}}, \quad \text{(A15)}$$

The orthogonal vector of R can be expressed as $v(R)=\{\boldsymbol{b_1},\cdots,\boldsymbol{b_P}\}$

Then, the extension factor of the domain G can be expressed as

$$\begin{aligned}\alpha(G)=\left[\alpha_1(G),\ldots,\alpha_p(G)\right]^T&=0.5T\left[|p_dv_1(R)|,\ldots,|p_dv_p(R)|\right]\\&=0.5T(p_{max}-p_{min})\left[|\boldsymbol{b_1}|,\ldots,|\boldsymbol{b_p}|\right],\end{aligned} \quad \text{(A16)}$$

References

1. Jao, C.K.; Wang, C.Y.; Yeh, T.Y.; Tsai, C.C.; Lo, L.C.; Chen, J.H.; Pao, W.C.; Sheen, W. WiSE: A System-Level Simulator for 5G Mobile Networks. *IEEE Wirel. Commun.* **2018**, *25*, 4–7. [CrossRef]
2. Lee, J.; Han, M.; Rim, M.; Kang, C.G. 5G K-SimSys for Open/Modular/Flexible System-level Simulation: Overview and Its Application to Evaluation of 5G Massive MIMO. *IEEE Access* **2021**, *1*, 99. [CrossRef]
3. Pratschner, S.; Müller, M.K.; Ademaj, F.; Nabavi, A.; Tahir, B.; Schwarz, S.; Rupp, M. Verification of the Vienna 5G link and system level simulators and their interaction. In Proceedings of the 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 11–14 January 2019; pp. 1–8.
4. Series, M. Guidelines for evaluation of radio interface technologies for IMT-2020. Report ITU, 2017: 2412-0. Available online: https://www.itu.int/pub/R-REP-M.2412-2017 (accessed on 28 July 2022).
5. OPNET Technologies. Available online: https://opnetprojects.com/ (accessed on 28 July 2022).
6. Puttonen, J.; Kurjenniemi, J. Satellite Network Simulator 3 (SNS3). In Proceedings of the Workshop on Simulation for European Space Programmes, Noordwijk, The Netherlands, 24 March 2015; Volume 24, p. 26.
7. Yang, L.Y.; Han, S.S.; Wang, X.; Li, Y.K.; Wang, F.Y. Computational Experiment Platforms for Networks: The State of the Art and Prospect. *Zidonghua Xuebao/Acta Autom. Sin.* **2019**, *45*, 1637–1654.
8. Li, J.; Xue, K.; Wei, D.S.L.; Liu, J.; Zhang, Y. Energy Efficiency and Traffic Offloading Optimization in Integrated Satellite/Terrestrial Radio Access Networks. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 2367–2381. [CrossRef]
9. Systems Tool Kit (STK)—AGI. Available online: https://www.agi.com/products/stk/ (accessed on 28 July 2022).
10. General Mission Analysis Tool (GMAT)—NASA—GSFC Open Source Software. Available online: https://opensource.gsfc.nasa.gov/projects/GMAT/ (accessed on 28 July 2022).
11. Available online: https://ww2.mathworks.cn/help/satcom/ref/satellitescenario.html (accessed on 28 July 2022).
12. Qi, J.; Li, Z.; Liu, G. Research on coverage and link of multi-layer Satellite Network based on STK. In Proceedings of the 2015 10th International Conference on Communications and Networking in China (ChinaCom), Shanghai, China, 15–17 August 2015; pp. 410–415. [CrossRef]
13. Zhang, S.; Chen, Z.; Sun, W.; Xiao, X.; Ke, Y. Study On The Connection Rate Of LEO Communication Satellite. In Proceedings of the 2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C), Hainan, China, 6–10 December 2021; pp. 610–614. [CrossRef]

14. Huanhuan, L.; Tao, S.; Hengchi, L.; Yongxin, F. Design and Implementation of Software Simulation for Air-Ground Data Link Communication. In Proceedings of the 2015 8th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 12–13 December 2015; pp. 379–382. [CrossRef]
15. Afonso, F.; Jerome, G.; Paolo, P. Topological Design, Routing and Hand-Over in Satellite Networks. In *Handbook of Wireless Networks and Mobile Computing*; John Wiley and Sons, Ltd.: Hoboken, NJ, USA, 2002; pp. 473–507.
16. Bhuyan, B.; Sarma, H.D.; Sarma, N. A Survey on Middleware for Wireless Sensor Networks. *Eurasip J. Wirel. Commun. Netw.* **2014**, *4*, 7–17.
17. Al-Roubaiey, A.A.; Sheltami, T.R.; Mahmoud, A.S.H.; Salah, K. Reliable Middleware for Wireless Sensor-Actuator Networks. *IEEE Access* **2019**, *7*, 14099–14111. [CrossRef]
18. Guessoum, Z. Adaptive agents and multiagent systems. *IEEE Distrib. Syst. Online* **2004**, *5*, 1–4. [CrossRef]
19. Lewis, F.L.; Cui, B.; Ma, T.; Song, Y.; Zhao, C. Heterogeneous Multi-Agent Systems: Reduced-Order Synchronization and Geometry. *IEEE Trans. Autom. Control* **2015**, *61*, 1391–1396. [CrossRef]
20. Kumari, G.P.; Kandan, B.; Mishra, A.K. Experience Sharing on SOA Based Heterogeneous Systems Integration. In Proceedings of the 2008 IEEE Congress on Services—Part I, Los Alamitos, CA, USA, 6–11 July 2008; pp. 107–108. [CrossRef]
21. Yang, Y. Application of SOA and Web Service in Implementing Heterogeneous System Integration. In Proceedings of the 2013 International Conference on Information Science and Cloud Computing Companion, Guangzhou, China, 7–8 December 2013; pp. 1–5. [CrossRef]
22. Hou, T.; Jia, Y.; Wu, Y.; Li, Y.; Han, J. Diagnosing Performance Issues in Microservices with Heterogeneous Data Source. In Proceedings of the 2021 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), New York City, NY, USA, 30 September 2021–3 October 2021; pp. 493–500. [CrossRef]
23. Afanasev, M.Y.; Fedosov, Y.V.; Krylova, A.A.; Shorokhov, S.A. An application of microservices architecture pattern to create a modular computer numerical control system. In Proceedings of the 2017 20th Conference of Open Innovations Association (FRUCT), St. Petersburg, Russia, 3–7 April 2017; pp. 10–19. [CrossRef]
24. Yan, J.; Xu, H.; Li, L. Modeling and applications for temporal point processes. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 3227–3228.
25. Becchi, M.; Byna, S.; Cadambi, S.; Chakradhar, S. Data-Aware Scheduling of Legacy Kernels on Heterogeneous Platforms with Distributed Memory. In *Proceedings of the 22nd Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'10), Santorini, Thira, Greece, 13–15 June 2010*; Heide, F.M., Phillips, C.A., Eds.; ACM: New York, NY, USA; pp. 82–91.
26. Becchi, M.; Byna, S.; Cadambi, S.; Chakradhar, S. Adaptive Partitioning for Irregular Applications on Heterogeneous CPU-GPU Chips. *Procedia Comput. Sci.* **2015**, *51*, 140–149.
27. Wang, Z.; Zheng, L.; Chen, Q.; Guo, M. CPU+GPU scheduling with asymptotic profiling. *Parallel Comput.* **2014**, *40*, 107–115. [CrossRef]
28. Shen, J.; Varbanescu, A.L.; Lu, Y.; Zou, P.; Sips, H. Workload Partitioning for Accelerating Applications on Heterogeneous Platforms. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 2766–2780. [CrossRef]
29. Huh, E.-N.; Welch, L.R.; Shirazi, B.A.; Cavanaugh, C.D. Heterogeneous resource management for dynamic real-time systems. In Proceedings of the 9th Heterogeneous Computing Workshop (HCW 2000) (Cat. No.PR00556), Cancun, Mexico, 1 May 2000; pp. 287–296. [CrossRef]
30. Feng, M.; Guomin, L.; Wenrong, G. Heterogeneous Network Resource Allocation Optimization Based on Improved Bat Algorithm. In Proceedings of the 2018 International Conference on Sensor Networks and Signal Processing (SNSP), Xi'an, China, 28–31 October 2018; pp. 55–59. [CrossRef]
31. Alhusaini, H.; Prasanna, V.K.; Raghavendra, C.S. A framework for mapping with resource co-allocation in heterogeneous computing systems. In Proceedings of the 9th Heterogeneous Computing Workshop (HCW 2000) (Cat. No.PR00556), Cancun, Mexico, 1 May 2000; pp. 273–286. [CrossRef]
32. Kim, J.-K.; Siegel, H.J.; Maciejewski, A.A.; Eigenmann, R. Dynamic Resource Management in Energy Constrained Heterogeneous Computing Systems Using Voltage Scaling. *IEEE Trans. Parallel Distrib. Syst.* **2008**, *19*, 1445–1457. [CrossRef]
33. Tian, J.; Liu, Q.; Zhang, H.; Wu, D. Multiagent Deep-Reinforcement-Learning-Based Resource Allocation for Heterogeneous QoS Guarantees for Vehicular Networks. *IEEE Internet Things J.* **2021**, *9*, 1683–1695. [CrossRef]
34. Al Qerm, I.; Pan, J. DeepEdge: A New QoE-Based Resource Allocation Framework Using Deep Reinforcement Learning for Future Heterogeneous Edge-IoT Applications. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 3942–3954. [CrossRef]
35. RIida, R.F.; Trindade, P.H.A.; Faria, B.; Aguayo, L.; Wyglinski, A.M. SatSysSim: A Novel Event-Driven Simulation Framework for DVB/RCS2 Performance Characterization. *IEEE Access* **2021**, *10*, 308–318. [CrossRef]
36. Zhang, X.; Wang, J.; Jiang, C.; Li, H.; Guo, Q.; Wang, X.; Ren, Y. Hardware-in-the-loop simulation system for space information networks. *J. Commun. Inf. Netw.* **2017**, *2*, 131–141. [CrossRef]
37. Connors, P.; Ryu, B.; Dao, S. Modeling and simulation of broad-band satellite networks. I. medium access control for QoS provisioning. *IEEE Commun. Mag.* **2019**, *37*, 72–79. [CrossRef]
38. Al-Hraishawi, H.; Lagunas, E.; Chatzinotas, S. Traffic simulator formultibeam satellite communication systems. In Proceedings of the 2020 10th Advanced Satellite Multimedia Systems Conference and the 16th Signal Processing for Space Communications Workshop (ASMS/SPSC), Frosinone, Italy, 8–10 September 2020; pp. 1–8.

39. Wang, Y.; Xu, J.; Jiang, L. Challenges of System-Level Simulations and Performance Evaluation for 5G Wireless Networks. *IEEE Access* **2017**, *2*, 1553–1561. [CrossRef]
40. Shao, S.; Hailes, P.; Wang, T.Y.; Wu, J.Y.; Maunder, R.G.; Al-Hashimi, B.M.; Hanzo, L. Survey of turbo, LDPC, and polar decoder ASIC implementations. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2309–2333. [CrossRef]
41. Fang, Y.; Bu, Y.; Chen, P.; Lau, F.C.; Al Otaibi, S. Irregular-mapped protograph LDPC-coded modulation: A bandwidth-efficient solution for 6G-enabled mobile networks. *IEEE Trans. Intell. Transp. Syst.* **2021**. [CrossRef]
42. Bioglio, V.; Condo, C.; Land, I. Design of polar codes in 5G new radio. *IEEE Commun. Surv. Tutor.* **2020**, *23*, 29–40. [CrossRef]
43. Raghav, S.; Ruggiero, M.; Marongiu, A.; Pinto, C.; Atienza, D.; Benini, L. GPU Acceleration for Simulating Massively Parallel Many-Core Platforms. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 1336–1349. [CrossRef]
44. Nikdel, Z.; Bing, G.; Neville, S.W. DockerSim: Full-stack simulation of container-based Software-as-a-Service (SaaS) cloud deployments and environments. In Proceedings of the 2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, Canada, 21–23 August 2017; pp. 1–6.
45. Steinbach, M.; Jindal, A.; Chadha, M.; Gerndt, M.; Benedict, S. TppFaaS: Modeling Serverless Functions Invocations via Temporal Point Processes. *IEEE Access* **2022**, *10*, 9059–9084. [CrossRef]
46. Aytekin, A.; Johansson, M. Exploiting Serverless Runtimes for Large-Scale Optimization. In Proceedings of the 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), Milan, Italy, 8–13 July 2019; pp. 499–501. [CrossRef]
47. Ramachandran, L.; Narendra, N.C.; Ponnalagu, K. Dynamic provisioning in multi-tenant service clouds. *Serv. Oriented Comput. Appl.* **2012**, *6*, 283–302. [CrossRef]
48. Zhou, C.; Gong, C.; Hui, H.; Lin, F.; Zeng, G. A task-resource joint management model with intelligent control for mission-aware dispersed computing. *China Commun.* **2021**, *18*, 214–232. [CrossRef]
49. Al-Azzoni, I.; Iqbal, S. Meta-Heuristics for Solving the Software Component Allocation Problem. *IEEE Access* **2020**, *8*, 153067–153076. [CrossRef]
50. Pohlmann, U.; Hüwe, M. Model-driven allocation engineering:Specifying and solving constraints based on the example of automo-tive systems. *Autom. Softw. Eng.* **2019**, *26*, 315–378. [CrossRef]
51. Aleti, S.; Bjornander, L.; Grunske; Meedeniya, I. ArcheOpterix: An extendable tool for architecture optimization of AADL models. In Proceedings of the ICSE Workshop Model-Based Methodologies Pervasive Embedded Softw, Los Alamitos, CA, USA, 16 May 2009; pp. 61–71.
52. OSATE (Open Source AADL Tool Environment). Available online: https://osate.org (accessed on 1 July 2022).
53. Aleti, L.; Grunske, I.; Meedeniya, I.; Moser, I. Let the ants deploy your software—An ACO based deployment optimisation strategy. In Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering, Auckland, New Zealand, 16–20 November 2009; pp. 505–509.
54. Guan, X.; Wan, X.; Choi, B.-Y.; Song, S.; Zhu, J. Application Oriented Dynamic Resource Allocation for Data Centers Using Docker Containers. *IEEE Commun. Lett.* **2016**, *21*, 504–507. [CrossRef]
55. Monsalve, A.; Landwehr, A.; Taufer, M. Dynamic CPU Resource Allocation in Containerized Cloud Environments. In Proceedings of the 2015 IEEE International Conference on Cluster Computing, Shenzhen, China, 4–7 May 2015; pp. 535–536.
56. Kaewkasi, C.; Chuenmuneewong, K. Improvement of Container Scheduling for Docker Using Ant Colony Optimization. In Proceedings of the 2017 9th international conference on knowledge and smart technology (KST), Chonburi, Thailand, 1–4 February 2017; pp. 254–259.
57. Li, Y.; Zhang, J.; Jiang, C.; Wan, J.; Ren, Z. PINE: Optimizing Performance Isolation in Container Environments. *IEEE Access* **2019**, *7*, 30410–30422. [CrossRef]
58. Waibel, P.; Hochreiner, C.; Schulte, S.; Koschmider, A.; Mendling, J. ViePEP-C: A Container-Based Elastic Process Platform. *IEEE Trans. Cloud Comput.* **2019**, *9*, 1657–1674. [CrossRef]
59. Jena, T.; Mohanty, J.R. GA-Based Customer-Conscious Resource Allocation and Task Scheduling in Multi-cloud Computing. *Arab. J. Sci. Eng.* **2017**, *43*, 4115–4130. [CrossRef]
60. Tao, Y.; Feng, L.; Zhang, L.; Liao, T.W. CLPS-GA: A library and pareto solution-based hybrid genetic algorithm forenergy-aware cloud service scheduling. *Appl. Soft Comput.* **2014**, *19*, 264–279. [CrossRef]
61. Sun, G.; Yu, H.; Anand, V.; Li, L. A cost efficient framework and algorithm for embedding dynamic virtual network requests. *Future Gener. Comput. Syst.* **2013**, *29*, 1265–1277. [CrossRef]
62. Cheng, X.; Su, S.; Zhang, Z.; Shuang, K.; Yang, F.; Luo, Y.; Wang, J. Virtual network embedding through topology awareness and optimization. *Comput. Netw.* **2012**, *56*, 1797–1813. [CrossRef]
63. Lu, J.; Turner, J. *Efficient Mapping of Virtual Networks onto a Shared Substrate*; Washington University in St Louis: St. Louis, MI, USA, 2006. [CrossRef]
64. Szeto, W.; Iraqi, Y.; Boutaba, R. A multi-commodity flow based approach to virtual network resource allocation. In Proceedings of the GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489), San Francisco, CA, USA, 1–5 December 2003; Volume 6, pp. 3004–3008.
65. Zhu, Y.; Ammar M, H. Algorithms for assigning substrate network resources to virtual network components. In Proceedings of the 25TH IEEE International Conference on Computer Communications. Barcelona, Spain, 23–29 April 2006; pp. 1–12.

66. Liu, W.; Xiang, Y.; Ma, S.; Tang, X. Completing Virtual Network Embedding All in One Mathematical Programming. In Proceedings of the 2011 International Conference on Electronics, Communications and Control (ICECC), Ningbo, China, 9–11 September 2011; pp. 183–185.
67. Schaffrath, G.; Werle, C.; Papadimitriou, P.; Feldmann, A.; Bless, R.; Greenhalgh, A. Network Virtualization Architecture: Proposal and Initial Prototype. In Proceedings of the 1st ACM workshop on Virtualized Infrastructure Systems and Architectures, New York, NY, USA, 17 August 2009; pp. 63–72.
68. Houidi, I.; Louati, W.; Ameur, W.B.; Zeghlache, D. Virtual network provisioning across multiple substrate networks. *Comput. Netw.* **2011**, *55*, 1011–1023. [CrossRef]
69. Wei, X.; Li, H.; Yang, K.; Zou, L. Topology-aware partial virtual cluster mapping algorithm on shared distributed infrastructures. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *25*, 2721–2730. [CrossRef]
70. Pei, J.; Hong, P.; Xue, K.; Li, D. Efficiently embedding service function chains with dynamic virtual network function placement in geo-distributed cloud system. *IEEE Trans. Parallel Distrib. Syst.* **2018**, *30*, 2179–2192. [CrossRef]
71. Zhang, Q.; Liang, X.; Zhu, Y.; Wang, X.; Palacharla, P.; Sarin, V.; Ikeuchi, T. Vertex-centric distributed computation for mapping virtual networks across domains. In Proceedings of the NOMS 2018–2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018; pp. 1–8.
72. Alhazmi, K.; Sharkh, M.A.; Shami, A. Drawing the cloud map: Virtual network provisioning in distributed cloud computing data centers. *IEEE Syst. J.* **2016**, *12*, 1480–1491. [CrossRef]
73. Zong, Y.; Ou, Y.; Hammad, A.; Kondepu, K.; Nejabati, R.; Simeonidou, D. Location-aware energy efficient virtual network embedding in software-defined optical data center networks. *J. Opt. Commun. Netw.* **2018**, *10*, B58–B70. [CrossRef]
74. Dahir, M.H.; Alizadeh, H.; Gözüpek, D. Energy efficient virtual network embedding for federated software-defined networks. *Int. J. Commun. Syst.* **2019**, *32*, e3912. [CrossRef]
75. Chen, X.; Li, C.; Jiang, Y. Optimization model and algorithm for energy efficient virtual node embedding. *IEEE Commun. Lett.* **2015**, *19*, 1327–1330. [CrossRef]
76. Zhang, Z.; Cao, H.; Su, S.; Li, W. Energy aware virtual network migration. *IEEE Trans. Cloud Comput.* **2020**. [CrossRef]
77. Chen, M.; Sun, Y.; Hu, H.; Tang, L.; Fan, B. Energy-saving and resource-efficient algorithm for virtual network function placement with network scaling. *IEEE Trans. Green Commun. Netw.* **2020**, *5*, 29–40. [CrossRef]
78. Zhang, X.; Xu, Z.; Fan, L.; Yu, S.; Qu, Y. Near-optimal energy-efficient algorithm for virtual network function placement. *IEEE Trans. Cloud Comput.* **2019**, *10*, 553–567. [CrossRef]
79. Ayoub, O.; Bovio, A.; Musumeci, F.; Tornatore, M. Survivable virtual network mapping with fiber tree establishment in filterless optical networks. *IEEE Trans. Netw. Serv. Manag.* **2021**, *19*, 37–48. [CrossRef]
80. Liu, Y.; Han, P.; Hou, J.; Zheng, J. Resource-Efficiently survivable IoT services provisioning via virtual network embedding in fiber-wireless access network. *IEEE Access* **2019**, *7*, 65007–65018. [CrossRef]
81. Li, X.; Gao, T.; Zhang, L.; Tang, Y.; Zhang, Y.; Huang, S. Survivable K-node (edge) content connected virtual optical network (KC-VON) embedding over elastic optical data center networks. *IEEE Access* **2018**, *6*, 38780–38793. [CrossRef]
82. Cao, H.; Zhao, H.; Luo, D.X.; Kumar, N.; Yang, L. Dynamic virtual resource allocation mechanism for survivable services in emerging NFV-enabled vehicular networks. *IEEE Trans. Intell. Transp. Syst.* **2021**. [CrossRef]
83. Chen, Y.; Zhang, G.; Hu, D.; Tao, Q. Multiscale emulation technology based on the integration of virtualization, physical and simulation networks. In Proceedings of the 2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC), Hangzhou, China, 23–25 June 2019; pp. 396–402.
84. Ricci, R.; Alfeld, C.; Lepreau, J. A solver for the network testbed mapping problem. *ACM SIGCOMM Comput. Commun. Rev.* **2003**, *33*, 65–81. [CrossRef]
85. Sheng, Z.; Qian, Z.; Song, G.; Lu, S. FELL: A flexible virtual network embedding algorithm with guaranteed load balancing. In Proceedings of the 2011 IEEE international conference on communications (ICC), Kyoto, Japan, 5–9 June 2011; pp. 1–5.
86. Boyang, L.; Muqing, W.; Haosen, Z. Virtual Network Embedding Based on Hybrid Adaptive Genetic Algorithm. In Proceedings of the 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, 6–9 December 2019; pp. 1197–1202.
87. Cheng, X.; Su, S.; Zhang, Z.; Wang, H.; Yang, F.; Luo, Y.; Wang, J. Virtual network embedding through topology-aware node ranking. *ACM SIGCOMM Comput. Commun. Rev.* **2011**, *41*, 38–47. [CrossRef]
88. Zhang, P.; Yao, H.; Li, M.; Liu, Y. Virtual network embedding based on modified genetic algorithm. *Peer—Peer Netw. Appl.* **2019**, *12*, 481–492. [CrossRef]
89. Zhang, P.; Yao, H.; Li, M.; Liu, Y. Multi-objective enhanced particle swarm optimization in virtual network embedding. *EURASIP J. Wirel. Commun. Netw.* **2016**, *2016*, 1–9. [CrossRef]
90. Zhang, S. Reliable virtual network mapping algorithm with network characteristics and associations. *IEEE Access* **2021**, *9*, 48121–48130. [CrossRef]
91. Cao, H.; Zhu, H.; Yang, L. Collaborative attributes and resources for single-stage virtual network mapping in network virtualization. *J. Commun. Netw.* **2019**, *22*, 61–71. [CrossRef]