Task Scheduling in Geo-Distributed Computing: A Survey

Yujian Wu, Shanjiang Tang, Ce Yu, Bin Yang, Chao Sun, Jian Xiao, Hutong Wu, and Jinghua Feng

(Survey Paper)

***Abstract*—Geo-distributed computing, a paradigm that assigns computational tasks to globally distributed nodes, has emerged as a promising approach in cloud computing, edge computing, cloud-edge computing, and supercomputer computing (SC). It enables low-latency services, ensures data locality, and handles large-scale applications. As global computing capacity and task demands increase rapidly, scheduling tasks for efficient execution in geo-distributed computing systems has become an increasingly critical research challenge. It arises from the inherent characteristics of geographic distribution, including heterogeneous network conditions, region-specific resource pricing, and varying computational capabilities across locations. Researchers have developed diverse task scheduling methods tailored to geo-distributed scenarios, aiming to achieve objectives such as performance enhancement, fairness assurance, and fault-tolerance improvement. This survey provides a comprehensive and systematic review of task scheduling techniques across four major distributed computing environments, with an in-depth analysis of these approaches based on their core scheduling objectives. Through our analysis, we identify key research challenges and outline promising directions for advancing task scheduling in geo-distributed computing.**

***Index Terms*—Geo-distributed, task scheduling, workflow scheduling, optimization.**

I. Introduction

In RECENT years, driven by the increasing distributed processing capacities and application requirements, geo-distributed computing has emerged as a new paradigm in diverse computing environments. From large-scale social networks processing billions of daily interactions to privacy-preserving federated learning systems and latency-sensitive Internet of Things (IoT) applications, modern systems inherently require computation and data processing across geographical locations. Frequently, the relevant data for these computational tasks and the computing nodes they occupy are geographically distributed.

Received 15 February 2025; revised 10 June 2025; accepted 10 July 2025. Date of publication 21 July 2025; date of current version 18 August 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFF1204101. Recommended for acceptance by A. C. Zhou. *(Corresponding author: Shanjiang Tang.)*

Yujian Wu, Shanjiang Tang, Ce Yu, Bin Yang, Chao Sun, Jian Xiao, and Hutong Wu are with the College of Intelligence and Computing, Tianjin University, Tianjin 300072, China (e-mail: wyuj@tju.edu.cn; tashj@tju.edu.cn; yuce@tju.edu.cn; yangbincic@tju.edu.cn; sch@tju.edu.cn; xiaojian@tju.edu.cn; wht@tju.edu.cn).

Jinghua Feng is with the National Supercomputing Center of Tianjin, Tianjin 300456, China (e-mail: fengjh@nscc-tj.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPDS.2025.3591010, provided by the authors.

Digital Object Identifier 10.1109/TPDS.2025.3591010

TABLE I
A COMPARISON OF DIFFERENT GEO-DISTRIBUTED COMPUTING ENVIRONMENTS

Feature	GDCC	EC	CEC	GDSC
Respond Latency	High Latency	Moderate Latency	Low Latency	Extremely Low
Workload Size	Virtually Unlimited	Relatively Small	Moderate Workload	Exascale, Heavy
Performance	High, Scalable	Low, Limited	Moderate	Exceptional, Scalable
Bandwidth	High Demand	Low Demand	Reduced Demand	Very High Demand
Task Type	General-Purpose (E.g., Web Services)	Real-Time, Latency-Sensitive	Latency-Sensitive & Compute-Intensive	Scientific Computing, Large-Scale

* GDCC: Geo-Distributed Cloud Computing.
* EC: Edge Computing. CEC: Cloud-Edge Computing.
* GDSC: Geo-Distributed Supercomputer Computing.

Geo-distributed computing distributes tasks across multiple locations to enable global scalability, leverage computational capacity and provide geographical redundancy for enhanced reliability. The paradigm also minimizes user-perceived latency by processing data closer to its source, and inherently supports regional data locality requirements that many modern applications demand. However, these characteristics also introduce unique challenges in resource management and data transfer that traditional centralized scheduling systems do not encounter. Moreover, each geo-distributed computing scenario has its unique characteristics and challenges. Fully utilizing computation capacities requires more customized solutions for each environment to ensure efficient task execution.

Different geo-distributed computing environments demand distinct scheduling strategies to balance among latency, workload, and network bandwidth. Table I illustrates these differences to better understand the unique features and requirements in each computing infrastructure. Researchers have developed numerous scheduling algorithms tailored for these geo-distributed computing systems, aiming to reduce overall makespan, minimize data transfer costs, or ensure fairness, fault-tolerance in scheduling. These approaches incorporate a range of techniques, including heuristic methods, mathematical models, and AI-based models, to enhance the execution performance of geo-distributed systems.

Despite these advancements, task scheduling in geo-distributed environments remain an active and challenging area

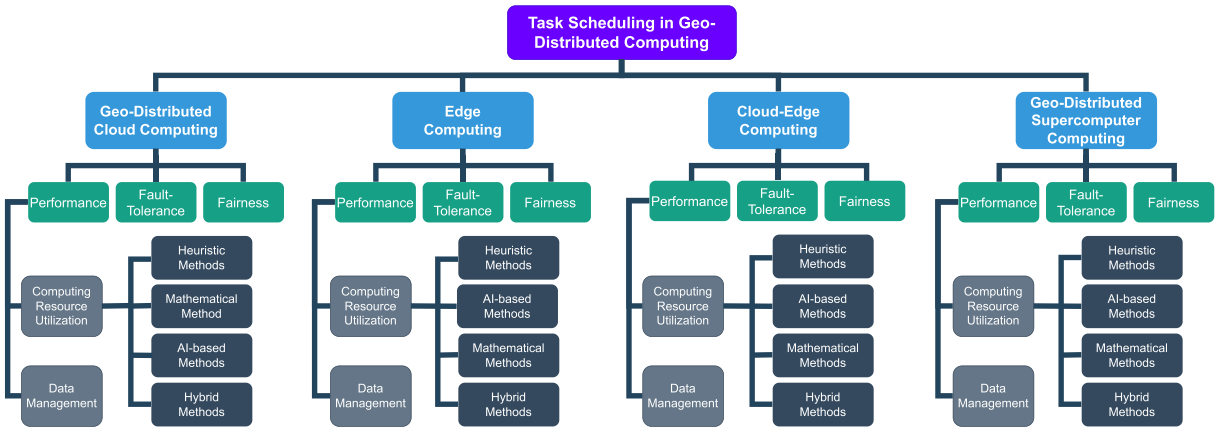

Fig. 1. An overview of geo-distributed task scheduling. We categorize scheduling strategies across Geo-Distributed Cloud Computing, Cloud-Edge Computing, Edge Computing, and Geo-Distributed Supercomputer Computing. In each scheduling infrastructure, we focus on objectives including performance, fault tolerance, and fairness, with scheduling methods including heuristic, AI-based, mathematical, and hybrid techniques.
Due to page limitations, the relevant content of the latter two scenarios is provided in Appendix A and B of the supplementary materials.

of research. Ongoing efforts focus on developing more efficient scheduling strategies, integrating emerging computing environments such as IoT, cloud-edge, and supporting new application paradigms like microservices. Furthermore, improving the usability and manageability of these systems is crucial, as it impacts the broader adoption and effectiveness of computing solutions in real-world applications. Still, with the emergence of new hardware, task scheduling in heterogeneous computing environments presents new opportunities and challenges in maximizing hardware utilization to enhance performance efficiency and system generality.

Many recent task scheduling surveys in cloud or edge computing have classified and compared scheduling strategies by algorithm types (e.g., heuristics, meta-heuristics or hybrid schemes) [1], [2], [3], [4], by centralized or distributed methods [5] or by application, technique, and metrics [6]. However, these studies are limited to a specific scheduling environment. Although [7], [8], [9] summarize scheduling methods across two or more distributed environments (e.g., cloud and grid environment), none comprehensively covers research on all types of geo-distributed computing environments, especially scheduling in super computer (grid) environment. This paper aims to fill this gap by summarizing the diverse geo-distributed scheduling strategies across four specific computing environments: geo-distributed cloud, cloud-edge, edge, and geo-distributed supercomputer computing. We include HPC environment as it focuses on extreme performance optimization and large-scale resource utilization, fundamentally differing from other geo-distributed computing paradigms.

The main contribution of this paper is twofold. First, we investigate the latest research advancements in geo-distributed task scheduling, classifying relevant works according to their scheduling environments. Second, we dive into each environment and classify the works based on three goals: performance, fault-tolerance, and fairness. Performance ensures efficient resource utilization and minimizes costs, fault-tolerance guarantees system reliability in failure-prone distributed systems, and fairness focuses on equitable resource allocation in multi-tenant settings. In the performance section, we further explore methods targeting computing resource utilization, such as heuristic, AI-based, mathematical, and hybrid approaches. Hybrid methods, such as AI combined with heuristic techniques, leverage the strengths of multiple paradigms. While computing resource utilization emphasizes efficient use of hardware, data transfer efficiency aims at storage and network optimizations to minimize latency and enhance I/O performance.

Fig. 1 illustrates the organization of the remainder of this survey. Section II discusses task scheduling techniques in the geo-distributed cloud computing environment. Section III covers scheduling techniques in the edge environment. For strategies in the cloud-edge computing and geo-distributed supercomputer computing, we put them in Appendix A and B of the supplementary materials. Section IV discusses the opportunities and challenges of task scheduling in geo-distributed computing. Finally, we conclude this survey in Section V.

II. Geo-Distributed Cloud Computing

Geo-distributed cloud computing (GDCC) operates across data centers (DCs) situated in diverse geographical locations, characterized by preemptible resources in a multi-tenant environment, infrastructure heterogeneity across regions and elasticity in resource scaling. Google's Borg system [10], for example, orchestrates applications and resources across its geo-distributed DCs. This architecture presents multiple challenges, including inter-DC network latency and bandwidth constraints, heterogeneous computing capacities and workloads, and regional regulatory compliance requirements. In particular, the electricity prices of each DC vary across time and location. This spatio-temporal diversity is demonstrated in Fig. 3. Effective task schedulers are expected to not only reduce electricity costs by allocating computing tasks appropriately, but also take into account resource utilization, performance, service level agreements (SLAs), and other operational expenses. Fig. 2 summarizes scheduling

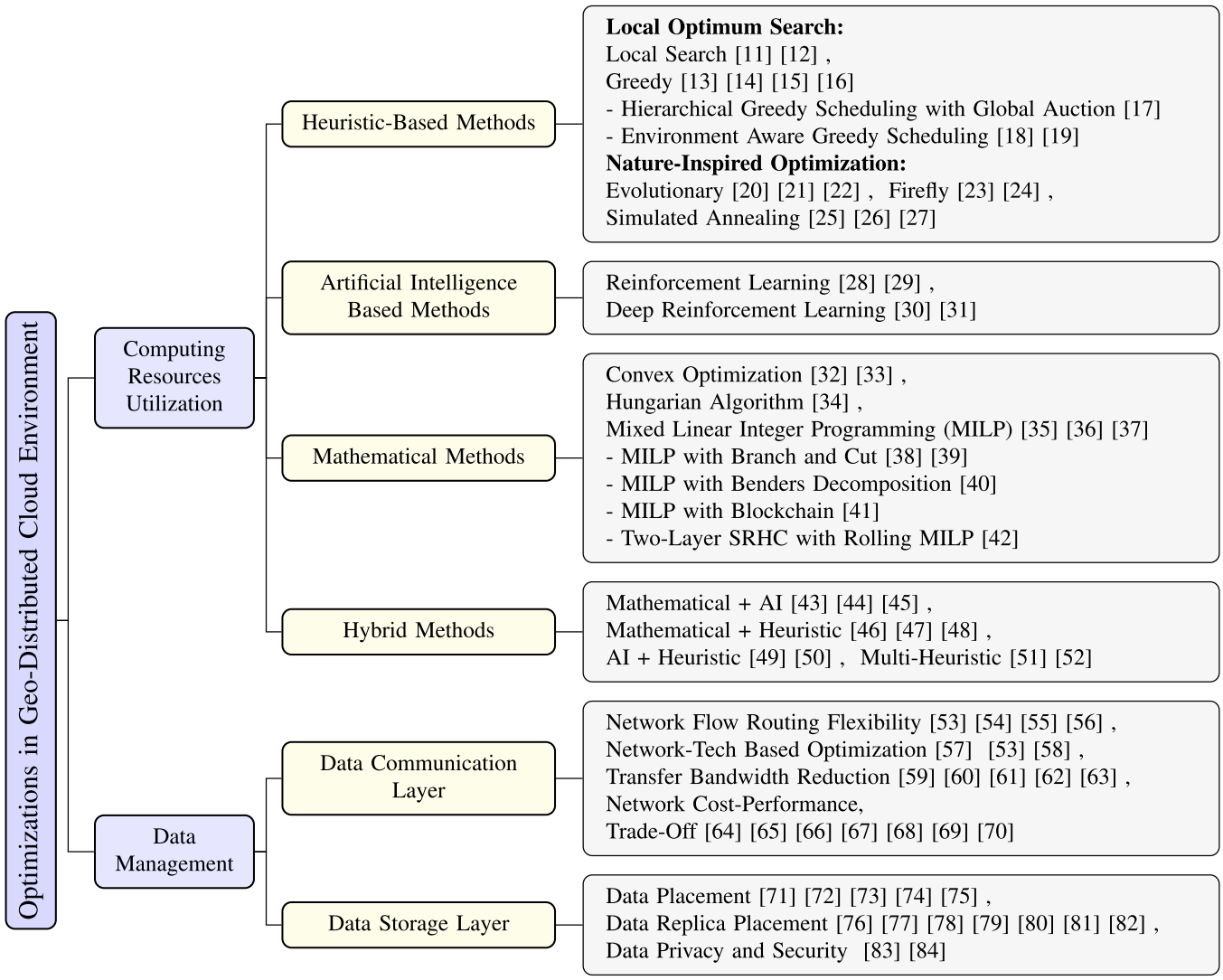

Fig. 2. Taxonomy of studies on optimizations under geo-distributed cloud computing infrastructure.

strategies addressing these geo-specific challenges in GDCC systems.

A. Performance

1) Computing Resources Utilization:

a) Heuristic-based methods: i) Local optimum: Local optimum refers to a solution to an optimization problem where, within a neighboring set of candidate solutions, no better solution exists. Unlike the global optimum, a local optimum may not be the best possible solution overall, but it is the best in its immediate vicinity.

Local Search Algorithms: Considering the spatio-temporal diversity of the electricity pricing to reduce energy cost, DEWS [11] employs Variable Neighborhood Descent (VND) to swap in-layer tasks and select geo-distributed DCs through three neighborhood structures, including task sequence swapping, DC selection, and VM selection. The approach further integrates Dynamic Voltage and Frequency Scaling (DVFS)-based energy optimization, enabling frequency adjustment of VMs and efficient utilization of task slack time. ECWSD [12], also designed for variable electricity environments, adopts a two-phase generate-and-optimize paradigm. An initial task sequence is produced via the Heterogeneous Earliest Finish Time (HEFT) algorithm, followed by refinement using an Adaptive Local Search (ALS) strategy, where the neighborhood scope is gradually reduced by limiting the number of swapped immediate successor task pairs in each iteration.

Greedy Algorithms: Due to the heterogeneity of computing nodes and tasks, the execution speeds of different tasks in a job are different. A straggler task with a slow execution speed will affect the execution progress of the job. To deal with straggling tasks, Li et al. [13] propose a two-phase speculative execution strategy that selects nodes with the strongest processing capability to create task replicas. First, it evaluates cluster load to identify straggler-affected jobs; then, it greedily chooses nodes with the highest computing power, storage capacity and memory resources to execute task replicas. This ensures the replicas can

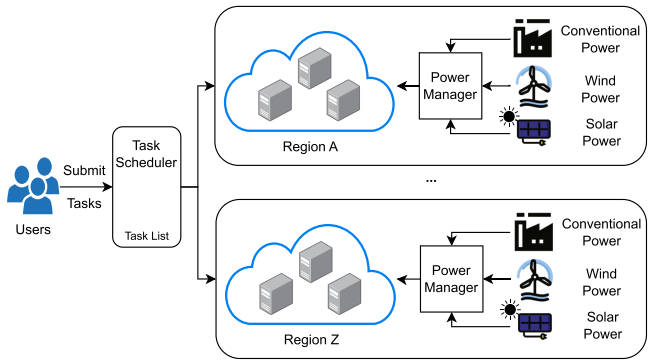

Fig. 3. An example of a geo-distributed computing architecture exploiting spatial-temporal diversity. Every geo-distributed region has its own power supply and power price. After tasks are submitted, the scheduler will assign tasks to or offload tasks from certain computing regions considering each region's power supply diversity.

be processed as quickly as possible to mitigate the impact of stragglers. Similarly, Real-Time Scheduling Algorithm using Task Duplication (RTSATD) [14] focuses on Big Data processing workflows by selecting tasks with minimum earliest start time (MESTF) while duplicating precursor tasks to the same instance in geo-distributed clouds. While task duplication strategies reduce workflow makespan, they also introduce computational overhead and potentially higher monetary costs, prompting alternative approaches that focus on directly identifying and managing straggler nodes rather than replicating tasks. Li et al. [15] first use statistical analysis of historical performance metrics, including authority category, urgency and length to detect them. Then they map tasks to resources through a priority-based, time-cost trade-off calculation for optimal resource utilization. This prevents task assignment to these identified straggling nodes while redistributing their existing tasks to normal nodes with available capacity. Beyond task management, efficient data access is also critical for job execution. Wang et al. [16] propose a three-phase dynamic scheduling framework that prioritizes data locality by scheduling tasks to the nearest available servers (rack-local, cluster-local, or remote).

Hierarchical Greedy Scheduling With Global Auction: To achieve global optimization in hyperscale environments, scheduling strategies evolve beyond localized greedy choices to incorporate hierarchical frameworks and auction-like competitive mechanisms. MAST [17] introduces a three-level hierarchical scheduler that decouples global queue management, regional resource allocation, and cluster-level orchestration. Jobs are placed via a distributed auction where regional ML schedulers compete using placement quality scores derived from resource availability and preemption costs, enabling exhaustive cross-region evaluation for final decisions.

Environment Aware Greedy Scheduling: Using renewable energy in task execution not only reduces energy costs, but is also environmentally friendly. However, renewable energy supply is often unstable and varying constantly. Padhi et al. [18] develop four scheduling algorithms based on uncertainty level (UNL) of renewable energy to optimize energy allocation using variable renewable and non-renewable energy sources. UNL categorizes uncertainty from low to high for users and from 1% to 100% for cloud providers, forming the basis for the following algorithms: *UNL-FABEF* reduces operational costs by optimizing energy usage predictions; *UNL-HAREF* maximizes renewable energy utilization and minimizes carbon emissions; *UNL-RR* evenly distributes tasks among DCs in a cyclical manner; and *UNL-MOSA* is a hybrid approach that dynamically adapts to changes in energy availability for efficient resource utilization and cost-effectiveness. By considering Computer Room Air Condition (CRAC) operations, Ali et al. [19] propose spatio-thermal-aware workload management algorithms that always select the lowest-cost DC from a sorted list based on cooling efficiency and electricity prices, while considering temperature variations (inside/outside DCs). These approaches use a zone-based (cool, warm, hot) allocation scheme to greedily select servers with minimum cooling requirements, reducing both cooling costs and service level agreement (SLA) violations in geo-distributed environments.

ii) Nature-inspired algorithms: These algorithms mimic natural processes to explore large search spaces and escape local optima for solving complex optimization problems. Distinct from the single-solution design of greedy methods, nature-inspired algorithms leverage a population-based structure to generate Pareto fronts for multi-objective trade-offs.

Evolutionary Algorithms: Taking advantage of spatial variations, Yuan et al. [20] propose an improved multi-objective evolutionary algorithm based on decomposition (IMEAD) decomposing the revenue-energy cost optimization problem into multiple sub-problems. Then it evolves solutions through genetic operators to determine optimal task splitting ratios and service rates under renewable energy constraints. Khalid et al. [21] also center on energy costs and formulate this as a constrained bi-objective optimization problem and leverage the Strength Pareto Evolutionary Algorithm (SPEA-II) to iteratively determine Pareto-optimal solutions for request dispatch and resource allocation, considering both computing and cooling costs under smart grid dynamics. Shifting the focus to multi-workflow scheduling in federated clouds, Wu et al. [22] propose a data locality-aware scheduling mechanism that first pre-processes tasks sharing the same datasets to reduce data transfer volume, then uses a customized Evolutionary Multi-objective Optimization (EMO) with an intensification strategy to minimize both makespan and rental costs while meeting deadline constraints.

Firefly Algorithms: Firefly algorithms are optimization techniques where solutions "attract" better ones, mimicking the behavior of fireflies. Handling geo-distributed large data with resource and cost optimization is a key challenge. Multivariate Metaphor based Metaheuristic Glowworm Swarm Map-Reduce Optimization (MM-MGSMO) [23] uses virtual machines (glowworms) and updates their positions based on multiple objective functions including bandwidth, storage, energy and computation costs, followed by MapReduce-based allocation to optimize resource utilization and workload distribution. Focusing on delay constraints and renewable energy utilization, Ammari et al. [24] address application scheduling in geo-distributed green DCs through a modified Firefly Algorithm (mFA) that dynamically adjusts attractiveness and introduces adaptive randomization

parameter with damping to maximize renewable energy usage across locations.

Simulated Annealing (SA) Algorithms: SA algorithms mimic the metal cooling process to gradually refine solutions. Yuan et al. [25] propose SA-based adaptive differential evolution (SADE) to balance task response time and energy cost in distributed DCs, which integrates Metropolis criterion and adaptive mutation with entropy-based crowding distance for better convergence. For green cloud DCs, Yuan et al. [26] develop SA-based biobjective differential evolution (SBDE) that optimizes both revenue and energy consumption by considering spatial variations in renewable power generation and electricity pricing. Targeting QoS in cloud environments, Yuan et al. [27] present an adaptive bi-objective differential evolution (ASBD) that minimizes both energy cost and task loss probability through genetic operations and adaptive elite archive updates. While sharing SA as their core optimization strategy, these methods differ in how they integrate SA with other techniques: SADE combines SA with differential evolution, SBDE incorporates SA into biobjective optimization, and ASBD adapts SA for elite archive-based evolution.

b) AI-based methods: AI-based methods stand out for their ability to autonomously adapt to dynamic and uncertain environments by learning optimal decision policies directly from environmental feedback. But they often require extensive training and well-designed reward structures to perform effectively.

Reinforcement Learning (RL): Graph partitioning, an important problem in graph analytics, involves analyzing large datasets spread across geo-distributed DCs. RLCut [28] is an adaptive graph partitioning method employing multi-agent learning to optimize hybrid-cut model decisions, considering both network bandwidth heterogeneity among DCs and graph dynamicity to balance partitioning effectiveness and overhead.

Scheduling AI-Generated Content (AIGC) workloads in the global cloud system needs to consider special characteristics of ML training, such as gang scheduling, locality of GPUs, intensive and exclusive GPU usage. Zhang et al.'s [29] algorithm leverages the advantages of multi-agent reinforcement learning (MARL) and Soft Actor Critic (SAC) algorithms to optimize GPU utilization while minimizing operational costs and carbon emissions. MARL eliminates the single point of failure in the central scheduling system and is scalable when the network grows, while SAC balances policy exploitation with action exploration optimally and has the advantage of addressing complex reward structures such as delayed rewards.

Deep Reinforcement Learning (DRL): Due to the uncertainty and complexity of energy availability and task arrival in green DCs, traditional heuristic algorithms encounter difficulties in geo-distributed task scheduling and resource allocation. Bi et al. [30] introduce an Improved Deep Q-learning Network (IDQN) that enables an agent to learn from a reward function and continuously select optimal green DCs and servers to maximize the reward, resulting in lower task rejection rates and energy costs. The IDQN agent perceives the real-time status of servers (CPU and memory resources) and individual task requirements within DCs at each decision step. Focusing on the same problem but within a broader hybrid cloud context, Zhao et al. [31] propose a Proximal Policy Optimization (PPO) based DRL approach, which automatically applies workload shifting and cloud-bursting in a hybrid multi-cloud environment consisting of multiple private and public clouds to maximize renewable energy utilization and avoid deadline constraint violations. Their agent observes a continuous state space that includes the time (affecting solar energy), predicted renewable energy availability, DC power consumption, incoming job attributes (changing workloads), server load across private DCs (resource availability), and recent public costs. Based on these, it dynamically decides whether to delay tasks to private servers, or utilize public cloud resources.

c) Mathematical methods: Mathematical methods offer interpretable solutions under strict constraints, but often struggle with scalability and adaptability in dynamic environments.

Convex Optimization: This is one of the mathematical approaches where the objective function is convex, meaning any local minimum is also a global minimum, ensuring efficient problem-solving. Kiani and Ansari [32] propose a profit-maximizing workload distribution strategy which decomposes workloads into green and brown components served by renewable and traditional energy sources respectively, optimizing both workload allocation and service rates while accounting for SLAs and electricity price diversity across regions. The strategy leverages a G/D/1 queuing model to capture workload distribution and proves the convexity of the optimization problem. Yuan et al. [33] later approach the profit maximization problem by formulating a geography-aware convex optimization that directly integrates ISP-specific task routing and a broader set of spatial cost and revenue variations, allowing efficient solution via interior point methods.

Hungarian Algorithm: Li et al. [34] propose a MapReduce scheduling framework optimizing both map and reduce phases: first the Hungarian algorithm assigns map tasks to idle containers by minimizing a cost matrix based on data locality levels, thus optimizing for reduced data transmission. Then For reduce tasks, it optimally assigns tasks to containers by minimizing the maximum predicted execution time based on a cost matrix of task completion times.

Mixed Linear Integer Programming (MILP): MILP models problems using linear equations while allowing discrete decision variables, enabling it to handle combinatorial complexity and ensure feasible solutions in scheduling tasks. Wang et al. [35] combine electrical and thermal system optimization in DC microgrids, which integrates scheduling with waste heat recovery, repurposing it for residential heating demands. By addressing the stochastic nature of renewable energy supply, delay-tolerant workloads, and thermal demand, their formulation minimizes total costs while ensuring system security, service quality and energy efficiency. Wang et al. [36] in contrast, concentrate on electric energy optimization without thermal integration, and formulate a Mixed Integer Nonlinear Programming (MINLP) model incorporating QoS constraints via an M/G/1 queuing network. They transform it into a tractable form and propose a strategy powered by both renewable and conventional energy, incorporating dynamic voltage and frequency scaling. Recently, Hao et al. [37] jointly consider computational workload scheduling, carbon emission, microgrid operation and characteristics of Uninterruptible Power Supply (UPS). Their method utilizes

the degree of freedom in computational workload scheduling to limit the nonlinear growth of UPS power losses and introduces carbon tax as a parameter in the optimization objective.

As the scale and complexity of DC scheduling problems grow, researchers have turned to specialized MILP solution techniques for specific operational challenges.

MILP With Branch and Cut: CASPER [38] is a carbon-aware scheduling and provisioning system for distributed web services. It formulates a multi-objective optimization problem utilizing spatial-temporal variability in energy sources and solves it using PuLP library, an interface to the Coin-or branch and cut (CBC) solver, to align computational workloads with available green energy across different regions.

MILP With Branch and Bound: To optimize power consumption based on demand response signals, OPRS [39] models the problem as a MILP, minimizing total operating costs across task delay scheduling, hybrid cooling, and UPS utilization. It employs a branch-and-bound algorithm which iteratively solves linear relaxations of the MILP. If a solution yields non-integer values for variables requiring integrality, the algorithm branches by creating two new subproblems that add constraints to drive these variables towards integer values.

MILP With Benders Decomposition: Integrating multiple DCs and the inter-DC network into a refined emission model and balancing emission reduction benefits against migration costs, Yang et al. [40] propose a large-scale MILP problem based on a spatio-temporal task migration mechanism and solve it using Benders decomposition algorithm which decouples task migration decisions and optical routing schemes across distributed DCs for carbon emission optimization.

Blockchain-Enabled Distributed MILP: Sajid et al. [41] design a decentralized energy-optimization system where DCs coordinate through a custom blockchain structure that enables direct workload migration based on real-time energy costs. Each DC employs MILP with conditional constraints to optimize across multiple energy sources (renewable/grid/battery/diesel) while using proof-of-work consensus to validate cost-based scheduling decisions. It replaces traditional front-end schedulers by enabling DCs to autonomously migrate workloads through blockchain-verified transactions when local energy costs exceed neighboring centers.

Two-Layer SRHC With Rolling MILP: DCs often consume lots of electricity and thus can be used to balance the power market. Recently, Cao et al. [42] develop a two-layer Stochastic Receding Horizon Control (SRHC) optimization framework for managing DC clusters as non-wire alternatives: the upper layer optimizes market bidding through stochastic programming while the lower layer executes spatial-temporal workload scheduling through MILP. This framework recursively solves finite-horizon optimization problems to handle uncertainties in regulating prices and workload delays, enabling DCs to participate in power market balancing.

d) Hybrid methods: Hybrid methods combine distinct approaches. The core strength is the ability to exploit complementary advantages of different paradigms, enabling them to tackle scheduling problems with improved flexibility, scalability, and solution quality. For example,

Mathematical + AI: Qin et al. [43] leverage Lyapunov optimization to transform time-coupled carbon emission constraints into a queue stability problem for geographical load balancing, and then employs both Generalized Benders Decomposition (GBD) and Deep Q-Network (DQN) to optimize joint energy consumption across servers and network traffic.

Nash equilibrium-based Intelligent Load Distribution (NILD) [44] combines non-cooperative game theory with Reinforcement Learning for workload management. This approach simultaneously minimizes DC operational costs and response latency across geographical locations. However, NILD does not consistently achieve global optima solutions. Game-Theoretic Deep Reinforcement Learning (GT-DRL) [45] advances carbon-aware scheduling by integrating location-specific renewable energy patterns into workload distribution across DCs. By synthesizing non-cooperative game theory with DRL, GT-DRL dynamically optimizes both carbon emissions and operational costs for AI inference workloads, adapting to real-time variations in electricity pricing and data transfer costs across geographical locations.

Mathematical + Heuristic: Hosseinalipour et al. [46] tackle energy optimization through a scale-adaptive framework for graph-structured tasks. They combine convex programming for small-scale networks with cloud crawler-based sub-graph extraction for large-scale geo-distributed environments, while employing online learning mechanisms to adapt to dynamic pricing scenarios. For distributed workflow scheduling, Li et al. [47] advance the efficiency of cloud workflows with a hypergraph partitioning based scheduling strategy, which incorporates the cloud's state and utilizes the Dijkstra algorithm with a Fibonacci heap. It significantly reduces both average task execution time and overall energy consumption, contributing to more balanced and sustainable cloud operations.

While the above work focuses on structural optimization, Yuan et al. [48] leverage spatial-temporal diversity in geo-distributed DCs and propose a spatial-temporal task scheduling. By formulating energy cost minimization as a nonlinear constrained optimization problem, it combines genetic algorithms with simulated annealing and particle swarm optimization to achieve optimal task scheduling while considering geographical variations in grid and renewable energy pricing.

AI + Heuristic: Turbo [49], a geo-distributed analytics system, leverages LASSO and GBRT to predict query execution time and intermediate output sizes in real-time. These predictions are subsequently utilized to guide its heuristic greedy policies (e.g., SCTF, MDRF) to dynamically reorder join operations at runtime. For geo-distributed, container-based cloud environments, Geo-aware Multi-Agent Task Allocation (GMTA) [50] employs a multi-agent and auction-based negotiation mechanism to optimize the scheduling of scientific workflows. GMTA initially partitions workflows heuristically to enhance parallelism. During the negotiation process, its agents apply a heuristic geo-aware cost model to dynamically determine whether critical tasks should be locally replicated or their data remotely accessed, thereby balancing computational and communication overheads.

Multi-Heuristic: Considering the diversity of electricity prices in distributed green clouds, profit-sensitive spatial scheduling (PS3) [51] maximizes profit using a Genetic-Simulated-Annealing-based Particle Swarm Optimization (GSPSO). PSO particles (solutions) are guided by GA-enhanced velocity updates and SA's Metropolis criterion then directly governs the acceptance of each particle's new position (scheduling decision). Under the same conditions, the Simulated Annealing-based Bees Algorithm (SBA) [52] minimizes energy cost for fine-grained scheduling by simulating bee foraging to find solutions. Differing from PS3, SBA integrates SA as an "SA-based selection" mechanism: SA's Metropolis criterion filters solutions by determining if new candidate solutions, generated by bees during their neighborhood search, are accepted to update a bee's current solution. It optimizes workload distribution, server speeds, and active server counts under strict response times.

2) Data Management:

a) Data communication layer: Efficiently and cost-effectively accessing the required data for geographically distributed computing tasks is essential, especially when the required data is distributed across various locations with limited cross-domain transfer bandwidth.

Network Flow Routing Flexibility: Due to the vast differences in network topology and bandwidth among DCs, a flexible routing approach becomes crucial in mitigating congestion and enhancing network utilization. Intelligent network routing strategies ensure balanced utilization of links between DCs, facilitating efficient and equitable distribution of data transfer loads, thus speeding up task execution (see Fig. 4).

Network flows will be generated to transfer the intermediate data between consecutive stages for further processing. These flows are collectively defined as a *coflow* of the data analytics job. Li et al. [53] propose a linear programming method to split and route data flows to multiple network paths and dynamically adjust sending rates to optimize bandwidth utilization across DCs. They treat the group of flows in a coflow that have the same pair of source and destination DCs as the basic unit in their multi-path routing model. For Map-Reduce jobs, in the shuffle phase, the entire set of network flows generated from map tasks to reduce tasks is referred to as a *coflow*. Li et al. [54] introduce Smart Coflow, which integrates endpoint flexibility into coflow scheduling, allowing for dynamic adjustment of data flow destinations based on current network conditions and DC availability. HPS+ [55] uses an augmented hyper-graph model to represent task-data and data-DC dependencies. It applies hyper-graph partitioning to minimize WAN data transfers and a Routing and Bandwidth Allocation (RBA) algorithm to coordinate data transfers and computation, prioritizing tasks with longer computing stages to reduce transfer times. However, it does not consider the de-allocation of network bandwidth. In the future, timely resource allocation and de-allocation could enhance network utilization and reduce the makespan.

Network congestion management is another key dimension of network flow optimization. CONA (CONgestion-Aware) [56] employs matrix-based traffic allocation and link grading strategies to maximize profit in geo-distributed transfers. While

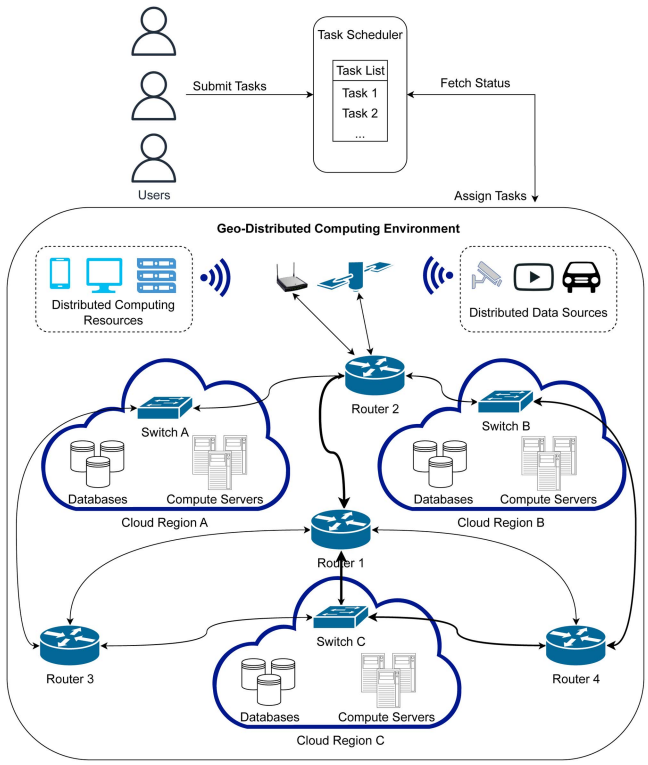

Fig. 4. An example of a geo-distributed computing environment focusing on distributed network architecture. After users submit tasks, the scheduler will assign tasks to different computing nodes according to computing and networking status. The thickness of the lines between routers and switches represents the relative size of the bandwidth. The length of the lines indicates the relative transmission distances.

CONA addresses general network congestion, modern distributed DL requires more sophisticated approaches.

Network Tech-based Optimization: VNF (Virtual Network Function) involves deploying network services as software instances rather than physical devices, allowing flexible network management. Gu et al. [57] address the deployment of VNFs and network flow scheduling in distributed DCs to minimize the total cost of Big Data processing while ensuring QoS.

SDN (Software Defined Network) is an architecture that allows for centralized control and dynamic management of network resources. Li et al. [53]'s design separates control and data planes by deploying proxy nodes in each DC and centralizing routing and scheduling decisions within a controller. It allows the controller to solve a LP problem to determine optimal multi-path traffic splitting percentages and sending rates for inter-DC flows, accelerating batch analytics jobs in communication stages. For geo-distributed stream data analytics, SAFA [58] focuses on SDN-driven, upfront worker node selection to minimize latency for continuous data streams. This P4-integrated framework facilitates efficient task allocation without modifying underlying stream processing systems. This contrasts with the above which uses SDN to optimize data transfer paths and rates for batch analytic jobs.

Network Bandwidth Optimization: MaxCompute [59] is a fast, fully managed, TB/PB-level data warehouse solution by

Alibaba. It provides users with a comprehensive data import solution and a variety of classic distributed computation models, which can efficiently process large-scale user data, reduce enterprise costs, and ensure data security. Huang et al. [60] propose *Yugong*, which works seamlessly with MaxCompute in very-large-scale production environments. By project migration, table replication and job outsourcing, the cross-DC bandwidth usage reduces significantly. The evaluation shows a 76% average total cross-DC WAN bandwidth reduction (hundreds of PBs daily), with specific replication optimizations consuming 41-45% less bandwidth than a baseline, and memory utilization imbalances between clusters halved.

Multi-level heterogeneities in network bandwidth and communication prices in geo-distributed DCs raise challenges to existing graph partitioning methods. To address it, Geo-Cut [61] first uses a cost-aware streaming heuristic to minimize inter-DC communication during edge assignment, followed by partition refinement to alleviate bottlenecks and optimize data transfer within budget constraints. For text analytics workloads, optimizing network bandwidth by reducing inherent data volume is also critical. POCLib [62] achieves 'near orthogonal processing on compression' for hierarchically compressed text by utilizing new indexing for word localization (e.g., word2rule, rule2location) combined with algorithms for local graph walks (partial traversals) and incremental data updates, enabling comprehensive direct processing (random access and traversal) on hierarchically compressed text.

Geo-distributed machine learning (Geo-DML) applications also face challenges with limited WAN bandwidth and data privacy laws, hindering efficient model training across dispersed DCs. RoWAN [63] (Routing and rate allocation in optical WAN) dynamically adjusts the network topology and allocates resources for each data flow. Additionally, they employ delayed SWRT (delayed Shortest Weighted Remaining Time) to prioritize and schedule multiple ML jobs effectively.

Network Transfer Cost and Performance Trade-Off: A crucial challenge in geo-distributed analytics (GDA) is efficiently managing the trade-off between cost and system performance. Xu et al. [64] address this challenge through a two-time scale approach that combines data placement optimization with query request admission control. It leverages Lyapunov optimization for online decision-making without requiring future traffic predictions. Another solution is *Kimchi* [65], a system that specifically targets the heterogeneity of monetary data transfer costs in multi-cloud environments. It employs mixed integer programming to determine cost-aware task placement for shuffle stages, allowing users to define their desired cost-performance preference. Delving into a more fundamental aspect of query processing, GDA-OPT [66] globally optimizes the query execution by combining join order and job location optimization through dynamic programming. Its cost model accounts for WAN costs, DC locations, and heterogeneous capabilities, while employing search space pruning techniques for large-scale GDA management. Specifically for MapReduce-based GDA workloads, *Cross-MapReduce* [67] optimizes inter-cluster data transfer by using a Global Reduction Graph (GRG) to analyze data dependencies, strategically determining the number and locations of global reducers before large-scale data movement. This contrasts with systems like Kimchi or GDA-OPT by directly minimizing data volume for MapReduce's aggregation phase via its GRG-based reducer placement.

Geo-Distributed ML (Geo-DML) also meets with this problem, Training Flow Adaptive Steering (TFAS) [68] is an online scheduling algorithm for Geo-DML jobs over dynamic and heterogeneous WANs. It utilizes a primal-dual framework within a linear programming model to optimize the allocation of network resources, expedite training completion and maximize ISP revenue. For geo-distributed DL training, unlike TFAS's network-wide resource allocation, HCEC (High-Convergence and Efficient-Communication) [69] optimizes a task's efficiency by adaptively managing its internal communication and computation scheduling. Adaptive Layerwise Communication (ALC) decides between N-layer or single-layer parameter synchronization based on real-time overhead analysis to minimize inter-DC communication delays. Experiments quantified a 32.9% communication optimization, reducing ResNet50/CIFAR-10 epoch processing time from around 28 seconds to around 19 seconds, alongside up to 37.9% overall training efficiency gains across models.

While the techniques above fulfill all user requests, these often lead to significant expenditure and high bandwidth consumption. Selectively accepting user requests can lead to higher profits and reduce bandwidth costs. For handling offline request submission, Yang et al. [70] propose *Metis* to alternately maximize service revenue under given bandwidth and minimize bandwidth cost under given requests. For online request submission, *OSA* is designed to maximize service profit by addressing the risk of incremental service costs exceeding revenue, making admission decisions in real-time.

b) Data storage layer: Proper storage of task-related data is crucial, as data is indispensable for computing jobs. However, the required data are often distributed across DCs, so well-designed placement strategies are crucial for efficient data storage and access.

Data Placement Optimization: The rapid growth of user data and the high cost of transferring large volumes across geo distributed DCs highlight the need for efficient data placement.

Refinement of Traditional Techniques: Li et al. [71] and Xie et al. [72] build new placement strategies upon classical methods (LP, Lagrangian relaxation) combined with heuristics (Floyd's, ant colony). Both algorithms outperform CRANE and Closet in terms of data transmission time.

Application of Graph Theory and Spectral Clustering: SpeCH (Spectral Clustering on Hypergraphs) [73] utilizes hypergraph spectral clustering to model multi-dimensional data associations. *SpectralApprox* improves efficiency with low-rank matrix approximations, while *SpectralDist* distributes computations across machines to handle large workloads.

Optimization of Critical Data Components: Parameter Servers (PS) store and update model parameters in Geo-DML, viewing model parameters as key distributed data. Li et al. [74] tackle this NP-hard problem using LP relaxation for fractional communication path solutions, then randomized rounding

for single path selection and bandwidth re-optimization, significantly cutting parameter transmission costs. Metadata also has a critical impact on the efficiency of workflow scheduling as it provides a global view of data location and enables task tracking during execution. Liu et al. [75] use relational DBMS, combined the hot metadata management strategies with three scheduling algorithms, OLB (Opportunistic Load Balancing), MCT (Minimum Completion Time) and DIM (Data-Intensive Multi-site task scheduling) to manage hot metadata.

Data Replica Placement Optimization: Data replica placement maintains data copies across DCs, which helps reduce access latency and improve overall efficiency. However, compared to the case without replication, replicas introduce additional complexity, making the placement more challenging.

Some approaches focus directly on improving the execution efficiency of individual tasks or overall job workflows by optimizing data locality or performing on-demand migration. For example, GEODIS [76] primarily focuses on minimizing the makespan, which employs LP and heuristics to balance data locality with data transfers. Some utilize the characteristics of data (access frequency, popularity) to manage replica placement. Li et al. [77] propose two distinct algorithms. DLO-migrate fetches data for non-node-locality tasks using idle network bandwidth, directly targeting task execution delays but risking migration overheads. DLO-predict periodically replicates predicted hot files to reduce data access delay. Chen et al. [78] utilize the golden division approach for Zipf-like replica distribution for MapReduce. They transform the challenge into a block-dependence tree construction problem and simplify it into a graph partitioning problem.

With the increasing scale and dynamism of systems, replica placement faces higher demands for scalability and adaptability and motivates the development of integrated frameworks that jointly optimize multiple objectives. Yu and Pan [79] propose a hypergraph-based data placement framework that models multiple metrics including data-node relationships and the associations of data groups (multi-item requests) without relaxation. Their multi-round scheme starts with greedy initial replica placement, then iteratively refines request routing based on replica locations and optimizes replica placement according to refined access patterns. Liu et al. [80] propose two schemes. The offline scheme determines the replica placement based on average read or write rates, offering scalability with linear computational complexity and a distributed implementation. However, its offline nature limits its immediate responsiveness. The online scheme complements this by adaptively handling bursty data requests without completely overriding the existing replica placement. This two-phase design (offline combined with online scheme) is intended to balance the stability of planning with responsiveness to dynamic changes.

AI methods can also be used to solve the placement problem. *GeoCol* [81] employs RL agents to study the placement decision to determine the existence and type of replicas at each DC, in order to adapt to changing access patterns and cost dynamics. GeoCol also splits data requests into sub-requests sent to different DCs, using Seasonal Auto-regressive Integrated Moving Average (SARIMA) to predict latency and determine the number and destination of sub-requests.

However, data replica may not always be the best choice. For data analysis tasks, Emara et al. [82] propose two strategies, one without replication and another with replication. They leverage the random sample partition data model to convert Big Data into sets of data blocks and distribute data blocks across DCs. Experimental results show that the strategy *without replication*, some data blocks are required to download from the remote DCs to a central DC for approximate analysis of the Big Data as a whole. The main advantage is its cost-effectiveness and reduced storage requirements, provided that initial data transfer latency is acceptable. In contrast, *with the replication* strategy, the data in each DC forms a random sample of the whole distributed data, as a sample of the data on each DC is enough to be representative of the whole distributed data. This method is better at supporting high-availability, low-latency localized analytics performance.

Data Security and Privacy: Data transfer and storage across geo-distributed DCs creates complex challenges, particularly in ensuring data transmission security and compliance with diverse regional regulations.

Nithyanantham et al. [83] introduce a hybrid DL framework which uses a DNN enhanced with Siamese training to safeguard against secondary data inference, effectively preserving user privacy during feature extraction and classification tasks. Zhou et al. [84] propose a privacy-centric process mapping where multi-level data privacy regulations dictate valid site placements by requiring equal or stricter data protection guarantees. Within these enforced privacy boundaries, it integrates the communication matrix for application processes with the varying network performance metrics of DCs, enabling optimized mapping of processes to nodes.

B. Fairness

Fairness-driven scheduling ensures resource allocation among tasks adheres to equity principles, prioritizing balanced outcomes over maximizing individual or system performance.

Chen et al. [85] apply *max-min fairness* to balance job completion times. They formulate the problem as a lexicographical minimization and leverage the totally unimodular property of its constraints to transform the original problem into an equivalent solvable LP problem, thereby ensuring fairness among competing jobs. While max-min fairness sometimes reduces system throughput, the evaluation in this work shows significant improvement in the worst job completion time compared to baselines like the default Spark scheduler.

In contrast to Chen et al.'s focus on job completion time fairness, Ebadifard et al. [86] address profit fairness among Cloud Service Providers (CSPs). Their approach involves a two-phase model prioritizing fairness before efficiency. It prioritizes CSPs with below-average profits in an "Essential Selling" (SE) status to enhance their participation and revenue, promoting fairer profit distribution. Next, a multi-objective fitness function evaluates these SE candidates, considering geo-distance (latency), file

dependency (migration costs and efficiency), and resource price. Compared to a price-only RT benchmark, this strategy improves execution time and maintains CSP profits without significant performance sacrifice.

C. Fault-Tolerance

Fault-tolerant scheduling leverages redundancy and predictive maintenance to improve reliability and resource efficiency, especially in dynamic environments. It involves two main aspects: fault recovery and proactive fault prevention.

Considering high network costs and latency, Xie et al. [72] design a predictive scheduler modeling transmission time and cost as key objectives. It uses a gray Markov model for resource prediction, and mitigates performance anomalies by factoring in potential error-induced overhead. Similarly, the two-level Approximate Dynamic Programming (ADP) [87] uses a virtualized monitoring model to predict server health, designed to avoid failures by scheduling tasks only on servers that are predicted to be healthy.

The two mechanisms above enable more robust task placement against execution anomalies, but they do not consider the recovery if faults occur. Li et al. [88]'s load-aware mechanism switches between proactive task cloning to guard against stragglers under light load, and reactive anomaly detection to recover from stragglers under heavy load. Then the Task Scheduling with Minimal Power and Execution Time (TSPT) algorithm places each task by optimizing for its deadline against the host's specific computing power and energy consumption to ensure an efficient and stable recovery.

III. Edge Computing

Edge computing is a geo-distributed computing paradigm that utilizes resources at the network edge to enable distributed computing near data sources (such as IoT devices and mobile devices). AWS Wavelength [89], for instance, extends AWS cloud capabilities to the edge of 5G networks, enabling ultra-low latency application deployment directly at the mobile edge. This distributed architecture effectively reduces communication latency, but it also introduces challenges: limited resources at edge nodes, unstable network connectivity, and high node heterogeneity. The scheduling process in an edge environment typically involves monitoring available resources, assessing workload requirements, and making real-time decisions to allocate tasks to the most suitable edge nodes. It allocates workloads efficiently across edge nodes while considering resource limitations. It also ensures low latency by keeping tasks close to data sources. The following examines various scheduling approaches, each addressing specific challenges under various edge computing scenarios (see Fig. 5).

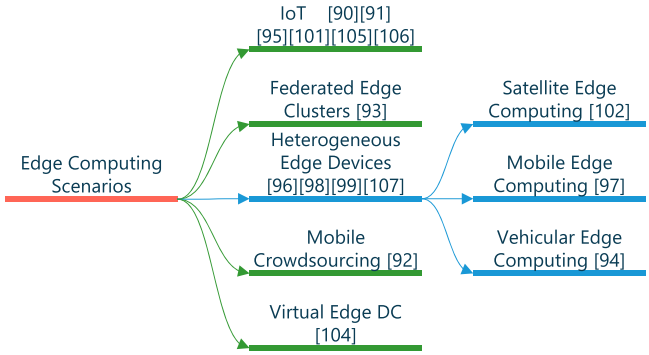

Fig. 5. An overview of specific edge computing scenarios.

A. Performance

1) Computing Resources Utilization: Many large-scale IoT applications need to analyze data distributed across sites to obtain final results. The problem is how to efficiently execute tasks among edge nodes and devices, considering the heterogeneity of resource capacities and prices across sites to ensure jobs finish before deadlines.

a) Heuristic-based methods: Gradient-based: Chen et al. [90] tackle the economic challenges of the commercial edge, where sites have heterogeneous resource capacities and prices. Their MCGL method uses a gradient adjustment strategy to navigate the trade-off between completion time and cost, finding an economical schedule that still meets a given deadline.

Distance-based: While the method above addresses the economic heterogeneity of commercial edge sites, managing hyperscale and dynamic topologies requires a different solution. Chatziliadis et al. [91] introduce NEMO, leveraging Euclidean embeddings of network topologies along with a set of heuristics to manage millions of nodes. This enables their distance-based calculations to efficiently manage millions of volatile nodes and adapt to changes in constant time.

Divide-and-Conquer: Mobile crowdsourcing leverages the collective efforts of individuals using mobile devices to gather data, complete tasks, and solve problems, often as part of IoT environments. An overview of such a system is shown in Fig. 6. Wang et al. [92] propose two approaches: Their local optimization algorithm first divides the complex task graph into distinct layers. It then conquers each layer's scheduling subproblem sequentially to assign the subtasks based on the workers' current spatial locations. This is complemented by a global method designed to co-optimize for both task completion time and the idle time of the mobile workers.

Affinity-based: Microservice is a software architecture style where a complex application is broken down into small and independently deployable services, each focusing on a specific function and communicating over the network. Many large-scale application development patterns are moving towards the agile microservice approach. Phare [93], based on affinity, prioritizes microservices with more stringent requirements and places them on the most convenient computing facilities.

b) AI-based methods: Deep Reinforcement Learning (DRL) Agent: Liu et al. [94] propose a multi-resource orchestration framework in vehicular edge computing (VEC) that combines multi-hop Vehicle-to-Vehicle (V2V) offloading

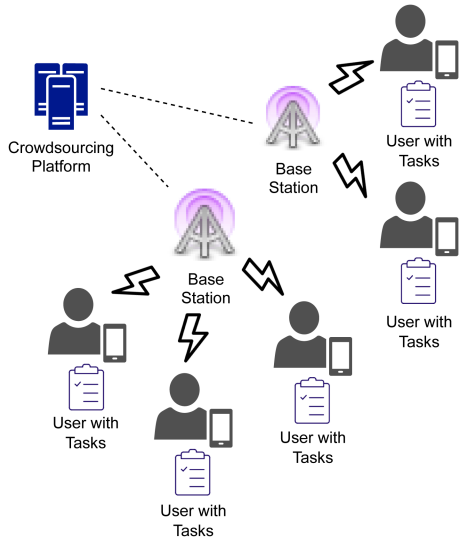

Fig. 6. An overview of the mobile crowd-sourcing system (MCS).

and service-migration-based Vehicle-to-Infrastructure (V2I) offloading. They employ an asynchronous advantage actor-critic (A3C) algorithm where multiple worker agents learn optimal scheduling policies through actor-critic networks. Focusing on macro and micro base stations, Ren et al.'s [95] agents are deployed within a hierarchical framework (BSCS and μBSCS), learning to make task offloading decisions that balance computing and caching resources across collaborative tiers to maximize long-term system utility. For serverless edge computing where rational, noncooperative edge nodes operate with only partial observability of the system, Tang et al. [96] model this as a Partially Observable Stochastic Game (POSG) and develop a multi-agent DRL algorithm based on Dueling Double Deep Recurrent Q-Network (D3RQN). It enables each agent to learn optimal scheduling and resource allocation strategies based on its local observations, including task generation state, data queue state, communication channel state, and previous computing resource allocation state in a competitive environment.

c) Mathematical methods: Mixed Integer Non-Linear Programming (MINLP): Li et al. [97] address task offloading in mobile edge computing with a focus on statistically guaranteed QoS to manage dynamic wireless conditions. The authors develop a statistical computation and transmission model as a MINLP with delay constraints and then leverage convex optimization and Gibbs sampling to balance task offloading and resource allocation.

Quadratic and Dynamic Programming: Targeting latency, resource consumption, and Quality of Results (QoR), Michailidou et al. [98] propose a three-objective task allocation in multi-query edge analytics. It models the edge's heterogeneous device capacities and variable network links, with adaptive techniques to revise allocations for new queries.

Modified Kuhn-Munkres: To leverage electric vehicles (EVs)' idle computing resources and store energy charged during off-peak hours, Liao et al. [99] propose an EV-assisted architecture incorporating a spatiotemporal workload offloading model that discretizes the optimization problem into smaller sub-problems. Then they deploy a modified Kuhn-Munkres algorithm for dynamic matching between EVs and service requests based on energy costs and QoS constraints.

d) Hybrid methods: Mathematical + Heuristic: Rossi et al. [100] use an Integer Linear Programming (ILP) formulation and a network-aware greedy heuristic for container-based application deployment. It selects the hosting VMs from a sorted list using a greedy approach. The list is sorted in ascending order, using the objective function as distance metric, the first VMs on this list minimize the adaption time.

2) Data Management: Network and storage layer scheduling algorithms enhance data management in distributed, heterogeneous edge systems by optimizing data flow and aggregation for efficient transfer, storage, and access.

a) Data communication layer: General Edge Network: Okita and *Okita∗* [101] are two online scheduling algorithms. *Okita* determines both worker and parameter server placement across edge sites to minimize network bandwidth usage, while *Okita∗* employs a non-preemptive fashion and optimizes this further by using dynamic programming to divide training data into time slots, making scheduling decisions based on data locality and wireless resource constraints.

Satellite Edge Computing (SEC) Network: Satellites, equipped with computing resources, have been envisioned as a key enabling technology to timely analyze stream data from IoT applications in remote regions on Earth. Streaming analytics with SEC enables real-time IoT data processing in remote areas. Xu et al. [102] formulate flow time minimization in SEC as an Integer Linear Programming (ILP) problem and propose offline and online algorithms using auxiliary graph construction and Lipschitz bandit techniques to handle satellite dynamics and data uncertainty.

Edge Compute First Networking (ECFN): ECFN integrates edge computing with networks to enable efficient data processing. Liu et al. [103] divide data processing into multiple parallel stages, where each stage optimizes cluster center selection and light-path provisioning to minimize job completion time. They further develop a routing and frequency slot reallocation scheme based on stage completion time to reduce bandwidth consumption during data transmission.

b) Data storage layer: Metadata Management: Dou et al. [104] create a virtual edge DC from pooled idle storage of edge servers. Its Intelligent Metadata Service (IMS) ensures low-latency access by partitioning file directories and migrating sub-trees to metadata servers closer to users, based on access patterns and latency.

B. Fairness

Fairness-focused scheduling methods address equitable resource distribution, ensuring that all users or tasks receive proportionate access to edge computing resources.

Dynamic Nash Bargaining Game: FairHealth [105], a 5G edge healthcare scheme that ensures long-term proportional

fairness in the Internet of Medical Things by addressing priority-aware and deadline-sensitive service characteristics. It employs a Lyapunov-based proportional-fairness resource scheduling algorithm that decomposes the long-term fairness problem into single-slot sub-problems, achieving a balance between service stability and fairness. This scheduling algorithm is complemented by a block-coordinate descent method for iteratively solving non-convex fair subproblems. Stemming from the Nash bargaining game formulation (P3), these per-slot fair subproblems are mixed-integer nonlinear programs (MINLP) with NP-hard complexity, addressed by the BCD method. It optimizes patient-MEC server associations and computation resource allocation rates to maximize a Nash product, which incorporates service priorities relative to local processing baselines and considers key resources like data size and compute capacities. Simulation results using a real-world telecom dataset validate this design, demonstrating a 74.44% improvement in the fairness index (Nash product) compared to classic global time-optimal schemes.

C. Fault-Tolerance

Fault-tolerant scheduling strategies are essential for ensuring reliable performance especially under conditions of dynamic workloads and potential failures of edge nodes.

Dynamic Model Partitioning: FTPipeHD [106] extends GPU-based pipeline parallelism to edge devices for fault-tolerant DNN training. It uses dynamic model partitioning to adapt to varying device capacities and a mixed weight replication strategy for quick recovery from device failures in distributed IoT environments.

Checkpoint With Replication: Xu et al. [107] introduce a hybrid approach for low-latency stream processing that combines checkpointing with active replication of high-risk operators to balance recovery speed and resource usage. By implementing this strategy alongside RL-based dynamic scaling, the framework ensures resilient stream processing, ensuring low latency processing of IoT data streams. Different from FTPipeHD which only considers proactive fault prevention, this approach combines both fault recovery with proactive prevention.

IV. Challenges and Open Issues

Aiming to assist researchers interested in geo-distributed computing and to promote deeper investigation into this domain, this section explores the research challenges, potential opportunities, and unresolved issues related to task scheduling in geo-distributed computing systems.

Security and Privacy: Geo-distributed tasks often involve handling massive amounts of data generated from multiple geo-distributed locations. Ensuring secure data transmission and compliant task execution has become a critical issue. These challenges are exacerbated in geo-distributed computing environments, particularly when managing sensitive data across multiple jurisdictions and heterogeneous edge devices. For example, deploying machine learning models across hospitals in different countries necessitates that schedulers manage not only computational resources but also heterogeneous data privacy regulations, such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). This challenge is particularly salient in federated learning applications within the healthcare domain [108]. The distributed nature of these systems introduces vulnerabilities at various levels, including data transmission between nodes and computation on untrusted edge devices. To address these privacy risks, industry solutions such as Google's Confidential Computing have been introduced. These approaches utilize Trusted Execution Environments (TEEs) to ensure data confidentiality even during processing on potentially untrusted edge nodes [109].

Existing security mechanisms are still limited in addressing these challenges due to the resource constraints of edge devices, the complexity of enforcing consistent security policies across diverse geographical regions with varying regulatory requirements, and the overhead of cryptographic operations in real-time applications. This necessitates the development of new security-aware scheduling algorithms that incorporate regional compliance requirements and ensure secure and efficient data handling during task execution.

Emerging Workload Diversity: Due to the increasing diversity in application types, geo-distributed computing faces growing scheduling challenges. Most existing scheduling approaches, while effective for traditional high-performance computing (HPC) and web service applications, struggle to handle emerging workload types, such as AI and multi-modal computational paradigms. These emerging workloads necessitate scheduling strategies tailored for geo-distributed computing environments and capable of exploiting the distributed computational capacities of geo-distributed infrastructures.

For instance, applications such as LLM inference and AR/VR-integrated intelligent assistants (e.g., ChatGPT's video-calling mode [110]) require coordination of multi-modal tasks, cross-region computational coordination, and the ability to leverage heterogeneous hardware such as CPUs, GPUs, and specialized accelerators. Similarly, time-sensitive AI applications, including conversational AI services (e.g., ChatGPT's voice-calling mode, 1-800-CHATGPT hotline) [111], require real-time response from servers. Both types of applications need strict adherence to Quality of Service (QoS) metrics, but often suffer from capacity limitations.

Currently, only a small number of LLM inference studies (e.g., [112]) have been conducted in geo-distributed environments, most optimization efforts remain confined to single-DC scenarios. The challenges are amplified in geo-distributed environments, where resource-demand imbalances, multi-stage processing pipelines, and network dynamics introduce additional complexity to workload scheduling. Existing geo-distributed scheduling approaches are often insufficient for optimally allocating LLM inferencing workload across geo-distributed regions. This is because LLM inference, being an auto-regressive process, has unique characteristics compared to traditional AI workloads. Specifically, the stateful nature of LLM inference relies on a KV Cache to expedite the process. Also, LLM

inferencing needs as low response latency as possible, presenting a distinct scheduling challenge.

Next Generation Geo-Distributed Computing: As computational hardware continues to advance, next-generation computing paradigms are poised to revolutionize computational capabilities, offering potential for solving complex problems.

Quantum computing leverages phenomena such as quantum superposition and entanglement to process massive computational tasks in parallel, demonstrating potential in addressing complex optimization and cryptographic problems that are difficult for classical computing systems to handle. This is relevant to the field of scheduling, as many scheduling problems are inherently combinatorial optimization challenges. For example, researchers have begun exploring the use of Quadratic Unconstrained Binary Optimization (QUBO) models to solve Flexible Job Shop Scheduling Problems (FJSP), employing quantum annealing devices such as Coherent Ising Machines (CIM) for computation [113]. Furthermore, quantum-inspired heuristic algorithms, drawing from quantum principles, have been shown to enhance scheduling efficiency on classical computers, outperforming traditional methods in terms of reducing waiting time and maximizing resource utilization [114]. These approaches can offer valuable insights for geo-distributed scheduling scenarios. By reformulating geo-distributed scheduling problems into QUBO-compatible structures, and applying quantum-inspired optimization frameworks, future systems may achieve faster convergence, better load balancing, and enhanced resilience to dynamic changes in task demands and network conditions.

Additionally, quantum communication, a critical aspect of quantum computing technology, leverages quantum entanglement and quantum key distribution. This mechanism enables ultra-secure data transmission, ensuring the data is transferred across geo-distributed computing systems securely [115]. However, integrating quantum processors into geo-distributed networks presents unique challenges. Due to the extreme sensitivity of quantum bits (qubits) to environmental noise, i.e., quantum decoherence, and the fact that the development of fault-tolerant quantum computers is still in its early stages, the current mainstream approaches [116], [117], [118] are hybrid models, in which classical computers control and coordinate geo-distributed, small-scale quantum processors.

V. Conclusion

Task scheduling in geo-distributed computing has attracted significant attention from both industry and academia due to its potential to leverage globally distributed computational resources and execute large-scale computational tasks. However, most existing surveys on task scheduling fail to differentiate between specific geo-distributed computing infrastructures. To address this gap, we present a comprehensive review of state-of-the-art task scheduling techniques across four distinct geo-distributed computing systems. We categorize scheduling algorithms based on different scheduling objectives (performance, fairness, fault-tolerance). Finally, we discuss the key challenges and open research issues in this field. We aim for this survey to serve as a valuable resource for researchers and practitioners, guiding continued exploration and innovation in this domain.

References

[1] M. Kumar, S. Sharma, A. Goel, and S. Singh, "A comprehensive survey for scheduling techniques in cloud computing," *J. Netw. Comput. Appl.*, vol. 143, pp. 1–33, Oct. 2019.

[2] P. Singh, M. Dutta, and N. Aggarwal, "A review of task scheduling based on meta-heuristics approach in cloud computing," *Knowl. Inf. Syst.*, vol. 52, no. 1, pp. 1–51, Jul. 2017.

[3] E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, "Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends," *Swarm Evol. Comput.*, vol. 62, Apr. 2021, Art. no. 100841.

[4] M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: A comprehensive analysis," *J. Netw. Comput. Appl.*, vol. 66, pp. 64–82, May 2016.

[5] A. Avan, A. Azim, and Q. H. Mahmoud, "A state-of-the-art review of task scheduling for edge computing: A delay-sensitive application perspective," *Electronics*, vol. 12, no. 12, pp. 2599, Jun. 2023.

[6] A. Arunarani, D. Manjula, and V. Sugumaran, "Task scheduling techniques in cloud computing: A literature survey," *Future Gener. Comput. Syst.*, vol. 91, pp. 407–415, Feb. 2019.

[7] Q. Luo, S. Hu, C. Li, G. Li, and W. Shi, "Resource scheduling in edge computing: A survey," *IEEE Commun. Surv. Tuts.*, vol. 23, no. 4, pp. 2131–2165, Fourth Quarter 2021.

[8] R. Ghafari, F. H. Kabutarkhani, and N. Mansouri, "Task scheduling algorithms for energy optimization in cloud environment: A comprehensive review," *Cluster Comput.*, vol. 25, no. 2, pp. 1035–1093, Apr. 2022.

[9] E. N. Alkhanak, S. P. Lee, R. Rezaei, and R. M. Parizi, "Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues," *J. Syst. Softw.*, vol. 113, pp. 1–26, Mar. 2016.

[10] A. Verma, L. Pedrosa, M. R. Korupolu, D. Oppenheimer, E. Tune, and J. Wilkes, "Large-scale cluster management at Google with Borg," in *Proc. Eur. Conf. Comput. Syst.*, Bordeaux, France, 2015, pp. 1–17.

[11] M. Hussain, L.-F. Wei, A. Rehman, F. Abbas, A. Hussain, and M. Ali, "Deadline-constrained energy-aware workflow scheduling in geographically distributed cloud data centers," *Future Gener. Comput. Syst.*, vol. 132, pp. 211–222, 2022.

[12] X. Li, W. Yu, R. Ruiz, and J. Zhu, "Energy-aware cloud workflow applications scheduling with geo-distributed data," *IEEE Trans. Serv. Comput.*, vol. 15, no. 2, pp. 891–903, Mar./Apr. 2022.

[13] C. Li, M. Song, Q. Zhang, and Y. Luo, "Cluster load based content distribution and speculative execution for geographically distributed cloud environment," *Comput. Netw.*, vol. 186, Feb. 2021, Art. no. 107807.

[14] H. Chen, J. Wen, W. Pedrycz, and G. Wu, "Big data processing workflows oriented real-time scheduling algorithm using task-duplication in geo-distributed clouds," *IEEE Trans. Big Data*, vol. 6, no. 1, pp. 131–144, Mar. 2020.

[15] C. Li, C. Zhang, B. Ma, and Y. Luo, "Efficient multi-attribute precedence-based task scheduling for edge computing in geo-distributed cloud environment," *Knowl. Inf. Syst.*, vol. 64, no. 1, pp. 175–205, Jan. 2022.

[16] J. Wang, X. Li, R. Ruiz, J. Yang, and D. Chu, "Energy utilization task scheduling for MapReduce in heterogeneous clusters," *IEEE Trans. Serv. Comput.*, vol. 15, no. 2, pp. 931–944, Mar. 2022.

[17] A. Choudhury et al., "MAST: Global scheduling of ML training across geo-distributed datacenters at hyperscale," in *Proc. 18th USENIX Symp. Operating Syst. Des. Implementation*, 2024, pp. 563–580. [Online]. Available: https://www.usenix.org/conference/osdi24/presentation/choudhury

[18] S. Padhi and R. B. V. Subramanyam, "Uncertainty level-based algorithms by managing renewable energy for geo-distributed datacenters," *Cluster Comput.*, vol, 27, pp. 5337—5354, Jan. 2024.

[19] A. Ali and Z. Özkasap, "Spatial and thermal aware methods for efficient workload management in distributed data centers," *Future Gener. Comput. Syst.*, vol. 153, pp. 360–374, Apr. 2024.

[20] H. Yuan, H. Liu, and J. Bi, "Revenue and energy cost-optimized biobjective task scheduling for green cloud data centers," *IEEE Trans. Automat. Sci. Eng.*, vol. 18, no. 2, pp. 817–830, Apr. 2021.

[21] S. Khalid and I. Ahmad, "Dual optimization of revenue and expense in geo-distributed data centers using smart grid," *IEEE Trans. Cloud Comput.*, vol. 11, no. 2, pp. 1622–1635, Second Quarter 2023.

[22] D. Wu, X. Wang, M. Huang, R. Zeng, and K. Yang, "Multi-objective optimization-based workflow scheduling for applications with data locality and deadline constraints in geo-distributed clouds," *Future Gener. Comput. Syst.*, vol. 157, pp. 485–498, Aug. 2024.

[23] S. Nithyanantham and G. Singaravel, "Resource and cost aware glow-worm MapReduce optimization based Big Data processing in geo distributed data center," *Wireless Pers. Commun.*, vol. 117, no. 4, pp. 2831–2852, Apr. 2021.

[24] A. C. Ammari, W. Labidi, F. Mnif, H. Yuan, M. Zhou, and M. Sarrab, "Firefly algorithm and learning-based geographical task scheduling for operational cost minimization in distributed green data centers," *Neurocomputing*, vol. 490, pp. 146–162, Jun. 2022.

[25] H. Yuan, J. Bi, J. Zhang, and M. Zhou, "Energy consumption and performance optimized task scheduling in distributed data centers," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 9, pp. 5506–5517, Sep. 2022.

[26] H. Yuan, J. Bi, and A. C. Ammari, "Biobjective task scheduling for distributed green data centers," *IEEE Trans. Automat. Sci. Eng.*, vol. 18, no. 2, pp. 731–742, Apr. 2021.

[27] H. Yuan, J. Bi, and M. Zhou, "Energy-efficient and QoS-Optimized adaptive task scheduling and management in clouds," *IEEE Trans. Automat. Sci. Eng.*, vol. 19, no. 2, pp. 1233–1244, Apr. 2022.

[28] A. C. Zhou, J. Luo, R. Qiu, H. Tan, B. He, and R. Mao, "Adaptive partitioning for large-scale graph analytics in geo-distributed data centers," in *Proc. IEEE 38th Int. Conf. Data Eng.*, Kuala Lumpur, Malaysia, 2022, pp. 2818–2830.

[29] S. Zhang, M. Xu, W. Y. Bryan Lim, and D. Niyato, "Sustainable AIGC workload scheduling of geo-distributed data centers: A multi-agent reinforcement learning approach," in *Proc. IEEE Glob. Commun. Conf.*. Kuala Lumpur, Malaysia, 2023, pp. 3500–3505.

[30] J. Bi, Z. Yu, and H. Yuan, "Cost-optimized task scheduling with improved deep Q-learning in green data centers," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*. Prague, Czech Republic, 2022, pp. 556–561.

[31] J. Zhao, M. A. Rodriguez, and R. Buyya, "A deep reinforcement learning approach to resource management in hybrid clouds harnessing renewable energy and task scheduling," in *Proc. IEEE 14th Int. Conf. Cloud Comput.*. Chicago, IL, USA, 2021, pp. 240–249.

[32] A. Kiani and N. Ansari, "Profit maximization for geographically dispersed green data centers," *IEEE Trans. Smart Grid*, vol. 9, no. 2, pp. 703–711, Mar. 2018.

[33] H. Yuan, J. Bi, and M. Zhou, "Geography-aware task scheduling for profit maximization in distributed green data centers," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1864–1874, Third Quarter 2022.

[34] X. Li, F. Chen, R. Ruiz, and J. Zhu, "MapReduce task scheduling in heterogeneous geo-distributed data centers," *IEEE Trans. Serv. Comput.*, vol. 15, no. 6, pp. 3317–3329, Nov./Dec. 2022.

[35] P. Wang, Y. Cao, Z. Ding, H. Tang, X. Wang, and M. Cheng, "Stochastic programming for cost optimization in geographically distributed internet data centers," *CSEE J. Power Energy Syst.*, vol. 8, no. 4, pp. 1215–1232, 2020.

[36] R. Wang, Y. Lu, K. Zhu, J. Hao, P. Wang, and Y. Cao, "An optimal task placement strategy in geo-distributed data centers involving renewable energy," *IEEE Access*, vol. 6, pp. 61948–61958, 2018.

[37] X. Hao, P. Liu, and Y. Deng, "Joint optimization of operational cost and carbon emission in multiple data center micro-grids," *Front. Energy Res.*, vol. 12, Feb. 2024, Art. no. 1344837.

[38] A. Souza et al., "CASPER: Carbon-aware scheduling and provisioning for distributed web services," Mar. 2024, *arXiv:2403.14792*.

[39] M. Zhao, X. Wang, and J. Mo, "Workload and energy management of geo-distributed datacenters considering demand response programs," *Sustain. Energy Technol. Assessments*, vol. 55, Feb. 2023, Art. no. 102851.

[40] T. Yang, H. Jiang, Y. Hou, and Y. Geng, "Carbon management of multi-datacenter based on spatio-temporal task migration," *IEEE Trans. Cloud Comput.*, vol. 11, no. 1, pp. 1078–1090, First Quarter 2023.

[41] S. Sajid et al., "Blockchain-based decentralized workload and energy management of geo-distributed data centers," *Sustain. Comput.: Inform. Syst.*, vol. 29, Mar. 2021, Art. no. 100461.

[42] Y. Cao, F. Cao, Y. Wang, J. Wang, L. Wu, and Z. Ding, "Managing data center cluster as non-wire alternative: A case in balancing market," *Appl. Energy*, vol. 360, Apr. 2024, Art. no. 122769.

[43] Y. Qin, W. Han, Y. Yang, and W. Yang, "Joint energy optimization on the server and network sides for geo-distributed data centers," *J. Supercomputing*, vol. 77, no. 7, pp. 7757–7790, Jul. 2021.

[44] N. Hogade, S. Pasricha, and H. J. Siegel, "Energy and network aware workload management for geographically distributed data centers," *IEEE Trans. Sustain. Comput.*, vol. 7, no. 2, pp. 400–413, Second Quarter 2022.

[45] N. Hogade and S. Pasricha, "Game-theoretic deep reinforcement learning to minimize carbon emissions and energy costs for AI inference workloads in geo-distributed data centers," 2024, *arXiv:2404.01459*.

[46] S. Hosseinalipour, A. Nayak, and H. Dai, "Power-aware allocation of graph jobs in geo-distributed cloud networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 4, pp. 749–765, Apr. 2020.

[47] C. Li, Y. Zhang, Z. Hao, and Y. Luo, "An effective scheduling strategy based on hypergraph partition in geographically distributed datacenters," *Comput. Netw.*, vol. 170, Apr. 2020, Art. no. 107096.

[48] H. Yuan and J. Bi, "Spatiotemporal task scheduling for heterogeneous delay-tolerant applications in distributed green data centers," *IEEE Trans. Automat. Sci. Eng.*, vol. 16, no. 4, pp. 1686–1697, Oct. 2019.

[49] H. Wang, D. Niu, and B. Li, "Turbo: Dynamic and decentralized global analytics via machine learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 6, pp. 1372–1386, Jun. 2020.

[50] M. Niu, B. Cheng, Y. Feng, and J. Chen, "GMTA: A geo-aware multi-agent task allocation approach for scientific workflows in container-based cloud," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 3, pp. 1568–1581, Sep. 2020.

[51] H. Yuan, J. Bi, and M. Zhou, "Profit-sensitive spatial scheduling of multi-application tasks in distributed green clouds," *IEEE Trans. Automat. Sci. Eng.*, vol. 17, no. 3, pp. 1097–1106, Jul. 2020.

[52] H. Yuan, M. Zhou, Q. Liu, and A. Abusorrah, "Fine-grained and arbitrary task scheduling for heterogeneous applications in distributed green clouds," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 5, pp. 1380–1393, Sep. 2020.

[53] L. Chen, S. Liu, and B. Li, "Optimizing network transfers for data analytic jobs across geo-distributed datacenters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 2, pp. 403–414, Feb. 2022.

[54] W. Li, X. Yuan, K. Li, H. Qi, X. Zhou, and R. Xu, "Endpoint-flexible coflow scheduling across geo-distributed datacenters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 10, pp. 2466–2481, Oct. 2020.

[55] L. Zhao, Y. Yang, A. Munir, A. X. Liu, Y. Li, and W. Qu, "Optimizing geo-distributed data analytics with coordinated task scheduling and routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 2, pp. 279–293, Feb. 2020.

[56] X. Tao, K. Ota, M. Dong, W. Borjigin, H. Qi, and K. Li, "Congestion-aware traffic allocation for geo-distributed data centers," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1675–1687, Third Quarter 2022.

[57] L. Gu, J. Hu, D. Zeng, S. Guo, and H. Jin, "Service function chain deployment and network flow scheduling in geo-distributed data centers," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2587–2597, Fourth Quarter 2020.

[58] H. Mostafaei and S. Afridi, "SDN-enabled resource provisioning framework for geo-distributed streaming analytics," *ACM Trans. Internet Technol.*, vol. 23, no. 1, pp. 18:1–18:21, Feb. 2023.

[59] "Alibaba maxcompute," 2024, Accessed Jul. 10, 2024. [Online]. Available: https://www.alibabacloud.com/zh/product/maxcompute

[60] Y. Huang et al., "Yugong: Geo-distributed data and job placement at scale," *Proc. VLDB Endowment*, vol. 12, no. 12, pp. 2155–2169, Aug. 2019.

[61] A. C. Zhou, B. Shen, Y. Xiao, S. Ibrahim, and B. He, "Cost-aware partitioning for efficient large graph processing in geo-distributed datacenters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 7, pp. 1707–1723, Jul. 2020.

[62] F. Zhang, J. Zhai, X. Shen, O. Mutlu, and X. Du, "POCLib: A high-performance framework for enabling near orthogonal processing on compression," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 2, pp. 459–475, Feb. 2022.

[63] L. Liu, H. Yu, G. Sun, L. Luo, Q. Jin, and S. Luo, "Job scheduling for distributed machine learning in optical WAN," *Future Gener. Comput. Syst.*, vol. 112, pp. 549–560, Nov. 2020.

[64] X. Xu et al., "Trading cost and throughput in geo-distributed analytics with a two time scale approach," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 2163–2177, Third Quarter 2022.

[65] K. Oh, M. Zhang, A. Chandra, and J. Weissman, "Network cost-aware geo-distributed data analytics system," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 6, pp. 1407–1420, Jun. 2022.

[66] A. Pradhan, S. Karthik, and S. Raghunandan, "Optimal query plans for geo-distributed data analytics at scale," in *Proc. 7th Joint Int. Conf. Data Sci. Manage. Data*, Bangalore India, 2024, pp. 247–251.

[67] S. M. Marzuni, A. Savadi, A. N. Toosi, and M. Naghibzadeh, "Cross-MapReduce: Data transfer reduction in geo-distributed MapReduce," *Future Gener. Comput. Syst.*, vol. 115, pp. 188–200, Feb. 2021.
[68] L. Fan, X. Zhang, Y. Zhao, K. Sood, and S. Yu, "Online training flow scheduling for geo-distributed machine learning jobs over heterogeneous and dynamic networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 10, no. 1, pp. 277–291, Feb. 2024.
[69] Y. Song, Y. Ai, X. Xiao, Z. Liu, Z. Tang, and K. Li, "HCEC: An efficient geo-distributed deep learning training strategy based on wait-free back-propagation," *J. Syst. Archit.*, vol. 148, Mar. 2024, Art. no. 103070.
[70] Z. Yang, Y. Cui, X. Wang, M. Li, and Y. Liu, "Less is more: Service profit maximization in geo-distributed clouds," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1925–1940, Third Quarter 2022.
[71] C. Li, Q. Cai, and Y. Lou, "Optimal data placement strategy considering capacity limitation and load balancing in geographically distributed cloud," *Future Gener. Comput. Syst.*, vol. 127, pp. 142–159, Feb. 2022.
[72] T. Xie, C. Li, N. Hao, and Y. Luo, "Multi-objective optimization of data deployment and scheduling based on the minimum cost in geo-distributed cloud," *Comput. Commun.*, vol. 185, pp. 142–158, Mar. 2022.
[73] A. Atrey, G. Van Seghbroeck, H. Mora, F. De Turck, and B. Volckaert, "SpeCH: A scalable framework for data placement of data-intensive services in geo-distributed clouds," *J. Netw. Comput. Appl.*, vol. 142, pp. 1–14, Sep. 2019.
[74] Y. Li, C. Fan, X. Zhang, and Y. Chen, "Placement of parameter server in wide area network topology for geo-distributed machine learning," *J. Commun. Netw.*, vol. 25, no. 3, pp. 370–380, Jun. 2023.
[75] J. Liu et al., "Efficient scheduling of scientific workflows using hot metadata in a multisite cloud," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1940–1953, Oct. 2019.
[76] M. W. Convolbo, J. Chou, C.-H. Hsu, and Y. C. Chung, "GEODIS: Towards the optimization of data locality-aware job scheduling in geo-distributed data centers," *Computing*, vol. 100, no. 1, pp. 21–46, Jan. 2018.
[77] C. Li, J. Zhang, T. Ma, H. Tang, L. Zhang, and Y. Luo, "Data locality optimization based on data migration and hotspots prediction in geo-distributed cloud environment," *Knowl.-Based Syst.*, vol. 165, pp. 321–334, Feb. 2019.
[78] W. Chen, B. Liu, I. Paik, Z. Li, and Z. Zheng, "QoS-Aware data placement for MapReduce applications in geo-distributed data centers," *IEEE Trans. Eng. Manag.*, vol. 68, no. 1, pp. 120–136, Feb. 2021.
[79] B. Yu and J. Pan, "A framework of hypergraph-based data placement among geo-distributed datacenters," *IEEE Trans. Serv. Comput.*, vol. 13, no. 3, pp. 395–409, May/Jun. 2020.
[80] K. Liu, J. Peng, J. Wang, W. Liu, Z. Huang, and J. Pan, "Scalable and adaptive data replica placement for geo-distributed cloud storages," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 7, pp. 1575–1587, Jul. 2020.
[81] H. Wang, H. Shen, Z. Li, and S. Tian, "GeoCol: A geo-distributed cloud storage system with low cost and latency using reinforcement learning," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst.*, DC, USA, 2021, pp. 149–159.
[82] T. Z. Emara and J. Z. Huang, "Distributed data strategies to support large-scale data analysis across geo-distributed data centers," *IEEE Access*, vol. 8, pp. 178526–178538, 2020.
[83] S. Nithyanantham and G. Singaravel, "Hybrid deep learning framework for privacy preservation in geo-distributed data centre," *Intell. Automat. Soft Comput.*, vol. 32, no. 3, pp. 1905–1919, 2022.
[84] A. C. Zhou, Y. Xiao, Y. Gong, B. He, J. Zhai, and R. Mao, "Privacy regulation aware process mapping in geo-distributed cloud data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 8, pp. 1872–1888, Aug. 2019.
[85] L. Chen, S. Liu, B. Li, and B. Li, "Scheduling jobs across geo-distributed datacenters with Max-Min fairness," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 3, pp. 488–500, Third Quarter 2019.
[86] F. Ebadifard and S. M. Babamir, "Federated geo-distributed clouds: Optimizing resource allocation based on request type using autonomous and multi-objective resource sharing model," *Big Data Res.*, vol. 24, 2021, Art. no. 100188.
[87] P. Zhang, X. Ma, Y. Xiao, W. Li, and C. Lin, "Two-level task scheduling with multi-objectives in geo-distributed and large-scale SaaS cloud," *World Wide Web*, vol. 22, no. 6, pp. 2291–2319, Nov. 2019.
[88] C. Li, J. Liu, M. Wang, and Y. Luo, "Fault-tolerant scheduling and data placement for scientific workflow processing in geo-distributed clouds," *J. Syst. Softw.*, vol. 187, May 2022, Art. no. 111227.
[89] Web AmazonServices, "Aws wavelength," Accessed: Jun. 01, 2025. [Online]. Available: https://aws.amazon.com/wavelength/
[90] Y. Chen, L. Luo, B. Ren, and D. Guo, "Geo-distributed IoT data analytics with deadline constraints across network edge," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22914–22929, Nov. 2022.
[91] X. Chatziliadis, E. T. Zacharatou, A. Eracar, S. Zeuch, and V. Markl, "Efficient placement of decomposable aggregation functions for stream processing over large geo-distributed topologies," *Proc. VLDB Endowment*, vol. 17, no. 6, pp. 1501–1514, 2024.
[92] L. Wang et al., "Compact scheduling for task graph oriented mobile crowdsourcing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 7, pp. 2358–2371, Jul. 2022.
[93] G. Castellano, S. Galantino, F. Risso, and A. Manzalini, "Scheduling multi-component applications across federated edge clusters with Phare," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 1814–1826, 2024.
[94] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan, and M. Xiao, "Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15513–15526, Dec. 2023.
[95] J. Ren et al., "Collaborative task offloading and resource scheduling framework for heterogeneous edge computing," *Wireless Netw.*, vol. 30, no. 5, pp. 3897–3909, Jul. 2024.
[96] Q. Tang et al., "Distributed task scheduling in serverless edge computing networks for the Internet of Things: A learning approach," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19634–19648, Oct. 2022.
[97] Q. Li, S. Wang, A. Zhou, X. Ma, F. Yang, and A. X. Liu, "QoS driven task offloading with statistical guarantee in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 278–290, Jan. 2022.
[98] A.-V. Michailidou, C. Bellas, and A. Gounaris, "Optimizing task allocation in multi-query edge analytics," *Cluster Comput.*, vol. 27, pp. 8289–8306, Apr. 2024.
[99] H. Liao, G. Tang, D. Guo, K. Wu, and L. Luo, "EV-Assisted computing for energy cost saving at edge data centers," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 9029–9041, Sep. 2024.
[100] F. Rossi, V. Cardellini, and F. L. Presti, "Elastic deployment of software containers in geo-distributed computing environments," in *Proc. IEEE Symp. Comput. Commun.*, Barcelona, Spain, 2019, pp. 1–7.
[101] J. Pang, Z. Han, R. Zhou, H. Tan, and Y. Cao, "Online scheduling algorithms for unbiased distributed learning over wireless edge networks," *J. Syst. Archit.*, vol. 131, Oct. 2022, Art. no. 102673.
[102] Z. Xu, G. Xu, H. Wang, W. Liang, Q. Xia, and S. Wang, "Enabling streaming analytics in satellite edge computing via timely evaluation of Big Data queries," *IEEE Trans. Parallel Distrib. Syst.*, vol. 35, no. 1, pp. 105–122, Jan. 2024.
[103] Z. Liu, X. Yuan, J. Yuan, J. Zhang, Z. Gu, and L. Zhang, "Multi-stage geo-distributed data aggregation with coordinated computation and communication in edge compute first networking," *J. Lightw. Technol.*, vol. 41, no. 8, pp. 2289–2300, Apr. 2023.
[104] W. Dou, B. Liu, C. Lin, X. Wang, X. Jiang, and L. Qi, "Architecture of virtual edge data center with intelligent metadata service of a geo-distributed file system," *J. Syst. Archit.*, vol. 128, Jul. 2022, Art. no. 102545.
[105] X. Lin, J. Wu, A. K. Bashir, W. Yang, A. Singh, and A. A. AlZubi, "FairHealth: Long-term proportional fairness-driven 5G edge healthcare in internet of medical things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 8905–8915, Dec. 2022.
[106] Y. Chen, Q. Yang, S. He, Z. Shi, J. Chen, and M. Guizani, "FTPipeHD: A fault-tolerant pipeline-parallel distributed training approach for heterogeneous edge devices," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 3200–3212, Apr. 2024.
[107] J. Xu and B. Palanisamy, "Cost-aware & fault-tolerant geo-distributed edge computing for low-latency stream processing," in *Proc. IEEE 7th Int. Conf. Collaboration Internet Comput.*, Atlanta, GA, USA, 2021, pp. 117–124.
[108] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Mach. Intell.*, vol. 2, no. 6, pp. 305–311, 2020.
[109] Google Cloud, "Encrypt workload data in-use with confidential GKE nodes," 2024, Accessed: May 26, 2025. [Online]. Available: https://cloud.google.com/kubernetes-engine/docs/how-to/confidential-gke-nodes
[110] K. Weil, J. Shannon, M. Qin, and R. Zellers, "Santa mode & video in advanced voice - 12 days of openai: Day 6," 2024, Accessed: Jan. 21, 2025. [Online]. Available: https://www.youtube.com/watch?v=NIQDnWlwYyQ

[111] K. Weil, A. Woodford, and A. Crookes, "1–800-chat-GPT - 12 days of openai: Day 10," 2024. Accessed: Jan. 21, 2025. [Online]. Available: https://www.youtube.com/watch?v=LWa6OHeNK3s
[112] A. Borzunov et al., "Distributed inference and fine-tuning of large language models over the internet," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 12312–12331.
[113] H. Okawa, X.-Z. Tao, Q.-G. Zeng, and M.-H. Yung, "Quantum-annealing-inspired algorithms for multijet clustering," *Phys. Lett. B*, vol. 864, 2025, Art. no. 139393.
[114] D. Ivanov, J. Smith, J. Wang, A. Petrov, S. Volkov, and D. Zhao, "Optimizing parallel processing with quantum-inspired task scheduling techniques," 2025. Accessed: May 25, 2025. [Online]. Available: https://www.researchgate.net/publication/389324244_Optimizing_Parallel_Processing_with_Quantum-Inspired_Task_Scheduling_Techniques
[115] P. Clark, "Toshiba achieves quantum communication over commercial fibre without cooling," *Financial Times*, 2025. Accessed: May 27, 2025. [Online]. Available: https://www.ft.com/content/51a65e45-302c-45fa-8bd1-c828a66b012d
[116] Web AmazonServices, "Amazon braket: Explore and experiment with quantum computing," 2025. Accessed: May 25, 2025. [Online]. Available: https://aws.amazon.com/cn/braket/
[117] IBM, "IBM quantum: Explore the quantum future," 2025. Accessed: May 25, 2025. [Online]. Available: https://quantum.ibm.com/
[118] Microsoft, "Azure quantum: Open ecosystem for quantum innovation," 2025. Accessed: May 25, 2025. [Online]. Available: https://azure.microsoft.com/en-us/products/quantum/

Yujian Wu received the bachelor's degree from the Hebei University of Technology, Tianjin, China, in 2024. He is currently working toward the MS degree with Tianjin University, Tianjin, China. His research interests include scheduling, load balancing and other resource allocation optimization problems.

Shanjiang Tang received the BS and MS degrees from Tianjin University (TJU), Tianjin, China, in July 2008 and Jan 2011, respectively, and the PhD degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2015. He is currently an associate professor with the College of Intelligence and Computing, Tianjin University, China. His research interests include parallel computing, cloud computing, Big Data analysis, and machine learning.

Ce Yu received the BS and MS degrees from Tianjin University, in 2002 and 2005, respectively, and the PhD degree in computer science from Tianjin University (TJU), Tianjin, China, in 2009. He is currently a professor and director of High Performance Computing Lab (HPCL) of Computer Science & Technology in Tianjin University. His main research interests include parallel computing, astronomy computing, cluster technology, Cell BE, multicore, grid computing.

Bin Yang received the BS and MS degrees from the Shandong University, Jinan, China, in 2015 and 2018, respectively, and the PhD degree in computer science and technology from Shandong University, in 2022. He was an associate researcher with Tsinghua University, Beijing, China. Currently he is an associate professor with Tianjin University, Tianjin, China. His research interests include high-performance computing, storage systems, performance analysis and modeling.

Chao Sun received the PhD degree in computer science from Tianjin University, Tianjin, China, in 2023. He is currently an engineer with High Performance Computing Center, Tianjin University, Tianjin, China. His main research interests include parallel computing and astronomy computing.

Jian Xiao received the PhD degree from Tianjin University, Tianjin, China. He is currently a senior engineer with the College of Intelligence and Computing, Tianjin University, Tianjin, China. He has backgrounds in both computational sciences and astroinformatics. His expertise includes numerical algorithms, high-performance computing. He was a visiting scholar with the Research Institute for Information Technology, Kyushu University, Japan.

Hutong Wu received the BS and MS degrees from Tianjin University (TJU), Tianjin, China, in July 2001 and March 2004, respectively. He is currently an associate professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests include cloud computing, visual analytics, and machine learning.

Jinghua Feng received the PhD degree from the National University of Defense Technology (NUDT), Changsha, China. He is currently a senior engineer and the chief engineer with the National Supercomputer Center in Tianjin, China. His research interests include high-performance computing, cloud computing, and artificial intelligence. His work focuses on intelligent operation and maintenance, heterogeneous computing resource scheduling, and system optimization.