Survey Of Distributed Task Offloading Optimization Based on Heuristic Algorithms in Edge Computing

Yuanrun Zheng *

Department of Nanjing University of Posts and Telecommunications, Nanjing, China

* Corresponding Author Email: 730297541@qq.com

Abstract. The explosive proliferation of terminal devices has generated massive volumes of data to be processed, exposing the limitations of traditional cloud computing in handling such ever-increasing data loads. The emergence of edge computing enables data processing at edge nodes, significantly improving the efficiency of data transmission and computation. Distributed task offloading based on heuristic algorithms is an important branch in edge computing. This survey provides a systematic review of distributed task offloading based on heuristic algorithms, comprehensively reviewing research on offloading problems empowered by ant colony optimization algorithms, particle swarm optimization algorithms, simulated annealing algorithms, and hybrid heuristic algorithms. The paper examines the effectiveness of task offloading under various algorithmic approaches, while investigating both the advantages of these algorithms and the challenges they face. This paper analyzes the inherent limitations of existing algorithms and identifies three core challenges confronting distributed task offloading empowered by heuristic algorithms. And proposes viable forward-looking optimization solutions addressing current research gaps, with these solutions focusing on interdisciplinary integration and multi-domain collaboration.

Keywords: Edge computing, Task offloading, Heuristic algorithms, Distributed based stations, Optimization strategies.

1. Introduction

The advent of cloud computing has fundamentally transformed IT delivery models, establishing itself as the core infrastructure of the digital economy. However, transmitting all data from devices to the cloud for processing consumes substantial network bandwidth due to the large volume of low-value-density information, resulting in significant resource inefficiencies. The substantial transmission distance introduces non-negligible feedback latency [1], which is intolerable in contemporary applications characterized by high dynamism. Edge computing emerges as an optimal—and often the only viable—solution for scenarios demanding ultra-low latency, bandwidth conservation, data localization guarantees, and offline operational capabilities. Edge computing adopts a three-tier "device-edge-cloud" collaborative architecture to distribute processing capabilities, relocating data processing, storage, and computational resources from the cloud closer to the data source. Edge computing addresses the inherent limitations of cloud computing in terms of real-time performance and bandwidth costs [2] by restructuring the geographical distribution of data processing, establishing itself as a critical foundation for modern smart city development.

Distributed task offloading represents a tightly coupled technological paradigm with edge computing, serving as the core mechanism for realizing edge computing efficacy. Distributed task offloading operates across spatial hierarchies and temporal granularities within the edge computing architecture. Through multidimensional analysis of feedback latency, transmission energy consumption, and other factors, the algorithm derives optimal offloading strategies to rationally distribute data across edge nodes, significantly enhancing system efficiency while reducing energy consumption and operational costs. Heuristic algorithms are widely employed in offloading decision-making due to their capability of rapidly searching for optimal solutions in complex spaces, exerting profound impacts on task offloading optimization. Distributed task offloading based on heuristic algorithms is an experience-driven scheduling approach that rapidly determines task execution locations, capable of identifying near-optimal offloading solutions within short timeframes. This method demonstrates significant advantages in addressing complex, dynamic edge network offloading problems with stringent timeliness requirements.

Edge computing has become a central driving force behind the intelligent transformation of industries, with recent years representing a pivotal juncture in its evolution from technological validation to widespread practical adoption. The efficacy of task offloading optimization is pivotal to the scalable deployment of edge computing systems. Distributed task offloading based on heuristic algorithms has been extensively studied, with the feasibility of its algorithmic models rigorously validated through numerous simulation experiments. However, challenges remain in terms of technological integration and production-oriented deployment. Positioned at this critical juncture, this survey provides a comprehensive review of recent advances in heuristic algorithm-empowered task offloading. We systematically analyze the strengths and limitations of existing approaches, and propose innovative pathways to address their shortcomings in industrial adoption and large-scale production deployment. Through this survey, beginners can gain a comprehensive understanding of recent research on heuristic algorithm-based distributed task offloading, establish a systematic knowledge framework in this field, and contribute to future industry advancements.

While existing surveys predominantly generalize challenges from perspectives such as algorithmic performance or scenario-specific characteristics, this paper innovatively identifies and elaborates on three fundamental dilemmas: dynamic topology tracking failure, multi-objective optimization trade-off dilemma and robustness issues. Breaking away from conventional optimization paradigms such as hybrid algorithm refinement and reinforcement learning empowerment, this study proposes a more promising interdisciplinary innovation framework with broader applicability. This survey demonstrates distinctive strengths in both systematic retrospection and forward-looking roadmap formulation.

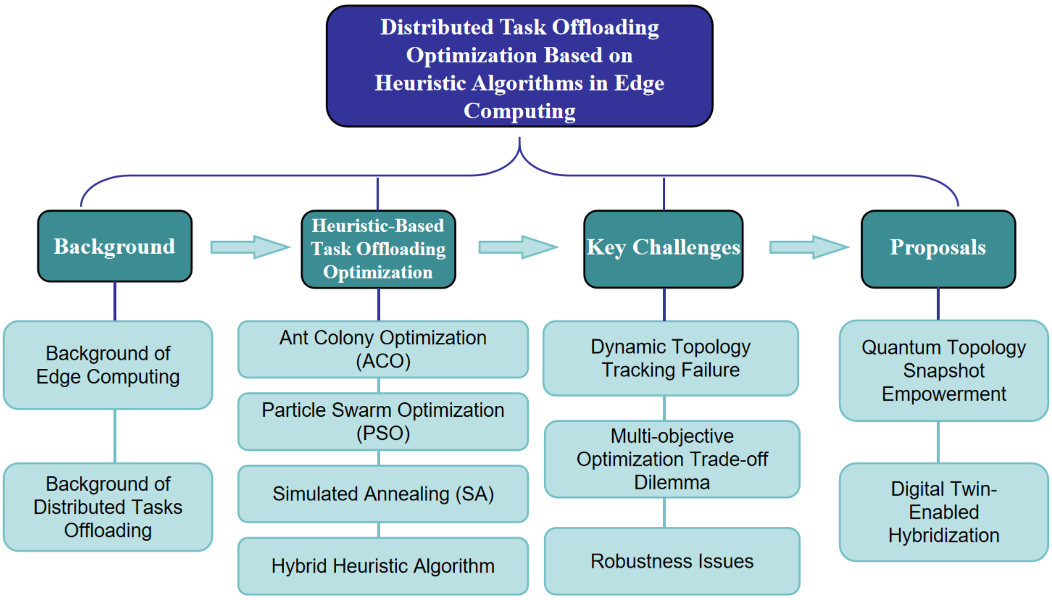

Fig. 1 Organizational Framework of the Survey

This survey is structured into four sections, with Figure 1 presenting the organizational framework of the survey. Chapter 2 introduces the concept of edge computing and elucidates the significance of distributed task offloading. Chapter 3 systematically reviews recent academic attempts in heuristic algorithm-based task offloading optimization, while providing comprehensive evaluations of each algorithm's performance. Chapter 4 examines the fundamental limitations of existing algorithms. Chapter 5 proposes actionable forward-looking solutions addressing current research gaps.

2. Background

2.1. Edge Computing

Edge computing represents a distributed computing paradigm that relocates data processing, storage, and applications from centralized clouds to edge nodes in closer proximity to end-users. This architectural shift effectively addresses critical bottlenecks of cloud computing in terms of latency, bandwidth, and security, thereby enabling real-time processing of massive-scale data. Edge computing adopts a three-tier architecture comprising cloud data centers, edge nodes, and terminal devices [1][3][4]. Cloud data centers possess massive data storage capacity and large-scale computational capabilities, responsible for global data analysis, model training, as well as long-term data storage and centralized management. Edge nodes, serving as the core infrastructure of edge computing architectures, are fundamentally heterogeneous distributed computing entities. Their computational power, storage, and network resource configurations exhibit significant gradient variations, spanning a continuous spectrum from metropolitan-level edge data centers to access-layer micro-infrastructure, specifically including: the infrastructure layer, network access layer, and terminal proximity layer. Terminal devices function as the physical entities that directly generate data, serving as both the data source and ultimate execution units in the three-tier edge computing architecture. Terminal devices perform fundamental computing locally through embedded capabilities while offloading computation-intensive tasks to edge nodes.

2.2. Distributed Task Offloading

Distributed task offloading is designed to address users' stringent timeliness requirements for data processing. By migrating computational workloads from resource-constrained terminal devices (with limited storage and computing capabilities) to edge nodes, this approach not only ensures timely data processing but also prevents cloud computing overload caused by centralized data aggregation while mitigating security risks such as data breaches. In edge computing environments, edge nodes operate under stringent computational resource constraints, while the task offloading process itself incurs substantial computational energy consumption and communication overhead [5]. To efficiently coordinate these resources, it is imperative to systematically address three core decision-making challenges in task offloading strategies: offloading decision-making, granularity control, and target node selection [6]. Heuristic algorithms can generate feasible solutions within polynomial time complexity. These algorithms demonstrate remarkable computational efficiency and dynamic adaptability when addressing critical challenges in edge computing environments, including terminal mobility, time-varying network states, and resource heterogeneity under multidimensional constraints [7].

3. Heuristic-Based Task Offloading Optimization

Heuristic algorithms have emerged as a core methodology for distributed task offloading decisions by virtue of their efficient search capabilities in complex solution spaces. By employing rule-driven or metaheuristic strategies, these algorithms generate near-optimal offloading solutions within polynomial time, significantly optimizing execution location selection and serving as a key enabling technology for edge intelligence. Heuristic algorithms encompass a diverse range of variants, among which four prominent approaches are specifically highlighted here: ant colony optimization-based, particle swarm optimization-based, simulated annealing-based, and hybrid heuristic algorithm-based distributed task offloading. Table 1 summarizes the key aspects of this section.

Table 1. Heuristic-Based Task Offloading Optimization

Algorithms	Characteristics	Advantages	Disadvantages	References
Ant Colony Optimization (ACO)	Pheromone-driven Parallel exploration	Competent for complex MOP	Slow convergence	[8][9][10][11][12][13][14]
Particle Swarm Optimization (PSO)	Swarm-coordinated Velocity-driven	Fast convergence	Premature convergence	[15][16][17][18][119][20][21]
Simulated Annealing (SA)	Temperature-scheduled	Global search capability	Slow convergence	[22][23][24][25][26]
Hybrid Heuristic Algorithm	Algorithmic synergy Self-adaptive mechanism	Enhanced solution quality	Design complexity Computational overhead	[27][28][29][30][31]

3.1 Distributed Task Offloading Based on Ant Colony Optimization Algorithms

The Ant Colony Optimization (ACO) algorithm transforms the task offloading problem into a path search problem, leveraging the collaborative mechanisms of ant colonies to identify optimal offloading strategies. Its positive feedback mechanism and parallel search characteristics endow the algorithm with unique advantages in adaptively handling complex dependency relationships, making it widely adopted for task offloading decision-making. The ACO algorithm demonstrably enhances task offloading success rates under high-user-density scenarios and elevated system loads [8]. The ant colony optimization algorithm achieves multi-objective optimization, effectively resolving system congestion caused by voluminous computational data while significantly reducing offloading latency [9]. In time-sensitive healthcare applications, the ACO has been deployed to schedule fog node tasks for load balancing, thereby accelerating task offloading response times [10]. Researchers have enhanced the pheromone update strategy to improve the ant colony algorithm [11], effectively reducing offloading costs under dynamic task quantity variations. The ACO demonstrates significant advantages in offloading decision efficiency for multi-objective optimization scenarios requiring simultaneous consideration of cost, node load balancing, and response time [12]. Enhanced ACO variants have been integrated with containerization technology to establish continuous multi-task offloading simulation systems, thereby improving task offloading efficacy [13]. Notably, the bi-objective ACO (bi-ACO) algorithm effectively addresses offloading problems with multiple constraints [14].

In ant colony optimization algorithms, each intelligent agent (artificial ant) makes autonomous decisions based on local information while achieving global coordination through pheromone exchange. This mechanism inherently aligns with the global optimization requirements of multi-node task offloading in edge computing. Such characteristics endow the algorithm with indisputable advantages in multi-objective optimization and complex task processing scenarios. Furthermore, the synergistic optimization capability of ACO: combining positive feedback mechanisms with heuristic search, endows it with superior optimization performance in resource-heterogeneous node environments. However, it exhibits several notable limitations, including slow convergence rates, poor stability, and inadequate dynamic adaptability. The ant colony optimization algorithm requires multiple iterations and pheromone updates to converge to a suboptimal solution, exhibiting low search efficiency that fails to meet the rapid convergence requirements for offloading decisions in highly dynamic edge environments. If the iteration count is forcibly reduced to accelerate the process, the algorithm becomes prone to local optima entrapment, significantly degrading solution quality.

Secondly, conventional ant colony optimization algorithms exhibit high sensitivity to initial parameter configurations, where improper settings can substantially compromise solution quality. The ant colony optimization algorithm's heavy reliance on historical information impedes its adaptability to rapidly changing dynamic environments, where outdated historical data may misguide current convergence directions. Furthermore, ACO incurs substantial communication and computational overheads—in large-scale distributed systems, the massive inter-node information exchanges consume significant network bandwidth, impose additional burdens on edge nodes, and inevitably increase feedback latency.

3.2 Distributed Task Offloading Based on Particle Swarm Optimization

The Particle Swarm Optimization (PSO) algorithm simulates the cooperative foraging behavior of bird flocks, transforming offloading decisions into particle search processes within the solution space, thereby efficiently identifying optimal task-resource allocation schemes. Compared with other methods, the PSO algorithm demonstrates superior performance in minimizing total execution time, enabling faster completion of offloading tasks [15]. The task offloading algorithm based on particle swarm optimization significantly reduces total system costs while effectively enhancing decision-making timeliness [16]. Researchers have further introduced dynamic weighting coefficients and adaptive learning factors to address multi-objective optimization, effectively accelerating convergence rates [17]. The Enhanced Improved PSO (EIPSO)-based computational offloading strategy establishes a dual-objective optimization model, effectively reducing both load imbalance and task offloading latency [18]. The adaptive hybrid PSO model achieves joint optimization of offloading decisions, data distribution, and load balancing, thereby enhancing global resource allocation capabilities [19]. The adaptive discrete PSO framework enables task offloading for a greater number of user equipment (UE), providing valuable references for 5G-enabled Mobile Edge Computing (MEC) system design [20]. Additionally, the incorporation of second-order oscillatory functions has been demonstrated to significantly enhance the optimization capacity of PSO algorithms [21].

As a population-based metaheuristic, particle swarm optimization inherently embodies swarm intelligence principles. This characteristic enables efficient resource allocation in task offloading scenarios through lightweight information exchange among nodes, effectively mitigating communication bottlenecks at central nodes while reducing computational overhead. Secondly, the algorithm's core mechanism—continuously updating particle velocities and positions—features computationally efficient logic that can rapidly generate near-optimal offloading strategies within limited iterations. This grants PSO notable advantages in convergence speed. However, it demonstrates relative limitations in convergence precision, algorithmic stability, and dynamic environment adaptability. In the particle swarm optimization algorithm, the global dominance of the velocity update formula intensifies progressively with iterations, compelling premature swarm convergence. This mechanism inherently restricts population diversity during evolutionary processes, consequently increasing susceptibility to local optima entrapment. The PSO algorithm relies on particles' historical optimal positions to explore global optima, yet in complex offloading environments with numerous nodes, this mechanism induces prohibitively long convergence times. Furthermore, the time-varying characteristics of edge node distributions render historical optimal data obsolete, ultimately compromising overall decision-making integrity. Moreover, critical parameters in the PSO algorithm—including evaporation factors and learning weights—require manual configuration without adaptive adjustment capabilities. This rigidity leads to imbalanced exploration-convergence capabilities in highly dynamic scenarios, consequently impairing optimal offloading decision-making. Particle position updates and fitness evaluations impose excessive computational burdens on resource-constrained edge devices, generating substantial resource pressure. This phenomenon becomes exacerbated with increasing node counts, where frequent inter-node communications consume significant bandwidth and markedly elevate feedback latency.

3.3 Distributed Task Offloading Based on Simulated Annealing Algorithms

Guided by the Metropolis criterion, the Simulated Annealing (SA) demonstrates dual advantages: the capability to escape local optima traps during initial search phases and concentrated fine-grained exploration within promising solution neighborhoods during later stages. This synergistic mechanism ensures statistically provable convergence to global optima [22]. Owing to its robust optimization capabilities, SA has been extensively adopted for task offloading decision-making. This algorithm demonstrably reduces both decision latency and energy consumption in multi-server single-task scenarios [23]. Researchers have further integrated conservative predictors to indirectly enhance task acceptance rates and reduce processing latency through algorithmic improvements [24]. In multi-edge-node environments, the Improved Simulated Annealing algorithm (iSA) demonstrates superior convergence speed and lower energy consumption via its joint optimization approach [25]. The SA-based Migratory Bird Optimization algorithm (SMBO) [26] achieves profit maximization within constrained timeframes through coordinated optimization of task offloading strategies.

Distributed task offloading based on simulated annealing algorithms effectively prevents local optima entrapment and demonstrates satisfactory convergence performance. Furthermore, unlike ACO and PSO, simulated annealing achieves optimization through iterative evolution of a single solution, eliminating the need to maintain population-wide information. This distinctive characteristic significantly reduces the implementation overhead of SA-empowered offloading algorithms, substantially decreasing node resource consumption. Consequently, SA-based approaches demonstrate superior deployability in resource-constrained distributed edge environments. However, it exhibits suboptimal adaptability in dynamic environments and underperforms in complex multi-node edge networks. The simulated annealing algorithm necessitates collecting resource states from all nodes during each iteration, incurring significant communication latency in large-scale multi-node edge networks. Moreover, its solution updates strictly depend on preceding iterative outcomes, fundamentally preventing genuine parallel optimization across distributed nodes. Simulated annealing-based task offloading requires manual configuration of temperature parameters, and its fixed cooling rate inherently limits adaptability to dynamic environments. The algorithm mandates exhaustive high-temperature exploration and low-temperature refinement phases to derive optimal solutions, consequently suffering from prohibitively slow convergence rates that fail to meet time-sensitive offloading decision requirements. Furthermore, environmental changes necessitate restarting the search process from the current temperature without leveraging historical optimization knowledge, resulting in redundant computations that squander substantial time and resources.

3.4 Hybrid Heuristic Algorithms

Hybrid heuristic algorithms integrate distinct algorithmic types to effectively mitigate the limitations inherent in individual approaches, thereby achieving superior optimization outcomes. The integration of simulated annealing with Binary Particle Swarm Optimization (BPSO) [27] leverages the Metropolis criterion to prevent conventional PSO from converging to local optima, while simultaneously optimizing computational resource allocation and uplink power distribution. Furthermore, the enhanced Genetic Binary PSO (GBPSO) algorithm [28] processes particle positions and velocities through multi-dimensional operations, significantly strengthening global search capabilities and improving offloading efficiency. The incorporation of Lévy flight into the PSO algorithm [29] enhances particle diversity, thereby achieving superior optimization of resource utilization across edge nodes. The Genetic Simulated Annealing-based PSO (GSAPSO) approach [30] synergistically combines genetic algorithms, simulated annealing, and particle swarm optimization, effectively reducing total energy consumption in edge servers while resolving offloading problems within shortened timeframes. Additionally, the integration of heuristic algorithms with deep reinforcement learning [31] demonstrates exceptional performance in both resource allocation and task offloading decision-making.

Distributed task offloading based on hybrid heuristic algorithms partially addresses the pronounced limitations of conventional heuristic approaches by mitigating their inherent trade-offs between advantages and disadvantages. The hybrid approach enables simultaneous achievement of rapid convergence and high-quality optimization. By incorporating adaptive weighting coefficients and learning factors [27], it mitigates excessive dependence on initial data configurations. However, such algorithms require establishing a unified objective function mapping framework to coordinate divergent convergence rates among constituent algorithms. Critical research challenges include minimizing inter-mathematical-model conflicts and enhancing cross-algorithm compatibility—key focus areas for advancing hybrid heuristic methodologies. Furthermore, hybrid architectures necessitate maintaining multiple algorithm states simultaneously, resulting in memory consumption that increases multiplicatively compared to standalone algorithms. When addressing ultra-large-scale tasks across numerous nodes, the computational complexity escalates dramatically. Balancing convergence performance with resource overhead remains a critical optimization challenge requiring further refinement in hybrid heuristic algorithms.

4. Key Challenges

4.1 Dynamic Topology Tracking Failure

In high-frequency dynamic scenarios where topological changes occur at millisecond-level intervals, the convergence latency of heuristic algorithms inherently lags behind real-time environmental variations. This fundamental mismatch between the rate of topological dynamics and the decision-making efficiency of heuristic approaches constitutes a critical performance bottleneck. In ultra-dynamic environments, topological transformations exhibit ultra-high-frequency characteristics, often involving non-gradual adjustments with intermittent abrupt transitions—a phenomenon overlooked in the design of most heuristic algorithms [32]. However, the decision-making logic of heuristic algorithms inherently relies on iterative optimization processes. Even particle swarm optimization—renowned for its convergence speed—requires multiple rounds of objective function evaluations and solution space exploration to approximate optimal solutions. The computational time required exhibits orders-of-magnitude disparity compared to topological change rates in ultra-dynamic scenarios. This latency directly results in algorithmic decisions being based on obsolete topological information that has undergone substantive alterations by the time of execution. Therefore, distributed task offloading systems based on heuristic algorithms urgently require enhanced adaptability to dynamic environments.

4.2 Multi-objective Optimization Trade-off Dilemma

In practical deployment scenarios, multi-objective optimization necessitates the joint consideration of latency, energy consumption, and cost within a unified framework. However, inherent challenges arise from fundamental disparities in the physical interpretations and dimensional units across these objectives, creating essential difficulties in quantitative normalization and commensurability. In the pursuit of optimal offloading strategies, the integration of heterogeneous objectives is critical, yet the selection of normalization methods introduces inherent subjectivity—different approaches can yield fundamentally divergent optimization outcomes. Secondly, latent objectives suffer from quantification deficiencies. Critical yet implicit metrics—such as data privacy preservation and node lifespan extension—resist precise numerical characterization. However, outright neglect of these offloading considerations risks generating optimization results that substantially deviate from practical requirements. Distributed task offloading based on heuristic algorithms typically employs fixed-weight approaches to consolidate multiple objectives into a single optimization target. However, such weight configurations heavily rely on empirical knowledge, and the predetermined weights inherently lack adaptability to abrupt scenario transitions. The dynamic variability of network topologies and resource states induces real-time shifts in optimization priorities across objectives. Meanwhile, inter-edge-node collaboration delays and information loss during communication cause distorted objective evaluations, ultimately undermining the reliability of foundational data for multi-objective trade-offs. Accurately characterizing real-time priority dynamics of optimization objectives and resolving multidimensional trade-offs remain critical challenges in current task offloading research.

4.3 Robustness Issues

Distributed task offloading based on heuristic algorithms follows structured iteration processes with convergence rules typically governed by static parameters. However, the core parameters governing practical task offloading persistently exhibit dynamic fluctuations with non-stationary characteristics. Consequently, empirical evidence consistently demonstrates that offloading solutions optimized for specific conditions frequently fail to maintain efficacy when deployed in divergent environments or under altered operational parameters [33]. In distributed systems, abrupt disturbances—including node failures, link interruptions, and malicious attacks—exhibit discontinuous patterns. Conversely, heuristic algorithms progressively approximate optimal solutions through iterative cycles, a process fundamentally reliant on historical state memorization. Abrupt environmental disturbances can disrupt algorithmic convergence progress, leading to either data loss or solution oscillations that severely compromise stability. Designing disturbance-resilient algorithms to enhance robustness remains a long-term research challenge requiring sustained investigation.

5. Proposals

5.1 Quantum Topology Snapshot Empowerment

Topology snapshots enable real-time recording of network or system topological states, while quantum bit superposition allows simultaneous evaluation of multiple topological configurations. Quantum-inspired sensor networks can concurrently capture state information from multiple edge nodes, achieving real-time synchronization of topology snapshots across distributed systems. Quantum computing-empowered heuristic algorithms efficiently integrate quantum parallelism with heuristic optimization capabilities, significantly enhancing topological data analysis efficiency while effectively addressing multi-node information exchange challenges in distributed task offloading. Quantum nonlocality enables instantaneous synchronization of topology snapshots across distributed nodes, while phase-locked entangled particles permit light-speed responses to environmental abruptions. This breakthrough transcends classical algorithmic communication bottlenecks, rendering real-time algorithmic adaptation feasible in dynamic environments.

5.2 Digital Twin-Enabled Hybridization

Digital twins construct virtual mirroring environments to simulate multiple offloading strategies, enabling rapid evaluation of their multi-objective performance and efficient screening of near-optimal solutions. Through multidimensional virtual simulation, digital twin technology circumvents objective trade-off failures induced by environmental dynamics and eliminates the need for repetitive heuristic algorithm parameter tuning in physical edge computing environments. This approach enhances the foresight of task offloading strategies, effectively ensuring their stability under dynamic multi-objective trade-off conditions. Furthermore, digital twins establish virtual sandboxes capable of conducting preemptive simulations during system idle periods, with optimized results cached to substantially reduce online computational overhead—thereby meeting stringent real-time requirements for task offloading. This mechanism fundamentally circumvents the inherent difficulties of executing complex heuristic algorithms on resource-constrained edge nodes in real-time.

6. Conclusion

As a core component of edge computing, distributed task offloading plays a crucial role in improving the timeliness of data processing and comprehensively utilizing the computing power of nodes. To comprehensively understand the data offloading problem, this paper systematically reviews the research on distributed task offloading based on heuristic algorithms in recent years, with a focus on discussing typical algorithms such as ant colony optimization, particle swarm optimization, simulated annealing, and hybrid heuristic algorithms. Meanwhile, this paper evaluates the performance of task offloading driven by various algorithms, analyzes their respective advantages and practical challenges, and summarizes three core problems faced by distributed task offloading empowered by heuristic algorithms: Dynamic Topology Tracking Failure, Multi-objective Optimization Trade-off Dilemma, and Robustness Issues. Based on the limitations in the current research field, two forward-looking optimization strategies based on interdisciplinary integration are proposed: Quantum Topology Snapshot Empowerment and Digital Twin-Enabled Hybridization. This paper conducts a multi-dimensional analysis of distributed task offloading based on heuristic algorithms, which not only helps beginners quickly grasp the development status of this field but also provides certain assistance for the advancement of the field.

References

[1] X. Song, Y. Wang, Z. Xie, and L. Xia, "A cloud-edge collaborative computing task scheduling and resource allocation algorithm for energy internet environment," KSII Transactions on Internet and Information Systems, vol. 15, no. 6, pp. 2282-2303, Jun. 2021.

[2] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in Proc. of the 6th International Conference on Advances in Future Internet, 2014.

[3] Y. Wang, X. Tao, X. Zhang, P. Zhang, and Y. T. Hou, "Cooperative task offloading in three-tier mobile computing networks: An ADMM framework," IEEE Transactions on Vehicular Technology, vol. 68, no. 3, pp. 2763-2776, Mar. 2019.

[4] Y.-D. Lin, Y.-C. Lai, J.-X. Huang, and H.-T. Chien, "Three-tier capacity and traffic allocation for core, edges, and devices for mobile edge computing," IEEE Transactions on Network and Service Management, vol. 15, no. 3, pp. 923-938, Sep. 2018.

[5] J. Bi, H. Yuan, S. Duanmu, M. Zhou and A. Abusorrah, "Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization," in IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3774-3785, 1 March1, 2021.

[6] J. Luo, Q. Qian, L. Yin, and Y. Qiao, "A game-theoretical approach for task offloading in edge computing," in Proc. of 2020 16th International Conference on Mobility, Sensing and Networking (MSN), 2020, pp. 756-761.

[7] S. Abuthahir and J. S. P. Peter, "Tasks offloading in vehicular edge computing network using meta-heuristic algorithms - A study of selected algorithms," in Proc. 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2024, pp. 1-6.

[8] T. Ahmed and J. Ahmed, "Delay minimization for offloaded tasks in UAV-assisted mobile edge computing using ant colony optimization," 2024 6th International Conference on Sustainable Technologies for Industry 5.0 (STI), Narayanganj, Bangladesh, 2024, pp. 1-6.

[9] Y. Sun, Z. Wu, K. Meng and Y. Zheng, "Vehicular task offloading and job scheduling method based on cloud-edge computing," in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 12, pp. 14651-14662, Dec. 2023.

[10] B. K. Alotaibi and U. Baroudi, "Offload and schedule tasks in health environment using ant colony optimization at fog master," 2022 International Wireless Communications and Mobile Computing (IWCMC), Dubrovnik, Croatia, 2022, pp. 469-474.

[11] H. Ge, J. Geng, Y. An, H. Feng, T. Zhou, and C. Huang, "Research on collaborative computational offload strategy based on improved ant colony algorithm in edge computing," in Proc. 5th Int. Conf. Nat. Lang. Process. (ICNLP), 2023.

[12] M. K. Hussein and M. H. Mousa, "Efficient task offloading for IoT-based applications in fog computing using ant colony optimization," in IEEE Access, vol. 8, pp. 37191-37201, 2020.

[13] J. Wang and H. Wang, "A secure data offloading strategy for UAV wireless networks based on improved ant colony algorithms," 2022 3rd International Conference on Electronics, Communications and Information Technology (CECIT), Sanya, China, 2022, pp. 57-61.

[14] Y. Wang, J. Zhu, H. Huang and F. Xiao, "Bi-objective ant colony optimization for trajectory planning and task offloading in UAV-assisted MEC systems," in IEEE Transactions on Mobile Computing, vol. 23, no. 12, pp. 12360-12377, Dec. 2024.

[15] D. Triyanto, I. W. Mustika, and Widyawan, "Delay-aware task offloading and bandwidth allocation using particle swarm optimization in mobile edge computing," in Proc. 16th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE), 2024.

[16] S. Li, H. Ge, X. Chen, L. Liu, H. Gong and R. Tang, "Computation offloading strategy for improved particle swarm optimization in mobile edge computing," 2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), Chengdu, China, 2021, pp. 375-381.

[17] Y. Wang, X. Li, S. Mao, B. Cai, J. He and J. Zhu, "Edge computing offload optimization strategy based on improved particle swarm," 2024 36th Chinese Control and Decision Conference (CCDC), Xi'an, China, 2024, pp. 5651-5656.

[18] S. Li, H. Ge, X. Chen, L. Liu, H. Gong and R. Tang, "Computation offloading strategy for improved particle swarm optimization in mobile edge computing," 2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), Chengdu, China, 2021, pp. 375-381.

[19] X. Cui and G. Chen, "Research on load balancing model of vehicle-to-everything communication transmission resources based on improved particle swarm optimization," 2023 IEEE 6th International Conference on Knowledge Innovation and Invention (ICKII), Sapporo, Japan, 2023, pp. 104-108.

[20] M. Wei, Z. Liu, W. Hu, S. Geng and X. Zhao, "Mobile edge computing task offloading based on ADPSO algorithm in multi-user environment," 2021 IEEE 4th International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China, 2021, pp. 126-130.

[21] D. Ye, X. Wang, and J. Hou, "An edge computing offloading algorithm based on second-order oscillatory particle swarm optimization," in Proc. 3rd Inf. Commun. Technol. Conf. (ICTC), 2022.

[22] A. Mahjoubi, K. -J. Grinnemo and J. Taheri, "An efficient simulated annealing-based task scheduling technique for task offloading in a mobile edge architecture," 2022 IEEE 11th International Conference on Cloud Networking (CloudNet), Paris, France, 2022, pp. 159-167.

[23] Y. Li, "Optimization of task offloading problem based on simulated annealing algorithm in MEC," 2021 9th International Conference on Intelligent Computing and Wireless Optical Communications (ICWOC), Chongqing, China, 2021, pp. 47-52.

[24] A. Mahjoubi, A. Ramaswamy and K. -J. Grinnemo, "An Online Simulated Annealing-Based Task Offloading Strategy for a Mobile Edge Architecture," in IEEE Access, vol. 12, pp. 70707-70718, 2024, doi: 10.1109/ACCESS.2024.3402611.

[25] T. Huang, S. Li and X. Gao, "Computing resource allocation and offloading method based on simulated annealing algorithm," 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Chongqing, China, 2020, pp. 276-282.

[26] H. Yuan, J. Bi, M. Zhou, J. Zhang and W. Zhang, "Profit-maximized task offloading with simulated-annealing-based migrating birds' optimization in hybrid cloud-edge systems," 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 2020, pp. 1218-1223.

[27] X. Chen and S. Zheng, "Resource allocation and task offloading strategy base on hybrid simulated annealing-binary particle swarm optimization in cloud-edge collaborative system," 2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 2022, pp. 379-383.

[28] Y. Deng, P. Wang and L. Li, "Task offloading for mixed cloud/fog computing in vehicular network using genetic particle swarm optimization," 2021 4th International Conference on Information Communication and Signal Processing (ICICSP), Shanghai, China, 2021, pp. 489-494.

[29] T. Gao, Q. Tang, J. Li, Y. Zhang, Y. Li, and J. Zhang, "A particle swarm optimization with Lévy flight for service caching and task offloading in edge-cloud computing," IEEE Access, vol. 10, pp. 76636–76647, 2022.

[30] J. Bi, H. Yuan, S. Duanmu, M. Zhou and A. Abusorrah, "Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization," in IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3774-3785, 1 March1, 2021.

[31] J. Bi, H. Yuan, S. Duanmu, M. Zhou and A. Abusorrah, "Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization," in IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3774-3785, 1 March1, 2021.

[32] A. Alkhateeb, I. Beltagy, and S. Alex, "Machine learning for reliable mmWave systems: Blockage prediction and proactive handoff," in Proc. GlobalSIP, 2018, pp. 1055–1059.

[33] S. S. Abuthahir and J. S. P. Peter, "Tasks offloading in vehicular edge computing network using meta-heuristic algorithms - A study of selected algorithms," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-10.