Cloud Computing in Mobile Edge Computing

Nandakumar S
BCA211355

SYNOPSIS

With the combination of Mobile Edge Computing (MEC) and the next generation cellular networks, computation requests from end devices can be offloaded promptly and accurately by edge servers equipped on Base Stations (BSs). However, due to the densified heterogeneous deployment of BSs, the end device may be covered by more than one BS, which brings new challenges for offloading decision, that is whether and where to offload computing tasks for low latency and energy cost. This paper formulates a multi-user-to-multi-servers (MUMS) edge computing problem in ultra-dense cellular networks. The MUMS problem is divided and conquered by two phases, which are server selection and offloading decision. For the server selection phases, mobile users are grouped to one BS considering both physical distance and workload. After the grouping, the original problem is divided into parallel multi-user-to-one-server offloading decision subproblems. To get fast and near-optimal solutions for these subproblems, a distributed offloading strategy based on a binary-coded genetic algorithm is designed to get an adaptive offloading decision. Convergence analysis of the genetic algorithm is given and extensive simulations show that the proposed strategy significantly reduces the average latency and energy consumption of mobile devices. Compared with the state-of-the-art offloading researches, our strategy reduces the average delay by 56% and total energy consumption by 14% in the ultra-dense cellular networks.

1.OBJECTIVE

The primary objectives of adaptive offloading in mobile-edge computing (MEC) using genetic algorithms are rooted in the overarching goal of optimizing system performance, enhancing user experience, and addressing critical challenges associated with edge computing. This multifaceted approach is driven by a set of interconnected objectives aimed at achieving efficiency, responsiveness, and sustainability in the dynamic landscape of ultra-dense 5G cellular networks.

One of the paramount objectives is the reduction of latency in data processing. The aim is to enable real-time interactions for applications that demand low-latency responses, including augmented reality, autonomous vehicles, and industrial control systems. By dynamically offloading computational tasks to edge servers based on genetic algorithm optimization, the system endeavors to ensure that users experience minimal delays, thereby enhancing the overall quality of service.

Optimizing the utilization of computational resources at the edge is another key objective. This involves load balancing to prevent resource bottlenecks and the efficient allocation of computing resources based on varying workloads. By intelligently distributing tasks among edge servers, the system aims to achieve an optimal resource utilization profile, contributing to improved system efficiency and performance.

Energy efficiency is a critical objective in the era of sustainable computing. By enhancing the energy efficiency of edge computing through optimized offloading decisions, the system endeavors to reduce the overall energy consumption of edge devices. This not only aligns with environmental sustainability goals but also has the potential to lower operational costs, making edge computing a more economical and eco-friendly solution.

The enhancement of the user experience is central to the objectives of adaptive offloading. By prioritizing low-latency communication and aligning offloading decisions with user preferences and application requirements, the system aims to provide users with seamless and responsive interactions. This user-centric focus ensures that the advantages of edge computing translate into tangible improvements in user satisfaction and engagement.

Furthermore, the system aims to exhibit adaptability to dynamic conditions. In ultra-dense 5G cellular networks, where network conditions, user mobility, and workloads are subject to constant change, an adaptive offloading strategy becomes essential. The genetic algorithms employed seek to dynamically adjust offloading decisions to accommodate variations in demand and network status, maintaining responsiveness in dynamic environments.

Load balancing is a critical facet of the objectives, preventing server congestion and promoting a balanced distribution of tasks across edge servers. This not only ensures the optimal use of resources but also guards against performance degradation and bottlenecks, contributing to the overall reliability and stability of the edge computing system.

Privacy preservation emerges as a crucial objective, acknowledging the sensitivity of user data. The system aims to implement robust mechanisms that safeguard user privacy during the offloading process. By incorporating privacy-preserving techniques, such as encryption and secure protocols, the offloading decisions prioritize the protection of user data, addressing growing concerns regarding data security and privacy.

2.INTRODUCTION

The development of mobile applications facilitates people's life, such as augmented reality and intelligent interconnection, etc. These applications are characterized by a large amount of computation and a low tolerance for latency. However, mobile devices are limited in computation and endurance capacity. Mobile edge computing (MEC) technology has been regarded as an effective solution to the above constraints. Mobile devices get powerful computing capacity and lower energy consumption when the computing tasks are offloaded to edge servers. Moreover, the integration of the MEC technology and next generation networks, such as the fifth/sixth generation networks (5G/6G), can save bandwidth resources and improve user experience.

Though enjoying the super-high data transmission rate brought by the next generation networks, mobile devices still face new problems, such as the network densification and low signal penetration. Many existing works formulated the MEC problem in 4G cellular networks as multi-user to one single-server model , and their concern was task local processing or remote offloading to the base station. Compared with 4G, the frequency of the 5G network is much higher, which means the faster its attenuation will be. Considering the development of antenna technology and the compensation of radio frequency technology, the 5G construction density of China Unicom and China Telecom should be about 1.7 times that of the 4G base station, and 3.2 times for China Mobile. The ultra density of base stations means overlapped coverage of mobile devices, which brings a new challenge for offloading decisions: which accessible BS is the best one to offload tasks to achieve low latency and energy consumption?

In this paper, we formulate the offloading decision problem in the ultra-dense BSs environment as a nonlinear integer optimization problem, which is proved to be NP-Hard .Then a suboptimal algorithm is designed to solve this problem. First, by considering the preference of both proximity and workload, mobile users are grouped to one BS with the highest preference as a destination to offload tasks.

With the result of grouping, the original problem is decomposed into several distributed 0-1 integer optimization sub-problems. After that, a binary-coded Genetic Algorithm Based Distributed Offloading Strategy (GABDOS) is proposed to solve these sub-problems in parallel. The main contributions of this paper are as follows:

1. A Multi-User-to-Multi-Servers edge computing offloading problem, termed as MUMS, is proposed, which is especially for the ultra-dense 5G cellular networks.

2. The multi-user-to-multi-servers problem is divided and conquered by two phases, which are server selection and offloading decision. For the server selection phases, mobile users are grouped to the BS with the highest preference, which fully considers physical distance and workload of base stations.

3. After the grouping, the original multi-user-to-multi-servers problem is converted into several parallel offloading decision subproblems, each of which can be formulated as a nonlinear integer optimization problem. To get a fast near-optimal solution, a binary-coded Genetic Algorithm Based Distributed Offloading Strategy (GABDOS) is designed to get an adaptive offloading decision.

4. Convergence analysis of the genetic algorithm is given and extensive simulations show the performance of our solution from the impact of user amount, user device CPU frequency, and trade-off parameter for offloading latency and energy of users.

3.HISTORY

The history of cloud computing in mobile applications is a journey marked by transformative milestones, shaping the landscape of digital technology and redefining the capabilities of our handheld devices.

Early Foundations (2000s): The convergence of cloud computing and mobile applications found its roots in the early 2000s. The introduction of 3G and the widespread use of smartphones paved the way for a move toward mobile applications with more features and complexity. However, developers faced difficulties in delivering advanced functionalities due to the constrained processing power and storage capacities of mobile devices.

Emergence of Cloud Services (2010s): The widespread use of cloud services in the 2010s brought about a paradigm shift. Reputable companies like Google Cloud, Microsoft Azure, and Amazon Web Services (AWS) began providing scalable and reasonably priced cloud solutions. This lessened the load on mobile devices and made room for more complex applications by enabling developers to offload processing tasks and store data on remote servers.

Mobile-Cloud Fusion (Mid-2010s): In the middle of the decade, mobile-cloud fusion became popular. Developers started using cloud services for computation as well as storage, which allowed resource-intensive tasks to be run on potent cloud servers. As a result, mobile applications became much more capable and performed better, which made it possible to implement innovations like augmented reality, machine learning, and real-time collaboration on mobile devices.

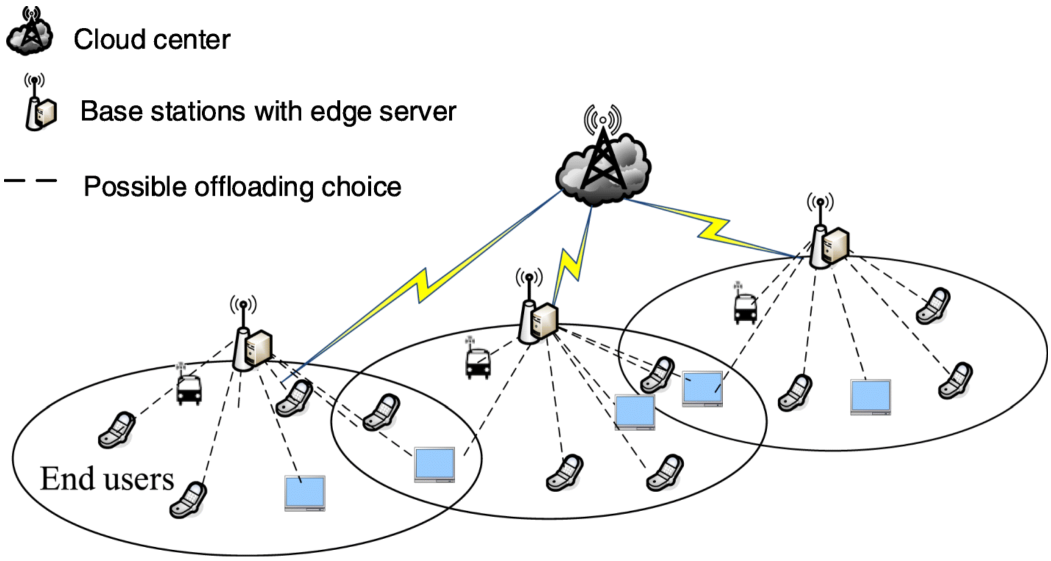
Edge Computing Integration (Late 2010s – Early 2020s): The need for reduced latency and enhanced productivity led to the integration of edge computing into the mobile-cloud ecosystem. By moving some computational operations closer to the user, edge computing has improved real-time interactions and decreased latency. The mobile application experience was further enhanced by this integration, especially in the gaming, healthcare, and Internet of Things domains.

Current Landscape (2020s): The modern cloud computing and mobile application relationship has developed into a dynamic alliance. Today's developers can quickly develop, test, and deploy applications thanks to the wide range of cloud-based tools and services available to them. Thanks to the widespread use of serverless computing and cloud-native architectures, developers can now build mobile apps that are responsive, scalable, and resilient.

As we continue with this seminar, it will become increasingly important to comprehend the past development of cloud computing in mobile applications in order to fully recognize the revolutionary possibilities that lie ahead. The progression from early limitations to the present day of smooth integration highlights the ongoing innovation and cooperation that characterize the meeting point of mobile applications and cloud co

4.WORKING

As shown in Fig., m base stations are denoted as set $B=\{b_1,b_2,\ldots,b_m\}$, each of which is fixed with an edge server. The set of m servers is marked as $S=\{s_1,s_2,\ldots,s_m\}$. End users $U=\{u_1,u_2,\ldots,u_n\}$ are distributed in the area with geometric distribution. The set of computing tasks generated by all users is denoted as $T=\{t_1,t_2,\ldots,t_n\}$. Because of the high density of 5G BSs, the coverage area of servers will overlap with each other. That is to say, most users will be covered by more than one BS.



About the mobility of end-users. In the area of high population density, users move at 2-3 kilometers per hour and vehicles run at about 20 kilometers per hour, whose shift is relatively small for a 5G cellular station with a coverage radius of 500 to 1000 meters. Therefore, it can be assumed that the end-users' position is approximately stationary in each given time slot.

As mentioned earlier, user devices transmit data with the BS through the wireless channel. The data transmission rate is determined by the general communication model, Shannon-Hartley theorem

$$r_{u_i,b_j} = \frac{W_{b_j}}{N}\log_2\left(1 + \frac{p_{u_i} g_{u_i,b_j}}{\omega + \sum_{k=1,k\neq i}^{N} p_{u_i} g_{u_i,b_j}}\right),$$

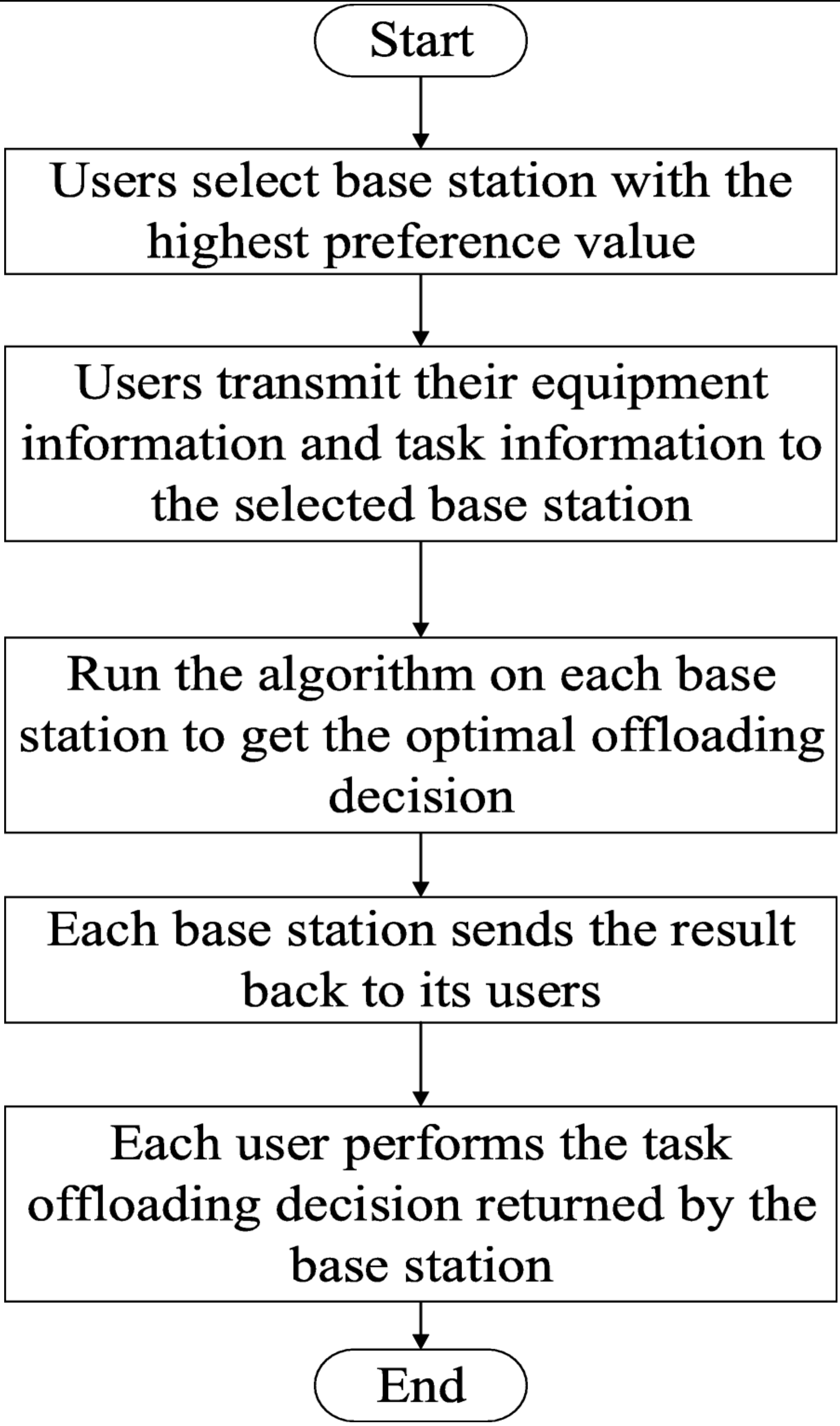
Where W_{bj} denotes the bandwidth of BS b_j, N denotes the number of users the BS b_j served, P_{ui} denotes the transmission power of user u_i,g_{ui},$b_j$denotes the channel gain between user u_i and BS b_j, and ω denotes background noise. Because the edge server is equipped on the BS, the transmission delay between them is very small, which can be regarded as zero. Therefore, the data transmission rate between the BS and the user equipment can directly be used to represent the data transmission rate between the user equipment and the edge server:

$$r_{u_i,s_j} = r_{u_i,b_j}\,.$$

5.TECHNICAL SPECIFICATION

Mobile users need to select one of the reachable edge servers for task offloading. To obtain the optimal offloading option is a combinatorial explosion problem. In this paper, a genetic algorithm based distributed offloading strategy is proposed. The general idea of the strategy is as follows and Fig. 2 gives an intuitive flow chart of process.

- First, according to the user grouping strategy based on preferences, mobile users are assigned to the BS with the highest preference as a destination to offload tasks. The preference is determined by a tradeoff computation of both distance and workload. On the one hand, the closer the distance between the BS and the user, the larger the preference value is. On the other hand, the smaller the BS's workload, the larger the preference value.


Start
Users select base station with the highest preference value
Users transmit their equipment information and task information to the selected base station
Run the algorithm on each base station to get the optimal offloading decision
Each base station sends the result back to its users
Each user performs the task offloading decision returned by the base station
End

Then, the original problem will be decomposed into multi-user-to-single-server subproblems, which means that each BS is responsible for only a few users. In each group, users either process locally or offload tasks. So the subproblems are 0-1 selection problems. Because these subproblems are independent of each other, they can be processed in a distributed parallel.

Finally, genetic algorithm (GA) is used on each edge server to solve the 0-1 selection problems and get near-optimal offloading decisions.

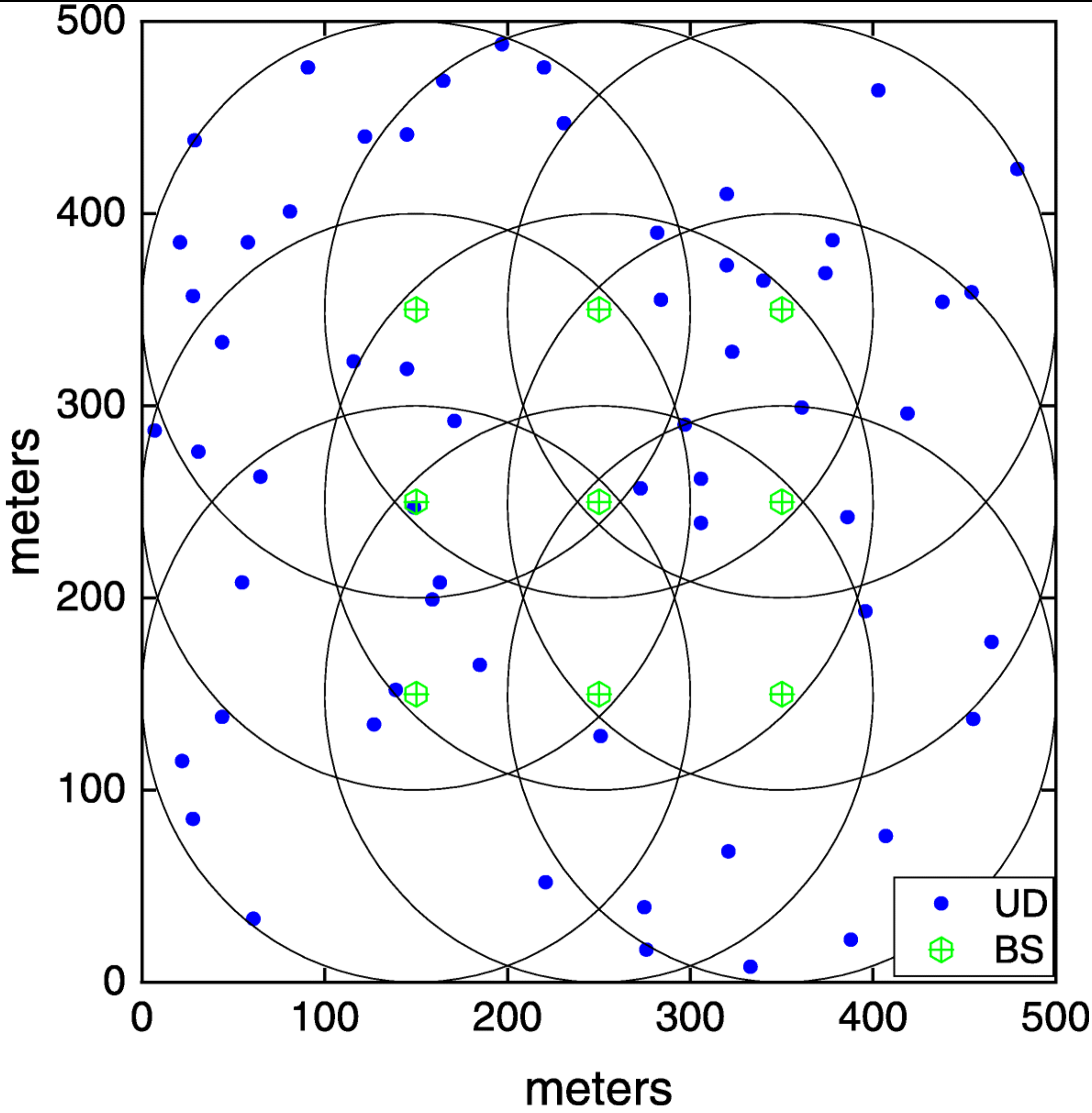
Algorithm 2: Genetic Algorithm Based Distributed Offloading Strategy

Input: Initial population: P_0
Population size: N
The largest generation: G
The maximum fitness set of each generation population: $F = \{f_1, f_2, \ldots, f_N\}$
Maximum fitness difference threshold: K
Crossover probability: C_P
Mutation probability: M_p

Output: Optimal individual O

$g \leftarrow 0$;
$F \leftarrow \emptyset$;
while $g < G$ **do**
  for $i = 1$ *to* N **do**
    Evaluate fitness of P_g;
    $f_g \leftarrow$ The maximum fitness;
    $O \leftarrow$ Maximum fitness individual;
  for $i = 1$ *to* N **do**
    Select operation to P_g;
  for $i = 1$ *to* $N/2$ **do**
    if *random(0,1)* $< C_p$ **then**
      Crossover operation to P_t;
  for $i = 1$ *to* N **do**
    if *random(0,1)* $< M_p$ **then**
      Mutation operation to P_g;
  if $g \neq 0$ && $f_g - f_{g-1} < K$ **then**
    break;
  $P_{g+1} \leftarrow P_g$;
  $g \leftarrow g + 1$;
return O;

Assuming there are N BSs and M users, the proposed offloading strategy can be divided into two parts. First, each user needs to traverse all the reachable base stations to select the base station with the highest preference. Therefore, the time complexity of this part is $O(N \times M)$. Then, each base station needs to calculate the optimal offloading decision. Since all base stations can process the step in parallel, the time complexity of a certain base station can represent the time complexity of this part. Since GA is adopted here, the time complexity of this part is mainly related to the parameters of GA. G is used to represent the generation of GA. P is used to represent the population.The size of the individuals is equal to the number of users. Therefore, the time complexity in this part can be summed up as $O(G \times P \times M)$. The time complexity of the latter part accounts for the main part of the whole algorithm. The convergence of GA has a great impact on its time complexity, so its convergence described in detail in the next section.


500
400
300
200
100
0
0
100
200
300
400
500
meters
meters
UD
BS

6.ADVANTAGES

1. **Dynamic Adaptability:** Adaptive offloading allows the system to dynamically adjust to changing network conditions and user demands. The genetic algorithm's evolutionary nature enables the optimization of offloading decisions over time, ensuring that the system remains adaptable to varying workloads and network states in ultra-dense environments.

2. **Efficient Resource Utilization:** The genetic algorithm can efficiently explore the vast solution space to find optimal offloading strategies. This leads to improved resource utilization within ultra-dense cellular networks, minimizing latency and maximizing the use of available computational resources at the edge.

3. **Load Balancing:** The adaptive offloading mechanism, guided by the genetic algorithm, facilitates effective load balancing across multiple base stations (BS) in ultra-dense networks. By distributing user workloads based on evolving conditions, it prevents network congestion, reduces bottlenecks, and ensures a more equitable utilization of resources among BS.

4. **Low Latency and High-Quality of Service (QoS):** The adaptive nature of the offloading decisions, driven by the genetic algorithm, contributes to low-latency connections. Users experience improved QoS as the system dynamically optimizes offloading strategies based on proximity and workload considerations, resulting in reduced communication delays.

5. **Scalability:** Genetic algorithms are inherently scalable and can handle complex optimization problems efficiently. In the context of ultra-dense networks with a large number of users and BS, the adaptive offloading solution using a genetic algorithm can scale effectively to address the increasing demands of the network.

6. **Fault Tolerance:** The adaptability of the offloading strategy provides a level of fault tolerance. If there are unexpected changes or failures in the network, the genetic algorithm can quickly adapt the offloading decisions to mitigate the impact on user experience and maintain the overall performance of the system.

7. **Energy Efficiency:** The genetic algorithm can optimize offloading decisions not only for low latency but also for reduced energy consumption. By intelligently distributing computational tasks based on proximity and workload, the adaptive offloading strategy contributes to energy-efficient operations in ultra-dense cellular networks.

In summary, the use of adaptive offloading in mobile-edge computing, coupled with a genetic algorithm, brings forth a range of benefits such as dynamic adaptability, efficient resource utilization, load balancing, low latency, scalability, fault tolerance, and energy efficiency. These advantages collectively contribute to the optimization of edge computing in the context of ultra-dense cellular networks.

7.DISADVANTAGES

1. **Computational Complexity:** Genetic algorithms, due to their iterative and evolutionary nature, can be computationally intensive. The optimization process involves multiple iterations and evaluations, potentially leading to increased processing overhead, especially in resource-constrained edge computing environments.

2. **Algorithm Tuning Challenges:** Configuring and fine-tuning the parameters of a genetic algorithm can be a non-trivial task. The effectiveness of the algorithm may heavily depend on the proper adjustment of parameters, and finding the optimal settings might require extensive experimentation.

3. **Real-time Constraints:** Genetic algorithms are generally designed for optimization over a period of time and may not always meet the strict real-time constraints of certain applications. In scenarios where immediate decision-making is crucial, the inherent time complexity of the genetic algorithm may pose challenges.

4. **Limited Exploration in Large Solution Spaces:** In ultra-dense networks with a vast solution space, genetic algorithms may face limitations in thoroughly exploring all possibilities. There is a risk of the algorithm converging to suboptimal solutions, especially if the diversity of the population is not effectively maintained.

5. **Communication Overhead:** The adaptive offloading decisions involve continuous communication between edge devices and base stations for exchanging information related to user proximity, workload, and optimization parameters. This communication overhead might strain network resources and affect overall system efficiency.

6. **Dependency on Quality of Fitness Function:** The performance of a genetic algorithm heavily relies on the definition of a suitable fitness function. If the fitness function does not accurately capture the optimization goals or if it is improperly defined, the algorithm may produce suboptimal results.

7. **Sensitivity to Initial Population:** Genetic algorithms are sensitive to the initial population. The effectiveness of the optimization process may be influenced by the quality and diversity of the initial set of solutions, and choosing an inappropriate initial population might hinder the algorithm's ability to converge to optimal solutions.

8. **Limited Explainability:** Genetic algorithms, being heuristic optimization techniques, might lack transparency and interpretability. Understanding the decision-making process of the algorithm can be challenging, which may be a concern in applications where interpretability is crucial.

8.APPLICATIONS

1. **Smart Cities:** Optimizing offloading decisions in smart city applications, such as traffic management and surveillance, enables real-time processing of data at the edge. Genetic algorithms help balance the computational load among edge servers, ensuring responsive and efficient smart city operations.

2. **Healthcare IoT:** In healthcare, MEC with adaptive offloading using genetic algorithms can enhance the processing of medical data from IoT devices at the edge. This ensures low-latency analytics, vital for applications like remote patient monitoring and real-time diagnostics.

3. **Industrial IoT (IIoT):** Adaptive offloading is crucial in IIoT scenarios where edge computing handles real-time control and monitoring. Genetic algorithms can optimize offloading strategies, improving latency and reliability in industrial processes.

4. **Autonomous Vehicles:** Edge computing with adaptive offloading is beneficial for autonomous vehicles. Genetic algorithms can optimize decisions related to data processing at the edge, contributing to reduced latency and improved safety in real-time decision-making for self-driving cars.

5. **Augmented Reality (AR) and Virtual Reality (VR):** AR and VR applications demand low latency to deliver immersive user experiences. Adaptive offloading using genetic algorithms helps distribute computational tasks efficiently, ensuring seamless AR and VR interactions at the edge.

6. **Smart Grids:** In energy management systems, adaptive offloading can optimize the processing of data from smart grids at the edge. Genetic algorithms contribute to efficient load balancing and decision-making for real-time energy monitoring and control.

7. **Retail Analytics:** Retailers benefit from MEC with adaptive offloading for processing customer behavior data at the edge. Genetic algorithms can optimize the offloading strategy to ensure timely insights for personalized marketing and inventory management.

8. **Agricultural IoT:** In precision agriculture, adaptive offloading assists in processing data from IoT sensors on the field. Genetic algorithms optimize offloading decisions for timely analysis, aiding in crop monitoring, irrigation control, and resource optimization.

9. **Edge-based Gaming:** For cloud gaming and mobile gaming, adaptive offloading with genetic algorithms ensures optimal distribution of computational tasks between edge servers and user devices. This minimizes latency and enhances the gaming experience.

10. **Emergency Response Systems:** Adaptive offloading is crucial in emergency response systems for real-time data processing from sensors and devices. Genetic algorithms help optimize resource allocation, ensuring rapid and effective decision-making during critical situations.

11. **Financial Services:** In financial applications, adaptive offloading using genetic algorithms can optimize the processing of real-time market data and transaction requests at the edge. This contributes to low-latency financial analytics and improved trading efficiency.

9.CONCLUSION

In conclusion, this paper introduces a novel approach, termed Multi-User-to-Multi-Servers Offloading Decision (MUMS), tailored for the complexities of edge computing in ultra-dense 5G cellular networks. The proposed solution addresses the challenge of optimizing offloading decisions by considering both proximity and workload of Base Stations (BS) in a multi-user environment. Through a systematic two-phase process, the MUMS problem is effectively conquered.

The first phase involves grouping users to BS based on a preference metric that intricately balances proximity and workload considerations. By doing so, the system ensures low-latency connections for users while also distributing the workload evenly among the available BS, contributing to improved overall network efficiency.

The second phase of the solution comprises decomposing the original MUMS problem into sub-problems with 0-1 selectivity. Subsequently, a parallel processing approach is employed, utilizing a Genetic Algorithm named GABDOS. This algorithm is designed to efficiently explore and solve the sub-problems simultaneously, providing a scalable and optimized solution for the offloading decision-making process.

Through rigorous verification, the proposed GABDOS scheme demonstrates its efficacy in achieving a balanced offloading decision. The evaluation focuses on key performance indicators, notably low latency and reduced user energy consumption. The results affirm the capability of the proposed solution to strike a harmonious balance between providing users with enhanced real-time responsiveness and ensuring the efficient utilization of network resources.

In summary, the paper presents a comprehensive and innovative framework for addressing the MUMS problem in ultra-dense 5G cellular networks. By incorporating proximity-aware user grouping and leveraging a parallelized Genetic Algorithm, the proposed scheme stands out as a promising solution for optimizing offloading decisions, ultimately contributing to a more responsive, energy-efficient, and well-balanced edge computing environment in the context of ultra-dense 5G networks.

10.FUTURE ENHANCEMENT

1. **Machine Learning Integration:** Integrate advanced machine learning techniques, such as reinforcement learning or deep learning, with genetic algorithms. This hybrid approach can enhance the adaptability and learning capabilities of the system, enabling it to better respond to dynamic network conditions and user behaviors.

2. **Edge-Cloud Collaboration:** Explore strategies for effective collaboration between edge computing and cloud resources. Develop mechanisms to dynamically offload tasks between the edge and cloud based on workload characteristics, ensuring optimal resource utilization and scalability.

3. **Energy-Aware Optimization:** Incorporate energy-aware optimization algorithms to minimize the environmental impact and improve the sustainability of edge computing. Consider offloading decisions that not only prioritize low latency but also take into account energy efficiency, especially in resource-constrained environments.

4. **Privacy-Preserving Offloading:** Enhance privacy-preserving mechanisms in offloading decisions. Develop algorithms that ensure sensitive user data remains secure during the offloading process, incorporating encryption techniques and decentralized approaches to protect user privacy.

5. **5G and Beyond Integration:** Adapt offloading strategies to leverage the capabilities of evolving network technologies, including 5G and beyond. Explore how the integration of advanced communication technologies can influence offloading decisions and improve overall system performance.

6. **Edge Security Enhancement:** Strengthen the security of edge computing environments by integrating robust security mechanisms. Consider the implications of offloading decisions on system vulnerabilities and develop strategies to mitigate potential security risks.

7. **Federated Learning for Edge Devices:** Explore federated learning approaches to enable edge devices to collaboratively train models without exchanging raw data. This can enhance the efficiency of model training at the edge while preserving data privacy.

8. **Quantum Computing Integration:** Investigate the potential benefits of integrating quantum computing in optimization processes. Quantum algorithms may provide superior optimization capabilities, and their integration could lead to more efficient and faster offloading decisions.

9. **Distributed Offloading Decisions:** Develop distributed offloading decision-making mechanisms that allow edge devices to collaboratively optimize task allocation. This approach can enhance scalability and efficiency in large-scale edge computing environments.

10. **Multi-Objective Optimization:** Extend the optimization criteria to include multiple objectives, such as minimizing latency, maximizing energy efficiency, and optimizing resource utilization simultaneously. Multi-objective genetic algorithms can provide a holistic approach to offloading decision optimization.

11. **Edge Orchestrator Platforms:** Create comprehensive edge orchestrator platforms that facilitate dynamic offloading decisions. These platforms can intelligently manage and coordinate offloading strategies across diverse edge devices and environments.

12. **Standardization and Interoperability:** Work towards standardizing offloading decision processes and ensuring interoperability among different edge computing systems. This can facilitate seamless integration and collaboration in heterogeneous edge environments

11.REFERENCES

1. Mach P, Becvar Z (2017) Mobile Edge Computing: A Survey on Architecture and Computation Offloading. IEEE Commun Surv Tutorials 19(3):1628–1656. https://doi.org/10.1109/COMST.2017.2682318.Chellappa, R., & Sarathy, R. (2005). Service-Oriented Computing and Cloud

2. Hsieh H-C, Chen J-L, Benslimane A (2018) 5g virtualized multi-access edge computing platform for iot applications. J Netw Comput Appl 115:94–102. https://doi.org/10.1016/j.jnca.2018.05.001.

3. Tomkos I, Klonidis D, Pikasis E, Theodoridis S (2020) Toward the 6g network era: Opportunities and challenges. IT Prof 22(1):34–38.

4. Bhushan N, Li J, Malladi D, Gilmore R, Brenner D, Damnjanovic A, Sukhavasi RT, Patel C, Geirhofer S (2014) Network densification: the dominant theme for wireless evolution into 5g. IEEE Commun Mag 52(2):82–89.

5. Du J, Yu FR, Chu X, Feng J, Lu G (2019) Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization. IEEE Trans Veh Technol 68(2):1079–1092. https://doi.org/10.1109/TVT.2018.2883156.

6. Sun H, Zhou F, Hu RQ (2019) Joint Offloading and Computation Energy Efficiency Maximization in a Mobile Edge Computing System. IEEE Trans Veh Technol 68(3):3052–3056. https://doi.org/10.1109/TVT.2019.2893094.

7. Qi Q, Wang J, Ma Z, Sun H, Cao Y, Zhang L, Liao J (2019) Knowledge-Driven Service Offloading Decision for Vehicular Edge Computing: A Deep Reinforcement Learning Approach. IEEE Trans Veh Technol 68(5):4192–4203. https://doi.org/10.1109/TVT.2019.2894437.

8. Poularakis K, Llorca J, Tulino AM, Taylor I, Tassiulas L (2019) Joint Service Placement and Request Routing in Multi-cell Mobile Edge Computing Networks. Proc IEEE INFOCOM 2019-April:10–18. https://doi.org/10.1109/INFOCOM.2019.8737385.

9. Tran TX, Pompili D (2019) Joint task offloading and resource allocation for multi-server mobile-edge computing networks. IEEE Trans Veh Technol 68(1):856–868. https://doi.org/10.1109/TVT.2018.2881191.

10. Li G, Cai J (2020) An online incentive mechanism for collaborative task offloading in mobile edge computing. IEEE Trans Wirel Commun 19(1):624–636. https://doi.org/10.1109/TWC.2019.2947046.

11. Li C, Tang J, Luo Y (2019) Dynamic multi-user computation offloading for wireless powered mobile edge computing. J Netw Comput Appl 131:1–15. https://doi.org/10.1016/j.jnca.2019.01.020.

12. Bai T, Wang J, Ren Y, Hanzo L (2019) Energy-efficient computation offloading for secure uav-edge-computing systems. IEEE Trans Veh Technol 68(6):6074–6087. https://doi.org/10.1109/TVT.2019.2912227.

13. Alghamdi I, Anagnostopoulos C, Pezaros DP (2019) On the optimality of task offloading in mobile edge computing environments In: 2019 IEEE Global Communications Conference (GLOBECOM), 1–6, Hawaii. ISBN 9781728109626. https://doi.org/10.1109/GLOBECOM38437.2019.9014081.