International Journal of Scientific Engineering and Science
Volume 7, Issue 7, pp. 30-33, 2023.
ISSN (Online): 2456-7361

A Genetic Algorithm for Task Offloading of Edge Computing towards Meteorological Radar Data Computing Scenario

Shuangqing Xue, Dongping Jing
Meteorological Bureau of Linfen, Shanxi, China (86)

Abstract— *We propose a genetic algorithm for task offloading of meteorological radar data computing with Edge Computing (EC) in this paper. The increasing volume of radar data and the complexity of computing tasks make it necessary to offload the computation to remote computing resources. The proposed Randomized Greedy Algorithm (RA) and Group Genetic Algorithm are designed to determine the optimal task offloading strategy, which considers the trade-off between computation cost and communication overhead. The algorithm uses a fitness function that considers the task transmission delay, and data loss rate. The algorithm also employs crossover and mutation operations to generate new offloading strategies and improve the system's overall performance. We evaluate the performance of the proposed algorithm using simulation experiments and demonstrate that it can effectively reduce the computation offloading delay and data loss rate, which is compared with the RA task distribution.*

Keywords— *Edge computing; task offloading; fitness function; delay; data loss rate.*

I. INTRODUCTION

The size and computational requirements of meteorological radar data depend on several factors, including the type of radar, working frequency, beam width, scanning angle, and scanning speed [1-3]. Generally, meteorological radar generates large amounts of data, typically ranging from several hundred megabytes to several gigabytes per second. Real-time processing of radar data requires powerful computational capabilities to handle this data. The algorithms used to process radar data are often complex, requiring efficient computing architectures and algorithm implementations to improve processing speed. Therefore, computing tasks for meteorological radar data usually require high computational power and a large amount of storage space to process and store the data [4-6]. Due to security, integrity, and confidentiality concerns, meteorological radar data is typically not processed using public computing devices but instead uses private computing devices owned by meteorological departments to meet the computational and processing needs of radar and other data. In particular, the use of deep learning for predicting heavy rain [7], detecting snowfall [8], and thunderstorm detection [9] has become a current hot topic, posing higher challenges to computational and network transmission capabilities. Therefore, it has become an important solution for meteorological departments to migrate computing tasks to different computing devices for radar data processing in the above-mentioned deep learning applications.

However, many departments with radar detection equipment face the following issues: first, they have limited computational resources or virtual machines, which need to handle data parsing, meteorological forecasting, and multi-source data fusion calculations while facing computational limitations. Second, a large amount of real-time data is transmitted through various data transmission links, and the remaining bandwidth often changes over time. Therefore, when migrating some computing tasks for radar data, it may result in data loss and excessively high processing delays. Therefore, this paper proposes a genetic algorithm-based optimization method for small-scale computational devices with limited bandwidth and remaining computational resources in each device, to further improve the system performance of computing task offloading. This algorithm will further support the application of artificial intelligence algorithms in meteorological radar data, fully leveraging the advantages of intelligent computing. The main contributions of this paper are as follows:

1) A computing task offloading model was established for the application of deep learning in radar data, providing a reference for utilizing distributed computing power distribution on small clusters.
2) A genetic algorithm was proposed to optimize multiple servers with different computational capabilities and multiple residual bandwidths of different links, effectively reducing system latency and data loss rates.
3) A simulation scenario for computing task offloading on small-scale clusters was established, and the algorithm was compared with a random greedy algorithm in terms of convergence, latency, and data loss rates.

The organizational structure of the paper is as follows: Chapter 2 introduces the system model for radar computing task offloading; Chapter 3 presents the genetic algorithm and the random greedy algorithm; Chapter 4 establishes the experimental simulation environment and presents experimental results; and Chapter 5 provides a summary.

II. THE SYSTEM MODEL FOR RADAR COMPUTING TASK OFFLOADING

Meteorological radar data is often continuous time series data, and the data generated by the radar and various sensors in the meteorological field are transmitted to various computing devices for calculation and processing. These devices mainly perform computing tasks including meteorological forecasting and warning, meteorological scientific research, meteorological data processing, and meteorological disaster emergency response. Let ∂^i represent the amount of task data required for deep learning in the i-th forecast and warning, and the data can be divided into sets $\{\partial|\partial_1^i, \partial_2^i, \dots, \partial_n^i\}$ according to the algorithm model. The computing devices $\{\alpha|\alpha_1^i, \alpha_2^i, \dots, \alpha_n^i\}$ that have computing capabilities and can obtain various data sub-items can perform distributed computing of the corresponding data parts ∂_x^i. Here, x represents a certain data segment that is obtained. Therefore, the calculation task offloading can be represented based on the edge computing framework as shown in Fig. 1.

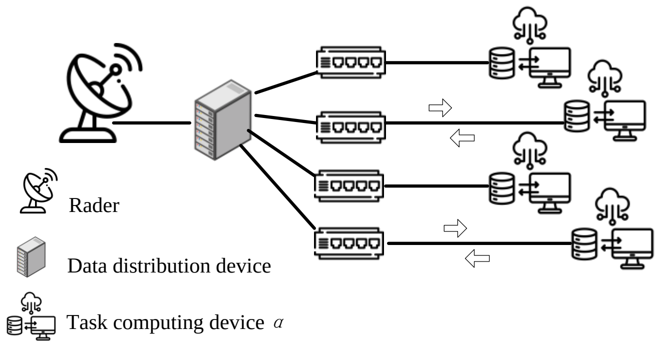

Fig. 1. The model of computing task offloading toward radar data

From the figure, it can be seen that radar data can be distributed to various nearby computing devices α_x for computation of the data portion ∂^i that applies to the nearby device, through the current receiving device (or data distribution devices). This is a typical edge computing framework. Through the α devices, the parallel computing can be effectively utilized, and the idle computing power of the nearby devices can be fully utilized. This model abandons the traditional cloud computing model for two main reasons. First, for latency-sensitive algorithms, the cloud computing model cannot meet the real-time computing requirements. Second, meteorological data cannot be transmitted over public networks due to privacy requirements. However, the decomposition problem of task data ∂^i results in varying sizes of computation tasks. Moreover, there exist various exchange devices in different paths, as shown in the figure, leading to different transmission delays δ. The transmission delay δ can be calculated by Formula 1:

$$\delta_x^i = T_x^j + f(\delta) \tag{1}$$

Where T_x^j represents the transmission delay by device j. And the f is the function for the data transmission delay. By splitting the computing tasks and assigning them to different edge computing devices, the latency can be effectively reduced, and it is also beneficial to improve the computing efficiency of each server.

III. Group Random Greedy Algorithm And Genetic Algorithm

In the radar data processing model shown in Figure 1, there are two problems to be addressed. Firstly, for each computation task fragment ∂_x^i, different computing devices α_x may have different transmission delays δ_x^i along their respective paths. Secondly, the amount of data that can be processed by each computing device α_x per unit time may be different, which can lead to data loss for some unprocessed data. Therefore, a reasonable optimization of the selection of different computing task segments ∂_x^i and computing devices α_x can to some extent reduce the delay and data loss rate in the calculation task offloading.

(a) Randomized Greedy Algorithm (RA): First, randomly select the φ objects reachable by the computing task path, denoted as $\{\alpha_i, \alpha_j, \alpha_k|\alpha_x\}$. Next, for the computation task fragment ∂_x^i, choose $min(\delta_x^i,\ D)$, where D is the data loss rate. In algorithm design, considering the real-time performance, it is assumed that data does not queue in the buffer during the arrival process waiting for computation. Therefore, the data loss rate can be mapped to formula 2:

$$D = \begin{cases} = 0 & f(D_{current}) - f(D_{piece}) \ge 0 \\ = |f(D_{curent}) - f(D_{piece})| & otherwise \end{cases} \tag{2}$$

(b) Genetic Algorithm (GA): Firstly, establish an individual with a length of L_n, where n is the number of available devices. Establish the initial population A and obtain a randomly generated matrix with a length of n through A, as shown in formula 3:

$$A=\begin{pmatrix} \omega_{00} & \cdots & \omega_{0n} \\ \vdots & \ddots & \vdots \\ \omega_{m0} & \cdots & \omega_{mn} \end{pmatrix} \tag{3}$$

Where ω represents the computing device to which the computing task fragment is transmitted. Each row represents the computing device assigned to each computing task fragment, and it represents a choice for task offloading. Therefore, m can represent the number of corresponding offloading schemes. After constructing the solution model of genetic algorithm, it is necessary to establish the corresponding fitness function to evaluate the quality of the current solution.

$$\begin{gathered} FitV = \sigma\big(\delta/max(\delta)\big) + \tau\big(D/max(D)\big) \\ min(FitV) \\ \delta \le \delta_x^i \\ D \le f(D_{curent}) \end{gathered} \tag{4}$$

Therefore, based on equation 4, the fitness value of all individuals in the matrix A can be calculated. The optimization goal is to find the minimum value of $FitV$. After sorting, the optimal value of $FitV_C$ in the current generation can be obtained, and the solution that corresponds to this optimal value is the optimal solution of the current generation. Based on the current generation, selection, crossover, and mutation operations can be performed. In the selection operation, we abandon the top-ranked θ solutions and re-copy the individuals that have not been abandoned to construct a complete A matrix. In the crossover operation, partial gene exchange is performed with a probability of ε_c to obtain a new individual, as shown in Figure 2.

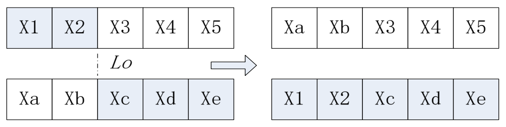

Fig. 2. The new individual solutions by crossover operation

In the mutation operation, the mutation is performed at a certain position with a probability of ε_m, as shown in Figure 3.

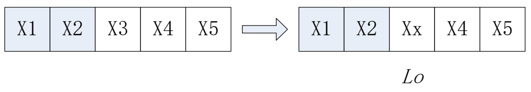

Fig. 3. A individual solution by mutation operation

After the crossover and mutation operations, a new population is obtained. By calculating its fitness value, the latest optimal value can be obtained. Through gradual iteration, the solution process approaches the local optimal solution required in the scheme, and a good decision can be obtained in a short time.

IV. Experiments And Result

For the scenario of radar data sources, we established a radar data source and ten computing devices. The calculation size of each task is designed to be 10Mbit. The tasks are divided into n parts according to a certain algorithm, and the remaining computing power of each computing device is considered to be in the range of 0-1Mbits. As it is assumed that real-time processing is required, data exceeding this range is discarded. Other key parameters are shown in Tab. 1.

Tab. 1 the key parameters

The symbol	value
m	100
ε_c	5%
ε_m	0.5%
L_n	10
θ	50
φ	3

Then we conducted experimental analysis on the convergence of the genetic algorithm, as shown in Figure 4.

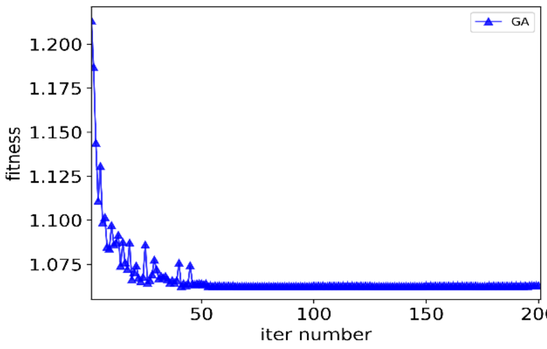

Fig. 4. The schematic diagram of algorithm convergence

As shown in Figure 1, it can be seen that with the increase of the number of iterations, the fitness value quickly converges. From the perspective of convergence and speed, the GA algorithm has good performance in dealing with the current radar data computation task offloading.

For a more intuitive comparison of the advantages of the GA algorithm, we simulated the system performance of the Group Greedy algorithm and the GA algorithm in the scenario, mainly comparing them in terms of cost (fitness), delay, and data loss rate under different numbers of computing devices.

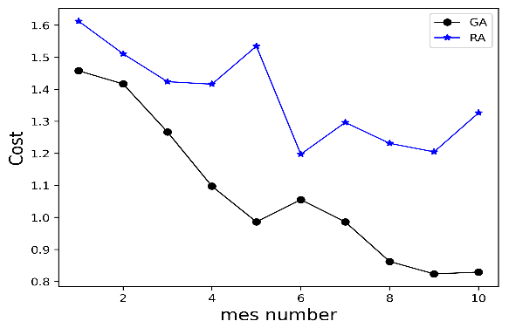

Fig. 5. The cost of simulation scenarios

Based on Figure 5, it can be observed that the overall cost of the system decreases as more computing devices are added. At the same time, the GA algorithm shows a significant advantage over the group random greedy algorithm in terms of performance.

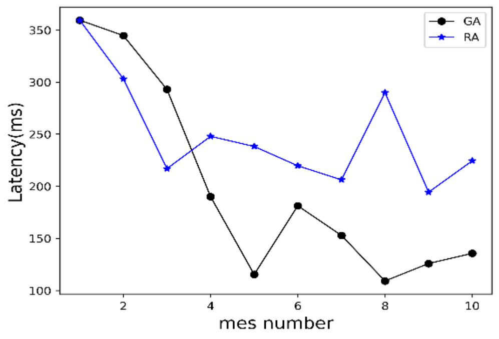

Fig. 6. The processing latency for RA and GA method

As shown in Figure 6, in the group random greedy algorithm, it has an advantage when there are few available computing devices. However, as the number of available computing devices increases, the algorithm itself exhibits some randomness. Therefore, the group random greedy algorithm can be chosen when there are fewer participating computing devices, while the GA algorithm can be chosen when there are more.

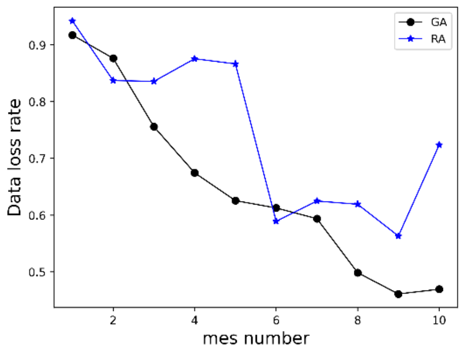

Fig. 7. The data lose rate for RA and GA

As shown in Figure 7, with the increase of the number of computing devices, the average data loss rate under the GA algorithm is lower than that under the group random greedy algorithm. The curve of the group random greedy algorithm shows significant fluctuations with the increase of the number of computing devices, which also reflects some randomness.

V. Related Work

In recent years, research on the optimization of resource allocation in edge computing has attracted much attention both domestically and internationally. In terms of the application of genetic algorithm (GA), there have been some studies that have explored its effectiveness in resource allocation optimization. For example, in [10], the authors proposed a GA-based approach for optimizing the energy consumption of a cloud-edge system in the context of mobile edge computing. In [11], the authors used GA to optimize the placement of virtual machines in a cloud-edge environment.

In terms of the application of greedy algorithms in edge computing, there have also been some studies. For example, in [12], the authors proposed a greedy algorithm for dynamic resource allocation in a cloud-edge environment. In [13], the authors proposed a greedy algorithm for load balancing in edge computing.

Overall, the effectiveness of GA and greedy algorithms in resource allocation optimization in edge computing has been demonstrated in various studies. However, there is still room for further research, particularly in the optimization of task partitioning and allocation. From the perspective of computing task offloading, on the one hand, they did not focus on radar data, so the proposed algorithm model is not suitable for the practical scenario of offloading radar data computing tasks. On the other hand, the modeled data flow and device quantity do not match the situation of a large number of small and medium-sized radar stations, thus lacking reference value. Therefore, the method proposed in this paper solves the algorithm operability of the scenario to some extent from the perspective of practical needs.

VI. Conclusions

In the context of radar devices, the use of cloud-based intelligent computing has certain limitations. Therefore, in this context, we propose the group random greedy algorithm and genetic algorithm based on the edge computing framework to solve the selection problem of radar data when multiple computing devices are available, in order to reduce system latency and data loss rate. In the simulation after modeling, it was found that the GA algorithm has an advantage in solving the problem in this scenario. When there are fewer computing devices, the group random greedy algorithm can achieve good performance with lower algorithm complexity. The next step of research work will focus on the division of radar computing tasks.

Acknowledgment

We would like to express our heartfelt gratitude to all those who have contributed to this research. Firstly, we would like to thank our supervisor for his/her valuable guidance and insightful comments throughout the entire research process. We also extend our sincere appreciation to all the participants who have generously devoted their time and effort to provide us with valuable data and insights. Finally, we would like to thank our families and friends for their unwavering support and encouragement. This research would not have been possible without their support. Also, we thank the lab funding from "Meteorological Information and Signal Processing Key Laboratory of Sichuan Higher Education Institutes of Chengdu University of Information Technology"; "Integrated Computing and Chip Security Sichuan Collaborative Innovation Center of Chengdu University of Information Technology", (NO. CXHCL202204).

Reference

[1] Chen, X., Yu, Y., & Gao, J. (2020). Design and implementation of the 915 MHz wind profiler radar system. Journal of Atmospheric and Solar-Terrestrial Physics, 201, 105225.
[2] Kuang, S., Liu, L., & Huang, Y. (2019). An efficient and accurate approach to estimating wind speeds and directions from dense radar networks. Remote Sensing, 11(8), 970.
[3] Rasti, M., & Tabatabaeenejad, A. (2019). Reconfigurable antenna design for ground-based radar systems: A review. Journal of Electromagnetic Waves and Applications, 33(8), 946-963.
[4] Li, Y., Zhang, Y., Cao, L., & Cui, Y. (2021). Design and implementation of a parallel computing system for weather radar data processing. Journal of Parallel and Distributed Computing, 151, 77-87.
[5] Yang, Z., Lu, X., & Li, X. (2019). Cloud-based weather radar data processing platform. Journal of Cloud Computing, 8(1), 1-12.
[6] Li, J., Li, R., Li, J., & Cao, J. (2018). Performance evaluation of an HPC cluster for weather radar data processing. International Journal of High Performance Computing and Networking, 12(1), 16-26.
[7] Li, H., Wang, L., & Gao, S. (2019). Rainstorm prediction based on deep learning and weather radar data. Journal of Physics: Conference Series, 1168(5), 052052.
[8] Hu, L., Qiao, Y., & Liu, D. (2021). Snowfall detection from weather radar data using convolutional neural network. Journal of Hydrology, 595, 126055.
[9] Xu, Y., Cui, L., & Li, Y. (2018). Convolutional neural networks for lightning detection from radar data. IEEE Transactions on Geoscience and Remote Sensing, 57(10), 7664-7677.
[10] T. Zhang, J. Zhou, Y. Zhang, et al., "Optimizing energy consumption of cloud-edge mobile system based on genetic algorithm," IEEE Access, vol. 8, pp. 195446-195458, 2020.
[11] X. Li, H. Li, and X. Li, "Virtual machine placement optimization in cloud-edge computing using genetic algorithm," Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 7657-7665, 2021.
[12] S. Liu, H. Song, Y. Xu, et al., "A dynamic resource allocation algorithm in cloud-edge environment based on the greedy algorithm," Journal of Ambient Intelligence and Humanized Computing, vol. 11, pp. 5329-5337, 2020.
[13] K. Wang, Y. Sun, C. Qiao, et al., "Load balancing algorithm in edge computing based on improved greedy algorithm," Journal of Intelligent and Fuzzy Systems, vol. 38, pp. 2457-2466, 2020.