

An Edge Computing Offload Method Based on NSGA-II for Power Internet of Things

Yue Ma¹, Xin Li¹, Jianbin Li²

¹State Grid Jibei Information & Telecommunication Company, Beijing, China

²School of Control and Computer Engineering, North China Electric Power University, Beijing, China

Email address:

lijb87@ncepu.edu.cn (Jianbin Li)

To cite this article:

Yue Ma, Xin Li, Jianbin Li. An Edge Computing Offload Method Based on NSGA-II for Power Internet of Things. *Internet of Things and Cloud Computing*. Vol. 9, No. 1, 2021, pp. 1-9. doi: 10.11648/j.iotcc.20210901.11

Received: March 4, 2021; **Accepted:** March 15, 2021; **Published:** March 22, 2021

Abstract: In the ubiquitous power Internet of things, all kinds of growing power terminal equipment and business applications will generate massive data, which will cause huge pressure to the master station, and high delay and security cannot meet the requirements of new business forms. Edge computing organically integrates computing, storage, and other resources on the edge of the network and responds to the task request of the network edge node timely and effectively according to the principle of nearest service. Due to the limited resources of edge nodes, such as power monitoring camera capability, resources, bandwidth, energy, etc., computing offload is a key problem of edge computing. To solve this problem, this paper proposes a method of edge computing offload based on genetic algorithm. Firstly, in the edge-computing scenario of the power Internet of things, we analyze the computing unloading problem model under the time sequence condition. Then, aiming at the optimal decision-making problem of energy consumption and time delay of terminal equipment, we creatively transform the problem of computational offload into the problem of multi-objective optimization. In the genetic algorithm, we use NSGA-II to achieve the multi-objective optimization of the decision-making. Through conversion, time delay and energy consumption, the optimization can be achieved. Finally, we designed a simulation experiment. The results show that the unloading decision of NSGA-II can reach the best. The results show that the results of NSGA-II can be distributed in a wider range.

Keywords: Edge Computing Offload, NSGA-II, Power Internet of Things

1. Introduction

In 2019, State Grid Co., Ltd. put forward a new goal of “three types and two networks”. The goal of power Internet of things construction has emerged, and the prelude of related construction has officially opened [10]. In the architecture of ubiquitous power Internet of things, the sensing terminal is responsible for state data collection and the edge computing is responsible for real-time analysis of state data [16]. With the rapid increase of the number of edge terminal devices and computing tasks, the traditional cloud computing performance appears bottleneck, which leads to the decrease of computing speed and the increase of delay and jitter [13, 28, 31]. While edge computing provides services to the nearest side, which reduces the pressure of the main station by putting a lot of computing to the execution end, meets the requirements of real-time data analysis and processing and

low delay business, reduces the operation and maintenance cost, and improves the system efficiency [14, 20, 25]. For example, for the real-time monitoring of transmission lines, many transmission lines are in remote areas, the network environment is poor, and it is difficult to real-time monitor manually. Holographic sensing equipment analyzes the data of sensors, surveillance cameras and other data directly to identify potential threats to the security of the power grid, including people, vehicles, fire, and electric shock. Through edge computing, only the information that needs to be early-warning is uploaded to the cloud, thus greatly reducing the cost of communication bandwidth and computing delay [2, 8, 12]. In edge computing, computing offload technology offloads the computing tasks of the terminal to the edge network, which solves the shortcomings of the devices in resource storage, computing performance and energy efficiency [27].

At the same time, compared with computing offload in cloud computing, edge computing offload solves the problems of network resource occupation, high delay, and extra network load [3, 7, 11]. The first step of unloading decision is to calculate whether the task should be unloaded or not. The key is to compare the cost of local processing task and edge server processing task, and then judge whether unloading is valuable. A poor offload decision may result in multiple processing delays and energy consumption [1, 18, 22, 23]. Therefore, how to design and calculate unloading scheme reasonably is a key problem of edge computing [26]. Energy and time delay are the key design goals of edge computing. Achieving one uninstall goal may affect the other. For example, executing a task on an edge server may reduce the response time of the task, but it may not save the battery power of the terminal device [17, 24, 29]. It is a nontrivial multi-objective optimization problem to study the trade-off between energy consumption and response time [30].

Common optimization algorithms include ant colony algorithm, particle swarm algorithm, genetic algorithm, etc [5, 6, 9]. Among many algorithms, genetic algorithm is very popular because of its great search ability and convergence speed. Multi-objective genetic algorithm is an evolutionary algorithm used to analyze and solve multi-objective optimization problems. Its core is to coordinate the relationship between various objective functions and find out the optimal solution set that makes each objective function reach a larger (or smaller) function value as much as possible [15, 19, 21]. NSGA-II algorithm is the most influential and widely used multi-objective genetic algorithm in multi-objective optimization [4].

Therefore, considering the need to adapt to different application scenarios, communication conditions and the computing resources of the base station, the decision-making of computing offload needs to be integrated. In this paper, an edge computing offload method based on NSGA-II is proposed for effective computing offload and resource allocation, to balance the time delay and energy consumption of task execution. The research goal of this paper is to reduce the energy consumption and time cost of the whole system as much as possible while considering that the edge server shares the huge task of cloud computing.

The structure of this paper is as follows. In the second section, the edge computing unloading method of power Internet of things based on NSGA-II is proposed. The third section introduces the experimental simulation and the comparison results. At last, we give the concluding remarks.

2. An Edge Computation Offload Method Based on Nsga-II

A. System modeling

1) *Network modeling*: Consider the scenario of multiple terminal devices in single plant area. The base station is deployed with a high-performance server, which can provide computing services for multiple terminal devices in the plant

area. The set of terminal devices is expressed as $N = \{1, 2, \dots, N\}$. Suppose that each terminal device has a computing intensive task to complete, which can be executed locally or uninstalled to the base station server. In this case, we only consider the two-state offloading of computing tasks, that is all tasks are intelligently selected to be offloaded to the base station server or executed locally. The so-called two-state unloading refers to the offloading decision is described as 0 and 1. 0 means execute locally; 1 means offload tasks to edge computing server.

We use time division multiplexing to share wireless links. Time division multiplexing can reduce the total delay of all tasks. In this way, a time frame is divided into multiple time slices on average for communication between terminal equipment and base station. Therefore, only one task in a time slice can be unloaded to the edge computing server for execution, which reduces the transmission delay of single task and reduces the waiting time of multiple tasks competing for computing resources on the server side.

If device n chooses to offload the task, the data transmission rate of the terminal device is

$$r_n = B \log_2 \left(1 + \frac{g_n P_n}{B N_0} \right) \quad (1)$$

where B is the bandwidth of the wireless channel, P_n represents the transmission power of the device n for uploading data, g_n represents the channel gain between the device n and the base station, N_0 is the variance of white Gaussian noise.

2) *Task modeling*: Consider that each device has a compute intensive task to calculate. The task corresponding to each terminal device n corresponds to a quadruple, $T_n \triangleq (v_n, s_n, a_n, t_n^{off})$. n is the number of devices and t_n^{off} is the upload time. Where v_n represents the amount of data for the task to be performed, including input parameters and program code. s_n represents the amount of computation, that is, the number of CPU execution rounds required to calculate the task. Because the computing power of the edge computing server is different from that of the device, the running time of the same task is also different. If tasks arrive randomly, the arrival time of each task is a_n . t_n^{off} indicates the task upload time. The information of these four parameters related to tasks can be estimated by task analyzer.

When the device has a task to execute, it needs to first send a request to the base station, which contains all the information of the task; then wait for the base station to decide to offload. Tasks that have arrived but have not been executed will wait for the scheduling of the base station. Since only one task can be executed at the same time on the server at the base station, there is a queue to store the tasks waiting to be executed. The uninstall decisions defined for device n are:

$$\alpha \in \{0, 1\} \quad (2)$$

If the base station decides to offload the task to the edge computing server, the base station will set $\alpha = 0$, otherwise $\alpha = 1$.

3) *Computing modeling*: Because the task can be executed at the base station end, it can also be executed at the device end. Therefore, the calculation model is divided into two parts: local computing model and offloading computing model. The local computing model represents the use of the computing resources of the terminal device itself to perform tasks. The offloading computing model represents that the base station decides to offload the task to the edge computing server and uses the computing resources of the server to process the task.

(1) Local computing model

If the base station decides to execute the computing task locally, define T_n as the delay of local computing, that is, the processing time of the local CPU. If the computing power of the device n is f_n , and the local execution delay is

$$T_n^l = \frac{S_n}{f_n} \quad (3)$$

Defined E_n as the energy consumption in the local execution process, that is, the energy of the local computing. So,

$$E_n^l = z_n (f_n^l)^2 D_n \quad (4)$$

z_n is the energy consumption density, and the parameter difference between different equipment is very small.

(2) Offloading computing model

If the edge computing server is selected to perform the task, the device will first uninstall the task to the edge computing server through the wireless channel, and then the edge computing server will perform the task. The whole unloading process consists of three parts. First, the device needs to upload the program code and parameter related information to the base station through the wireless channel, and then the base station forwards the relevant information to the edge computing server.

Then, the edge computing server needs to allocate corresponding computing resources to perform tasks. Finally, the base station returns the computing results to the device.

Therefore, the delay of offloading processing is divided into three parts. The first part is the transmission delay caused by the input data v_n . The data transmission delay depends on the available offloading rate r_n .

$$T_n^l = \frac{v_n}{r_n} \quad (5)$$

The energy consumption of the equipment in this process is:

$$E_{n,t}^o = P_n T_{n,t}^o = \frac{(P_n B_n)}{r_n} \quad (6)$$

Any time slice after sending task related information to the base station may offload task data to the base station server.

Therefore, the time when the device uploads data through the wireless channel is recorded as t_n^{off} . Since only one task can be executed on the edge computing server at the same time, other tasks may be scheduled on the local device and offloaded to the edge computing server at a certain time slice in the future, or they may have been offloaded to the edge computing server for execution. Considering that there are usually multiple tasks waiting to be executed on the edge computing server, there is a first in first out queue in the server to store the offloaded tasks. Once the executing task is finished, the server will get the next task from the queue. Define the time when the task starts executing at the edge calculation server as t_n^{exe} , so

$$t_n^{exe} \geq t_n^{off} + T_n^l \quad (7)$$

The second part of the execution delay is the processing delay of the edge computing server s_n . When the edge computing server executes a task, assuming it occupies all computing resources, the execution delay T_n^e is

$$T_n^e = \frac{s_n}{f^c} \quad (8)$$

f^c represents the computing resources of the edge computing server.

During the period when the edge computing server performs the tasks, the device is in the state of waiting for receiving. Define the power of the device in this state as P_n^i , then the energy consumed by the device in this period is:

$$E_{n,p}^o = P_n^i T_{n,p}^o = \frac{P_n^i D_n}{f_n} \quad (9)$$

The third part of the offload processing delay is the return delay. The prior researches show that the amount of data in the calculation results is very small compared with the amount of data offloaded. Therefore, the return delay is ignored in this paper. The total delay in the calculation of the unloading is

$$T_n = \sum (1-\alpha) T_n^l + \alpha T_n^e \quad (10)$$

and the total energy consumption of unloading is

$$E_n = \sum_n (1-\alpha) E_n^l + \alpha E_n^e \quad (11)$$

B. Problem modeling

In this paper, we consider both time delay and energy consumption, and try to find the optimal solution set which can make the two objective functions of delay and energy consumption reach a relatively small value as much as possible. The device selects one of the computing ways of local computing or offloading computing. The problem model is as follows:

$$\begin{aligned}
T &= \min \sum_n (1-\alpha)T_n^l + \alpha T_n^e \\
E_n &= \min \sum_n (1-\alpha)E_n^l + \alpha E_n^e \\
s.t. \text{ C1: } \alpha_n &\in \{0,1\}, \forall n \\
\text{C2: } t_n^{exe} &\geq t_n^{off} + T_n^t
\end{aligned} \tag{12}$$

Constraint C1 guarantees that each task can only be processed by local computing or offloading to edge computing server. Constraint C2 means that the task needs to be offloaded to edge computing server first before it can be executed.

C. Multi-objective genetic algorithm

Genetic algorithm is a computational model simulating the natural selection and genetic mechanism of Darwinian biological evolution. It is a method to search the optimal solution by simulating the natural evolution. Its main characteristics are direct operation on the structure object, without the limitation of derivation and function continuity; inherent implicit parallelism and better global optimization ability; using probabilistic optimization method, the search space can be obtained and guided automatically without certain rules, and the search direction can be adjusted adaptively. Genetic algorithm takes all individuals in a population as objects and uses randomization technology to guide the efficient search of a coded parameter space. Among them, selection, crossover, and mutation constitute the genetic operation of genetic algorithm; parameter coding, setting of initial population, design of fitness function, design of genetic operation and setting of control parameters constitute the core content of genetic algorithm.

Multi-objective genetic algorithm is an evolutionary algorithm used to analyze and solve multi-objective optimization problems. Its core is to coordinate the relationship between various objective functions and find out the optimal solution set that makes each objective function reach a larger (or smaller) function value as much as possible. Among many multi-objective optimization genetic algorithms, NSGA-II algorithm is a multi-objective optimization algorithm based on Pareto optimal solution, which is the most influential and widely used multi-objective genetic algorithm. After its appearance, the algorithm has become one of the basic algorithms in multi-objective optimization problems because of its simple, effective, and obvious advantages. The main advantages (improvements) of the algorithm are as follows [4]:

1) A fast-non-dominated sorting algorithm is proposed, which reduces the complexity of calculating the non-dominated order and reduces the complexity of the optimization algorithm from the original mN^3 to mN^2 . m is the number of objective functions and N is the size of the population.

2) The elite strategy is introduced to expand the sampling space. The combination of the parent population and the offspring population will generate the next generation population through competition, which is conducive to the

maintenance of the excellent individuals in the parent generation, to ensure that those excellent individuals are not discarded in the evolution process, to improve the accuracy of the optimization results. And by storing all individuals in layers, the best individuals will not be lost, and the population level can be rapidly improved.

3) Introduce crowding degree and crowding degree comparison operator, which not only overcomes the defect of NSGA that needs to specify the shared parameters artificially, but also takes crowding degree as the comparison criterion between individuals in the population, so that the population individuals in the quasi Pareto domain can be evenly extended to the whole Pareto domain, thus ensuring the diversity of the population.

Because the problem proposed in this paper is a NP-hard problem, the complexity of solving the optimal solution is high. Considering the idea of multi-objective optimization rather than single objective optimization, the efficiency of using NSGA-II method to get the optimal solution is high. In the multi-objective optimization problem, it is usually necessary to measure the impact of each objective variable on the overall objective, but when considering the multi-objective method, NSGA-II method not only uses the proportion or weight of each objective to measure, the measurement standard is more balanced and comprehensive.

The population is defined as P , P is made up of $\text{Individual} = \{C1, C2, \dots, Cn\}, C \in \{0,1\}$, where C represents the decision of offloading by an edge device. It mainly includes initialization, model and parameter setting, genetic algorithm solving and iteration. The code of algorithm design is shown as follows. The core of the algorithm is NSGA-II. The flow chart of NSGA-II is shown as Figure 1.

```

P = [random(ind) for i in range (pop-size) ]
while generations < max generations:
    for ind not evaluated in P :
        evaluate(ind)
    mated_offspring = mate(P, CXPB)
    mutated_offspring = mute(P, CXPB)
    mutated_offspring = mutate (mated, offspring,
                                MUTPB)
    offspring = mutated_offspring
    P = select_NSII(P + offspring )
return P

```

3. Simulation Experiment

To verify that the proposed method based on NSGA-II can be effectively applied in the power IOT, the following simulation experiments are done. The simulation experiment of this paper uses Python platform to train the genetic algorithm model. The results show that the unloading decision of NSGA-II can reach the optimum. The results show that the results of NSGA-II can be distributed in a wider range.

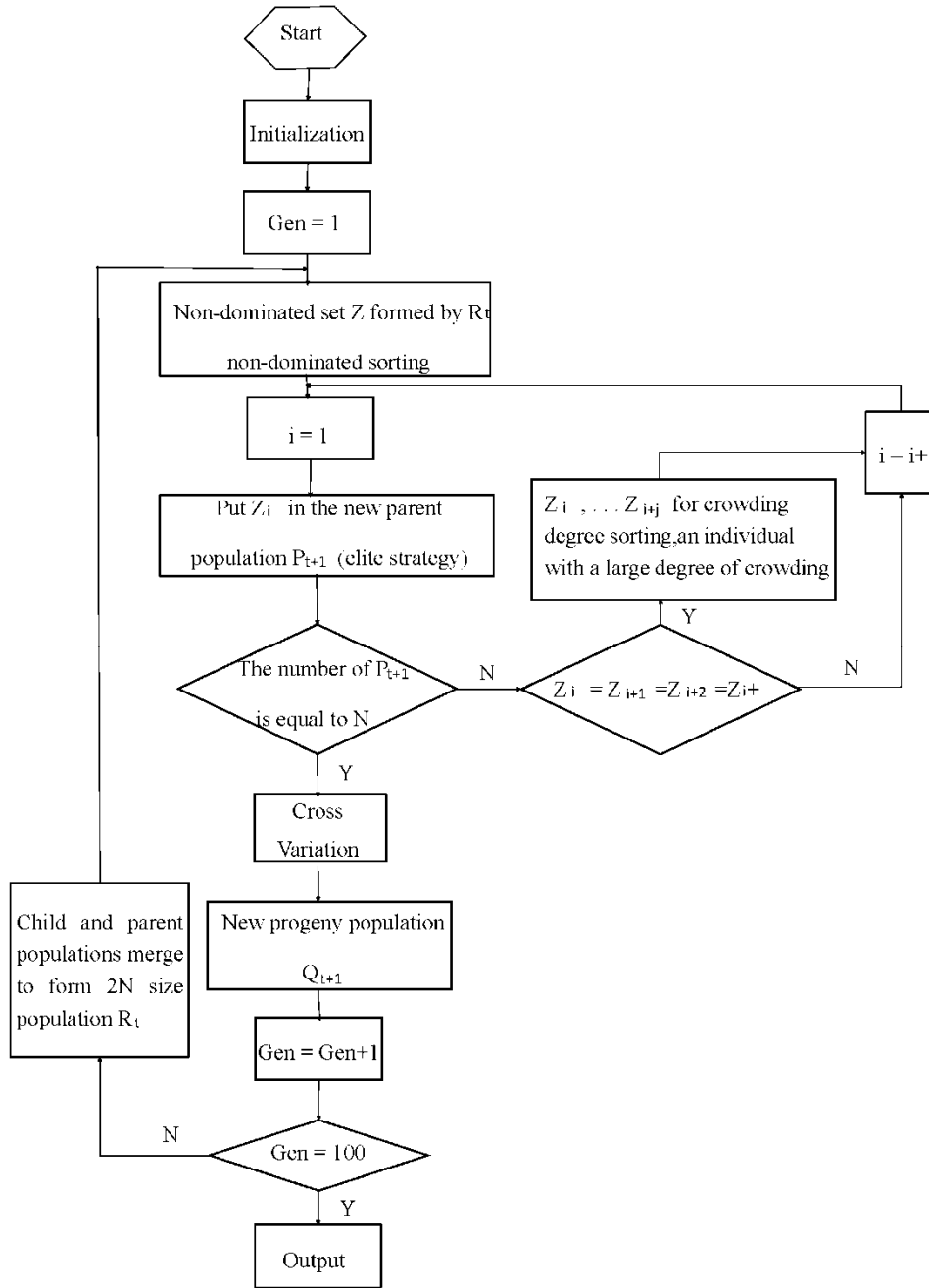


Figure 1. The flow chart of NSGA-II.

3.1. Simulation Settings

Suppose that there are n edge nodes, one base station and one edge computing server in the wireless network scenario of a single plant area. The topology is as shown in the figure. In this experiment, we assume that each task has only two kinds of decision-making. The calculation tasks are performed locally and unloaded to the edge calculation server, which are represented by 0 and 1 respectively. Each node has limited computing power but can compute its own tasks. The edge computing server can perform computing tasks, and it is assumed that its performance can be as large as possible, which is larger than the calculation amount of

any node but smaller than the total calculation capacity of all nodes. Base station adopts centralized edge computing unloading strategy, which plays a scheduling role in edge computing. This paper assumes that the computing power is large enough, the scheduling queue is small, and the scheduling time can be ignored. As a decision-maker, the base station decides the calculation execution right of the node. When calculating the time delay, because the return time delay is very small, it can be ignored. The parameters set in the experiment are shown in the following table.

3.2. Simulation Results

In this experiment, the standard deviation of individual

fitness in the population of genetic algorithm changes with the population evolution algebra as shown in the figure, and the individual fitness includes time delay and energy consumption. The blue dotted line indicates that the standard deviation of time delay changes with the population evolution algebra, and the red solid line indicates that the standard deviation of energy consumption changes with the population evolution algebra. In the early stage of population evolution, the standard deviation of time delay and energy consumption fluctuated greatly and became stable after 35 generations. The evolution algebra selected in this experiment is 500 generations.

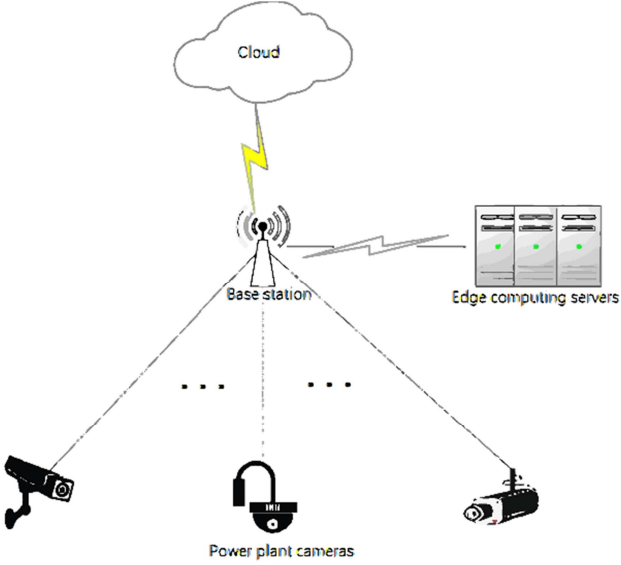


Figure 2. Network topology.

The decision to calculate the offload decides the execution of the edge device. The results of NSGA-II (red dot set), random method (black dot set), all unloading (blue dot set) and all local (green dot) of GA are comprehensively considered, as shown in the figure. The random method represents the result of random selection of unloading decision of an edge device. The closer to the origin point is the better result (considering the best situation of overall unloading considering the delay and energy consumption), and the closer to the upper right corner is the worse result. Through the analysis, we can see that the random results, all the offloading results and all the local results are all in the upper right of the results of using GA, therefore the results are the best.

If the results of NSGA-II and random method are considered, see the figure below. The red point set represents the decision result of NSGA-II, and the black point set represents the decision result of random method. According to the above rules, the black point sets are all in the upper right of the red point set, which means that the result of NSGA-II algorithm is better than that of random method.

Consider introducing a ray from the origin in the first quadrant, which starts from the origin and passes through the red point set and the black point set, as shown in the following figure. The graph is equivalent to the figure above,

but the representation method is different. As a result, the closer to the origin (that is, the closer to the left) the better the result. If the time series passing through the point set is used, the earlier the ray passes through the point, the better the result will be. Through the analysis, it can be concluded that the red points are all on the lower left side of the black point set, which is closer to the origin or the ray passes through earlier than the black point set, and the result of the red point set is better.

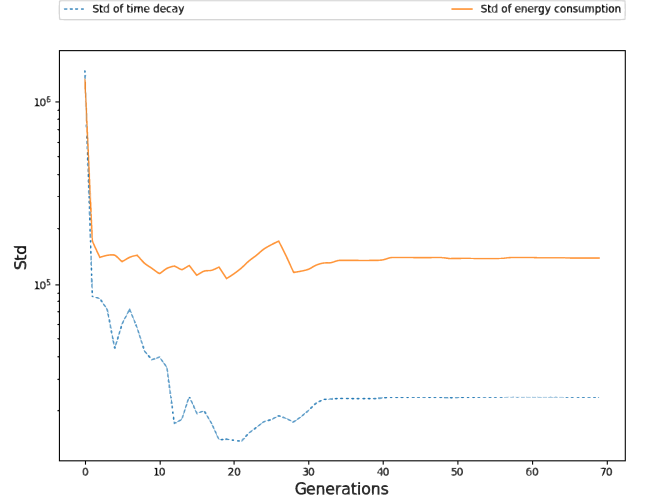


Figure 3. The standard deviation of individual fitness changes with population evolution algebra. The blue dotted line and the red solid line represent the standard deviation of time delay and energy consumption as the population evolution algebra changes. After 35 generations, the standard deviation of individual fitness tends to be stable.

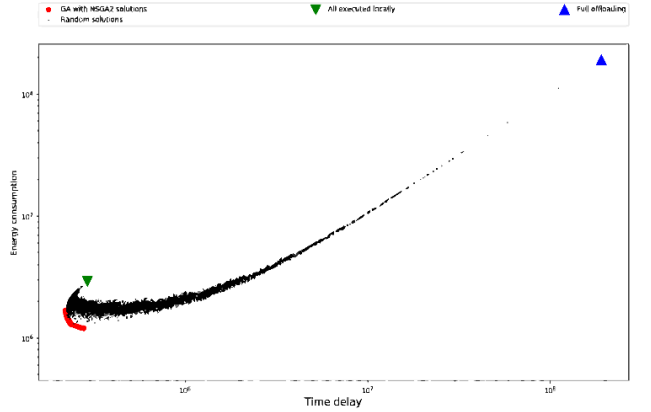


Figure 4. Comparison of NSGA method, random method, all offloading, and all local.

In this experiment, we consider using NSGA-II and tournament algorithm to get the decision results of time delay and energy consumption in the selection process. The results are compared as shown in the figure below. The red dot indicates that the NSGA-II selection method is used in the selection process, and the green dot indicates that the Tournament method is used. The results are presented in the form of dots and distributed in the region discretely. Through the analysis, when closer to the time delay has a greater impact on the decision results, the results of NSGA-II are

distributed in a wider range, and closer to the side of energy consumption, which means that the use of NSGA-II in the selection process can make us get a wider range of optimal solutions. In the same way, similar analysis results appear in

the case that the closer energy consumption has a greater impact on the decision results. This characteristic reflects that NSGA-II considers the comprehensive influence of multi-objective, rather than one or several of them dominating.

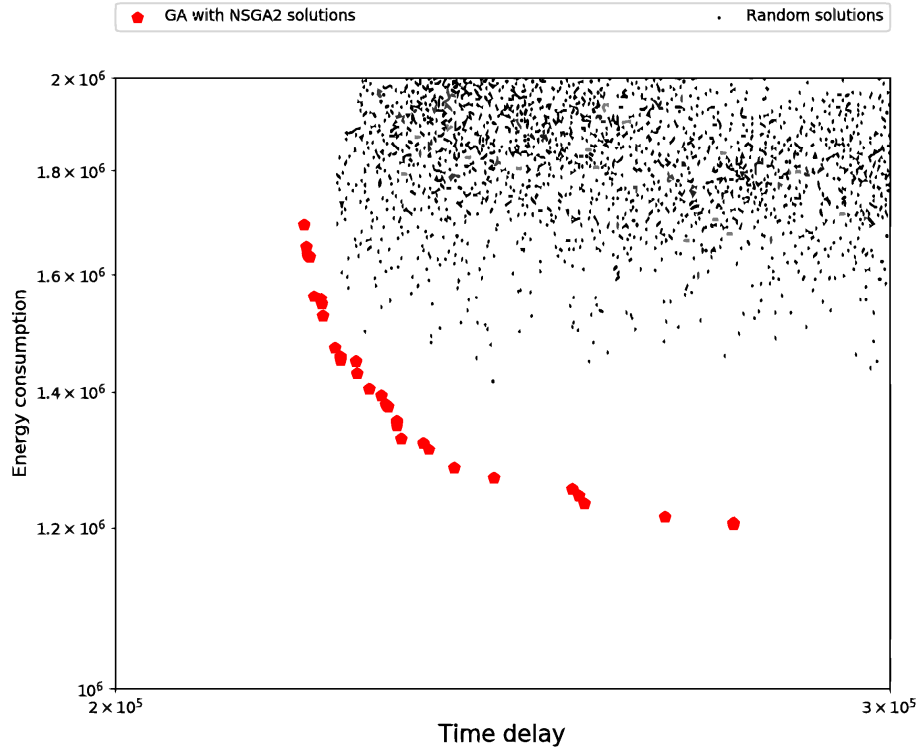


Figure 5. Comparison of NSGA method and random method.

To sum up, the experiments show that, compared with the multi-objective optimization method of stochastic method, the unloading decision of NSGA-II can achieve the best. The results of NSGA-II can be distributed in a wider range than that of common tournament selection methods.

Table 1. Parameter setting.

Parameter	Range of value	Parameter	Range of value
n	(5,100)	$t_n^{off} - a_n$	(0,10s)
V_n	(300,500)	f_n	5GHz/s
s_n	(500,1000)	P_n	100MW
a_n	(0,60s)	P'_n	100MW
B	20MHz	g_n	10dB
n_0	-88dB	$I_n(f'_n)^2$	1MJ/kg(L)

4. Conclusion

In this paper, we propose an edge computing offload method of power Internet of things based on multi-objective genetic algorithm NSGA-II. First, in the edge computing scenario of simple power Internet of things, we analyze the computing offload problem model under the condition of time division multiplexing. Then, aiming at the target of energy consumption and time delay of terminal equipment, we use NSGA-II to make the decision of edge computing offloading. This method mainly considers multi-objective optimization to carry out effective computing offloading and

resource allocation, as well as energy consumption and time delay of terminal equipment. Finally, we performed a simple simulation experiment. The results of single objective optimization and multi-objective optimization are compared. The results show that the optimal unloading can be achieved by considering both energy consumption and time delay. Advantage comparison is shown in Table 2.

But there are still some problems in this paper. First of all, choosing 500 generations in the genetic algebra setting of genetic algorithm wastes computing resources to a certain extent. Then, the multi-objective optimization problem of genetic algorithm is essentially a search problem without self learning ability, which is also needed in the target optimization problem of computational offload. Finally, the setting of the experimental scene is relatively simple, without considering the more complex scene.

Table 2. Advantage comparison.

Items	Artificial intelligence	Time sequence	Multi-objective optimization
Wang [26]	×	×	×
Zhang [29]	×	✓	✓
May [19]	✓	×	✓
Zhang [30]	✓	✓	×
Cai [2]	✓	✓	×
Our	✓	✓	✓

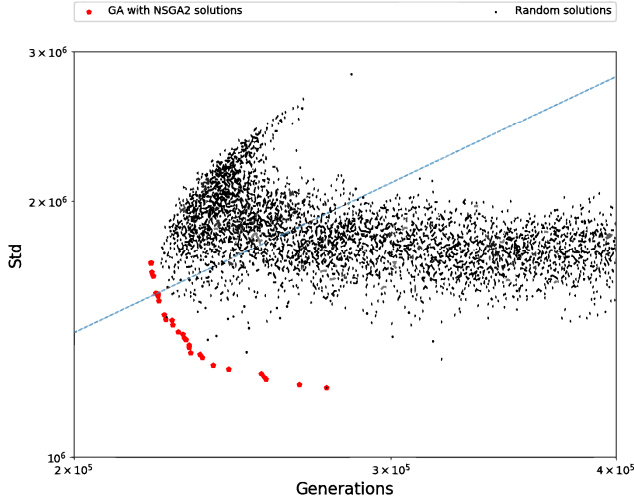


Figure 6. Time series diagram of ray.

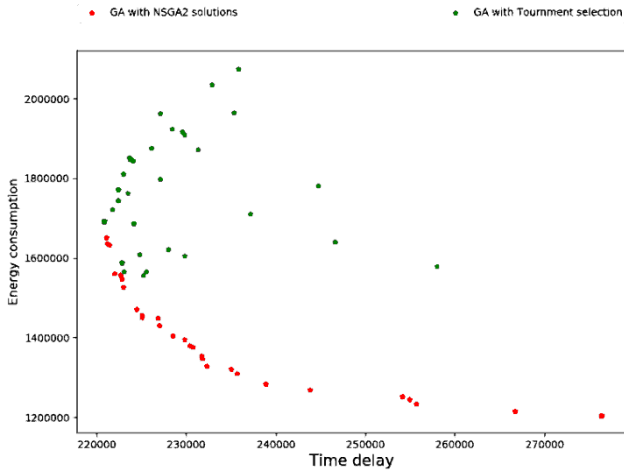


Figure 7. Comparison of decision results of time delay and energy consumption obtained by NSGA-II and tournament algorithm in selection process.

Acknowledgements

This work is supported by the Science and technology project support of State Grid Corporation of China (Contract No.: “5700-201914241A-0-0-00” and “Z181100005118016”). Jianbin Li is the corresponding author of this work.

References

- [1] A Ahmed and E Ahmed. A survey on mobile edge computing. 2016 10th International Conference on Intelligent Systems and Control (ISCO), 2016.
- [2] Y M Cai, S Y Feng, H W Du, M X Liu, X H Ding, W L Ji. Novel Edge-ware Adaptive Data Processing Method for the Ubiquitous Electric Power Internet of Things. High Voltage Engineering, 2019.
- [3] E Cuervo, A Balasubramanian, D Cho, et al. MAUI: making smartphones last longer with code offload. Proceedings of the 8th international conference on Mobile systems, applications, and services. ACM, 2010.
- [4] K Deb, A Pratap, S Agarwal, et al. A fast and elitist multi objective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 2002, 6 (2): 182-197.
- [5] M Dorigo, V Maniezzo, A Colomi. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, man, and cybernetics, Part B: Cybernetics, 1996, 26 (1): 29-41.
- [6] R Eberhart and J Kennedy. A new optimizer using particle swarm theory. MHS’95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Ieee, 1995: 39-43.
- [7] H Flores, P Hui, S Tarkoma, et al. Mobile code offloading: from concept to practice and beyond. IEEE Communications Magazine, 2015, 53 (3): 80-88.
- [8] S Hagen and A Kemper. Model-based planning for state related changes to infrastructure and software as a service instances in large data centers. Proceedings of 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD). Miami, USA: IEEE, 2010.
- [9] J H Holland. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992.
- [10] K He. The Key Technologies of IOT with Development and Applications. Radio frequency world, 2010, (1): 33-35.
- [11] L Jiao, R Friedman, X Fu, et al. Cloud-based computation offloading for mobile devices: State of the art, challenges and opportunities. Future Network and Mobile Summit, 2013.
- [12] X C Jiang, Y D Liu, X F Fu, P Xu, S Q Wang, G H Sheng. Construction Ideas and Development Trends of Transmission and Distribution Equipment of the Ubiquitous Power Internet of Things. High Voltage Engineering, 2019.
- [13] J Q Li. National Grid Corporation fully deploys ubiquitous power Internet of Things recommendations. 2019.
- [14] F M Liu, P Shu, H Jin, et al. Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications. IEEE Wireless Communications, 2013.
- [15] Q Long, C Wu, T Huang, et al. A genetic algorithm for unconstrained multi-objective optimization. Swarm and Evolutionary Computation, 2015, 22: 1-14.
- [16] Y J Lv. The origin and development trend of the Internet of Things. Information and communication technology, 2010, 4 (2): 4-8.
- [17] P Mach and Z Becvar. Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys and Tutorials, 2017, 19 (3): 1628-1656.
- [18] Y Mao, C You, J Zhang, et al. A survey on mobile edge computing: the communication perspective. IEEE Communications Surveys and Tutorials, 2017, PP (99): 1-1.
- [19] G May, B Stahl, M Taisch, et al. Multi-objective genetic algorithm for energy-efficient job shop scheduling. International Journal of Production Research, 2015, 53 (23): 7071-7089.
- [20] M Mukherjee, L Shu, D Wang. Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges. IEEE Communications Surveys and Tutorials, 2018.

- [21] T Murata and H Ishibuchi. MOGA: Multi-objective genetic algorithms. IEEE international conference on evolutionary computation, 1995, 1: 289-294.
- [22] J H Park, K H Lee, G J Jin, et al. Loading/unloading decision system of ship block in the shipyard. Journal of the Institute of Electronics Engineers of Korea CI, 2010, 47 (6): 40-46.
- [23] W Y Poe and J B Schmitt. Node deployment in large wireless sensor networks: coverage, energy consumption, and worst-case delay. Asian Internet Engineering Conference. ACM, 2009: 77-84.
- [24] W Shi, J Cao, Q Zhang, et al. Edge computing: Vision and challenges. IEEE Internet of Things Journal, 2016, 3 (5): 637-646.
- [25] State Grid Internet Department. Ubiquitous power Internet of Things construction overall plan. 2019.
- [26] C Wang, F R Yu, C Liang, et al. Joint computation offloading and interference management in wireless cellular networks with mobile edge computing. IEEE Transactions on Vehicular Technology, 2017, 66 (8): 7432-7445.
- [27] R C Xie, X F Lian, Q M Jia, T Huang, Y J Liu. Survey on computation offloading in mobile edge computing. Journal on Communications, 2018.
- [28] X Yang. Building a ubiquitous power Internet of Things to meet more individual needs. State Grid News, 2019.
- [29] K Zhang, Y Mao, S Leng, et al. Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. IEEE access, 2016, 4: 5896-5907.
- [30] J Zhang, X Hu, Z Ning, et al. Energy-latency trade on for energy-aware offloading in mobile edge computing networks. IEEE Internet of Things Journal, 2017, 5 (4): 2633-2645.
- [31] Y Zhen, X Z Li, H Q Ou, L K Zeng. Internet of Things and Smart Grid. Engineering Applications, 2012.

Biography

Yue Ma is the deputy general manager of State Grid Hebei Communication Company.

Xin Li is the deputy director of data technology center of State Grid Jibei communication company.

Jianbin Li worked in the electronic tax management center of the State Administration of Taxation from July 1995 to January 2015 as the director of the information security management department; he worked in Central South University from January 2015 to December 2017 as the first Dean of the Institute of information security and big data, director of the collaborative innovation center of medical big data 2011 of Hunan University, specially appointed professor and doctoral supervisor; since January 2018, Professor, doctoral supervisor, member of Academic Committee, Dean of energy and power big data research institute, School of control and computer engineering, North China Electric Power University.