Energy efficiency optimisation modelling for security robots by edge computing

Muchun Zhou, Baochuan Fu* and Baoping Jiang

School of Electronics and Information Engineering,
Suzhou University of Science and Technology,
Suzhou 215009, China
and
Anhui Jianzhu University,
Hefei 230022, China
Email: muchunzhou1@163.com
Email: fubc@163.com
Email: baopingj@163.com
*Corresponding author

Abstract: In recent years, as indoor security robots are widely used in large public building places, robots have shouldered the pressure of solving security risks, patrol monitoring and fire warning. Mobile robots with rich sensors accomplish many computation-intensive tasks which account for a large proportion of the total energy consumption, thus affecting the service life of their batteries. Most security robots use micro-control units to maintain low energy consumption and hardware complexity, greatly limiting the local computing capacity of robots. In the case of analysing real-time video data and a large amount of sensor data, offloading a numerous intensive computing tasks to edge servers has become an extensive solution. This paper proposes a system model containing multi-robot terminals and multi-edge servers which uses a simulated annealing algorithm based on exchanging two different edge servers. This algorithm realises energy efficiency optimisation for security robots under minimum latency and power limitation by offloading partial computation-intensive tasks to edge servers. The feasibility of the proposed algorithm is also verified by the simulation results.

Keywords: edge computing; security robot; computation offloading; energy efficiency optimisation.

Reference to this paper should be made as follows: Zhou, M., Fu, B. and Jiang, B. (2022) 'Energy efficiency optimisation modelling for security robots by edge computing', *Int. J. Simulation and Process Modelling*, Vol. 18, No. 1, pp.36–44.

Biographical notes: Muchun Zhou is a postgraduate student in School of Electronics and Information Engineering at Suzhou University of Science and Technology. His research consists of edge computing, mobile cloud computing and internet of things.

Baochuan Fu is the Head of the Jiangsu Province Key Laboratory of Intelligent Building Efficiency at Suzhou University of Science and Technology. His research interests include network intelligent control, intelligent building and mathematical modelling in complex systems.

Baoping Jiang received his PhD in Control Theory from the Ocean University of China, Qingdao, China, in 2019. He was a joint training PhD candidate in the Department of Mechanical Engineering, Polytechnic di Milano, Milan, Italy from 2017–2019. He joined the Suzhou University of Science and Technology, Suzhou, China in 2019. He is an Associate Professor. His research interests include sliding mode control, stochastic systems, etc.

1 Introduction

With the development of urbanisation, the urban population has increased rapidly. The emergence of large numbers of people is bound to bring a series of high energy consumption (Korzhakov et al., 2021) and public security problems, especially in large densely populated building spaces, e.g., commercial squares, bus stations, railway stations, airports, schools and hotels. Aiming to create a safe and low carbon emissions indoor environment is becoming increasingly critical. Therefore, the current situation of public security in cities must be greatly improved. Presently, the public security problems raised in the above scenarios are solved with the large-scale deployment of security robots, but some new problems have also emerged.

Security robots (Bailin, 2019) have abundant sensors, e.g., infrared thermometers, high-definition cameras, smell detectors, audio signal receiver transmitters and the thermal imaging system. These mobile sensors incessantly detect the surrounding environment and produce video and audio data. (Lin et al., 2021), propose a wildfire hotspot detection system using flying robot and provide a comprehensive and reliable fire detection algorithm. However, a single security robot's terminal has limited resources (Ge, 2018), such as low battery capacity and weak data computing capability. These restrictions limit the operation time of security robots or hinder their decision-making in complex situations.

Cloud computing has attracted great attention in recent decades. (Zhao et al., 2020) discuss a health status evaluation method for wind turbines employing the advantages of the cloud model in dealing with uncertain information. In traditional cloud computing model, the robots can use the rich computing power of the cloud to offload some computation-intensive tasks to the cloud server (Barbarossa et al., 2014) and minimise the energy consumption of mobile terminals. But the lengths of cable and wireless networks between security robot terminals and cloud computing centres are long, it will cause network congestion and increase latency which affects robot real-time response. (Grigorik et al., 2013), propose a game theory method to realise efficient computing offloading of mobile cloud computing. This paper designed a decentralised computing offloading mechanism to realise the Nash equilibrium of the game, which can effectively reduce computing consumption. However, the energy-saving optimisation of large-scale mobile devices is well solved, and the problem of latency is not solved. In some specific application scenarios, (Rahman et al., 2017) study the smart factory environment, in which smart devices offload task-intensive data to the cloud server, and they only consider the energy consumption of resources. (Wang et al., 2016) discuss a Stackelberg game-based resource allocation strategy in the intelligent warehouse environment where the tasks are assigned according to bandwidth allocation cost. However, the problem of latency in resource allocation is not investigated while allocating resources for robotic tasks.

According to Tolia et al. (2016), in some low-latency scenarios, users are dissatisfied when the delay exceeds 80 ms and lose interest when the delay exceeds 1 s. In addition, maintaining the stable performance of ultra-long distance transmission tasks at the cloud side within a short time is difficult due to practical reasons, such as network routing path selection, packet processing of network routing nodes and security supervision of the network content (Tolia et al., 2006). The application of mobile edge computing technology (Barbarossa et al., 2014) has compensated for the shortcoming of cloud computing mentioned above. The research on computation offloading in edge computing (Al-Shuwaili and Simeone, 2017) has recently attracted the attention of many scholars, especially in the terms of minimising energy consumption (Ge et al., 2012) when the offloading target satisfies the latency minimisation constraint. In (Sardellitti et al., 2015), a multi-terminal system where multiple mobile users ask for computation offloading to a common cloud server was considered. This paper formulates the offloading problem as the joint optimisation of the radio resources (i.e., transmission precoding matrices of the terminals) and the computational resources (i.e., the CPU cycles/second assigned by the cloud to each terminal) to minimise the overall energy consumption of users while meeting latency constraints. A study (Wang et al., 2016) investigated partial computation offloading by jointly optimising the computational speed of smart mobile devices, the transmission power of devices, and the offloading ratio with two system design objectives: minimising the energy consumption of smart mobile devices and the latency of application execution. To address the energy consumption minimisation problem, the present study considered it as a convex problem, adopted a variable substitution technique and obtained the optimal solution. To address the non-convex and non-smooth LM problem, this work proposed a locally optimal algorithm (Wu et al., 2019) with a univariable search technique.

In Geng et al. (2018), the problem of energy-efficient computation offloading on multicore-based mobile devices running multiple applications was considered. This paper formalises the problem as a mixed-integer nonlinear programming problem that is NP-hard and then proposes a novel heuristic algorithm to jointly solve the offloading decision and task scheduling problem. The basic idea is to prioritise tasks from different applications to ensure that application time constraints and task-dependency requirements are satisfied. In Li and Wang (2018), the problem of energy-aware edge server placement was studied, and a more effective placement scheme with low energy consumption was explored. This work formulated the problem as a multi-objective optimisation problem and devised a particle-swarm-optimisation-based energy-aware edge server placement algorithm to find the optimal solution. Finally, in Li and Wang (2018), the placement model of edge server was considered, and the problem of reducing energy consumption was transformed into a multivariable optimisation problem. That is, the optimal solution was obtained by designing an energy optimisation algorithm based on particle swarm optimisation. Finally, simulation shows that compared with other algorithms, the energy consumption could be reduced by 10% under the constraint of meeting the maximum latency.

Combined with edge computing, this paper proposes an edge computing model based on a security robot. In this model, the security robot takes advantage of edge servers with abundant computational resources to offload partial tasks for computing. Generally, a security robot and an edge server have a one-to-one correspondence, but security robots have different computing powers and edge servers have different data processing rates. These factors contribute to the problem of distributing computation-intensive tasks (or edge servers) correctly for security robots under maximum latency and power constraints. Solving this problem can improve the cruise capability and reliability of robots. To solve this problem, we propose a simulated annealing algorithm (Wu et al., 2019b) based on exchanging two different edge servers and determine the optimal match scheme. Finally, the simulation results verify the effectiveness of the proposed algorithm scheme.

Notations: In this paper, E is denoted as the edge server's set. M means the number of the edge servers. R is denoted as the robots' task set. K means the number of the security robots.

2 Computation offloading system model

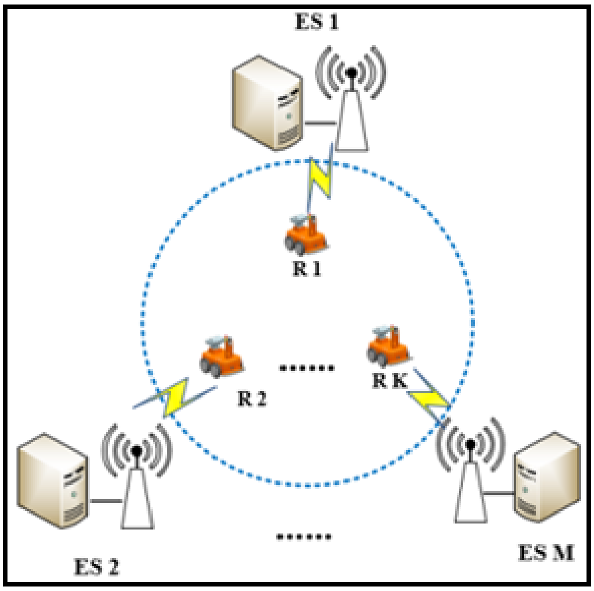
2.1 Model Components

On the basis of the analysis of the security robot system above, this paper proposes a security robot model combined with edge computing as shown in Figure 1. The model consists of two parts.

Security robots. Computation-intensive tasks, such as simultaneous localisation and mapping (SLAM) (Taheri and Xia, 2021) algorithms, are offloaded to edge servers via the wireless network embedded in the robots' vehicle.

Edge servers. The edge servers handle the partial computation-intensive tasks offloaded by the selected security robots located around the site.

Figure 1 Computation offloading model of security robots (see online version for colours)



2.2 Model workflow

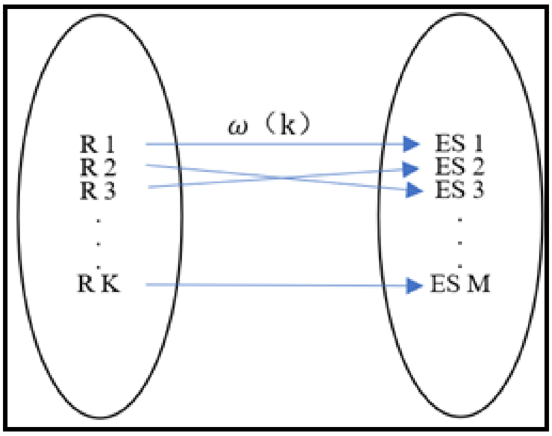
Numerous computing tasks are generated by security robots when performing patrolling services and executing SLAM algorithms (Chen et al., 2021). This model assumes that each robot connects to an edge server, and each edge server only undertakes the tasks of one security robot. The simulated annealing algorithm is used in this model for the robot terminals to achieve minimal energy consumption.

3 Problem formulation

The edge server group is denoted as set $E = \{1, 2, 3, ..., M\}$, and each security robot task is represented as set $R = \{1, 2, 3, ..., K\}$. We assume that $M \geq K$, i.e., z. the number of edge servers is greater than or equal to the number of mobile security robots. To achieve one-to-one robot and edge server correspondence,

We set up $(M\text{-}K)$ virtual tasks $S^{tot} = 0$. Figure 2 shows the mapping relationship between the security robots and the edge servers, i.e., $\omega(k) = m$ means the K_{th} task of the security robot offloads to the m_{th} edge server.

Figure 2 Mapping relationship between security robot and edge server (see online version for colours)



The order of power gain of M edge server channels is assumed to be

$$g_1 > g_2 > \ldots > g_M \tag{1}$$

Under the conditions above, the minimum transmission power (Sarker et al., 2019) (transmission time is t, s) sent by the security robot to the edge server is

$$P^{tot}\left(\{S_k\}_{\forall k \in R}, t, \omega(k)\right) = Wn_0 \sum_{i=1}^{K} \frac{1}{g_i(\omega(k))} 2^{\frac{1}{tw}\sum_{m=1}^{K} S_m} - \frac{Wn_0}{g_M} \tag{2}$$

In equation (2), parameter S_k (Mbits) represents the number of tasks transmitted by each security robot to the edge server through the wireless network; parameter W (MHz) denotes the channel bandwidth; parameter n_0 (W/MHz) denotes the power density of the background noise; and parameter g_i (dimensionless parameters) denotes the random channel gain. Then, the total energy spent by the security robots to offload tasks to the edge servers is

$$E_{up} = tP^{tot}\left(\{S_k\}_{\forall k \in R}, t, \omega(k)\right) \tag{3}$$

Thus, the overall latency $\left(d_k^{ove}\right)$ of the K_{th} security robot to complete its assigned task is

$$d_k^{ove} = Max\left\{\frac{S_k^{tot} - S_k}{u_{L,k}}, t + \frac{S_k}{u_w(k)}\right\} \tag{4}$$

We use equation (4) to quantify the total latency of the security robot. $u_{L,k}$ (bits/s) is the local computing speed of the K_{th} security robot, $u_{\omega(k)}$ (Mbits/s) is the computing speed of the edge server selected by the K_{th} security robot, $\frac{S_k^{tot} - S_k}{u_{L,k}}$ is the local computing latency of the K_{th} security robot to conduct the remaining tasks $S_k^{tot} - S_k$ and $t + \frac{S_k}{u_\omega(k)}$ is the total latency of the security robot to offload part of its tasks to the selected edge server.

Thus, the energy optimisation problem (EM-i) can be concluded as follows:

$$\begin{aligned}&(EM\text{-}i): \ Min\ tP^{tot}\left(\{S_k\}_{\forall k\in R}, t, \omega(k)\right)\\ &+\sum_{k\in R}\frac{S_k^{tot} - S_k}{u_{L,k}}\rho\left(u_{L,k}\right)^3 = tP^{tot}\left(\{S_k\}_{\forall k\in R}, t, \omega(k)\right)\\ &+\sum_{k\in R} S_k^{tot} - S_k\rho\left(u_{L,k}\right)^2\end{aligned} \tag{5}$$

$$S.t.: \ P^{tot}\left(\{S_k\}_{\forall k\in R}, t, \omega(k)\right) \le P^{\max} \tag{6}$$

$$d_k^{ove} = Max\left\{\frac{S_k^{tot} - S_k}{u_{L,k}}, t + \frac{S_k}{u_w(k)}\right\} \le T_k^{\max}, \forall k\in R \tag{7}$$

$$u_{L,k} \le u_L^{max} \tag{8}$$

Variables: $\{S_k\}_{\forall k\in R}$, t, $\omega(k)$ and $\{u_{L,k}\}_{\forall k\in R}$

In EM-i, tP^{tot} $(\{S_k\}_{\forall k\in R}, t, \omega(k))$ is the transmission energy consumption of the security robots' offloading tasks $\sum_{k\in R}\frac{S_k^{tot} - S_k}{u_{L,k}}\rho\left(u_{L,k}\right)^3$ represents the local CPU computing energy consumption of the security robots (based on the dynamic voltage adjustment principle, the computational power of the local CPU is the third power of the computing speed which is determined by the chip structure ρ). Constraint (6) indicates that the power transmitted by the network cannot exceed the power budget of the security robot (P^{max}). Constraint (7) indicates that the total latency of each security robot cannot exceed the maximum latency $\left(T_k^{max}\right)$. Constraint (8) indicates that the total local CPU calculation rate of the security robot cannot exceed u_L^{max}.

4 Allocation method based on simulated annealing algorithm

4.1 Decomposition of problem

Solving optimisation problem EM-i is a great challenge under the condition that the security robot and the edge server are not fixed matched. The most direct method is to make arbitrary combinations of tasks, calculate the minimum energy consumption of each combination, and then determine the most suitable situation.

This method is undoubtedly feasible for the scenario with few robots and edge servers but complicated to use in cases with many robots and edge servers. To overcome the challenges, a simulated annealing algorithm can provide a powerful method that a slow decrease in the probability of accepting worse solutions as the solution space is explored, and accept worse solutions allows for a more extensive search for global optimal solution (Çetin and Keçebaş, 2021). It means, the temperature progressively decreases from an initial value to zero. At each iteration, the algorithm randomly selects a mapping relationship, measures its energy consumption, and moves to another solution according to the temperature-dependent probabilities of selecting better or worse match. Through this algorithm, the local matching server can achieve an approximate solution to the global minimum energy consumption.

In the given state of this set of matching relations, mapping relation ω (m) is given. From inequality (7), we can conclude that

$$\begin{aligned}&\frac{S_k^{tot} - S_k}{u_{L,k}} \le T_k^{max}, \ \forall k\in R\\ &\frac{S_k^{tot} \text{ - } S_k}{T_k^{max}} \le u_{L,k}, \ \forall k\in R\end{aligned} \tag{9}$$

Inequality (9) represents the minimum computation speed of the robots required by the tasks processed locally. In addition,

$$\begin{aligned}&t + \frac{S_k}{u_{\omega(k)}} \le T_k^{max}\\ &S_k \le (T_k^{max} - t)u_{w(k)}, \ \forall k\in R\end{aligned} \tag{10}$$

Inequality (9) is substituted into objective function Em-i which can be transformed into (P1)

$$\begin{aligned}&P1: \ Min\ tP^{tot}\left(\{S_k\}_{\forall k\in R}, t\right) + \sum_{k\in R}\left(S_k^{tot} - S_k\right)\rho\left(\frac{S_k^{tot} - S_k}{T_k^{\max}}\right)^2\\ &S.t.: P^{tot}\left(\{S_k\}_{\forall k\in R}, t\right) \le P^{\max}\end{aligned} \tag{11}$$

$$0 \le S_k \le \left(T_k^{\max} - t\right)u_{w(k)}, \ \forall k\in R \tag{12}$$

$$0 \le \frac{S_k^{tot} - S_k}{T_k^{\max}} \le u_L^{\max} \tag{13}$$

Variable: $t\ and \{S_k\}_{\forall k\in R}$

Problem (P1) is still a non-convex optimisation problem. To solve this problem, we regard variable t as a constant under constraint condition $0 \le t \le T_k^{\max}, \forall k\in R.$ Within this constraint range, the minimum energy consumption of (P1) determined by the matching relation can be solved through linear search using the CVX of the MATLAB software (Gustafsson et al., 2015).

In the initialisation, the algorithm generates the initial solution at random. It means that the security robot group randomly matches the edge servers. Under this matching condition, the optimal energy consumption is solved. Then, in order to avoid the local optimal solution, we generate a new solution by perturbation. That is a new edge server group is selected for the security robots to match, so as to generate a new optimal energy consumption solution. The difference outcome is made between the newly matching scheme of the optimal energy consumption and that of the original scheme and determine whether to accept the original solution directly or the new solution by the Metropolis criterion.

In this paper, we set the initial annealing temperature T_0 , the number of iteration in each cooling and annealing speed q. Accordingly, the updating way of temperature is:

$$T_{i+1} = T_i * q \tag{14}$$

In equation (14) T_i is the i_{th} temperature, q is the annealing speed and its value range is (0, 1).

4.2 Algorithm flow

Step 1 Setting the initial solution: At the beginning of the algorithm, we set the initial annealing temperature $T = T_0$, the lowest annealing temperature T_{min} and the number of iteration Q in each cooling. An matching mapping $\{\omega(m) = k\}_{\forall m\in R, \forall k\in E}$ between a security robot and an edge sever is randomly generated, and the minimal energy consumption of the problem about EM-i is solved under this mapping. We make $E_{best} = E^{\min}_{\{\omega(m)=k\}_{\forall m\in R,\forall k\in E}}$ and update T to the new temperature T_i.

Step 2 Generating the difference between the new solution of the minimal energy consumption and current solution of that: A updating matching relation is generated by disturbing the original matching relation, and the minimum energy consumption $E^{min}_{\{\omega(m)=k\}^{upd}_{\forall m\in R,\forall k\in E}}$ will be solved. The corresponding difference is:

$$\Delta = E^{min}_{\{\omega(m)=k\}^{upd}_{\forall m\in R,\forall k\in E}} - E^{min}_{\{\omega(m)=k\}_{\forall m\in R,\forall k\in E}} \tag{15}$$

Step 3 Judging whether the new solution can be accepted: If $\Delta \le 0$, we update the original solution $E_{best} = E^{min}_{\{\omega(m)=k\}^{upd}_{\forall m\in R,\forall k\in E}}$. Otherwise, we accept the new solution as the current optimal outcome according to the Metropolis criterion:

$$P = \begin{cases} 1\ ,\ E^{min}_{\{\omega(m)=k\}^{upd}_{\forall m\in R,\forall k\in E}} < E^{min}_{\{\omega(m)=k\}_{\forall m\in R,\forall k\in E}} \\ e^{-\frac{\Delta}{T}}, E^{min}_{\{\omega(m)=k\}^{upd}_{\forall m\in R,\forall k\in E}} \ge E^{min}_{\{\omega(m)=k\}_{\forall m\in R,\forall k\in E}} \end{cases} \tag{16}$$

Cooling is carried out in accordance with formula (14). At the temperature T_{i+1}, acceptance of new solutions are solved in step 2 and 3.

Step 4 Finding the feasible solution: we judge whether the temperature T reaches the lowest temperature T_{min}. If it reaches the minimal temperature, the algorithm will be terminated. Otherwise, we return to step (2) to continue the algorithm.

5 Simulation results and analysis

5.1 Simulation scenario and parameters setting

Table 1 Simulation parameters and values

Parameters	*Physical meaning*	*Values*
M	Number of edge servers	5
K	Number of security robots	5
W	Channel bandwidth /MHz	8
n_0	The power density of background noise /W/MHz	10^{-8}
P^{max}	Maximum power of security robot /W	50
T_k^{max}	Maximum computing latency/s	[2, 3 ,4, 5, 6]
u_L^{max}	Maximum computation speed of the robot	8 Mbits/s
T_0	Initial temperature	300
$T_{\min}$	Final temperature	0.02
q	Speed of annealing	0.95

This paper verifies the performance of the proposed algorithm in the scenario with five security robots and five edge servers. We establish a 5-ES scene in which five edge servers are evenly distributed around the circle with a radius of 500 m, and security robots are randomly distributed in this circular scene. To better observe the convergence effect, two groups of experiments were conducted, and the variable was set to the total number of tasks of each security robot. According to the distance model, the continuous channel gain from the security robot to the edge server was generated. The random channel power gain we used $\{g_i\}_{i\in E}$ = {9.1831 × 10^{-6}, 5.4139 × 10^{-6}, 3.2571 × 10^{-6}, 2.4311 × 10^{-6}, 2.2623 × 10^{-6}, 2.2623 × 10^{-6}}. We set the bandwidth W = 8 MHz, n_0 = 10^{-8} and five edge servers' computation speed $u_{\omega k}$ = [12, 15, 11, 10, 9] Mbit/s. The maximum power of a security robot was 5 J/s. The robots' maximum latency $T_k^{max} = [2, 3, 4, 5, 6]$ s. We set the two groups of total tasks for each security robot $S_k^{max} = [5, 6, 7, 8, 9]$ M and [11, 12, 13, 10, 8] M and the maximum computation speed of the robots to $u_L^{max} = 8$ Mbits/s. Initial temperature T was set to 300. All simulations ran on Intel(R) Core (TM)i7 6300HQCPU@2.30 GHz PC. The complete simulation parameters are shown in Table 1. The simulation parameters come from (Zhong et al., 2020).

Figure 3 shows the convergence graph of the energy optimisation used by the simulated annealing algorithm which is based on exchanging two different edge servers in the 5-ES scenario. The horizontal axis is iterations i, and the vertical axis is the minimum energy consumption obtained by the optimisation algorithm in this proposed scenario (J). The discrete results are connected to facilitate the observation of the convergence effect of the final energy. According to the simulation results, when the total task of the security robots is $S_k^{tot} = [5, 6, 7, 8, 9]M$, iteration i is 72, and the time consumed by the algorithm is 0.654×10^3s. When i is 28, the energy consumption converges to 0.0904 J. At this point, the mapping relations between security robots and edge servers are $\omega_1 = 4$, $\omega_2 = 5$, $\omega_3 = 3$, $\omega_4 = 2$ and $\omega_5 = 1$. Figure 4 shows the numbers of tasks of the security robots are $S_k^{tot} = [11, 12, 13, 10, 8]M$, i is 73, and the time consumed by the running is 0.725×10^3s. When i is 33, the energy consumption converges to 0.2024 J. At this point, the mapping relations between security robots and edge servers are $\omega_1 = 5$, $\omega_2 = 2$, $\omega_3 = 1$, $\omega_4 = 3$ and $\omega_5 = 4$.

To verify the feasibility of this algorithm, permutation on all the tasks $S_k^{tot} = [5, 6, 7, 8, 9]M$ was performed, and the corresponding minimal energy consumption $\left(E^{min}_{\{\omega(m)\}_{m\in R}}\right)$ was solved. To simplify the expression, this paper defines [a, b, c, d, e] as the security robots whose total amounts of tasks are a, b, c, d and e, respectively, and these tasks are offloaded to the corresponding edge servers.

Figure 3 $E^{\min}_{\{\omega_m^*\}_{m\in R}}$ -i convergence graph (total task amount [5, 6, 7, 8, 9] m) (see online version for colours)

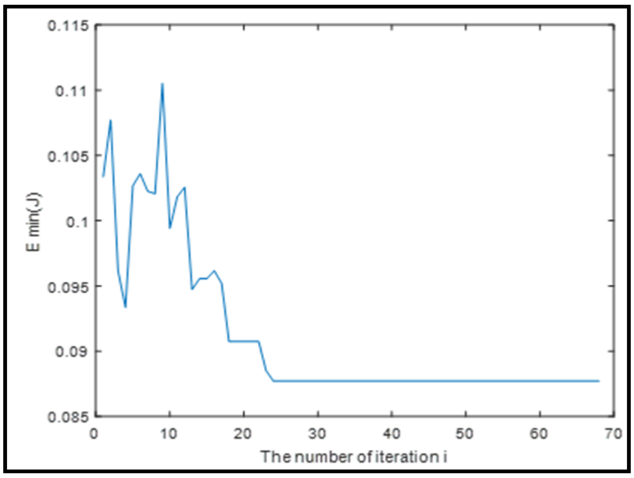


Table 2 The minimal energy consumption of entire permutation in 5-ES scenario

Permutation scheme	*E_{min} (J)*	*Permutation scheme*	*E_{min} (J)*	*Permutation scheme*	*E_{min} (J)*	*Permutation scheme*	*E_{min} (J)*
[5,6,7, 8,9]	0.1055	[5 6 8 7 9]	0.1043	[5 6 8 9 7]	0.1041	[5 6 9 7 8]	0.1034
[5 7 6 9 8]	0.1032	[5 7 8 6 9]	0.1042	[5 7 9 6 8]	0.1039	[5 7 9 8 6]	0.1023
[5 8 6 9 7]	0.1017	[5 8 7 6 9]	0.1012	[5 8 7 9 6]	0.1008	[5 8 9 6 7]	0.1019
[5 8 9 7 6]	0.1015	[5 9 6 7 8]	0.1010	[5 9 6 8 7]	0.1004	[5 9 7 6 8]	0.0988
[5 9 7 8 6]	0.0995	[5 9 8 6 7]	0.0993	[5 9 8 7 6]	0.0987	[6 5 7 9 8]	0.0982
[6 5 7 8 9]	0.0976	[6 5 8 7 9]	0.0974	[6 5 8 9 7]	0.1037	[6 5 9 7 8]	0.1035
[6 5 9 8 7]	0.1028	[6 7 5 8 9]	0.1024	[6 7 5 9 8]	0.1017	[6 7 8 5 9]	0.1015
[6 7 8 9 5]	0.1044	[6 7 9 5 8]	0.1005	[6 7 9 8 5]	0.1007	[6 8 5 7 9]	0.1022
[6 8 5 9 7]	0.1017	[6 8 7 5 9]	0.1003	[6 8 7 9 5]	0.1003	[6 8 9 5 7]	0.0982
[6 8 9 7 5]	0.0985	[6 9 5 7 8]	0.0998	[6 9 5 8 7]	0.0995	[6 9 7 5 8]	0.0980
[6 9 7 8 5]	0.0981	[6 9 8 5 7]	0.0969	[6 9 8 7 5]	0.0972	[7 6 5 8 9]	0.1034
[7 6 5 9 8]	0.1032	[7 6 8 5 9]	0.1006	[7 6 8 9 5]	0.1006	[7 6 9 5 8]	0.0995
[7 6 9 8 5]	0.0997	[7 5 6 8 9]	0.1015	[7 5 6 9 8]	0.1013	[7 5 8 6 9]	0.0997
[7 5 8 9 6]	0.0991	[7 5 9 6 8]	0.0986	[7 5 9 8 6]	0.0982	[7 8 6 5 9]	0.0981
[7 8 6 9 5]	0.0980	[7 8 5 6 9]	0.0990	[7 8 5 9 6]	0.0982	[7 8 9 6 5]	0.0954
[7 8 9 5 6]	0.0948	[7 9 6 5 8]	0.0957	[7 9 6 8 5]	0.0959	[7 9 5 6 8]	0.0966
[7 9 5 8 6]	0.0962	[7 9 8 6 5]	0.0942	[7 9 8 5 6]	0.0936	[8 6 7 5 9]	0.0984
[8 6 7 9 5]	0.0966	[8 6 5 7 9]	0.0962	[8 6 5 9 7]	0.0971	[8 6 9 7 5]	0.0971
[8 6 9 5 7]	0.0962	[8 7 6 5 9]	0.0971	[8 7 6 9 5]	0.0970	[8 7 5 6 9]	0.0980
[8 7 5 9 6]	0.0973	[8 7 9 6 5]	0.0945	[8 7 9 5 6]	0.0938	[[8 5 6 7 9]	0.0985
[8 5 6 9 7]	0.0980	[8 5 7 6 9]	0.0976	[8 5 7 9 6]	0.0969	[8 5 9 6 7]	0.0957
[8 5 9 7 6]	0.0952	[8 9 6 7 5]	0.0928	[8 9 6 5 7]	0.0924	[8 9 7 6 5]	0.0920
[8 9 7 5 6]	0.0914	[8 9 5 6 7]	0.0923	[8 9 5 7 6]	0.0931	[9 6 7 8 5]	0.0952
[9 6 7 5 8]	0.0950	[9 6 8 7 5]	0.0943	[9 6 8 5 7]	0.0939	[9 6 5 7 8]	0.0968
[9 6 5 8 7]	0.0966	[9 7 6 8 5]	0.0941	[9 7 6 5 8]	0.0928	[9 7 8 6 5]	0.0230

Table 2 The minimal energy consumption of entire permutation in 5-ES scenario (continued)

Permutation scheme	E_{min} *(J)*	*Permutation scheme*	E_{min} *(J)*	*Permutation scheme*	E_{min} *(J)*	*Permutation scheme*	E_{min} *(J)*
[9 7 8 5 6]	0.0916	[9 7 5 6 8]	0.0947	[9 7 5 8 6]	0.0942	[9 8 6 7 5]	0.0919
[9 8 6 5 7]	0.0915	[9 8 7 6 5]	0.0914	[9 8 7 5 6]	0.0904	[9 8 5 6 7]	0.0931
[9 8 5 7 6]	0.0952	[9 8 5 7 6]	0.0950	[9 5 6 7 8]	0.0943	[9 5 6 8 7]	0.0939
[9 5 7 6 8]	0.0952	[9 5 7 8 6]	0.0950	[9 5 8 6 7]	0.0944	[9 5 8 6 7]	0.0939

A total of 120 data was generated by the entire permutation. From the Table 2, the optimal allocation scheme was [9 8 7 5 6], and the total energy consumption obtained at this time was 0.0904 J, which is consistent with the results obtained by the proposed algorithm.

5.2 *Comparison of algorithm results*

Figure 5 validates the accuracy of our proposed algorithm in solving this energy optimisation problem compared to LINGO (i.e., a commercial optimisation package). All tested cases in Figure 5 show that our proposed algorithm can achieve the results almost same as those from LINGO's global solver with a narrow difference.

Figure 5 Proposed algorithm accuracy compared with lingo

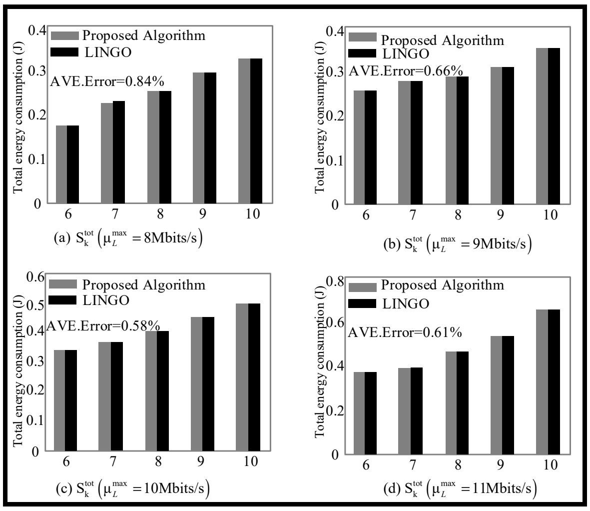


Figure 6 shows the energy consumption comparison between the algorithm proposed in this paper that security robots match appropriate edge servers to optimise energy consumption and the scheme that all computing tasks are locally processed by security robots under different u_L^{max} conditions. The data shows that the overall energy saving compared to the local scheme is nearly 20%. Specifically, the scheme that all computing tasks are processed locally means that in the 5-ES scenario, all security robot terminals use the local computing resources of intelligent devices to process energy-intensive tasks without offloading edge computing tasks. The results of Figure 5 show that the energy consumption obtained by the algorithm proposed in this paper can be much better than that of the scheme where all tasks are processed locally.

Figure 6 Contrast experiment between proposed algorithm and local scheme under different u_L^{max} conditions (see online version for colours)

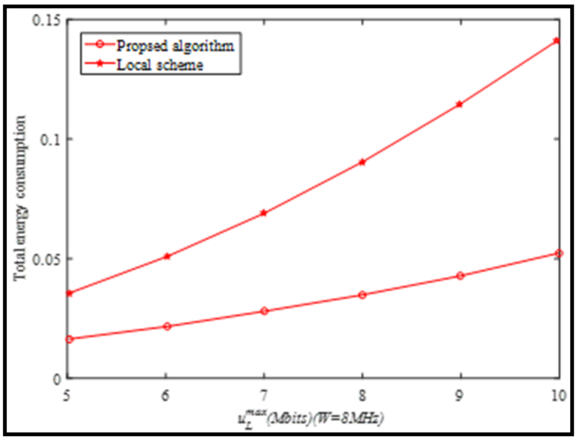


Figure 7 Contrast experiment between proposed algorithm and non-match edge computing server scheme under different u_L^{max} conditions

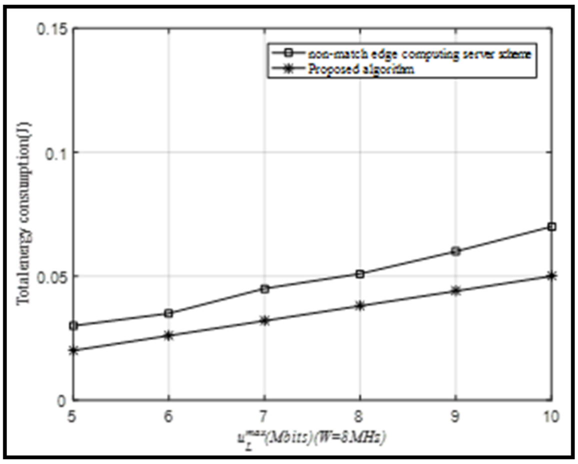


Figure 7 shows the energy consumption comparison between the algorithm proposed in this paper that security robots match appropriate edge servers to optimise energy consumption and non-match edge computing server's scheme under different u_L^{max} conditions in the 5-ES scenario. The data shows that the overall energy saving compared to the non-match scheme is nearly 12%. This simulation experiment verifies the significance of matching the right edge computing servers. This is due to the difference in the ability of each edge computing servers to handle various types of tasks.

Figure 8 Contrast experiment between proposed algorithm and traditional algorithm under different s^{tot} conditions

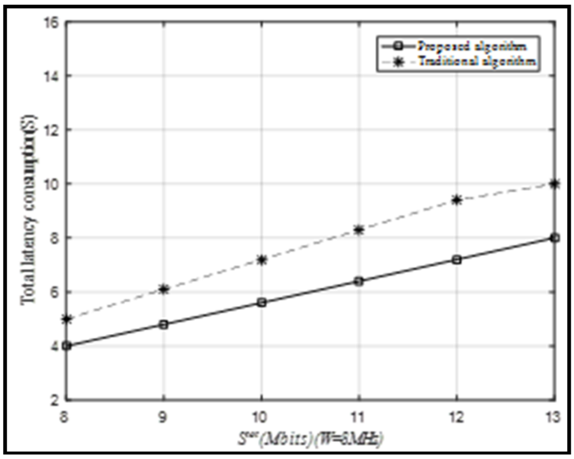


Figure 8 shows the comparison between our proposed algorithm and the traditional algorithm in terms of total latency in 3-ES system. The data shows that the latency of the edge computing model is nearly 25 % better than that of the cloud's model. Specifically, the traditional algorithm means all robots, terminals use the local computing and cloud computing resources to deal with computing tasks. It can be analysed from the figure that the latency obtained by the algorithm proposed in this paper is better than that obtained by the traditional algorithm. This advantage is due to the rich computing resources of edge servers and their low latency offloading of computing task.

6 Conclusions

In this paper, the issue of energy optimisation for security robots by using edge computing technology has been discussed. First, the scenario of multiple robots and edge servers has been considered, and simulated annealing algorithm based on exchanging two different edge servers has been proposed. Second, this approach has found the optimal match between security robots and edge servers and has achieved minimum energy consumption for security robots. Finally, the simulation results have shown that our proposed algorithm is better than the local scheme and non-match scheme in energy saving. In the future, we plan to consider a complex scenario where a single security robot's task can be simultaneously offloaded to different edge servers to further optimise the energy consumption.

Acknowledgements

The research is funded by the National Science Foundation of China, grant number 61672731, 61803279, 51874205, in part by Anhui Province Key Laboratory of Intelligent Building and Building Energy Saving under grant IBES2020KF01

References

Al-Shuwaili, A. and Simeone, O. (2017) 'Energy-efficient resource allocation for mobile edge computing-based augmented reality applications', *IEEE Wireless Communications Letters*, Vol. 6, No. 3, pp.398-401.

Bailin (2019) 'How will security patrol robots be defined in the 5G era?', *China Security*, Vol. 162, No. 6, pp.51-53.

Barbarossa, S., Sardellitti, S. and Di Lorenzo, P. (2014) 'Communicating while computing: distributed mobile cloud computing over 5G heterogeneous networks', *IEEE Signal Processing Magazine*, Vol. 31, No. 6, pp.45–55.

Bui, D.-M., Tu, N.A. and Huh, E.-N. (2021) 'Energy efficiency in cloud computing based on mixture power spectral density prediction', *Journal of Supercomputing*, Vol. 77, No. 3, pp.2998–3023.

Çetin, G. and Keçebaş, A. (2021) 'Optimization of thermodynamic performance with simulated annealing algorithm: a geothermal power plant', *Renewable Energy*, Vol. 172, No. 2, pp.968–982.

Chen, Y., Zhang, N., Zhang, Y., Chen, X., Wu, W. and Shen, X. (2021) 'Energy efficient dynamic offloading in mobile edge computing for internet of things', *IEEE Transactions on Cloud Computing*, Vol. 9, No. 3, pp.1050–1060.

Ge, G. (2018) 'Review of research on security patrol robot', *Computing Knownledge Technology*, Vol. 14, No. 12, pp.178–179+182.

Ge, Y., Zhang, Y., Qiu, Q. and Lu, Y-H. (2012) 'A game theoretic resource allocation for overall energy minimization in mobile cloud computing system', in: *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, (Redondo Beach, California, USA: Association for Computing Machinery)

Geng, Y., Yang, Y. and Cao, G. (2018) 'Energy-efficient computation offloading for multicore-based mobile devices', in: *IEEE INFOCOM - IEEE Conference on Computer Communications*, pp.46–54.

Grigorik, I. (2013) *High Performance Browser Networking: What Every Web Developer Should Know about Networking and Web Performance*, O'Reilly Media, Inc., China.

Gustafsson, M., Tayli, D., Ehrenborg, C., Cismasu, M. and Nordebo, S. (2015) 'Tutorial on antenna current optimization using MATLAB and CVX'.

Korzhakov, A., Oskin, S., Korzhakov, V. and Korzhakova, S. (2021) 'The simulation of heat supply system with a scale formation factor to enable automation of greenhouse geothermal heat supply system, *Machines*, Vol. 9, No. 3, pp.64.

Li, Y. and Wang, S. (2018) 'An energy-aware edge server placement algorithm in mobile edge computing', in: *IEEE International Conference on Edge Computing (EDGE))*, pp.66–73.

Pande, V., Marlecha, C. and Kayte, S. (2016) 'A review-fog computing and its role in the internet of things', *International Journal of Engineering Research and Applications*, Vol. 6, No. 10, pp.2248–96227.

Rahman A, J Jin, Cricenti A.L. et al. 'Communication-aware cloud robotic task offloading with on-demand mobility for smart factory maintenance', *IEEE Transactions on Industrial Informatics*, Vol. 2019, No. 5, pp.2500–2511.

Sardellitti, S., Scutari, G. and Barbarossa, S. (2015) 'Joint optimization of radio and computational resources for multicell mobile-edge computing', *IEEE Transactions on Signal and Information Processing over Networks*, Vol. 1, No. 2, pp.89–103.

Sarker, V.K., Queralta, J.P., Gia, T.N., Tenhunen, H. and Westerlund, T. (2019) 'Offloading SLAM for indoor mobile robots with edge-fog-cloud computing', in: *1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT))*, pp.1–6.

Taheri, H. and Xia, Z.C. (2021) 'SLAM; definition and evolution', *Engineering Applications of Artificial Intelligence*, Vol. 97, No. 3, p.104032.

Tolia, N., Andersen, D.G. and Satyanarayanan, M. (2006) 'Quantifying interactive user experience on thin clients', *Computer*, Vol. 39, No. 3, pp.46–52.

Wang, L. (2015) *A Pricing Mechanism for Task Oriented Resource Allocation in Cloud Robotics*, Springer International Publishing, Switzerland.

Wang, Y., Sheng, M., Wang, X., Wang, L. and Li, J. (2016) 'Mobile-edge computing: partial computation offloading using dynamic voltage scaling', *IEEE Transactions on Communications*, Vol. 64, No. 10, pp.4268–4282.

Wu, Z., Jiang, B. and Kao, Y. (2019a) 'Finite-time $\mathscr{H}\infty$ filtering for Itô stochastic Markovian jump systems with distributed time-varying delays based on optimisation algorithm', *IET Control Theory and Applications,* Vol. 13, No. 5, pp.702–710.

Wu, Z., Karimi, H.R. and Dang, C. (2019b) 'A deterministic annealing neural network algorithm for the minimum concave cost transportation problem', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, No. 10, pp.4354–4366.

Zhao, W. and Cui, M. (2020) 'Real-time health status evaluation for electric power equipment based on cloud model', *International Journal of Simulation and Process Modelling*, Vol.15 No.1/2, pp.134–144.

Zhong, R., Liu, X. Liu, Y., Chen, Y. and Wang, X. (2020) *Path Design and Resource Management for Noma Enhanced Indoor Intelligent Robots*, arxiv preprint arxiv.

Zhongjie, L., Bohdanov., D., Liu, H. and Wotton, M. (2021) 'Autonomous wildfire hotspot detection using a fixed wing UAV', *International Journal of Aerospace System Science and Engineering*, Vol. 1, No. 1, pp.68–84.