HR-kESP: A Heuristic Algorithm for Robustness-Oriented k Edge Server Placement

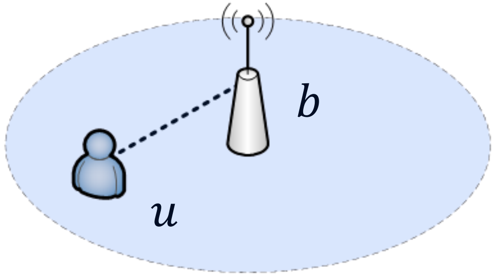
Haiquan Hu, Jifu Chen and Chengying Mao

School of Software and IoT Engineering
Jiangxi University of Finance and Economics

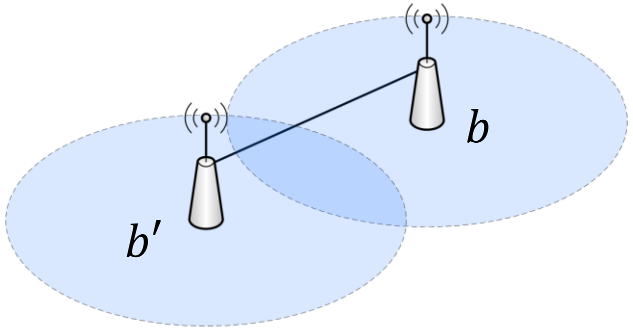
The 23rd International Conference on Algorithms and Architectures for Parallel Processing (*ICA3PP 2023*)

The k Edge server placement (***k*ESP**) problem is to select k suitable base stations as the server deployment locations in a given mobile network.

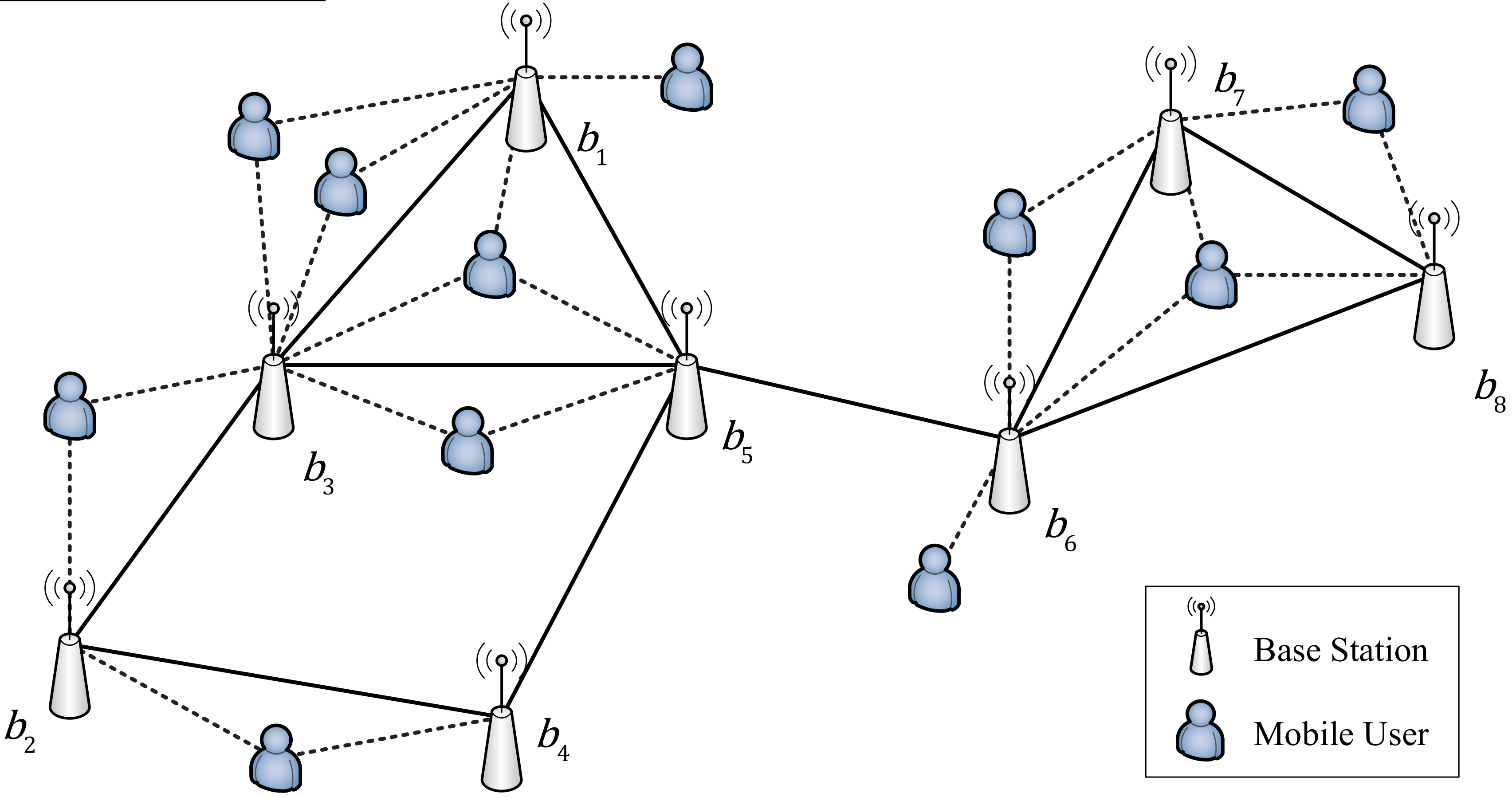
The **coverage** of the base station



The accessing relationship between **a user and a base station**.



The accessing relationship between **two base station**.



The **Robustness-Oriented** k ESP is to construct a edge server network to achieve the following two objective:

- More **user coverage**
- Higher anti-failure capability of network (network **robustness**)

Network robustness can be defined as the capability of the server network to maintain the original quality of service (Qos) in the face of sudden failures.

- The server network can operate steadily at **close to the theoretical best performance**
- Avoid high economic losses and network maintenance costs

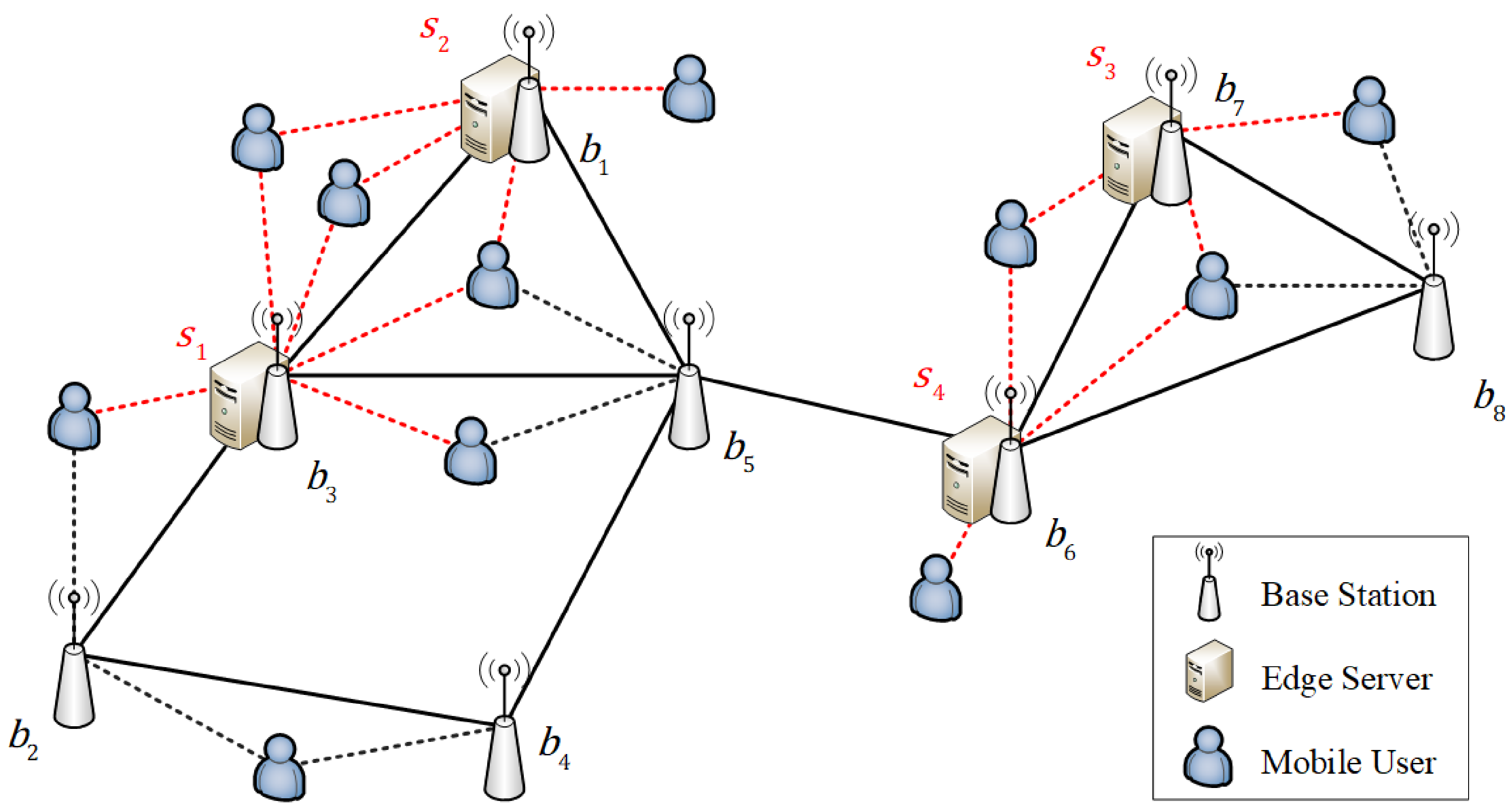
There is a robustness-oriented kESP solution (server deployment scheme $\mathbf{p}$).

Server budget: $k = 4$

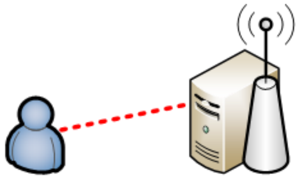
Deployment scheme: $\{b_1, b_3, b_6, b_7\}$

User coverage: **10**

Network robustness: **5**

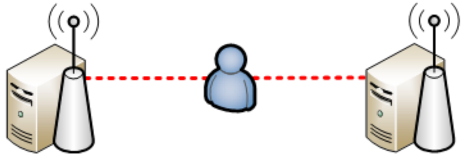


User Coverage $coverage(\mathbf{p})$ is the number of users severed by edge servers

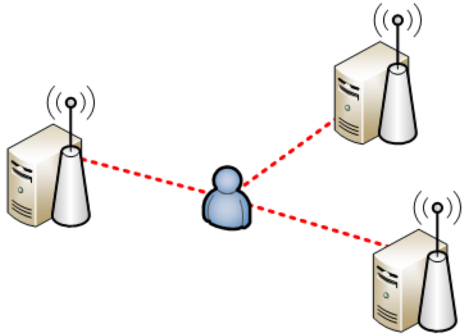


$coverage(\mathbf{p}) = 1$

Network Robustness $robust(\mathbf{p})$ is the number of times which the users are repeatedly served by edge servers.



$robust(\mathbf{p}) = 1$



$robust(\mathbf{p}) = 2$

For the kESP solution $\mathbf{p} = \{p_1, p_2, \dots, p_m\}$ (m is the number of base station). The **robustness-oriented** k edge server placement problem can be formulated as follows:

$$maximize\ CR(\mathbf{p}) = \sqrt{\omega_\mathrm{c} \cdot \big(coverage(\mathbf{p})\big)^2 + \omega_\mathrm{r} \cdot (robust(\mathbf{p}))^2}$$

while satisfying the following constraints:

$$\boxed{\omega_\mathrm{c} = \omega_\mathrm{r} = 0.5}$$

$$p_i \in \{0,1\}$$

Each base station has two states whether it deploys servers ($p_i = 1$) or not ($p_i = 0$).

$$\sum_{i=1}^{m} p_i = k$$

And the number of servers is equal to the server budget k.

The Main Contributions:

- A **H**euristic algorithm for **R**obustness-oriented $\boldsymbol{k}$ **E**dge **S**erver **P**lacement named **HR-kESP** is proposed to effectively solve the kESP considering user coverage and network robustness

- A **simulation-based evaluation method** for the **network robustness** is proposed. After simulating the failure randomly, the network robustness is evaluated by the variation of the network performance (User Survival Rate).

- Intensive experiments are conducted on the widely-used EUA dataset[1] by considering various application scenarios, and the results indicate that HR-kESP **outperforms other representative algorithms in most cases**.

1. Lai, P., He, Q., Abdelrazek, M., et al.: Optimal edge user allocation in edge computing with variable sized vector bin packing. In: Proceedings of the 16th International Conference on Service-Oriented Computing (ICSOC 2018), pp. 230-245. (2018)

The main thought of HR-kESP

The distribution of **service requirements** is very similar to the distribution of **users and base stations**

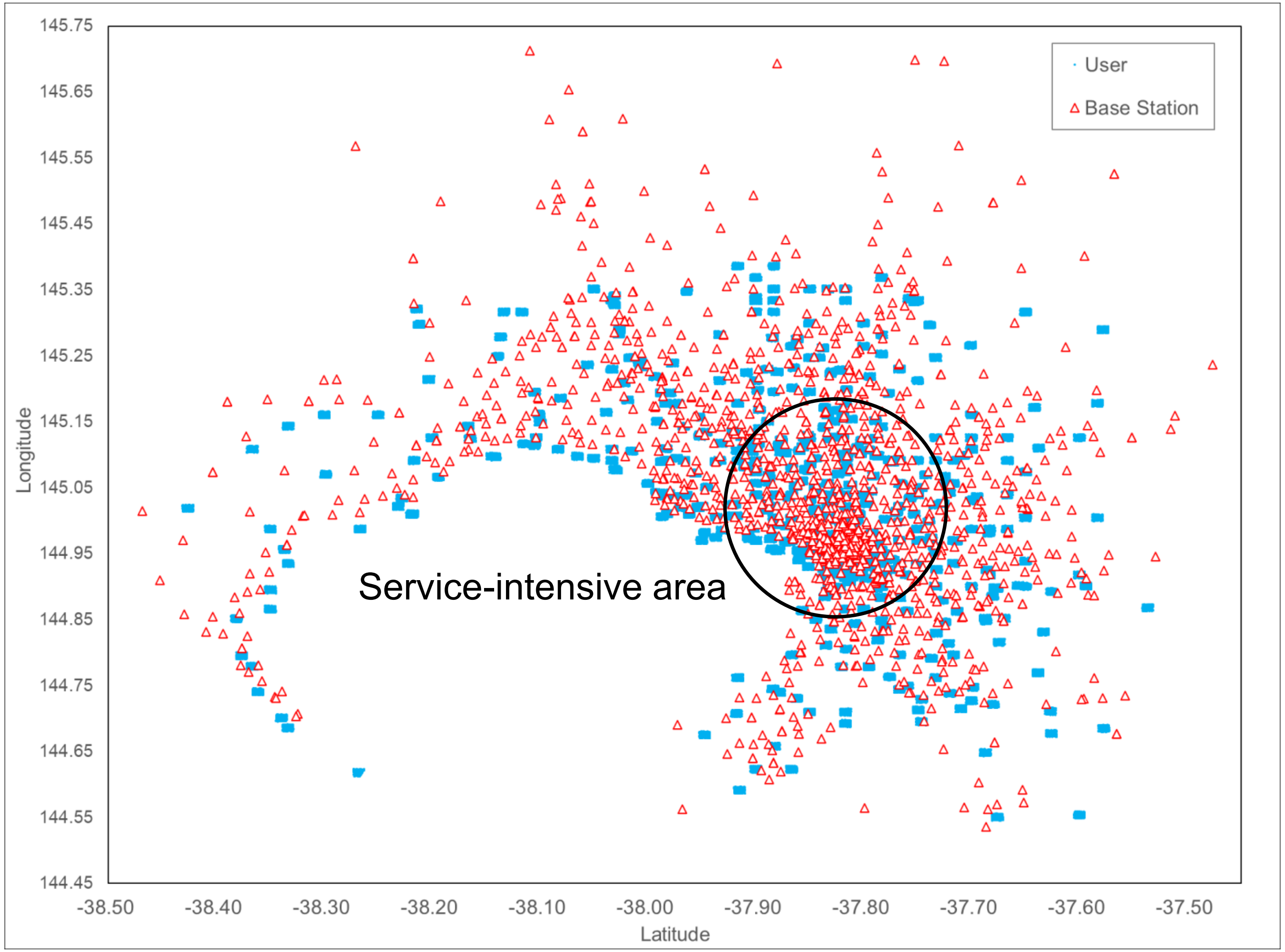
The basic idea:

The server deployment scheme is expanded gradually from the **service-intensive area** to the **service-dispersive area**.

The step of the HR-kESP$_0$ (basic version):

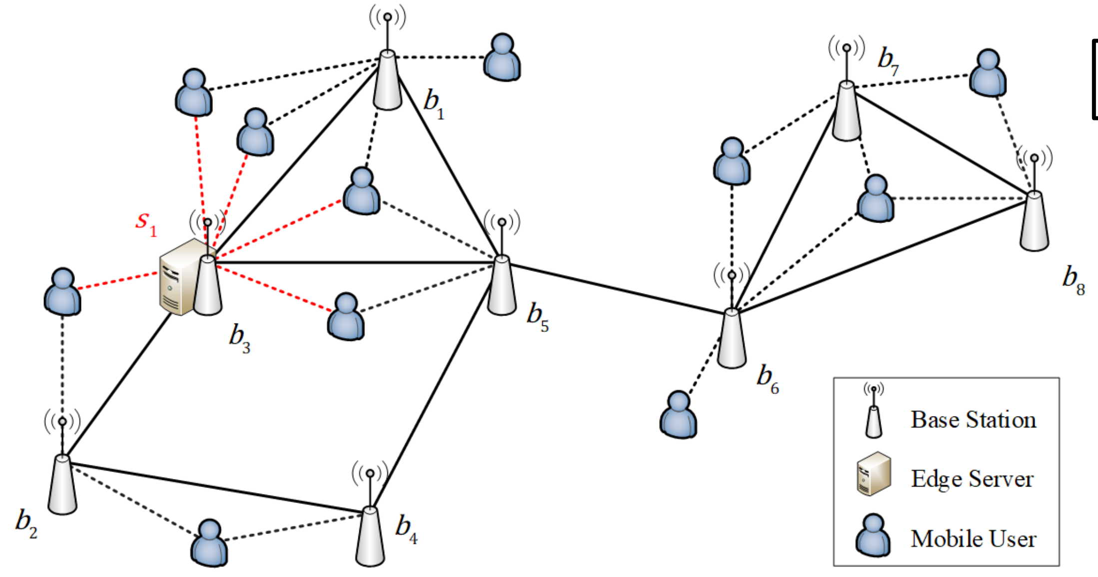
1. Select the **most service-intensive** base station as the starting point of the server deployment scheme.
2. Evaluate the neighbors of selected base station, and the one with the largest robustness increment is selected to expand the server deployment scheme.
3. Repeat step 2 until the number of servers reaches the budget.

Heuristic Information: Robustness Increment

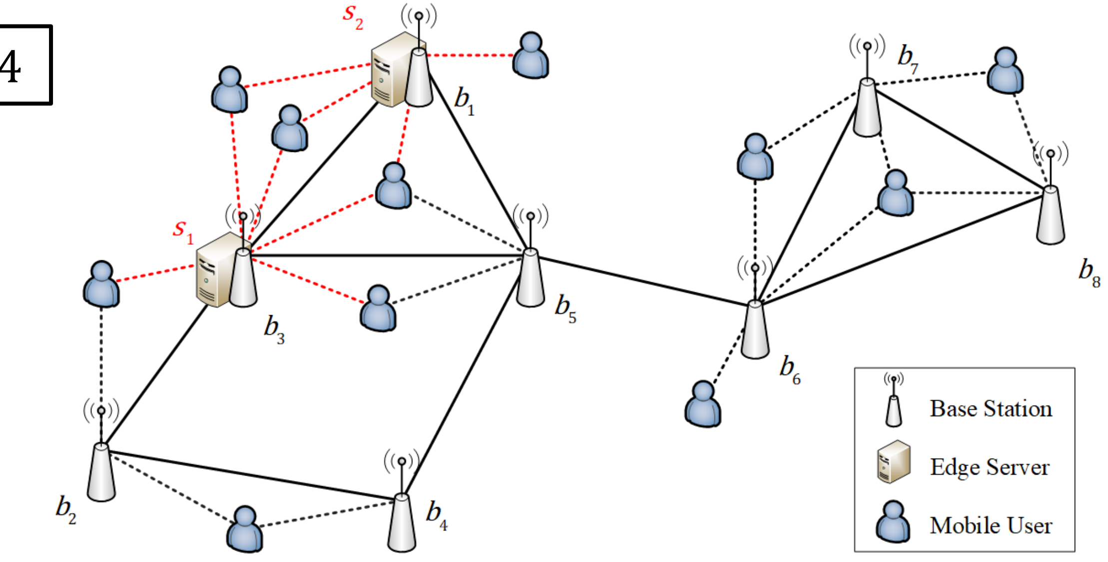


The distribution of users and base stations on the EUA dataset

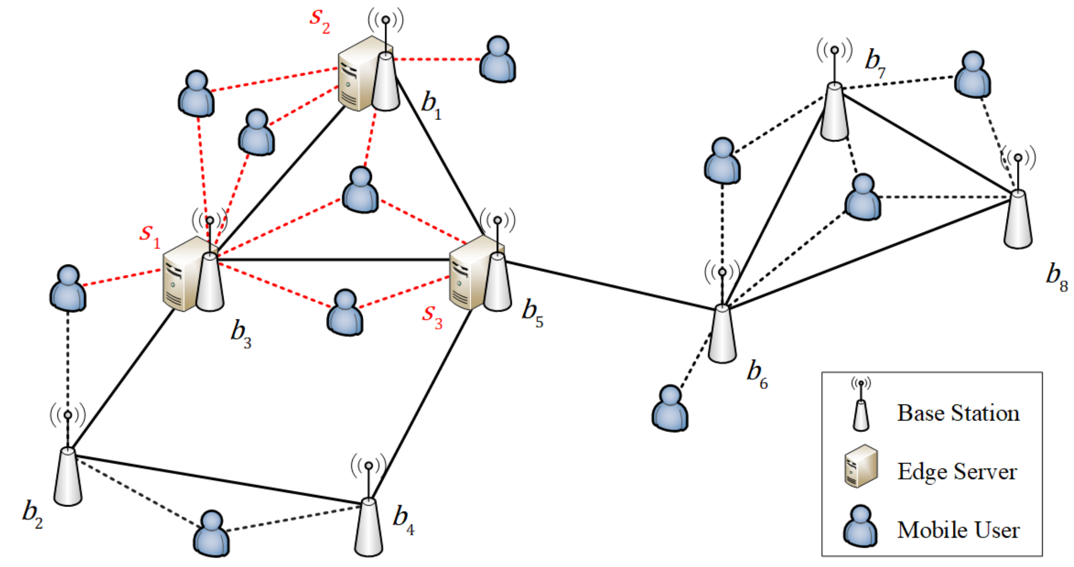
$k = 4$



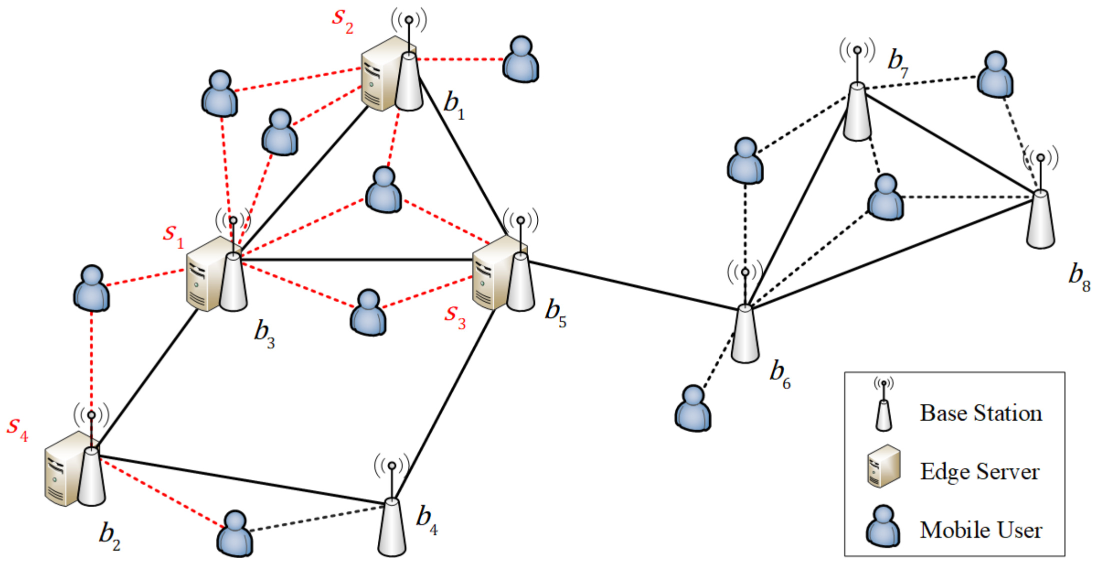
(a) Start with b_3, which is the most serviced intensive.



(b) b_1 has the most robustness increment, among b_3's neighbors $\{b_1, b_2, b_5\}$



(c) b_5 has the most robustness increment, among $\{b_1, b_3\}$'s neighbors $\{b_2, b_4, b_5\}$



(d) b_2 has the most robustness increment, among $\{b_1, b_3, b_5\}$'s neighbors $\{b_2, b_4, b_6\}$

The shortage of HR-kESP$_0$

- It is suitable for the case that there is **only one service-intensive area**. In the actual scenario, there may be **more than one** service-intensive area.

- With the expansion of the server deployment scheme to the service-dispersive area, the **robustness increment** of candidate base station will **decrease gradually**.

Instead of just expanding the server deployment scheme, it is also important to find new service-intensive area.

HR-kESP$_0$ ⇨ **HR-kESP**

The step of the HR-kESP (improved version):

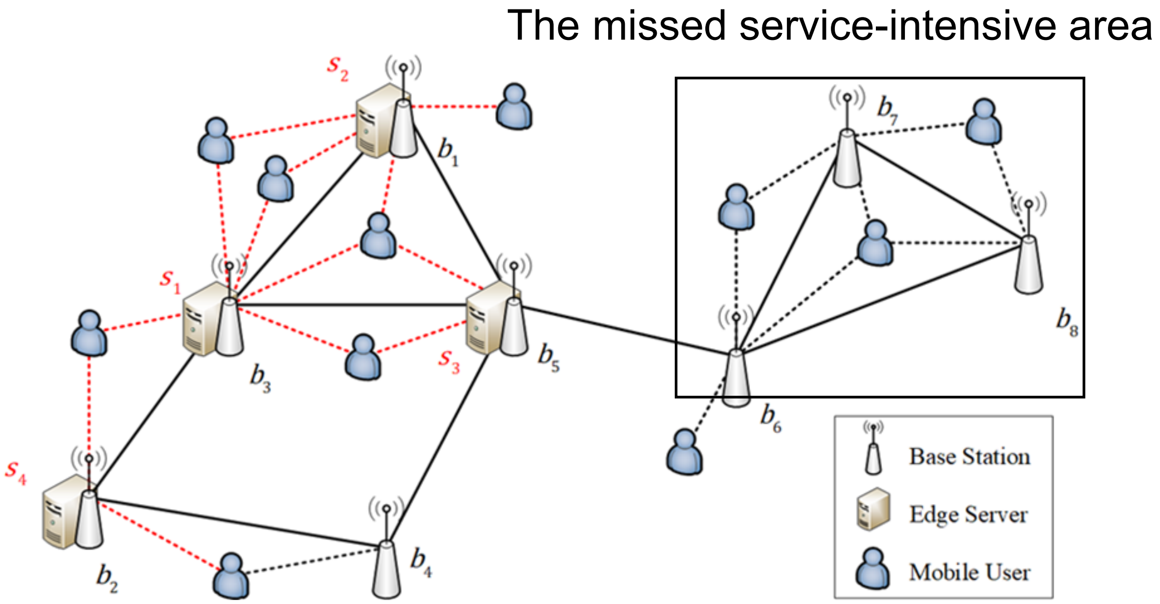
1. Select the most service-intensive base station as the starting point of the server deployment scheme.

2. Evaluate the neighbors of selected base station, and the one with the largest robustness increment as b_{near}.

3. Evaluate the others (unselected base stations except for the neighbors mentioned), and the one with the largest potential robustness increment as b_{far}.

4. Select the best one between b_{near} and b_{far} to extend the server deployment scheme.

5. Repeat steps 2-4 until the number of servers reaches the budget.

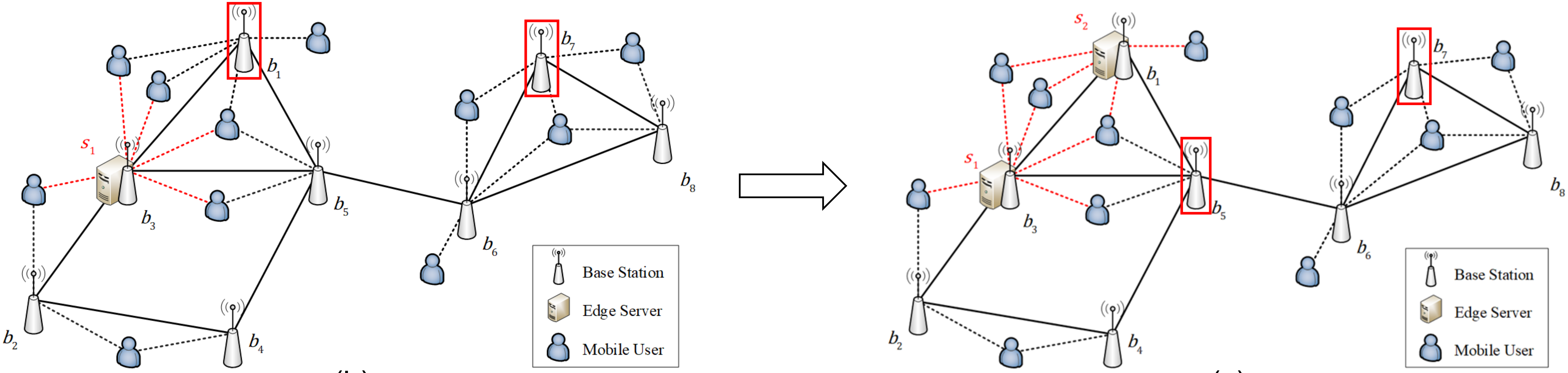


(a) The previous solution of HR-kESP$_0$ misses another service-intensive area.

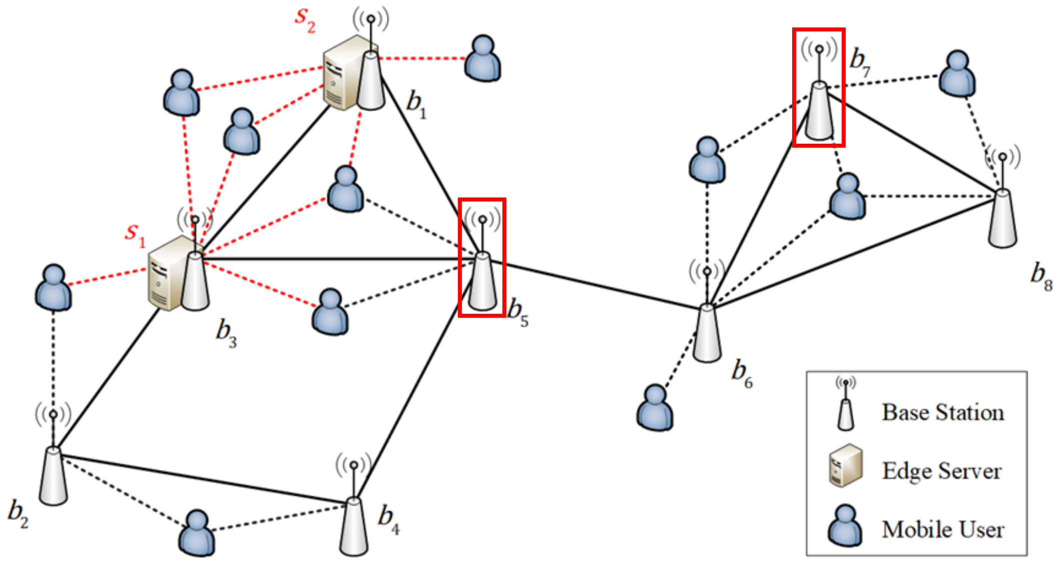
1. The starting base station is still b_3. Its neighbors is $N_{\text{near}} = \{b_1, b_2, b_4, b_5\}$, and others is $N_{\text{far}} = \{b_6, b_7, b_8\}$.
2. Evaluate the base stations of N_{near} and N_{far}, then find the b_{near} which is b_1 and the b_{far} which is b_7.
3. Select the best (b_1) of them to expand the solution.
4. Update the N_{near}, N_{far} and selected b_{near}.

$N_{\text{near}} = \{b_2, b_4, b_5\}$ $\quad N_{\text{far}} = \{b_6, b_7, b_8\}$

$b_{\text{near}} = b_5$ $\quad b_{\text{far}} = b_7$

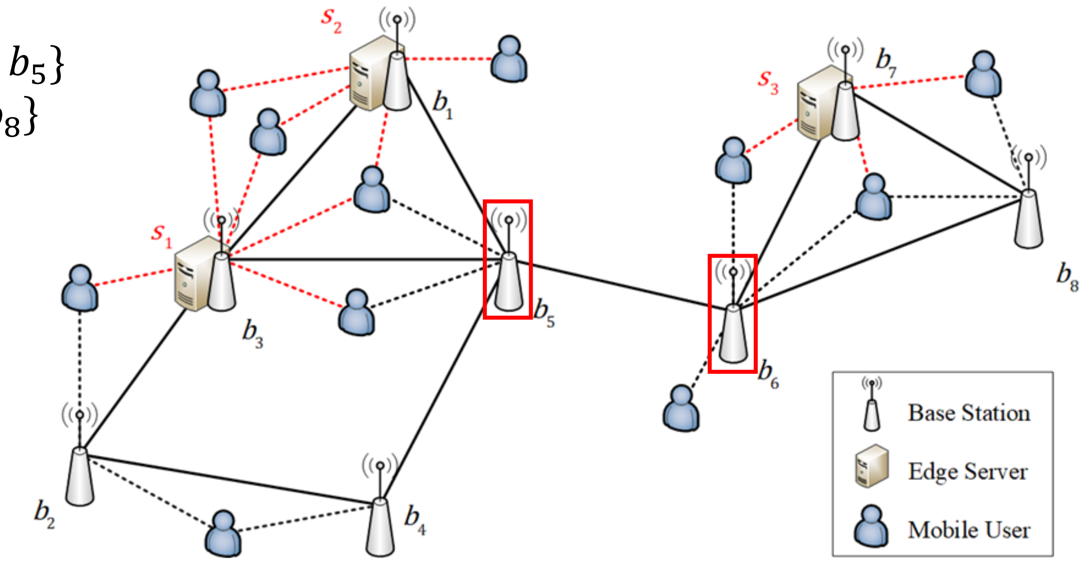


(b) (c)

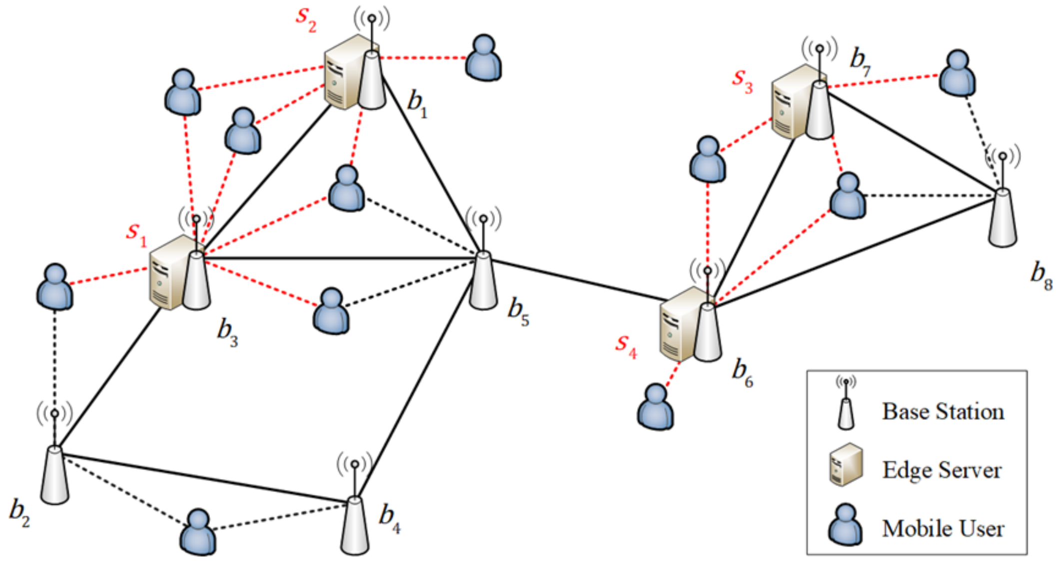


(a)

$N_{\text{near}} = \{b_2, b_4, b_5\}$
$N_{\text{far}} = \{b_6, b_7, b_8\}$
$b_{\text{near}} = b_5$
$b_{\text{far}} = b_7$



(b)



(c)

Since b_7 is better than b_5, there is an opportunity to deploy the third server in another service-intensive area.

According to the same steps, the final selected base station is b_6.

The solution of HR-kESP is $\{b_1, b_3, b_6, b_7\}$.

Experimental Evaluation

The EUA dataset[1] contains the locations of 1465 real-world base stations within metropolitan Melbourne in Australia.

Benchmark algorithm

- *Random*, randomly select k base stations;
- *Greedy*, select the top-k base stations according to the metric $CR(\mathbf{p})$;
- *ESP-A*[2], an approximate solution for kESP problem;
- *HR-kESP$_0$*, the basic version of HR-kESP.

The network robustness evaluation method

- Some servers are randomly shut down to simulate the sudden failures.
- The default proportion of failed servers is 50%.
- **User Survival Rate** is the measure of network robustness, which is the ratio of the number of the covered users after some server failures to its numbers before.

1. https://github.com/swinedge/eua-dataset
2. Cui, G., He, Q., Chen, F., et al.: Trading off between user coverage and network robustness for edge server placement. IEEE Transactions on Cloud Computing, 10(3), 2178-2189 (2022).

Table 1. Experimental Parameter Settings.

Parameters	m	k	n
Group 1	5, 10, . . . , 35	4	80
Group 2	20	1, 2, . . . , 8	80
Group 3	20	4	20, 40, . . . , 160

- m, the number of base stations
- k, the servers budgets
- n, the number of users

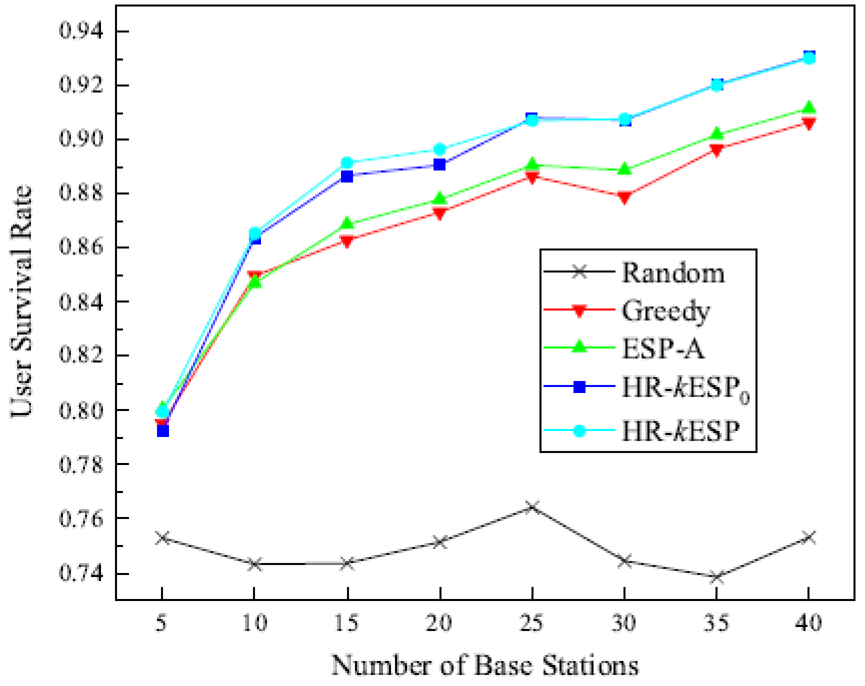


Fig. 3. User Survival Rate after Failures with Different Numbers of Base Stations.

Table 2. The Performance ($CR(\mathbf{p})$) of Algorithm with Different Numbers of Base Stations.

Algorithm\\m	5	10	15	20	25	30	35	40
Random	185.86	174.18	178.29	180.18	198.11	180.05	168.11	184.35
Greedy	239.02	350.36	393.46	429.27	472.57	466.55	495.26	514.79
ESP-A	228.72	345.92	398.10	436.62	478.43	476.33	506.70	522.41
HR-kESP$_0$	233.34	350.42	400.68	436.08	**493.53**	489.34	518.96	**531.86**
HR-kESP	**235.38**	**353.79**	**404.71**	**441.10**	491.34	**490.65**	**520.93**	530.33

The Result of Group 1

Table 3. The Performance ($CR(\mathbf{p})$) of Algorithm with Different Server Budgets.

Algorithm\\k	1	2	3	4	5	6	7	8
Random	31.68	62.32	112.29	186.74	287.55	426.51	569.98	763.73
Greedy	52.02	105.01	237.88	433.37	693.00	929.68	1244.87	1597.61
ESP-A	**53.00**	**106.20**	246.50	441.28	704.48	933.31	1248.36	1570.70
HR-kESP$_0$	**53.00**	105.62	**251.01**	449.66	716.03	967.16	1272.51	1603.55
HR-kESP	51.91	105.49	250.49	**455.72**	**720.87**	**982.07**	**1286.63**	**1621.23**

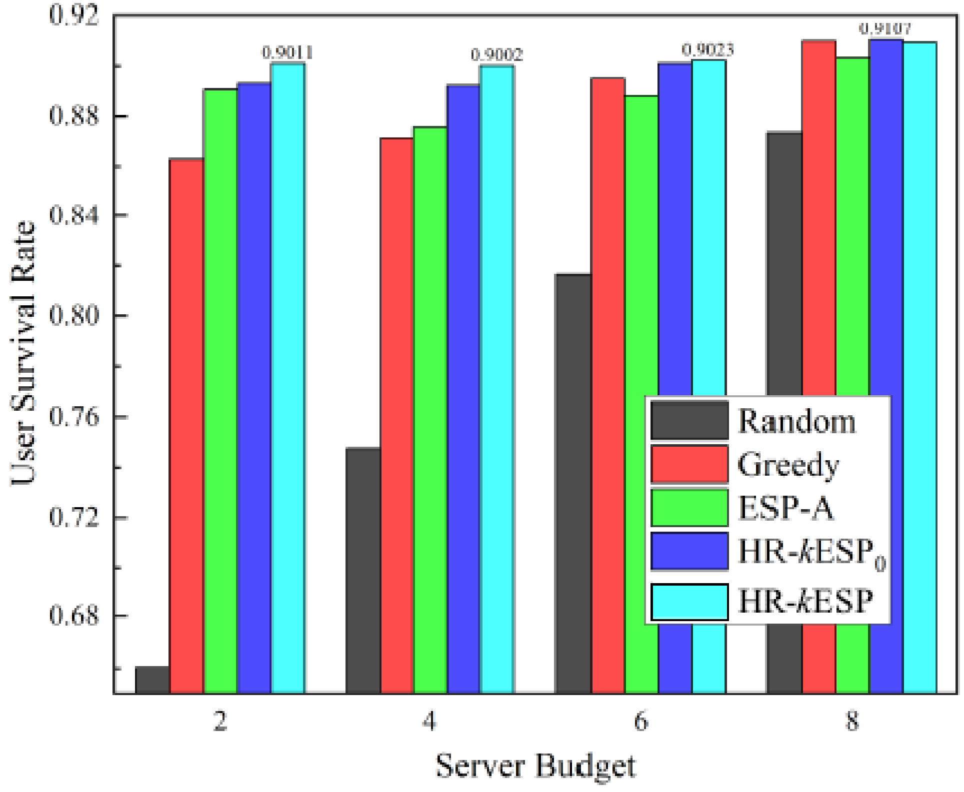


Fig. 4. User Survival Rate after Failures with Different Server Budgets.

The Result of Group 2

Table 4. The Performance ($CR(\mathbf{p})$) of Algorithm with Different Numbers of Users.

Algorithm\\n	20	40	60	80	100	120	140	160
Random	43.65	91.16	141.12	188.31	219.96	253.56	309.23	350.02
Greedy	118.34	221.56	323.57	434.84	533.93	652.92	737.79	849.94
ESP-A	119.94	226.03	329.31	440.27	546.91	666.93	751.18	868.78
HR-kESP$_0$	**123.61**	231.40	335.98	446.82	554.25	671.64	**776.13**	880.68
HR-kESP	122.76	**234.73**	**337.69**	**447.61**	**558.88**	**675.64**	771.60	**890.16**

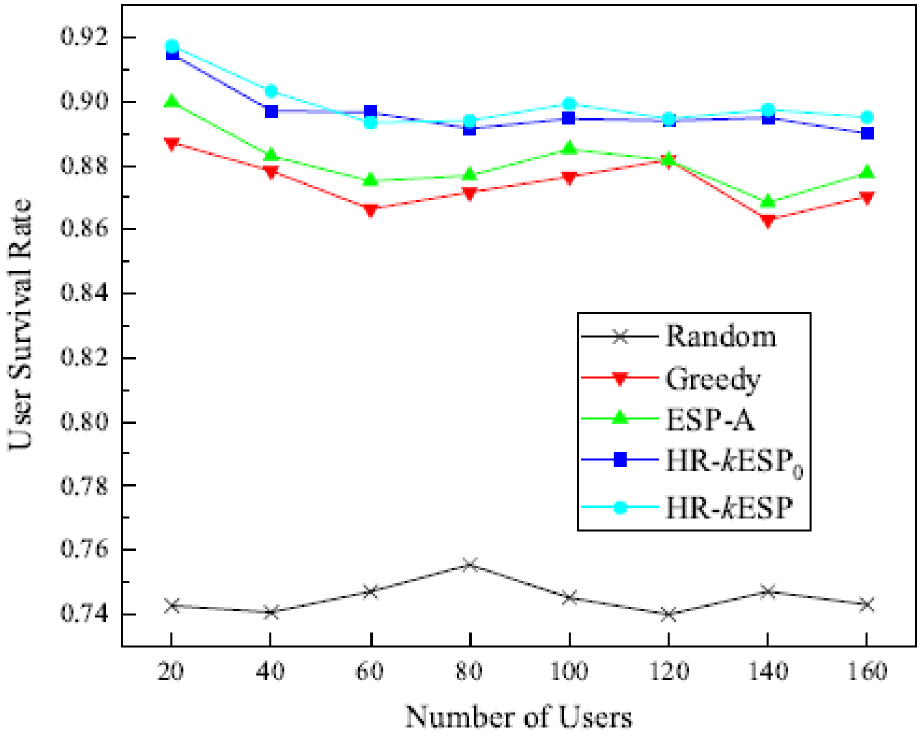


Fig. 5. User Survival Rate after Failures with Different Numbers of Users.

The Result of Group 3

Conclusion

- For the **Robustness-oriented kESP** problem, a heuristic algorithm named **HR-kESP** is proposed.

  According to the **distribution** of the base stations and the users, from the service-intensive area to the service-dispersive area , the server deployment scheme is expanded gradually by one of **two candidates** which are respectively selected from two subsets (adjacent or non-adjacent base stations).

- A **evaluation method** for the **Network Robustness** is designed.

  A metric named User Survival Rate is used to measure the network robustness. Assume the original number of covered users is n_{before}. After some server failures, the number of covered users is n_{after}.

$$User\ Survival\ Rate = \frac{n_{after}}{n_{before}}$$

- The experiments are conducted on the **public EUA dataset**, and the results indicate that HR-kESP outperforms other benchmark algorithms in most cases.

Thank you for your listening!