NETLAB
NETWORKED SYSTEMS RESEARCH LABORATORY
University of Glasgow | School of Computing Science

School of Computing Science
Essence: Pervasive & Distributed Intelligence

University of Glasgow

Task offloading in Mobile Edge Computing: An Optimal Stopping Theory approach

Essence Lab Talk, Thursday 4 March 2021

Ibrahim Alghamdi

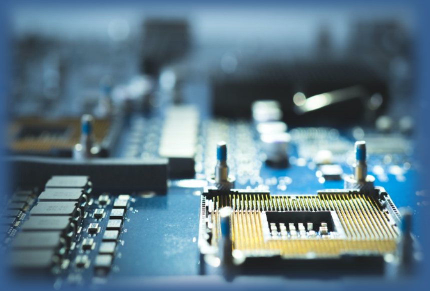
Outline

- Introduction
  - Background
  - Motivation & Challenge
  - Related work and contribution
- Task offloading decision making
  - System Model
  - Problem Formulation
  - Maximizing the Probability of Offloading to the Best Server
  - Minimizing the Expected Total Delay of Task Offloading
- Performance evaluation
  - Simulation
  - Real data set evaluation
  - Real implementation
- Future work

New forms of mobile nodes

The Requirements of the Emerging Applications

- Require higher computing/networking resources:
  - Latency-sensitive applications (virtual reality)
  - Powerful CPUs (data analytics using machine learning)
  - Need more storages (sensing and collecting data)
- These requirements contradict with the mobile nodes capabilities.



Computation offloading

- Sending the computing task to an external server.
- The Cloud was the initial place for offloading.
- The Mobile cloud computing.
- Higher cost;
  - More delay.
  - More load on the network.

From the Cloud to the Edge

- Move the Cloud resources closer to the user.
- There different names in the literature:
  - edge computing.
  - cloudlet.
  - fog computing.
- *Mobile Edge Computing*.

Motivation

- The deployment of MEC servers.[1]
- MEC servers' load have large variation.[2]

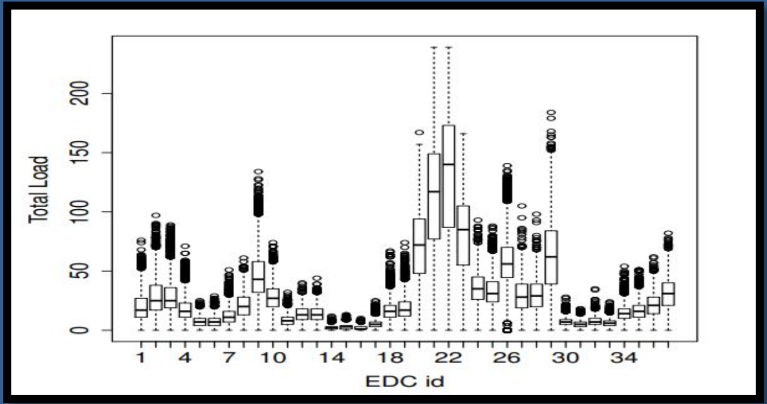


Figure 1: Workload in 37 EDCs according to the simulation in [2]

[1] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal et al., "Mobile-edge computing introductory technical white paper," *White Paper, Mobile-edge Computing (MEC) industry initiative*, 2014.
[2] C. N. Le Tan, C. Klein, and E. Elmroth, "Location-aware load prediction in edge data centers," *in 2nd FMEC*. IEEE, 2017, pp. 25–31.

Motivation Example: MEC in RSU

- *Autonomous, Smart Vehicles:*

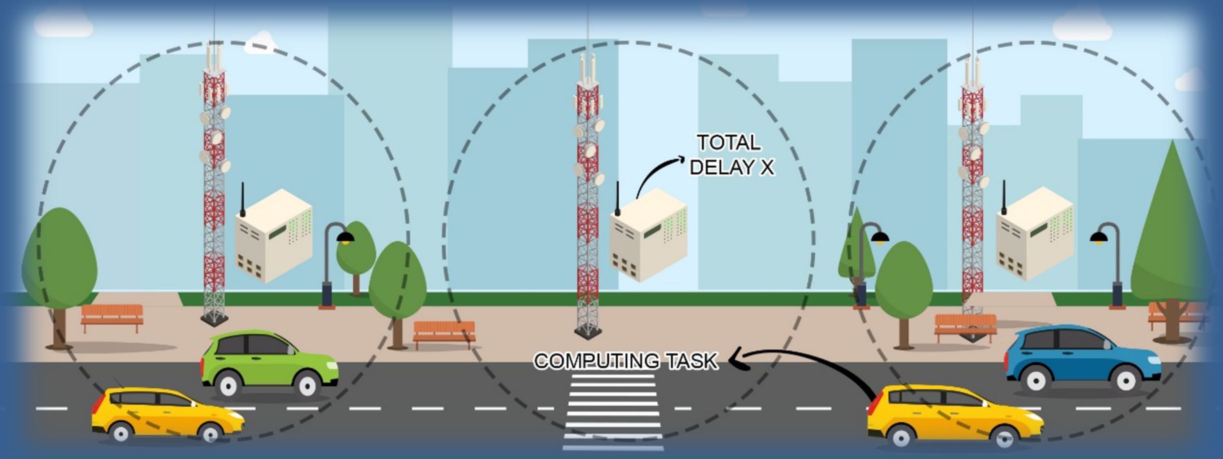
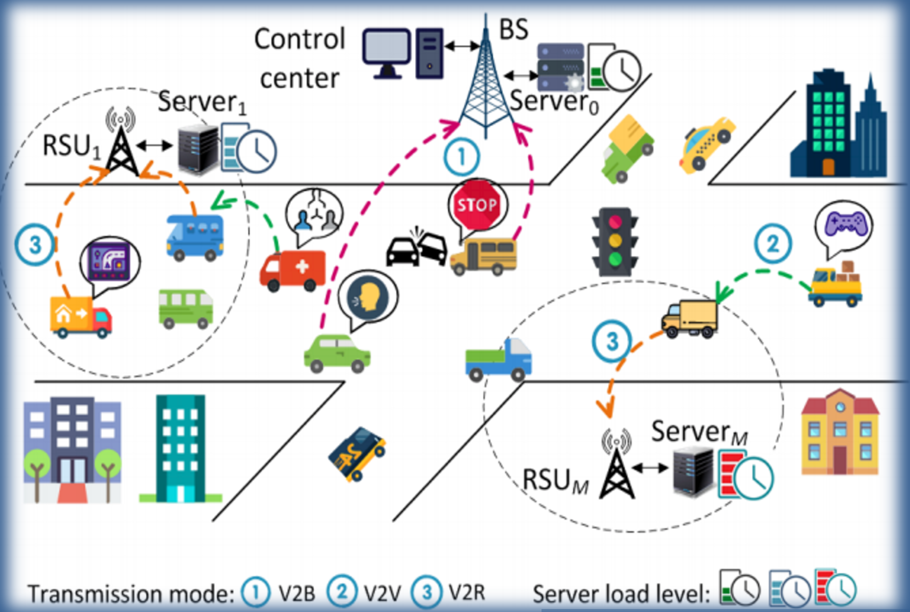


Figure 2: MEC environment. [3,4]

[3] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading,"IEEE Vehicular Technology Magazine, vol. 12,no. 2, pp. 36–44, 2017.

[4] R. Akmam Dziyauddin, D. Niyato, N. Cong Luong, M. A. M. Izhar,M. Hadhari, and S. Daud, "Computation offloading and content caching delivery in vehicular edge computing: A survey,"arXiv, pp. arXiv–1912,2019.

Naïve Approach



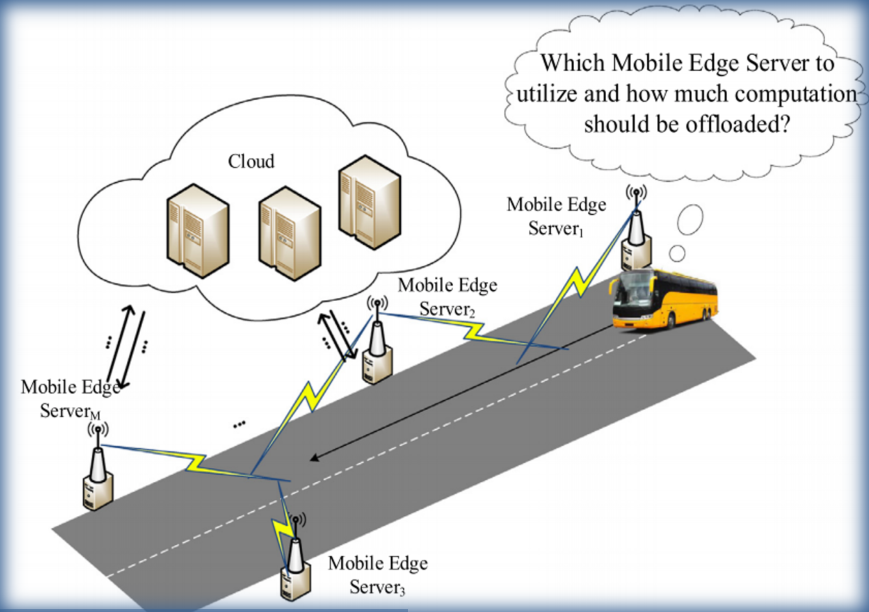


Figure 3: Centralised offloading. [5,6]

[5] W. Tang, X. Zhao, W. Rafique, L. Qi, W. Dou, and Q. Ni, "An offloading method using decentralized p2p-enabled mobile edge servers in edgecomputing,"Journal of Systems Architecture, vol. 94, pp. 1–13, 2019.
[6] K. Zhang, Y. Zhu, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Deep learning empowered task offloading for mobile edge computing in urban informatics,"IEEE Internet ofThings Journal, vol. 6, no. 5, pp. 7635–7647, 2019

V2V Approach

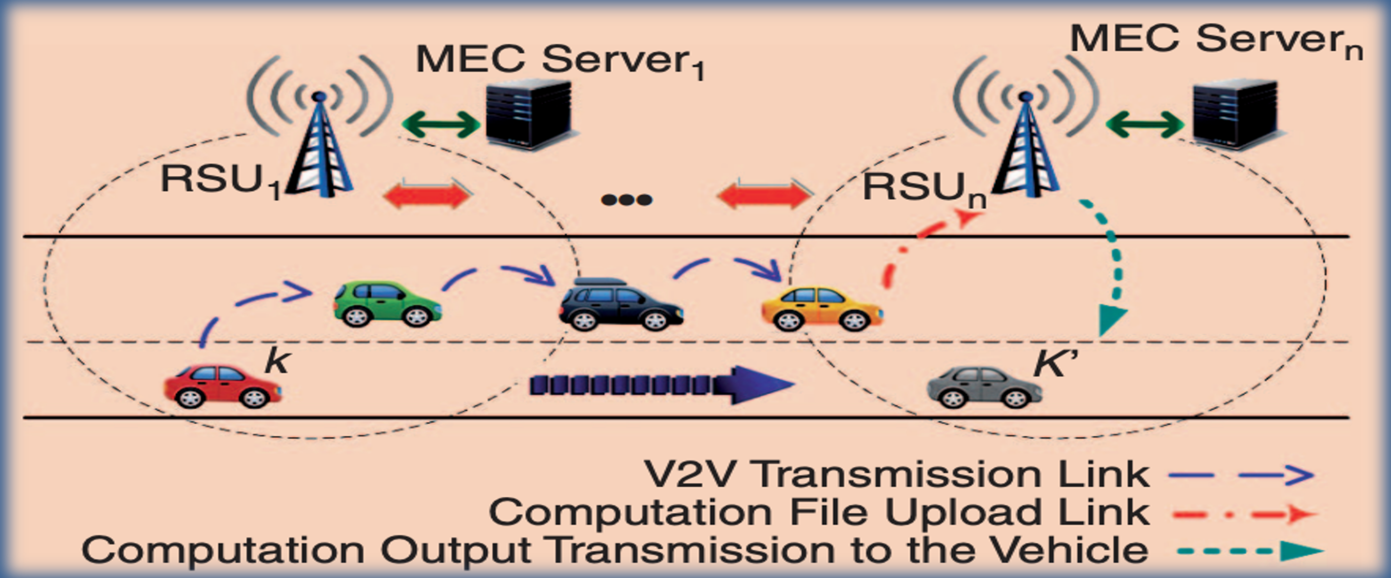


Figure 4: V2V offloading method. [3]

[3] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, “Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading,”IEEE Vehicular Technology Magazine, vol. 12,no. 2, pp. 36–44, 2017.

Contributions

- Offloading decision.
  - Independent
- Considerations:
  - Mobility:
    - Higher chance of meeting better resources. [7]
  - Deadline:
    - We must offload before T. [5]
  - Sequential:
    - Optimality found in the optimal stopping theory

[7] S. Zhou, Y. Sun, Z. Jiang, and Z. Niu, "Exploiting moving intelligence: Delay-optimized computation offloading in vehicular fog networks,"IEEE Communications Magazine, vol. 57, no. 5, pp. 49–55, 2019.
[5] W. Tang, X. Zhao, W. Rafique, L. Qi, W. Dou, and Q. Ni, "An offloading method using decentralized p2p-enabled mobile edge servers in edgecomputing,"Journal of Systems Architecture, vol. 94, pp. 1–13, 2019.

Task offloading decision-making

- System Model
- Problem formulation
- The proposed models

Setting/system model [3]

- MEC servers deployed along the user path.
- Mobile node moves in 1D mobility model.
- Computing task to be offloaded to one of the MEC servers.
- The mobile node only knows about the current MEC (the one in the range of mobile node).
- Processing time X.

Figure 5: Context

[3] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," IEEE Vehicular Technology Magazine, vol. 12, no. 2, pp. 36–44, 2017.

Problem Statement (1)

- A key problem:
  - Deciding which server to select?
- Giving
  - we have load variance for edge servers over time,
  - users are moving, and knows only about the server in the range of it.
- Applying the Optimal Stopping Theory

Problem Statement (2)

Objective 1: Maximizing the Probability of Offloading to the Best Server.

Objective 2: Minimizing the Expected Execution Time when Offloading.

Specifically: find an offloading rules that achieve the previous two objectives.

Maximizing the Probability of Offloading to the Best Server (1)

- Goal:
  - Max (P_n^*)
- Assumption:
  - We know the number of options servers/times.
  - No recall is allowed.
- This is cast as a Best-Choice Problem (BCP) [8] .

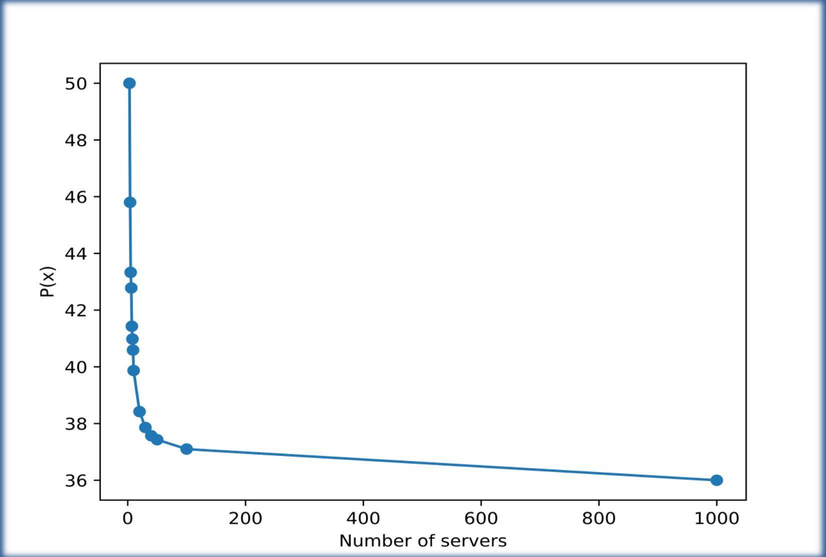
[8] T. S. Ferguson, "Optimal Stopping and Applications," http://www.math.ucla.edu/ tom/Stopping/Contents.html, March 2019.

The Offloading Rule

- Let M be the best server among r_n-1 servers.
- Based on the BCP, the optimal offloading policy is to
  - reject the first $r_n - 1$ servers (times).
  - offload first server that is better than M.
- $r_n = \min\{r \geq 1: \frac{1}{r} + \frac{1}{r+1} + \ldots + \frac{1}{n-1} \leq 1\}$ for n ≥ 2 ..(1)
- $P^*(r_n) = \frac{r_n - 1}{n} \sum_{k=r_n}^{n} \frac{1}{k-1}$ ...(2)
- In the case where there is a relatively high number of servers, r=n/e and the probability is around 0.368. [8]

[8] T. S. Ferguson, “Optimal Stopping and Applications,” http://www.math.ucla.edu/ tom/Stopping/Contents.html, March 2019.

Cont'd



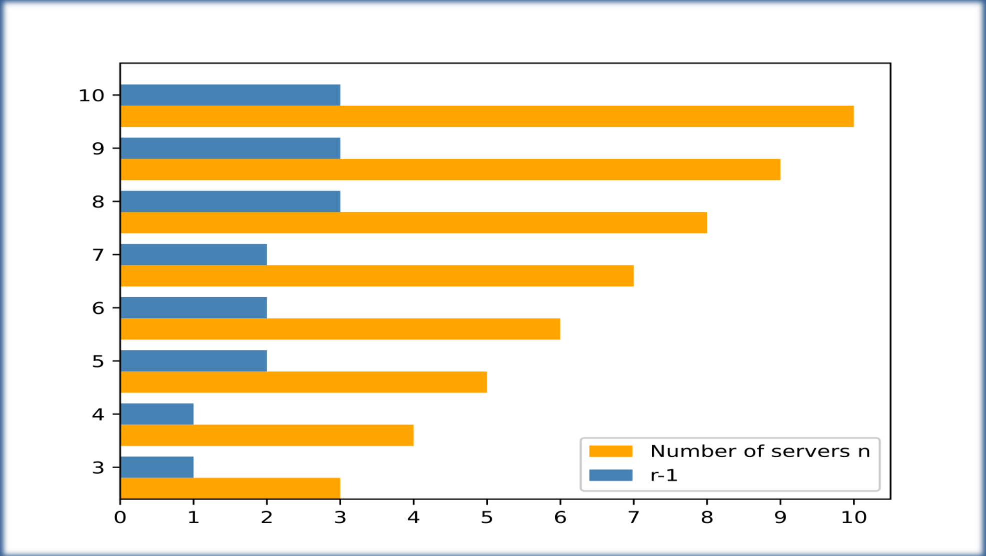


Figure 6: The probability of offloading to the best (left) and the value of *r−1*(right) for different numbers of MEC servers *n*.

[8] T. S. Ferguson, "Optimal Stopping and Applications," http://www.math.ucla.edu/ tom/Stopping/Contents.html, March 2019.

Maximizing the Probability of Offloading to the Best Server (2)

- Goal:
  - Max (P_n^*)
- Assumption:
  - We know the probability distribution function of X.
    - MEC server operator
    - historical data
  - Data quality indicator:
    - $f_k = \begin{cases} 1 - \frac{k}{n+1}, & 1 \leq k < n \\ 0, & k \geq n \end{cases}$
    - 1 is fresh (when k=0), 0 is very old data (when k=n)
  - No recall is allowed.
- Odd-sum model [9] .

[9] F. T. Bruss, "Sum the odds to one and stop," Annals of Probability, pp. 1384–1391, 2000.

Odds Algorithm

- Odds algorithm:
  - maximise the probability of stopping at the last success.
- Offload to MEC server with specific threshold.
  - For example, the mobile node needs processing time less than 50 ms.
- The Odds in general:
  - $r_k = \frac{P_k}{1 - P_k}$
- The Odds in our case:
  - $r_k = \frac{P_k f_k}{(1 - P_k) f_k}$

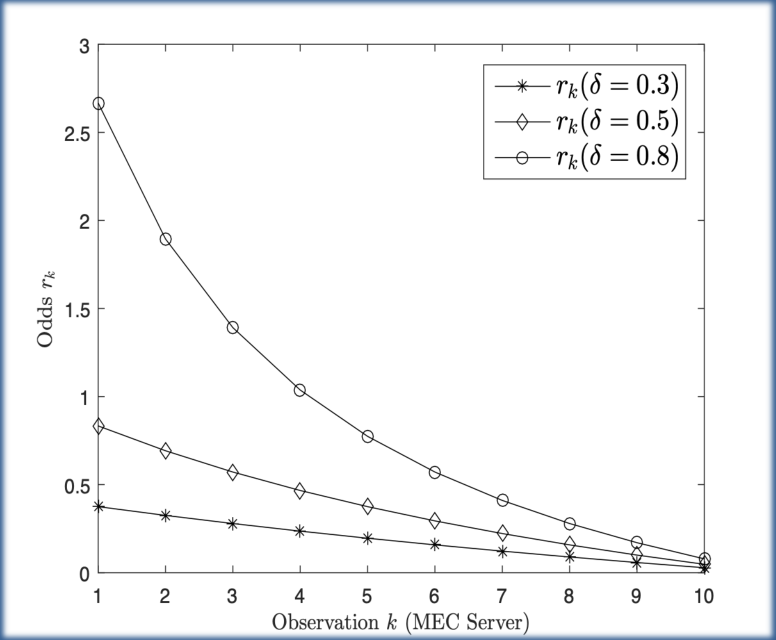


Figure 7: The odds against observation.

So what are the offloading rules now?

- The Odds-algorithm sums up the Odds in reverse order:
  - $r_n + r_{n-1} + r_{n-2} + \cdots + r_s$
- **Let us denote that this happens at observation s:**
  - $R_s = r_n + r_{n-1} + r_{n-2} + \cdots + r_s$
- **Example:**
  - **s = 4 in the**
  - **s =1**
- **The offloading rule:**
  - **reject all observation before s**
  - **After s, start looking for the server that meets the requirements.**

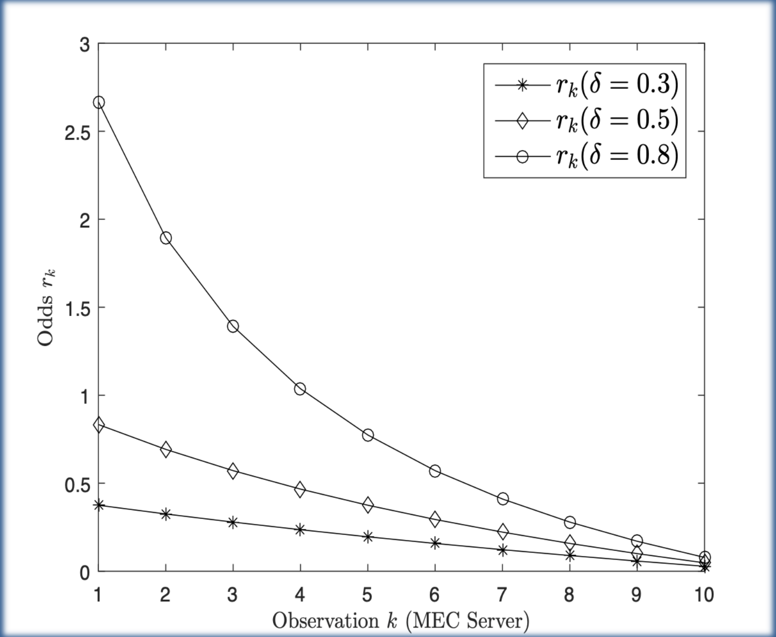


Figure 7: The odds against observation.

Minimising the Expected Processing Time (1)

- Delay-Tolerant Sequential Decision Making (DTO)
- Assumption:
  - We have an idea about the the load (execution time) of the MEC servers, i.e. X.
  - We know the number of options servers/times.
- Finite Horizon Optimal Stopping Problem

Cont'd

- Find the optimal stopping time:
  - $k = \inf\{k: X_k < \mathbb{E}[X_{k+1}]\}$
- At each observation take the minimum between:
  - current processing time
  - or the expected processing time in the next time
- We provide an estimate of the optimal offloading time.
- The optimal offloading time is determined by the values $a_1, a_2, \ldots, a_n$ by which the mobile node decides either to offload or not.

Cont’d

- The values of the threshold a is calculated through the backward indication starting from the last observation.

$$a_k = \frac{1}{1+r}\left(a_{k+1}(1 - F(a_{k+1})) + \int_0^{a_{k+1}} u dF(X)\right)$$

$$a_n = \frac{1}{1+r}\int_0^1 u dF(X) = \frac{1}{1+r}\mathbb{E}[X]$$

Cont'd

- The decision values (black points).
- Simulated server processing time (blue points) vs.
- The optimal data offloading time when k= 27,29,46,47,48and 50 where X < a.
- We offload at k=27

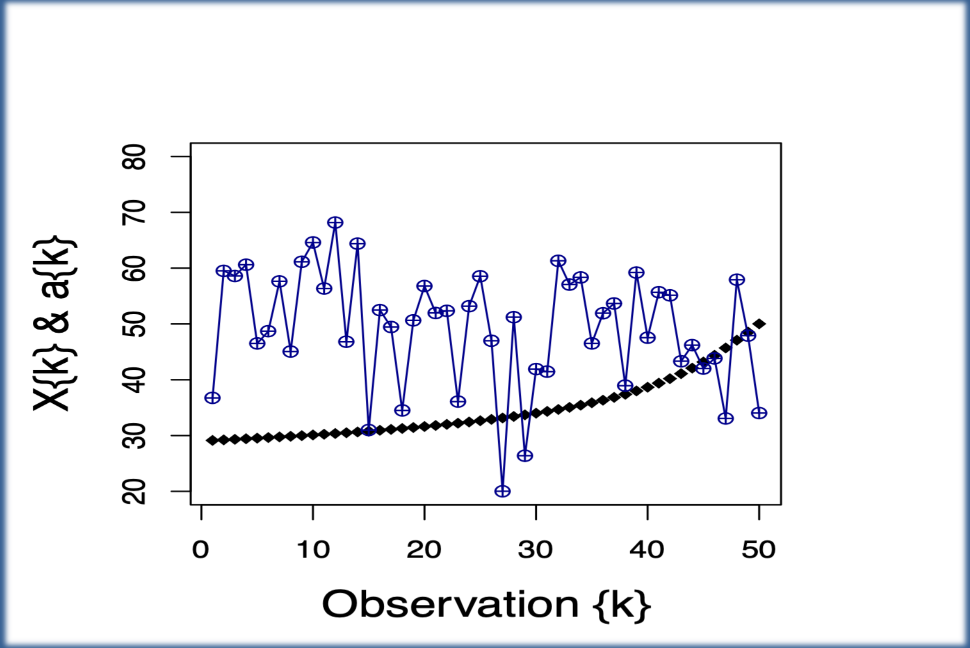


Figure 8:The decision values a and X vs observation k

Minimising the Expected Processing Time (2)

- Cost-based Optimal Task Offloading Policy (COT)
- Goal:
  - We desire to find when to offload and which server that minimizes the total expected cost.
- Assumption:
  - We have an idea about the the load of the MEC servers, i.e. X.
  - The mobile node pays c cost units per observation when it has not yet offloaded the task/data.

$$\boldsymbol{Y = X + ck}$$

Cont'd

$$Y = X + ck \ldots (1)$$

- The node minimizes the expected cost Y by offloading at the first server such that:

$$k^* = \min\{k > 0 : X_k \leq V^*\} \ldots (2)$$

where the V^* is the solution of:

$$\int_{V^*}^{\infty} (x - V^*)\, dF(x) = c \ldots (3)$$

- *where* F(x) *is the CDF of X.*

Cont’d

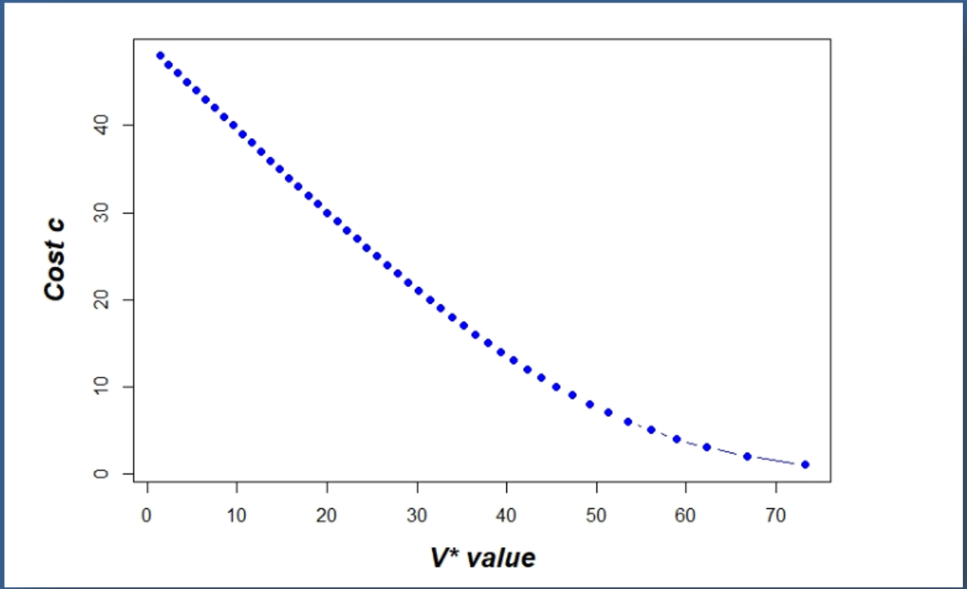


Figure 9: The V value vs. cost c for X with Mena = 50 and SD = 10.

RECAP!

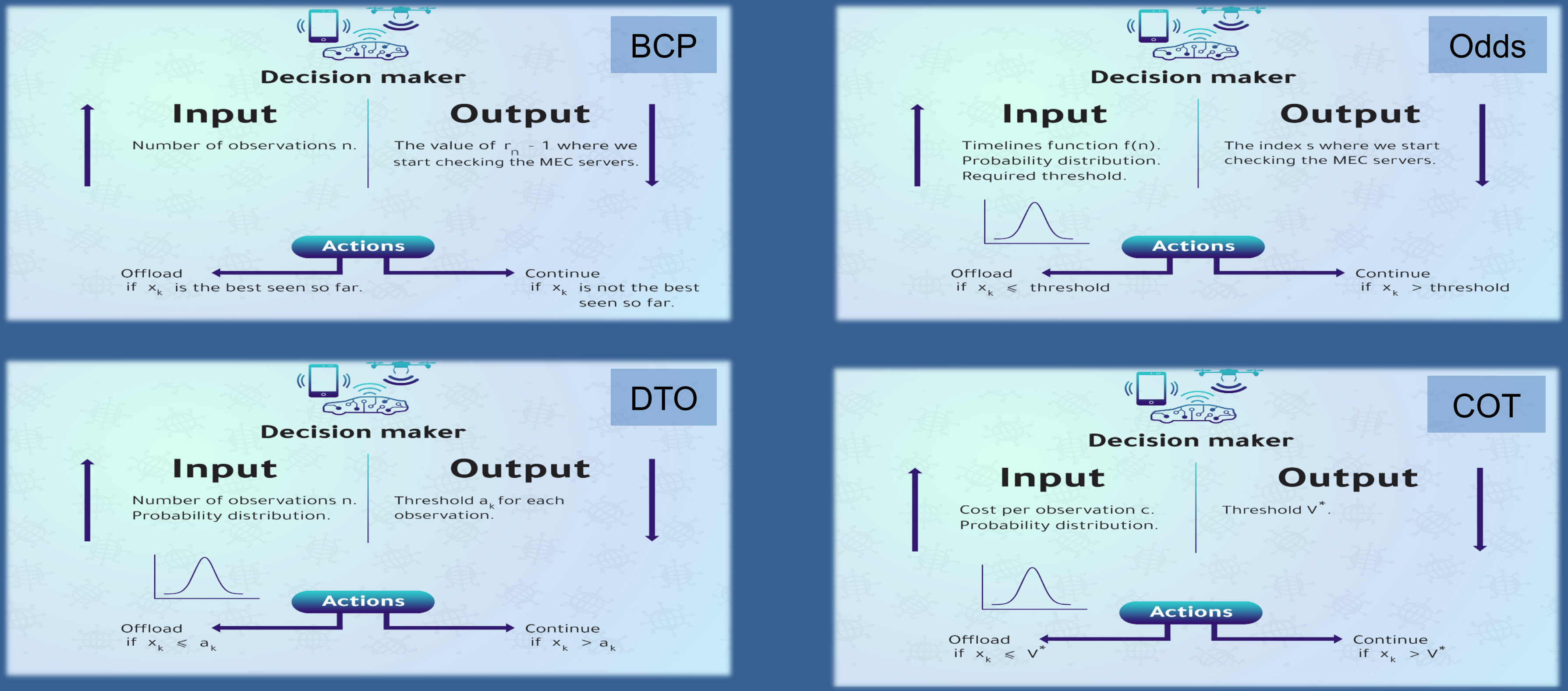


Figure 10: Summary of the OST based model.

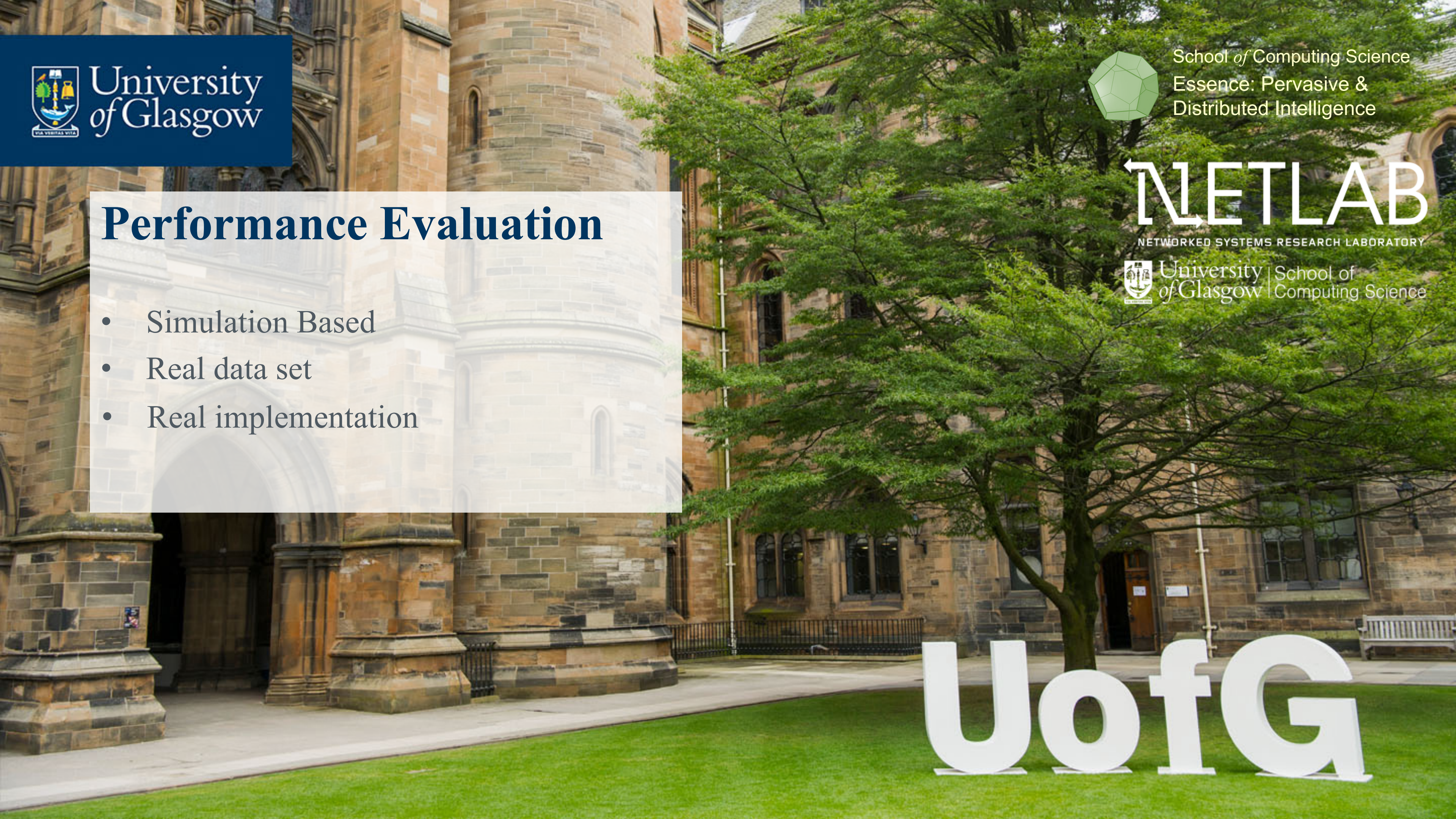
University of Glasgow

School *of* Computing Science
Essence: Pervasive & Distributed Intelligence

NETLAB
NETWORKED SYSTEMS RESEARCH LABORATORY
University of Glasgow | School of Computing Science

Performance Evaluation

- Simulation Based
- Real data set
- Real implementation



Performance Assessment

- Approaches:
  - Simulation Based
  - Real data set
  - Real implementation
- Comparison:
  - Best Choice Problem (BCP)
  - Odds
  - Delay-Tolerant Sequential Decision-making (DTO)
  - COT
  - Random.
  - p-model with different probability $p=0.8$
  - ***The optimal.***

Performance Evaluation (1): Simulation

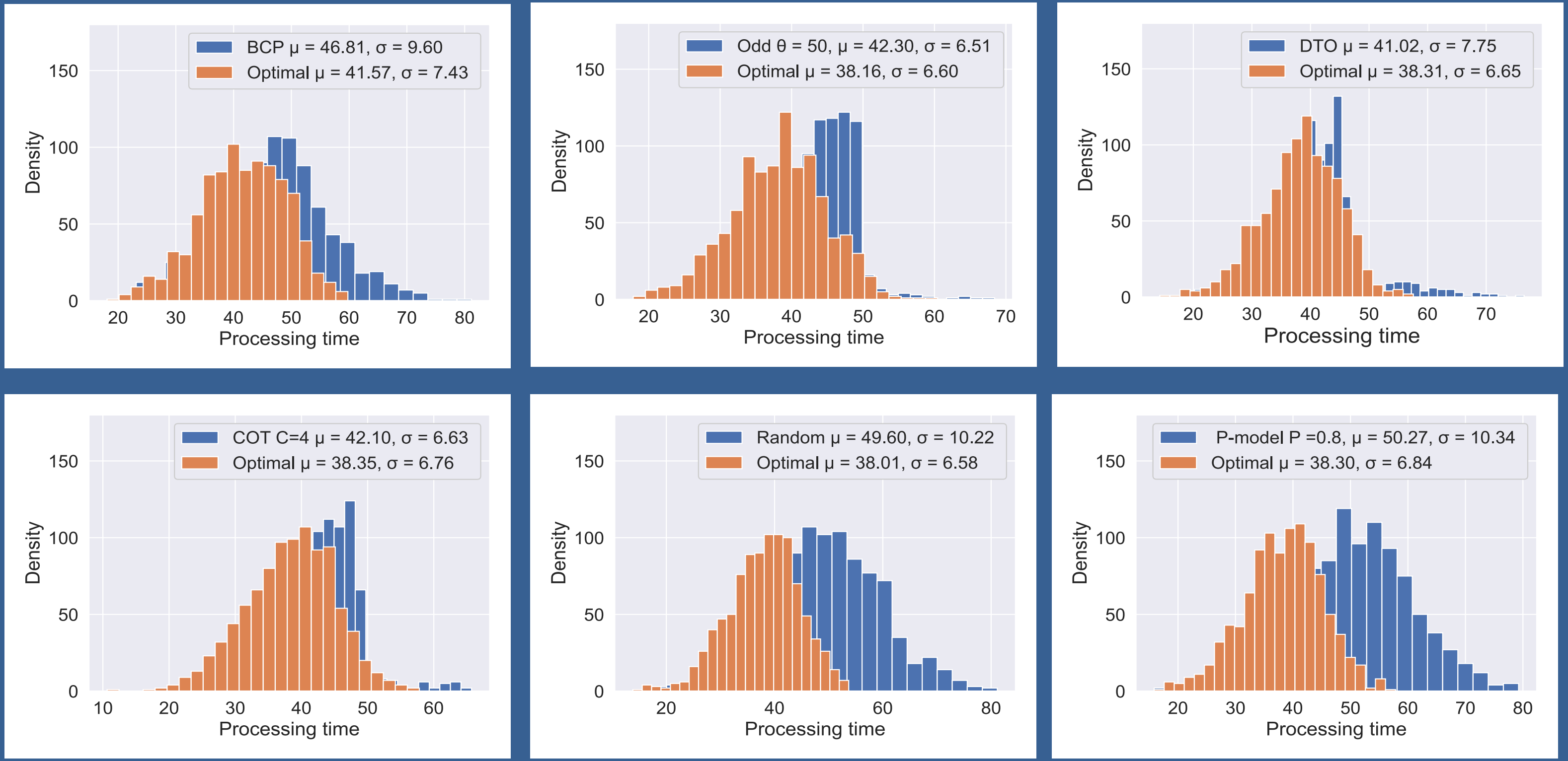


Figure 11: Simulation results for normally distributed processing time.

Cont’d

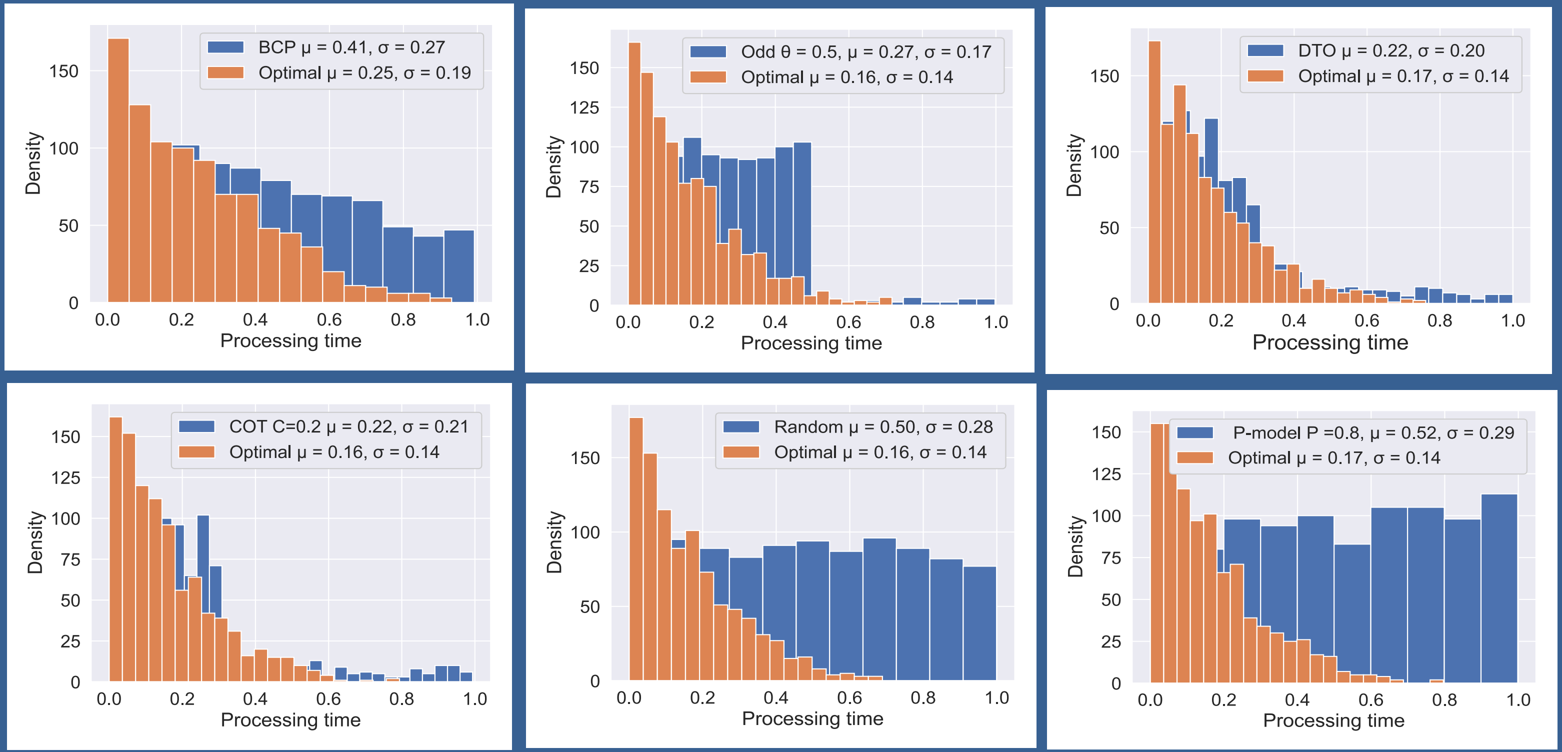


Figure 12: Simulation results for uniformly distributed processing time.

Performance Evaluation (2): data set

- We used the real dataset of taxi cabs' movements in Rome [1].
- Real Server utilisation [2]

Cap id	Movement time	Location	Machine name	CPU utilization
156	2014-02-05 00:11:01	(41.8911, 12.49073)	m_1939	(51)
156	2014-02-05 00:11:11	(41.89905,12.4899)	m_1936	(47)
156	2014-02-05 00:11:22	(41.8994,12.48940)	m_1941	(20)
156	2014-02-05 00:11:31	(41.8994,12.489401)	m_1941	(37)

Table 5.2: A sample of the data set used in the experiment.

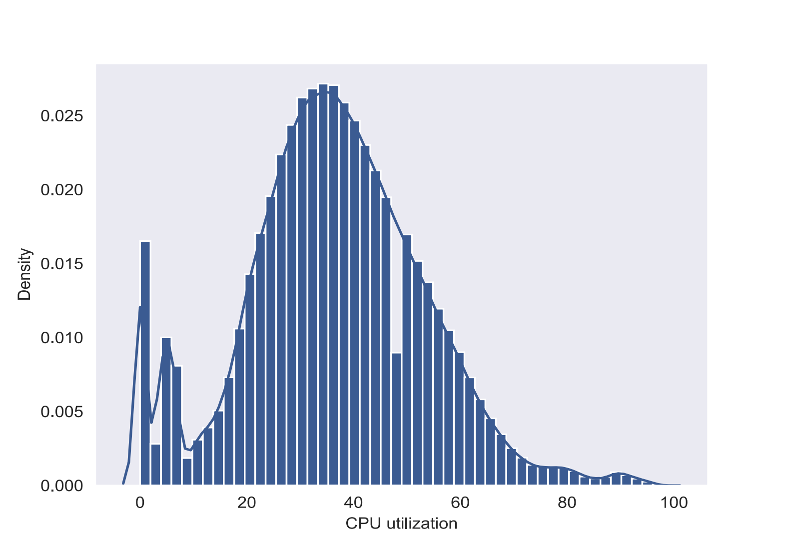
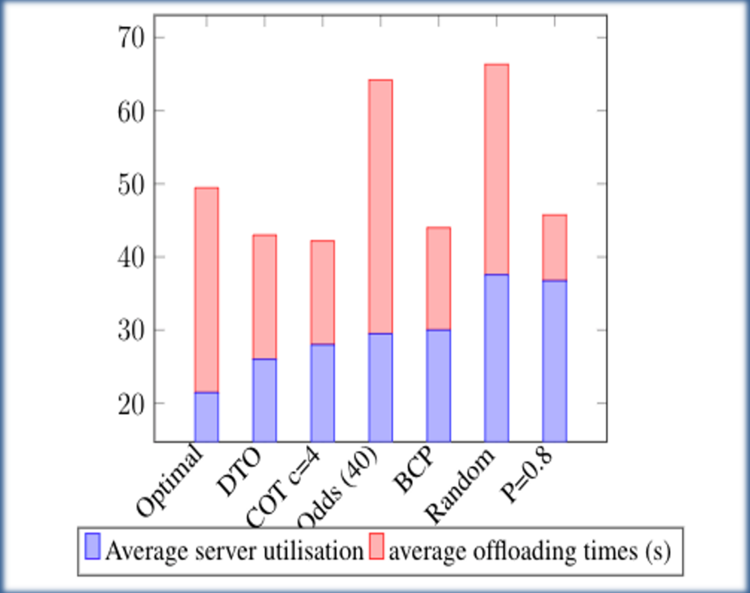


Figure 13: Server utilisation.

[1] L.Bracciale,M.Bonola,P.Loreti,G.Bianchi,R.Amici,andA.Rabuffi, "CRAWDAD dataset roma/taxi (v. 2014-07-17)," Downloaded from https://crawdad.org/roma/taxi/20140717, Jul. 2014.
[2] https://github.com/alibaba/clusterdata/blob/master/cluster-trace-v2018/trace_2018.md

Cont’d



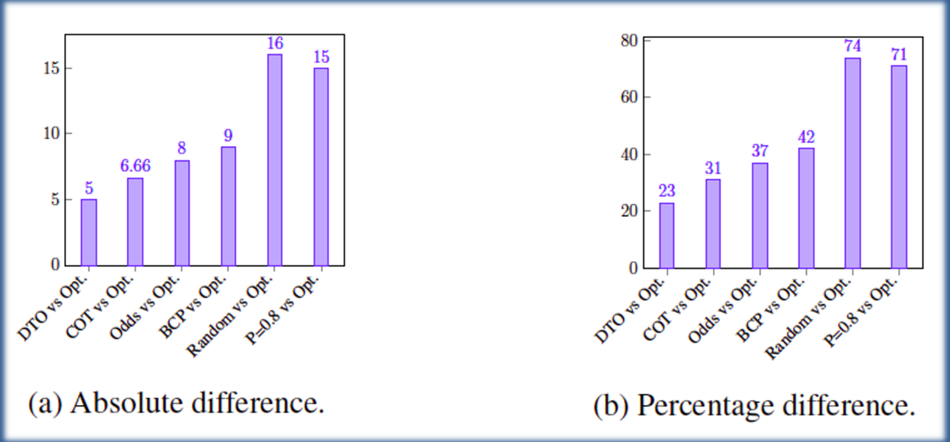


Figure 14: Real data set results.

Performance Evaluation (3): Real Implementation

- Machines:
  - MacBook Pro (new generation)
  - MacBook Pro (old one)
  - VM
  - Raspberry Pi
- Mobility
  - Each time, change the order of the list.
- Random variable:
  - Average execution time (long run)
- Task
  - Image recognition task

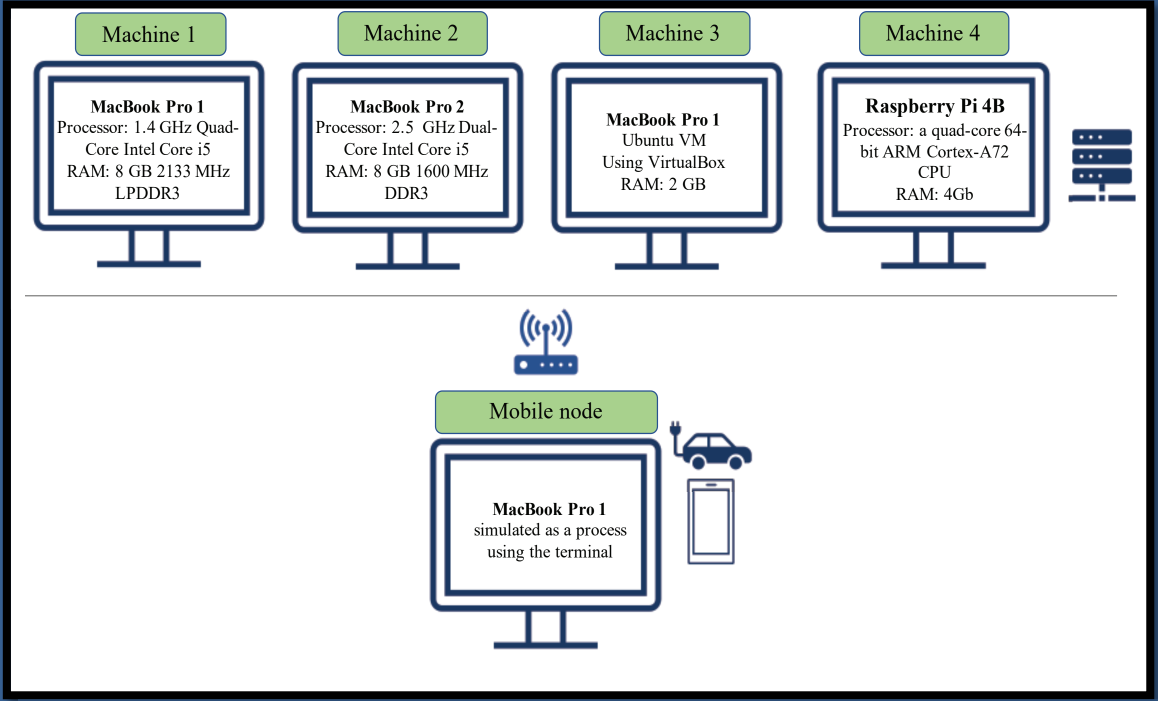


Figure 15: Machines used in the experiment.

Cont'd

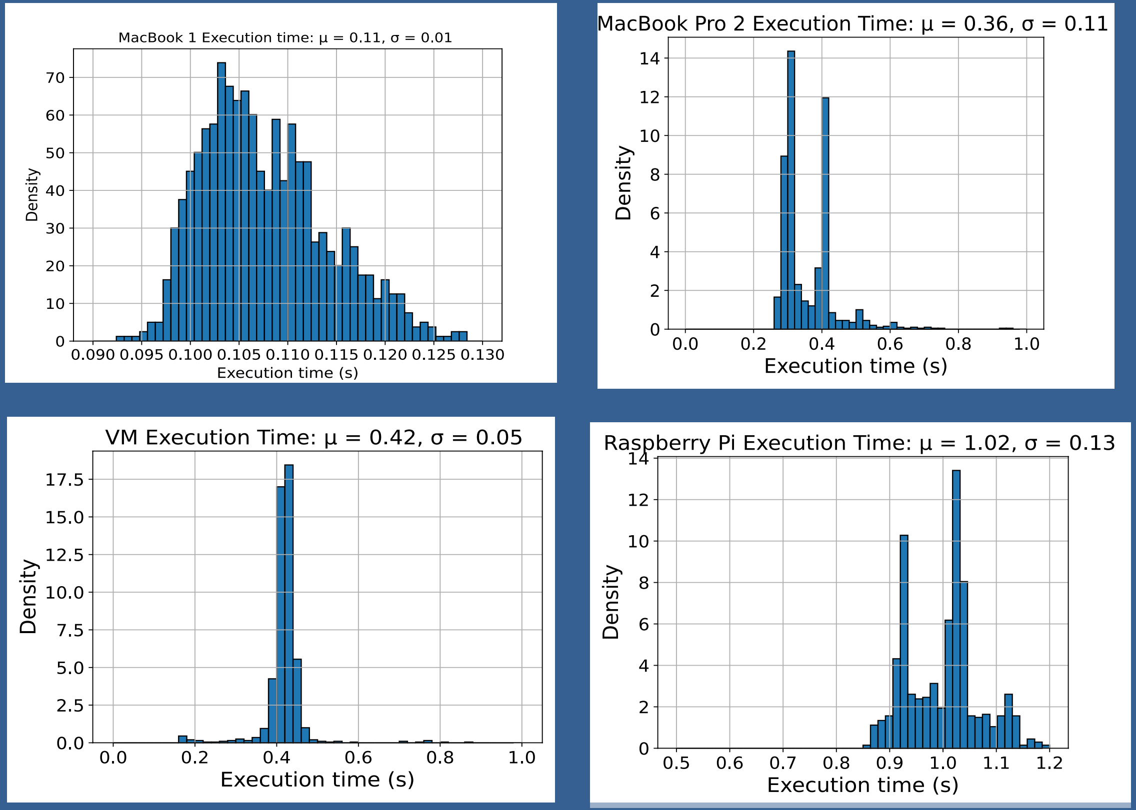


Figure 16: processing time for each machine.

Cont’d

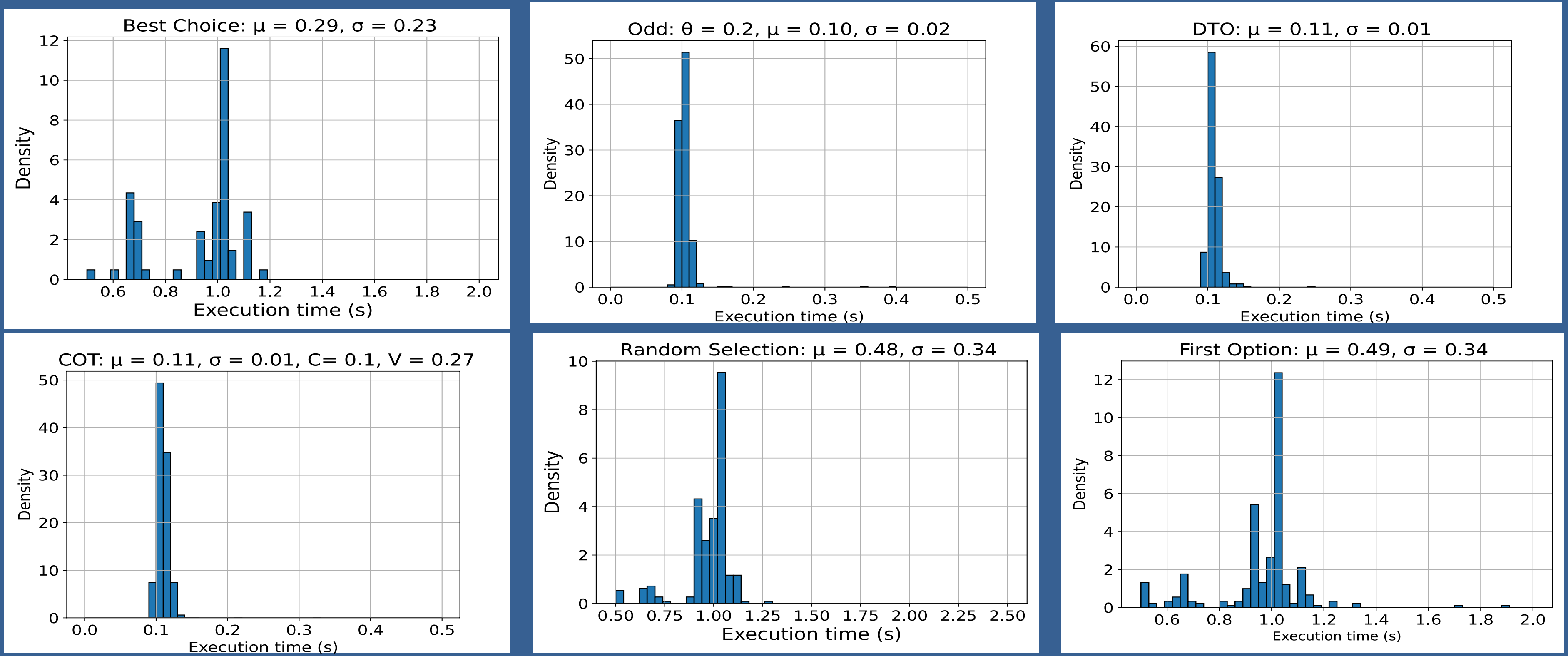


Figure 17: processing time for each model.

Future Work

- Competitive Scenarios
- Different probability distribution:
- Different random variables

University of Glasgow

School *of* Computing Science

Essence: Pervasive & Distributed Intelligence

Thank you!
Questions

i.alghamd.1@research.gla.ac.uk

http://www.dcs.gla.ac.uk/essence/