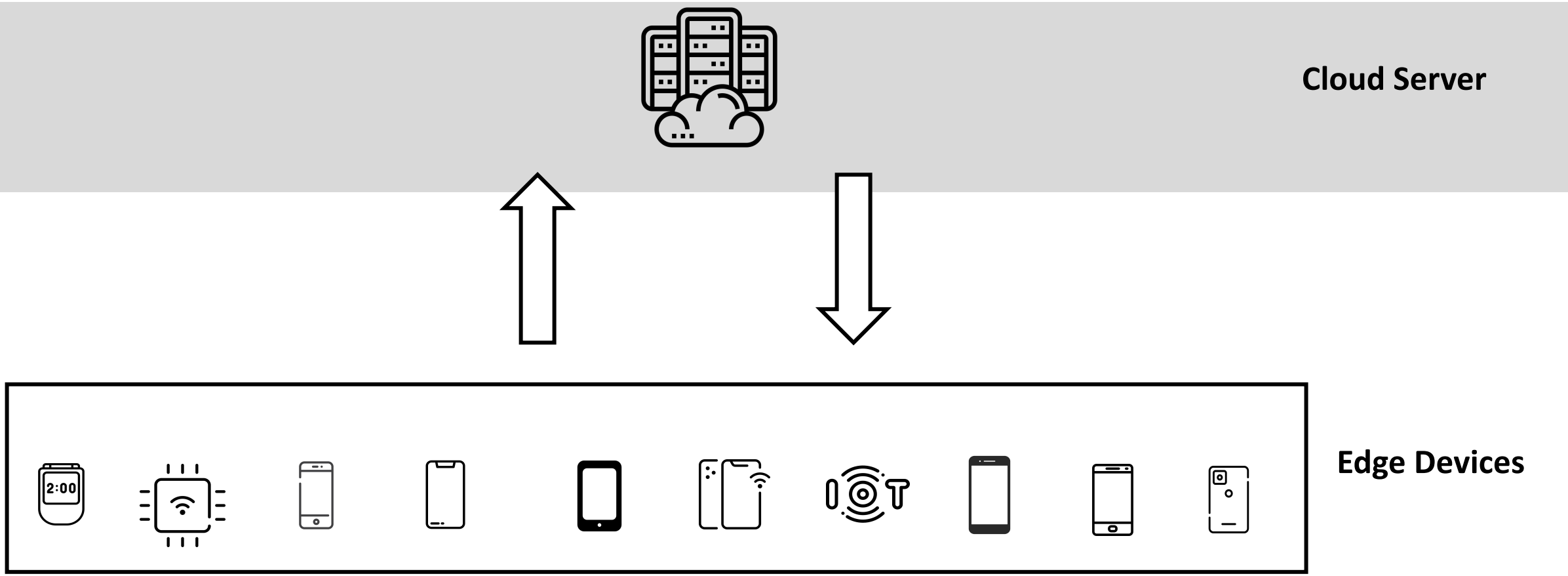
A Genetic Algorithm to schedule task in edge computing

Manish Pandey

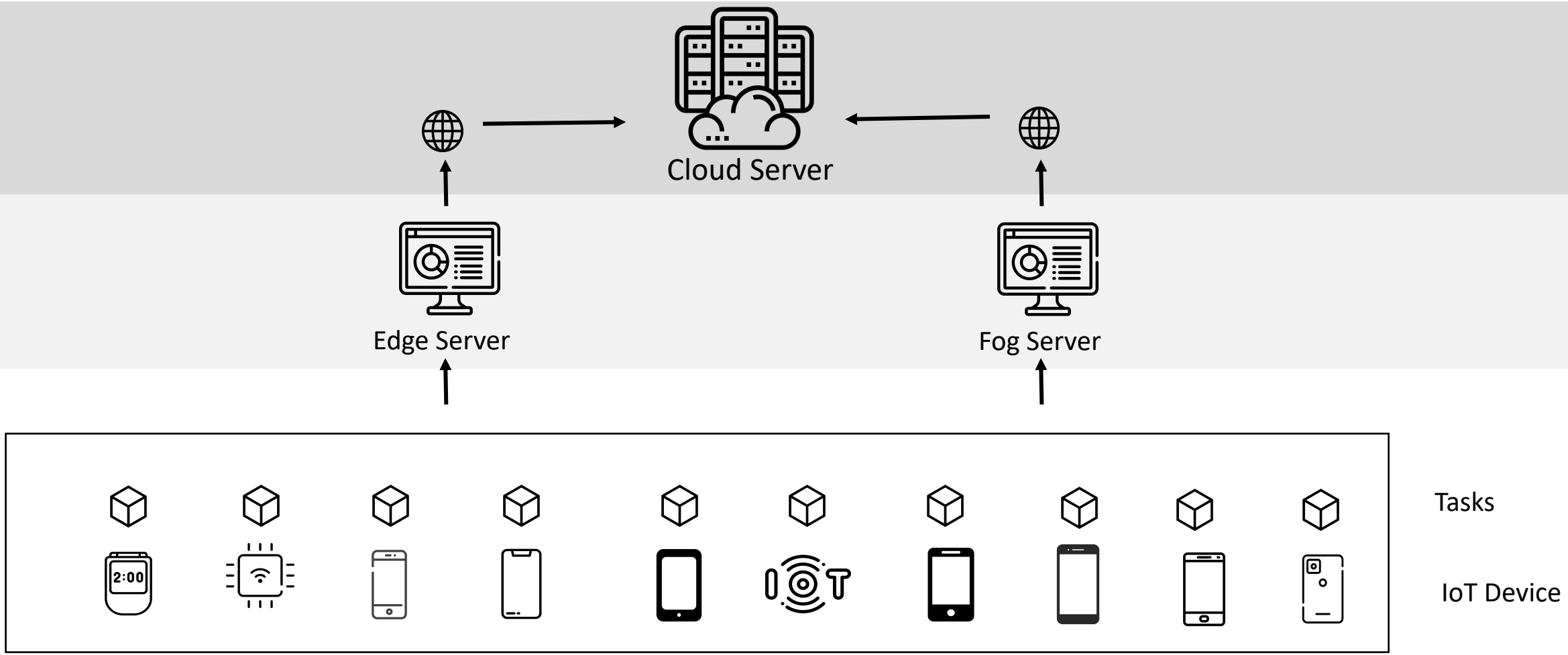
2021327158

School of Computer Science and Engineering

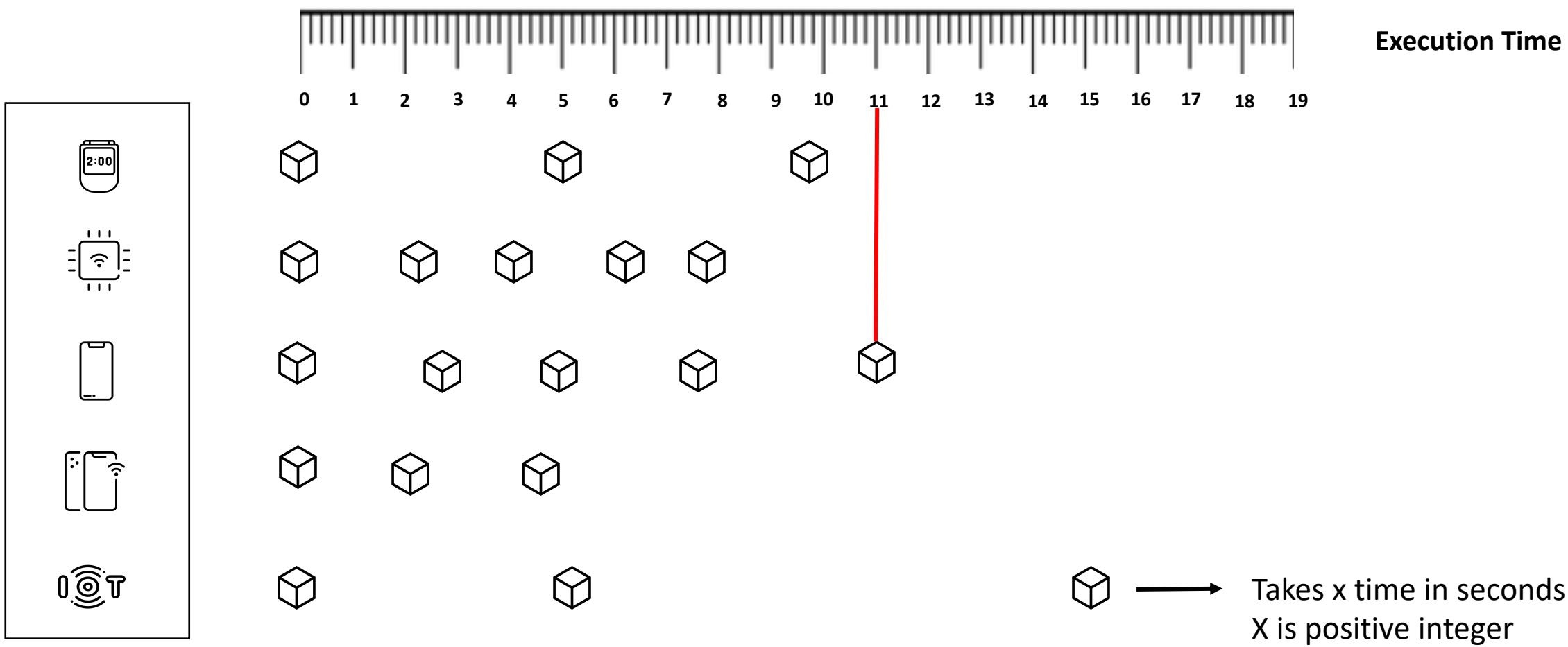
Background



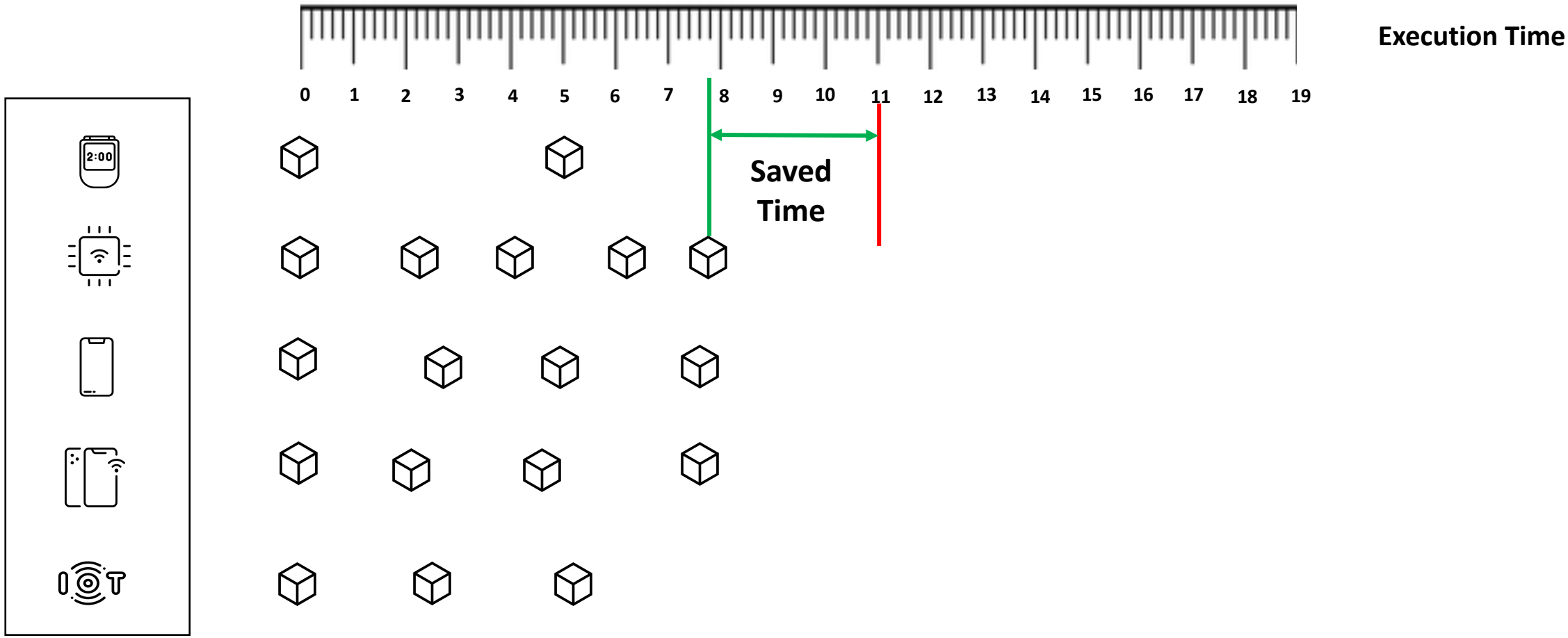
Introduction



Task Scheduling in Edge Devices



Task Scheduling Results after Genetic Algorithm

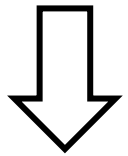


Task Representation

- Edge Device 0: T0, T6
- Edge Device 1: T2, T4, T8,
- Edge Device 2: T3, T9, T10, T12
- Edge Device 3: T1, T11
- Edge Device 4: T5, T7

[0, 3, 1, 2, 1, 4, 0, 4, 1, 2, 2, 3, 2]

Individual



[[0, 3, 1, 2, 1, 4, 0, 4, 1, 2, 2, 3, 2],
[0, 4, 3, 1, 4, 2, 2, 3, 3, 1, 1, 0, 1],
…..
[2, 0, 0, 3, 3, 1, 4, 4, 0, 1, 2, 0, 1],

Population

Is it really a combinatorial optimization?

- The task scheduling problem is NP-hard even if there are 3 jobs and 3 machines [1]
- Task scheduling problem is a combinatorial optimization problems
- We are assigning task (t1, t2, .. tn) on various m edge devices
- Execution time depends on how the task are assigned to edge devices.
- The goal is to minimize the total execution time

[1] Sotskov, Yu N., and Natalia V. Shakhlevich. "NP-hardness of shop-scheduling problems with three jobs." *Discrete Applied Mathematics* 59.3 (1995): 237-266.

Input Parameters

- Population Size (popSize): 1000
- Number of edge devices (noOfEdge): 10
- Number of tasks (noOfTask): 150
- Execution Time of each task: Currently execution time is set randomly from 2 to 50 seconds
- Threshold (threshold): 200s
- Maximum Repeat (maxRepeat): 10
- Maximum Generation (maxGen): 100
- Maximum Running Time in seconds (maxTime): 300s

Stopping Criteria

- The fitness score value is less than the input threshold.
- If the score value remains constant for more than maxRepeat input
- If the number of generations is greater than maximum Generation
- We stop the function when the program exceeds the fixed time interval.

Fitness Function

$$Edge\ Devices\ (D)\ = \{d_1, d_2,\ ....,\ d_n\}$$
$$Task\ (T)\ = \{t_1, t_2,\ ....,\ t_m\}$$
$$Execution\ Time\ (E)\ = \{e_1, e_2,\ ....,\ e_m\}$$

$$No\ of\ Task\ in\ edge\ devices\ (X)\ =\ \left\{\ x_1, x_2,\ ....,\ x_z \mid \left(\sum_{i=1}^{z} x_i\ =\ m\right)\ \wedge (|X|\ =\ |D|)\ \right\}$$

$$Score\ =\ max\left\{\ \sum_{i}^{x} t_i e_i \mid \forall_c\ \in X\ \right\}$$

Selection Operator

- Used Tournament Selection
- Compared fitness value of new selected population with fitness bounds
- Append random selection if new population is empty

Crossover and Mutation

Individual 1	1357|30983721
Individual 1	3476|98738715
Offspring 1	1357|98738715
Offspring 2	3476|30983721

Single Point Crossover

- Used Single point crossover
- Used random crossover point
- Performed crossover on random population (75%)
- Swap for mutation
- Used Elitism for generation selection

Experiment Results

```
The output is
Edge Devices 0: 28 32 42 46 89 90 91 92 102 114 115 142 144 148
Edge Devices 1: 3 6 22 25 26 38 45 52 53 63 68 73 97 107 110 126
Edge Devices 2: 1 9 27 30 35 40 54 58 60 113 116 121 129 132 141
Edge Devices 3: 14 48 50 62 94 95 103 105 118 124 131 134 147
Edge Devices 4: 7 47 51 59 75 78 81 93 96 100 109 119 137 139 140
Edge Devices 5: 0 8 12 31 36 41 49 57 70 72 77 108 143 146
Edge Devices 6: 10 11 17 44 64 67 80 85 86 98 101 111 117 128 130 138 149
Edge Devices 7: 2 5 15 24 33 37 61 65 69 83 84 88 122 123 135 145
Edge Devices 8: 20 29 34 39 43 56 66 74 76 82 99 104 125 133 136
Edge Devices 9: 4 13 16 18 19 21 23 55 71 79 87 106 112 120 127
Total execution time is 147.89s
Initial execution time is 458.62s
Best execution time is 398.71s
```

Limitation

- Selection algorithm needs improvement
- Fitness function is simple

Future Work

- We will add task dependency
- We will use task execution time per edge devices
- We will improve the fitness function

Code Available

https://github.com/Man1ish/csgeneticalgorithm

Run project

```
root@manish:~$ python app.py
```

Thank you