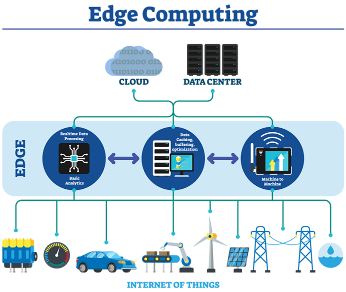
A Stackelberg Game Approach for Managing AI Tasks in a Mobile Edge Cloud System with Multiple Applications

Ettore Busani

advisor: Prof. Ardagna, coadvisor: Dr. Sedghani

Edge computing provides support to smart devices in processing the massive amount of data that they produce by providing additional computational power through a fast network.



Mobile Edge Cloud System

In this thesis we consider a Mobile Edge Cloud (MEC) system [2] consisting of three main components:

- Edge Platform with limited edge servers,
- a large number of heterogeneous mobile users,
- a pool of unlimited cloud VMs.

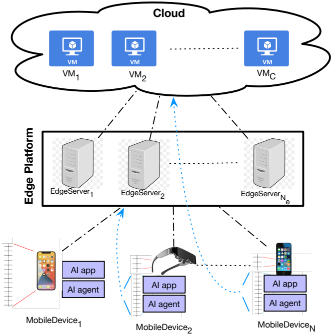


Figure: Mobile Edge Cloud System Model

User's Problem

In this framework, users are modelled as independent agents that choose the deployment that maximize their utility function and is compliant to the energy and memory constraints of their mobile device.

$$\max_{x_i^{(k)}} \sum_{k=1}^{2} x_i^{(k)} \left(U_i - cost_i^{(ak)}(\mathbf{r}) \right)$$

Subject to:

$$\sum_{k=1}^{2} x_i^{(k)} \leq 1,$$

Energy and memory constraints,

$$x_i^{(k)} \in \{0, 1\} \qquad \forall k = 1, 2$$

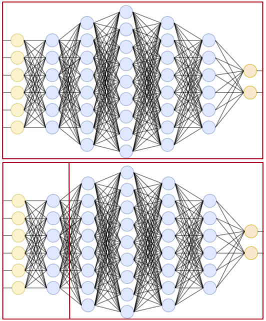


Figure: User's possible deployments

Platform Problem

$$\max_{r, n_e^{(a)}, n_c^{(a)}, y_i^e, y_i^c} \quad P_e =$$

$$= \text{Revenue}(r) - \text{Cost}\left(n_e^{(a)}, n_c^{(a)}\right)$$

Subject to:

$\mathbf{x}_i^*(\mathbf{r}) =$ solution of User's Problem

$$\sum_{a \in \mathcal{A}} n_e^{(a)} \leq N_e,$$

$\forall a \in \mathcal{A}:$

$R_i \leq \bar{R}^{(a)} \quad$ (time-response const.),

$L_e^{(a)} < n_e^{(a)},$

$L_c^{(a)} < n_c^{(a)},$

$y_i^e + y_i^c = x_i^{(2)} \quad \forall i \in \mathcal{U}$

$r^{(a2)} = r^{(1)} + \gamma^{(a)} r, \quad \gamma^{(a)} > 1$

$r_{min} \leq r \leq r_{max}$

$y_i^e, y_i^c \in \{0, 1\} \qquad \forall\ i \in \mathcal{U},$

$n_e^{(a)}, n_c^{(a)} \in \mathbb{Z}_+.$

Platform Problem

$$\max_{r, n_e^{(a)}, n_c^{(a)}, y_i^e, y_i^c} \quad P_e =$$

$$= \text{Revenue}(r) - \text{Cost}\left(n_e^{(a)}, n_c^{(a)}\right)$$

Subject to:

$$\mathbf{x}_i^*(\mathbf{r}) = \text{solution of User's Problem}$$

$$\sum_{a \in \mathcal{A}} n_e^{(a)} \leq N_e,$$

$$\forall a \in \mathcal{A}:$$

$$R_i \leq \bar{R}^{(a)} \quad \text{(time-response const.)},$$

$$L_e^{(a)} < n_e^{(a)},$$

$$L_c^{(a)} < n_c^{(a)},$$

$$y_i^e + y_i^c = x_i^{(2)} \quad \forall i \in \mathcal{U}$$

$$r^{(a2)} = r^{(1)} + \gamma^{(a)} r, \quad \gamma^{(a)} > 1$$

$$r_{min} \leq r \leq r_{max}$$

$$y_i^e, y_i^c \in \{0, 1\} \qquad \forall\ i \in \mathcal{U},$$

$$n_e^{(a)}, n_c^{(a)} \in \mathbb{Z}_+.$$

Platform Problem

$$\max_{r, n_e^{(a)}, n_c^{(a)}, y_i^e, y_i^c} P_e =$$

$$= \text{Revenue}(r) - \text{Cost}\left(n_e^{(a)}, n_c^{(a)}\right)$$

Subject to:

$\mathbf{x}_i^*(\mathbf{r}) =$ solution of User's Problem

$$\sum_{a \in \mathcal{A}} n_e^{(a)} \leq N_e,$$

$\forall a \in \mathcal{A}:$

$\boxed{R_i \leq \bar{R}^{(a)}}$ (time-response const.),

$$L_e^{(a)} < n_e^{(a)},$$

$$L_c^{(a)} < n_c^{(a)},$$

$$y_i^e + y_i^c = x_i^{(2)} \quad \forall i \in \mathcal{U}$$

$$r^{(a2)} = r^{(1)} + \gamma^{(a)} r, \quad \gamma^{(a)} > 1$$

$$r_{min} \leq r \leq r_{max}$$

$$y_i^e, y_i^c \in \{0, 1\} \qquad \forall\ i \in \mathcal{U},$$

$$n_e^{(a)}, n_c^{(a)} \in \mathbb{Z}_+.$$

Time response constraints

The average server side response time for user-i is modelled as a M/G/1 queue [3]:

$$R_i = \sum_{k=1}^{2} D_i^{(k)} x_i^{(k)} + \frac{\delta^{(a)} x_i^{(2)}}{B_i} + \frac{D_e^{(a)} x_i^{(2)} y_i^e}{1 - \frac{L_e^{(a)}}{n_e^{(a)}}} + \frac{D_c^{(a)} x_i^{(2)} y_i^c}{1 - \frac{L_c^{(a)}}{n_c^{(a)}}}$$

Time response constraints

The average server side response time for user-i is modelled as a M/G/1 queue [3]:

$$R_i = \sum_{k=1}^{2} D_i^{(k)} x_i^{(k)} + \frac{\delta^{(a)} x_i^{(2)}}{B_i} + \frac{D_e^{(a)} x_i^{(2)} y_i^e}{1 - \frac{L_e^{(a)}}{n_e^{(a)}}} + \frac{D_c^{(a)} x_i^{(2)} y_i^c}{1 - \frac{L_c^{(a)}}{n_c^{(a)}}}$$

Problem Complexity

The Stackelberg game is a mixed-integer nonlinear program (MINLP) and it can be solved by a global solver. Finding the solution is very slow because of the large number of variables and constraints:

- **N. of variables:** $2\left(|\mathcal{U}| + |\mathcal{A}|\right) + 1$
- **N. of constraints:** $4|\mathcal{U}| + 5|\mathcal{A}| + 2$

Proposed Solution

The optimal r is within a finite set of values $\mathcal{R}$

$\forall r \in \mathcal{R}$

Solve a simplified platform problem and return set of *elite solutions*

Use *elite solutions* as proxy to generate complete solutions. Return the one with highest profit

Perform a Tabu Search to further optimize the final profit

Proposed Solution

The optimal r is within a finite set of values $\mathcal{R}$

$\xrightarrow{\forall r \in \mathcal{R}}$

Solve a simplified platform problem and return set of *elite solutions*

Use *elite solutions* as proxy to generate complete solutions. Return the one with highest profit

$\rightarrow$

Perform a Tabu Search to further optimize the final profit

Proposed Solution

The optimal r is within a finite set of values $\mathcal{R}$

$\forall r \in \mathcal{R}$

Solve a simplified platform problem and return set of *elite solutions*

Use *elite solutions* as proxy to generate complete solutions. Return the one with highest profit

Perform a Tabu Search to further optimize the final profit

Proposed Solution

The optimal r is within a finite set of values $\mathcal{R}$

$\forall r \in \mathcal{R}$

Solve a simplified platform problem and return set of *elite solutions*

Use *elite solutions* as proxy to generate complete solutions. Return the one with highest profit

Perform a Tabu Search to further optimize the final profit

Tabu Search

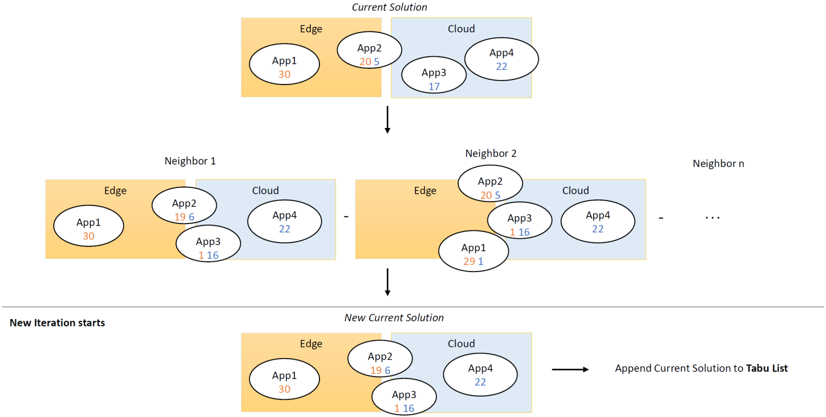


Figure: Tabu search algorithm.

Comparison with BARON

We compared the solution found by our method with the one found by BARON 22.3.21 [1]. We considered a system with 3 applications and n. of users in the range $\{10, 20, \ldots, 100\}$. We measured:

- **Platform profit** of final solution.
- **Time required** to reach the final solution.

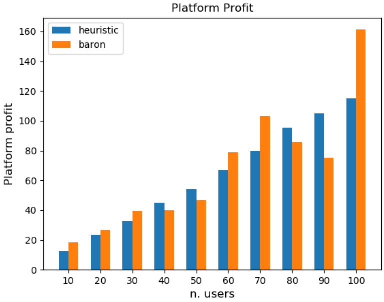


Figure: Heuristic vs BARON comparison

Comparison with BARON

We compared the solution found by our method with the one found by BARON 22.3.21 [1]. We considered a system with 3 applications and n. of users in the range $\{10, 20, \ldots, 100\}$. We measured:

- **Platform profit** of final solution.
- **Time required** to reach the final solution.

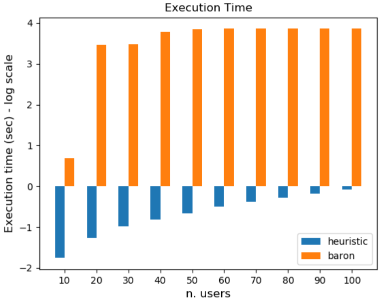


Figure: Heuristic vs BARON comparison

Heuristic Approach Scalability

We measured the time required by our heuristic approach to reach the final solution for increasing problem complexity by varying:

- **N. of users** in the range $\{50, 100, \ldots, 1000\}$.
- **N. of applications** in the range $\{1, 2, \ldots, 7\}$

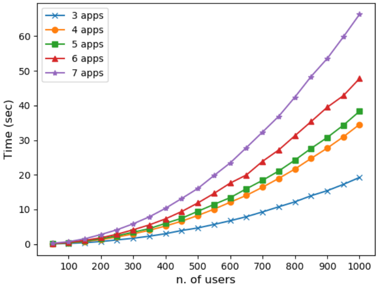


Figure: Execution time of heuristic approach

Heuristic Approach Scalability

We measured the time required by our heuristic approach to reach the final solution for increasing problem complexity by varying:

- **N. of users** in the range $\{50, 100, \ldots, 1000\}$.
- **N. of applications** in the range $\{1, 2, \ldots, 7\}$

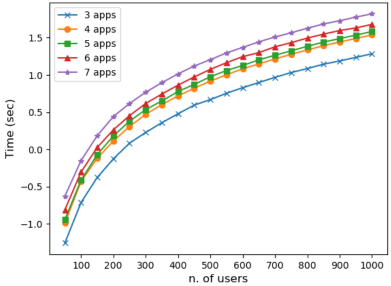


Figure: Execution time of heuristic approach

Tabu Search

We measured the current and best profit profiles of tabu search for different combinations of the hyperparameters. Unfortunately, the increase in profit achieved in most cases is not significant.

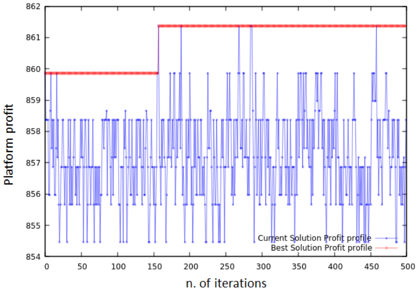


Figure: tabu search with random neighbor selection, long memory scenario.

Tabu Search

We measured the profit increase (in percentage) achieved by tabu search for systems with 500 users, 5 applications and progressively increased variance of the parameters among users. We tested all the possible combinations of the hyperparameters.

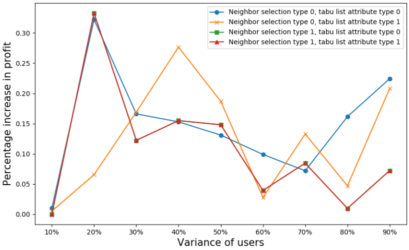


Figure: Profit increase by tabu search for different users' variance.

Conclusion and Future Work

Overall, we have developed a fast and efficient way to approximately solve the Stackelberg Game loosing less than 5% on average with respect to the optimal solution. Moreover, our model can be generalized to account for multiple deployments for each AI application. Although our current implementation of tabu search is not effective in this framework, changes can be made to the algorithm to expand the neighborhood hoping to achieve better results.

References

[1] Minlp: Baron. `https://minlp.com/baron-solver.`

[2] H. Sedghani, D. Ardagna, M. Passacantando, M. Z. Lighvan, and H. S. Aghdasi. An incentive mechanism based on a Stackelberg game for mobile crowdsensing systems with budget constraint. *Ad Hoc Networks*, 123:102626, 2021.

[3] U. Tadakamalla and D. A. Menasce. Autonomic resource management for fog computing. *IEEE TCC*, pages 1–1, 2021.

Relaxed Problem

$$\min_{n_e^{(a)}, n_c^{(a)}, \Lambda_e^{(a)}, \Lambda_c^{(a)}} \quad c_e n_e + c_c n_c$$

Subject to:

$$\sum_{a \in \mathcal{A}} n_e^{(a)} \leq N_e$$

$\forall a \in \mathcal{A}$:

$$\frac{\Lambda_e^{(a)}}{\Lambda^{(a)}} \frac{D_e^{(a)} n_e^{(a)}}{n_e^{(a)} - D_e^{(a)} \Lambda_e^{(a)}} + \frac{\Lambda_c^{(a)}}{\Lambda^{(a)}} \frac{D_c^{(a)} n_c^{(a)}}{n_c^{(a)} - D_c^{(a)} \Lambda_c^{(a)}} \leq \bar{\bar{R}}^{(a)}$$

$$D_e^{(a)} \Lambda_e^{(a)} < n_e^{(a)}$$

$$D_c^{(a)} \Lambda_c^{(a)} < n_c^{(a)}$$

$$\Lambda_e^{(a)} + \Lambda_c^{(a)} = \Lambda^{(a)},$$

$$n_e^{(a)}, n_c^{(a)} \geq 0$$

$$\Lambda_e^{(a)}, \Lambda_c^{(a)} \geq 0.$$

Optimal r

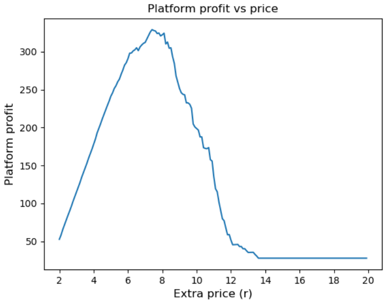


Figure: Platform profit function for 200 users and 4 applications.

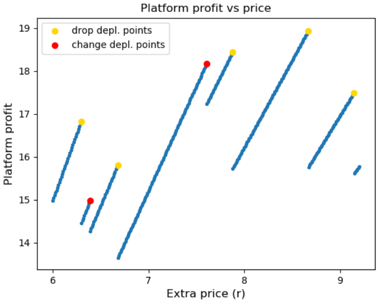


Figure: Close up frame of platform profit function for 20 users and 2 applications.

Comparison with BARON

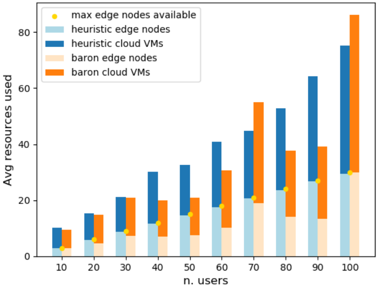


Figure: Average of resources used.

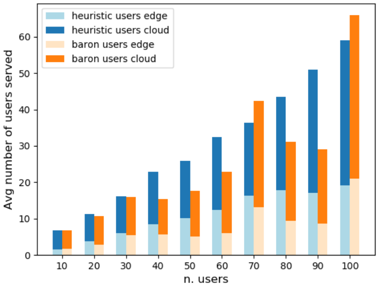
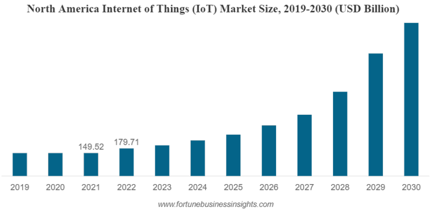


Figure: Average of users served.

Edge computing and IoT

Edge computing has experienced increased popularity parallel to growth of Internet of Things market size[a].

[a] source: https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307



Tabu search - neighbors generation process

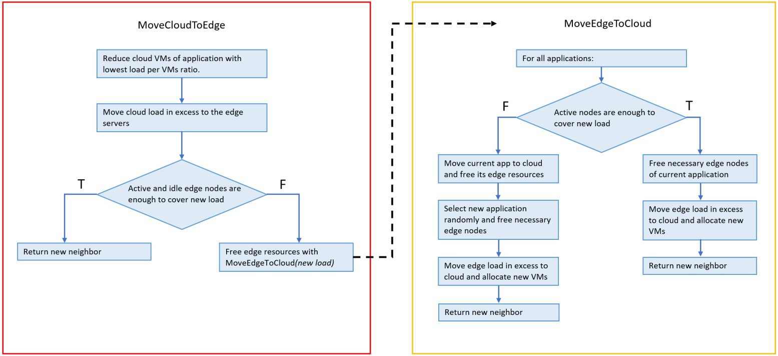


Figure: Flowchart of neighbors generation process.

Time execution of tabu search

We measured the time required by tabu search for system with increasing size.

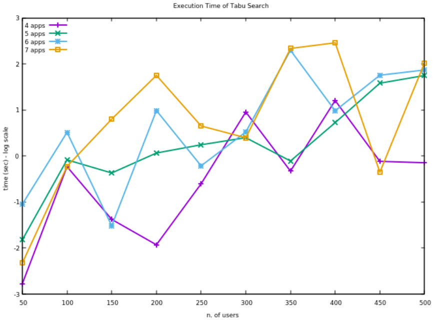


Figure: Execution time of tabu search with 500 max iterations.

Current and best profit profiles of tabu search.

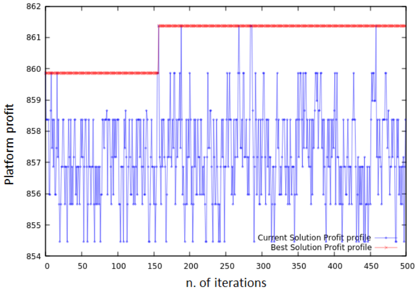


Figure: Random neighbor selection, long memory scenario

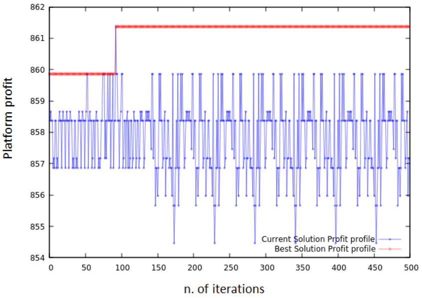


Figure: Best profit neighbor selection, long memory scenario

Current and best profit profiles of tabu search.

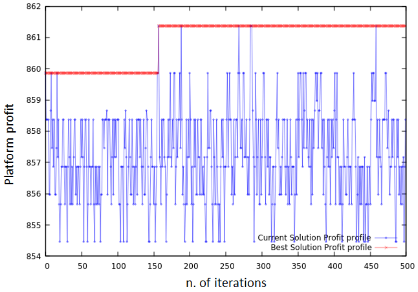


Figure: Random neighbor selection, short memory scenario

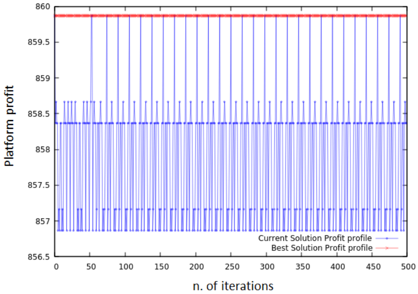


Figure: Best profit neighbor selection, short memory scenario