Task Offloading using Multi-armed bandit Optimization in Autonomous Mobile Robots

Anis Ur Rahman[1], Asad Waqar Malik[2], Hasan Ali Khattak[2], Moayad Aloqaily[3]

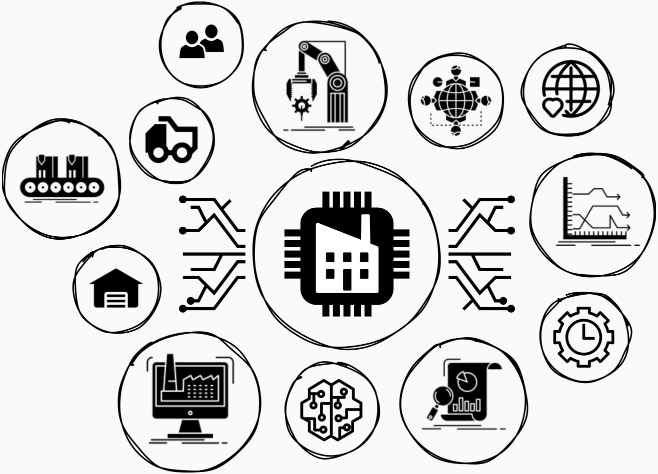
[1] University of Jyväskylä, Finland
[2] National University of Sciences and Technology, Pakistan
[3] Mohamed bin Zayed University of Artificial Intelligence, United Arab Emirates
email:anis.u.rahman@jyu.fi

Table of contents

Introduction and motivation

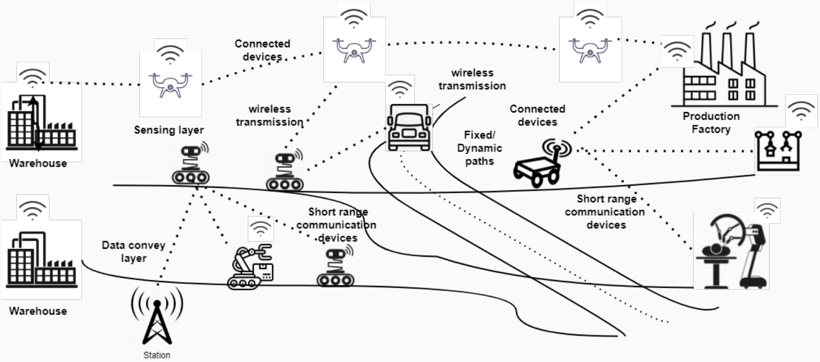
Motivation: Industry 5.0

Industry 5.0 is a **human-centered** and **collaborative** approach to manufacturing, where humans and machines work together in cooperation.



Motivation: Collaborative Industrial Environment

- **Develop** a framework for **collaborative task offloading** using different attributes of participating robots.



System Model

System Model: Overview

Environment

- **fenceless rectangular smart factory floor**,
- **workstations** spread across the floor in a **grided** manner.
  - set of **mobile** robots M,
  - set of **stationed** robots S, and
  - set of **edge** nodes E.

System Model: Overview

Environment

- **fenceless rectangular smart factory floor**,
- **workstations** spread across the floor in a **grided** manner.
  - set of **mobile** robots M,
  - set of **stationed** robots S, and
  - set of **edge** nodes E.

Robot generates a **task** represented as a **tuple**,

$$(x, y, w, p, d)$$

- x: **input** task size in KBs,
- y: **output** task size in KBs,
- w: **CPU cycles** required in Mbits,
- p: task **priority** (soft, medium, hard), and
- d: task **deadline**.

System Model: Overview

Environment

- **fenceless rectangular smart factory floor**,
- **workstations** spread across the floor in a **grided** manner.
  - set of **mobile** robots M,
  - set of **stationed** robots S, and
  - set of **edge** nodes E.

The problem

There are **limited on-device computational capabilities**, the generated tasks are **offloaded** across the **resource-sharing network**.

System Model: Goal

Each robot maintains a list of **neighbouring resources**, ones within i^{th} source robot's **data-transmission range**,

$$X_{-i} \subseteq \{M, S, E\} - i$$

System Model: Goal

Each robot maintains a list of **neighbouring resources**, ones within i^{th} source robot's **data-transmission range**,

$$X_{-i} \subseteq \{M, S, E\} - i$$

Our goal

Minimize **total service delay**, the sum of **communication** and **computation** costs.

Assumptions

- Data rate remains **stable** and **constant** once direct connection is established.
- Use a **linear** cost function for total service delay.

System Model: Constraints

Three cases for **communication cost** for task offloading:

1. source robot to neighbouring **mobile** robots,
2. source robot to neighbouring **stationed** robots, and
3. source robot to the neighbouring **edge** node.

System Model: Constraints

Three cases for **communication cost** for task offloading:

1. source robot to neighbouring **mobile** robots,
2. source robot to neighbouring **stationed** robots, and
3. source robot to the neighbouring **edge** node.

Constraints I: Communication

- Total **channel capacity** is predicted by the Shannon–Hartley theorem.
- The channel capacity determines **communication delay**,
  - $D^{\uparrow}$: **uplink** delay, and
  - $D^{\downarrow}$: **downlink** delay.

System Model: Constraints

Constraints I: Communication

Constraints II: Computation

- f: available **computational capacity** of resource,
- D_w: **residence time** in task queue before resource provisioning, and
- D_c: **computation delay** is time taken to compute task on provisioned computational resource.

Algorithm Design

Algorithm: Device selection

Multi-arm bandit (MAB) approach

- To **select** the **nearby under-utilized devices** for task offloading.
- Typically, MAB techniques make decisions:
    1. over time under **uncertainty**, and
    2. exhibit **simplicity**.

Algorithm: Device selection

Multi-arm bandit (MAB) approach

- To **select** the **nearby under-utilized devices** for task offloading.
- Typically, MAB techniques make decisions:
  1. over time under **uncertainty**, and
  2. exhibit **simplicity**.

- Assuming,
  - K possible actions or arms, and
  - T total rounds.
- At every round, an arm is chosen and reward is collected.

Algorithm: Device selection

Multi-arm bandit (MAB) approach

- To **select** the **nearby under-utilized devices** for task offloading.
- Typically, MAB techniques make decisions:
  1. over time under **uncertainty**, and
  2. exhibit **simplicity**.

- Assuming,
  - K possible actions or arms, and
  - T total rounds.
- At every round, an arm is chosen and reward is collected.

Bandit feedback

At every round, there is **auxiliary feedback** associated with every action,

- **gain** knowledge of the service delay of the chosen arm, and
- **builds** a context of the neighbouring environment.

Algorithm: Device selection

Multi-arm bandit (MAB) approach

- To **select** the **nearby under-utilized devices** for task offloading.
- Typically, MAB techniques make decisions:
  1. over time under **uncertainty**, and
  2. exhibit **simplicity**.

- Assuming,
  - K possible actions or arms, and
  - T total rounds.
- At every round, an arm is chosen and reward is collected.

The challenge

- Need to **balance** **exploration-exploitation trade-off**, not spending too much effort on exploring information.

Algorithm: Adaptive online learning

The solution

To learn a **balance** between **exploration** and **exploitation** using **online reinforcement learning.**

Consider a task is generated at robot i at time t,

1. At every iteration, a_t^i is available **action-set**, **local** or **offload.**
   - **list** i^{th} robot's **neighbouring resources**, X_{-i}.
   - **adjust** action a_t^i based on **earlier action** a_{t-1}^i.
2. After T iterations **accumulated expected reward** should be highest, corresponding to **improved throughput** of the collaborative resource-sharing network.

Algorithm: Adaptive online learning

Considering an **action** a is available at **time instant** t, i.e. $a_t = a$,

1. **Estimate quality** of this action is,

$$\mu(a) = \mathbb{E}[\rho_t | (a_t = a)]$$

Algorithm: Adaptive online learning

Considering an **action** a is available at **time instant** t, i.e. $a_t = a$,

1. **Estimate quality** of this action is,

$$\mu(a) = \mathbb{E}[\rho_t | (a_t = a)]$$

2. **Select** a suitable **action** a_t,
   - **pure-greedy.** selects an action a_t within the available action set as, $\max_a(\mu(a))$.
   - **ϵ-greedy.** explores **other options** with ϵ probability.
   - **ϵ-decay.** explores **other options** with ϵ probability but this exploration **diminishes** with passing iterations.

Algorithm: Adaptive online learning

Considering an **action** a is available at **time instant** t, i.e. $a_t = a$,

1. **Estimate quality** of this action is,

$$\mu(a) = \mathbb{E}[\rho_t | (a_t = a)]$$

2. **Select** a suitable **action** a_t,
3. **Compute cumulative reward** for it at step t,

$$Q(a_t) = Q_{a_{t-1}} + \frac{1}{t}(\rho_t - Q_{a_{t-1}})$$

Algorithm: Adaptive online learning

Considering an **action** a is available at **time instant** t, i.e. $a_t = a$,

1. **Estimate quality** of this action is,

$$\mu(a) = \mathbb{E}[\rho_t | (a_t = a)]$$

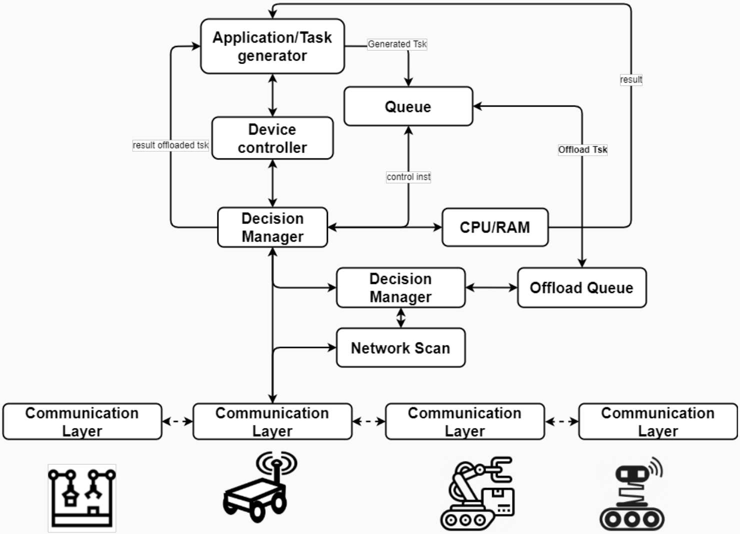
2. **Select** a suitable **action** a_t,
3. **Compute cumulative reward** for it at step t,

$$Q(a_t) = Q_{a_{t-1}} + \frac{1}{t}(\rho_t - Q_{a_{t-1}})$$

The optimization goal

$$P1 : \max_{a_1, \cdots, a_T} \frac{1}{T} \sum_{t=1}^{T} Q(a_t)$$

Algorithm: Framework



Results

Results: Simulation setup

Description	Value
Simulator	AnyLogic PLE v8.5
Simulation area	3 sq.km
Road Network	Industrial
Total simulation time	1 hr
Simulation repetitions	5 (five) times
Device arrival rate	100–500s
Communication range	100m
System	1.7 GHz Intel Core i5
RAM	4 GB
OS	macOS Mojave

Results: Performance metrics

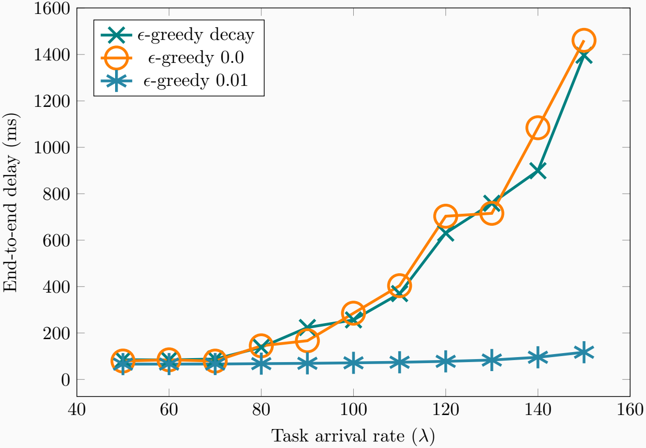
Performance metrics

1. **End-to-end delay.** is sum of **communication**, **queuing** at the device and **computation** delay.
2. **Delivery rate.** is **ratio** of completed jobs **delivered** to total jobs **offloaded** from source node.
3. **Transmission delay.** is **delay** caused due to the **data rate**.

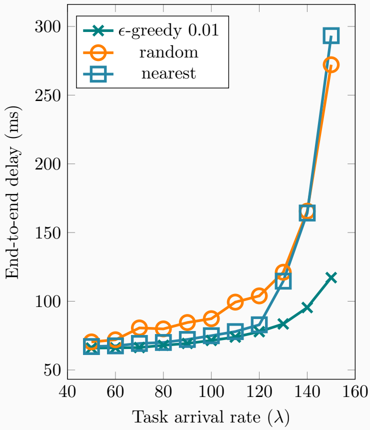
Other comparison methods:

1. **Nearest** selects device at minimum euclidean distance between devices from source.
2. **Random** selects an action randomly.

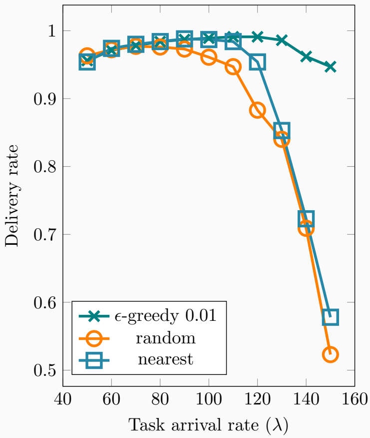
Results: Performance



Results: Performance

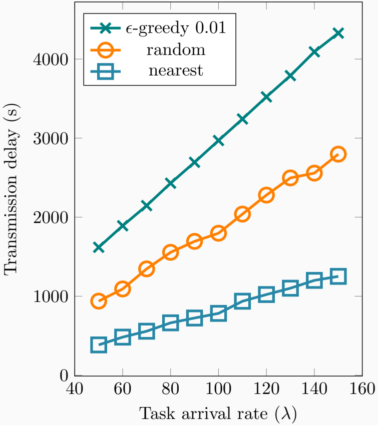


(a)

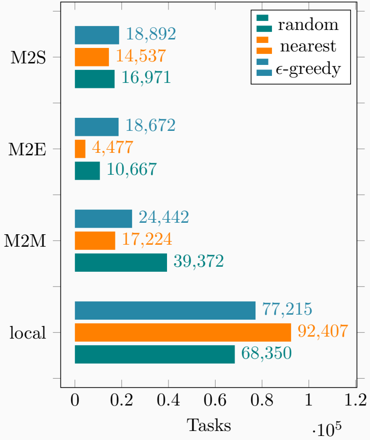


(c)

Results: Performance



(a)



(b)

Conclusions

Conclusions

- Provision of a framework for **collaborative task offloading** for **smart industry.**
- Support an **online reward-oriented** mechanism considering **resource capacity constraints** and **service delay.**
- Simulate a **resource-sharing network** among **collaborating robots.**
- Systematic evaluation of **task completion times** to assess framework **efficiency**.
- Demonstrate improved **task delivery rates.**