Edge-Enabled Two-Stage Scheduling Based on Deep Reinforcement Learning for Internet of Everything

Professor: Li-Der Chou
Presenter: Chih-Jou Tai

IEEE Internet of Things Journal

Outline

- Introduction
- Related Work
- Problem Formulation
- System Architecture
- Algorithm
- Experiment and Analysis
- Pros and cons

Introduction

- Problem Statement:
  - Managing complex scheduling in IoE with multiple flowlines.
- Research Objectives:
  - Optimize Computing & Communication Resources
  - DRL-TSS for Efficiency
  - Makespan Minimization

Outline

- Introduction
- Related Work
- Problem Formulation
- System Architecture
- Algorithm
- Experiment and Analysis
- Pros and cons

Related Work

- Reinforcement Learning in IoT Systems
  - Heterogeneous networks
  - Internet of Vehicles (IoV)
  - NarrowBand IoT networks
  - Multiagent learning strategy
  - Markov decision process

Related Work

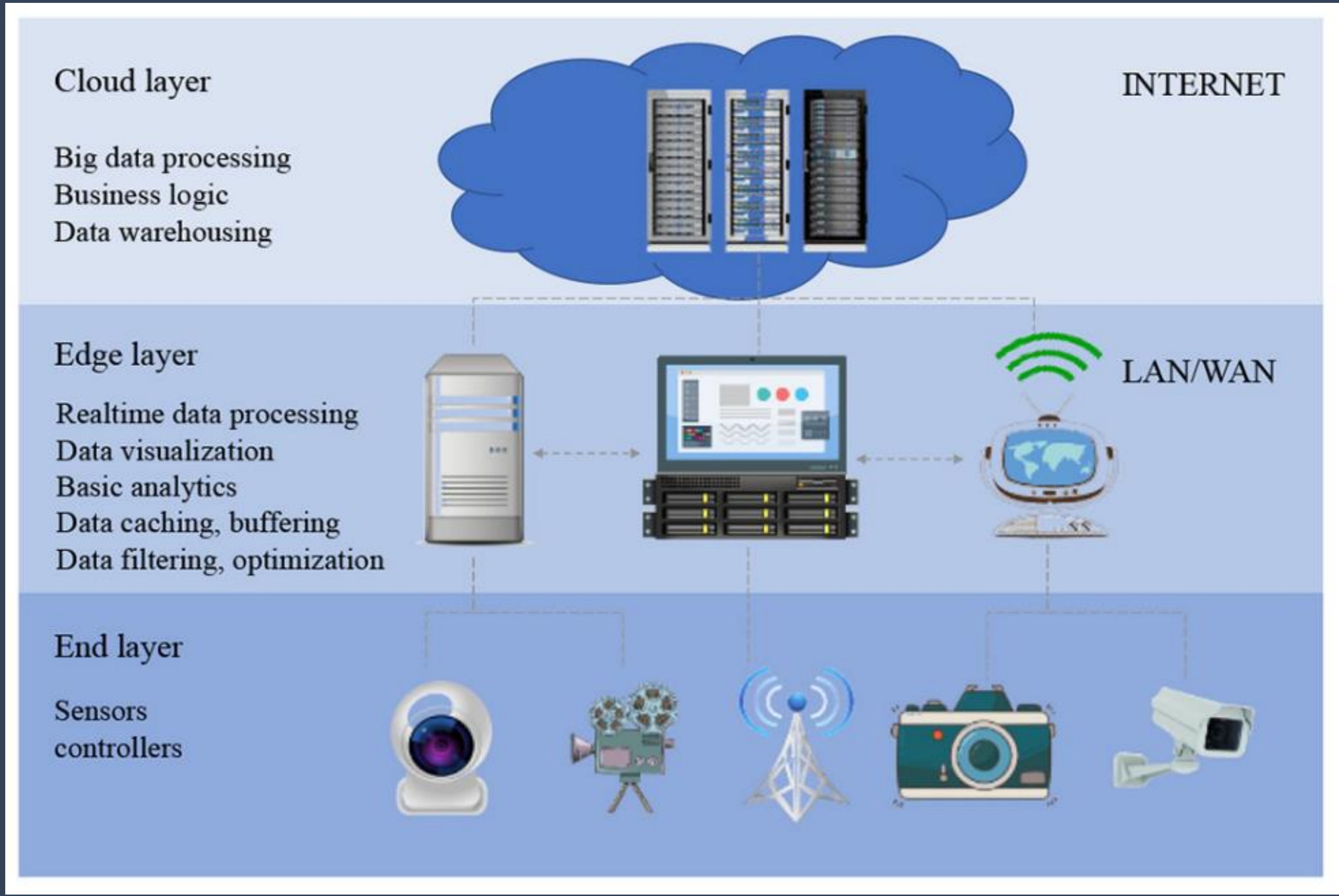
- Task Scheduling for IoT Applications
  - Mixed linear programming method
  - Hybrid algorithm
  - Model-free scheduling
  - Enhanced deep Q-learning
  - Two-step scheduling

Outline

- Introduction
- Related Work
- Problem Formulation
- System Architecture
- Algorithm
- Experiment and Analysis
- Pros and cons

Problem Formulation

- End–Edge–Cloud IoT Systems



Problem Formulation

- Problem Formulation
  - tasks J={J1,J2,…,Jn}
  - executors E={E1,E2,…,Em}
  - Each task Ji contains two operations {Oi1,Oi2} with the duration {di1,di2}
  - the completion time {ci1,ci2}

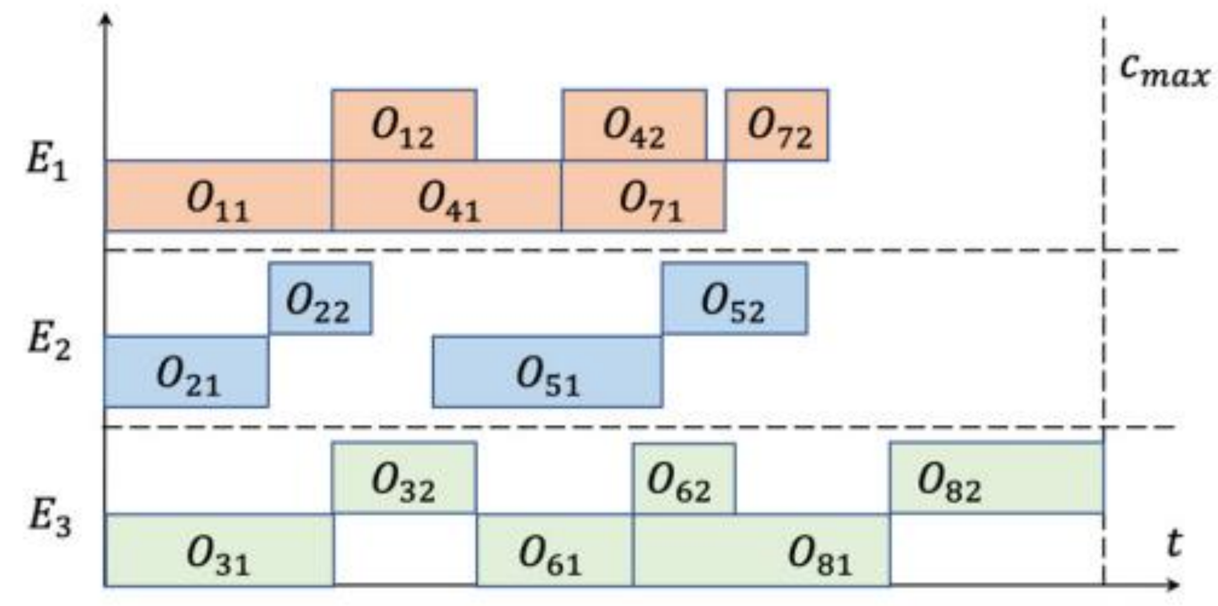


Fig. 2. Example of executed tasks in two-stage scheduling.

Outline

- Introduction
- Related Work
- Problem Formulation
- System Architecture
- Algorithm
- Experiment and Analysis
- Pros and cons

System Architecture

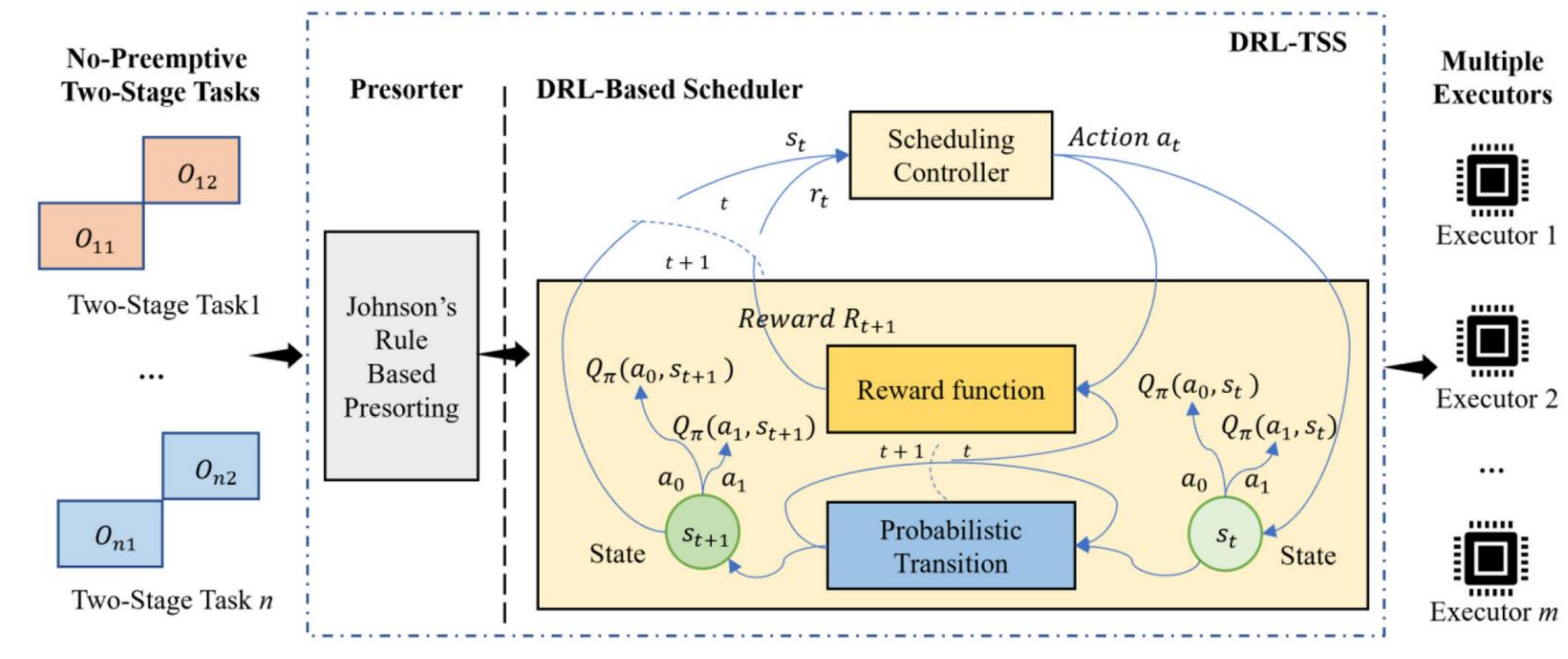


Fig. 3. Framework of DRL-TSS.

Outline

- Introduction
- Related Work
- Problem Formulation
- System Architecture
- Algorithm
- Experiment and Analysis
- Pros and cons

Algorithm

- Johnson's Rule-Based Presorter
  - Theorem 1: Johnson’s list is an optimal solution for a two-stage, single-executor scheduling problem.
  - Theorem 2: A subset of Johnson’s list is also Johnson’s list.

Algorithm 1 Johnson’s Rule-Based Presorting

Input: Task list $J = \{J_1, J_2, \dots, J_i, \dots, J_n\}$, in which each task J_i contains two operations $\{O_{i1}, O_{i2}\}$ with execution durations $\{d_{i1}, d_{i2}\}$
Output: Sorted task list J' in Johnson’s order

Initialize two task groups $G1 = \varnothing$, $G2 = \varnothing$, and $J' = \varnothing$
for each J_i in J **do**
  if $d_{i1} \leq d_{i2}$ **then** $G1 = G1 \cup J_i$
  else $G2 = G2 \cup J_i$
  end if
end for
Sort all tasks in $G1$ in ascending order based on the duration time d_{i1} for each task $J_i \in G1$
Sort all tasks in $G2$ in descending order based on the duration time d_{i2} for each task $J_i \in G2$
Merge the two task lists by appending $G2$ behind $G1$ as $J' = G1 \cup G2$
return J'

Algorithm

- Deep Reinforcement Learning for Edge-Enabled Scheduling

Algorithm 2 Training of DRL-TSS

Input: Sorted task list $J = \{J_1, J_2, \ldots, J_t, \ldots, J_n\}$ in Johnson's order
Output: Two-stage scheduling model M

Initialize action value function Q and target action value function Q' with weights $\theta' = \theta$ by Eq. (9)
Initialize learning step σ, greedy exploration probability ϵ, and discounting factor γ
Initialize experience replay buffer set D
for episode $eps = 1$ to $MaxBatchSize$ **do**
  for $t = 1$ to n **do**
    Select random action a_t that assigns J_t to a random executor with probability ϵ, otherwise $a_t = \text{argmax}_a Q(s_t, a; \theta)$
    Obtain state s_{t+1} by executing action a_t, and calculate reward r_t by Eq. (6)
    Store transition (s_t, a_t, r_t, s_{t+1}) in replay buffer D
    Sample random minibatch of transition (s_j, a_j, r_j, s_{j+1}) from D
    Calculate
$$y_j = \begin{cases} r_j & \text{if } eps \text{ terminates at } (j+1) \\ r_j + \gamma \max_{a'} Q'(s_{j+1}, a'; \theta') & otherwise \end{cases}$$
    Calculate error $e_j = (y_j - Q(s_j, a_j; \theta))^2$ and conduct gradient descent step by e_j
    Reset $Q' = Q$ in every σ steps
  end for
end for

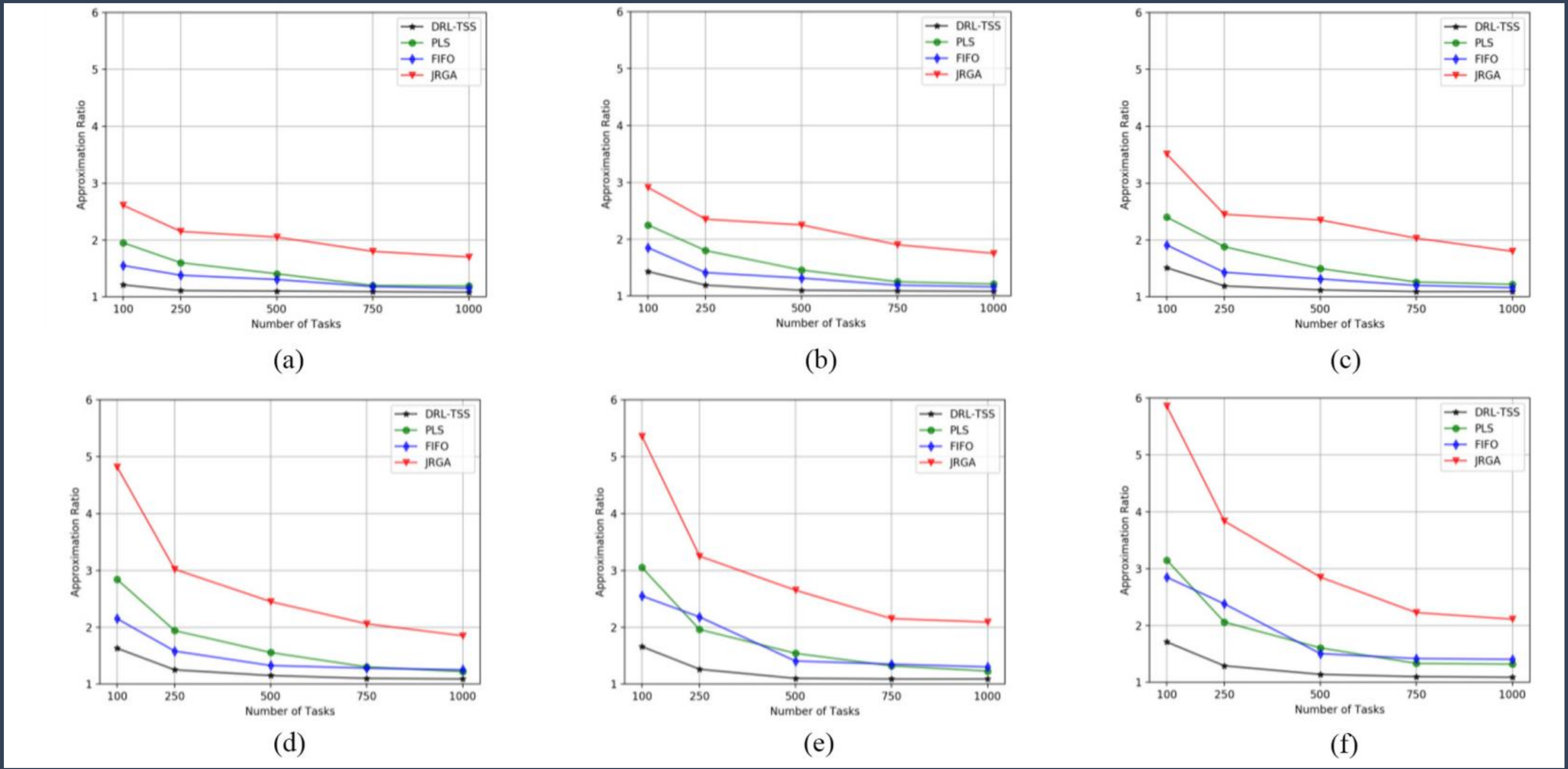
Outline

- Introduction
- Related Work
- Problem Formulation
- System Architecture
- Algorithm
- Experiment and Analysis
- Pros and cons

Experiment and Analysis

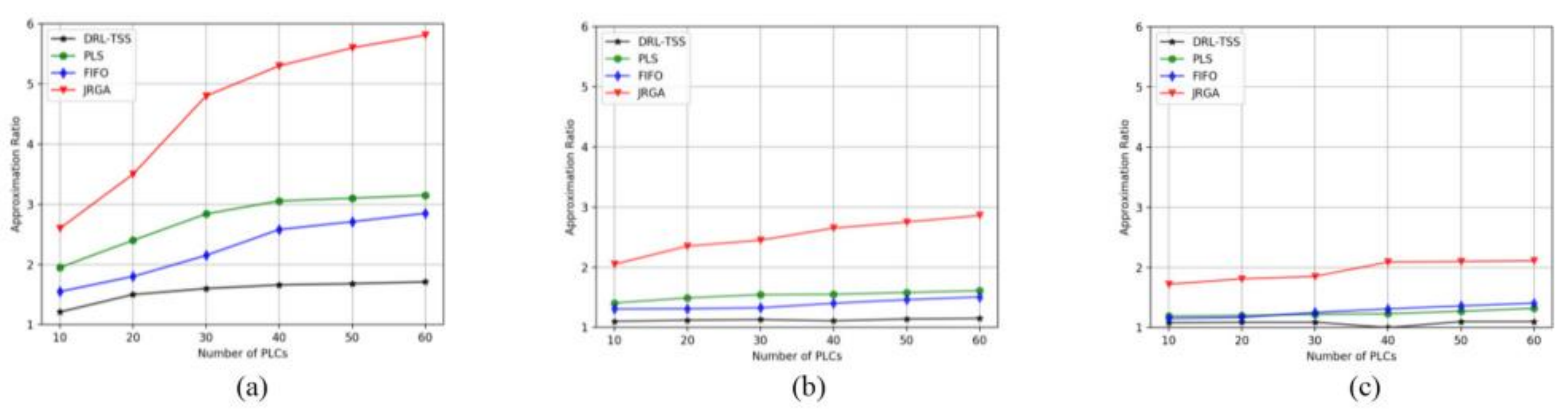
- Experiment Design Highlights:
  - Three task categories:
    light (10-100 μs), medium (100-1000 μs), heavy (1000-10000 μs)
  - Proportions: 30% light, 45% medium, 25% heavy
  - Random workload assigned to executors (10-100 μs)

Experiment and Analysis



Experiment and Analysis

- Experiment Design Highlights:



Outline

- Introduction
- Related Work
- Problem Formulation
- System Architecture
- Algorithm
- Experiment and Analysis
- Pros and cons

Pros and cons

- Pros:
  - The introduction of DRL-TSS results in an approximation ratio close to the optimal makespan.
  - It demonstrates strong performance even when dealing with a high number of tasks and multiple executors.
- Cons:
  - The "Related Work" section lacks a detailed comparison with existing techniques.
  - The paper's focus on industrial IoT scheduling challenges might limit its applicability in other domains.

Q & A