Indraprastha Institute of
Information Technology Delhi

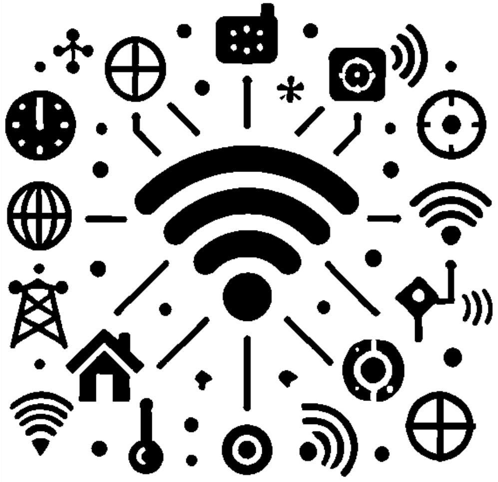
Technische Universität München

A Deadline-Aware Scheduler for Smart Factory using WiFi 6

Mohit Jain, Anis Misra, Andreas Wiese,
Syamantak Das, **Arani Bhattacharya**, Mukulika Maity

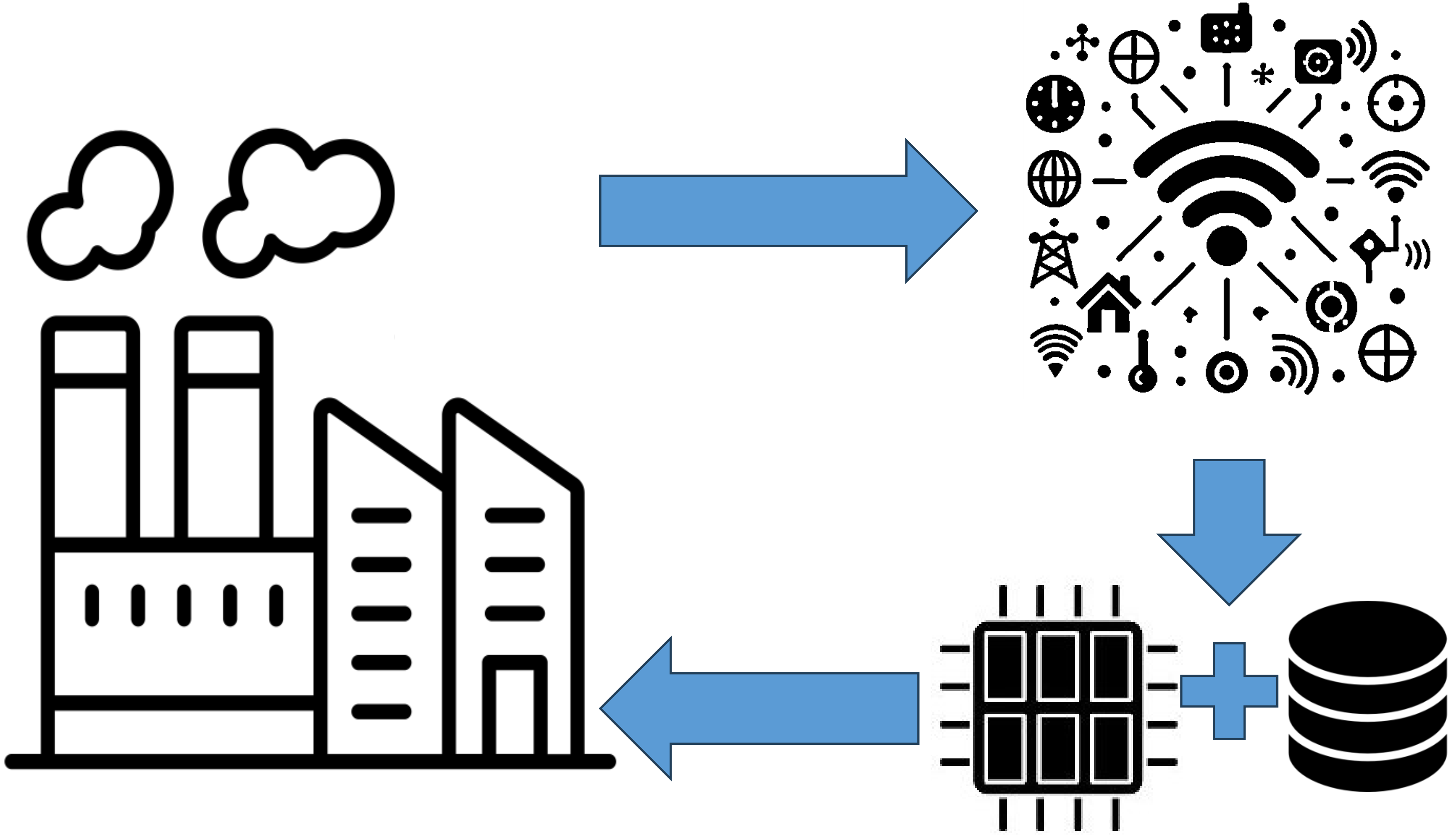
ACM MobiHoc 2024

Smart Factories are Considered to be Factories of the Future



Collects data from multiple sensors

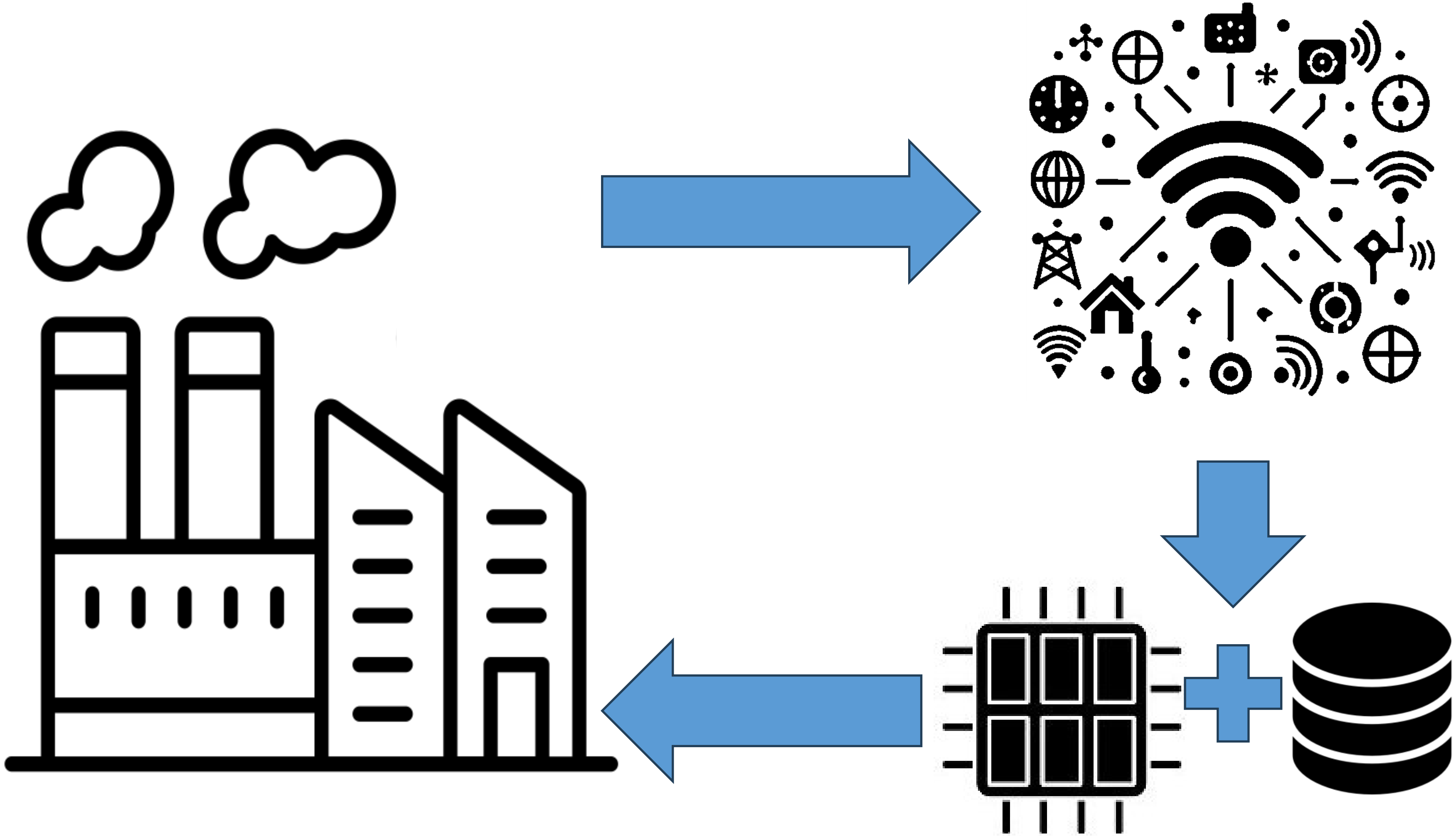
Smart Factories are Considered to be Factories of the Future



Collects data from multiple sensors

CPU + Storage for intelligent decisions

Smart Factories are Considered to be Factories of the Future

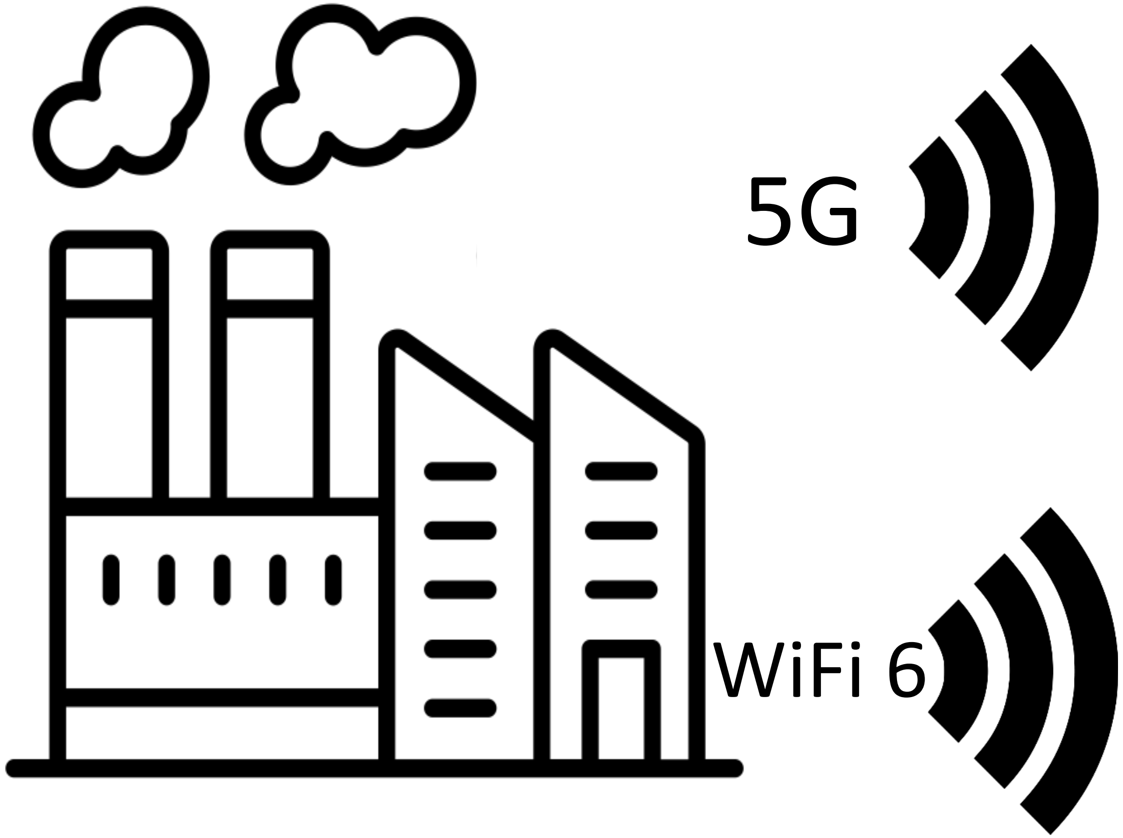


Collects data from multiple sensors

CPU + Storage for intelligent decisions

Requires reliable and real-time communication over wireless network

WiFi 6/6E and 5G Offer the Best Potential to Enable Smart Factories



Uses centralized structure, with prioritization and resource reservation for classes of traffic

Support for smart factories widely studied

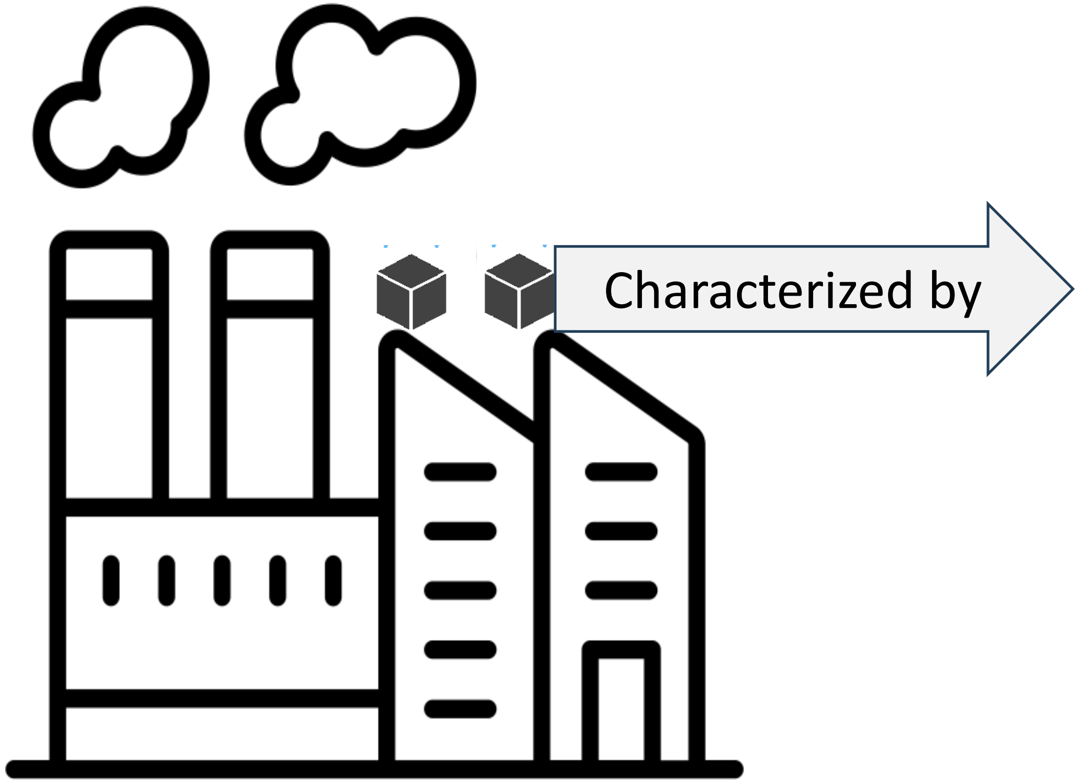
Expensive; access to large compute power

Traditionally decentralized; unlicensed spectrum

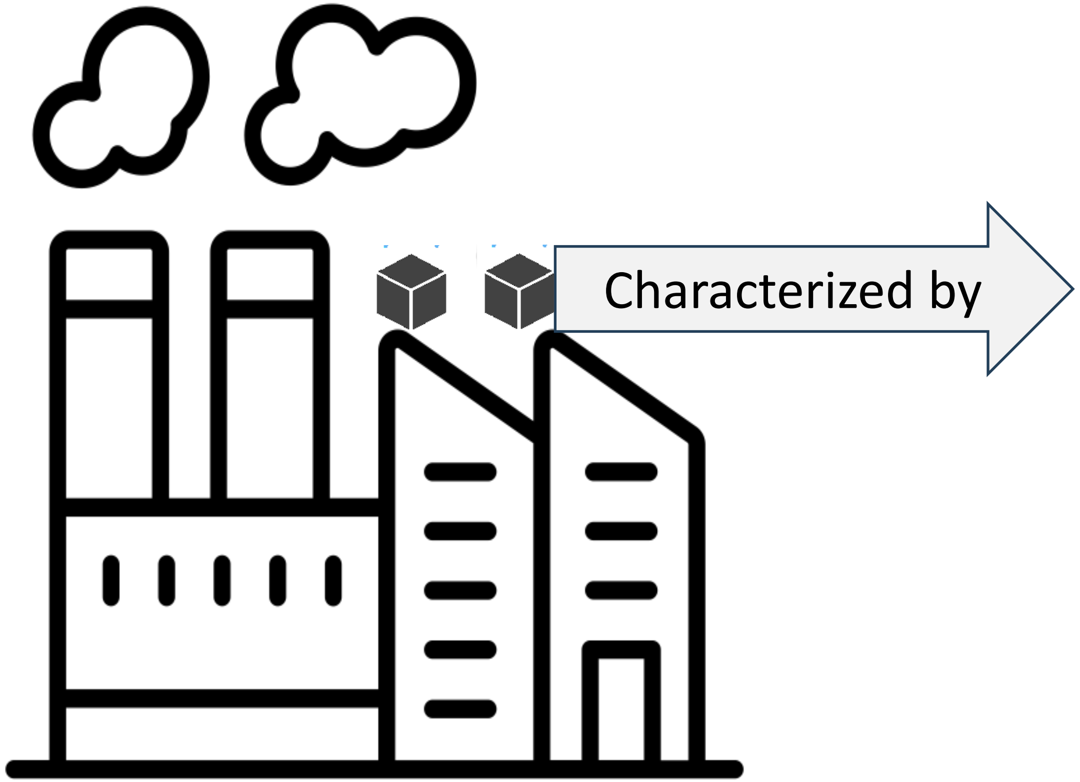
WiFi 6 has introduced partially centralized control: a specific type of broadcast packet allows access point (AP) to centrally control

Low cost of setup and operation make WiFi 6 attractive for smaller factories

Requirement of Smart Factories
Specific properties that make this problem unique



Requirement of Smart Factories
Specific properties that make this problem unique



Packets have known fixed deadlines

Requirement of Smart Factories
Specific properties that make this problem unique

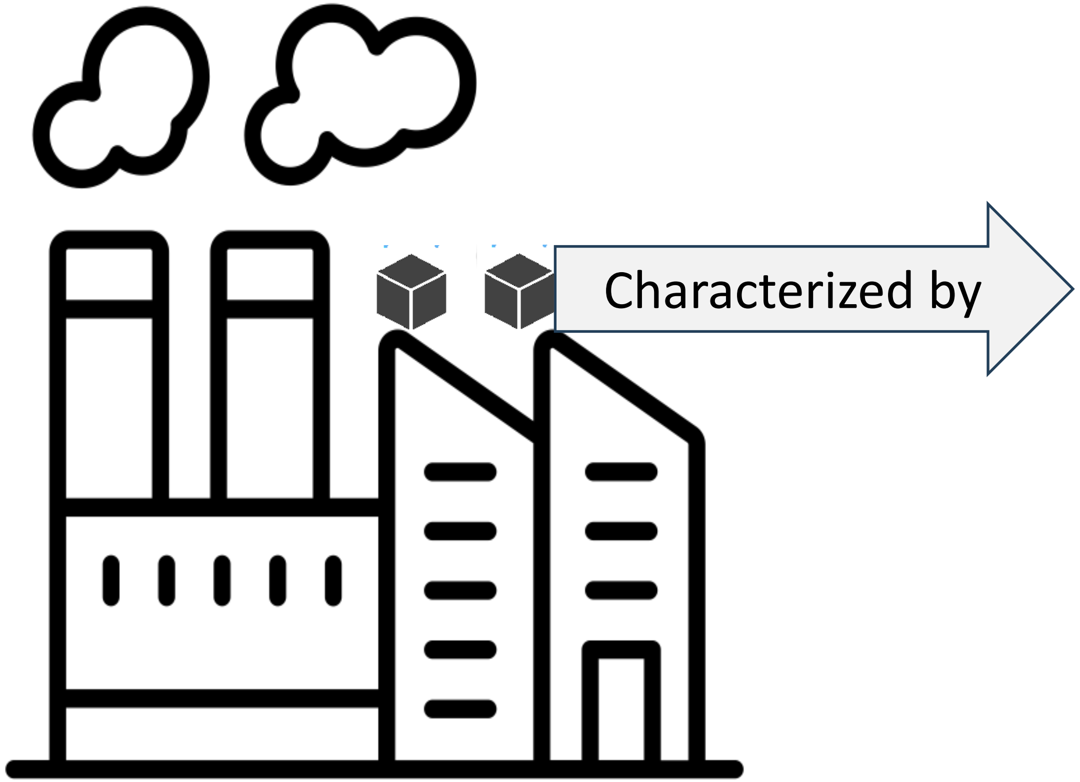
Characterized by

Packets have known fixed deadlines

!!! vs !

Some packets are highly critical, others are only good to deliver

Requirement of Smart Factories
Specific properties that make this problem unique



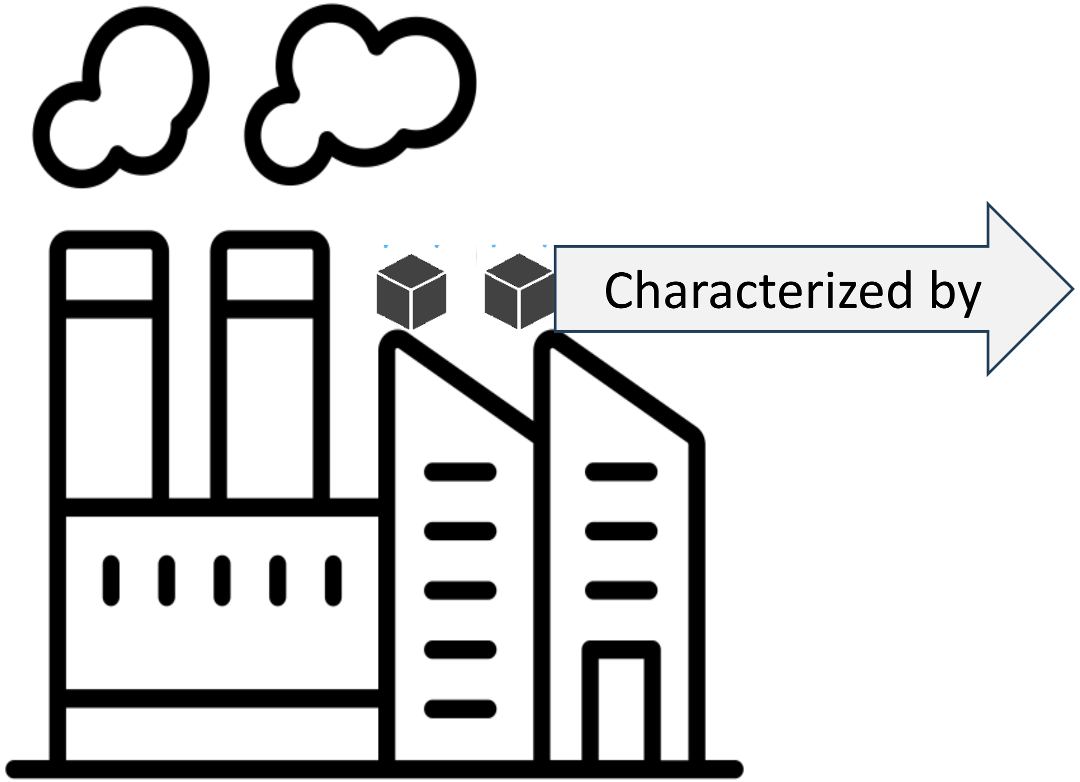
Packets have known fixed deadlines

!!! vs !

Some packets are highly critical, others are only good to deliver

Packets arrive periodically, in known fixed intervals distinct for each user

Requirement of Smart Factories
Specific properties that make this problem unique



Packets have known fixed deadlines

!!! vs !

Some packets are highly critical, others are only good to deliver

Packets arrive periodically, in known fixed intervals distinct for each user

These properties can be used to design intelligent scheduling of packets

Example: The case of Wind Turbine

Report of smart meters and control traffic is much more critical and stringent than logging and video surveillance

Appl.	Gen. rate (pkts/sec)	Size (B)	Dead-line (ms)	Profit	# Nodes
Smart meters	1.25	100	16	10	15
Status info	2.5	100	16	20	15
Reporting & logging	0.75	500	1000	30	15
Data polling	1	500	16	10	15
Control traffic	937.5	100	16	160	20
Video surveil-lance	2000	1500	1000	10	10

Packets have known fixed deadlines

!!! vs !

Some packets are highly critical, others are only good to deliver

Packets arrive periodically, in known fixed intervals distinct for user each

Example: The case of Wind Turbine

Report of smart meters and control traffic is much more critical and stringent than logging and video surveillance

Appl.	Gen. rate (pkts/sec)	Size (B)	Dead-line (ms)	Profit	# Nodes
Smart meters	1.25	100	16	10	15
Status info	2.5	100	16	20	15
Reporting & logging	0.75	500	1000	30	15
Data polling	1	500	16	10	15
Control traffic	937.5	100	16	160	20
Video surveil-lance	2000	1500	1000	10	10

Packets have known fixed deadlines

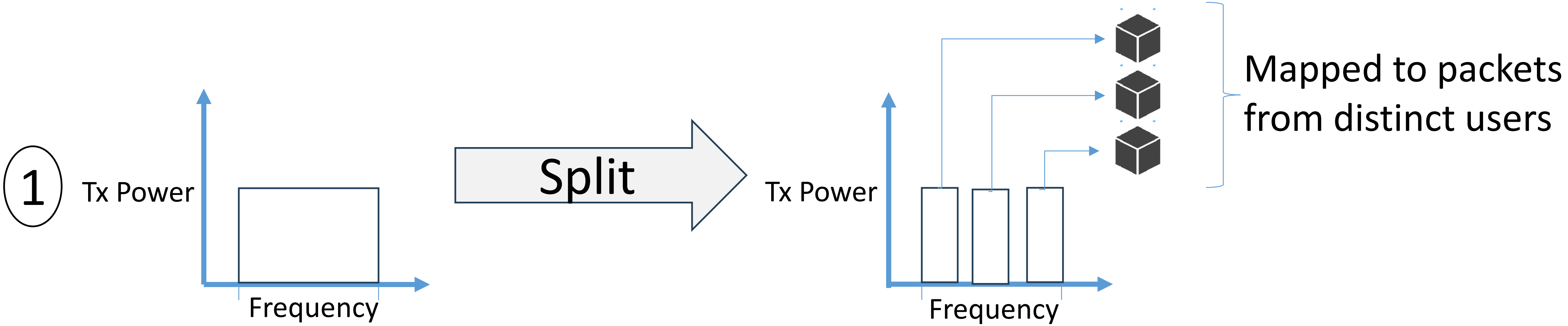
!!! vs !

Some packets are highly critical, others are only good to deliver

Packets arrive periodically, in known fixed intervals distinct for user each

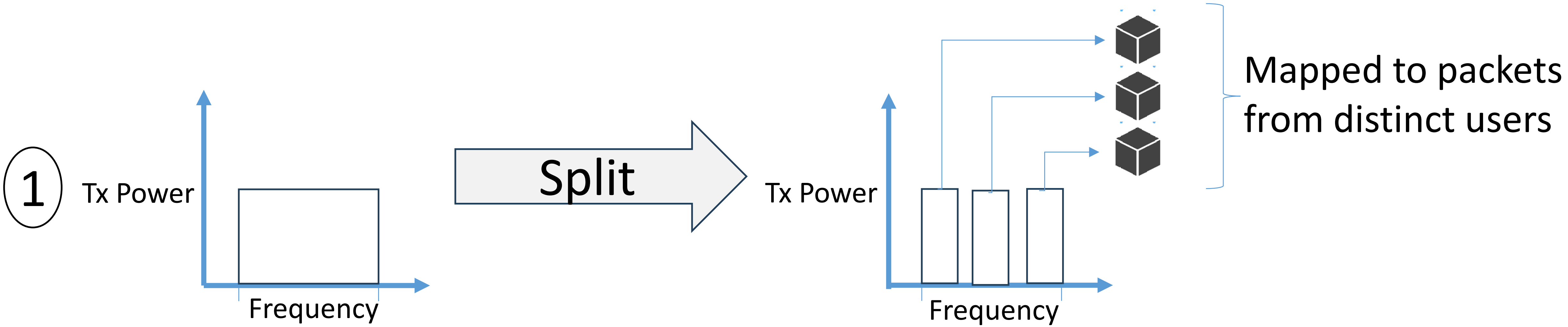
Other examples include metal processing and bottle filling

Challenges of Scheduling Packets over WiFi 6

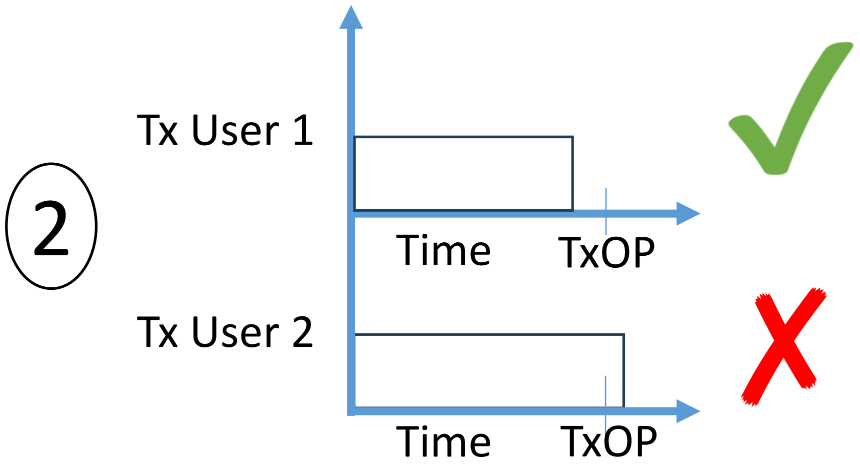


Bandwidth is split into discrete RUs

Challenges of Scheduling Packets over WiFi 6

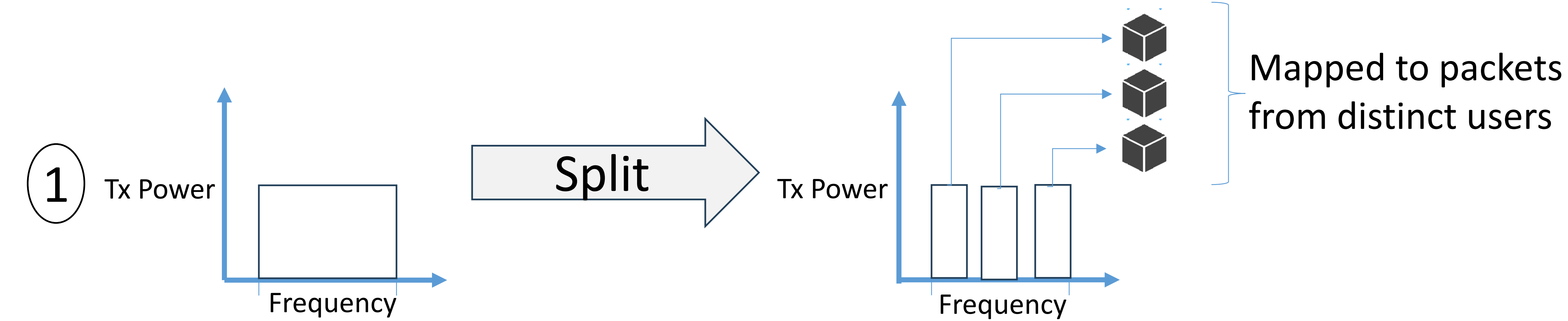


Bandwidth is split into discrete RUs

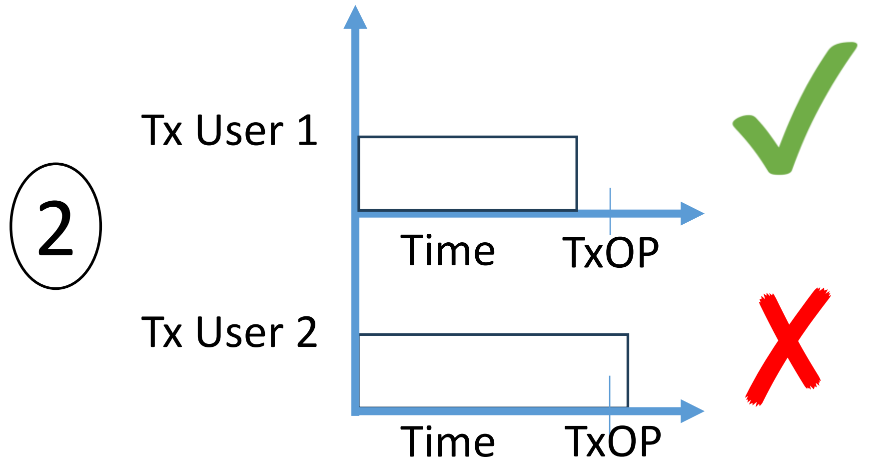


Parallel transmissions must finish within TxOP

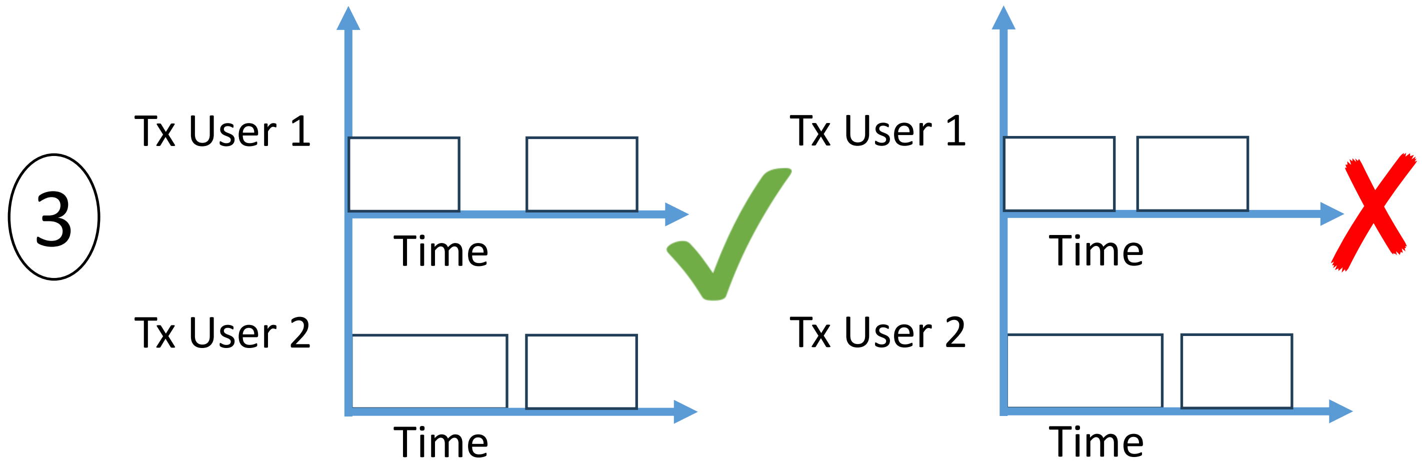
Challenges of Scheduling Packets over WiFi 6



Bandwidth is split into discrete RUs



Parallel transmissions must finish within TxOP



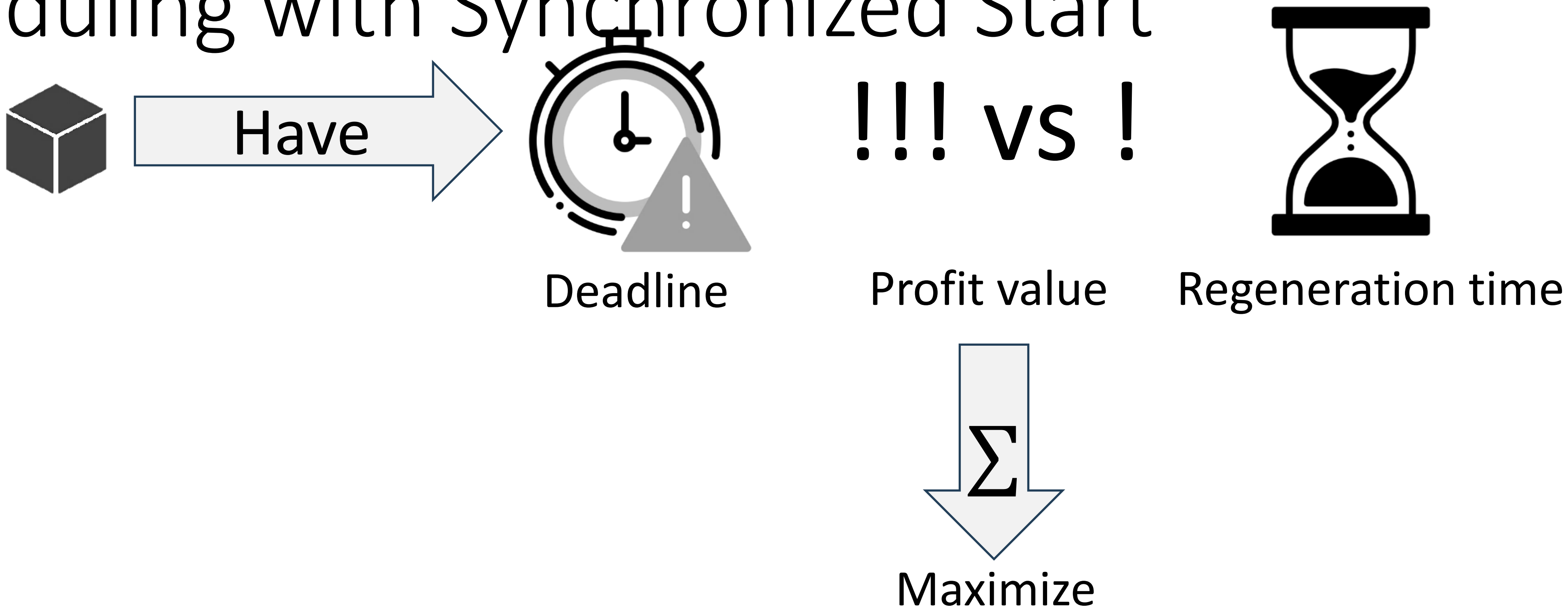
Start of each transmission must be synchronized

DPMSS: Deadline-aware Parallel Machines Scheduling with Synchronized Start

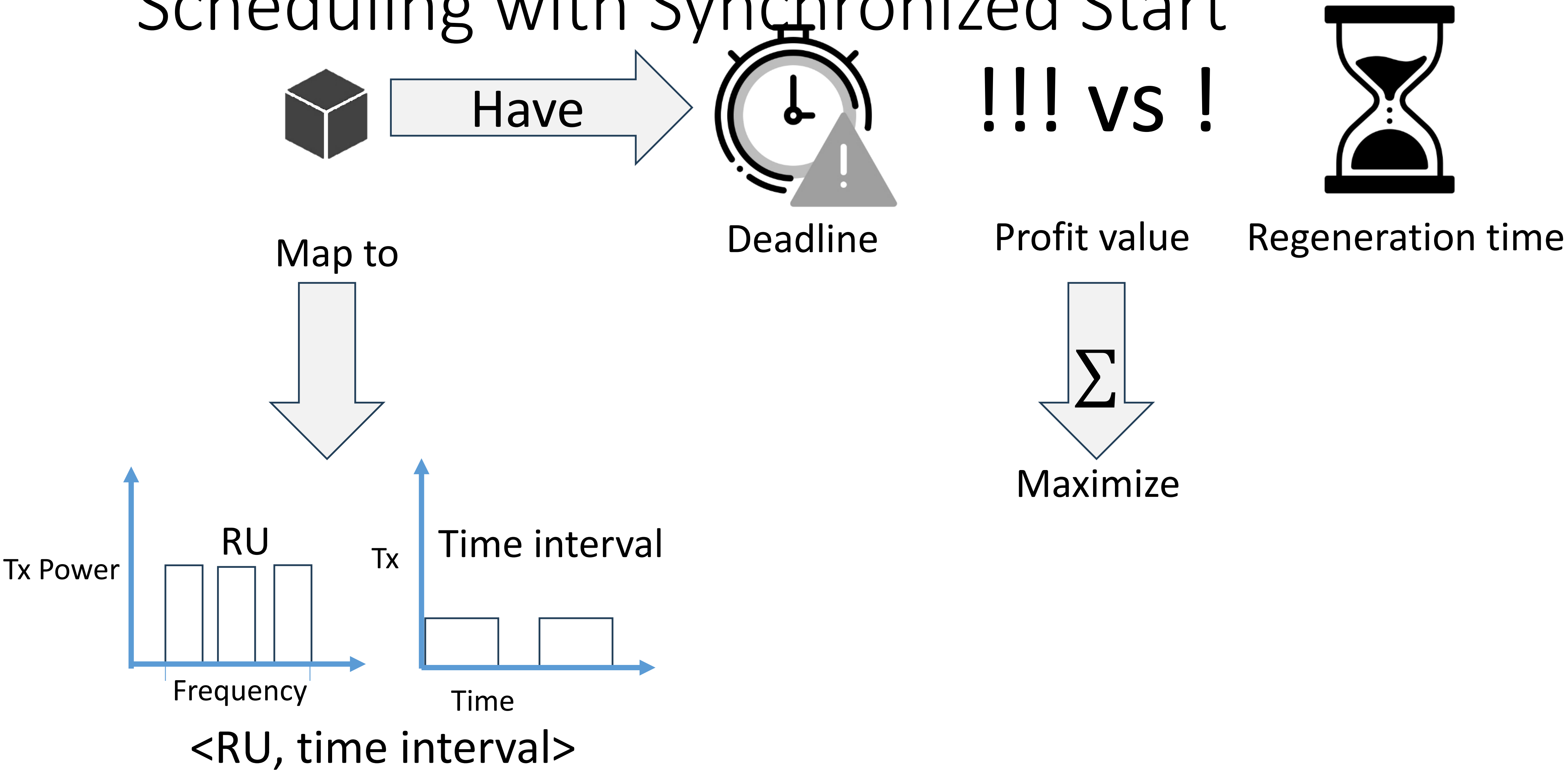
DPMSS: Deadline-aware Parallel Machines Scheduling with Synchronized Start

Deadline Profit value Regeneration time

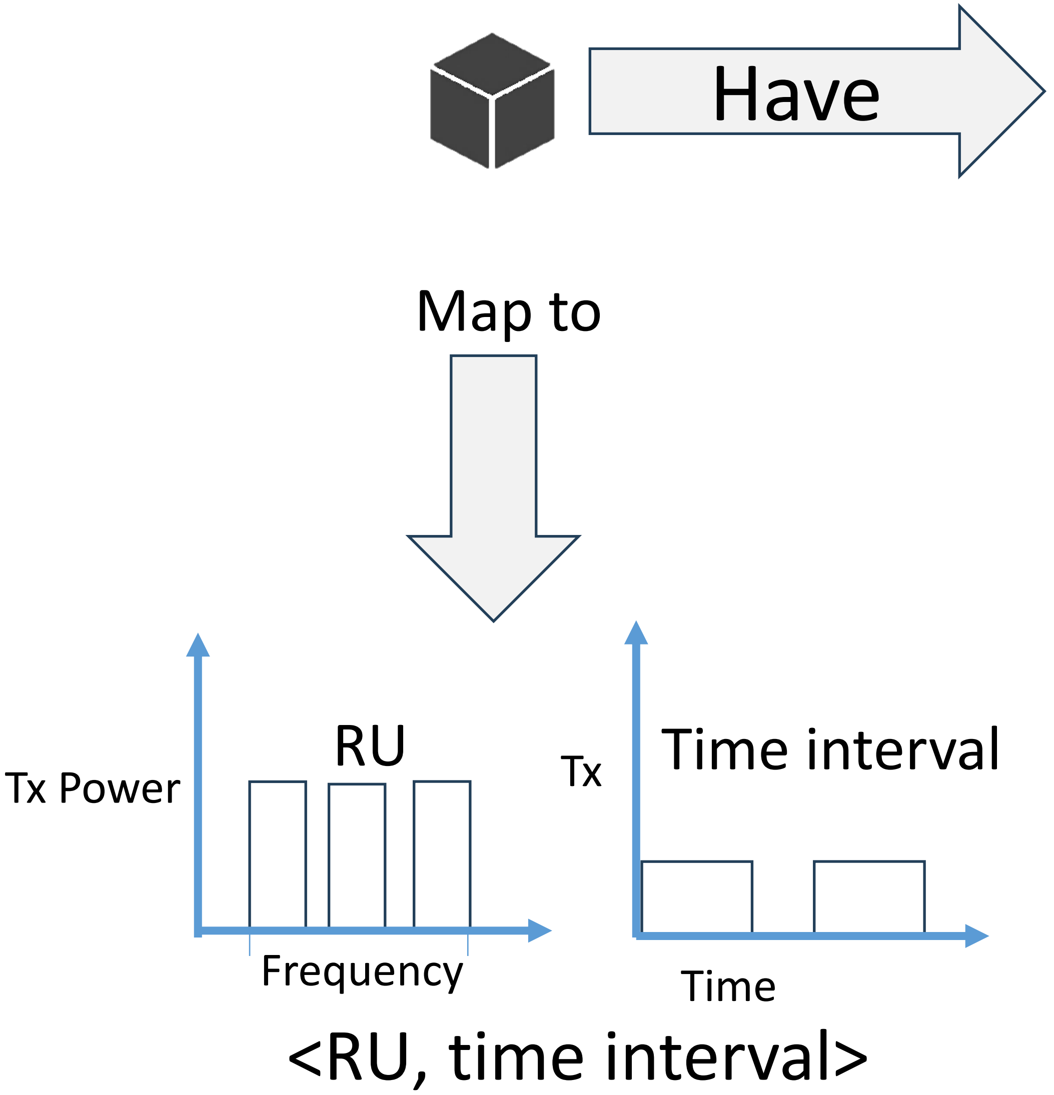
DPMSS: Deadline-aware Parallel Machines Scheduling with Synchronized Start

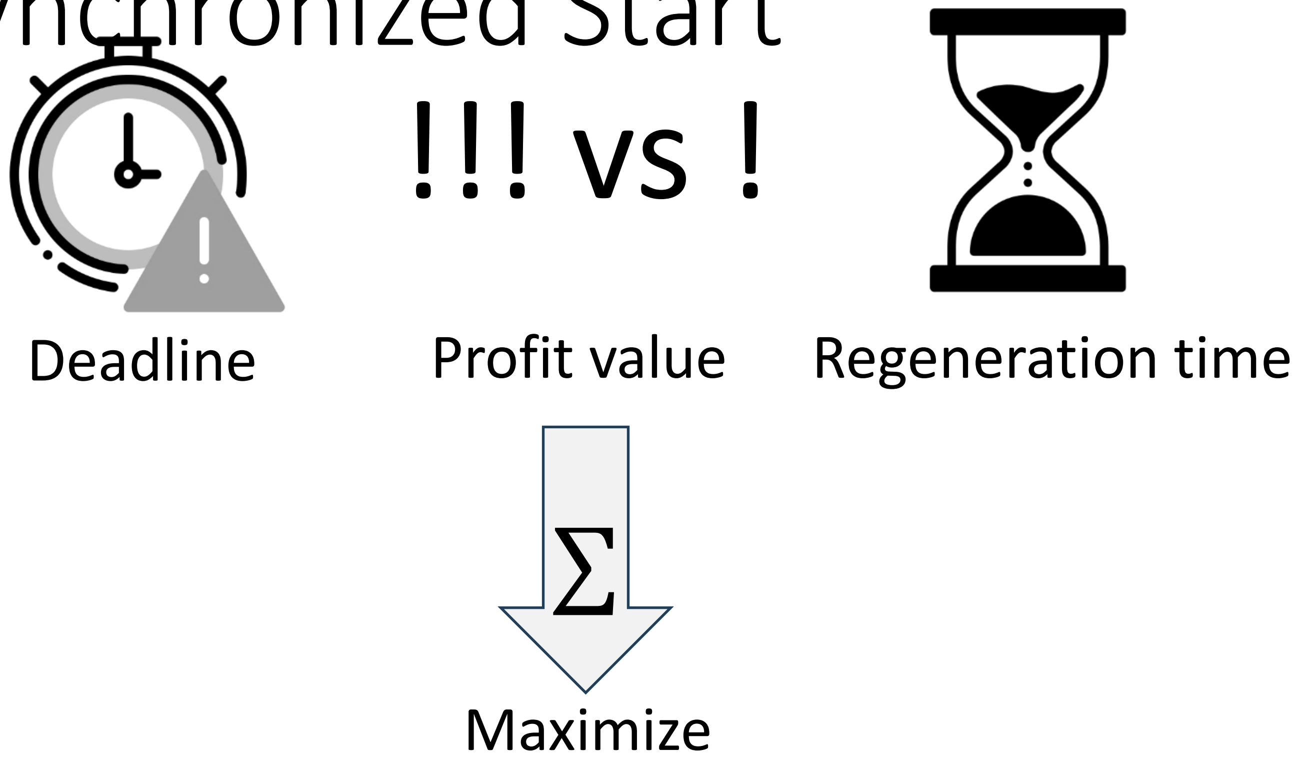


DPMSS: Deadline-aware Parallel Machines Scheduling with Synchronized Start



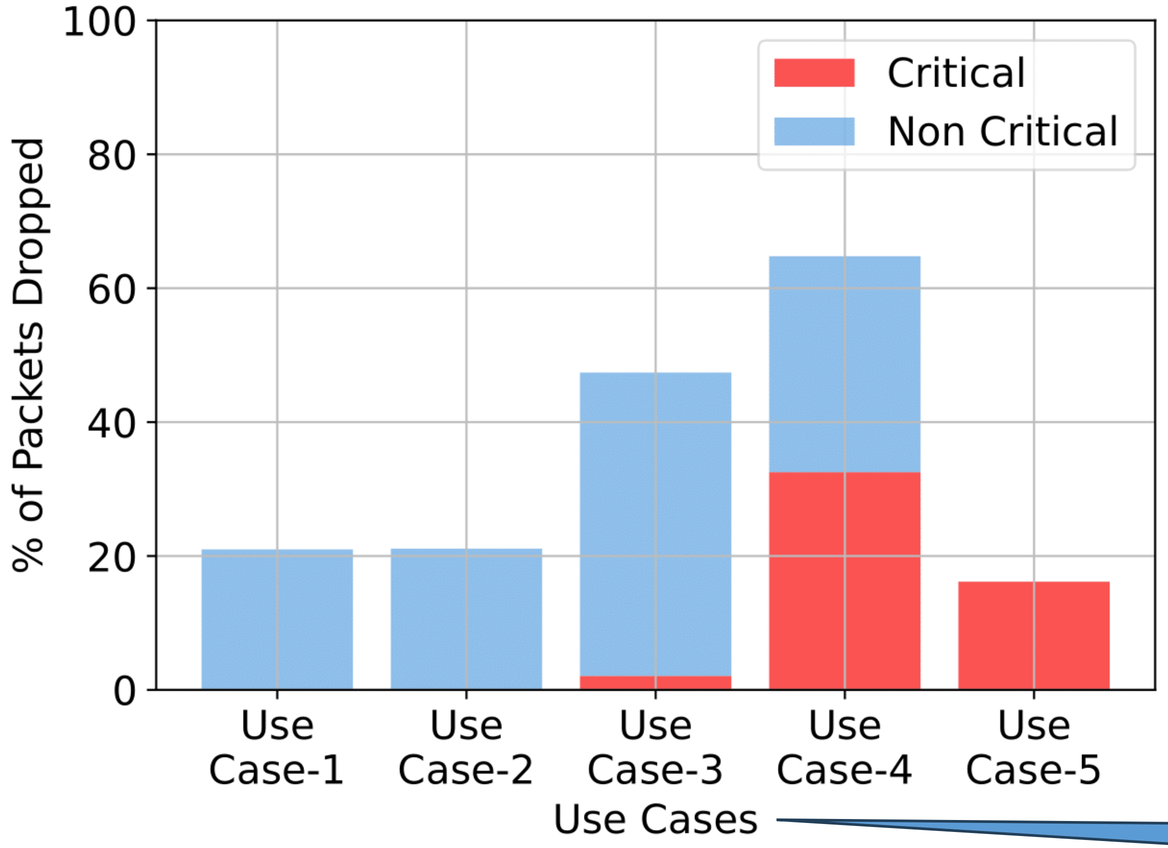
DPMSS: Deadline-aware Parallel Machines Scheduling with Synchronized Start





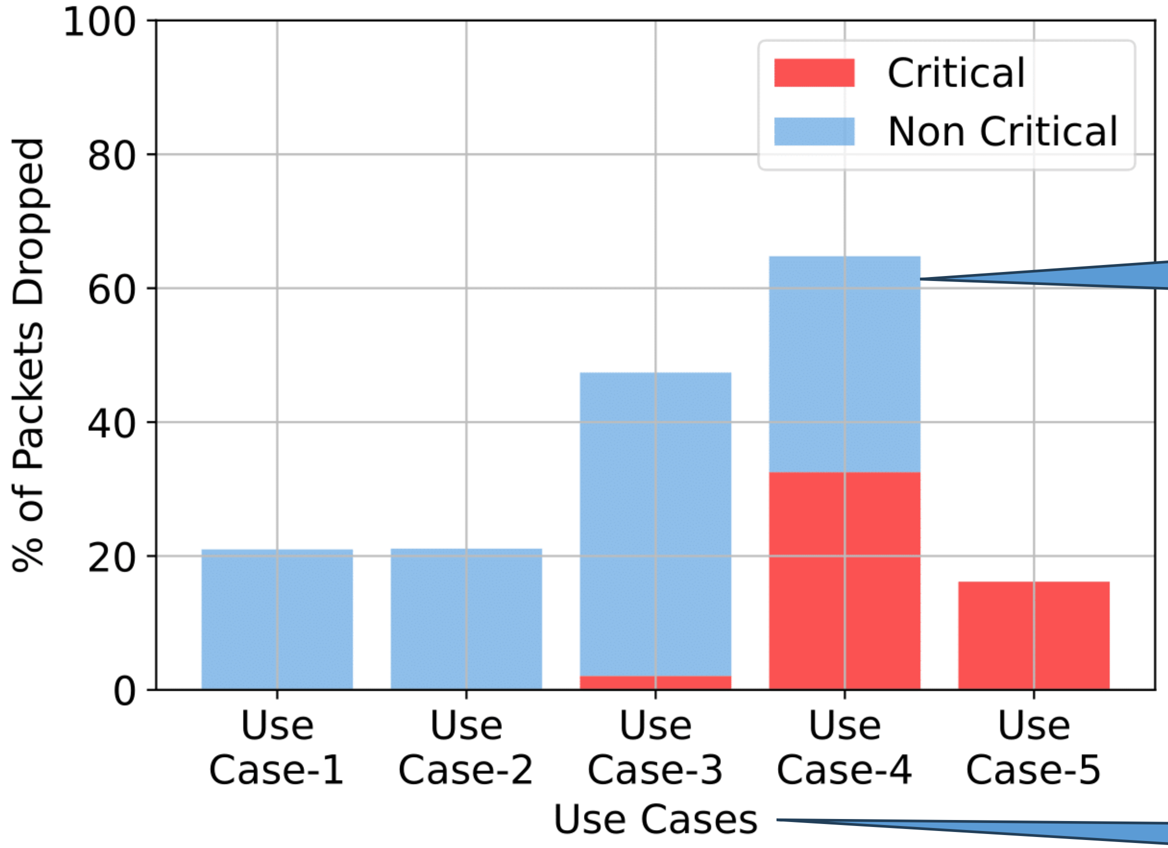
Can be formally defined as an integer linear programming problem, with decision variables being RUs and time intervals used

Simple Approach of Scheduling based on Deadline Provides Poor Results



Represent trace of different factories

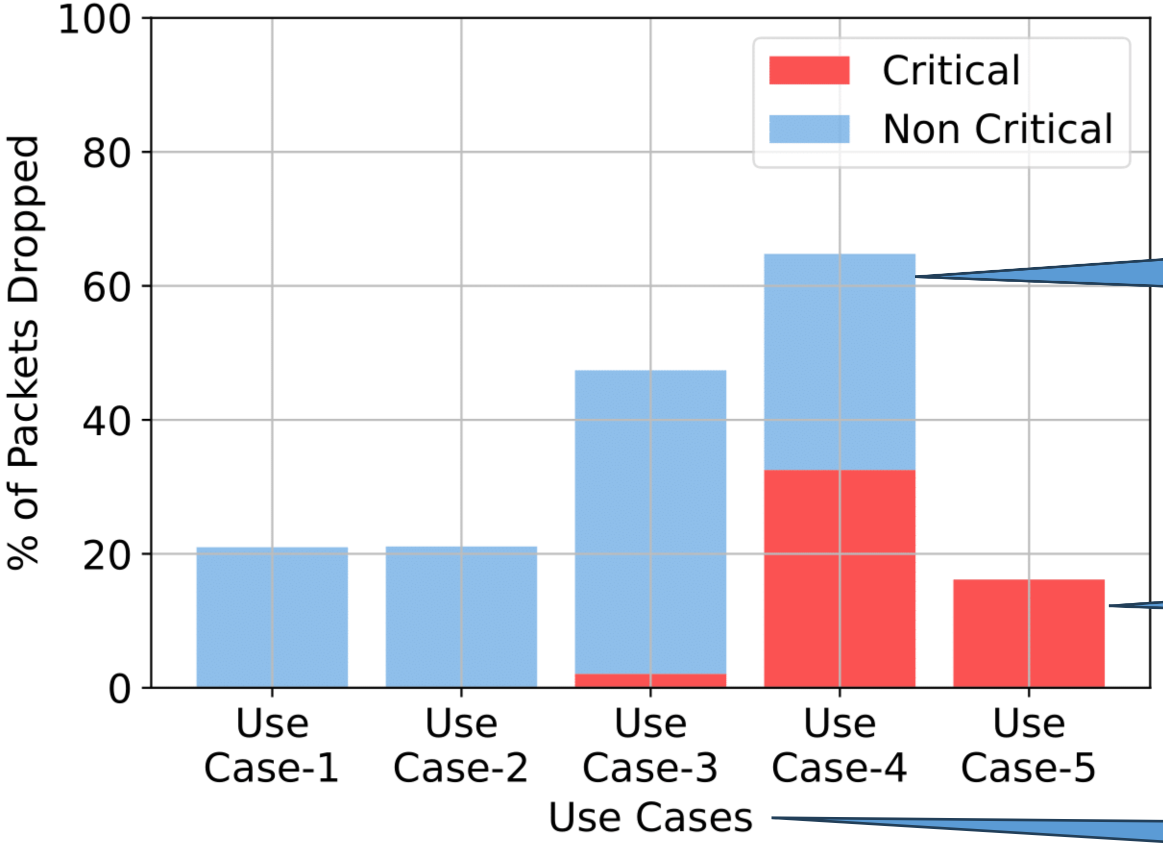
Simple Approach of Scheduling based on Deadline Provides Poor Results



High number of packet losses

Represent trace of different factories

Simple Approach of Scheduling based on Deadline Provides Poor Results

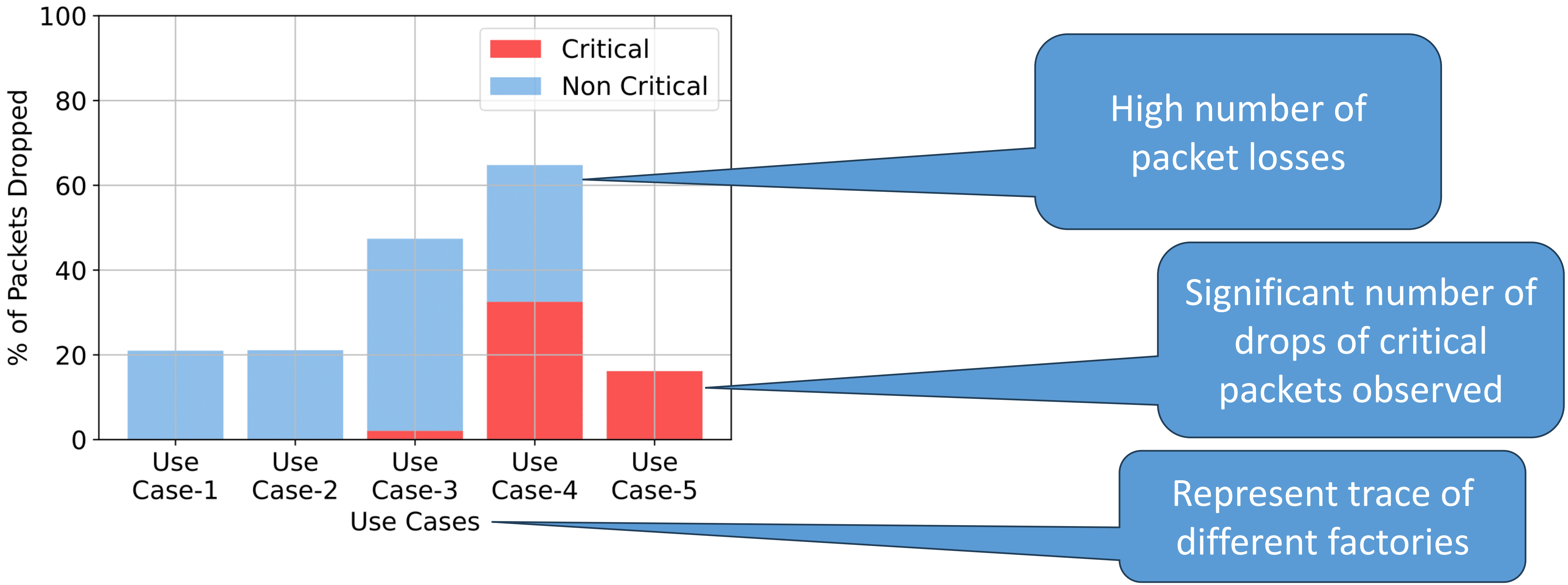


High number of packet losses

Significant number of drops of critical packets observed

Represent trace of different factories

Simple Approach of Scheduling based on Deadline Provides Poor Results

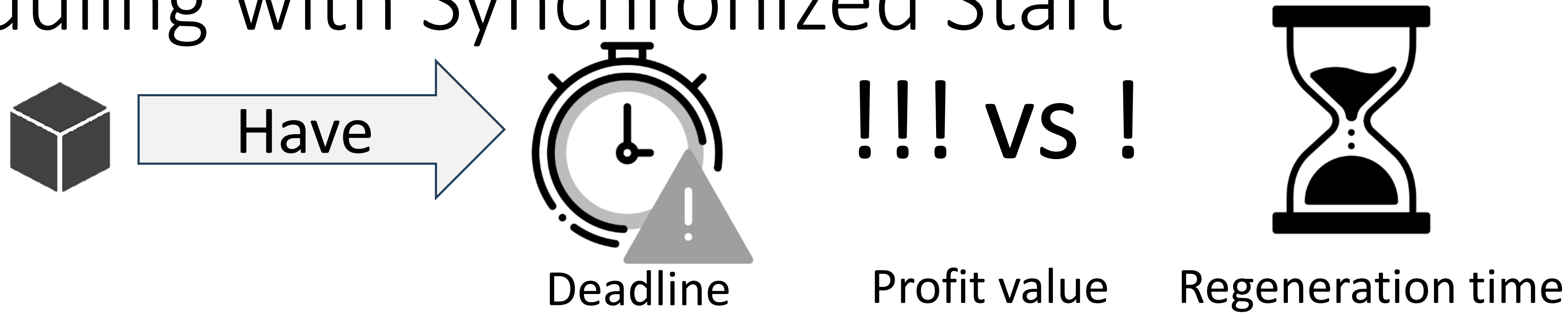


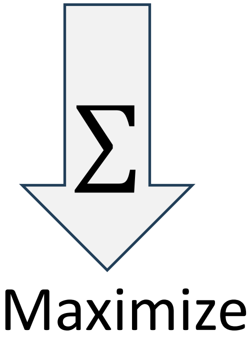
Better strategies of scheduling are needed to avoid dropping of critical packets

Content

- Introduction and Problem Formulation: Using WiFi 6 in Smart Factory
- Solution Approach
- Results
- Summary

DPMSS: Deadline-aware Parallel Machines Scheduling with Synchronized Start

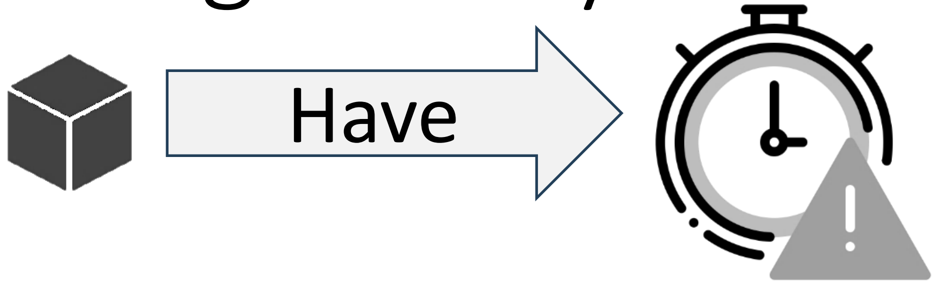




Obvious Solution: Group jobs/packets of similar deadlines and profits, and then choose the right RU/machine for each group

Can perform arbitrarily bad compared to optimal

DPMSS: Deadline-aware Parallel Machines Scheduling with Synchronized Start

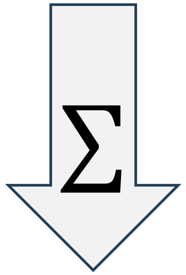


Deadline

!!! vs !

Profit value

Regeneration time



Maximize

Obvious Solution: Group jobs/packets of similar deadlines and profits, and then choose the right RU/machine for each group

Can perform arbitrarily bad compared to optimal

Novel problem even in the context of scheduling due to synchronized start

DPMSS: Feasibility and Hardness

DPMSS: Feasibility and Hardness

Feasible Solution:
(1) Disjoint time intervals
(2) A set of packets/jobs mapped to a set of RUs/machines for each time interval

Within an interval, no machine is assigned for than a job
Total bandwidth allocated within be within the budget
No job belongs to two time intervals

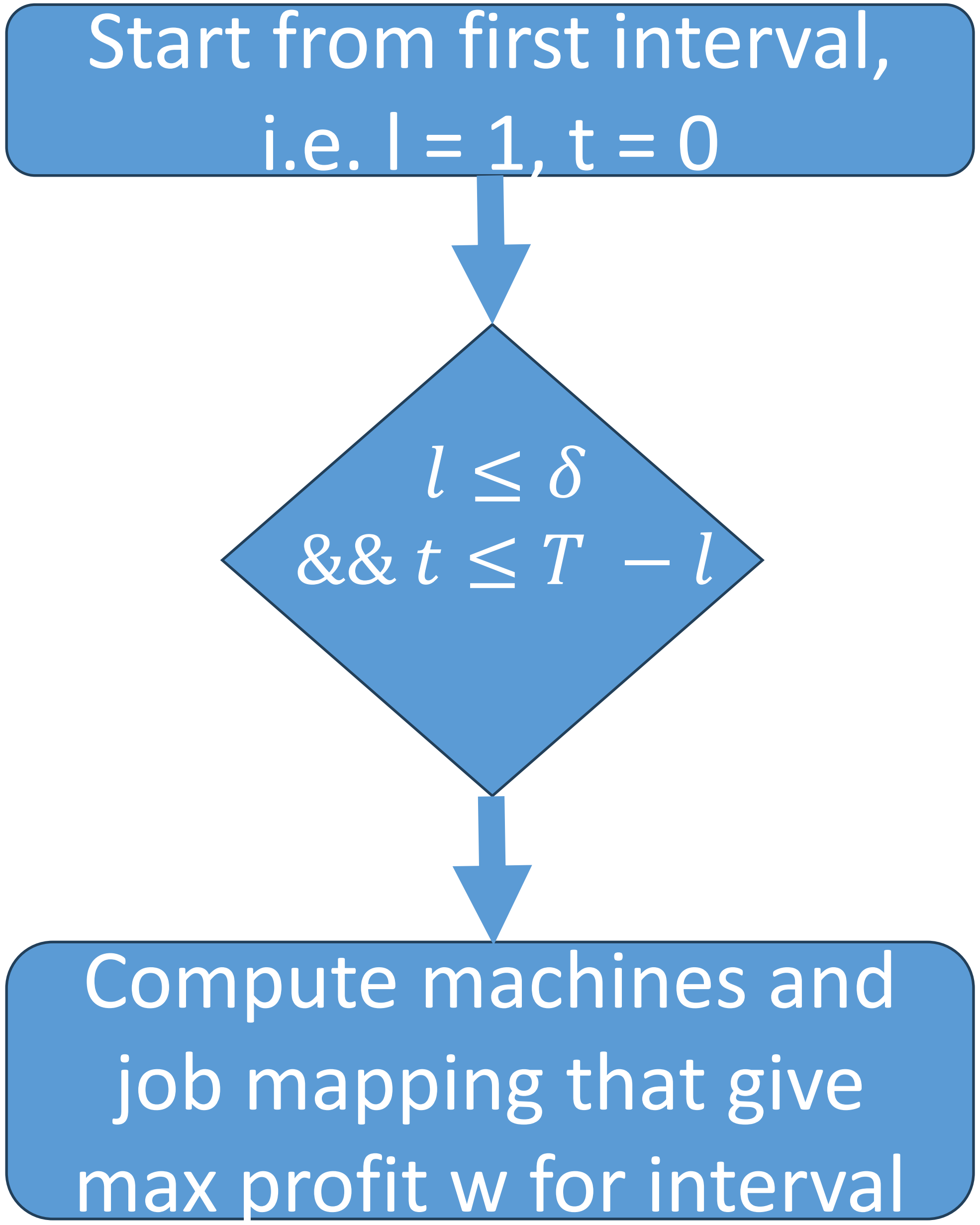
DPMSS: Feasibility and Hardness

Feasible Solution:
(1) Disjoint time intervals
(2) A set of packets/jobs mapped to a set of RUs/machines for each time interval

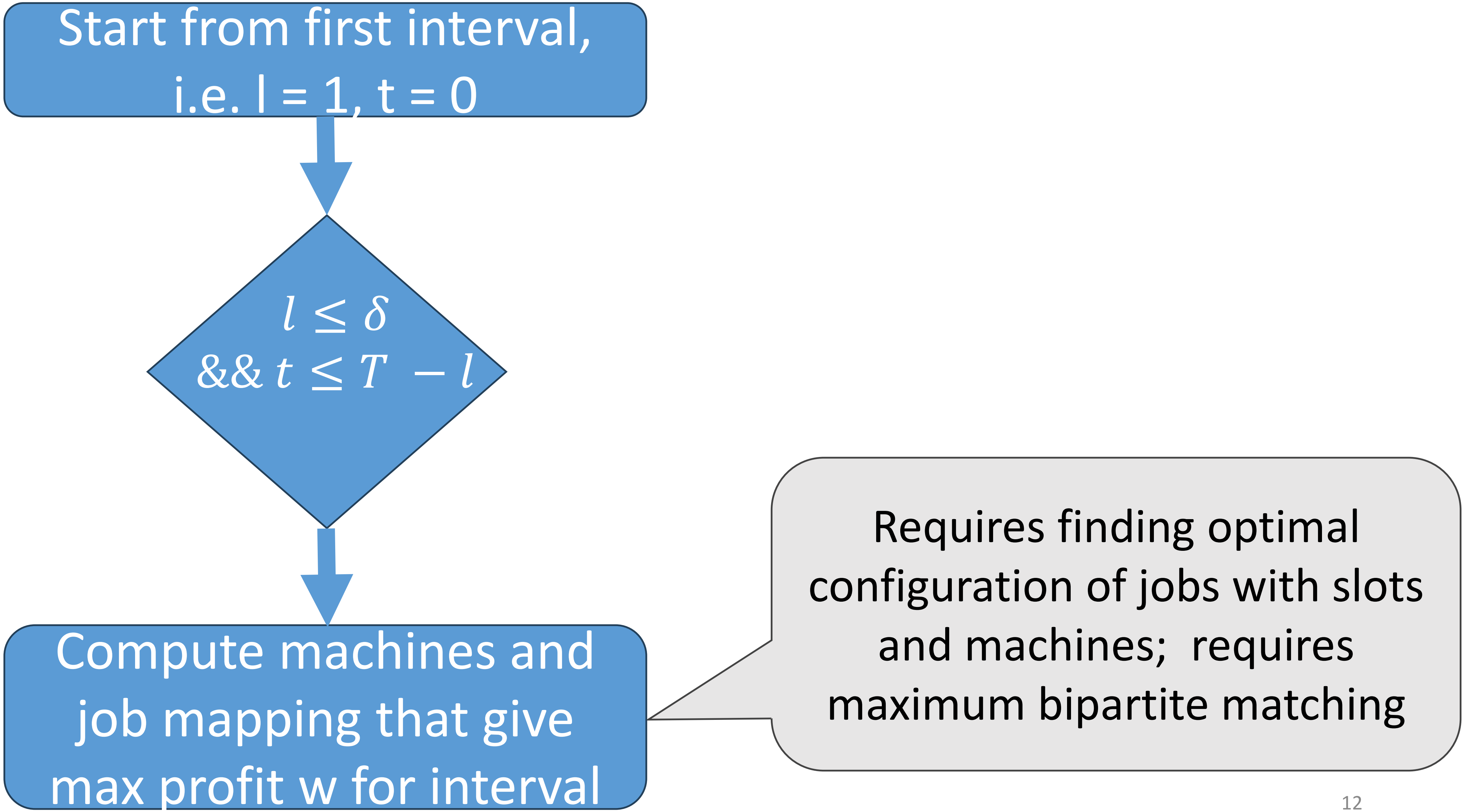
Within an interval, no machine is assigned for than a job
Total bandwidth allocated within be within the budget
No job belongs to two time intervals

DPMSS is (strongly) NP-Hard
Follows from single machine non-preemptive scheduling

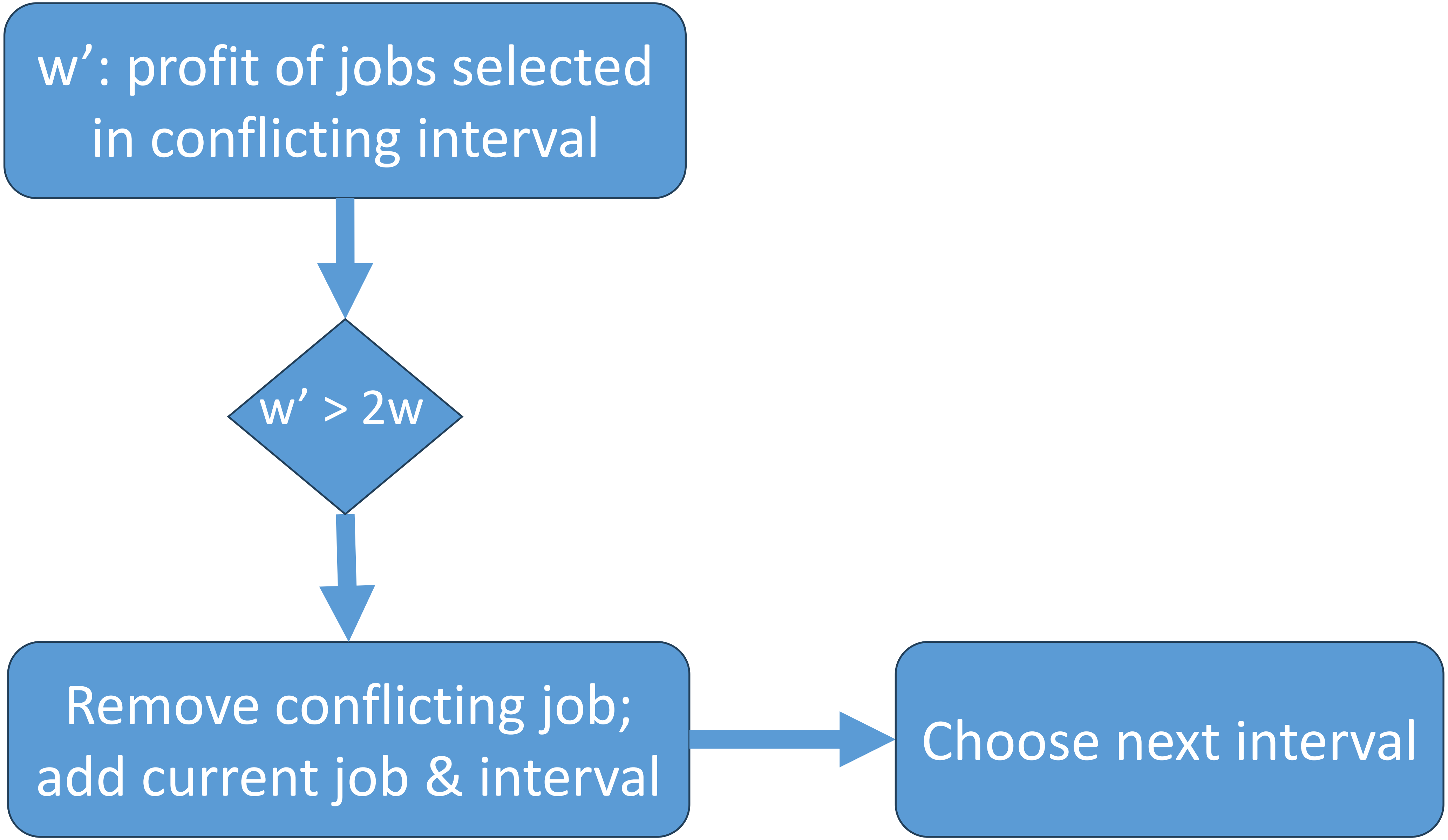
Algorithmic Strategy: first for fixed RUs



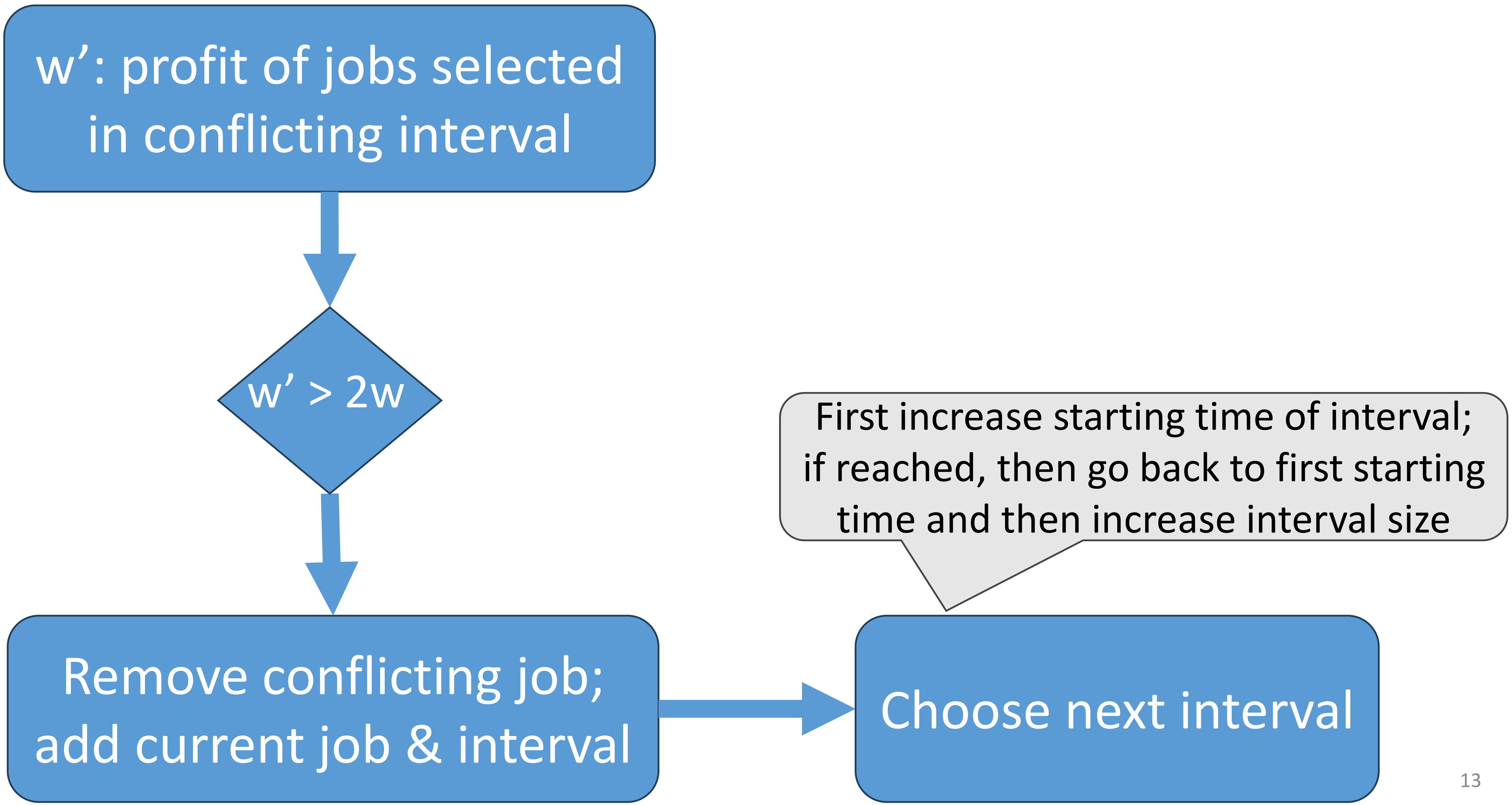
Algorithmic Strategy: first for fixed RUs



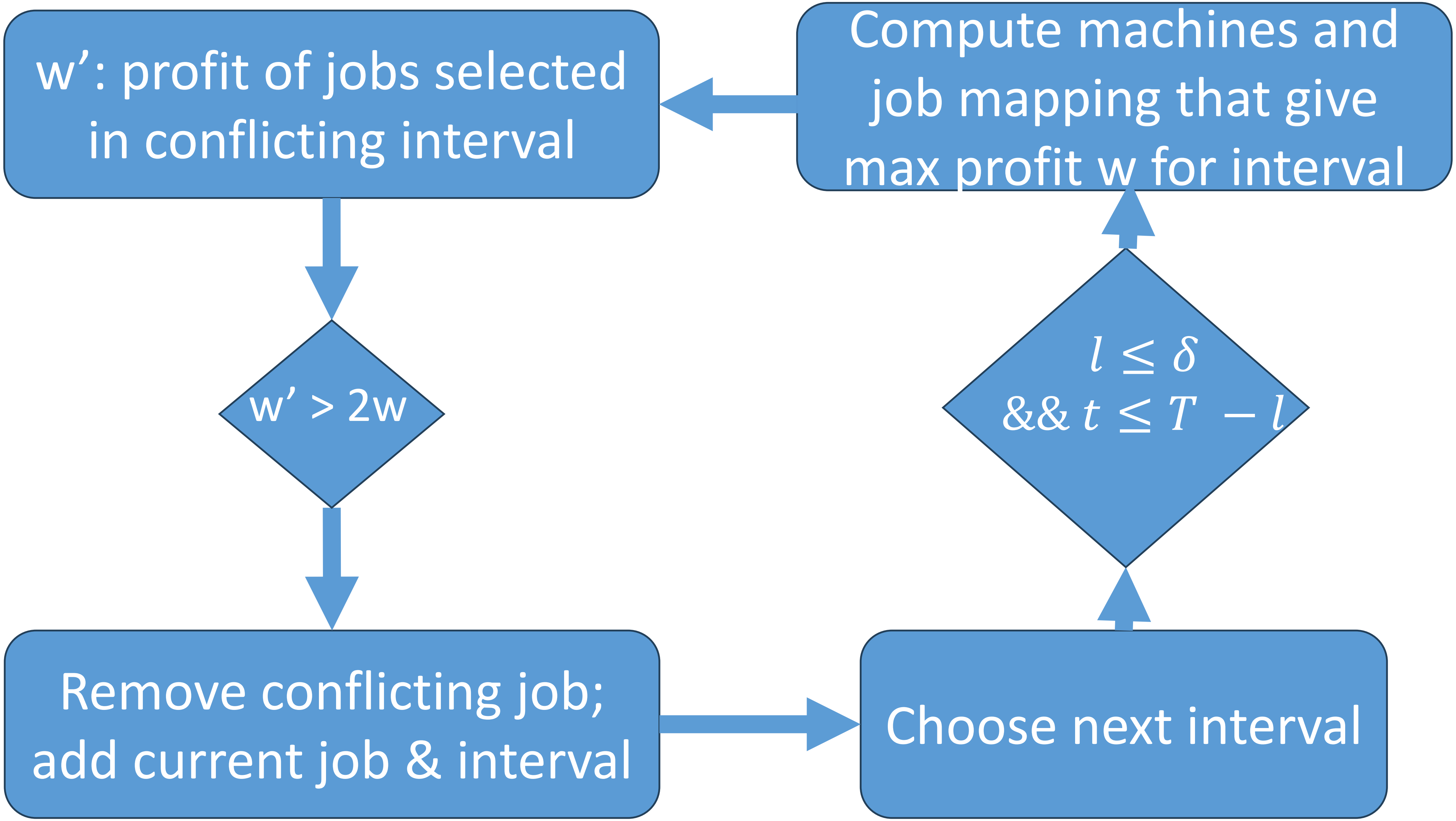
Algorithmic Strategy: first for fixed RUs



Algorithmic Strategy: first for fixed RUs



Algorithmic Strategy: first for fixed RUs



Properties of our Algorithm

- Our algorithm is feasible
  - No conflicting intervals are chosen
  - Only admissible jobs are chosen
- Process of local search
  - We look at smaller intervals first, and schedule as much as we can
  - A larger interval is acceptable only if it provides twice as much profit
- DPMSS provides a 12-approximate solution

Idea of 12-Approximation

Enumeration across four categories of packets:

- J_1 – Jobs/Packets never chosen by DPMSS; chosen by optimal solution
- J_2 – Jobs/Packets added initially but then discarded; also chosen by optimal solution
- J_3 – Jobs/packets that are present in DPMSS; also chosen by optimal solution
- J_4 – Jobs/Packets that are present in DPMSS; not chosen by optimal solution
- $w(J^*) = w(J_1) + w(J_2) + w(J_3)$; $w(J_a) = w(J_3) + w(J_4)$

Idea of 12-Approximation

Enumeration across four categories of packets:

- J_1 – Jobs/Packets never chosen by DPMSS; chosen by optimal solution
- J_2 – Jobs/Packets added initially but then discarded; also chosen by optimal solution
- J_3 – Jobs/packets that are present in DPMSS; also chosen by optimal solution
- J_4 – Jobs/Packets that are present in DPMSS; not chosen by optimal solution
- $w(J^*) = w(J_1) + w(J_2) + w(J_3)$; $w(J_a) = w(J_3) + w(J_4)$

Lemma 1: $w(J_1) \leq 5[w(J_2) + w(J_3) + w(J_4)]$

Obtained by enumerating all possible intervals

Idea of 12-Approximation

Enumeration across four categories of packets:

- J_1 – Jobs/Packets never chosen by DPMSS; chosen by optimal solution
- J_2 – Jobs/Packets added initially but then discarded; also chosen by optimal solution
- J_3 – Jobs/packets that are present in DPMSS; also chosen by optimal solution
- J_4 – Jobs/Packets that are present in DPMSS; not chosen by optimal solution
- $w(J^*) = w(J_1) + w(J_2) + w(J_3)$; $w(J_a) = w(J_3) + w(J_4)$

Lemma 1: $w(J_1) \leq 5[w(J_2) + w(J_3) + w(J_4)]$

Obtained by enumerating all possible intervals

Lemma 2: $w(J_2) \leq w(J_a)$

Obtained by quantifying the highest possible profit that is lost by discarding

Idea of 12-Approximation

Enumeration across four categories of packets:

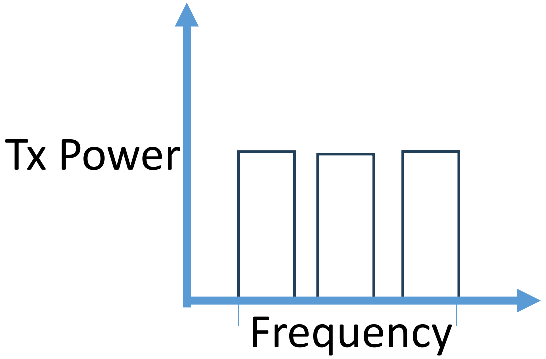
- J_1 – Jobs/Packets never chosen by DPMSS; chosen by optimal solution
- J_2 – Jobs/Packets added initially but then discarded; also chosen by optimal solution
- J_3 – Jobs/packets that are present in DPMSS; also chosen by optimal solution
- J_4 – Jobs/Packets that are present in DPMSS; not chosen by optimal solution
- $w(J^*) = w(J_1) + w(J_2) + w(J_3)$; $w(J_a) = w(J_3) + w(J_4)$

Lemma 1: $w(J_1) \leq 5[w(J_2) + w(J_3) + w(J_4)]$
Obtained by enumerating all possible intervals

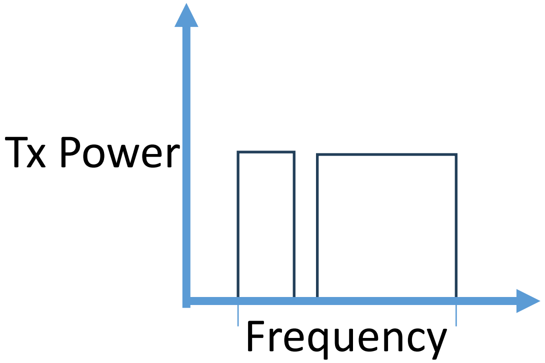
Lemma 2: $w(J_2) \leq w(J_a)$
Obtained by quantifying the highest possible profit that is lost by discarding

Theorem: DPMSSF provides a 12-approximate solution
$w(J^*) \leq 12\, w(J_a)$

Generalizability to Heterogeneous RUs

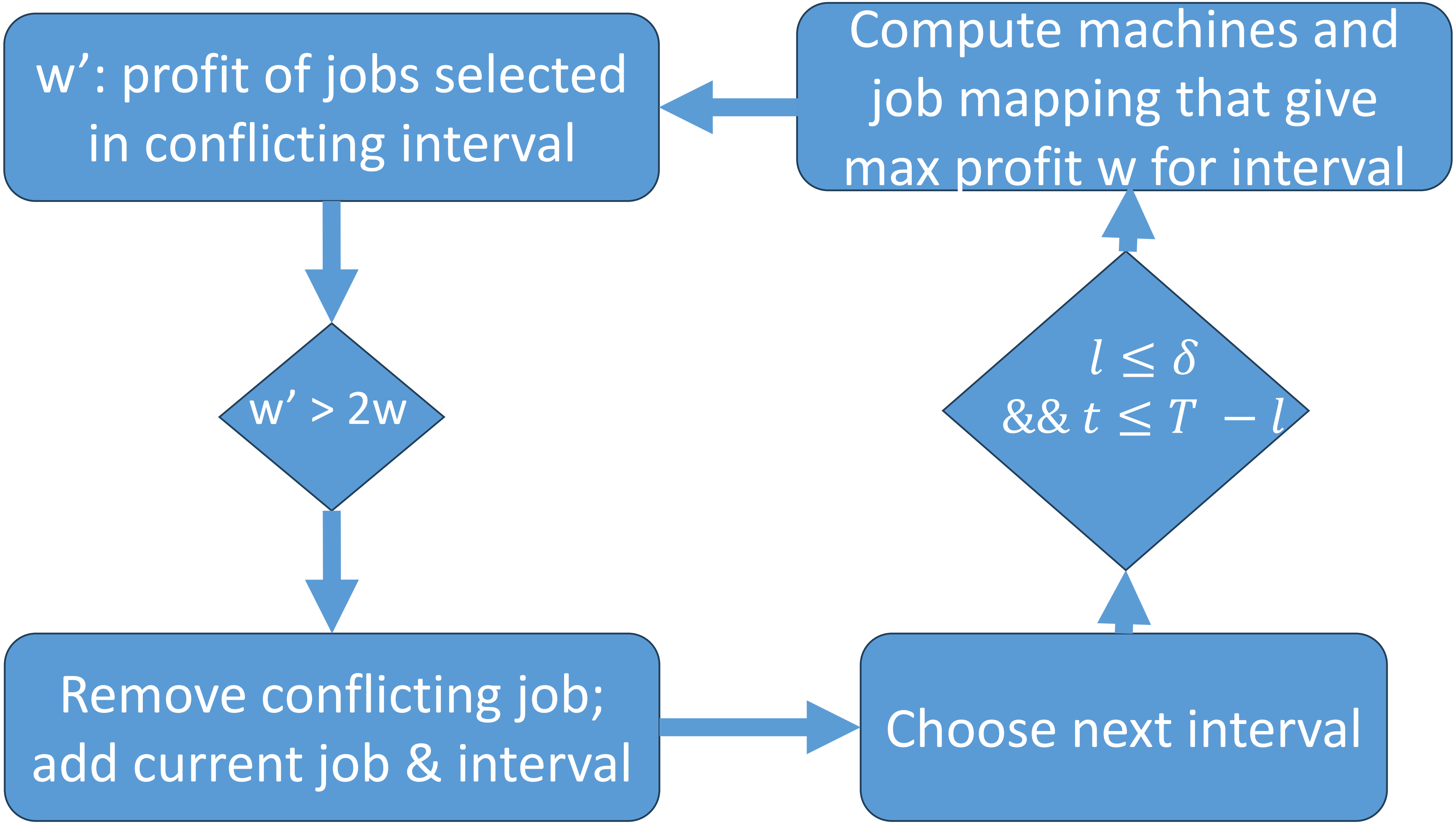


vs

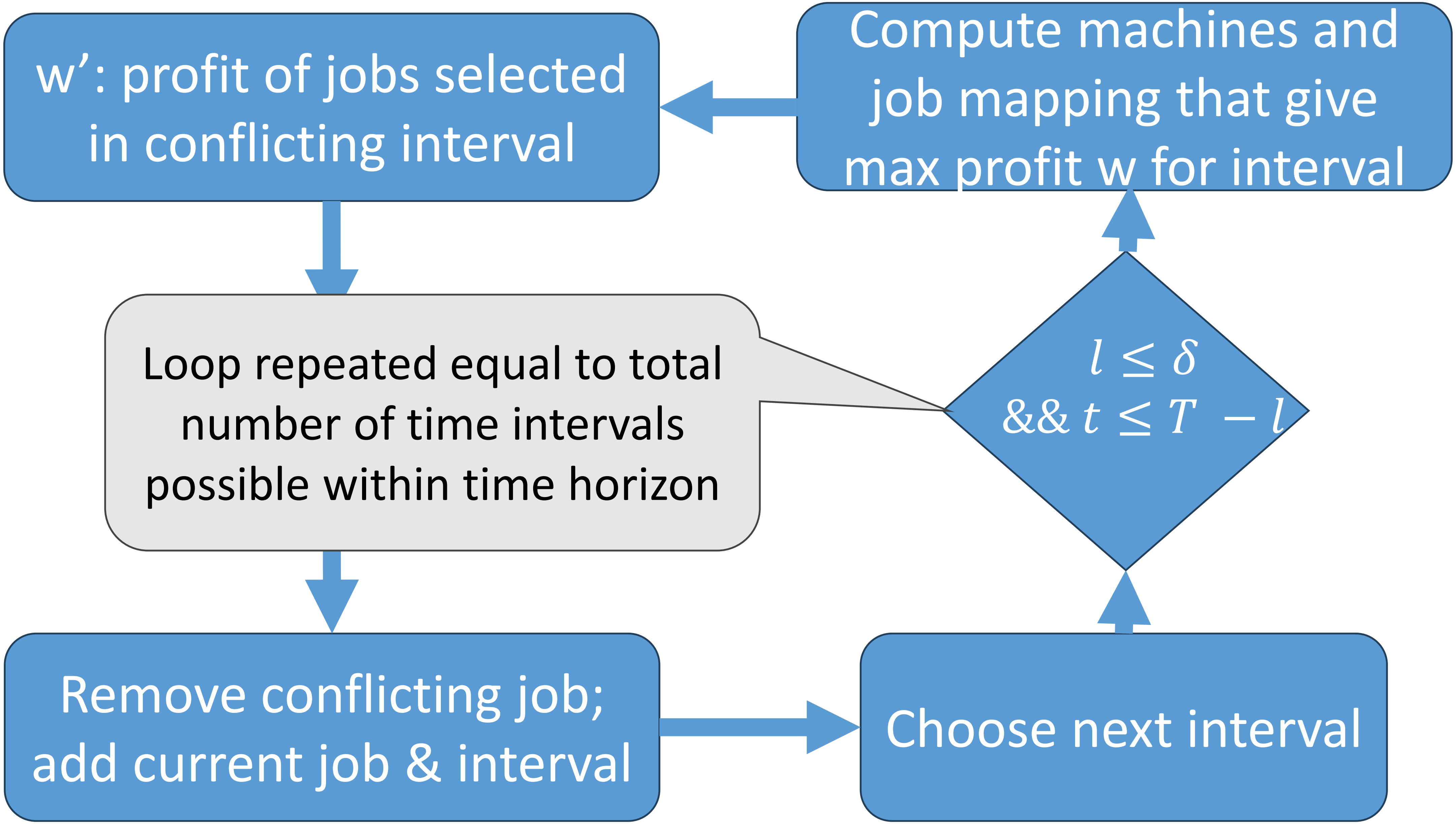


- Adds a knapsack constraint to bipartite matching
  - Best configuration can no longer be solved by bipartite matching
- Can be solved using a polynomial task-approximation scheme (PTAS)
  - Leads to (12 + ε)-approximation for DPMSS
- In practice, we solve using exhaustive search
  - Most cases solved within 300 ms and all cases within 1s on Raspberry Pi 3B

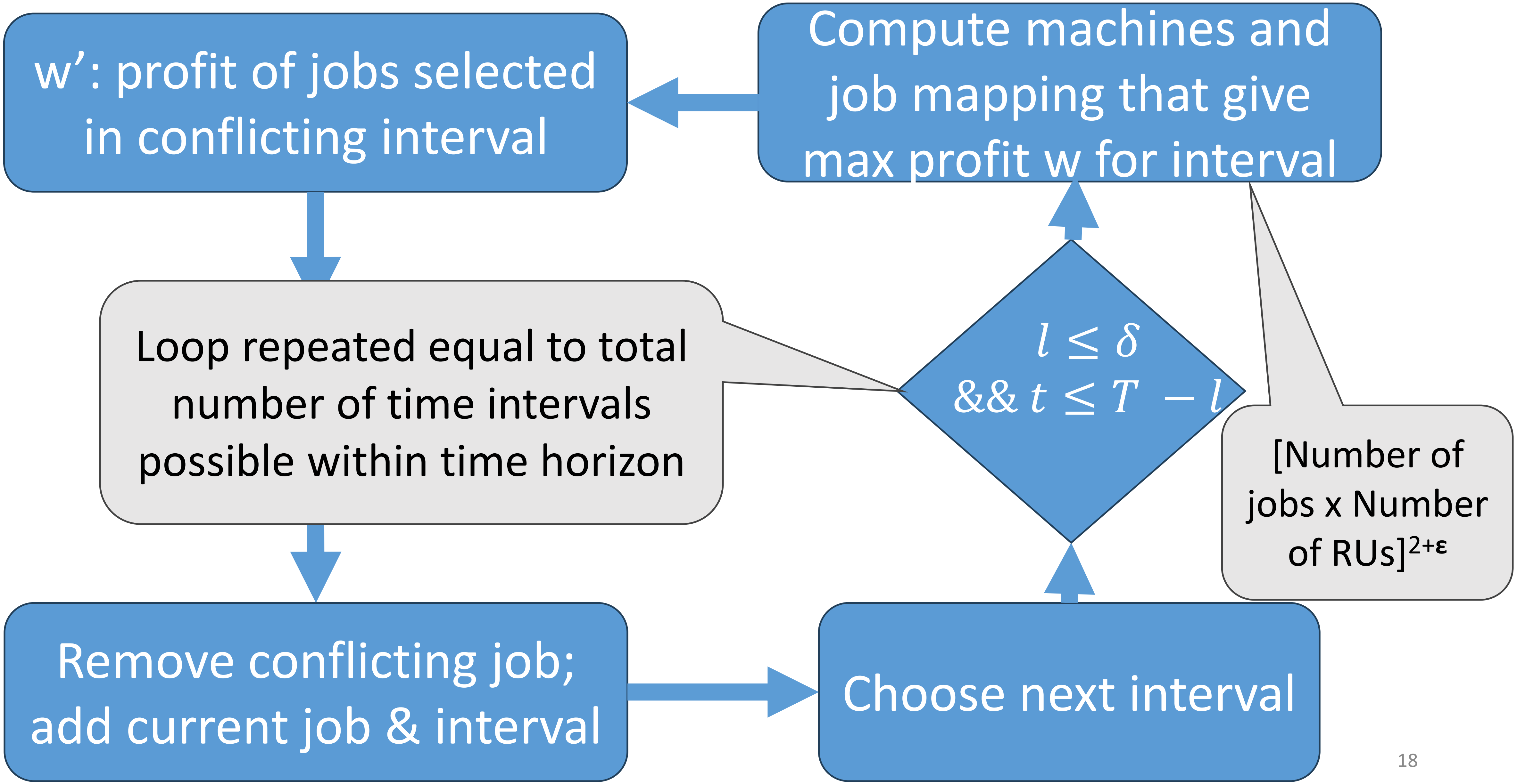
Time Complexity



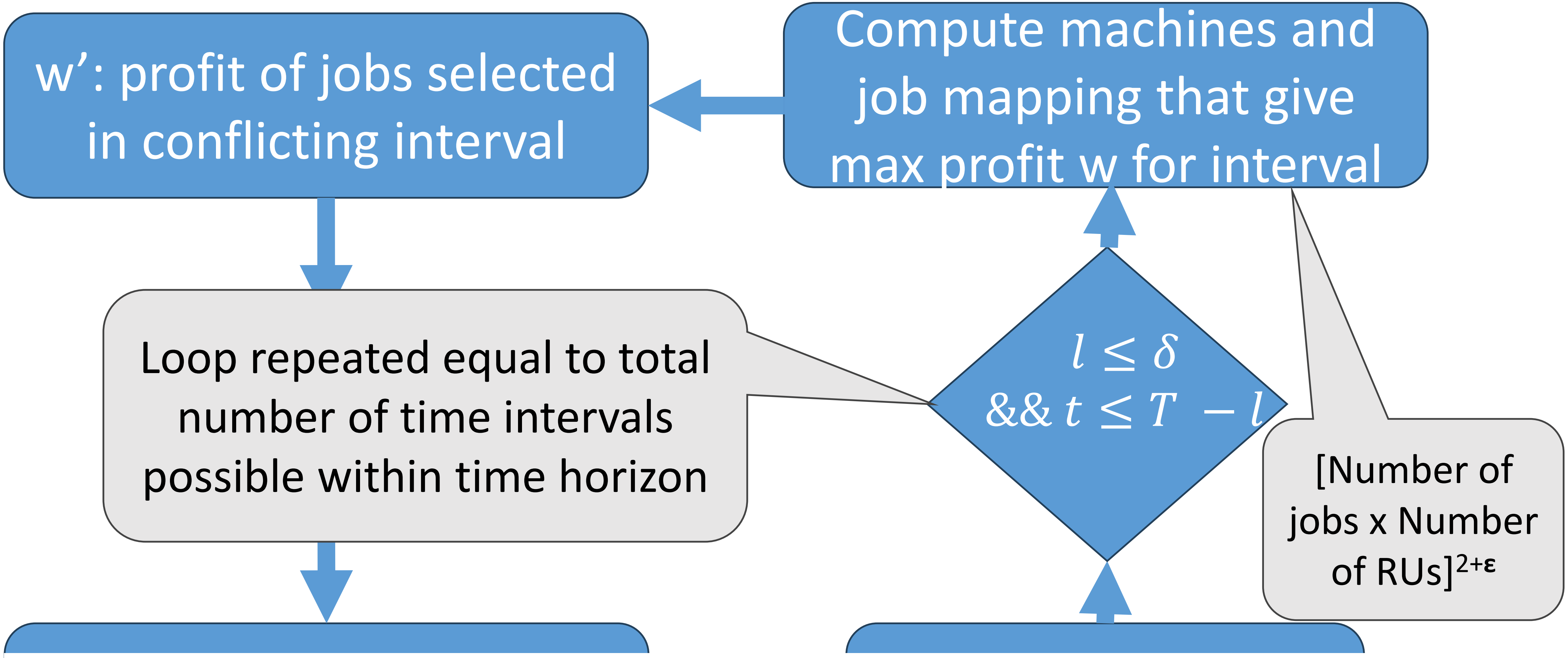
Time Complexity



Time Complexity



Time Complexity

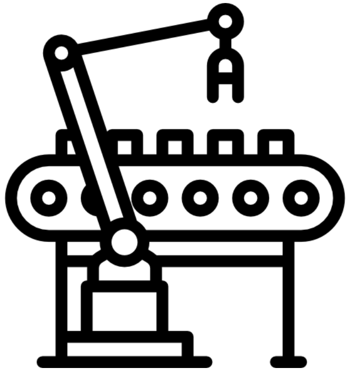


Overall time complexity:
O(number of intervals x (number of jobs x number of RUs)$^{2+\varepsilon}$)

Content

- Introduction and Problem Formulation: Using WiFi 6 in Smart Factory
- Solution Approach
- Results
- Summary

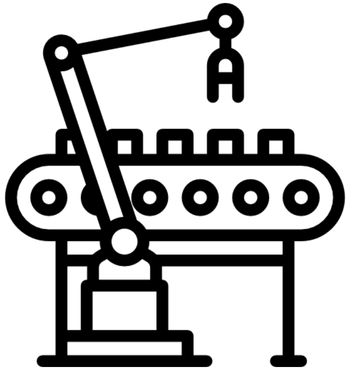
Results: Use Cases Taken from Variety of Sources



Management of Factory Robots

Metal Processing Site

Results: Use Cases Taken from Variety of Sources



Management of Factory Robots

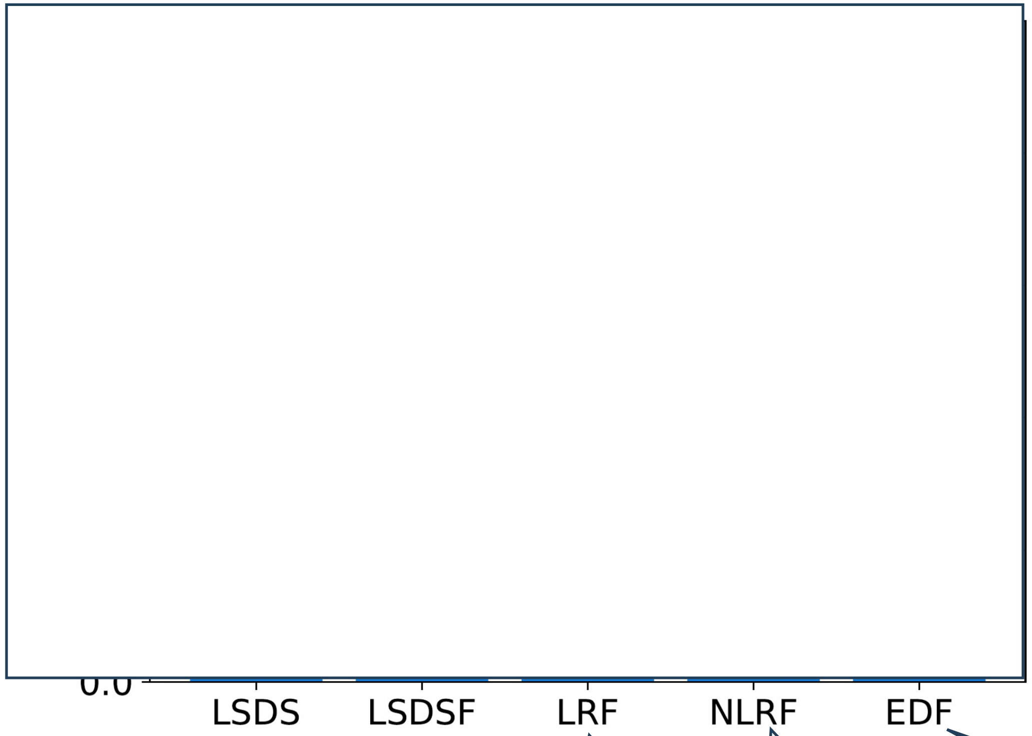
Metal Processing Site

Use cases cover a wide variety of deadlines, number of packets and profit

Simulation Settings

- Implementation: In C++ with code borrowed from ns3 (open-sourced)
- Channel Model: Both good and bad channels are considered
  - Using suitable modulation and coding scheme
- Time Horizon: 200ms
  - Any packets not scheduled within 200ms are assumed dropped
  - Leads to some loss of optimality; but we empirically observe it is very small

Results: Baselines



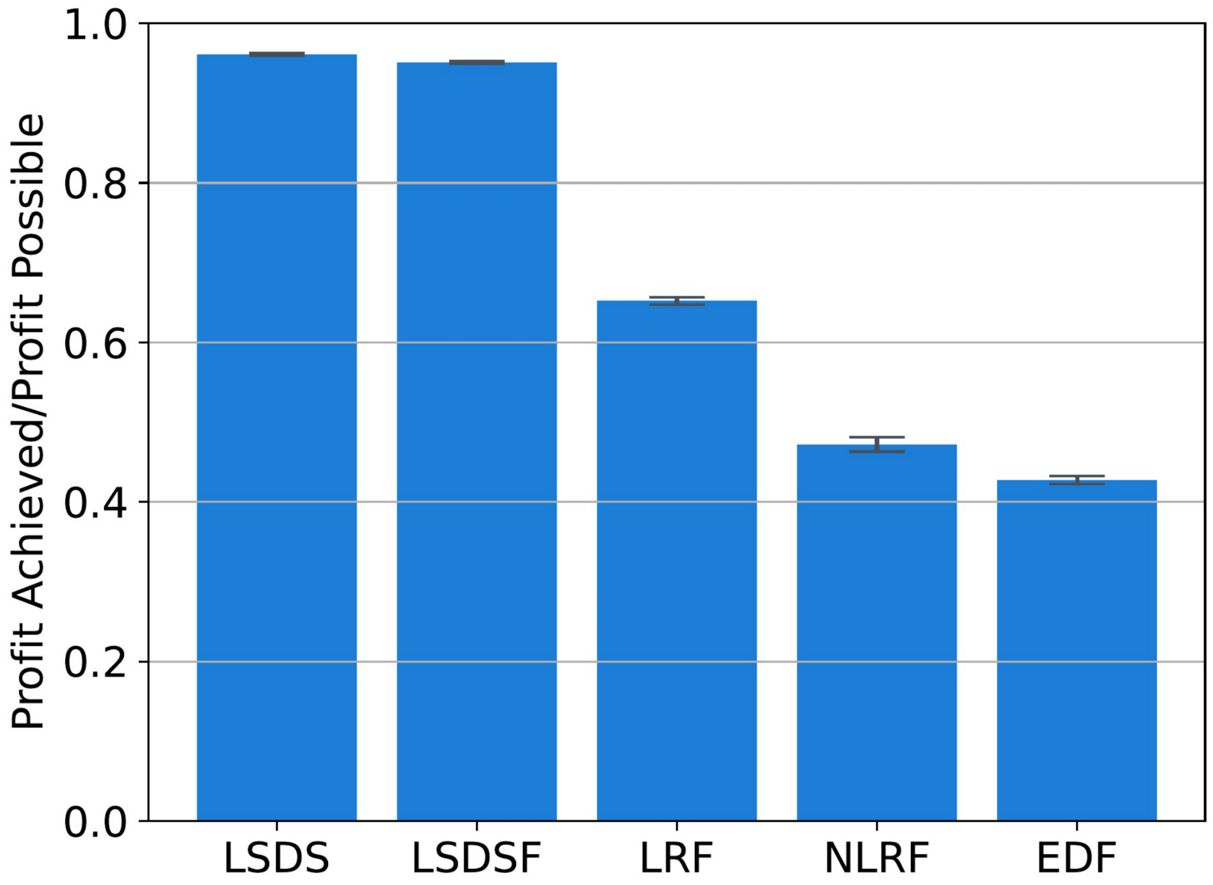
Largest Profit-To-Deadline Ratio First

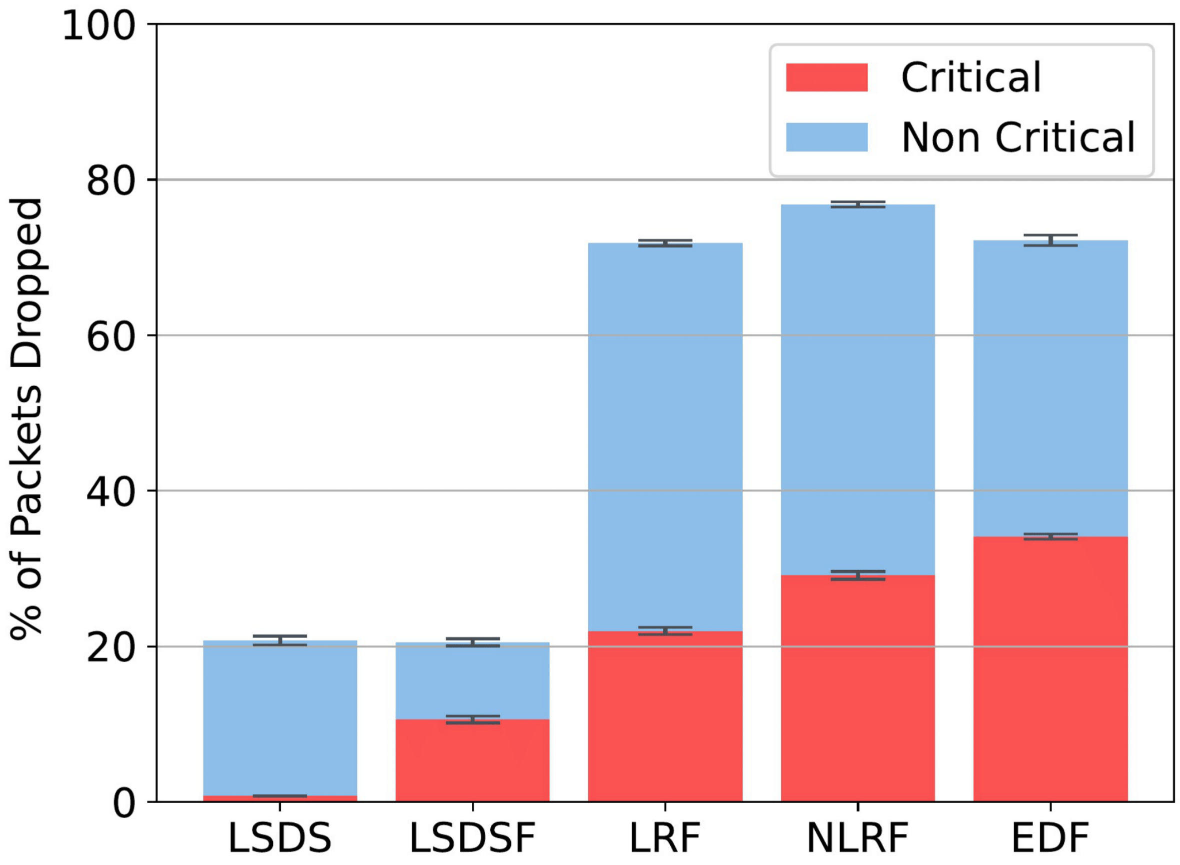
Non-starving version of LRF

Earliest Deadline First

Exhaustive search over RU configurations

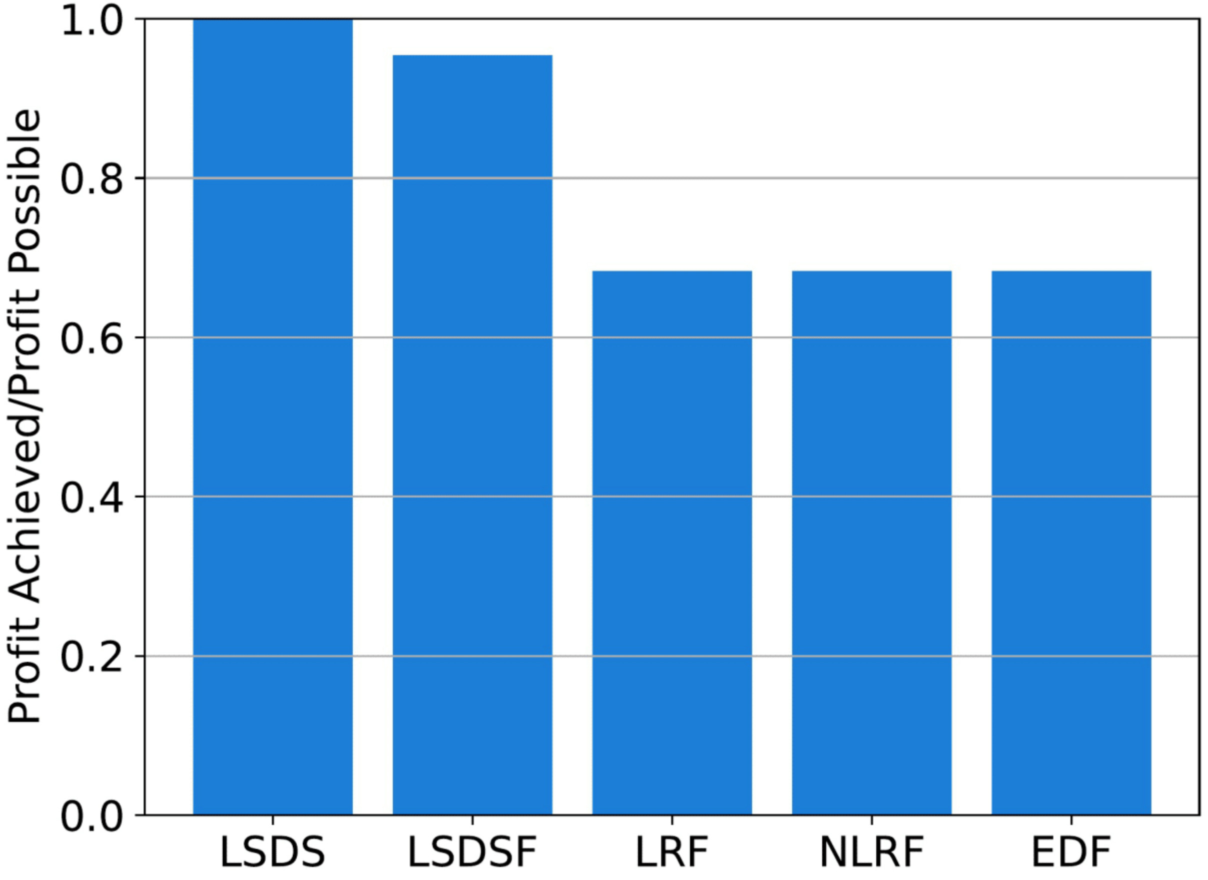
Scheduling of Packets in Industrial Robotic Control

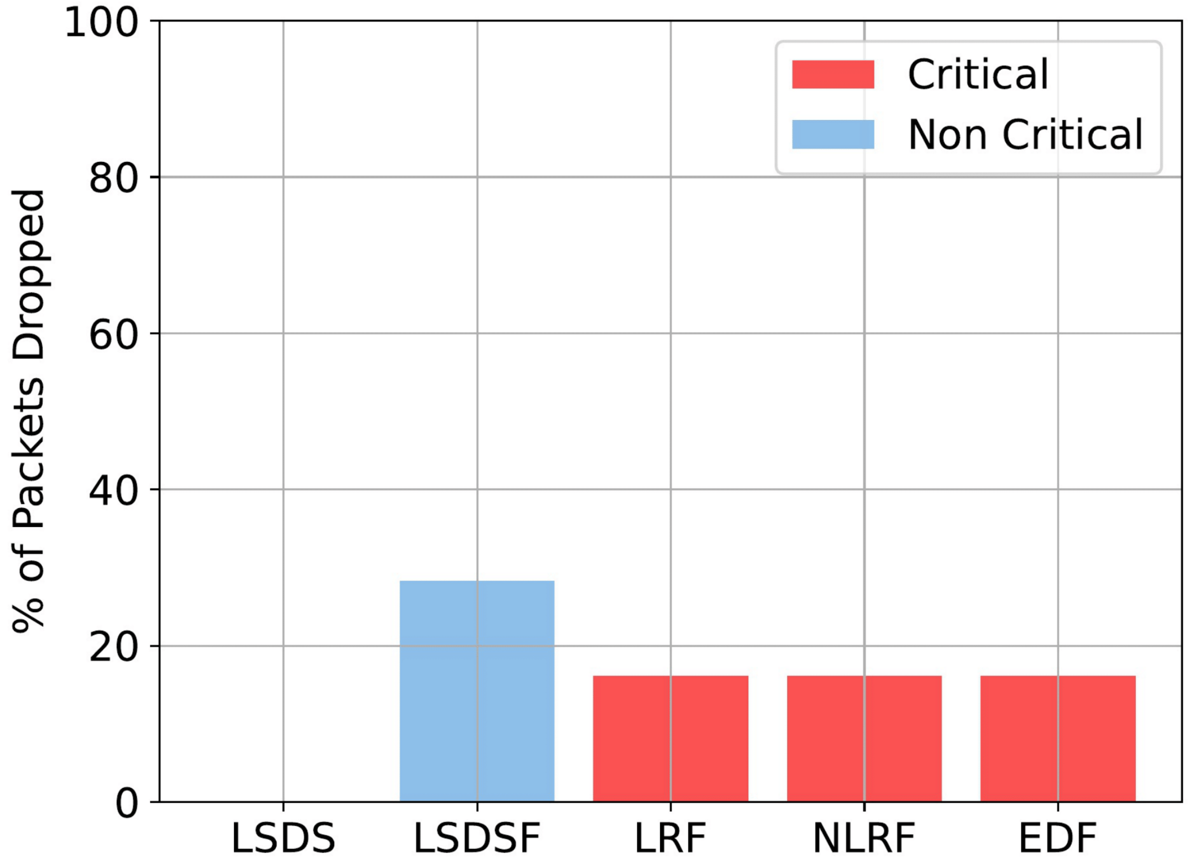




Both total drops and critical packet drops are far lower than baselines

Scheduling of Packets in Metal Processing





Both total drops and critical packet drops reduce to 0

Summary

- Smart factories require connectivity with specific requirements
- WiFi 6 can satisfy such requirements using specific techniques
- Scheduling packets in the above scenario is NP-Hard
- We propose a local-search based algorithm to schedule packets
- Our algorithm always provides profit greater 1/12 of the optimal

Summary

- Smart factories require connectivity with specific requirements
- WiFi 6 can satisfy such requirements using specific techniques
- Scheduling packets in the above scenario is NP-Hard
- We propose a local-search based algorithm to schedule packets
- Our algorithm always provides profit greater 1/12 of the optimal

Thank You

arani@iiitd.ac.in