A Stretching Algorithm for Parallel Real-time DAG Tasks on Multiprocessor Systems

Manar QAMHIEH Laurent George* Serge Midonnet

University Paris-Est Marne-la-Vallée, France
LIGM / ESIEE*

Versailles France RTNS 2014 8-10 Oct. 2014

Presentation Outline

- General Introduction
- Related Work & Problem Description
- DAG Stretching Algorithm
- Resource Augmentation Bound for global preemptive EDF
- Simulation-based Evaluations
- Conclusion and Perspective

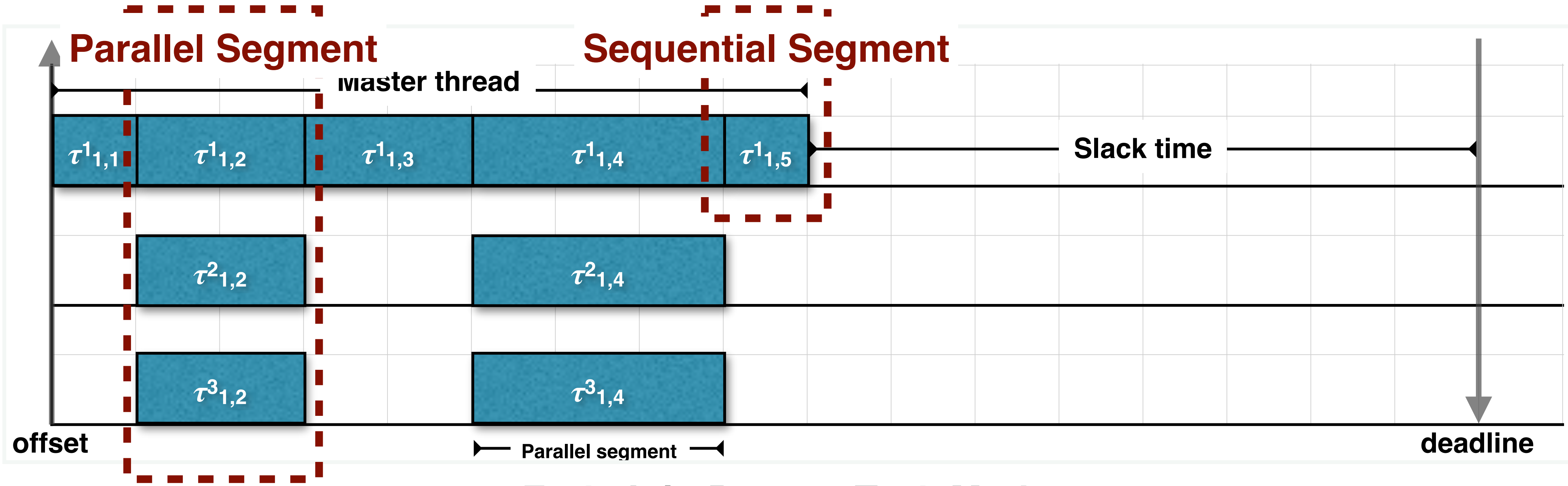
General Introduction

- **Hard Real-time Systems**
  - Failure in respecting timing characteristics of the real-time system leads to catastrophic results.
- **Multiprocessor Systems**
  - execution platform consists of homogeneous unit-speed processors.
- **Global Preemptive Scheduling**
  - the execution of a job can be interrupted by higher priority jobs, and job migration is allowed between processors.

Software Parallelism

- **Parallel programming** is proposed as an evolution of software so as to get advantage of multiprocessor architecture.
- **Parallel programming APIs**: OpenMP, pThreads
- Many **real-time parallel task models**: Fork-join (FJ) and Multi-Threaded Segment Model, Directed Acyclic Graphs (DAG).

Related Work: FJ-Stretching Algo.

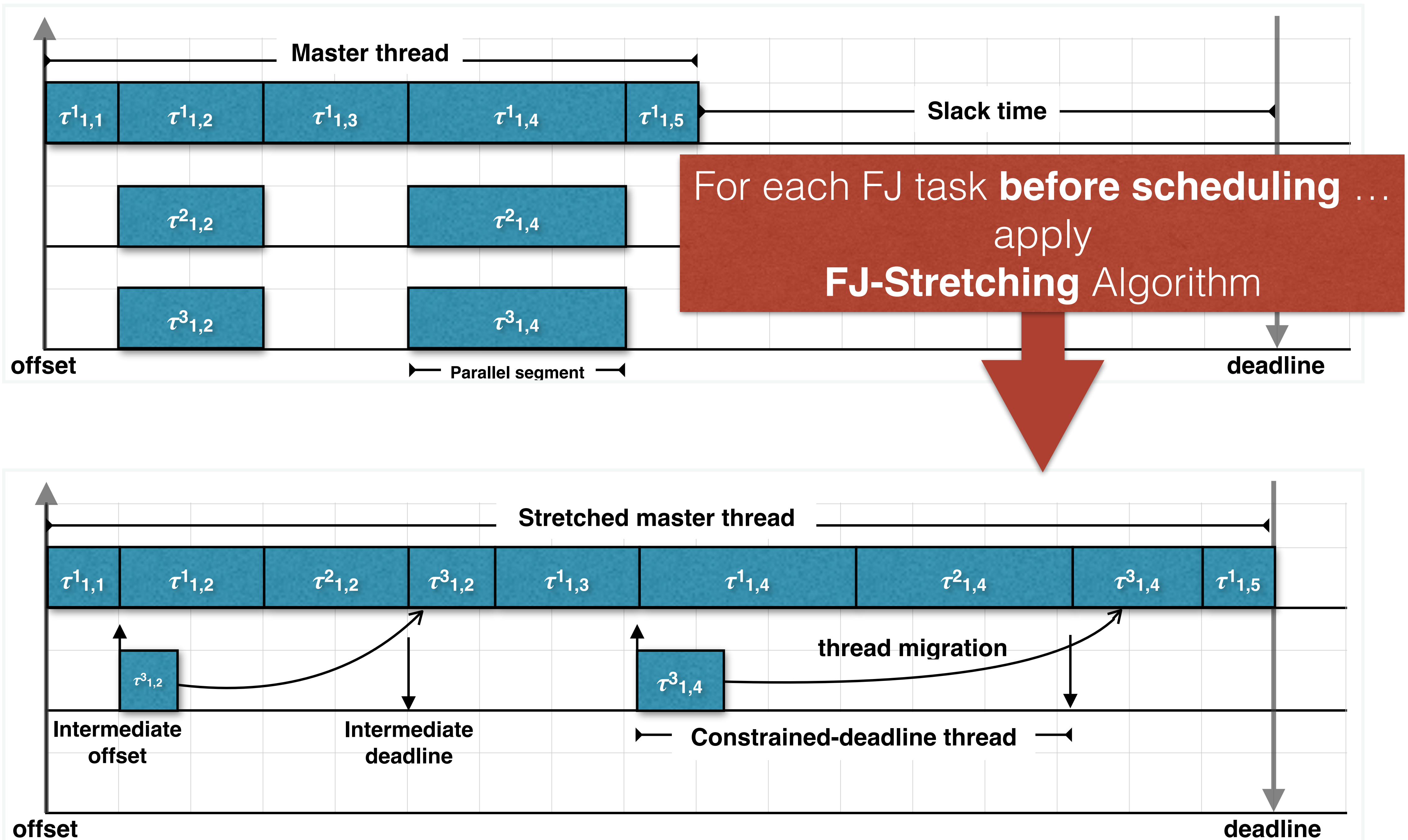


Fork-Join Parallel Task Model

Fork-join task *sets on multiprocessor systems can have schedulable utilization bounds slightly greater than and arbitrarily* ***close to uniprocessor schedulable utilization bounds****. From the perspective of schedulability, it is therefore desirable to* ***avoid such task structures*** *as much as possible.*[1]

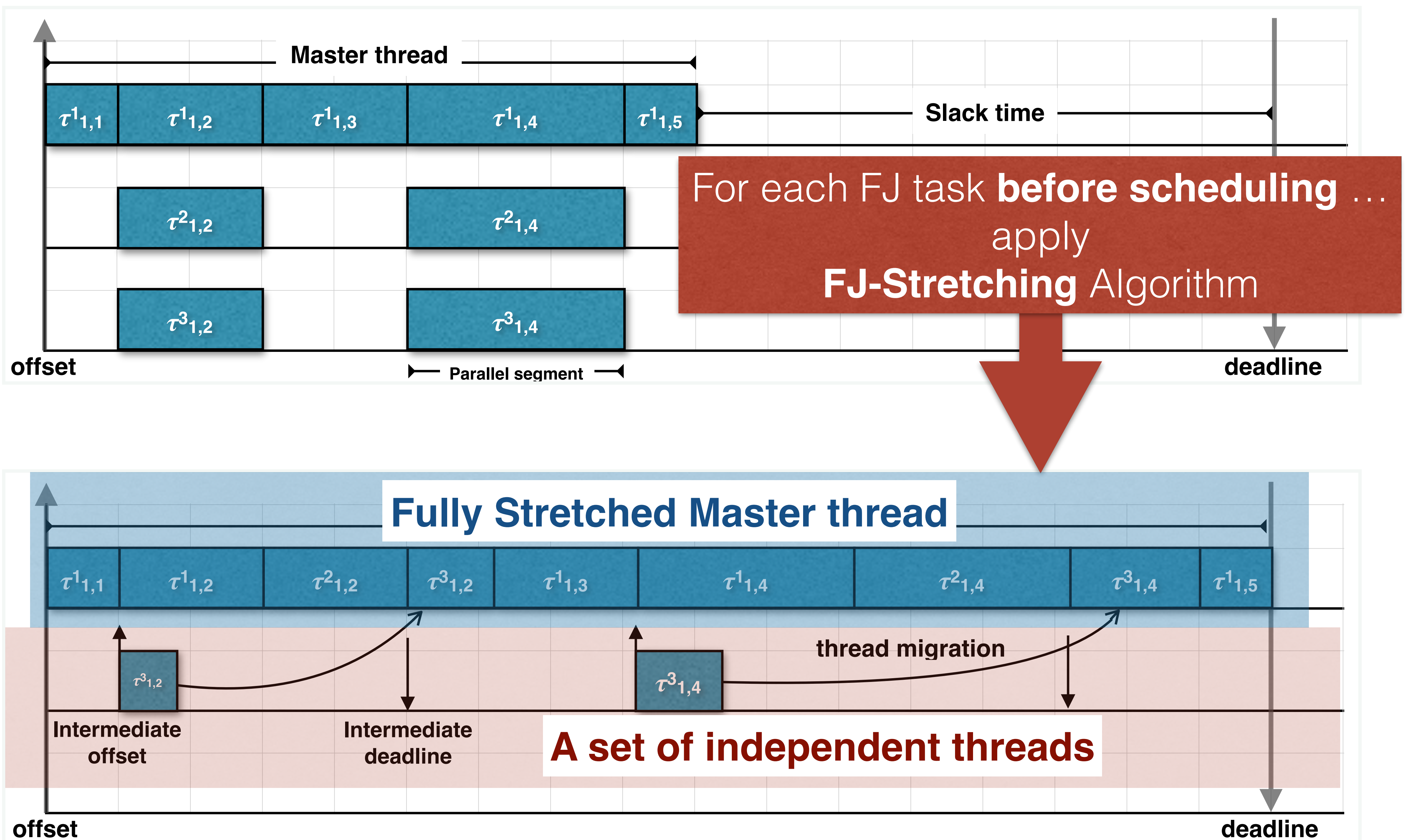
[1] *"Scheduling Parallel Real-Time Tasks on Multi-core Processors"*. Lakshmanan et al., ECRTS 2010

Related Work: FJ-Stretching Algo.



[1] *"Scheduling Parallel Real-Time Tasks on Multi-core Processors"*. Lakshmanan et al., ECRTS 2010

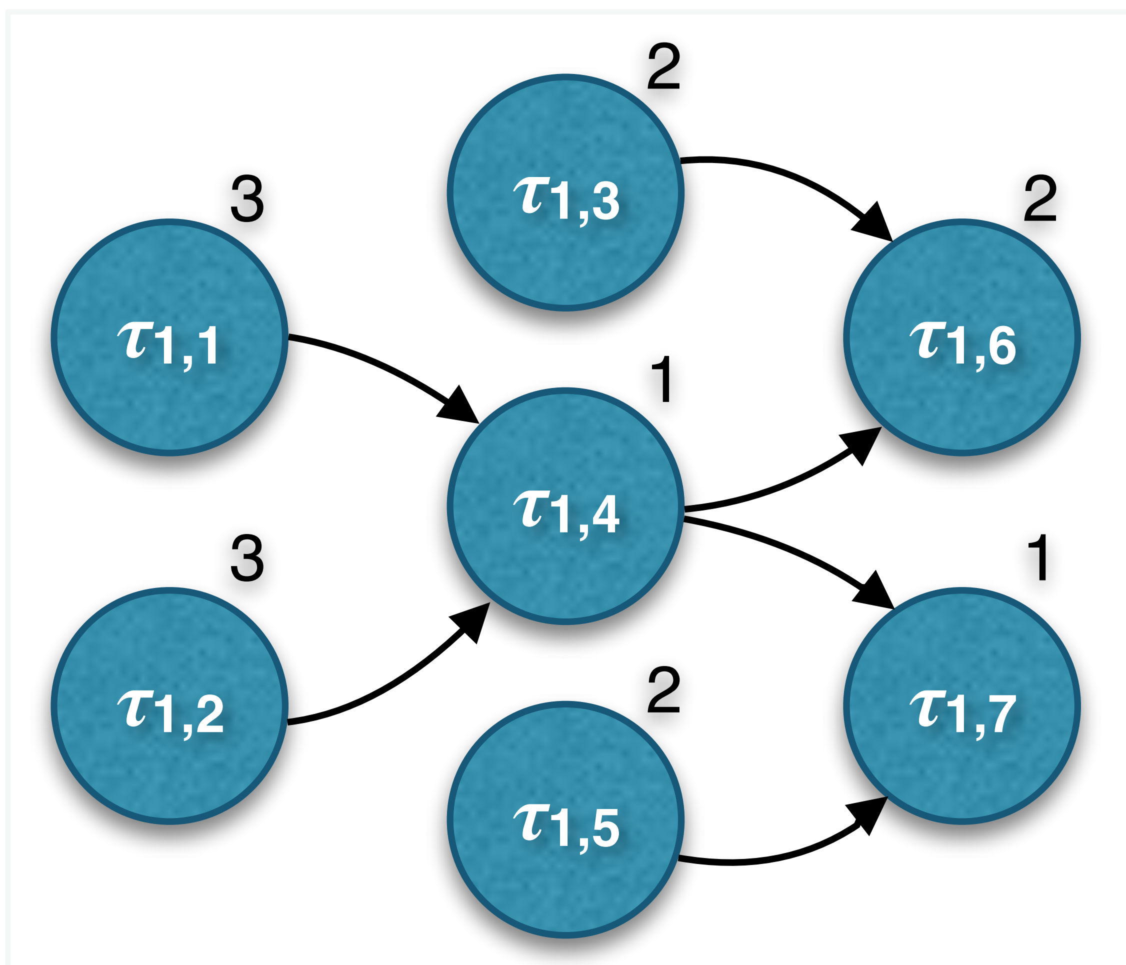
Related Work: FJ-Stretching Algo.



[1] *"Scheduling Parallel Real-Time Tasks on Multi-core Processors"*. Lakshmanan et al., ECRTS 2010

Contribution: DAG-Str Algorithm

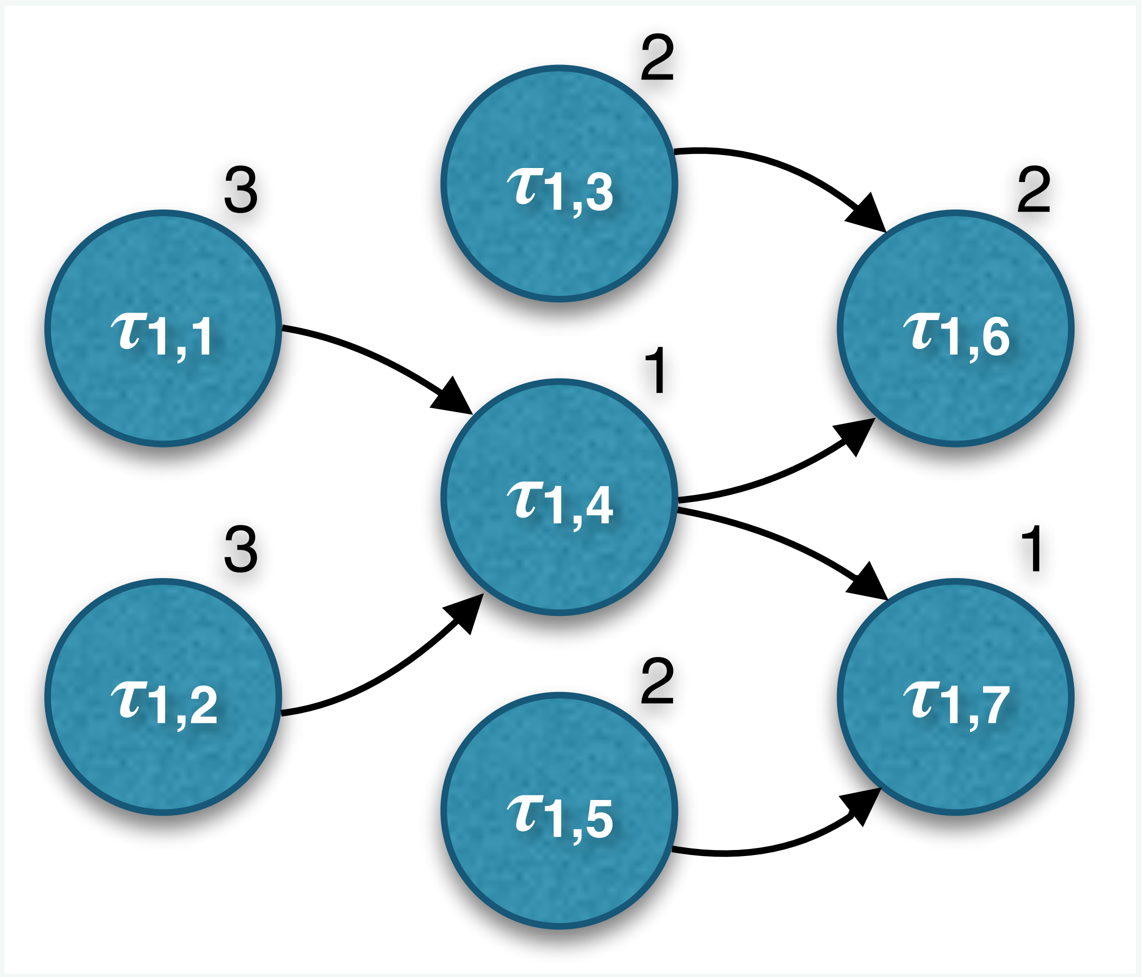
We propose a **DAG-Str (DAG Stretching) Algorithm**



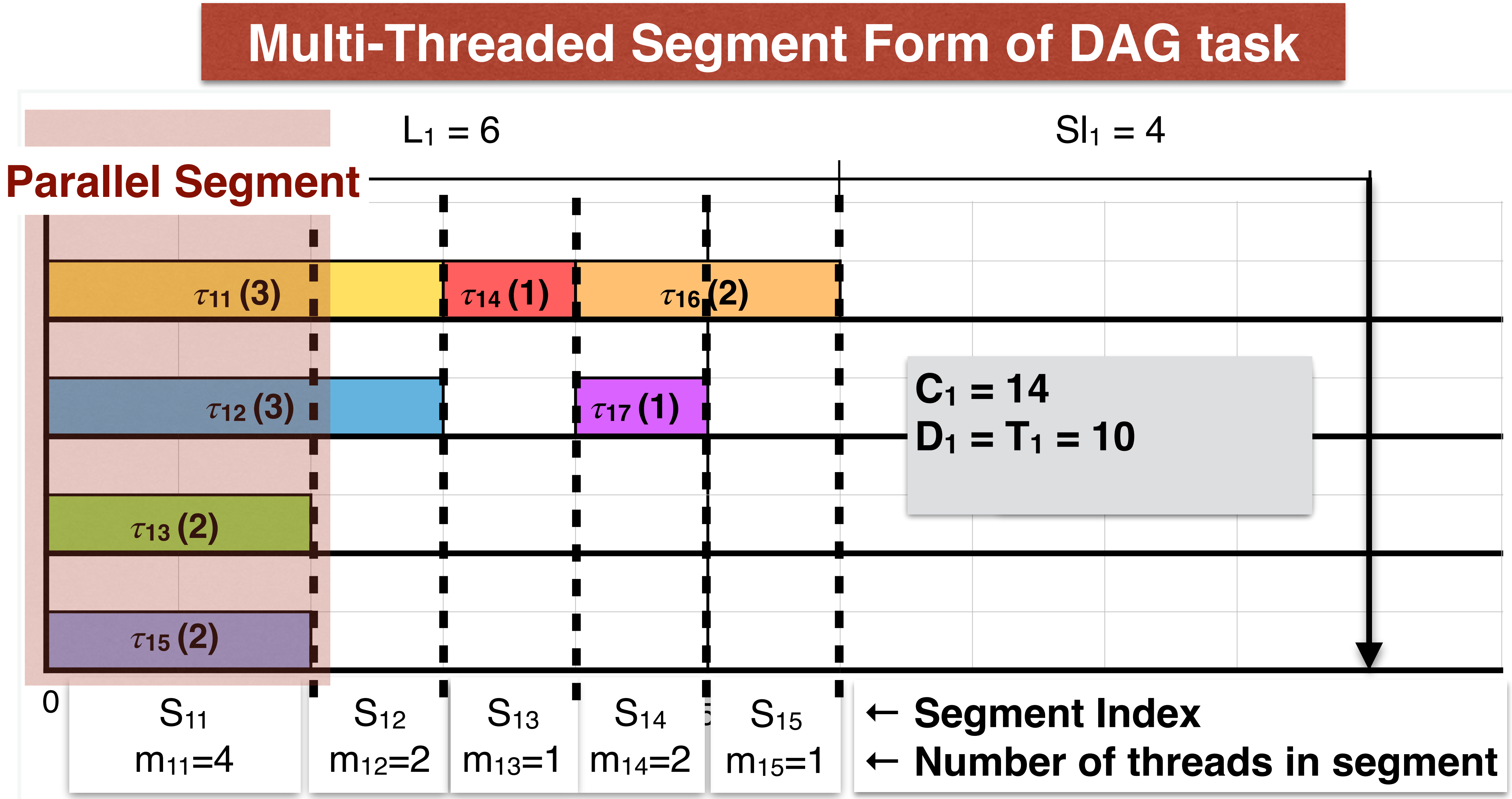
Contribution: DAG-Str Algorithm

We propose a **DAG-Str (DAG Stretching) Algorithm**

- A DAG task (periodic, implicit-deadline):
    - Set of subtasks with specific WCETs
    - Set of precedence constraints between subtasks
    - timing parameters: offset, relative deadline, period

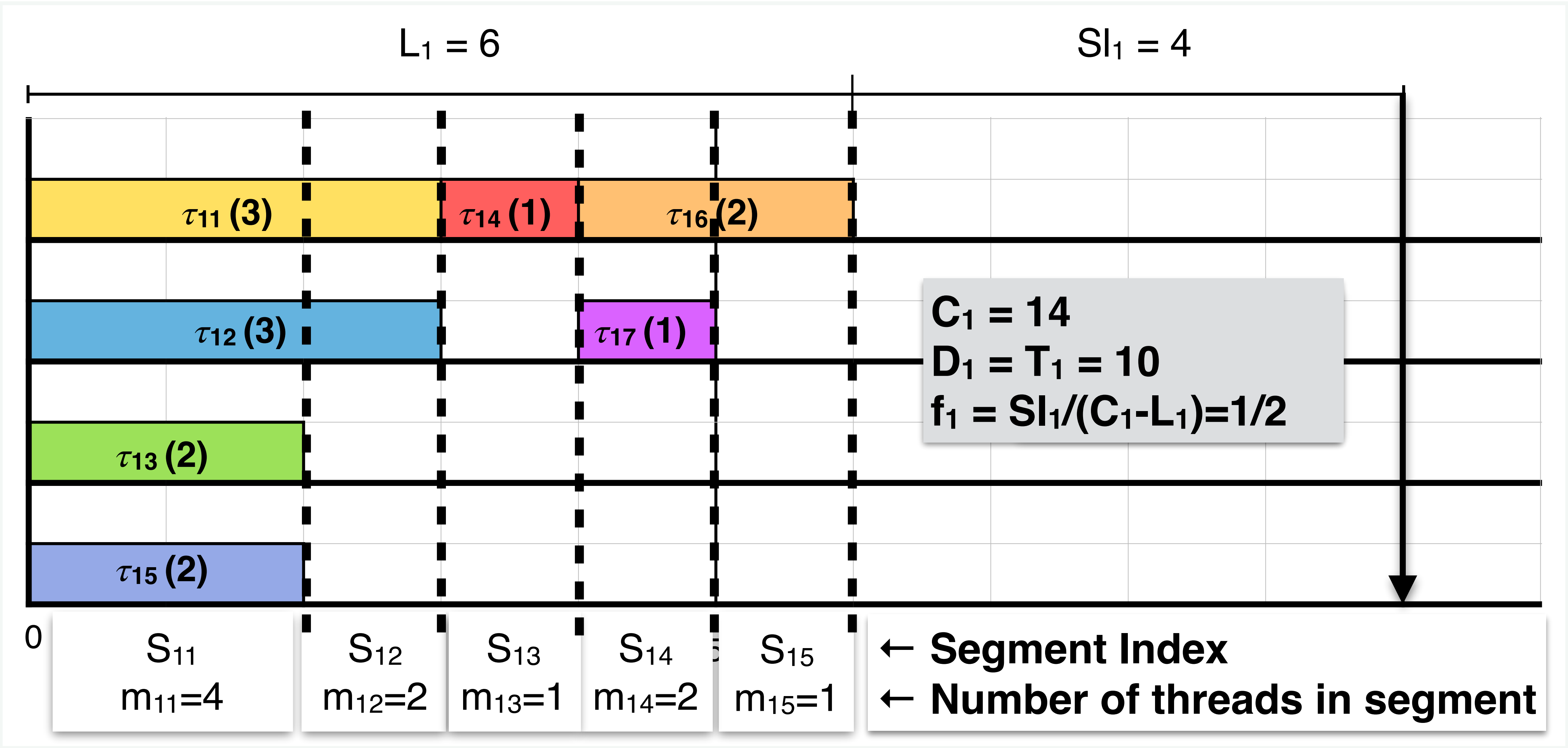


DAG-Str Algorithm



DAG-Str Algorithm

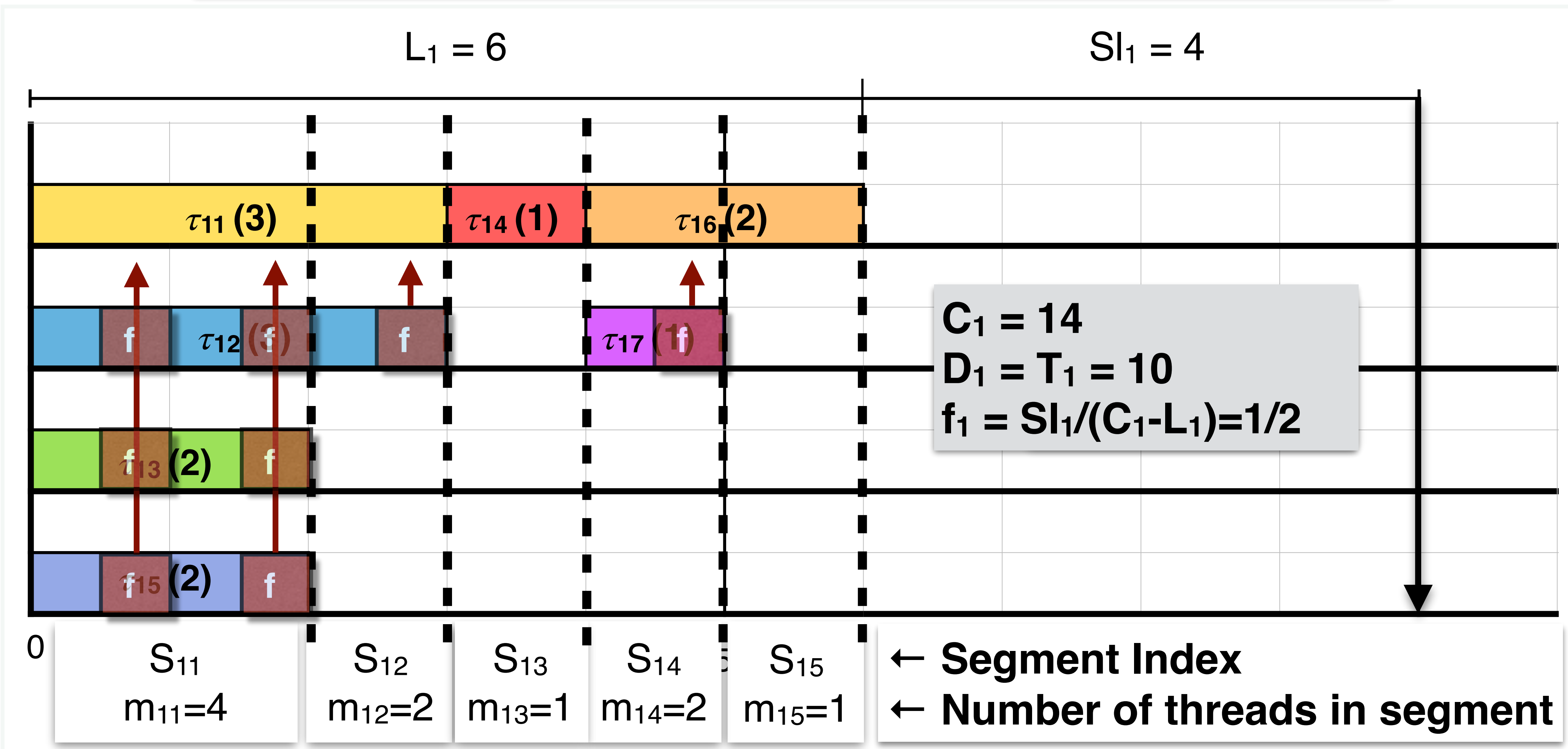
Calculate a unit factor f_1 of the DAG task



Uniform slack filling on a Thread-level

DAG-Str Algorithm

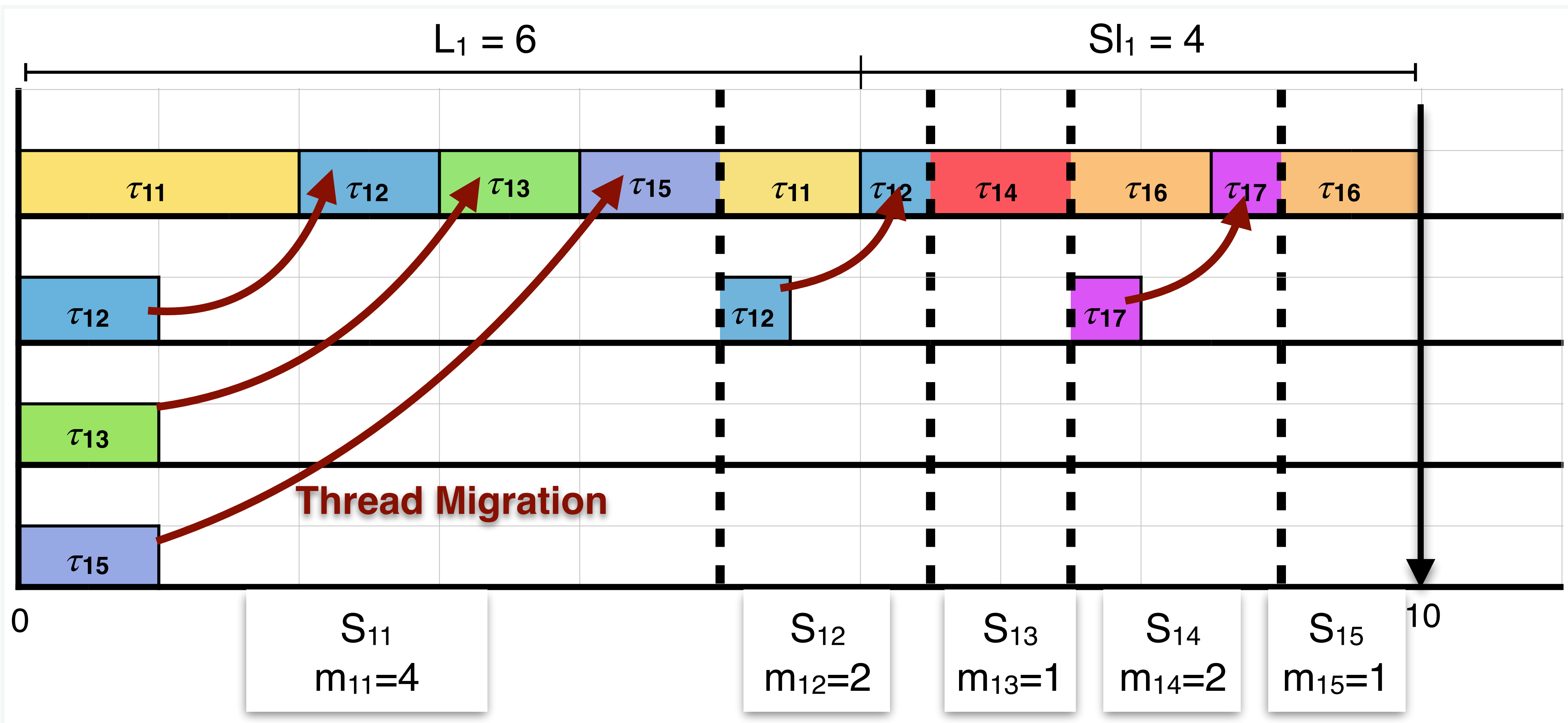
Calculate a unit factor f_1 of the DAG task



Uniform slack filling on a Thread-level

DAG-Str Algorithm

Calculate a unit factor f_1 of the DAG task

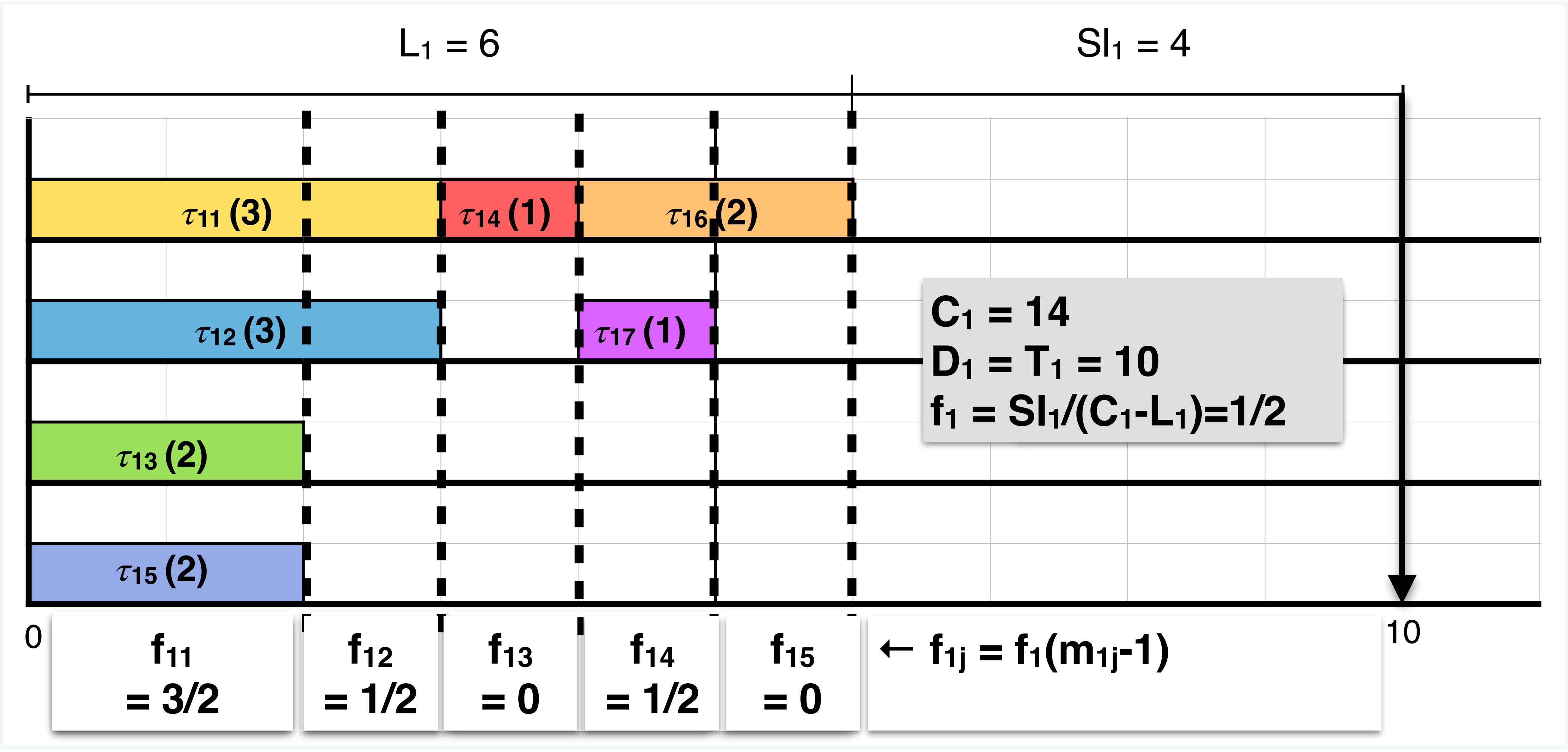


A lot of thread **migration** and **preemptions** are caused.

Solution: fill slack time based on a **Segment-Level**.

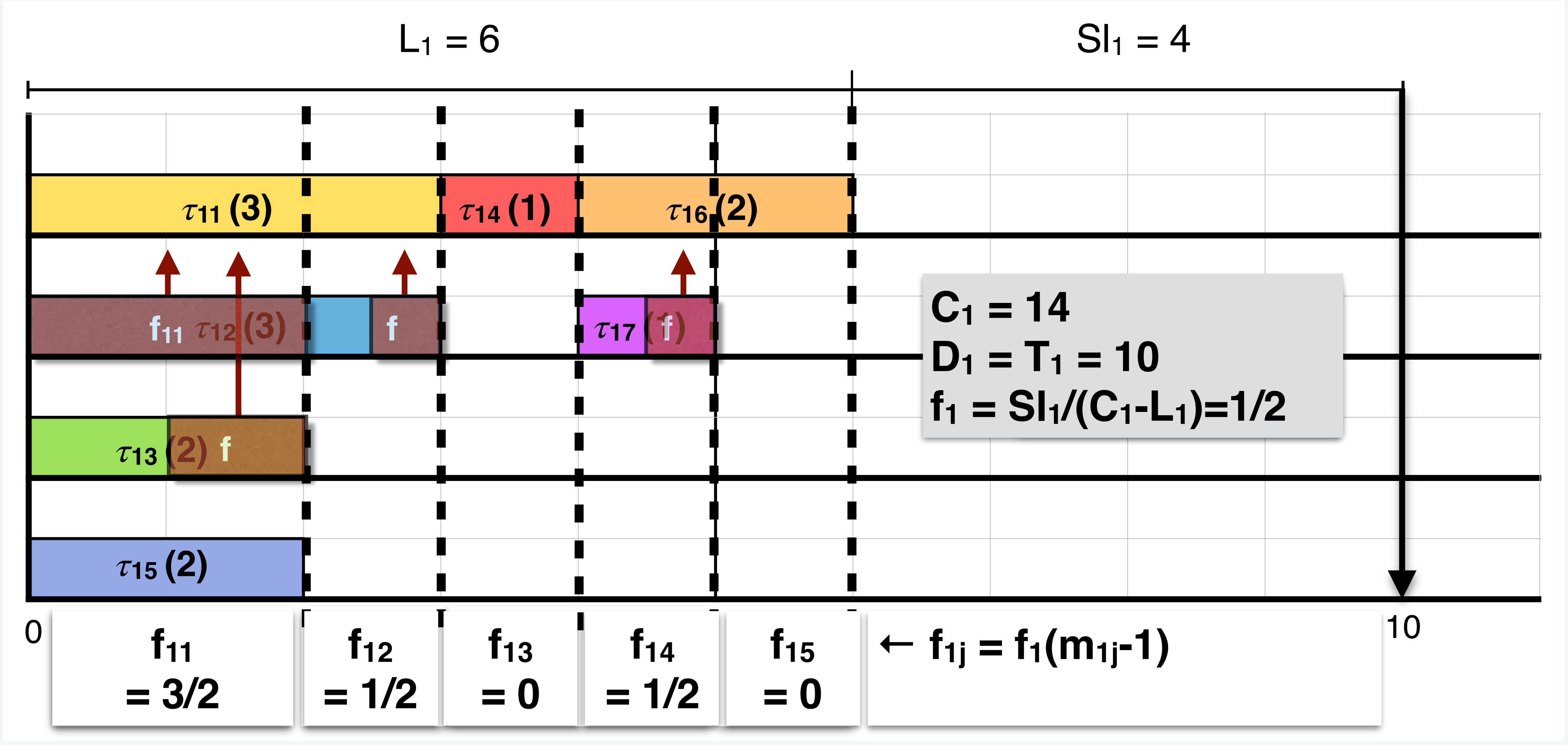
DAG-Str Algorithm

Calculate a segment factor f_{1j} of the DAG task



DAG-Str Algorithm

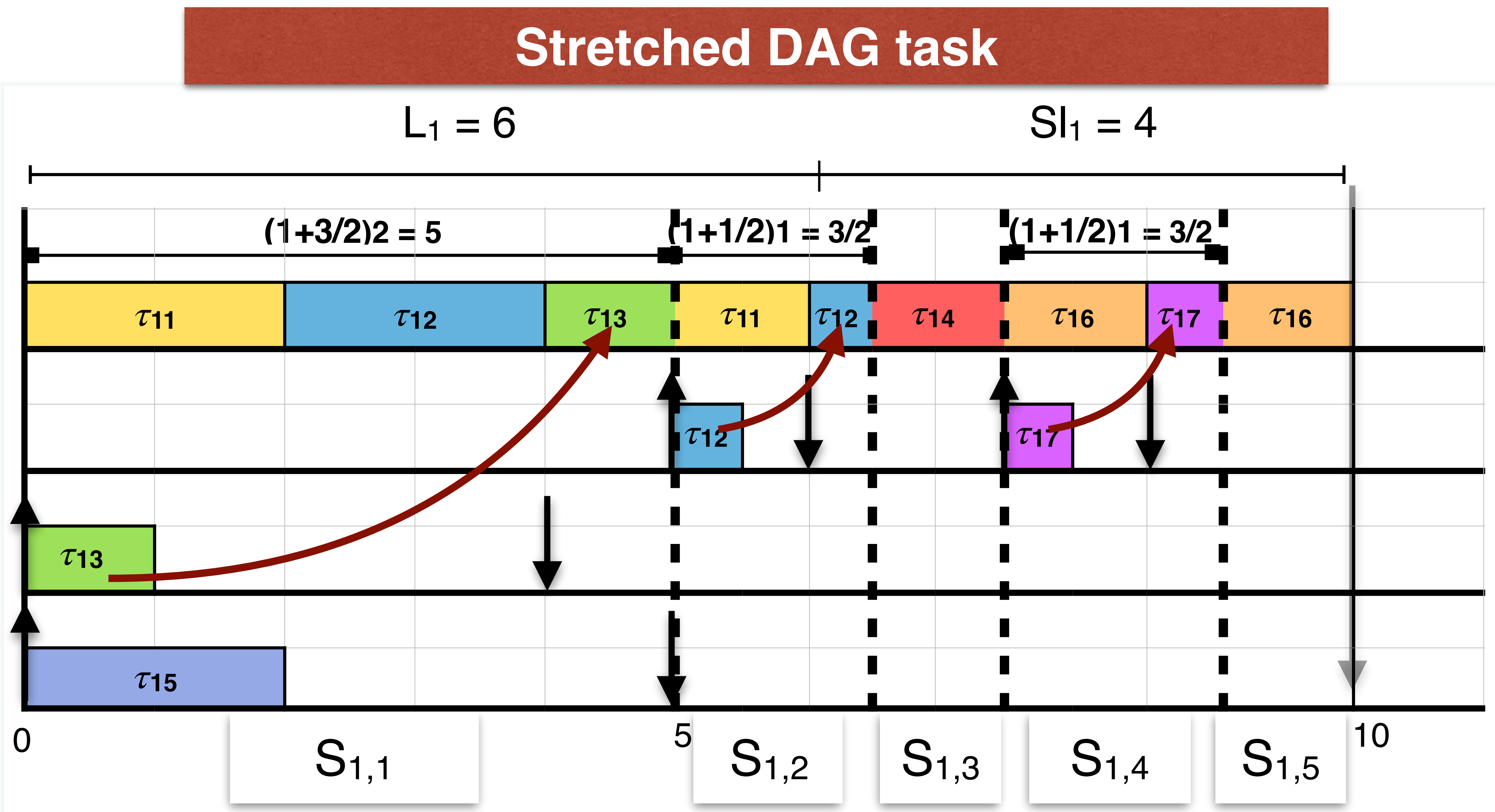
Calculate a segment factor f_{1j} of the DAG task



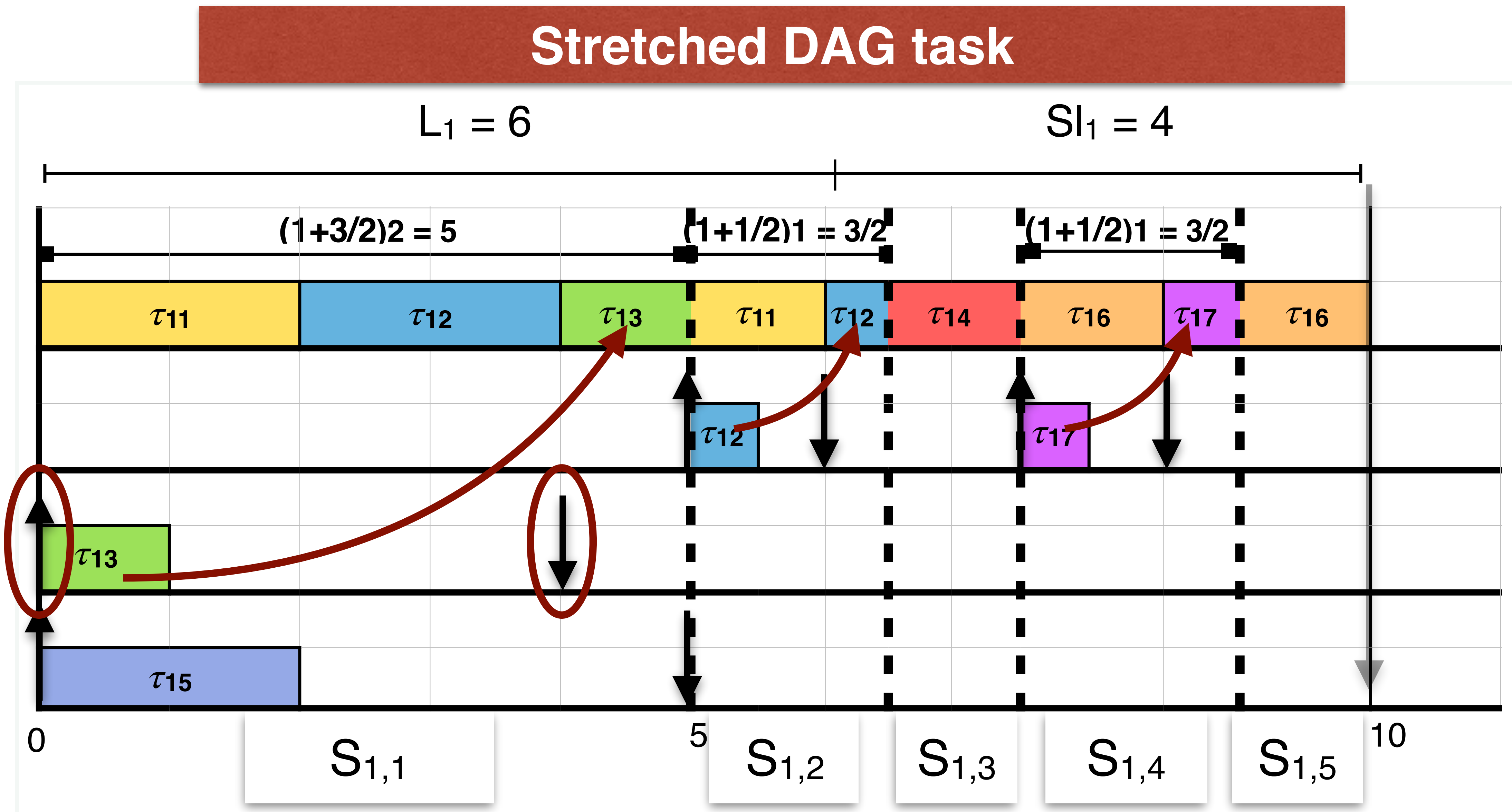
(f_{1j} * c_{1j}) execution units added to **master thread** from each segment S_{1j}

One thread/segment migrates between processors

DAG-Str Algorithm



DAG-Str Algorithm



Intermediate **offsets** and **deadlines** are assignes to parallel **threads**

DAG-Str Algorithm

- Results of DAG-Str algorithm when applied on a DAG task whose:
  - Utilization > 1:
    - Fully-Stretched threads with utilization equal to 1
      → Assigned to dedicated processors.
    - Periodic independent constrained-deadline threads, with intermediate offset and relative deadline.
      → Scheduled using any multiprocessor scheduling algorithm.
  - Utilization == 1 → Fully-stretched thread which is assigned to a dedicated processor.
  - Utilization < 1 → Periodic independent implicit-deadline thread scheduled using any multiprocessor algorithm.
- We consider Global Preemptive Earliest-Deadline First (GEDF) Scheduling algorithm.

Resource Augmentation Bound for GEDF

We prove a resource Augmentation Bound of global preemptive EDF equal to $(3+\sqrt{5})/2$

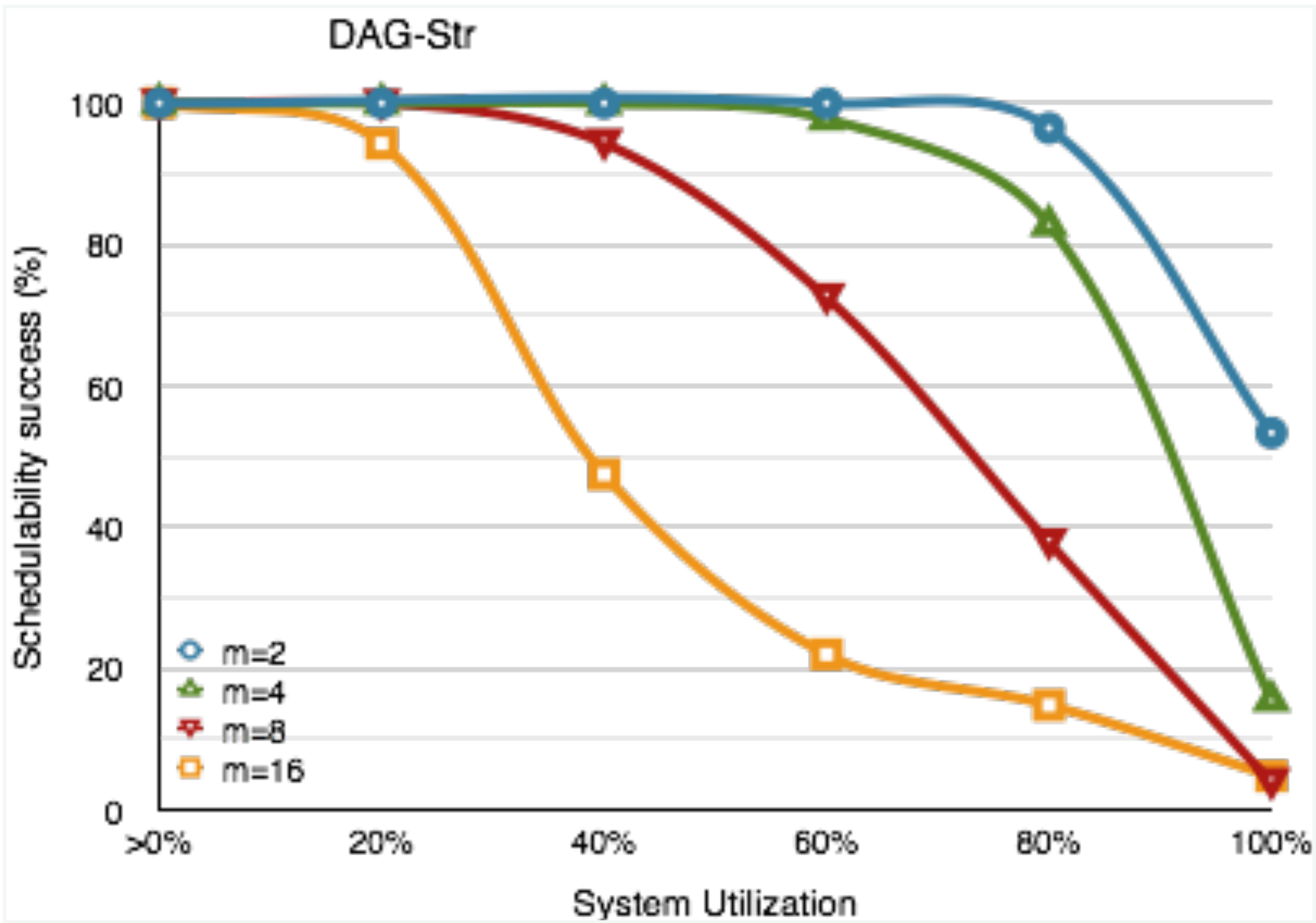
for all task sets with $n < \varphi \times \overline{m}$, where:

n	number of tasks in the set
φ	golden ratio where $\varphi = (1+\sqrt{5})/2$
$\overline{m}$	number of processors after stretching

The same value was proved for Global EDF for DAG tasks for large m in [1].

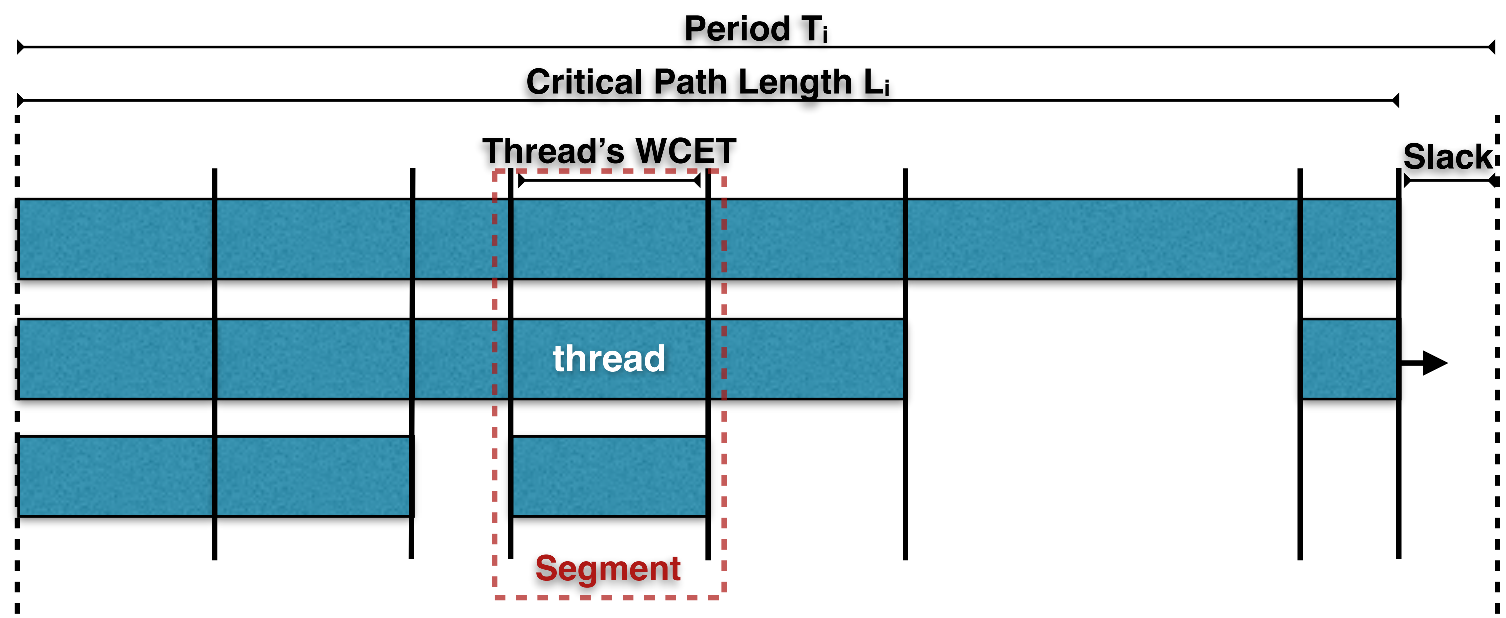
[1] *"Analysis of Federated and Global Scheduling for Parallel Real-time Tasks"*. Li *et al.*, ECRTS 2014

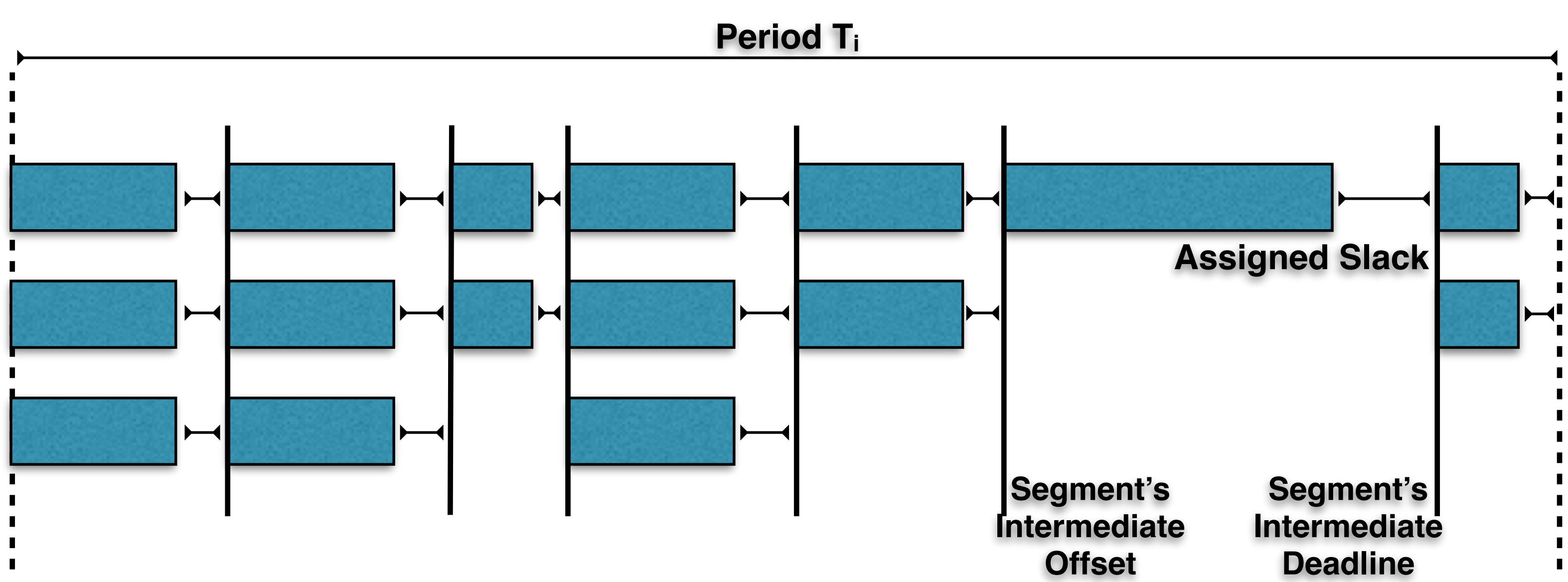
Simulation Results



Simulation Results

Compare with **Decomposition Algorithm (DCMP)[2]** for DAG tasks.

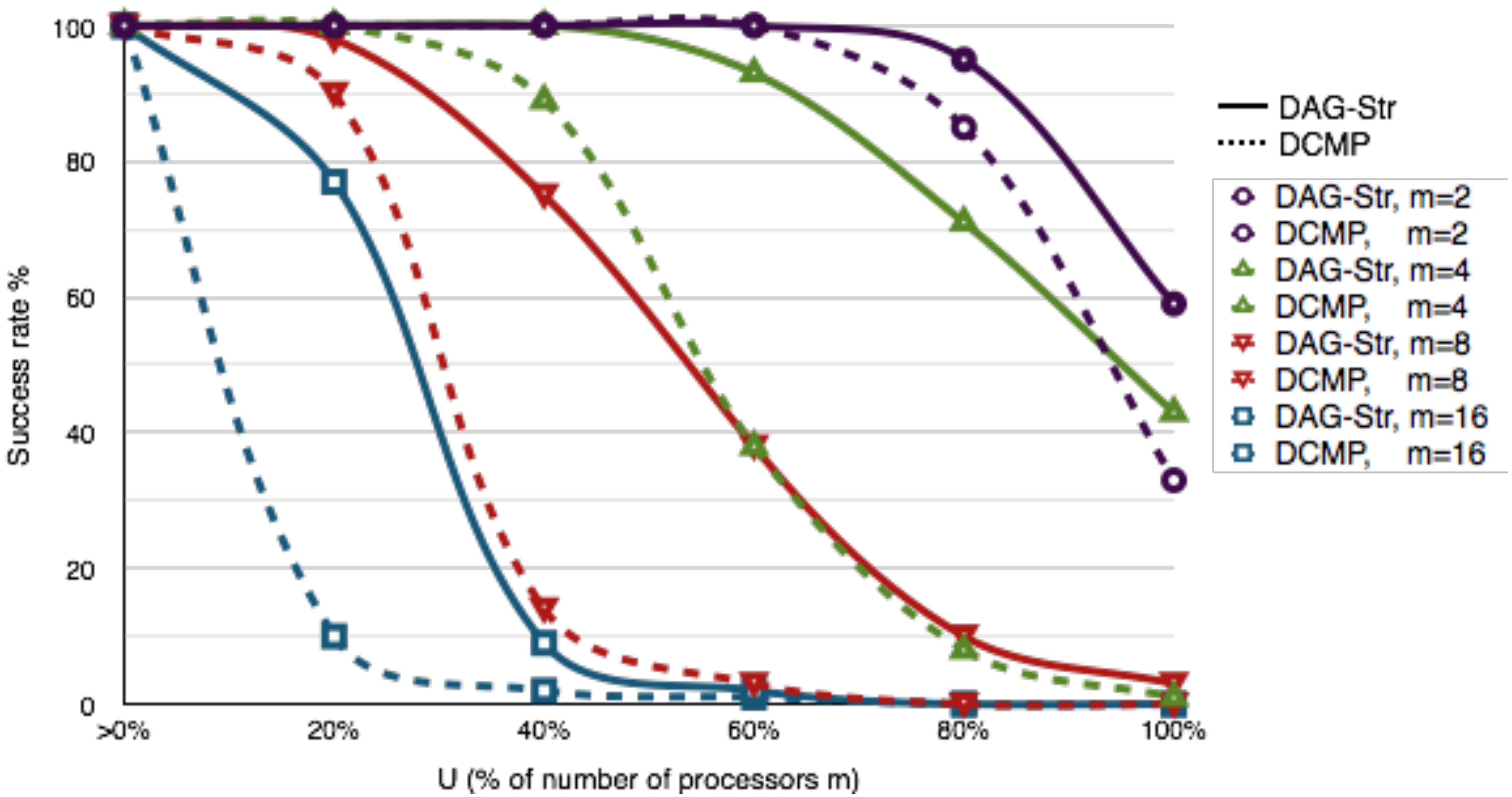




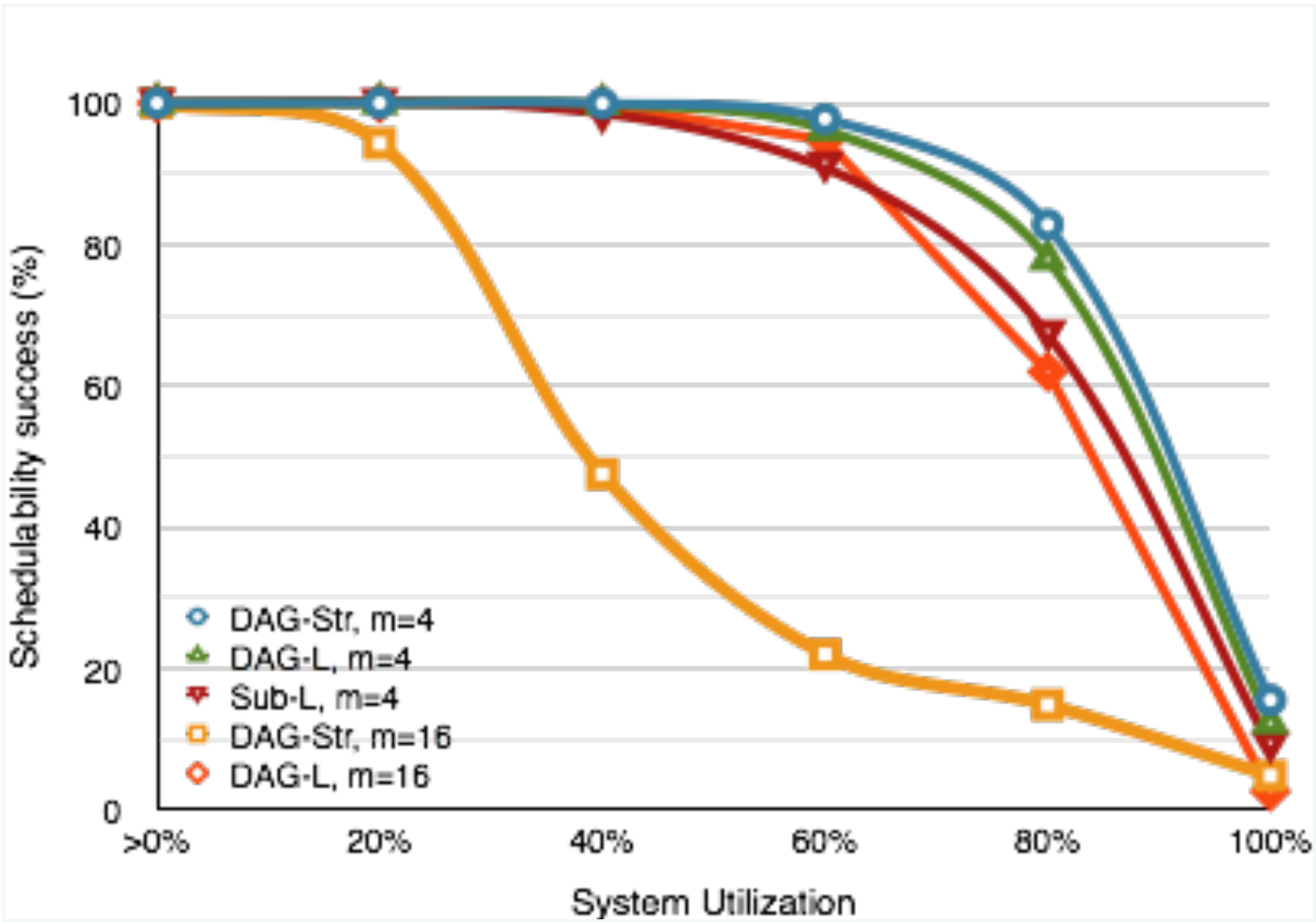
Assigns slack time for segments based on a density threshold

[2] *"Parallel real-time scheduling of DAGs"*. Saifullah et al., IEEE Transactions on Parallel and Distributed Systems 2014

Simulation Results



Simulation Results



Conclusion & Perspectives

- **Conclusions**:
  - DAG-Str algorithm which is more general than FJ Stretching algorithm
  - Resource Augmentation Bound for GEDF scheduling algorithm.
  - Simulation-based evaluations of the stretching algorithm.
- **Perspectives**
  - Optimize DAG stretching algorithm to reduce forced migrations and preemptions.
  - Analyze Fixed Priority Scheduling Algorithm.

Thank You
Questions?