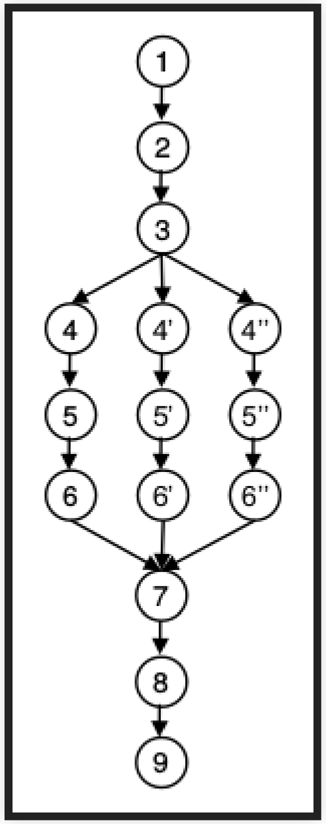
ENHANCING SCHEDULING ROBUSTNESS WITH TASK COMPLETION FEEDBACK AND RESOURCE REQUIREMENT BIASING

Nicolas Grounds, Ph.D., John K. Antonio, Ph.D.,
University of Oklahoma

OVERVIEW

- Problem Introduction & Past Research Summary
- Feedback and Biasing Concepts
- Simulation Results Overview & Conclusions

PROBLEM INTRODUCTION

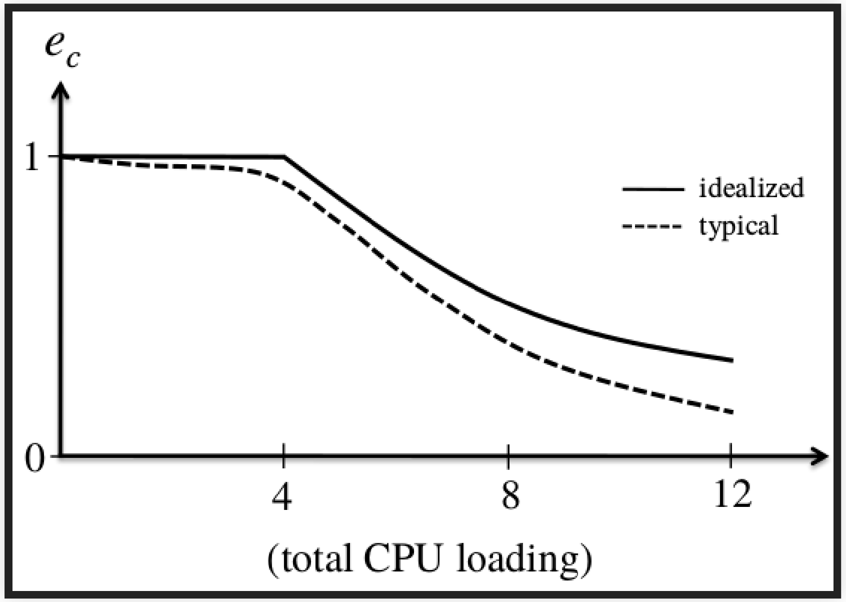


Scheduling Workflow Tasks in Cluster of Multi-core, Memory-Managed Machines (2009) [1]

Tasks are Individually-Schedulable Units of a Workflow

Workflows are a Precedence-Oriented DAG of Tasks Representing a "Job" Submitted to the System by a User along with Deadline

PROBLEM INTRODUCTION



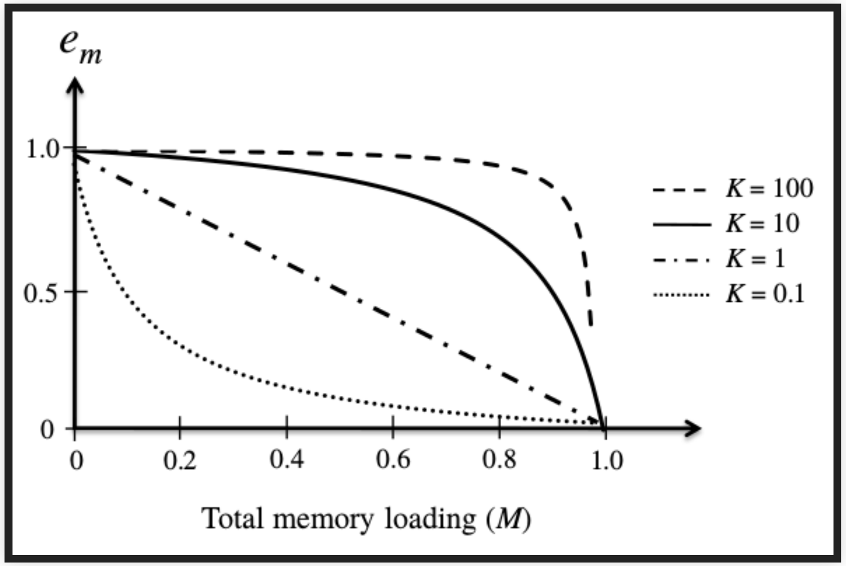
Tasks defined as utilization of two resources provided by a Machine:

1. CPU Utilization

Efficiency (Productivity) of a Machine based on cumulative tasks' CPU utilization, C, and Machine CPU core count (4)

$$e_c = \max\{1, \frac{4}{C}\}$$

PROBLEM INTRODUCTION



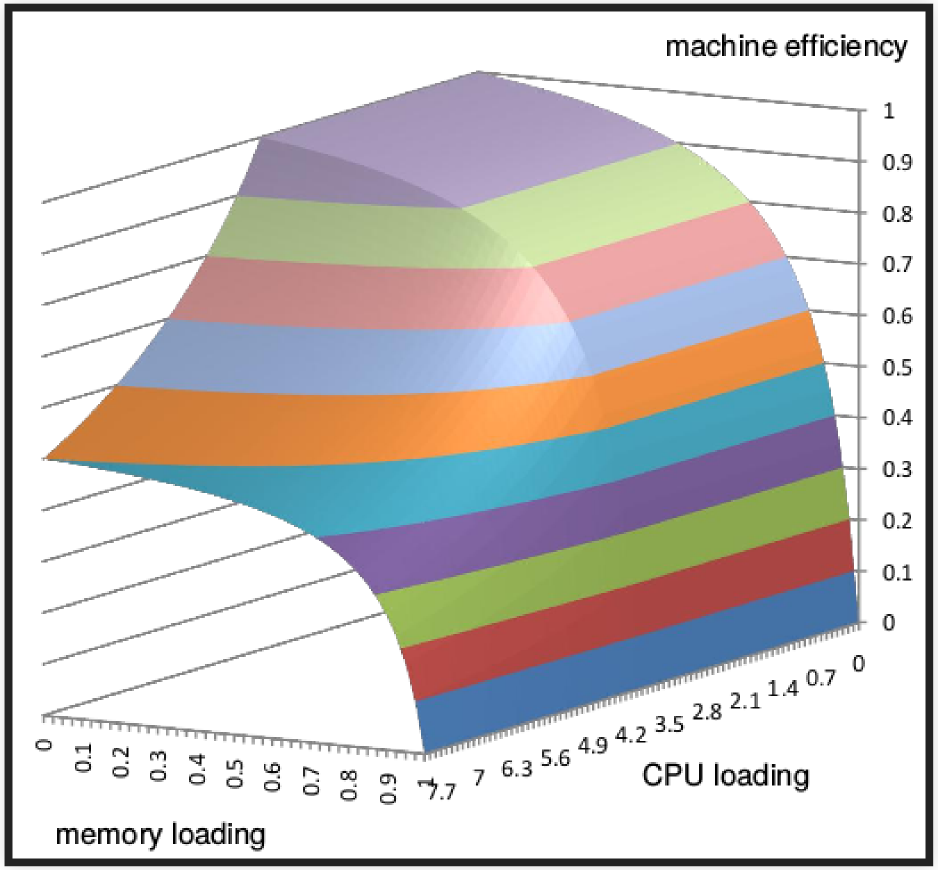
Tasks' second resource utilization:

2. Memory Utilization

Efficiency (Productivity) of a Machine based on cumulative tasks' Memory utilization:

$$e_m = \frac{K}{K + \frac{1}{\frac{1}{M} - 1}}$$

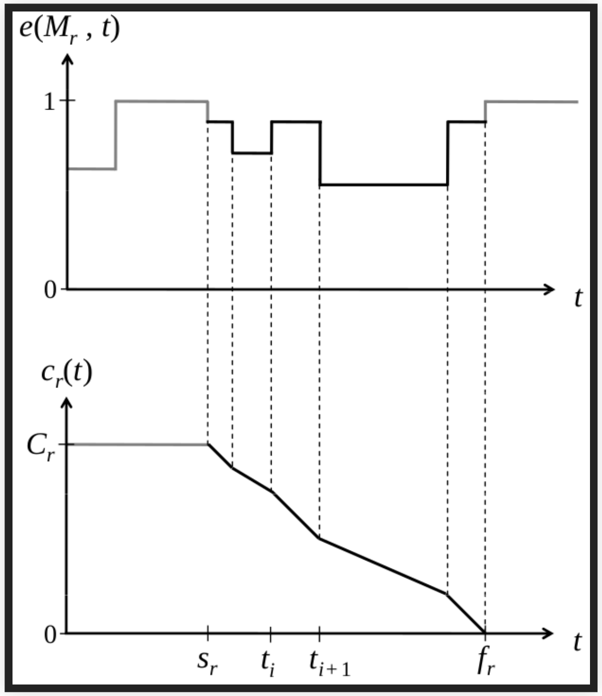
PROBLEM INTRODUCTION



Machines' Efficiency/Productivity based on CPU and Memory loading of executing tasks

$$e = e_c e_m$$

PROBLEM INTRODUCTION

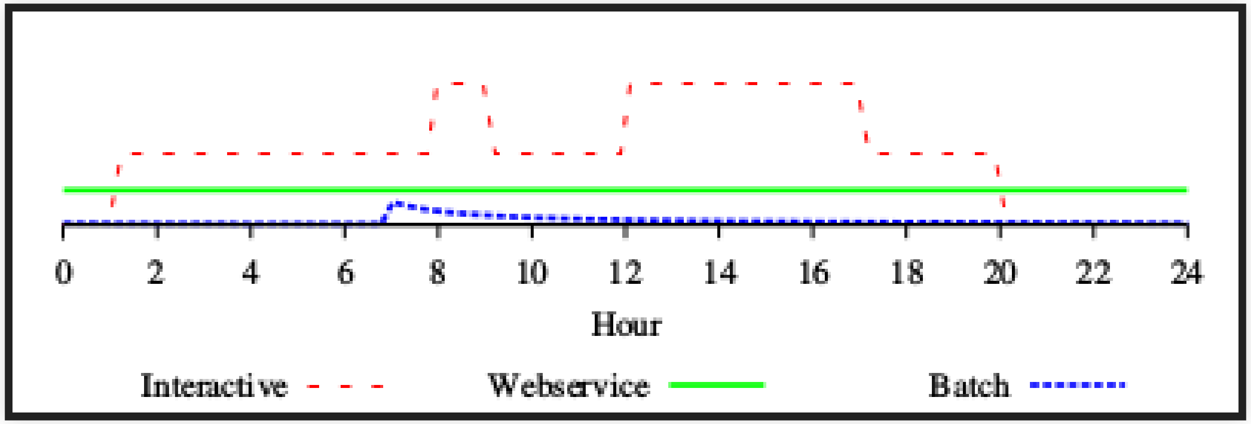


Task also defined by an amount of "work" that must be done (i.e. CPU cycles). The executing machine's efficiency/productivity impacts the rate of the amount of work done over time.

PROBLEM INTRODUCTION

Three types of Workflows modeled (after 3-region small interactive jobs, semi-interactive medium jobs, and batch-oriented large jobs), differentiated by:

- Amount of work required by its Tasks
- Number of tasks (total and concurrent)
- Arrival Rate



PROBLEM INTRODUCTION

Three Task-Scheduling Algorithms:

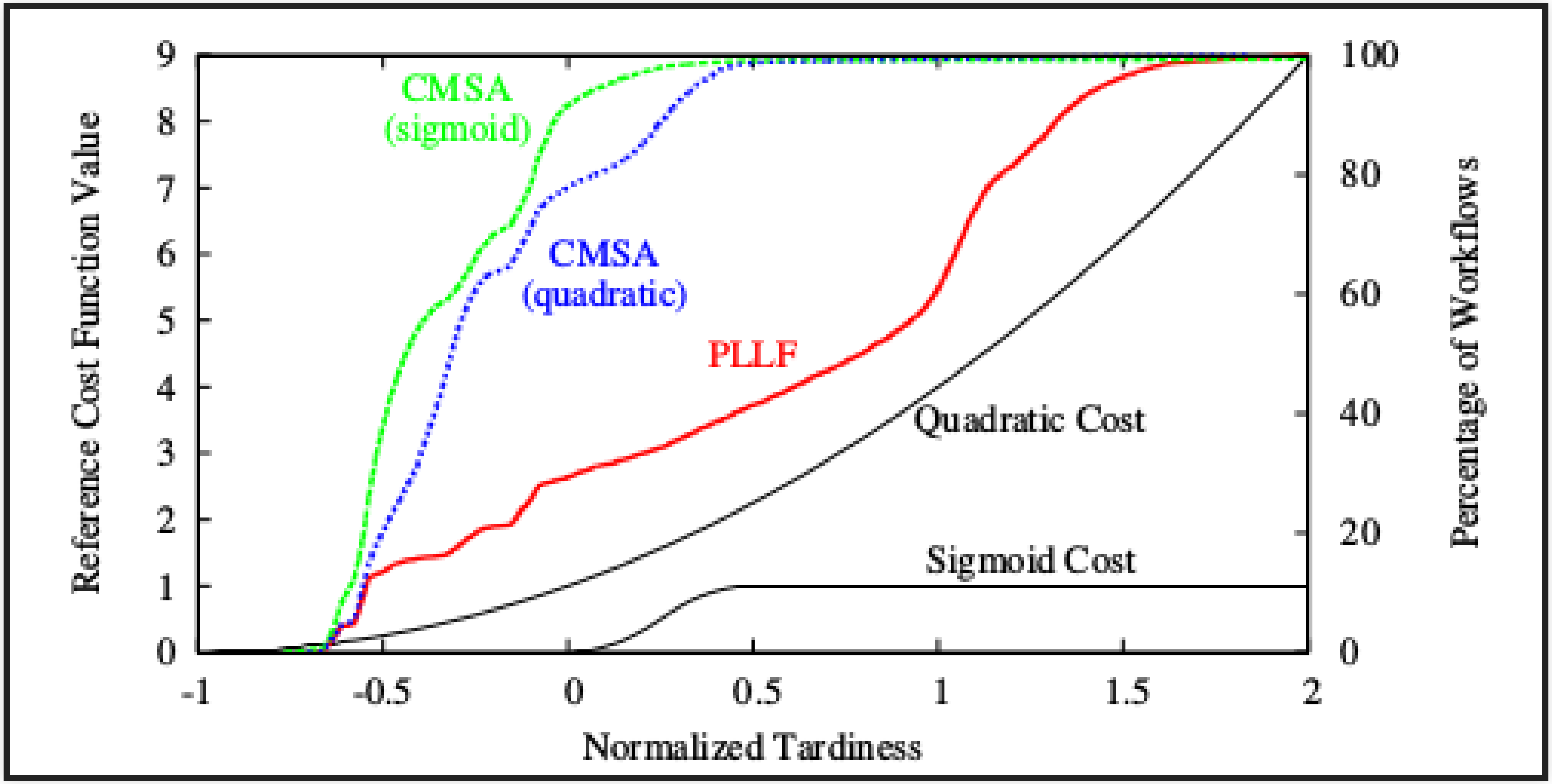
1. First-Come, First-Served (FCFS)

   Prioritizes Tasks based on Earliest-Arriving Workflow
2. Proportional Least Laxity First (PLLF)

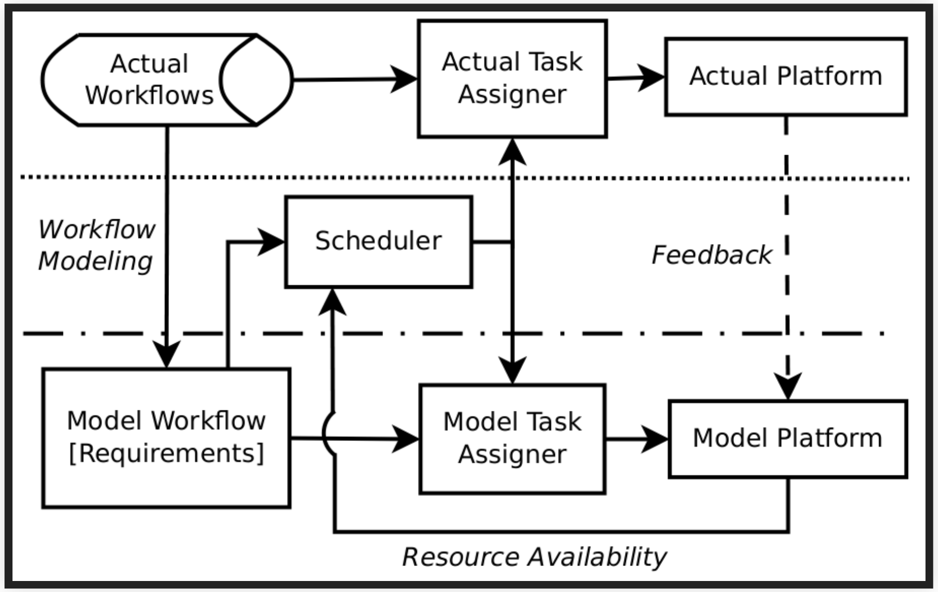
   Prioritizes Tasks based in projected laxity (normalized tardiness) of Workflow
3. Cost-Minimization Scheduling Algorithm (CMSA) [2]

   Prioritizes Tasks based on Cost Function trade-off of completing executing tasks vs. starting more tasks

PAST PERFORMANCE RESULTS



MODELING ERROR



In order to evaluate the impact of error in the information provided to scheduling algorithms, we distinguish Model Platform (used for scheduling decision-making) from Actual Platform (where performance is measured)

DEFINING ROBUSTNESS

Robustness to error in the model can mean:

- Does the Scheduler produce the same schedule (tasks assigned to the same machine at the same time vs. the case of no error)?
- Does the Scheduler achieve the same performance outcome (e.g., makespan, percent workflows late)?

Our research focuses exclusively on the latter

SIMULATION STUDIES

Using Simulation software, the amount of error in the Model (tasks' resource requirements) was taken from a uniformly-distributed random number distribution [-X,X] where X varied from 0.1% to 90%

Each result represented averaged value across 10 simulations where a unique seed was used to generated random error for each tasks' requirement

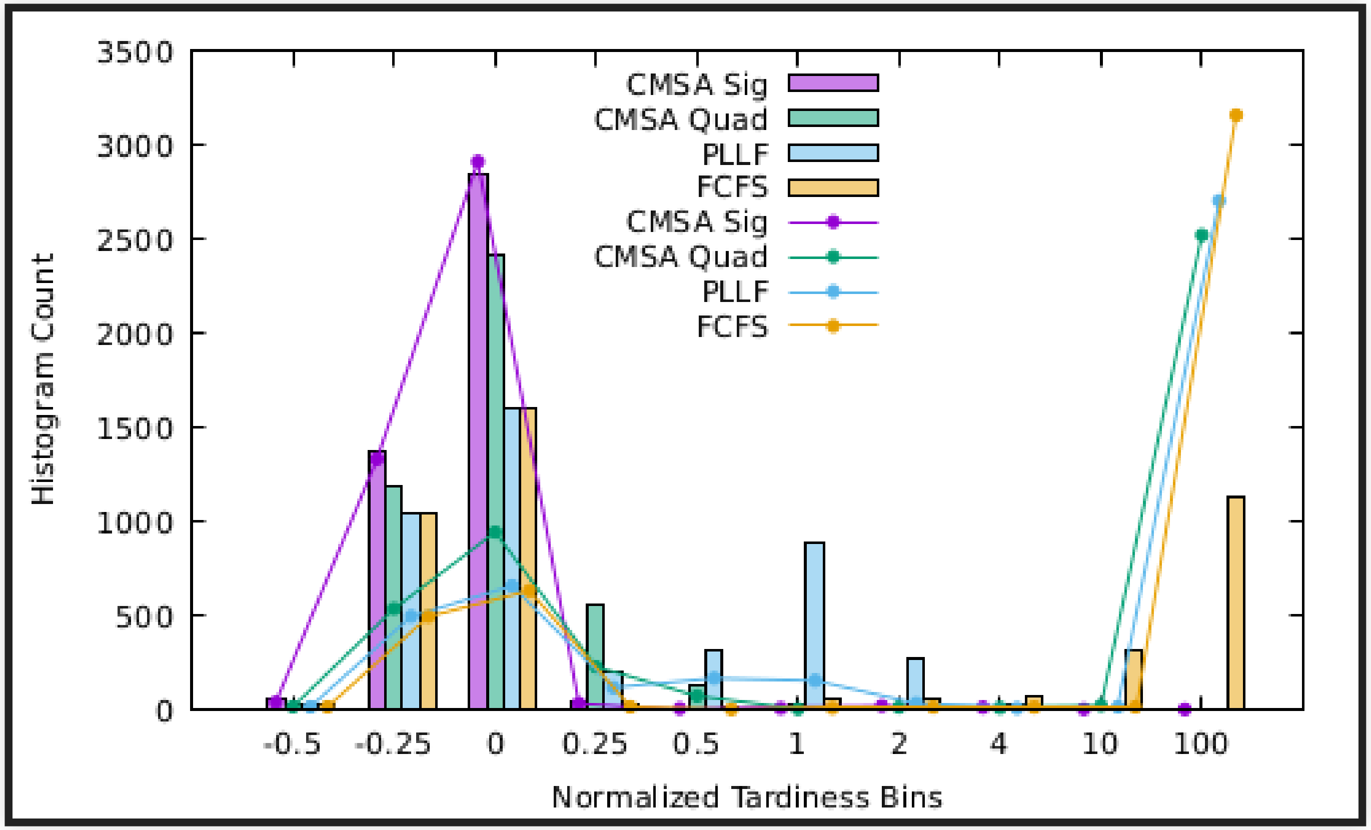
PAST RESULTS

Percentage of all Workflows Late

Error, X	CMSA (Quad)	CMSA (Sig)	FCFS	PLLF
0.0	16.54	1.77	38.49	38.72
0.1%	30.52	1.92	61.39	50.96
0.5%	34.75	1.84	59.48	52.39
1%	30.86	2.99	60.35	49.39
5%	28.87	8.99	63.60	49.54
10%	32.11	15.60	60.77	52.65
50%	34.34	24.96	52.85	55.68
90%	32.93	26.12	54.36	58.64

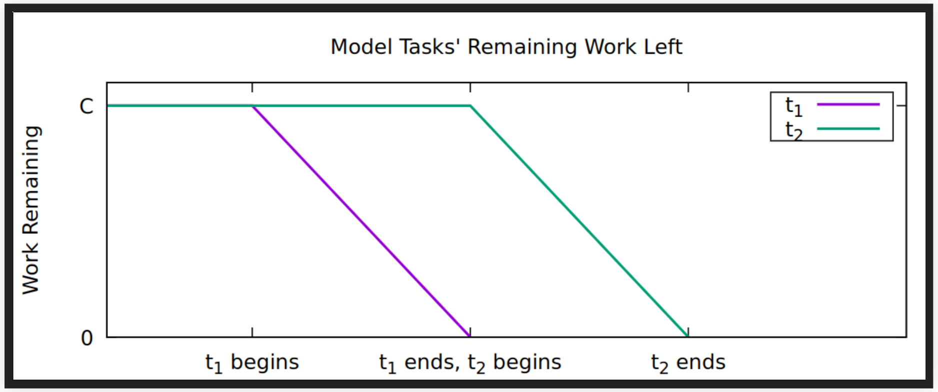
PAST RESULTS

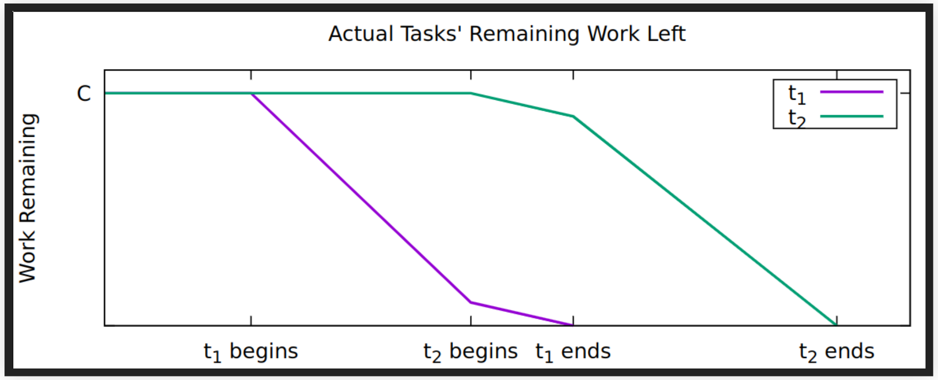
Effect on Tardiness of Small Error, $X = 0.001$



Bars represent no error, lines error $X = 0.001$

MODELING ERROR EFFECTS

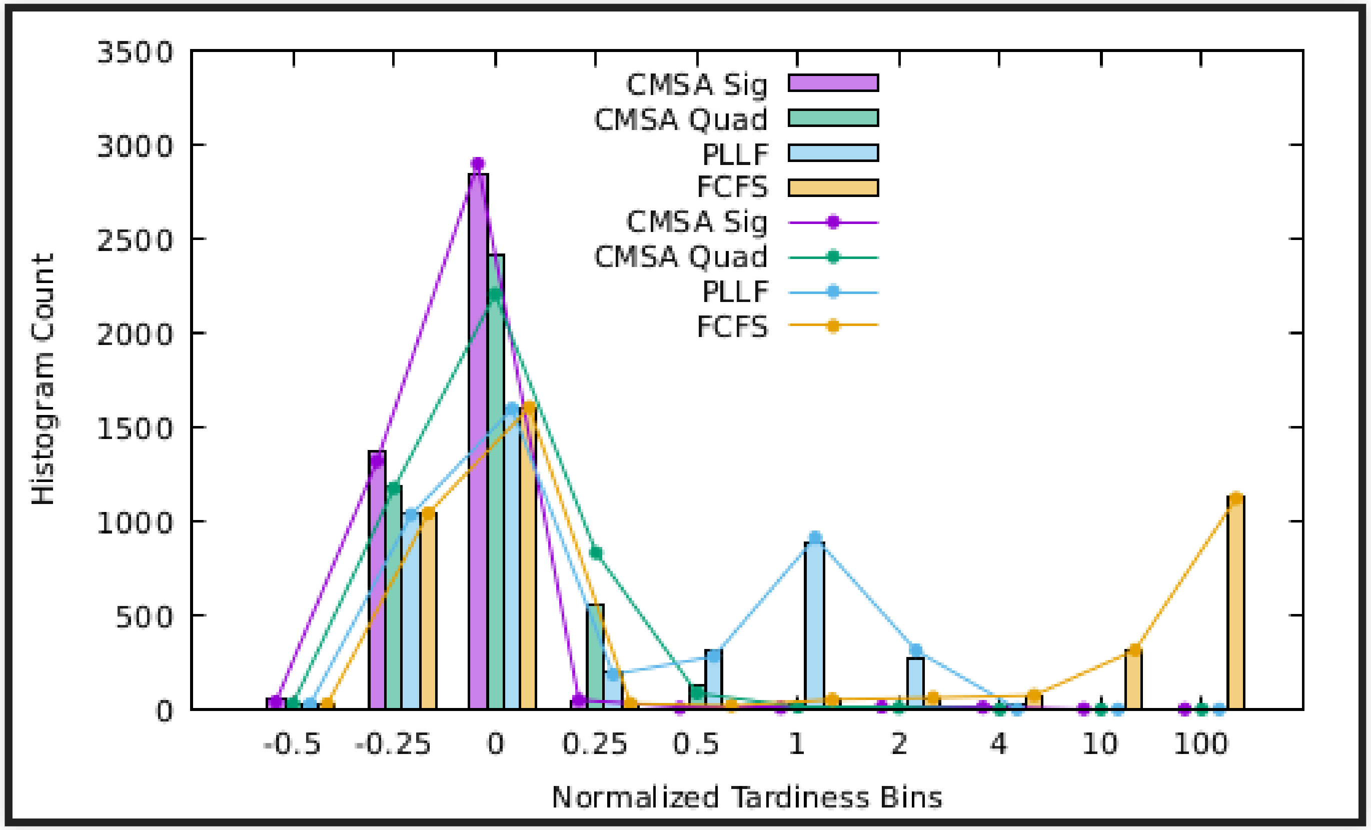




Effect of model underestimating task requirement (hence finish time) causes Scheduling algorithms to assign more task(s) to the machine, resulting in efficiency / production reduction. This effect can be compounding.

PAST RESULTS OF FEEDBACK

Effect of Including Actual Platform Feedback even with Large Error, $X = 0.9$



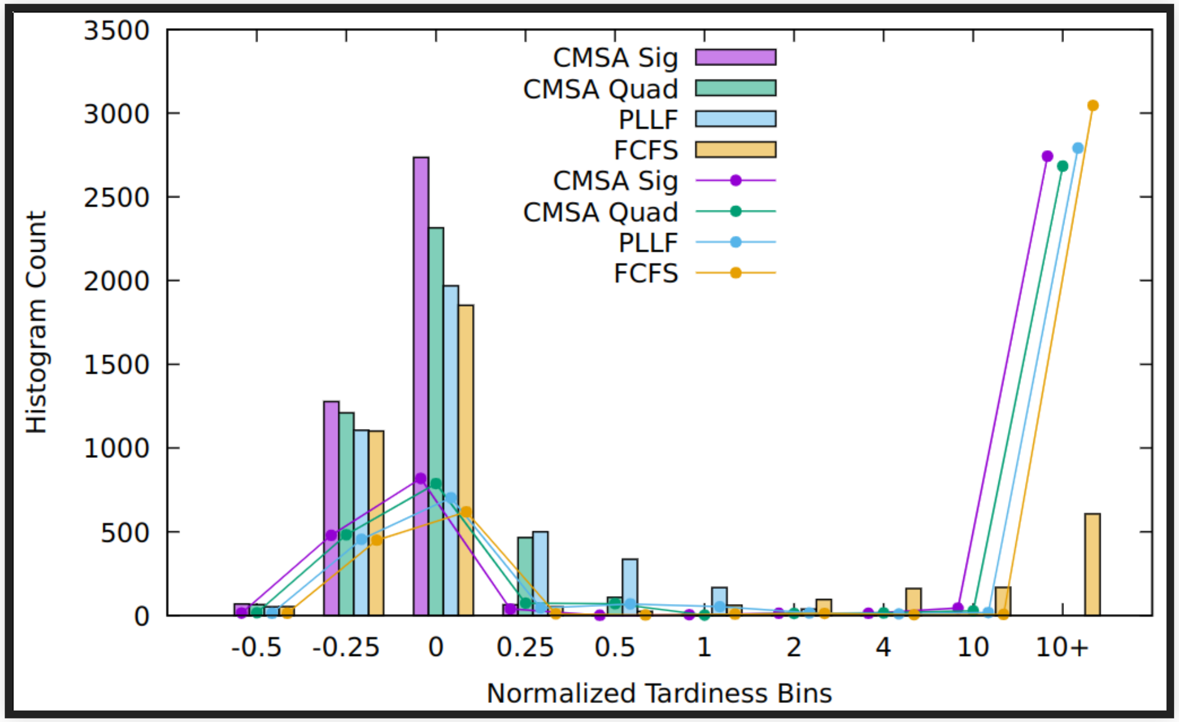
Bars represent no error, lines error $X = 0.9$

CURRENT RESEARCH

- Is full feedback of task completion necessary or can similar results be achieved with partial feedback?
- Can biasing model requirements to avoid underestimating help avoid the compounding problem of modeling tasks complete too early and over-allocating future tasks to machines?

PARTIAL FEEDBACK

Even full feedback doesn't help as long as we allow model to consider tasks complete (early) and Scheduler to assign more tasks to the machine ($X = 0.001$):



Bars represent no error, lines error $X = 0.9$

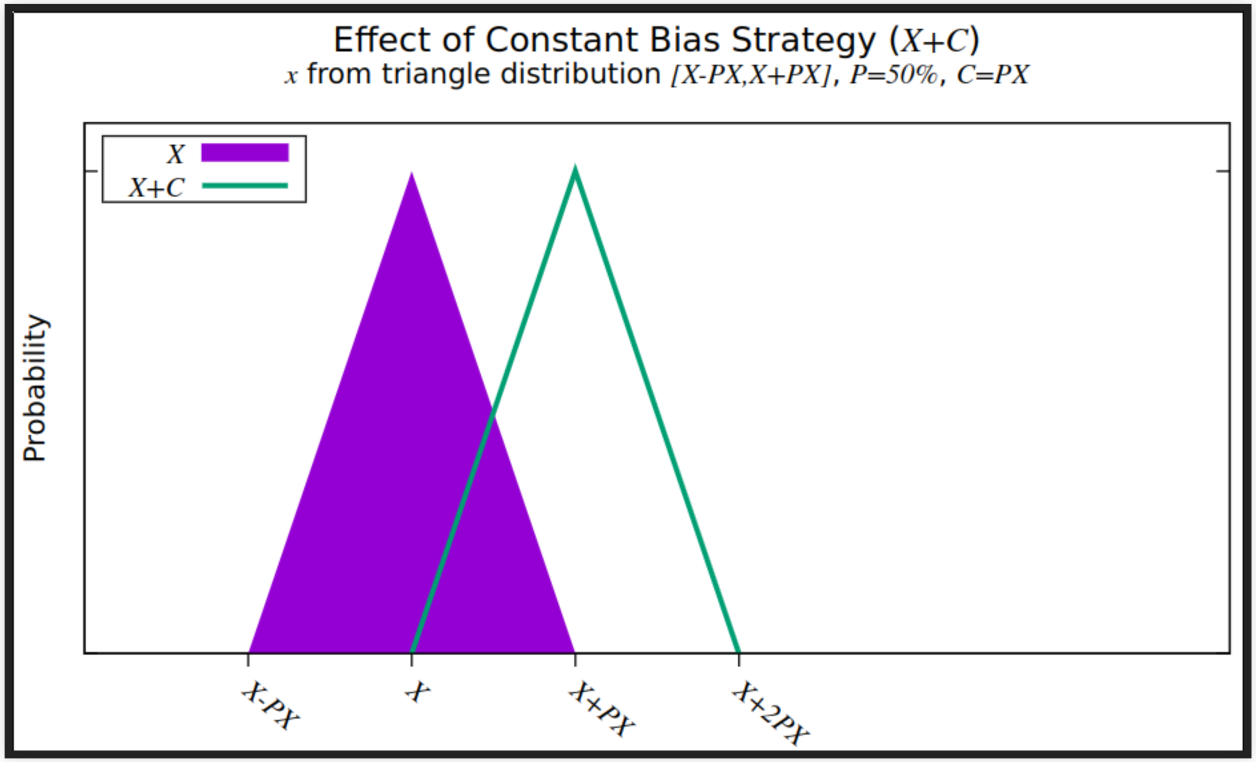
MODEL BIASING

Since model contains terms known to have error, three bias strategies are proposed to counter-act the probability of underestimation:

MODEL BIASING

1. Constant bias strategy:

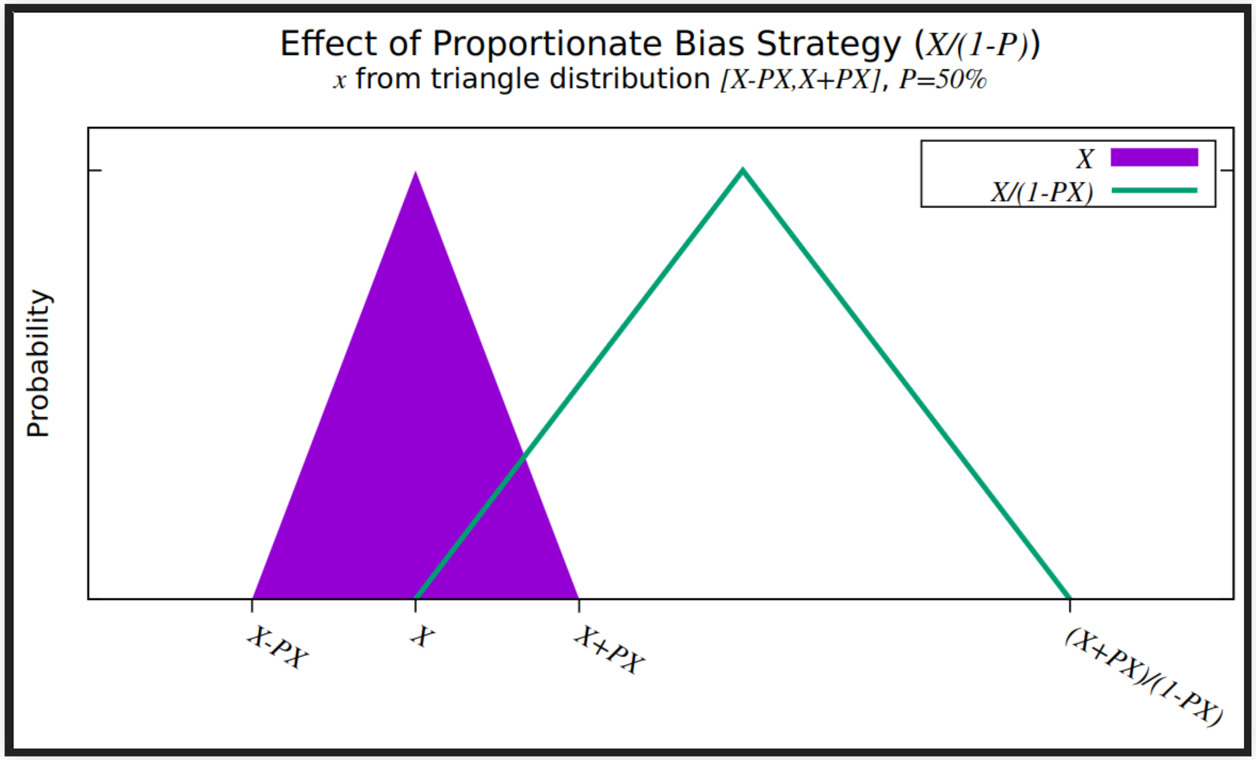
$$\hat{X}^b \xleftarrow{c} \hat{X} + C$$



MODEL BIASING

2. Proportionate bias strategy:

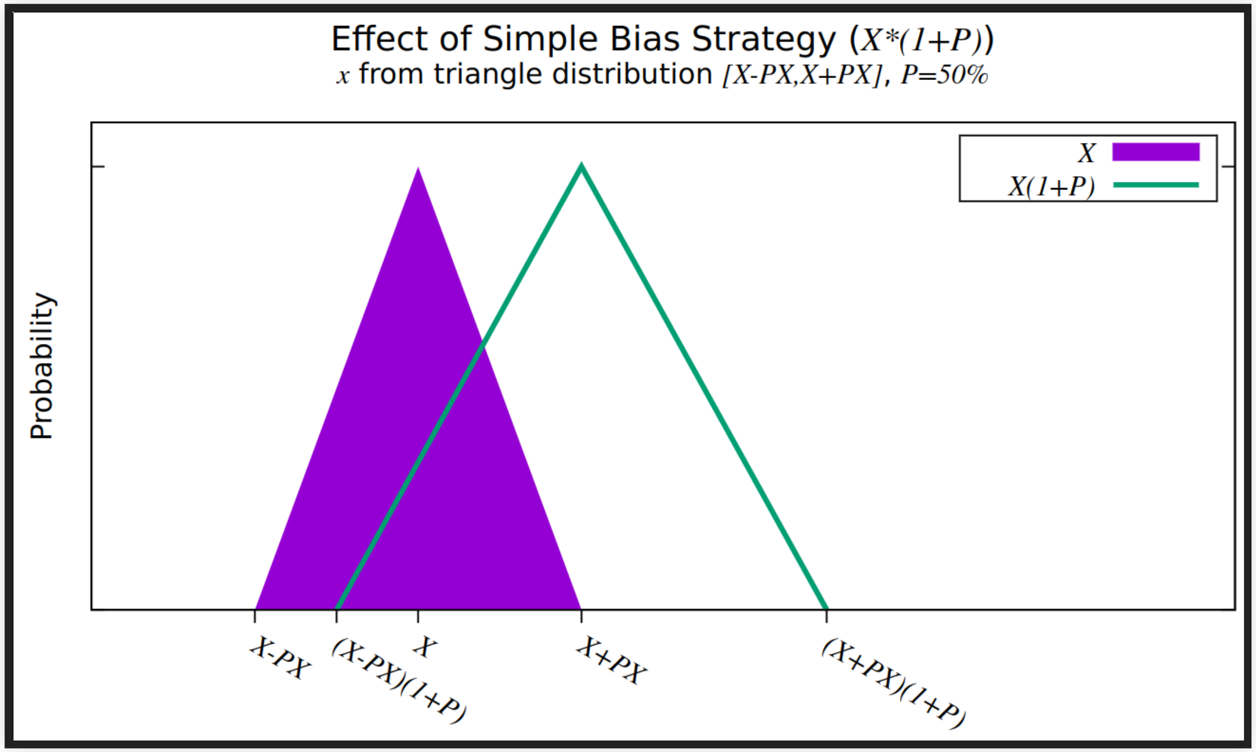
$$\hat{X}^b \xleftarrow{p} \hat{X}/(1-P)$$



MODEL BIASING

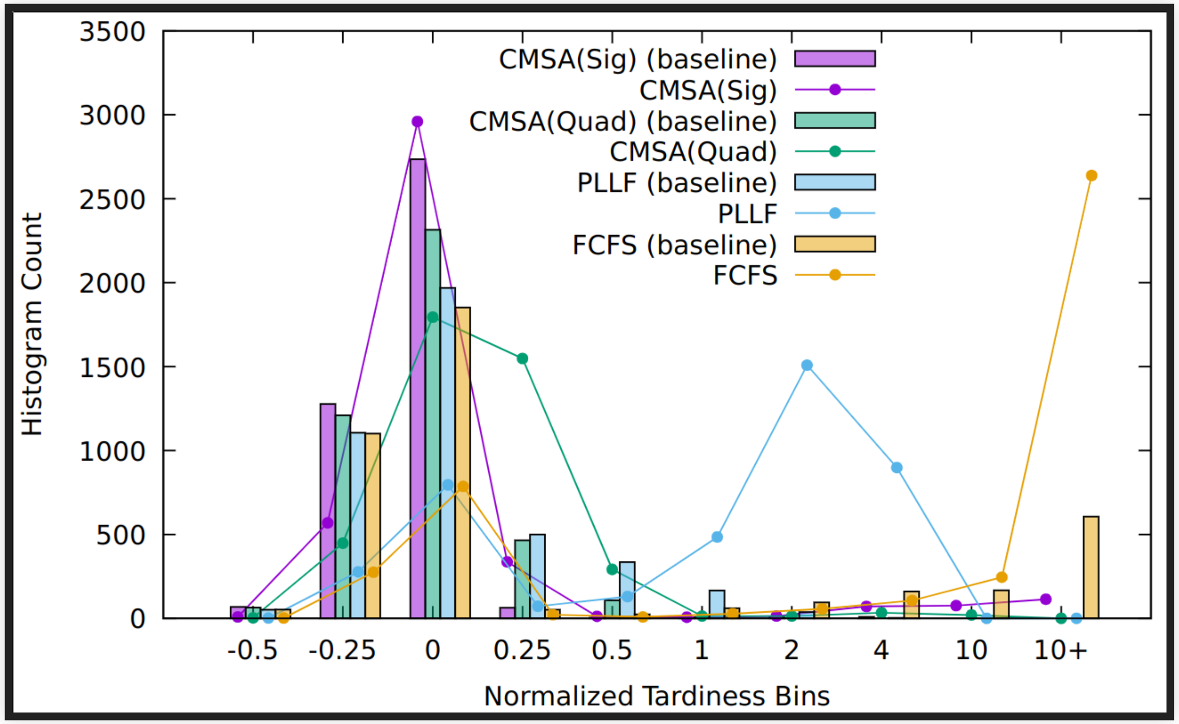
1. Simple bias strategy:

$$\hat{X}^b \overset{s}{\leftarrow} \hat{X}(1+P)$$



CONSTANT BIAS RESULTS

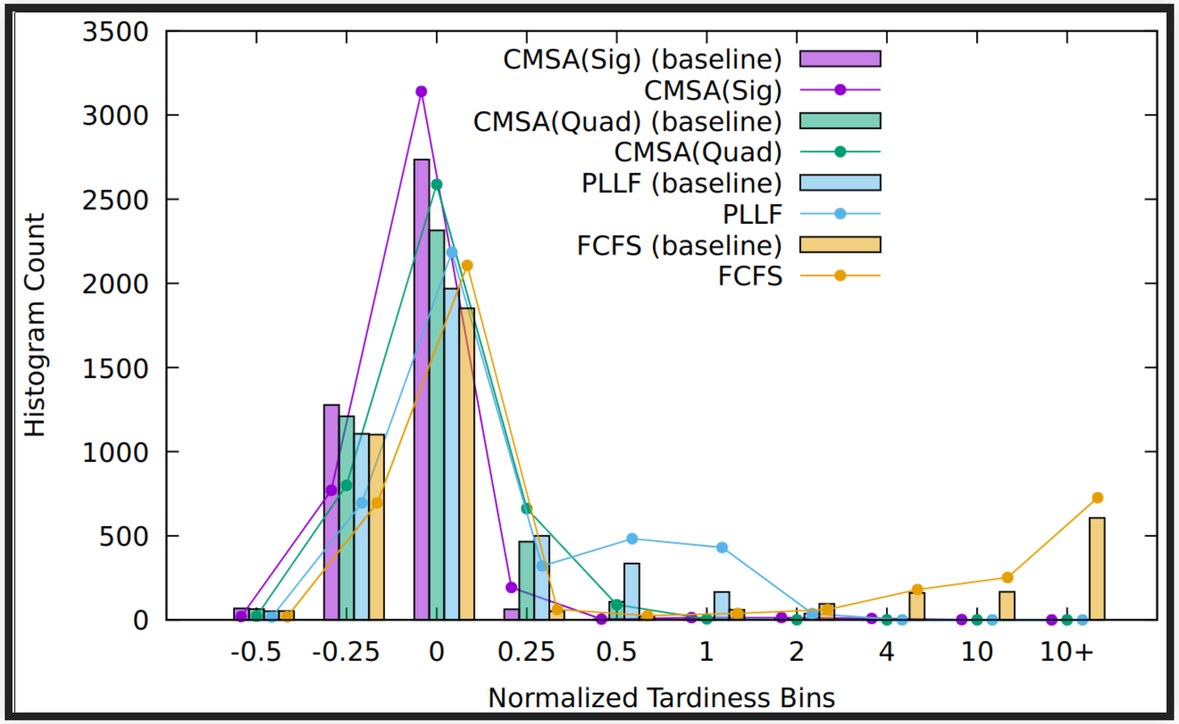
Constant bias strategy achieves somewhat robust performance for CMSA for up to 50% error ($X = 0.5$), less for FCFS and PLLF:



Bars represent no error, lines error $X = 0.5$

PROPORTIONATE BIAS RESULTS

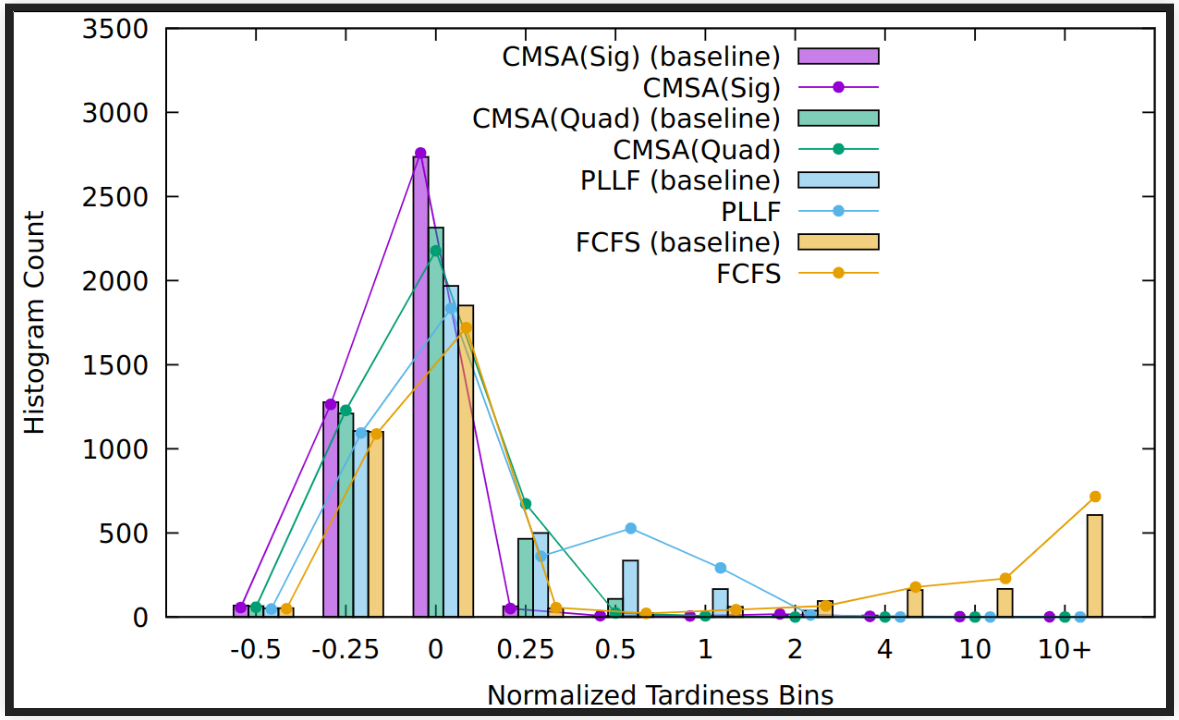
Proportionate bias strategy achieves relatively robust performance for all algorithms but only up to 5% error ($X = 0.05$):



Bars represent no error, lines error $X = 0.05$

SIMPLE BIAS RESULTS

Simple bias strategy achieves relatively robust performance for all algorithms but only up to 5% error ($X = 0.05$):



Bars represent no error, lines error $X = 0.05$

CONCLUSION

Most scheduling algorithms studied have little robustness with respect to even small amount of error [3]

Underestimated work requirement (i.e. negative error in model) leads to overloading machines, reducing efficiency, compounding the problem of underestimating

Full feedback without model modeling tasks as complete achieves robustness for all studied algorithms

Bias strategies can achieve moderate robustness to [not large] model error when full feedback isn't available, but require various level of knowledge of the error bounds

FUTURE WORK

- Non-Uniform Error
- Error in Other Aspects of Modeled Tasks / Workflows
- Viability of biasing if error bounds is unknown

THANK YOU

References:

1. Shrestha, H.K., Grounds, N., Madden, J., Martin, M., Antonio, J.K., Sachs, J., Zuech, J., Sanchez, C.: Scheduling workflows on a cluster of memory managed multicore machines. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA ’09) (July 2009)
2. Grounds, N., Antonio, J.K., and Muehring, J.: Cost-minimizing scheduling of workflows on a cloud of memory managed multicore machines. In MartinGilje Jaatun, Gansen Zhao, and Chunming Rong, editors, *Cloud Computing*, volume 5931 of *Lecture Notes in Computer Science*, pages 435–450. Springer Berlin Heidelberg, 2009.
3. Grounds, N., and Antonio, J.K.: A Model-Based Scheduling Framework for Enhancing Robustness. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '18) (July 2018)