

Cloud GPU Rendering Scheduling Problem (CGRSP)

A mathematical formulation and empirical evaluation on heterogeneous GPU fleets

Michael Liem (M11215818) | Dept. of CSIE, NTUST
Advisor: Prof. Yuan-Shin Hwang | Co-advisor: Prof. Vincent F. Yu

Why scheduling rented GPUs is hard

Heterogeneous fleet

RTX 3090, A5000, A4000 and A4500 differ in speed and hourly price, so the same job costs and takes differently per node.

Availability

Stock status drives provisioning delay, and execution time is stochastic (log-normal per complexity level).

Mixed deadlines

Assumed mix of 20% rush (1 h) and 80% standard (8 h) renders, stress-tested on a saturated hectic day, sharing $C_{\max} = 5$ workers.

Question: which dispatch policy keeps deadlines and waits low at minimal rental cost?

Contributions

- 1 Complete CGRSP formulation with stock-driven availability and bi-objective QoS-cost
- 2 Calibrated, constraint-checked discrete-event simulator (6,627 real jobs)
- 3 Ten-policy comparison: 30 paired seeds, Holm-corrected tests
- 4 Two schedulers: RH (anticipative) and CADR (cost-aware, low tardiness)
- 5 Regime map: EDF vs SPT crossover, C_max lever, monthly scale
- 6 Deployment rules grounded in paired tests

Formulation: QoS first, cost as tiebreaker

lex min (C_QoS , C_Cost)

$$C_{\text{QoS}} = \sum [\alpha_1 \cdot \text{wait}_j + \alpha_2 \cdot (1 - \delta_j)] \quad C_{\text{Cost}} = \sum c_j$$

Constraints

Each job executes exactly once over the service horizon. Dispatch follows submission; completion equals start plus execution. One job holds a node at a time.

Stochastic inputs

Execution time $\sim \text{LogNormal}(\text{complexity}, \text{GPU})$.
Provisioning delay depends on stock status and time of day.

Worked example: three jobs, two GPU types

Job	Complexity	Deadline	Placed on	Runtime
j1	Low	1 h (tight)	RTX 3090	59.7 s
j2	Medium	8 h	RTX A4000	68.7 s
j3	Low	8 h	RTX A4000	60.0 s

QoS first, cost only among QoS-equivalent options.

Outcome

All start at $t = 0$ with zero wait and meet every deadline, so cost decides placement. Total rental cost is about \$0.017 and utilisation is 3 of 5 slots (60%).

Simulator calibrated on real RunPod jobs

6,627

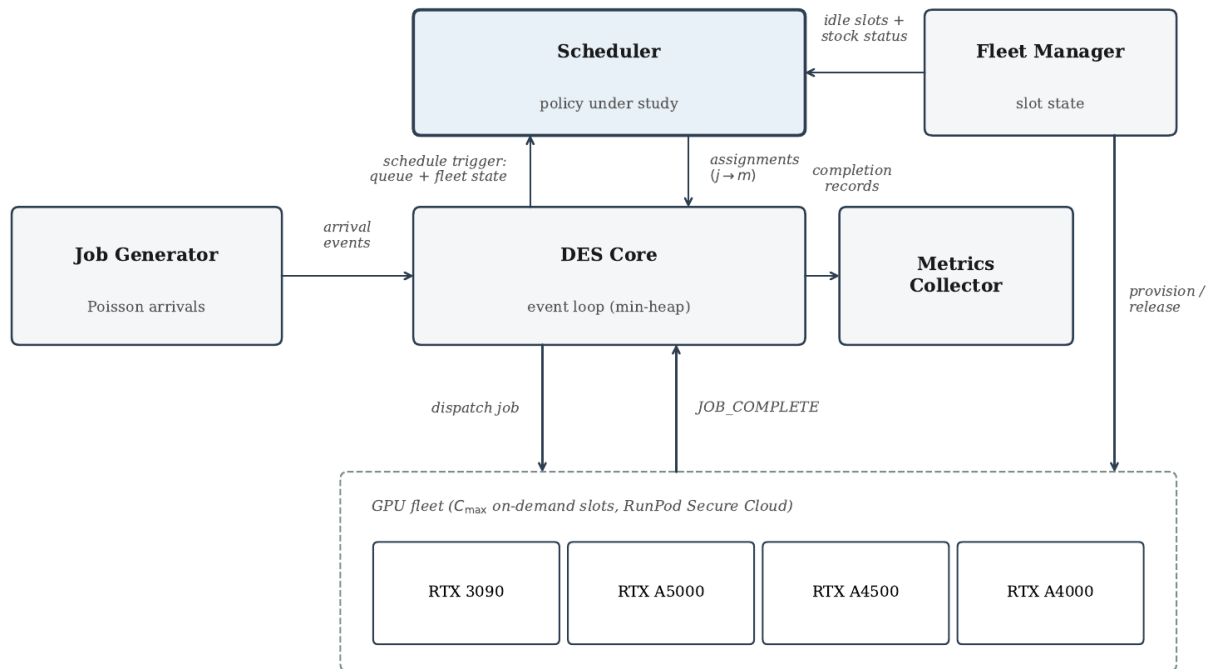
real rendering jobs (Jul
2025 to Jun 2026)

10

schedulers compared

30

seeds per scenario



Methods and validation

Statistics

Every scheduler runs on the same 30 seeds (paired). Paired t-test with Wilcoxon backup, Holm-Bonferroni over a family of 24 comparisons, Cohen's d for effect size. Ties are reported with their p-values.

Simulator checks

Log-normal service times fitted per complexity and GPU stratum (3 sparse strata use an estimated mean). A constraint verifier re-checks all ten policies on the seed-42 hectic run, and the baseline sweep row reproduces Phase 3 exactly.

Ten policies, two of them proposed

Rolling-Horizon (RH)

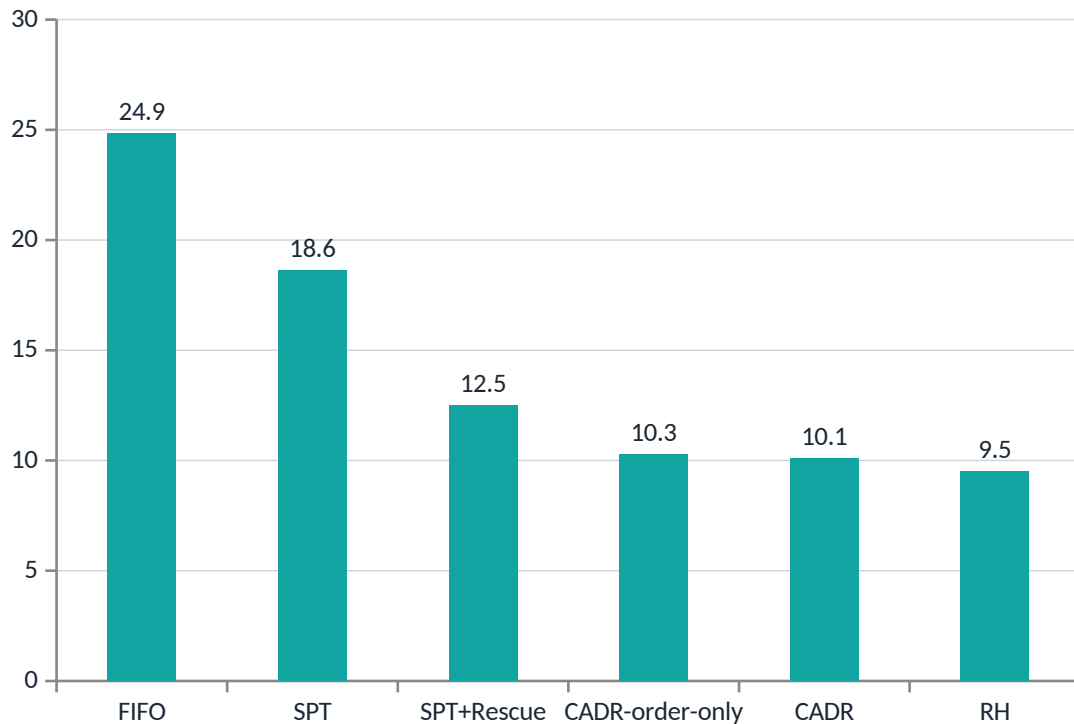
Plans over the slot-availability timeline and anticipated arrivals. Keeps SPT order for jobs with slack, promotes jobs that would otherwise miss, demotes doomed ones. Reserves idle capacity for rush orders below saturation.

CADR

Classifies jobs by a scale-free critical ratio and picks the cheapest deadline-feasible node. Matches RH on miss rate with far lower tardiness.

Baselines: FIFO, SPT, EDF, LCF, Balanced, Random, SPT+Rescue, Adaptive.

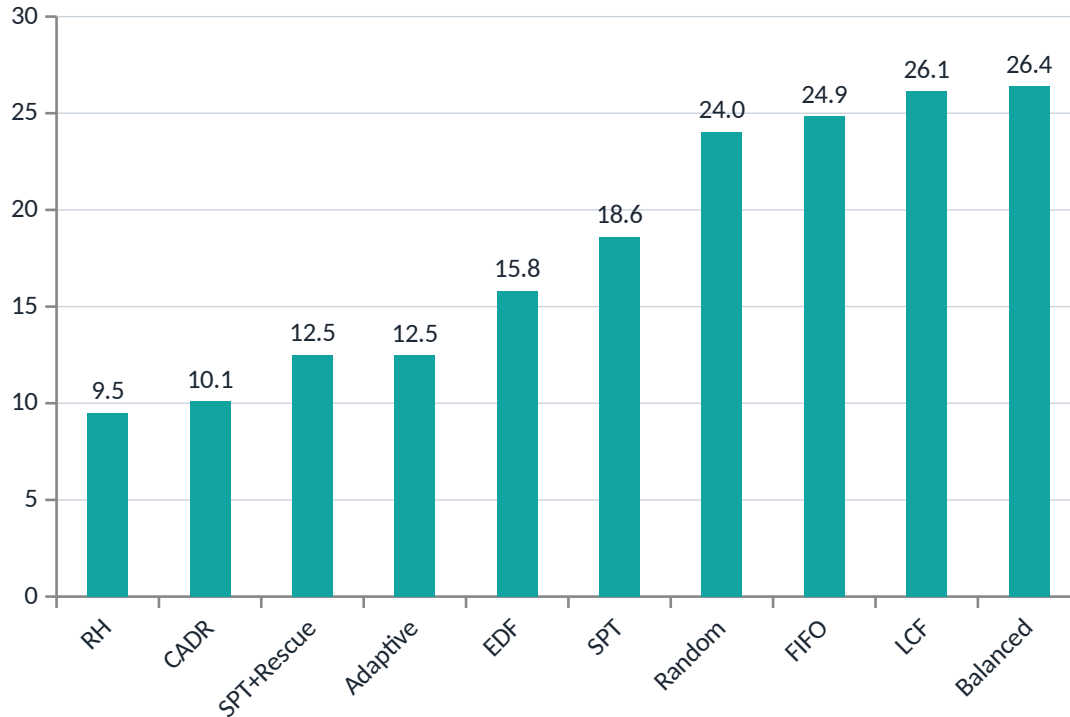
Ablation: what each ingredient buys



Reading it

Hectic day, 30 seeds. SPT cuts wait; rescue and deadline-risk ordering cut miss. RH beats SPT+Rescue on miss after Holm correction, but not on wait.

Hectic-day benchmark: deadline miss rate (%)



9.5%

RH lowest miss, tied with CADR

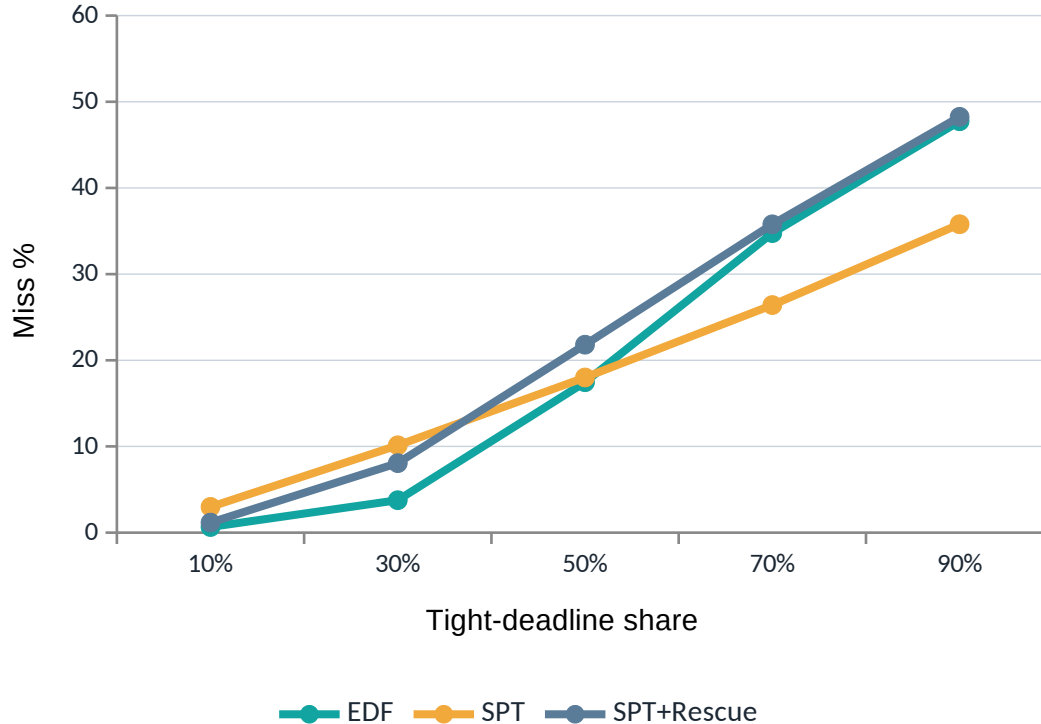
6.4 min

CADR tardiness vs 30.9 min for RH

140 min

RH wait, tied with SPT (141)

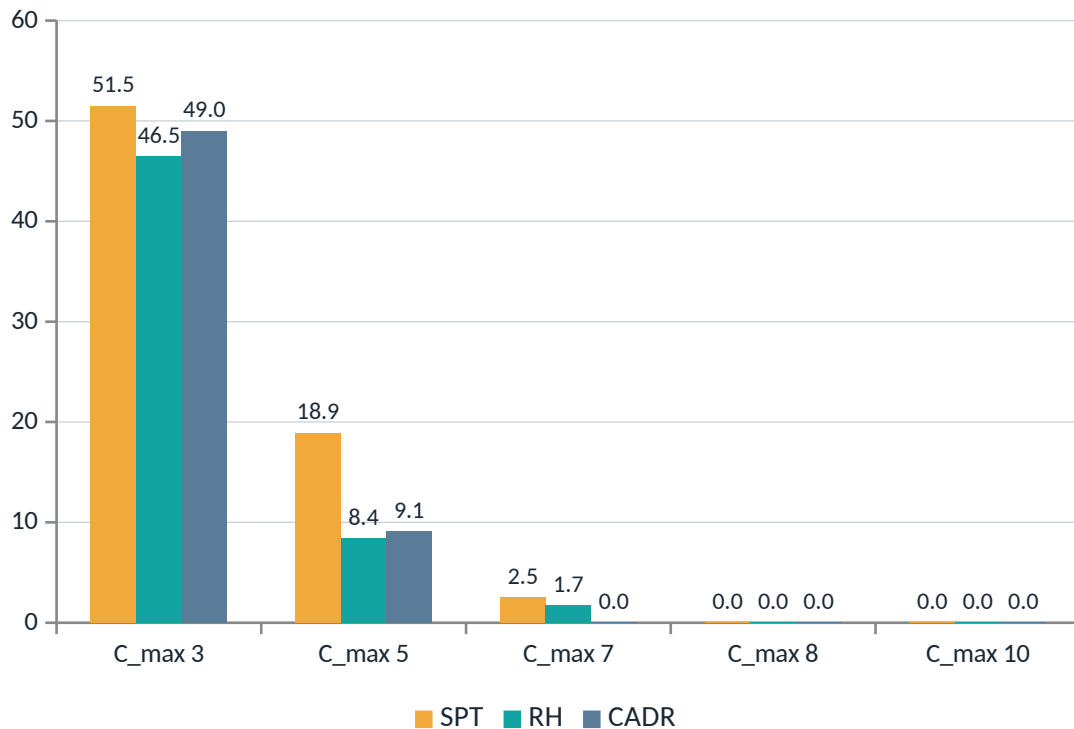
Where EDF and SPT trade places



Crossover

Surge-day load. EDF has significantly lower miss at 30% tight or less, the two tie at 50%, and SPT wins at 70% or more.

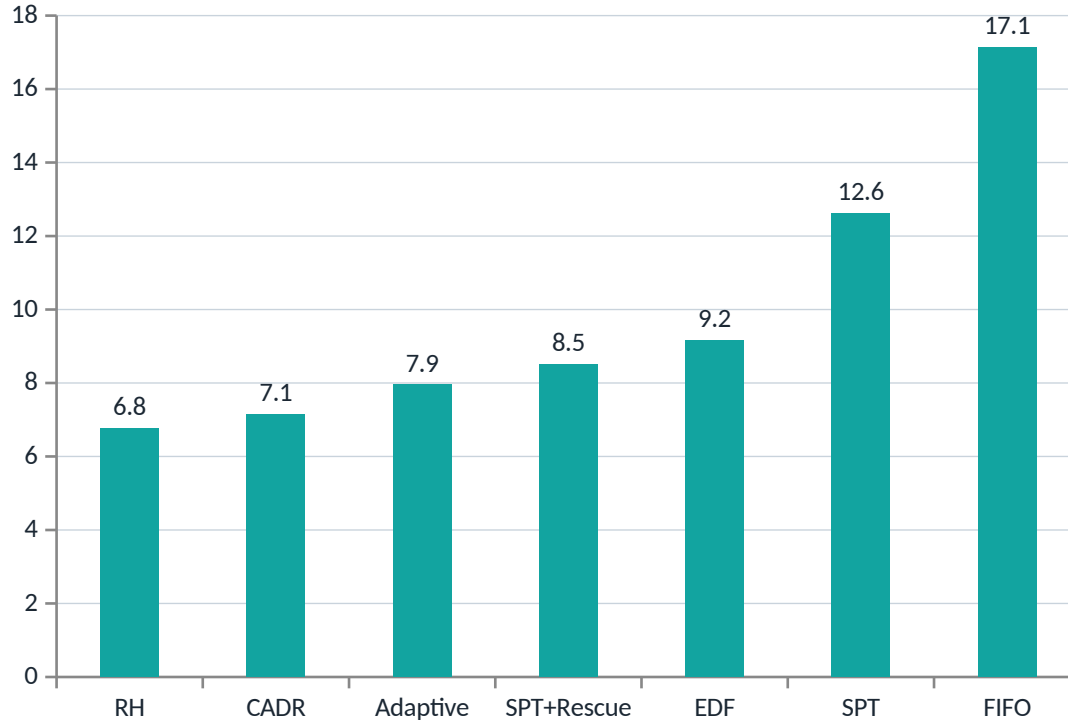
Concurrency cap is the main lever



Takeaway

Hectic day, 10 seeds (baseline differs from the 30-seed Phase 3 values). Miss falls under 3% at C_max 7 and to 0% for all three at 8, our current quota, where the policies tie. More slots, not a better ordering, remove the saturation misses.

Monthly: RH and CADR have the two lowest miss rates



30-day, 10 runs

One continuous 30-day calendar with backlog carried across days and drained at the end. Miss: CADR 7.14% vs RH 6.76%. Tardiness: 6.3 vs 19.3 min.

Snapshots contradict the assumed ranking

Policy	Assumed ranking	Snapshot-ordered ranking
CADR	10.09%	3.04%
SPT+Rescue	12.50%	3.24%
RH	9.50%	6.24%
SPT	18.60%	9.46%
FIFO	24.85%	17.90%

Deadline miss rate, hectic day, 30 paired seeds

Level sweep, x0.75 to x1.00: deadline-aware beats deadline-blind and RH is lowest, tied with CADR at x1.00 and ahead only at x0.75. No policy copes at x0.50 or below; several reach zero miss at x1.25.

Concede

Snapshots list the 3090 94% of the time, the A4000 63%. That is listing share, not fulfilment; beta stays an assumption.

Consequence

RH ties CADR at baseline ($p = 0.484$). Under snapshot order CADR beats RH ($p < 0.0001$) and ties SPT+Rescue. Default to CADR.

Deployment guidance

Default: CADR

Ties RH on Phase 3 miss, has close monthly miss point estimates, and far lower tardiness.

Specialist: RH

Scarce-availability regime when only missed-deadline count or wait matters.

Raise C_max

At saturation, lift the cap above RunPod's default of 5 (our quota is 8) to cut residual misses.

SPT remains a strong low-wait baseline. At our quota of 8 the policies tie on miss; the scheduler matters when the cap binds.

Limitations and future work

Limitations

Simulation only: no tight optimality bound (exact ILP intractable at about 950 jobs and not attempted even at toy size; the capacity bound certifies zero unavoidable misses, not the stochastic optimum). Hectic and surge loads are assumed stress loads (busiest recorded day: 415 jobs). Availability probabilities and delays are assumed; snapshots show scarce stock and rank GPU types in the reverse of the assumed order. Pods price snapshot, single-studio workload, non-preemptive jobs.

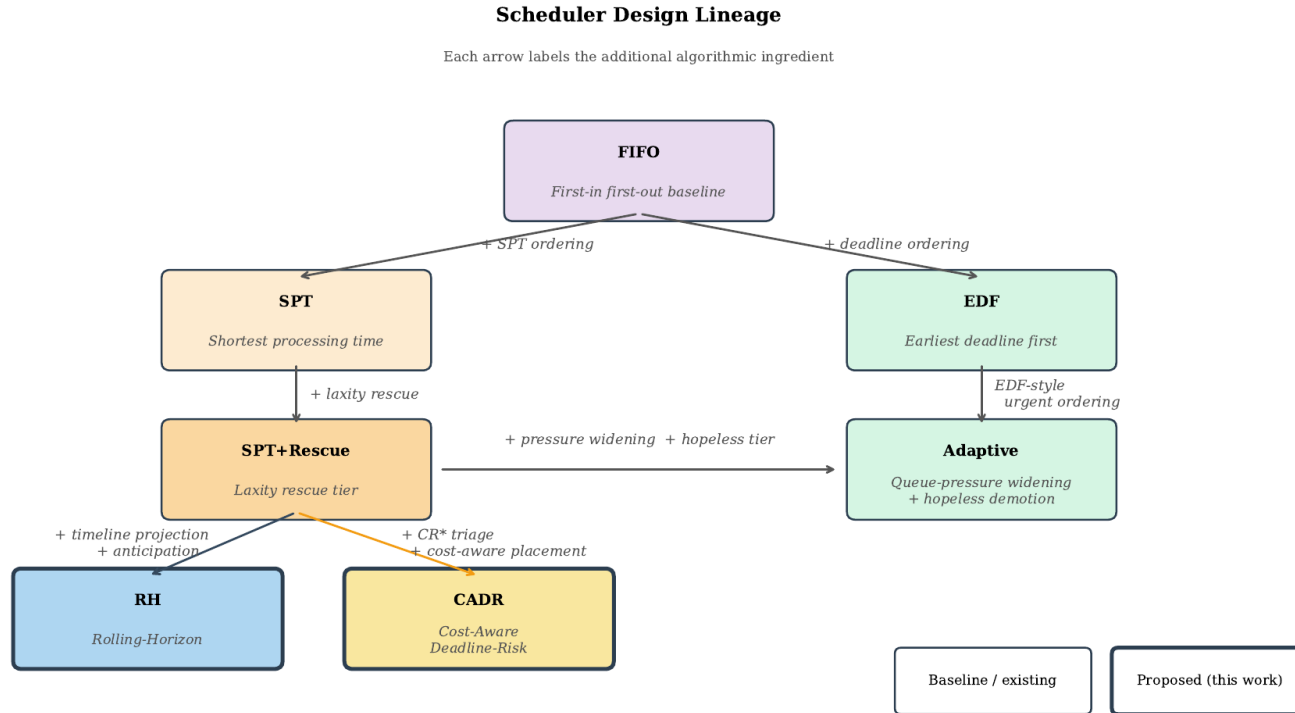
Future work

Validate on a live pipeline, add spot and reclaim dynamics, test learning-based dispatch, and extend to multi-provider fleets.

Thank you

Questions and discussion

Backup: scheduler lineage



Backup: Phase 3 full results

Scheduler	Wait (min)	Miss %	Tardiness (min)	Cost (\$)
RH	140.24	9.50	30.86	6.48
CADR	145.70	10.09	6.43	6.47
SPT+Rescue	145.85	12.50	6.64	6.49
Adaptive	169.64	12.49	4.45	6.48
SPT	141.03	18.60	11.99	6.45
EDF	174.16	15.81	5.74	6.48
LCF	173.36	26.14	28.50	6.53
Balanced	173.24	26.39	28.30	6.52
FIFO	169.32	24.85	26.69	6.50
Random	172.80	24.04	35.18	6.50

Hectic day, 30 seeds.

Backup: deadline sweep, miss %

Tight share	FIFO	SPT	EDF	LCF	Balanced	Random	SPT+Rescue
10%	5.26	2.98	0.65	5.30	5.21	4.68	1.16
30%	16.53	10.13	3.76	16.66	16.44	14.23	8.06
50%	27.23	18.00	17.45	27.43	27.06	23.70	21.81
70%	38.12	26.41	34.76	38.33	37.80	32.75	35.78
90%	48.91	35.79	47.74	49.26	48.50	42.07	48.25

Surge day, 30 seeds per point.

Backup: C_max sweep, wait / miss

C_max	FIFO	SPT	EDF	SPT+Rescue	RH	CADR
3	392.6 / 57.6	358.7 / 51.5	395.4 / 51.9	379.0 / 50.1	369.8 / 46.5	377.4 / 49.0
5	164.5 / 22.9	141.7 / 18.9	176.7 / 16.4	145.0 / 12.4	135.4 / 8.4	142.3 / 9.1
7	46.0 / 6.6	39.0 / 2.5	45.5 / 0.0	38.8 / 0.0	38.5 / 1.7	39.2 / 0.0
8 (current)	29.0 / 0.5	24.7 / 0.0	29.2 / 0.0	24.7 / 0.0	34.9 / 0.0	24.9 / 0.0
10	9.4 / 0.0	8.3 / 0.0	9.2 / 0.0	8.3 / 0.0	13.7 / 0.0	8.2 / 0.0

Wait in minutes / miss %, hectic day, 10 seeds.

Backup: Phase 4 monthly results

Scheduler	Wait (min)	Miss %	Tardiness (min)	Cost (\$/mo)
RH	102.85	6.76	19.30	48.56
CADR	103.10	7.14	6.26	48.88
SPT+Rescue	101.67	8.50	4.74	48.98
Adaptive	117.83	7.95	3.64	49.15
SPT	101.13	12.61	10.16	48.75
EDF	118.62	9.15	3.97	50.43
LCF	118.22	17.35	18.75	48.97
Balanced	119.85	17.30	19.39	49.59
FIFO	118.50	17.14	18.62	54.36
Random	119.84	16.37	24.33	52.40

30 days, 10 runs. Miss values are point estimates; no paired comparison of policies at monthly scale.