The 25th International Conference on
Algorithms and Architectures for Parallel Processing

MPPTS: Multi-factor Predictive Priority Task Scheduling

Algorithm for Heterogeneous Systems

Authors: Luyao Jiao, Hamza Djigal, Abdoul Fatakhou Ba, Abdullahi UwaisuMuhammad

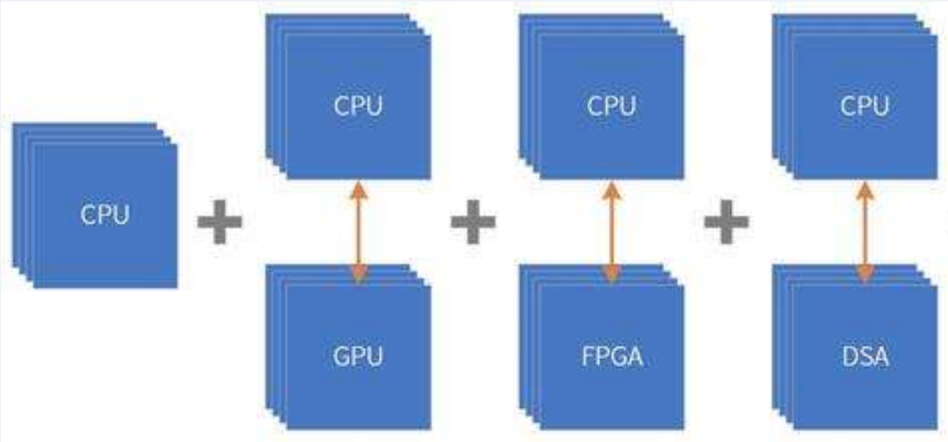
CONTENTS

01 Background

Heterogeneous Computing Systems (HCS):
There are computing resources of different types and performances in the system, which work together to complete a task.

Typically includes:

- Multiple processing units
- Different computing speeds
- Different communication capabilities (bandwidth and latency).



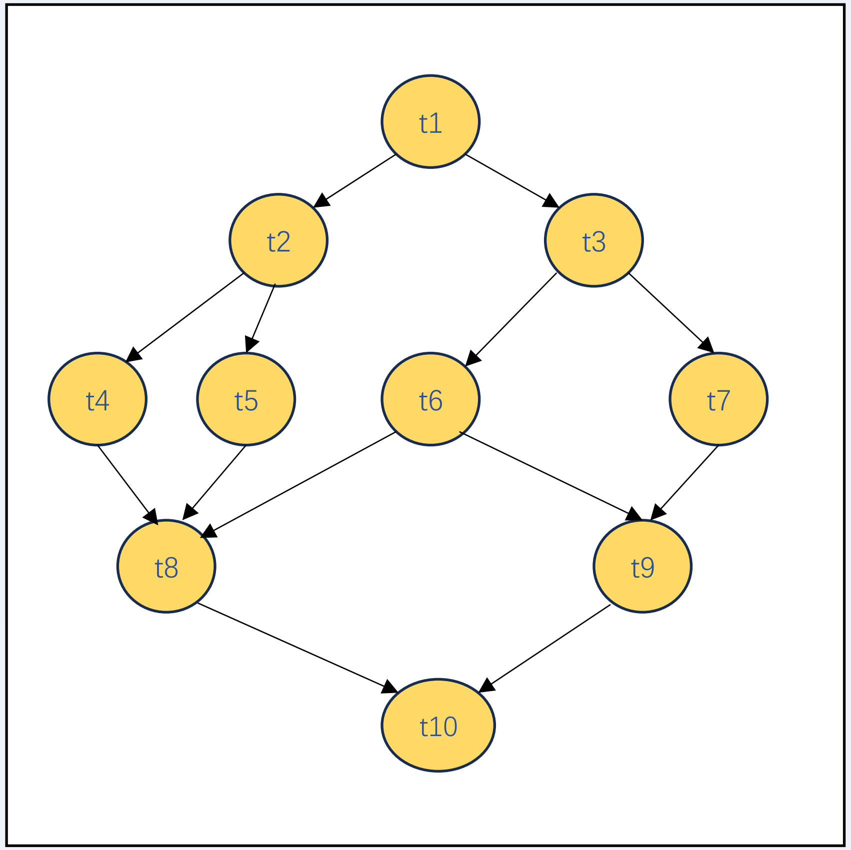
01 Background

Workflow:

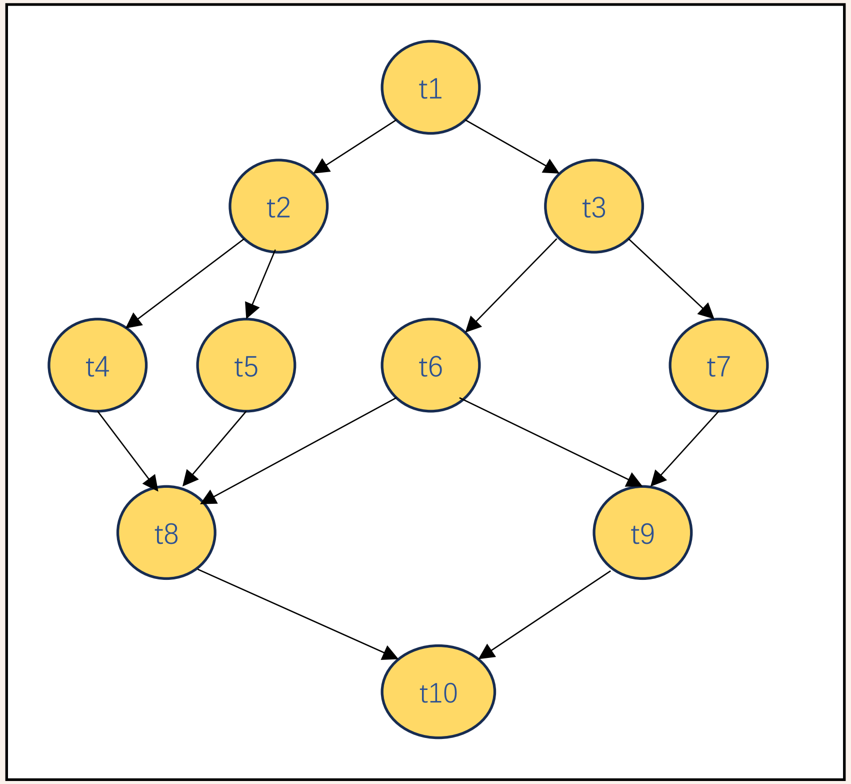
- It consists of many interdependent tasks, each of which is assigned to a different computing node for execution.

- In research, we can simplify the workflow into a Directed Acyclic Graph (DAG).
- The nodes represent tasks, and the arrows represent data dependencies.

- We hope to use a scheduling algorithm to properly arrange the execution order and allocation of these tasks, minimizing the completion time of the entire workflow.



02 TASK-SCHEDULING PROBLEM



Terminology

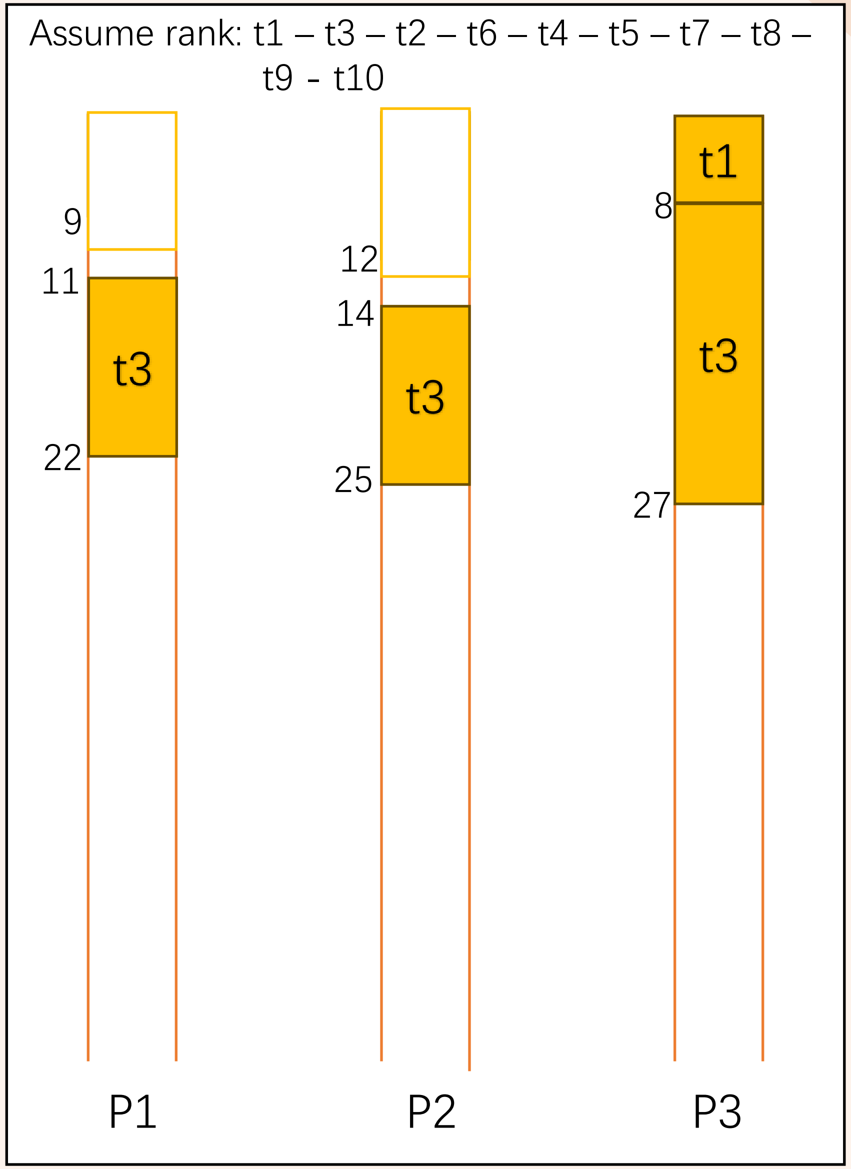
$\omega(v_i, p_j)$, $c_{i,k}$

EST (n_i, p_j)

EFT (n_i, p_j)

Rank

Scheduling



PREVIOUS WORK: HEFT[1]

average communication cost on all processors rank value of the successor task

- Task prioritizing phase:

Sorting tasks by decreasing order of:

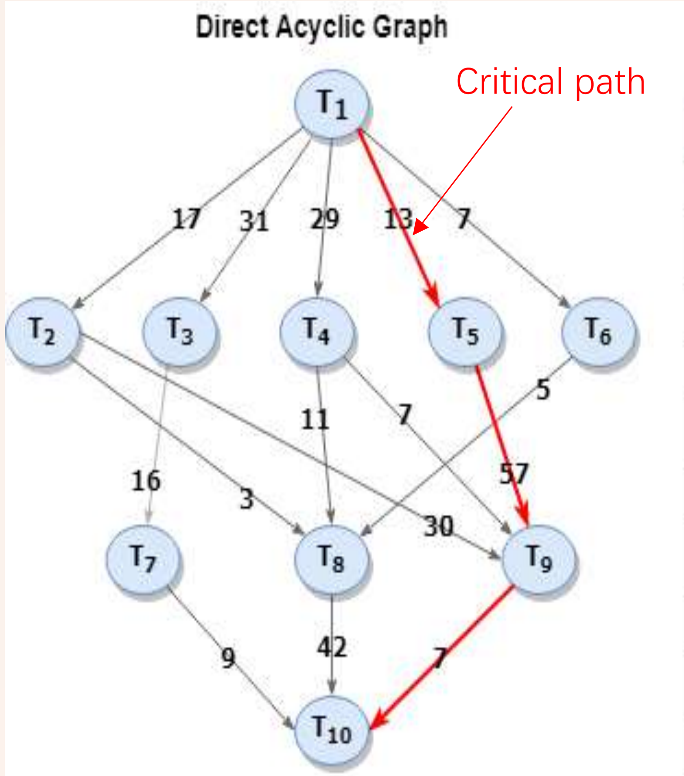
$$rank_u(n_i) = \overline{w_i} + \max_{n_j \in succ(n_i)} (\overline{c_{i,j}} + rank_u(n_j)),$$

Execution rank of tasks average computation cost on all processors

- Processor selection phase:

Arrange task to that processor with the minimum EFT

Disadvantage: it selects a processor for a task t_i based only on the current optimal solution without considering any influence of an assignment for all successor tasks.



Computation Costs

Task	P1	P2	P3
T_1	22	21	36
T_2	22	18	18
T_3	32	27	43
T_4	7	10	4
T_5	29	27	35
T_6	26	17	24
T_7	14	25	30
T8	29	23	36
T_9	15	21	8
T_{10}	13	16	33

1) $\boldsymbol{rank_u(t_{10})} = (13 + 16 + 33)/\,3 = \mathbf{20.66}$

2) $rank_u(t_9) = (15 + 21 + 33)/3 + \{\boldsymbol{rank_u(t_{10})} + 7\} = \mathbf{42.3}$

[1] H. Topcuoglu, S. Hariri, and M.-Y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," IEEE Trans. Parallel Distrib. Syst., vol. 13, no. 3, pp. 260–274, 2002. cited By 1626.

PREVIOUS WORK: PPTS[2]

PCM value of successor tasks

computation cost on processor γ

- Task prioritizing phase:

Sorting tasks by decreasing order of $rank_{PCM}$:

$$PCM(t_i, p_j) = \max_{t_k \in succ(t_i)} \left[\min_{p_\gamma \in p} \{PCM(t_k, p_\gamma) + w(t_i, p_\gamma) + w(t_k, p_\gamma) + \overline{c_{i,k}}\} \right],$$

$$\overline{c_{i,k}} = \begin{cases} 0, & \text{if } p_w = p_k \\ (\text{some value}), & \text{otherwise} \end{cases}$$

average computation cost on all processors

$$rank_{PCM}(t_i) = \frac{\sum_{j=1}^{p} PCM(t_i, p_j)}{p}$$

- Processor selection phase:

Arrange task to that processor with the minimum LA_{EFT}

$$LA_{EFT}(t_i, p_j) = EFT(t_i, p_j) + PCM(t_i, p_j)$$

Disadvantage: a task with less immediate successor tasks may be scheduled before a task with more immediate successor tasks.

[2] Djigal, H., Feng, J., Lu, J.: Task scheduling for heterogeneous computing usinga predict cost matrix. In: Workshop Proceedings of the 48th International Con-ference on Parallel Processing. ICPP Workshops '19, Association for ComputingMachinery, New York, NY, USA (2019).

PREVIOUS WORK: AEFT [3]

computation cost of task I on processor j   average computation cost on all processors

- Task prioritizing phase:

Sorting task by decreasing order of rank$_{IOCT}$:

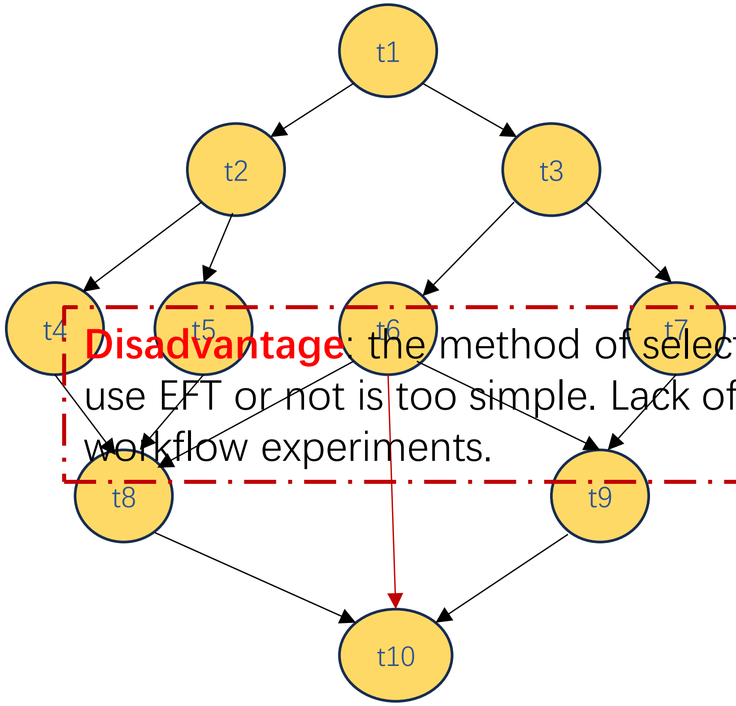
$$IOCT(v_i, p_j) = \omega(v_i, p_j) + \max_{v_j \in succ(v_i)} [\min \{IOCT(v_j, p_n) + \overline{c_{i,j}}\}],$$

$$\overline{c_{i,j}} = 0 \quad if \quad p_w = p_k$$

- Processor selection phase:

If $\boxed{\text{out-degree}}$ of the task is bigger than the number of processors, the task is assigned to the processor that yields the minimal EFT. If not, the task is assigned to the processor that yields the minimal I_{EFT} .

$$I_{EFT}(v_i, p_j) = EFT(v_i, p_j) + IOCT(v_i, p_j)$$



Disadvantage: the method of selecting whether use EFT or not is too simple. Lack of enough real workflow experiments.

Assume the number of processors is 2.

For task6: use EFT
Other tasks: use I_{EFT}

[3] Wang, M., Chen, J., Wang, H., Gao, Z., Bian, W., Qiao, S.: An en-hanced list scheduling algorithm for heterogeneous computing using anoptimized predictive cost matrix. Future Generation Computer Systems166, 107733 (2025).

03 OUR ALGORITHM

Algorithm 1 Pseudo Code of MPPTS Algorithm

Input: Workflow $G(T, E)$; a set Q of P heterogeneous processors;
Output: $S = \{t_i, p_j, EST(t_i, p_j), EFT(t_i, p_j)\}$
Compute $EPCM$ and $rank_{EPCM}$ for each task using (20) and (22) respectively;
Create an empty list $List_{EPCM}$ and put t_{entry} as initial task;
while $List_{EPCM}$ is not empty **do**
  Select t_i, the task with the highest $rank_{EPCM}$ value from $List_{EPCM}$;
  for all processors $p_j \in p$ **do**
    Compute $EFT(t_i, p_j)$ using insertion-based scheduling policy;
    Compute $CC_{EFT}(t_i, p_j) = EFT(t_i, p_j) + EPCM(t_i, p_j) + w(t_i, p_j)$;
  end for
  Assign task t_i to the processor p_j that minimizes CC_{EFT} for t_i;
  Update $List_{EPCM}$;
end while

EPCM value of successor tasks on processor γ

computation cost of successor task on processor γ

Enhanced Predict Cost Matrix (EPCM)

$$\boxed{\mathrm{EPCM}(t_i, p_j) = w(t_i, p_j) + \max_{t_k \in \mathrm{succ}(t_i)} \left[\min_{p_\gamma \in P} \{\mathrm{EPCM}(t_k, p_\gamma) + w(t_k, p_\gamma) + c_{i,k}\} \right]}$$

Computation and Communication-aware Earliest Finish Time

$$CC_{EFT}(t_i, p_j) = EFT(t_i, p_j) + EPCM(t_i, p_j) + w(t_i, p_j)$$

computation cost of current task on processor γ

01 EPCM

This helps better reflect real scientific workflow communication overhead, giving a more accurate estimate of makespan. It is very important in heavy communication workflows.

02 CC EFT

This provides a comprehensive scheduling score that reflects both the nearby and downstream implications of processor assignment, reduces the overdependence on future task predictions and ensures proper attention to the computation requirements of the current task.

04 EXPERIMENTS AND RESULTS

1. Randomly Generated Application Graphs

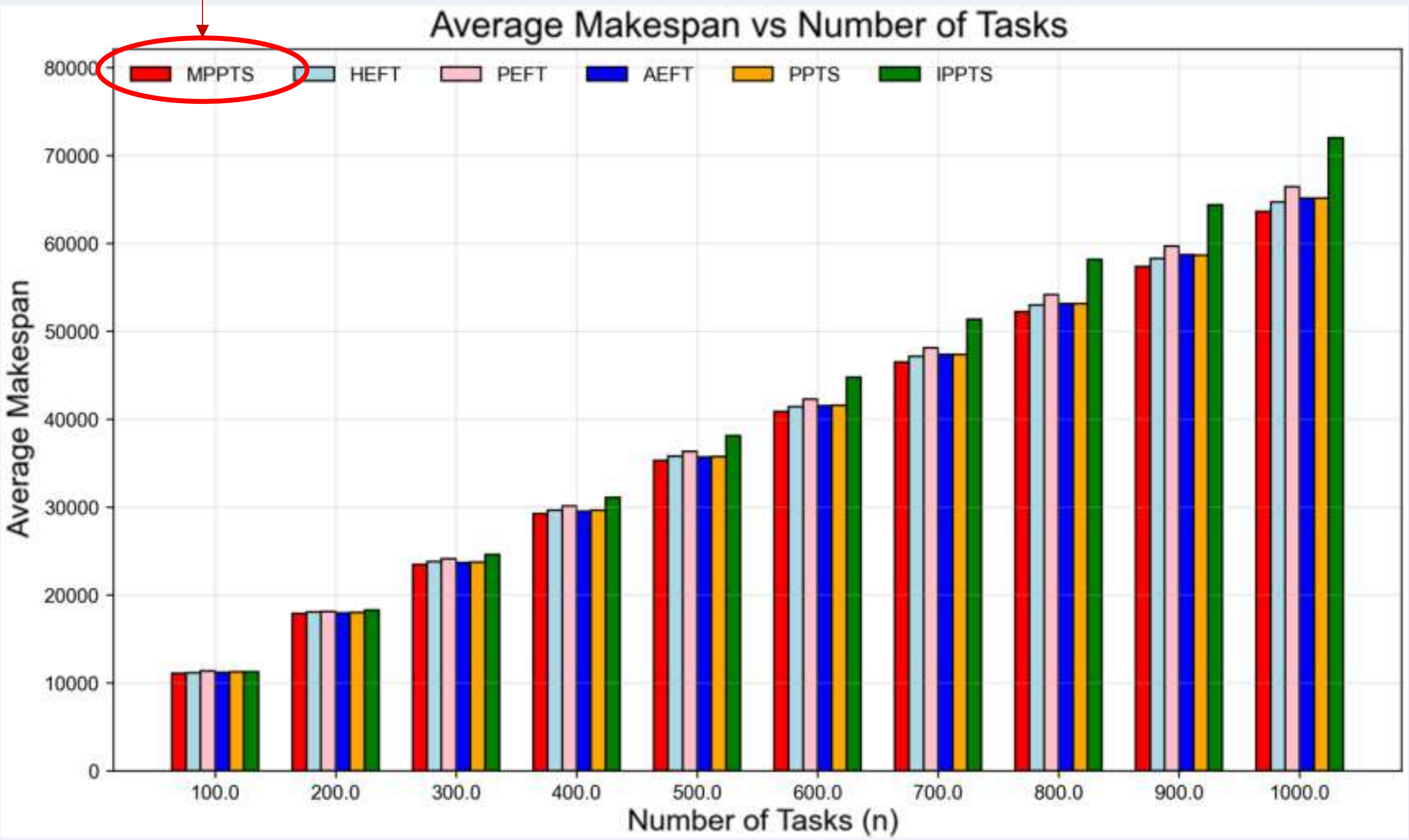
We set 450 kinds of different randomly generated graphs; each type includes 25 different graphs. Totally 11,250 graphs used.

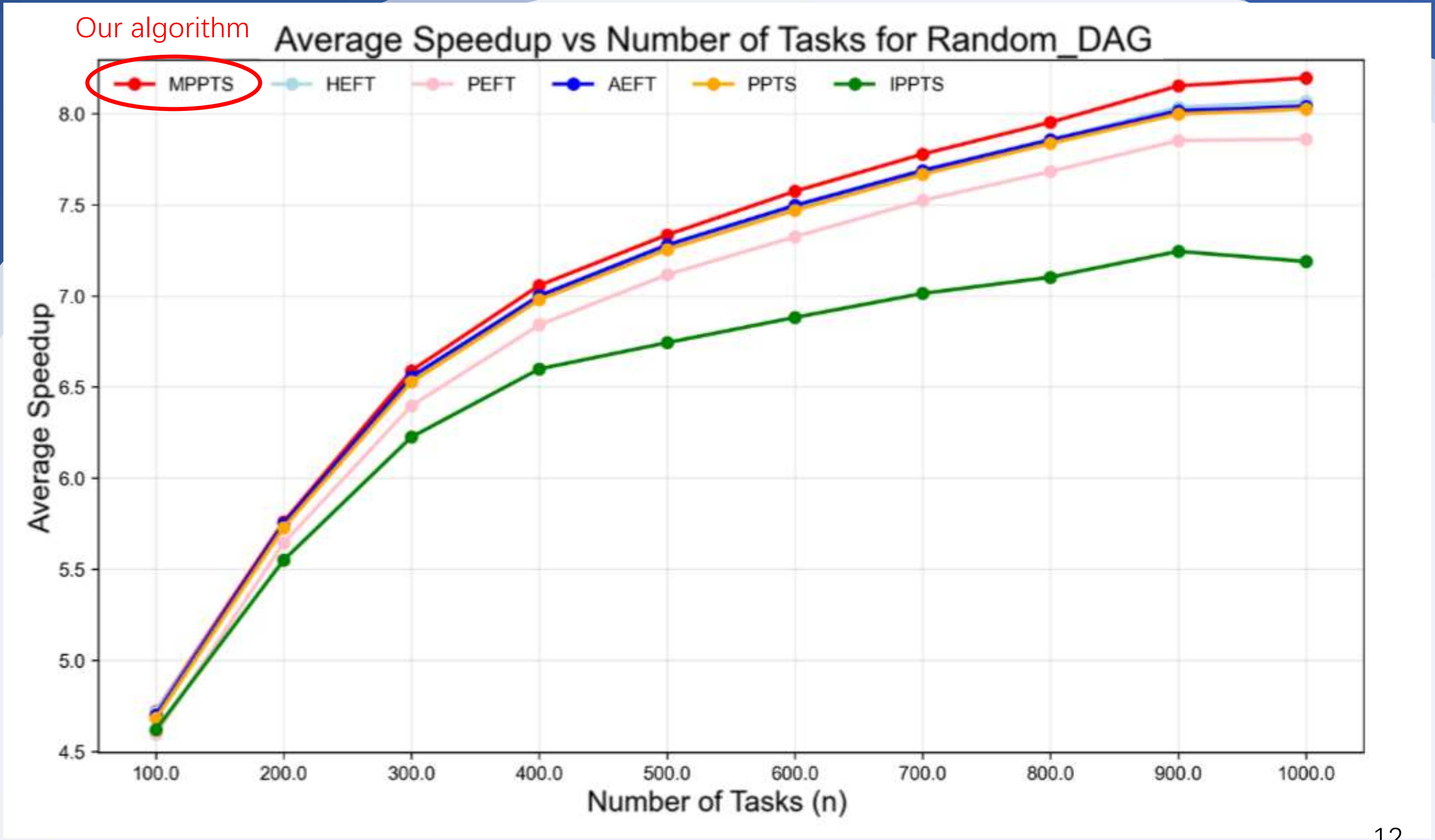
Parameters	Values
n	100, 200, 300, 400, 500, 600, 700, 800, 900, 1000
CCR	0.1, 0.5, 2, 5, 10
β	0.1, 0.2, 0.5
Processors	4, 8, 16
α	0.5, 1, 2
Out-degree	5, 6, 7, 8, 9, 10
Average computation cost of a task	Randomly obtained from 100 to 500

Controls the heterogeneity of processor speeds.

Indicates the closeness of interdependencies between tasks.

Our algorithm





Our algorithm
Average Speedup vs Number of Tasks for Random_DAG
MPPTS
HEFT
PEFT
AEFT
PPTS
IPPTS
Average Speedup
Number of Tasks (n)
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
100.0
200.0
300.0
400.0
500.0
600.0
700.0
800.0
900.0
1000.0

Our algorithm

-	Relation	MPPTS	HEFT	PEFT	AEFT	PPTS	IPPTS	combined
MPPTS	better	*	73.60%	77.70%	72.40%	74.40%	80.30%	75.70%
	equal		0.40%	0.00%	0.10%	0.10%	0.00%	0.10%
	worse		26.00%	22.20%	27.50%	25.50%	19.60%	24.20%
HEFT	better	26.00%	*	77.70%	53.10%	59.40%	80.60%	59.40%
	equal	0.40%		0.00%	0.10%	0.10%	0.00%	0.10%
	worse	73.60%		22.30%	46.80%	40.50%	19.40%	40.50%
PEFT	better	22.20%	22.30%	*	23.60%	21.40%	78.80%	33.70%
	equal	0.00%	0.00%		0.10%	0.20%	0.00%	0.10%
	worse	77.70%	77.70%		76.30%	78.50%	21.20%	66.30%
AEFT	better	27.50%	46.80%	76.30%	*	54.50%	82.40%	57.50%
	equal	0.10%	0.10%	0.10%		0.30%	0.00%	0.10%
	worse	72.40%	53.10%	23.60%		45.20%	17.50%	42.40%
PPTS	better	25.50%	40.50%	78.50%	45.20%	*	82.10%	54.40%
	equal	0.10%	0.10%	0.20%	0.30%		0.10%	0.20%
	worse	74.40%	59.40%	21.40%	54.50%		17.80%	45.50%
IPPTS	better	19.60%	19.40%	21.20%	17.50%	17.80%	*	19.10%
	equal	0.00%	0.00%	0.00%	0.00%	0.10%		0.00%
	worse	80.30%	80.60%	78.80%	82.40%	82.10%		80.80%

better

04 EXPERIMENTS AND RESULTS

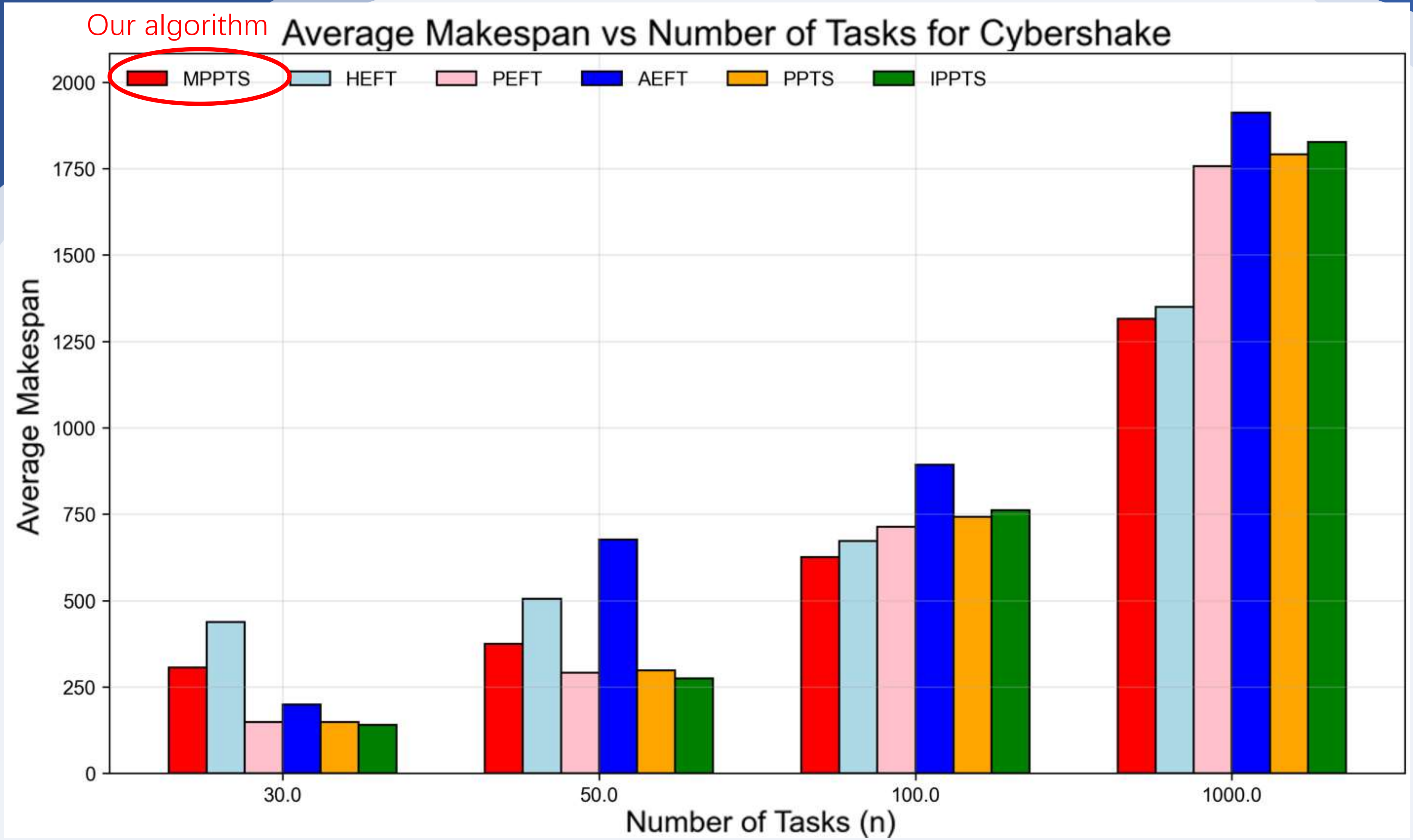
2. Real-World Application Graphs

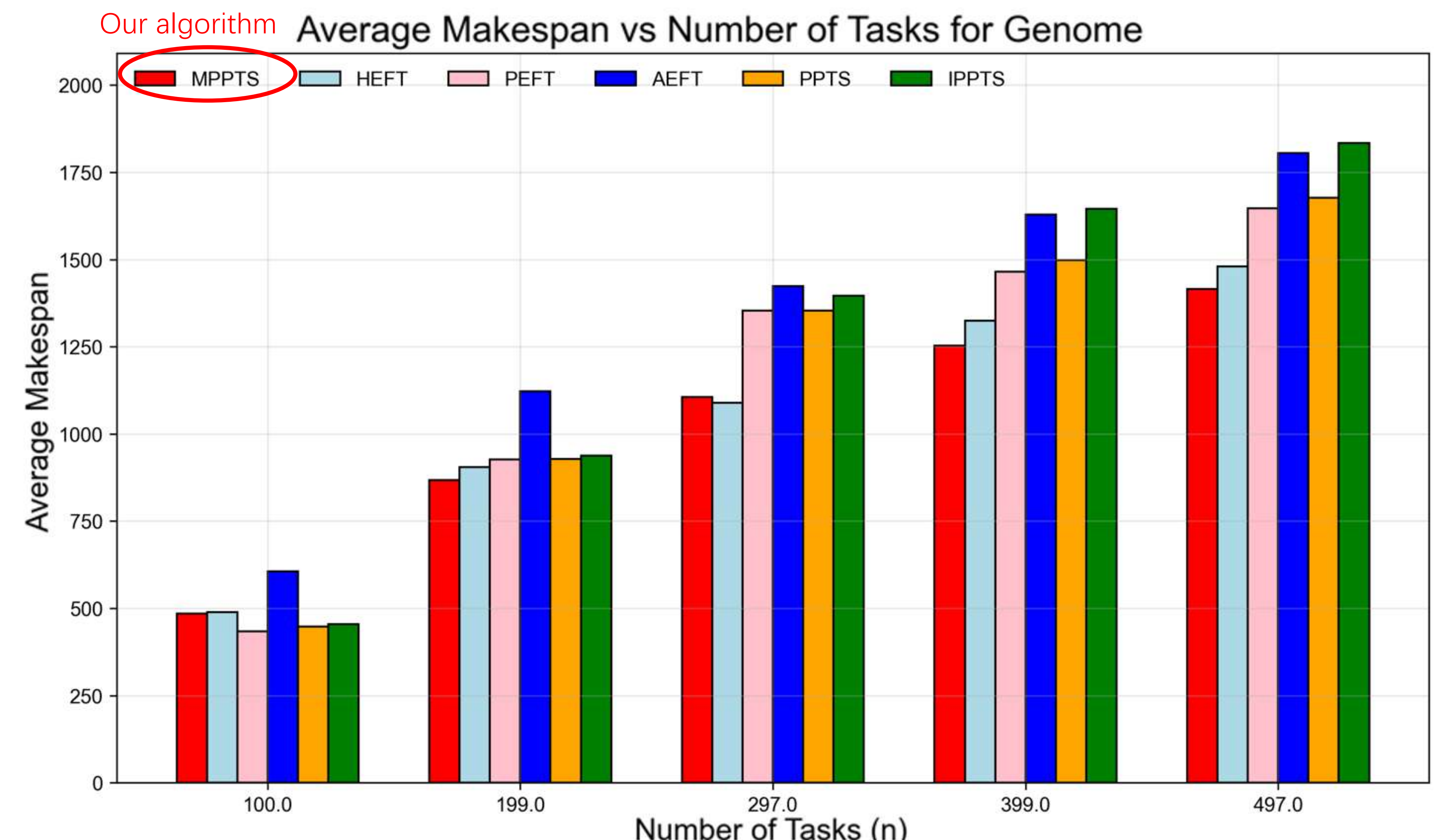
We did experiments with 4 kinds of different workflow graphs, totally 2700 graphs used.

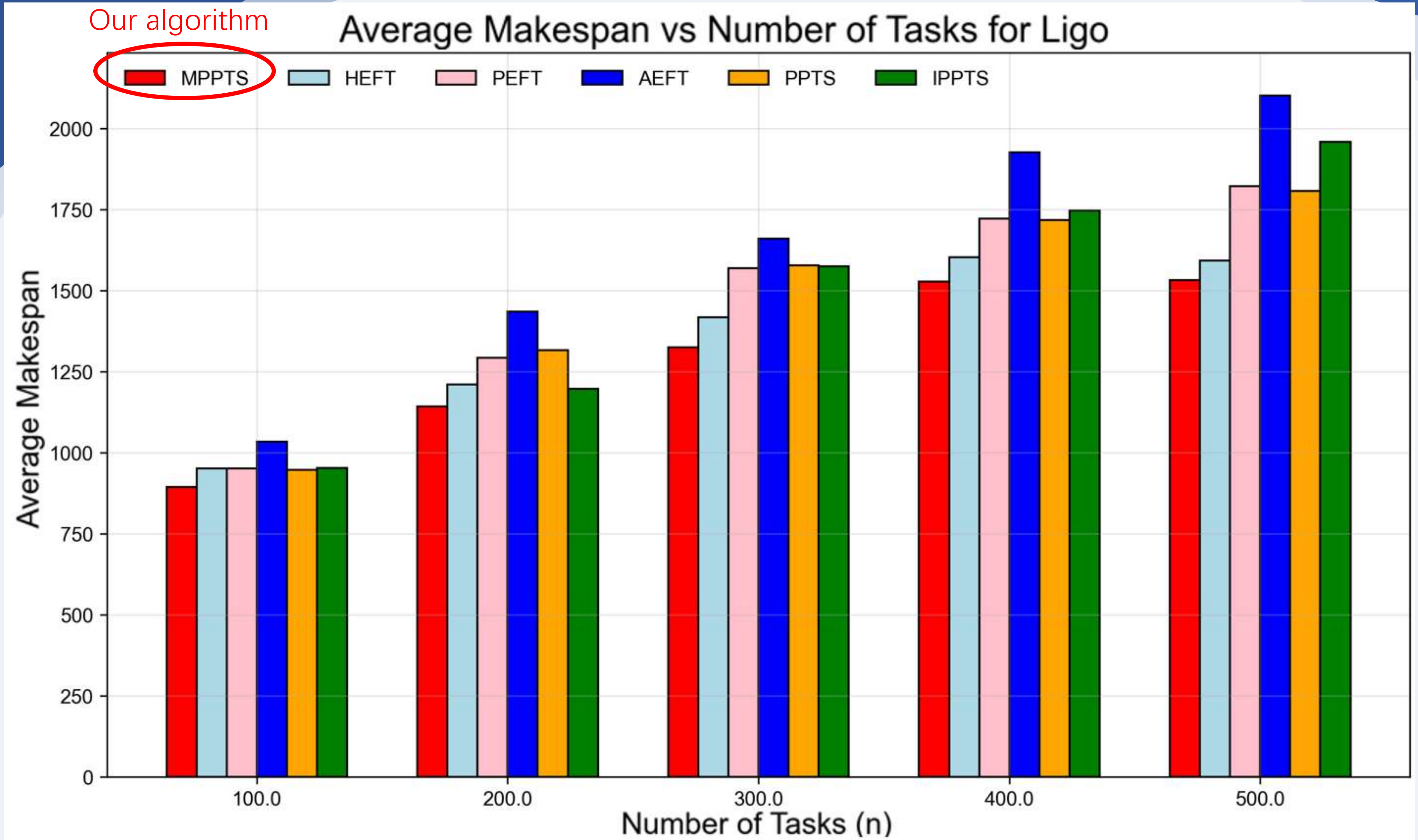
Parameters	Values			
	Cybershake	Genome	Ligo	Montage
n	30, 50, 100, 1000	100, 199, 297, 399, 497	100,200,300,400,500	25,50,100,1000
CCR	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5
β	0.1, 0.5, 0.9	0.1, 0.5, 0.9	0.1, 0.5, 0.9	0.1, 0.5, 0.9
Processors	8	8	8	8
Number of experiments	600	750	750	600

04 EXPERIMENTS AND RESULTS

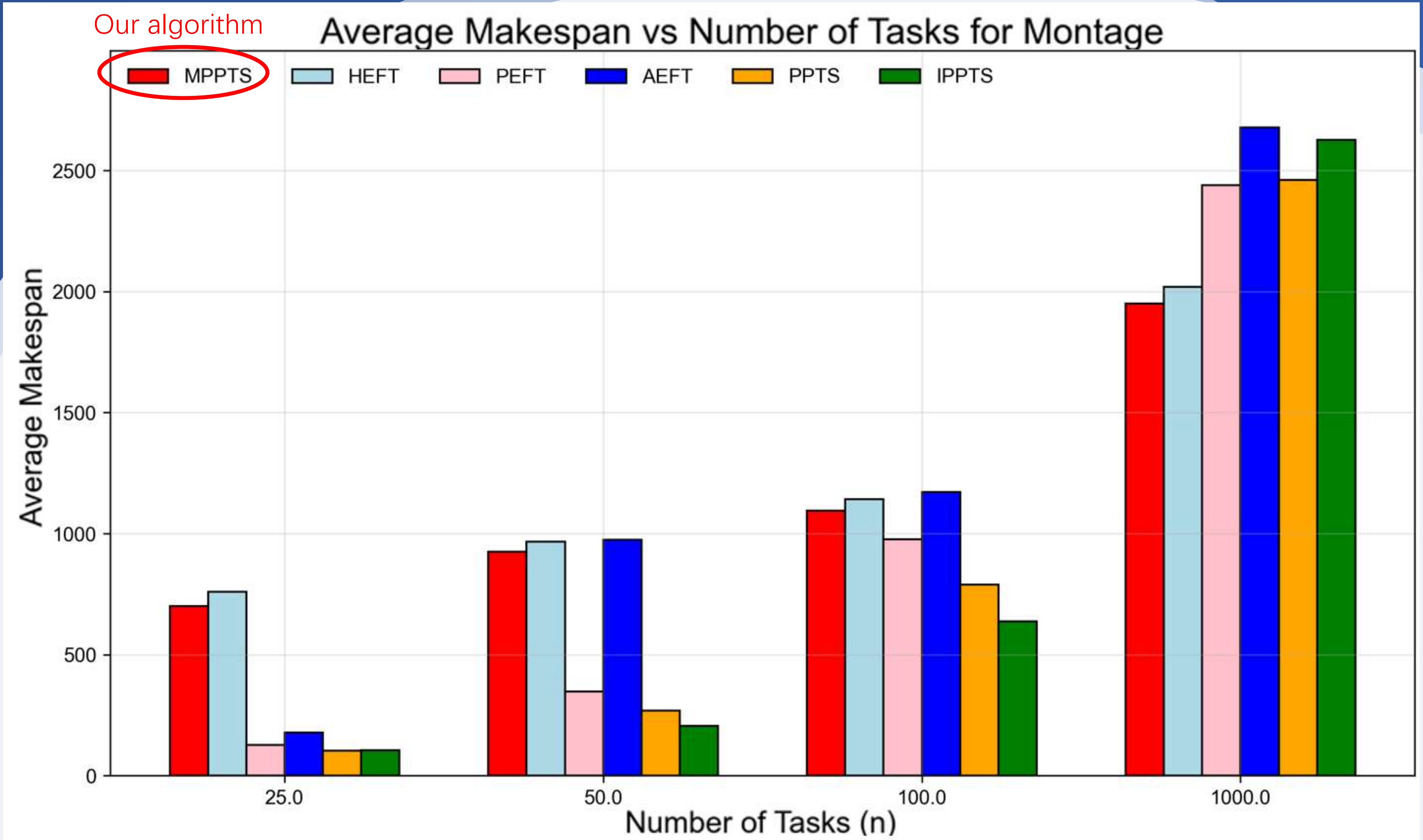
- CyberShake is used to process large earthquake datasets (the Green strain tensor) and supports the assessment and early warning of earthquake hazards.
- Genome is used in bioinformatics to process large-scale genome sequencing data and automate DNA sequence analysis.
- LIGO is a large-scale physics experiment and observatory designed to detect cosmic events that involve gravitational waves, such as the mergers of black holes and neutron stars.
- Montage is a widely used workflow application in the field of astronomy, specifically designed to assemble astronomical images into sky mosaics.


Our algorithm
Average Makespan vs Number of Tasks for Cybershake
MPPTS
HEFT
PEFT
AEFT
PPTS
IPPTS
Average Makespan
2000
1750
1500
1250
1000
750
500
250
0
30.0
50.0
100.0
1000.0
Number of Tasks (n)


Our algorithm
Average Makespan vs Number of Tasks for Genome
MPPTS
HEFT
PEFT
AEFT
PPTS
IPPTS
Average Makespan
2000
1750
1500
1250
1000
750
500
250
0
100.0
199.0
297.0
399.0
497.0
Number of Tasks (n)



Our algorithm
Average Makespan vs Number of Tasks for Ligo
MPPTS
HEFT
PEFT
AEFT
PPTS
IPPTS
Average Makespan
2000
1750
1500
1250
1000
750
500
250
0
100.0
200.0
300.0
400.0
500.0
Number of Tasks (n)



Our algorithm
Average Makespan vs Number of Tasks for Montage
MPPTS
HEFT
PEFT
AEFT
PPTS
IPPTS
Average Makespan
0
500
1000
1500
2000
2500
25.0
50.0
100.0
1000.0
Number of Tasks (n)

Our algorithm

better

Cybershake		MPPTS	HEFT	PEFT	AEFT	PPTS	IPPTS	combined
	better		50.40%	59.40%	79.70%	59.00%	60.80%	61.90%
MPPTS	equal	*	30.00%	14.80%	10.20%	15.20%	15.30%	17.10%
	worse		19.60%	25.80%	10.10%	25.90%	23.90%	21.10%
	better	19.60%		54.60%	77.20%	54.90%	57.20%	52.70%
HEFT	equal	30.00%	*	7.00%	5.20%	7.00%	7.80%	11.40%
	worse	50.40%		38.40%	17.60%	38.20%	35.00%	35.90%
	better	25.80%	38.40%		72.30%	46.90%	54.80%	47.60%
PEFT	equal	14.80%	7.00%	*	14.90%	22.50%	24.20%	16.70%
	worse	59.40%	54.60%		12.80%	30.60%	21.00%	35.70%
	better	10.10%	17.60%	12.80%		12.60%	20.90%	14.80%
AEFT	equal	10.20%	5.20%	14.90%	*	17.30%	16.10%	12.70%
	worse	79.70%	77.20%	72.30%		70.20%	63.10%	72.50%
	better	25.90%	38.20%	30.60%	70.20%		39.90%	41.00%
PPTS	equal	15.20%	7.00%	22.50%	17.30%	*	39.10%	20.20%
	worse	59.00%	54.90%	46.90%	12.60%		21.00%	38.90%
	better	23.90%	35.00%	21.00%	63.10%	21.00%		32.80%
IPPTS	equal	15.30%	7.80%	24.20%	16.10%	39.10%	*	20.50%
	worse	60.80%	57.20%	54.80%	20.90%	39.90%		46.70%

Our algorithm

Genome		MPPTS	HEFT	PEFT	AEFT	PPTS	IPPTS	combined
MPPTS	better	*	51.90%	69.20%	87.00%	73.70%	79.60%	72.30%
	equal		14.30%	1.70%	1.20%	2.10%	1.80%	4.20%
	worse		33.80%	29.00%	11.80%	24.20%	18.60%	23.50%
HEFT	better	33.80%	*	66.70%	85.20%	68.30%	77.10%	66.20%
	equal	14.30%		1.60%	1.10%	1.80%	1.70%	4.10%
	worse	51.90%		31.70%	13.70%	29.90%	21.30%	29.70%
PEFT	better	29.00%	31.70%	*	76.20%	52.00%	69.40%	51.70%
	equal	1.70%	1.60%		3.30%	5.80%	5.30%	3.50%
	worse	69.20%	66.70%		20.60%	42.20%	25.40%	44.80%
AEFT	better	11.80%	13.70%	20.60%	*	21.60%	45.20%	22.60%
	equal	1.20%	1.10%	3.30%		4.70%	4.40%	2.90%
	worse	87.00%	85.20%	76.20%		73.70%	50.40%	74.50%
PPTS	better	24.20%	29.90%	42.20%	73.70%	*	58.20%	45.60%
	equal	2.10%	1.80%	5.80%	4.70%		27.00%	8.30%
	worse	73.70%	68.30%	52.00%	21.60%		14.80%	46.10%
IPPTS	better	18.60%	21.30%	25.40%	50.40%	14.80%	*	26.10%
	equal	1.80%	1.70%	5.30%	4.40%	27.00%		8.00%
	worse	79.60%	77.10%	69.40%	45.20%	58.20%		65.90%

better

Our algorithm

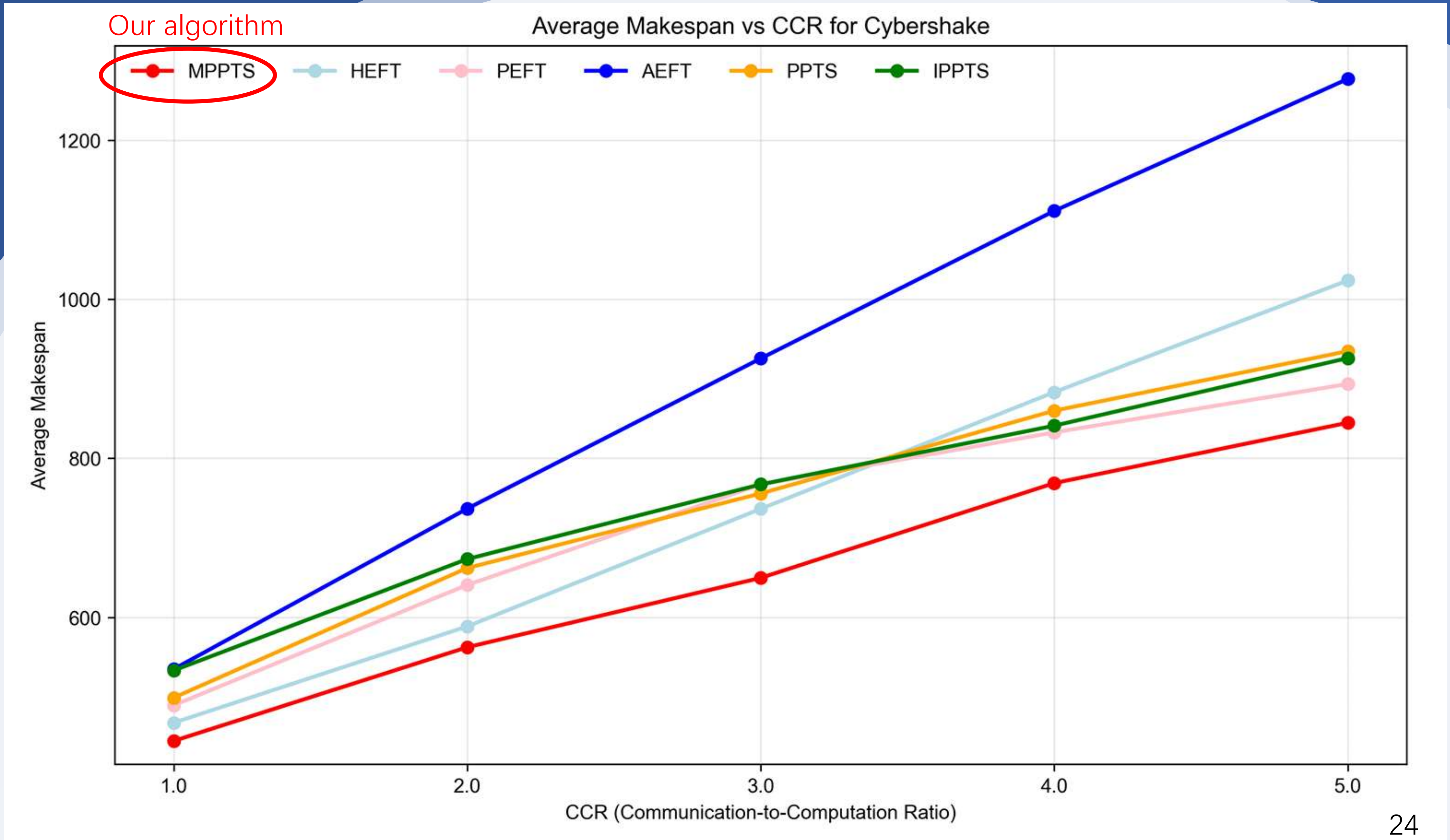
Ligo		MPPTS	HEFT	PEFT	AEFT	PPTS	IPPTS	combined
	better		64.70%	85.40%	91.60%	83.50%	80.50%	81.10%
MPPTS	equal	*	0.30%	0.00%	0.00%	0.00%	0.00%	0.10%
	worse		35.00%	14.60%	8.40%	16.50%	19.50%	18.80%
	better	35.00%		73.50%	83.00%	73.20%	72.00%	67.30%
HEFT	equal	0.30%	*	0.00%	0.00%	0.00%	0.00%	0.10%
	worse	64.70%		26.50%	17.00%	26.80%	28.00%	32.60%
	better	14.60%	26.50%		77.60%	48.40%	57.20%	44.90%
PEFT	equal	0.00%	0.00%	*	0.00%	0.10%	0.00%	0.00%
	worse	85.40%	73.50%		22.40%	51.40%	42.80%	55.10%
	better	8.40%	17.00%	22.40%		21.40%	33.60%	20.60%
AEFT	equal	0.00%	0.00%	0.00%	*	0.10%	0.00%	0.00%
	worse	91.60%	83.00%	77.60%		78.40%	66.40%	79.40%
	better	16.50%	26.80%	51.40%	78.40%		59.80%	46.60%
PPTS	equal	0.00%	0.00%	0.10%	0.10%	*	0.10%	0.10%
	worse	83.50%	73.20%	48.40%	21.40%		40.10%	53.30%
	better	19.50%	28.00%	42.80%	66.40%	40.10%		39.40%
IPPTS	equal	0.00%	0.00%	0.00%	0.00%	0.10%	*	0.00%
	worse	80.50%	72.00%	57.20%	33.60%	59.80%		60.60%

better

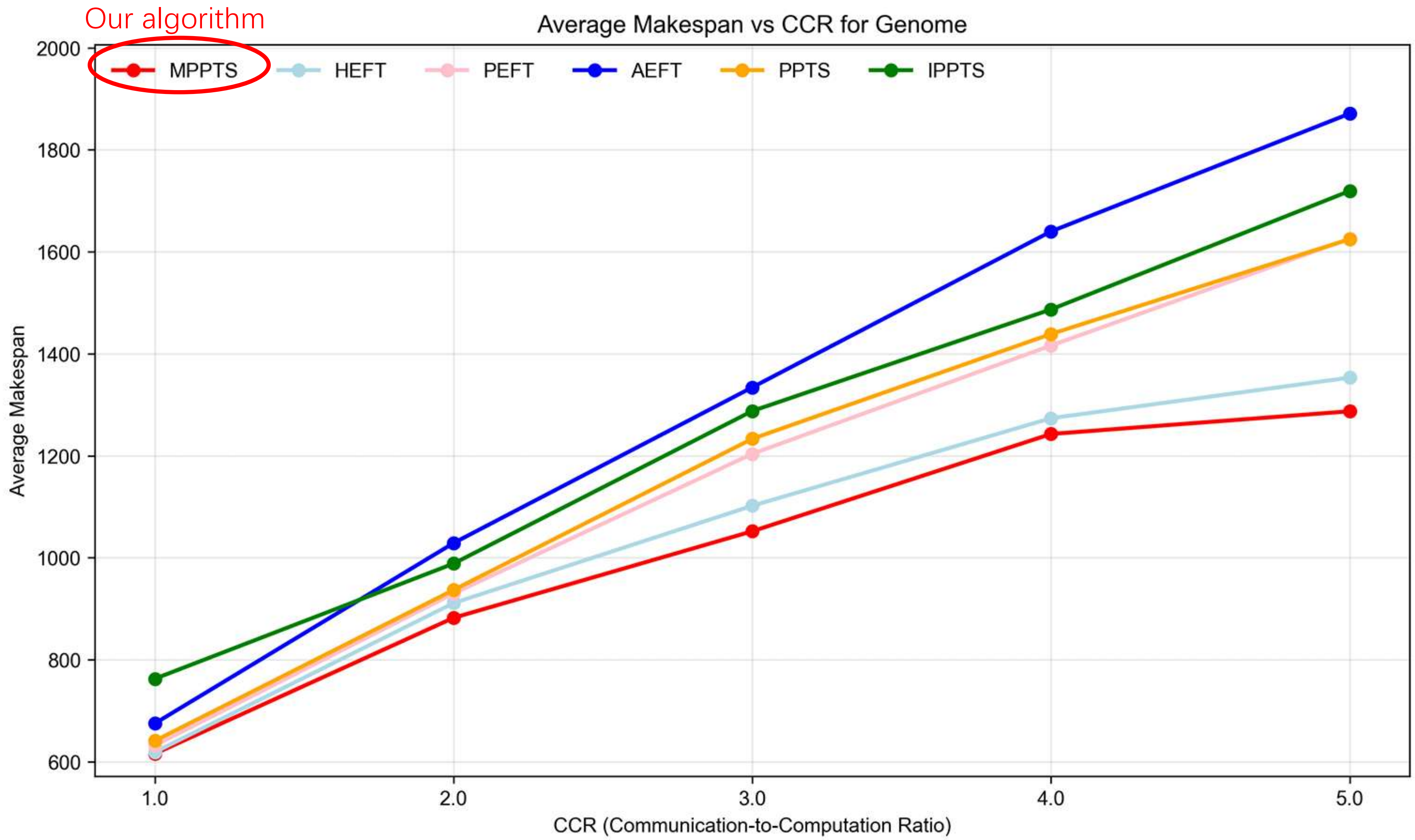
Our algorithm

Montage		MPPTS	HEFT	PEFT	AEFT	PPTS	IPPTS	combined
MPPTS	better		64.10%	34.40%	59.80%	29.10%	29.70%	43.40%
	equal	*	15.60%	0.30%	0.80%	0.50%	0.30%	3.50%
	worse		20.40%	65.20%	39.40%	70.40%	69.90%	53.10%
HEFT	better	20.40%		31.10%	53.40%	27.60%	27.90%	32.10%
	equal	15.60%	*	0.00%	0.20%	0.00%	0.00%	3.20%
	worse	64.10%		68.90%	46.40%	72.40%	72.10%	64.80%
PEFT	better	65.20%	68.90%		77.20%	26.30%	34.10%	54.30%
	equal	0.30%	0.00%	*	14.40%	29.40%	29.00%	14.60%
	worse	34.40%	31.10%		8.40%	44.30%	36.90%	31.00%
AEFT	better	39.40%	46.40%	8.40%		4.90%	14.70%	22.80%
	equal	0.80%	0.20%	14.40%	*	17.40%	15.70%	9.70%
	worse	59.80%	53.40%	77.20%		77.70%	69.60%	67.50%
PPTS	better	70.40%	72.40%	44.30%	77.70%		37.40%	60.40%
	equal	0.50%	0.00%	29.40%	17.40%	*	35.20%	16.50%
	worse	29.10%	27.60%	26.30%	4.90%		27.50%	23.10%
IPPTS	better	69.90%	72.10%	36.90%	69.60%	27.50%		55.20%
	equal	0.30%	0.00%	29.00%	15.70%	35.20%	*	16.00%
	worse	29.70%	27.90%	34.10%	14.70%	37.40%		28.80%

better



Our algorithm
Average Makespan vs CCR for Cybershake
MPPTS
HEFT
PEFT
AEFT
PPTS
IPPTS
Average Makespan
1200
1000
800
600
1.0
2.0
3.0
4.0
5.0
CCR (Communication-to-Computation Ratio)



Our algorithm
Average Makespan vs CCR for Genome
MPPTS
HEFT
PEFT
AEFT
PPTS
IPPTS
Average Makespan
2000
1800
1600
1400
1200
1000
800
600
1.0
2.0
3.0
4.0
5.0
CCR (Communication-to-Computation Ratio)

05 CONCLUSION

In this study, we present a novel scheduling algorithm, MPPTS, designed to solve the limitations of existing heuristic algorithms.

Experimental results in both randomly generated and real-world scientific workflows (CyberShake, Genome, LIGO, and Montage) demonstrate that MPPTS achieves a lower makespan, higher frequency of best scheduling results, and higher processor efficiency. MPPTS outperforms in most real-world scientific workflows, especially for large-scale tasks.

05 FUTURE WORK

We will explore broader applications using additional real-world workflows and investigate the integration of machine learning techniques to further improve adaptive scheduling capabilities.

In particular, we will conduct comprehensive evaluations on practical distributed and cloud platforms to validate their robustness and effectiveness in realistic environments.

REFERENCE

[1] H. Topcuoglu, S. Hariri, and M.-Y. Wu, “Performance-effective and low-complexity task scheduling for heterogeneous computing,” IEEE Trans. Parallel Distrib. Syst., vol. 13, no. 3, pp. 260–274, 2002. cited By 1626.

[2] Djigal, H., Feng, J., Lu, J.: Task scheduling for heterogeneous computing using a predict cost matrix. In: Workshop Proceedings of the 48th International Conference on Parallel Processing. ICPP Workshops ’19, Association for Computing Machinery, New York, NY, USA (2019).

[3] Wang, M., Chen, J., Wang, H., Gao, Z., Bian, W., Qiao, S.: An enhanced list scheduling algorithm for heterogeneous computing using an optimized predictive cost matrix. Future Generation Computer Systems166, 107733 (2025).