Joint Prediction and Matching for Computing Resource Exchange Platforms

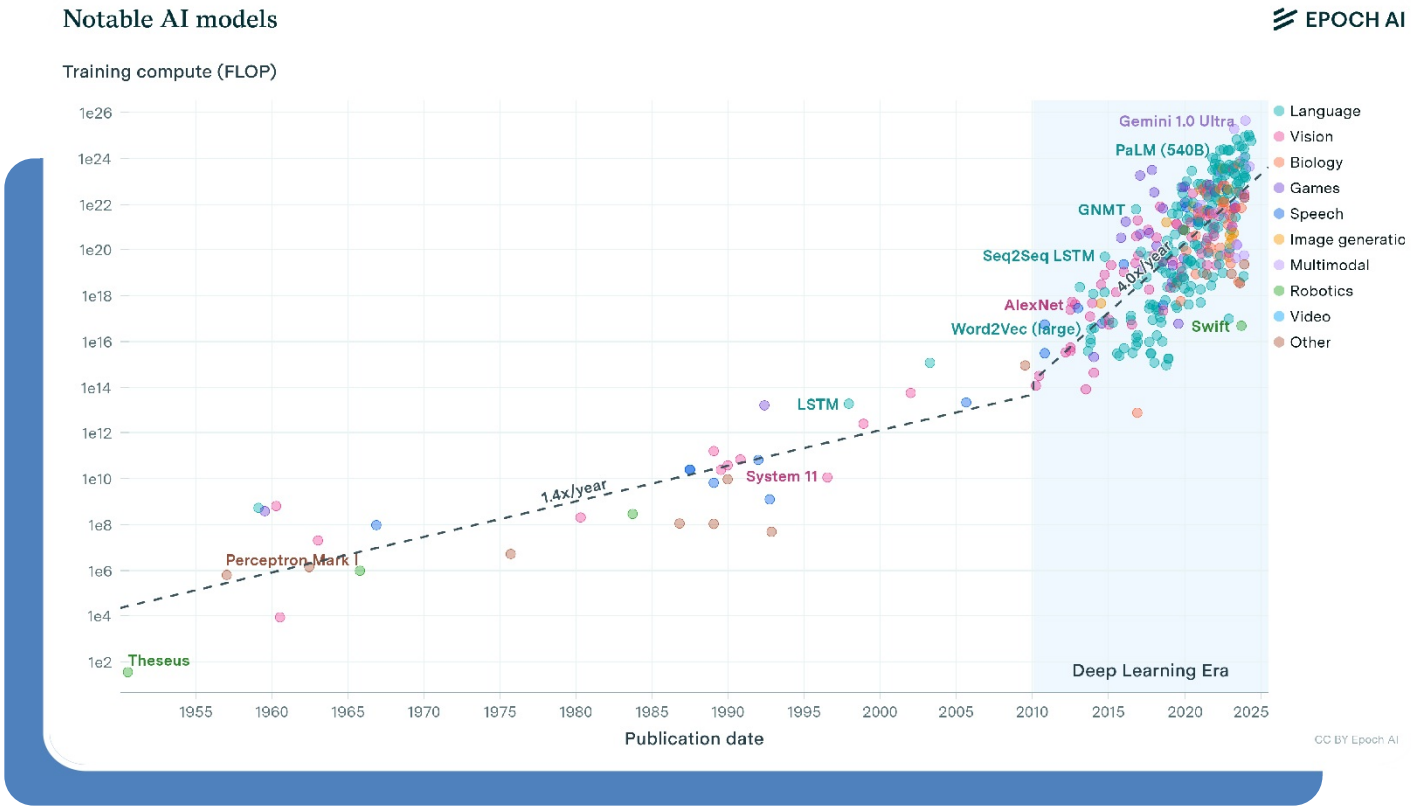
Da Huo[1], Zhenzhe Zheng[1], Xiaoyao Huang[2], Hao Chen[3],
Jianfeng Hu[3], Zhiyong Yan[3], Fan Wu[1], Jie Wu[2]

[1]Shanghai Jiao Tong University
[2]Cloud Computing Research Institute, China Telecom
[3]China Telecom Cloud Technology Co.Ltd., Beijing 100033

2025-9-10

Contents

Background: AI Computing Demand



Exponential Growth in AI Compute Needs.

[1]J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn and P. Villalobos, “Compute Trends Across Three Eras of Machine Learning,” 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 2022, pp. 1-8, doi: 10.1109/IJCNN55064.2022.9891914.

Background: Insufficient Computing Resources

Limitations of Centralized Cloud Services

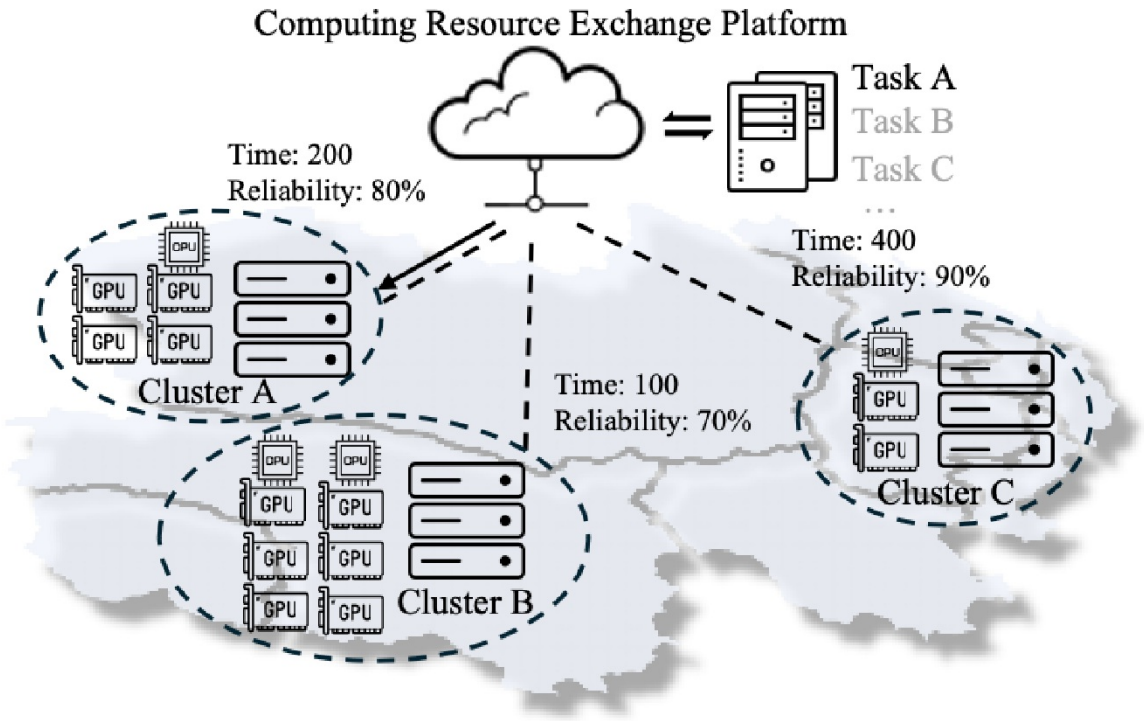
Underutilization of Enterprise Computing Resources

The Difficulty of Resource Sharing Without Platforms

Computing Resource Exchange Platforms

Pipeline:

- **Cluster performance prediction:** Computing tasks have varying preferences for different computing clusters, such as time and reliability.
- **Cluster-task matching:** Based on preference predictions, the platform allocates tasks to the optimal computing cluster.



Problem Definition

This corresponds to the following two optimization problems.

- **Cluster performance prediction:** Minimize the MSE loss between the predicted and true values on the set of tasks $\mathcal{Z}$.

$$\min_{\omega} \mathbb{E}_{\mathbf{z}\in\mathcal{Z}}[\|\mathbf{t} - m_{\omega}(\mathbf{z})\|_2^2], \qquad \min_{\phi} \mathbb{E}_{\mathbf{z}\in\mathcal{Z}}[\|\mathbf{a} - m_{\phi}(\mathbf{z})\|_2^2].$$

- **Cluster-task matching:** Solve the optimal bipartite matching problem to minimize the total execution time under the reliability constraint.

$$\begin{aligned} &\min_{\mathbf{X}} f(\mathbf{X}, \mathbf{T}), \\ &\text{s.t. } g(\mathbf{X}, \mathbf{A}) \geq 0, \\ &\sum_{i=1}^{M} \mathbf{x}_i - \mathbf{1}_N = 0, \\ &x_{ij} \in \{0, 1\}, \quad \forall i \in \mathcal{M}, \forall j \in \mathcal{N}, \end{aligned}$$

$$f(\mathbf{X}, \mathbf{T}) = \max_{i\in\{1,2,\cdots,M\}} \mathbf{x}_i^{\top} \mathbf{t}_i.$$

$$g(\mathbf{X}, \mathbf{A}) = \frac{1}{MN} \sum_{i=1}^{M} \mathbf{x}_i^{\top} \mathbf{a}_i - \gamma.$$

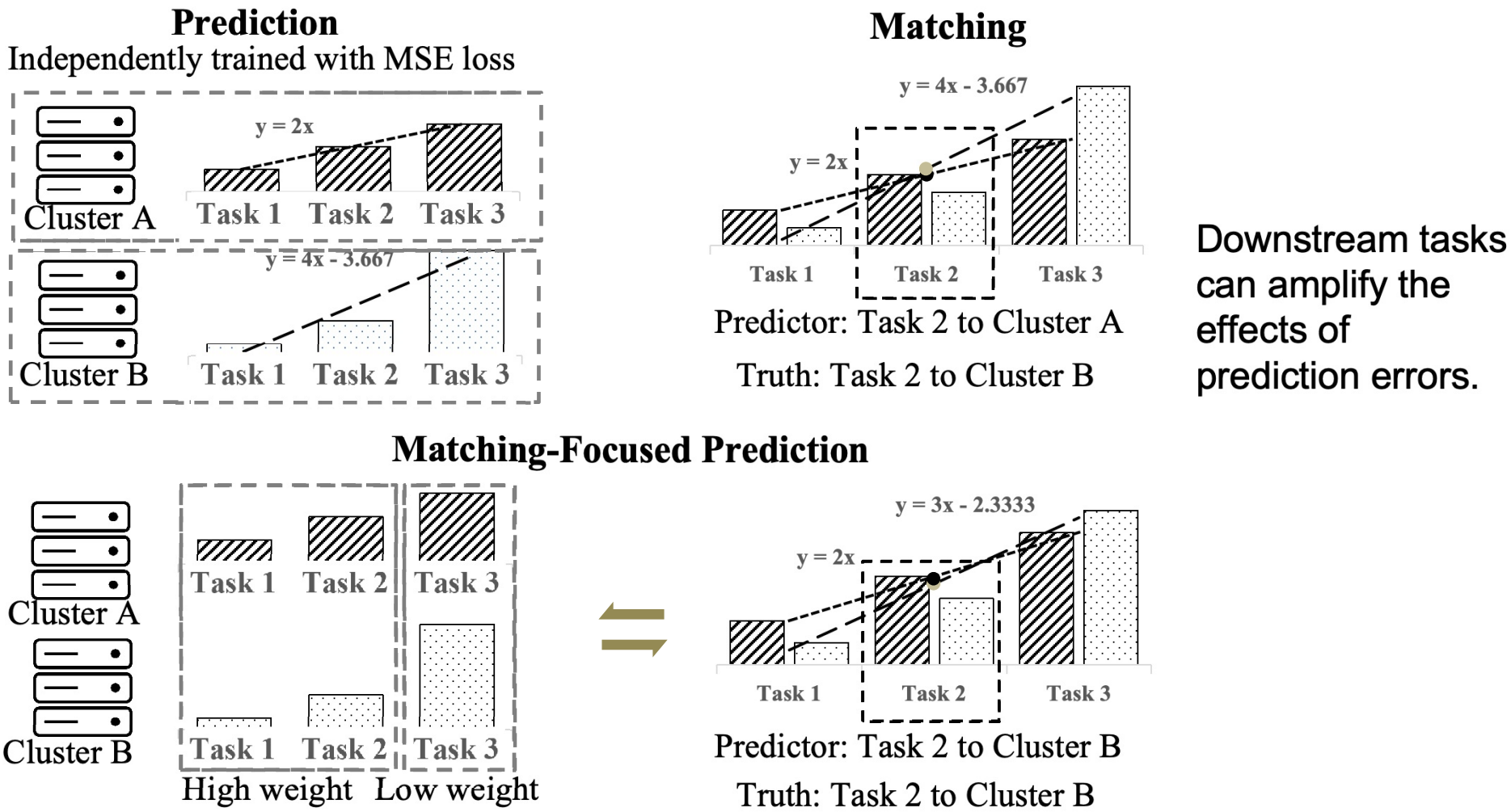
Predict-then-Matching Limitations

The objectives of the two optimization problems in the Predict-then-Matching framework are not aligned.

- **Independent predictors:** Optimized for MSE.

- **Misalignment with downstream matching:** Wrong decisions even with small prediction errors.

Predict-then-Matching Limitations: An Example



Our Solution: Joint Prediction and Matching

We integrate prediction and matching into a bi-level optimization problem.

- **The upper-level:** Select the optimal parameters to minimize the distance between the matching solution based on predicted values and the matching solution based on true values, which we refer to as regret.

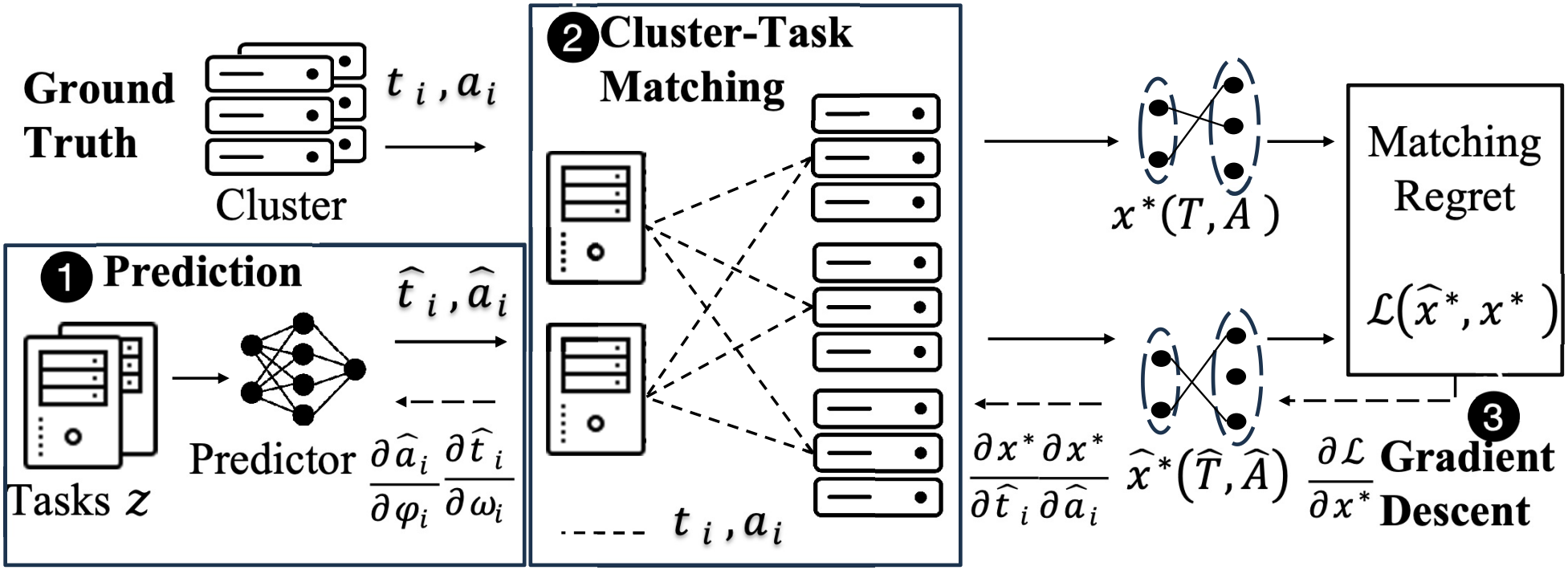
- **The lower-level:** Select the optimal matching decision that minimizes the cost function within the feasible domain under the reliability constraint.

$$\min_{\boldsymbol{\omega},\boldsymbol{\phi}} \frac{1}{N}\left(f\left(\mathbf{X}^*\left(\hat{\mathbf{T}},\hat{\mathbf{A}}\right),\mathbf{T}\right)-f\left(\mathbf{X}^*\left(\mathbf{T},\mathbf{A}\right),\mathbf{T}\right)\right),$$

$$\text{s.t. } \mathbf{X}^*\left(\mathbf{T},\mathbf{A}\right)=\underset{\mathbf{X}\in\mathcal{F}_{\mathbf{A}}}{\operatorname{argmin}} f\left(\mathbf{X},\mathbf{T}\right).$$

MFCP: System Overview

Matching-Focused Cluster Performance Predictor.



MFCP: System Overview

Matching-Focused Cluster Performance Predictor.

- **Forward:** Prediction -> Matching -> Computing regret.
- **Backward:** Propagate gradients through relaxed optimization.
- **Key challenge:** Backpropagate the gradient of the second term.

$$\frac{d\mathcal{L}}{d\omega_i} = \frac{d\mathcal{L}}{dX^*(\hat{T}, \hat{A})} \frac{dX^*(\hat{T}, \hat{A})}{d\hat{t}_i} \frac{d\hat{t}_i}{d\omega_i}.$$

Chain rule decomposition of the gradient.

Challenge 1: Non-differentiability

The inherent non-differentiability of the optimization problem.

- Matching is discrete.
- Execution time $f(\mathbf{X}, \mathbf{T})$ uses **max**, which is non-smooth.
- Reliability constraints yields zero or infinite gradient.

Solution to Challenge 1

- Continuous relaxation (binary -> [0,1]).
- Smooth approximation of max (log-sum-exp).

$$\tilde{f}(\mathbf{X}, \mathbf{T}) = \frac{1}{\beta} \log \left(\sum_{i=1}^{M} e^{\beta \mathbf{x}_i^\top \mathbf{t}_i} \right).$$

- Interior-point method for reliability constraints.

$$F(\mathbf{X}, \mathbf{T}, \mathbf{A}) = \tilde{f}(\mathbf{X}, \mathbf{T}) - \lambda \log(g(\mathbf{X}, \mathbf{A})),$$

$$\min_{\mathbf{X}} F(\mathbf{X}, \mathbf{T}, \mathbf{A}),$$

$$\text{s.t.} \sum_{i=1}^{M} \mathbf{x}_i - \mathbf{1}_N = 0, \quad \mathbf{x}_i \in [0, 1].$$

Challenge 2: Gradient Computation

As the matching optimization is convex.

- The Lagrangian of the optimization is:

$$L(\mathbf{X}, \boldsymbol{\nu}, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2) = F(\mathbf{X}, \mathbf{T}, \mathbf{A}) + \boldsymbol{\nu}^\top \left(\sum_{i=1}^{M} \mathbf{x}_i - \mathbf{1}_N \right) + \sum_{i=1}^{M} \sum_{j=1}^{N} \mu_{ij}^1 x_{ij} + \sum_{i=1}^{M} \sum_{j=1}^{N} \mu_{ij}^2 (1 - x_{ij}),$$

- The optimal matching must satisfies the following KKT conditions.

$$\Phi(\mathbf{X}, \mathbf{T}, \mathbf{A}, \boldsymbol{\nu}, \boldsymbol{\mu}^1, \boldsymbol{\mu}^2) = \begin{bmatrix} \nabla_{\mathbf{X}} L \\ \sum_{i=1}^{M} \mathbf{x}_i - \mathbf{1}_N \\ \boldsymbol{\mu}^1 \odot \mathbf{X} \\ \boldsymbol{\mu}^2 \odot (\mathbf{1} - \mathbf{X}) \end{bmatrix} = 0.$$

Challenge 2: Gradient Computation

As the matching optimization is convex.

- Then we can obtain the analytical gradients by taking the total derivative.

$$\begin{bmatrix} \nabla^2_{\mathrm{XX}} F & \mathbf{D}_{\boldsymbol{\nu}}^T & \mathbf{I} & -\mathbf{I} \\ \mathbf{D}_{\boldsymbol{\nu}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{U}_1 & \mathbf{0} & \mathbf{X}_d & \mathbf{0} \\ -\mathbf{U}_2 & \mathbf{0} & \mathbf{0} & \bar{\mathbf{X}}_d \end{bmatrix} \begin{bmatrix} d\mathbf{X} \\ d\boldsymbol{\nu} \\ d\boldsymbol{\mu}^1 \\ d\boldsymbol{\mu}^2 \end{bmatrix} = - \begin{bmatrix} \nabla^2_{\mathrm{XT}} F d\mathbf{T} + \nabla^2_{\mathrm{XA}} F d\mathbf{A} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

Extension: Non-Convex Matching

Considering scheduling algorithms, the execution time often does not scale linearly with the computational workload.

- Considering that tasks assigned to the same cluster share resources, faster execution can be achieved through internal cluster scheduling.
- For example, after parallel execution of two tasks with costs t_1 and t_2, the total cost becomes $\zeta(t_1 + t_2)$, where the speedup ratio $0 < \zeta < 1$.
- Without loss of generality, we use the function $\zeta(\boldsymbol{x} \cdot \mathbf{1}_N)$ to represent this speedup ratio, focusing primarily on the impact of the workload on this ratio.

Extension: Gradient Computation

Estimate the gradient of the matching algorithm via zero-order methods.

Algorithm 1: Optimal Matching by Gradient Descent

Input: Objective function $F(\mathbf{X}, \mathbf{T}, \mathbf{A})$, execution time martix $\mathbf{T}$ and reliablity martix $\mathbf{A}$.

Output: The optimal matching $\mathbf{X}^*$.

Initialize $\mathbf{X}$;

while *Iter* < *Epochs* **do**

  $\mathbf{X} \leftarrow \mathbf{X} - \eta \nabla_x F(\mathbf{X}, \mathbf{T}, \mathbf{A})$;

  $\mathbf{X}(:, j) \leftarrow \text{softmax}\,(\mathbf{X}(:, j))$ for $1 \leq j \leq N$;

return $\mathbf{X}^* \leftarrow \mathbf{X}$.

Experimental Setup

- Dataset: Xirang platform logs.
- Models: GNN for task features, predictors with FC layers.
- Metrics: Regret, Reliability, Utilization.
- Baselines: TAM, TSM, UCB, MFCP-AD, MFCP-FG.

Ablation Study

- Linear vs. max loss.
- Hard vs. soft penalty.
- Gradient computation vs. zeroth-order estimation.

Table 1: The Ablation study of MFCP.

Metric	Regret	Reliability	Utilization
(1)	1.555 ± 0.030	0.878 ± 0.005	0.387 ± 0.002
(2)	1.274 ± 0.035	0.681 ± 0.015	0.430 ± 0.008
(3)	0.937 ± 0.062	0.890 ± 0.0011	0.487 ± 0.007
MFCP	$\mathbf{0.894} \pm 0.035$	$\mathbf{0.886} \pm 0.006$	$\mathbf{0.488} \pm 0.003$

Overall Performance

- Results across different cluster combination settings.
- MFCP achieves lowest regret, higher utilization.

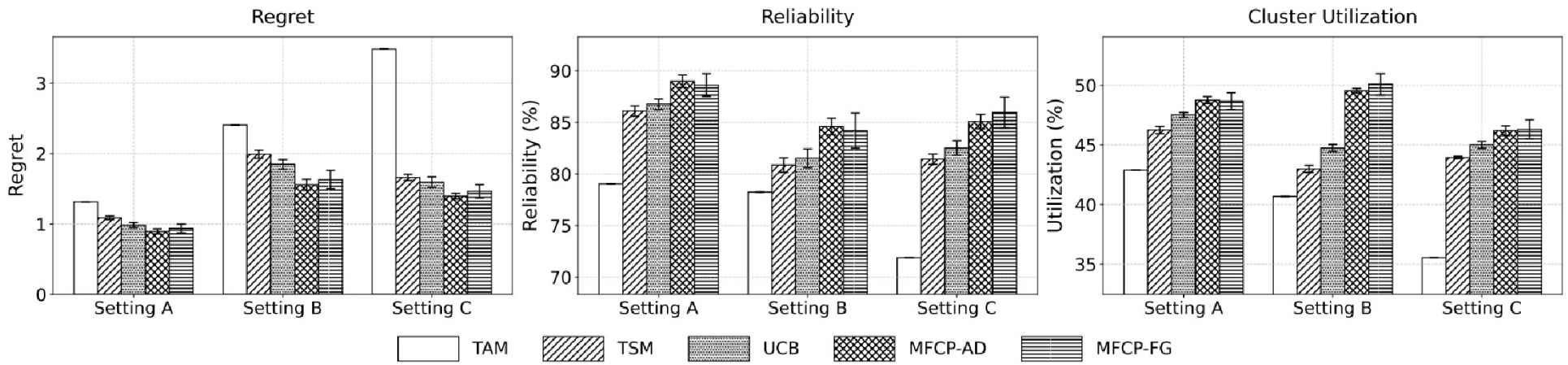


Figure 4: Overall experiment results. The figure illustrates the performance of the three metrics for the five methods under three cluster combination settings. Error bars represent 10× the actual standard deviation for visibility.

Scalability

- Performance with more tasks.
- MFCP remains lowest regret with higher utilization.

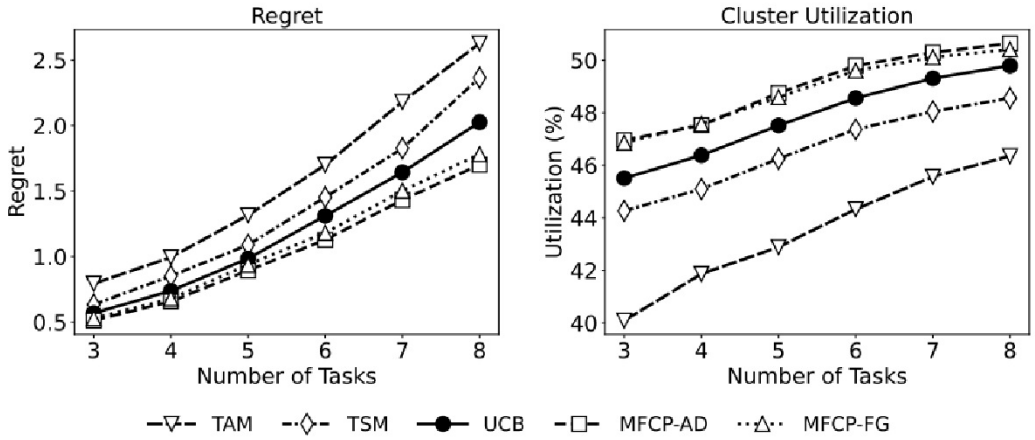


Figure 5: Experiment results with different number of tasks. MFCP-AD and MFCP-FG exhibit similar performance.

Parallel Execution Scenario

- We modeled the speedup ratio in the form of an exponential decay curve to evaluate the method's performance under parallel task scenarios.
- MFCP-FG outperforms baselines under non-convex conditions.

Table 2: Performance on parallel task execution settings

Method	Regret	Reliability	Utilization
TAM	3.032 ± 0.000	0.759 ± 0.000	0.485 ± 0.000
TSM	2.014 ± 0.035	0.832 ± 0.003	0.547 ± 0.001
UCB	1.835 ± 0.064	0.847 ± 0.003	0.553 ± 0.002
MFCP-FG	1.496 ± 0.081	0.851 ± 0.005	0.560 ± 0.003

Conclusion

- MFCP integrates prediction with matching, and forms it as **a bi-level optimization problem.**

- By computing **the gradient of the optimal solution of the matching optimization problem with respect to its parameters**, this lower-level optimization problem can be incorporated into the end-to-end training of the predictor.

- For convex matching optimization problems, we can use the **KKT conditions to compute the gradients**, whereas for more complex non-convex optimization problems, we can employ **zero-order gradient methods for estimation**.

Thank you