Paper Review :

Large-Scale Dynamic Scheduling for Flexible Job-Shop With Random Arrivals of New Jobs by Hierarchical Reinforcement Learning

산업경영공학과
이태희

2024.06.28

목차

- Problem Formulation
- Method
- Computational Experiment

Ⅰ. Problem Formulation

❖ Notation

- Successive new n' jobs : $\boldsymbol{J} = \{J_1, J_2, \ldots, J_{n'}\}$ that dynamically arrive ← $n' \in \{20, 100, 200, 1000\}$
- m Machines in system : $\boldsymbol{M} = \{M_1, M_2, \ldots, M_m\}$ ← $m \in \{10, 20\}$
- Consecutive Operations to process Job i : $O_i = \{O_{i1}, O_{i2}, \ldots, O_{in'_i}\}$ ← $n'_i \sim \mathrm{DU}(m - 5, m + 5)$
- A subset of eligible machines that can process Operation O_{ij} : $\boldsymbol{M_{ij}} \subseteq \boldsymbol{M}$
- Processing time of operation O_{ij} on machine k : p_{ijk} $\forall M_k \in \boldsymbol{M_{ij}}$ ← $p_{ijk} \sim \mathrm{DU}(0, 99)$
- Optimization goal : minimize *makespan* = C_{max}

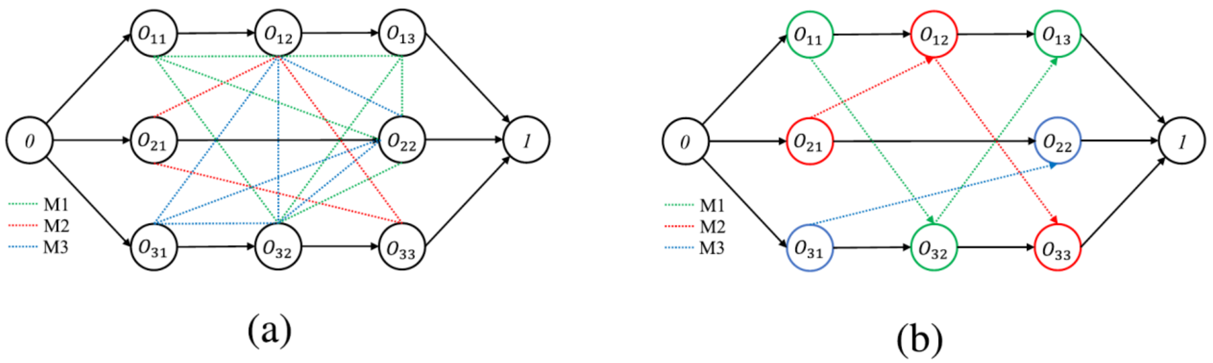


Fig. 1. Disjunctive graph representation of FJSP. (a) Disjunctive graph for an FJSP instance. (b) Example of a feasible solution.

Ⅱ. Method

❖ **Proposing Framework**

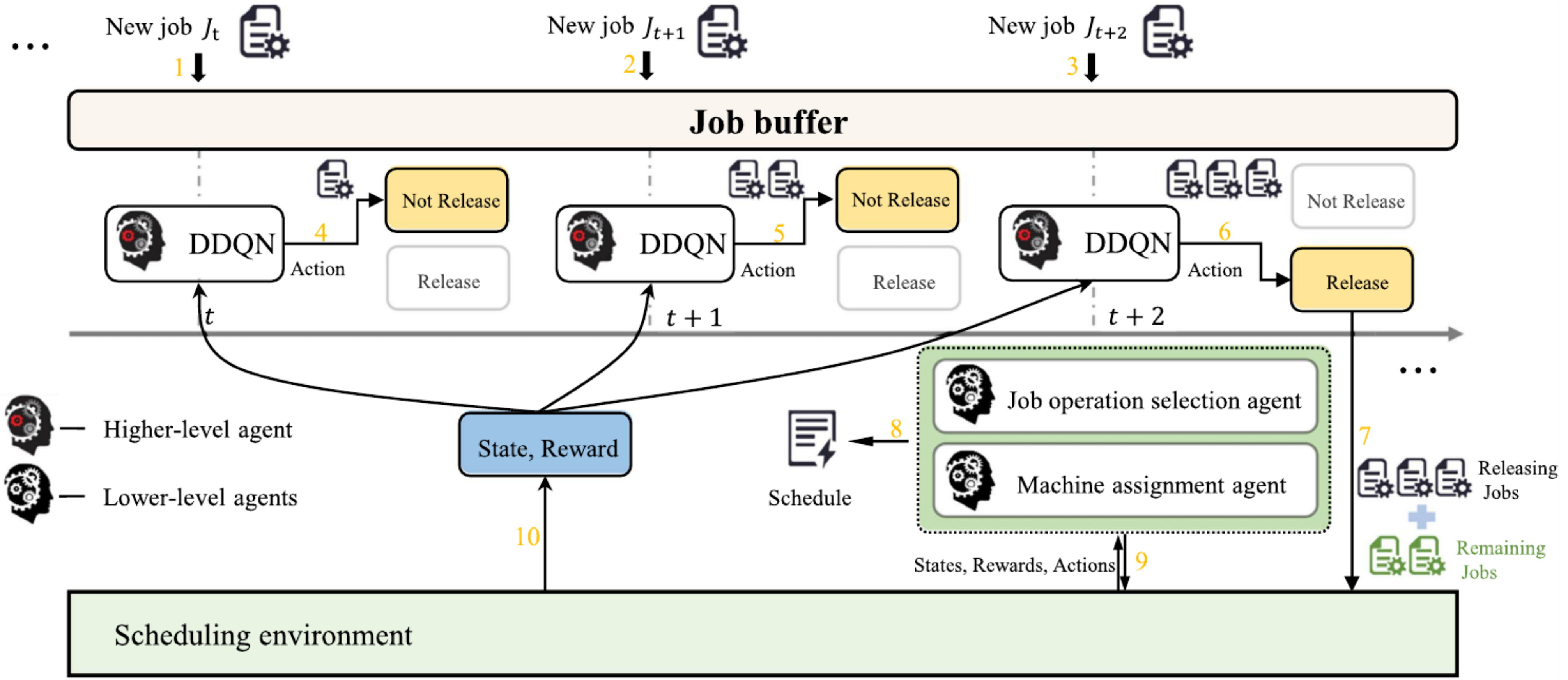
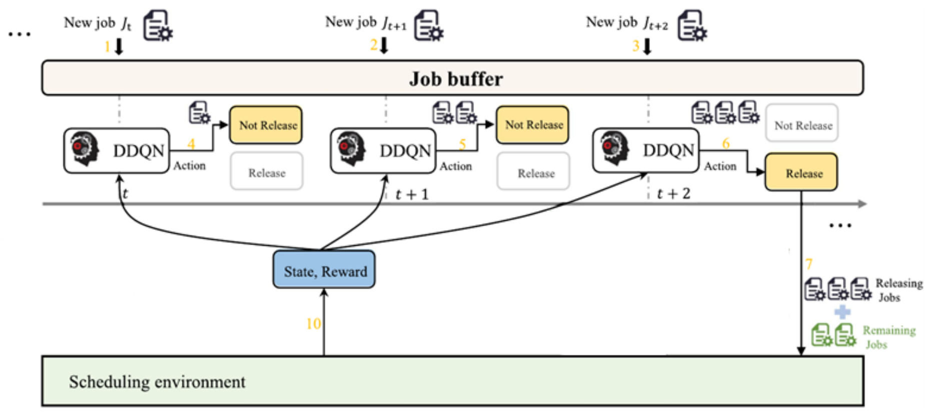


Fig. 2. End-to-end hierarchical reinforcement learning framework for DFJSP.

Ⅱ. Method

❖ Higher-level Agent

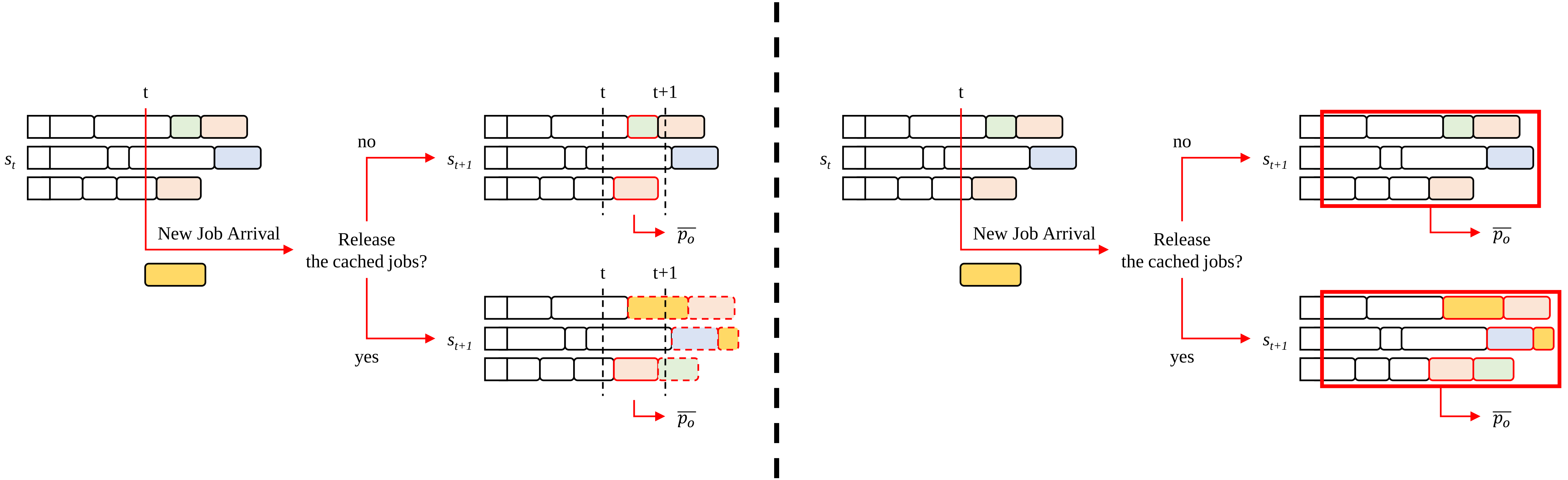


1. Markov Decision Process
   - State : (1 + m) values
     - Number of the cached jobs n_{new}
     - Variance value of completion time of all machines at the timestep
   - Action :
     - Whether to release the cached jobs, 1 if the action is to release
   - Reward :
     - Average processing time of each operation between the timestep t and $t + 1$
2. Training Method
   - Double Deep Q Network
3. Q-Network
   - Multilayer perceptron (MLP)

Ⅱ. Method

❖ Higher-level Agent

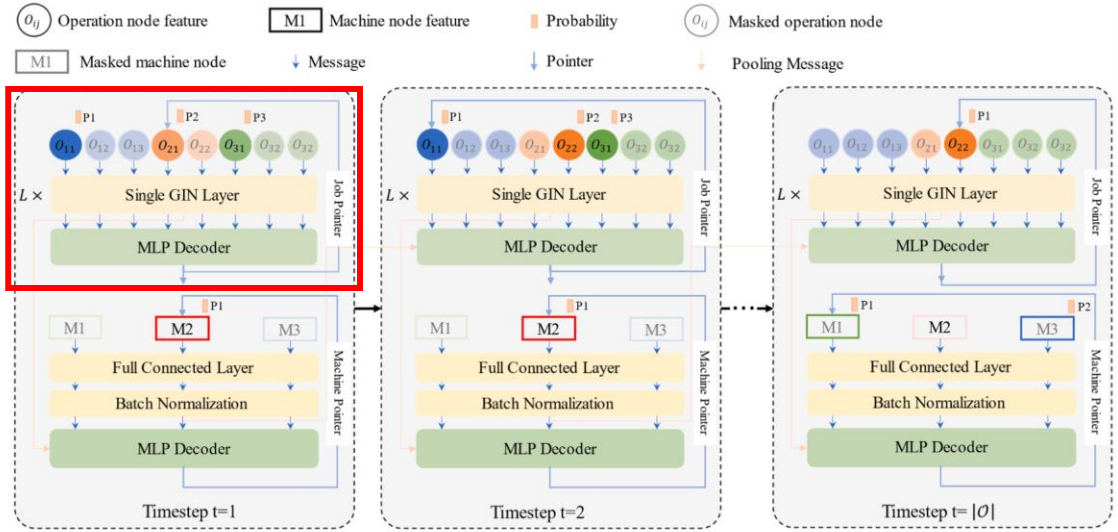
- Reward :
  - **Average processing time of each operation** between the timestep t and $t + 1$



Ⅱ. Method

❖ **Lower-level Agent(2022)** – Job Operation Selection Agent

1. Markov Decision Process
   - State : graph $G_t = (O_{new} \cup O_{remain}, C \cup D_t^o, D)$
     - C : A set of conjunctive arcs which represent consecutive operations from the same job [$\mathrm{LB}_t(O_{ij})$, $\mathrm{I}_t(O_{ij})$]
     - D : A set of disjunctive arcs, in which each arc connects a pair of operations that can be processed by the same machine
     - D_t^o : The disjunctive arcs which have been assigned directions till timestep t
   - Action :
     - A Specific eligible operation
   - Reward :
     - A joint reward shared by lower-level agents
2. Policy Network
   - Encoder : **Graph Isomorphism Network***
   - Decoder : MLP

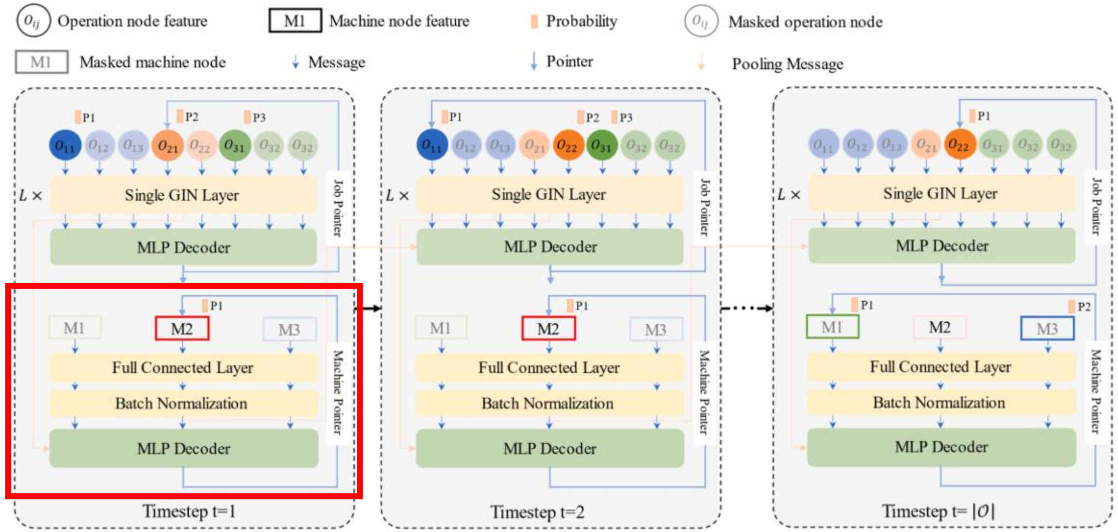


* K. Xu et al., (2019), "How Powerful are Graph Neural Networks", *ICLR 2019*, Poster

Ⅱ. Method

❖ **Lower-level Agent(2022)** – Machine Assignment Agent

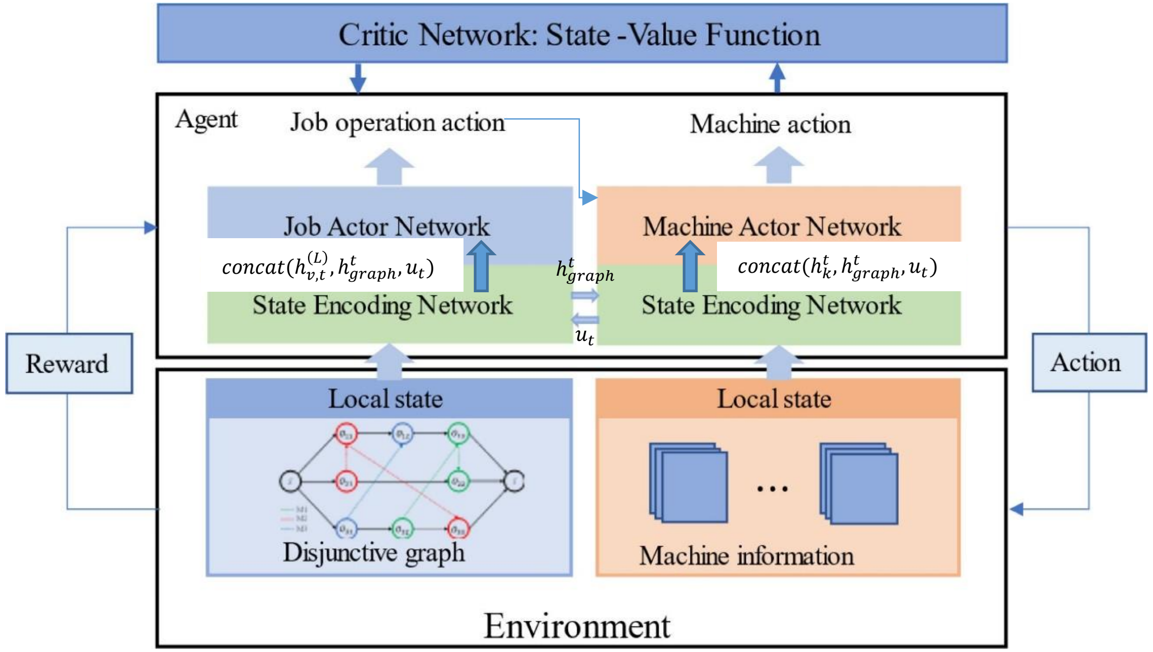
1. Markov Decision Process
   - State : 2 values × M_{ij}
     - The completion time for machine k before timestep t
     - p_{ijk}
   - Action :
     - A compatible machine for the operation selected by the job operation selection agent at timestep t
   - Reward :
     - A joint reward shared by lower-level agents
2. Policy Network
   - Encoder : MLP
   - Decoder : MLP



Ⅱ. Method

❖ Lower-level Agent(2022)

- Training Method : Proximal Policy Optimization
- A joint reward :
  - $LB_t(O_{ij}) = \begin{cases} \text{the completion time,} & \text{if } O_{ij} \text{ is scheduled} \\ LB_t(O_{i(j-1)}) + \min_k p_{ijk}, & \text{otherwise} \end{cases}$
  - $H_t = \max_{ij} LB_t(O_{ij})$
  - Reward $r_t^{step} = -(H_{t+1} - H_t)$



Ⅲ. Computational Experiment

- Dataset : abstracted from the actual production situation (CRCC High-Tech Equipment Corporation Limited)
- Baselines to be compared :
  1. Periodic(ΔT) Rescheduling + Dispatching Rules
  2. Learning to choose Dispatching Rules (L2D)
     - State : The combined state of lower-level agents
     - Action : The composite dispatching rules with top-k performance
     - Reward : Higher-level agent's reward
     - Training Method : DDQN
  3. ΔT Rescheduling + Ant Colony Optimization
  4. Gurobi solver with a time limit of 3600s
- Experimentation Hardware : Intel Xeon E5 V3 2600 CPU + a single Nvidia Tesla V100 GPU

Ⅲ. Computational Experiment

- Parameter settings of the simulator

TABLE III

PARAMETER SETTINGS OF THE SIMULATOR

Parameter	Value
Total number of machines (m)	$\{10, 20\}$
Number of initial jobs at the beginning (n)	$\{10\}$
Number of new arrived jobs (n')	$\{20, 100, 200, 1000\}$
Number of compatible machines of each operation	Unif $[1, m]$
Number of operations in each job	Unif $[m - 5, m + 5]$
Processing time of an operation on a compatible machine	Unif $[0, 99]$
The load factor of job shop, U	$\{1, 2, 4\}$
The rescheduling period (interval time), $\triangle T$	$\{\triangle t_{\text{avg}} \times n' / 10\}$

- Arrival time of new Job i : t_i = Poisson($\lambda = \Delta t_{avg} = \frac{\mu_a \mu_o}{U \times m}, k = n'$)

  μ_a = Average processing time of all operations; μ_a = Average operation number of each job

Ⅲ. Computational Experiment

- Main) Comparisons between the proposed HRL Algorithm with previous the Baseline methods

TABLE IV
MAIN RESULTS OF ALL METHODS ON TEST DATASETS

Size		HRL	Meta-heuristic	L2D	Dispatching rules							
		Ours	ACO		FIFO+SPT	MWKR+SPT	MOPNR+SPT	FIFO+EET	MWKR+EET	LWKR+SPT	MOPNR+EET	LWKR+EET
20-10-1	Obj.	**1286.86**	1395.23	1454.31	1506.71	1557.27	1736.26	2375.35	3478.81	3664.94	6861.59	7620.78
	Gap	**0.00%**	8.48%	13.06%	17.07%	21.00%	34.91%	84.56%	170.30%	184.77%	433.15%	492.14%
	Time (s)	3.92	487.5	1.28	0.93	0.96	1.02	0.95	0.95	**0.89**	0.98	1.03
100-10-1	Obj.	**5211.35**	5485.54	5536.69	5756.8	6648.51	6181.89	9808.56	11847.09	12681.01	24254.14	24604.53
	Gap	**0.00%**	5.26%	6.24%	10.47%	27.59%	18.63%	88.23%	127.35%	143.35%	365.44%	372.17%
	Time (s)	16.89	3402.35	8.15	7.75	7.57	7.23	7.79	6.98	7.22	**6.85**	7.55
100-20-2	Obj.	**2802.32**	–	3112.56	3182.43	3587.523	3447.31	8760.18	11400.81	14732.66	45477.84	42167.01
	Gap	**0.00%**		11.06%	13.56%	28.01%	23.01%	212.60%	306.83%	425.73%	1522.86%	1404.71%
	Time (s)	38.23		21.68	19.28	19.67	18.93	**18.56**	19.74	19.38	19.22	20.05
20-10-2	Obj.	**918.32**	1025.38	1078.31	1155.46	1229.55	1318.11	1898.77	2740.22	3712.33	5178.43	7278.04
	Gap	**0.00%**	11.66%	17.43%	25.87%	33.94%	43.59%	106.84%	198.50%	304.39%	464.10%	692.82%
	Time (s)	4.08	456.33	1.21	1.1	0.95	**0.93**	1.03	0.95	0.94	**0.93**	0.94
100-10-2	Obj.	**2934.25**	3218.09	3384.19	3466.69	4084.56	4109.43	6792.14	9288.82	11635.56	23599.87	21482.51
	Gap	**0.00%**	9.68%	15.33%	18.16%	39.21%	40.06%	131.50%	216.59%	296.58%	704.36%	632.19%
	Time (s)	16.35	3456.48	8.38	7.79	7.16	7.15	7.97	7.15	7.19	7.26	**7.13**
100-20-4	Obj.	**1838.3**	–	2284.02	2391.86	3237.07	2897.36	7806.9	10885.06	14580.31	45762	40704.61
	Gap	**0.00%**		24.26%	30.11%	76.09%	57.61%	324.68%	492.12%	693.14%	2389.36%	2114.25%
	Time (s)	37.45		21.22	19.24	18.62	19.52	20.39	19.24	18.93	20.09	19.06

Note: We name the instance size in the format of "number of dynamic jobs n' - number of machines m - job shop load factor U".

Job sequencing : First In First Out, Most WorK Remaining, Most Operation Number Remaining, Least WorK Remaining | Machine assignment : Shortest Processing Time, Earliest End Time

Ⅲ. Computational Experiment

- Ablation 1) Can both Higher-level and Lower-level agents generalize to Large-scale instances?

  → Generalization ability

TABLE V
GENERALIZATION PERFORMANCE TESTING OF OUR METHOD

Size		Ours	L2D	FIFO+SPT	MOPNR+SPT	MWKR+SPT	FIFO+EET	MWKR+EET	LWKR+SPT	LWKR+EET	MOPNR+EET
200-10-2	Obj.	**5296.25**	6115.36	6305.00	7091.07	7753.23	14799.19	18206.84	21275.45	42843.84	45994.58
	Gap	**0.00%**	15.46%	19.05%	33.89%	46.40%	179.44%	243.78%	301.73%	708.98%	768.48%
	Time (s)	**38.56**	26.42	24.31	**22.10**	21.89	25.73	22.32	22.36	22.32	22.91
200-20-4	Obj.	**3089.68**	3789.25	3910.02	4766.13	5213.91	13568.95	18732.04	27892.50	78655.56	92011.70
	Gap	**0.00%**	22.64%	26.55%	54.25%	68.75%	339.17%	506.27%	802.76%	2445.75%	2878.03%
	Time (s)	**79.83**	67.58	58.84	60.38	**57.51**	64.52	61.49	61.40	59.03	63.38
1000-20-4	Obj.	**13018.52**	13976.39	14219.61	16979.21	18935.89	61606.16	81650.11	377840.66	476172.61	
	Gap	**0.00%**	7.35%	9.22%	30.42%	45.45%	373.21%	527.18%	926.55%	2802.33%	3557.65%
	Time (s)	**493.25**	433.57	389.34	396.52	365.81	414.04	361.84	**341.76**	342.21	385.53

Job sequencing : First In First Out, Most WorK Remaining, Most Operation Number Remaining, Least WorK Remaining | Machine assignment : Shortest Processing Time, Earliest End Time

Ⅲ. Computational Experiment

- Ablation 2) Does the Higher-level agent lean to split the DFJSP from a long-term view?

  → Robust performance

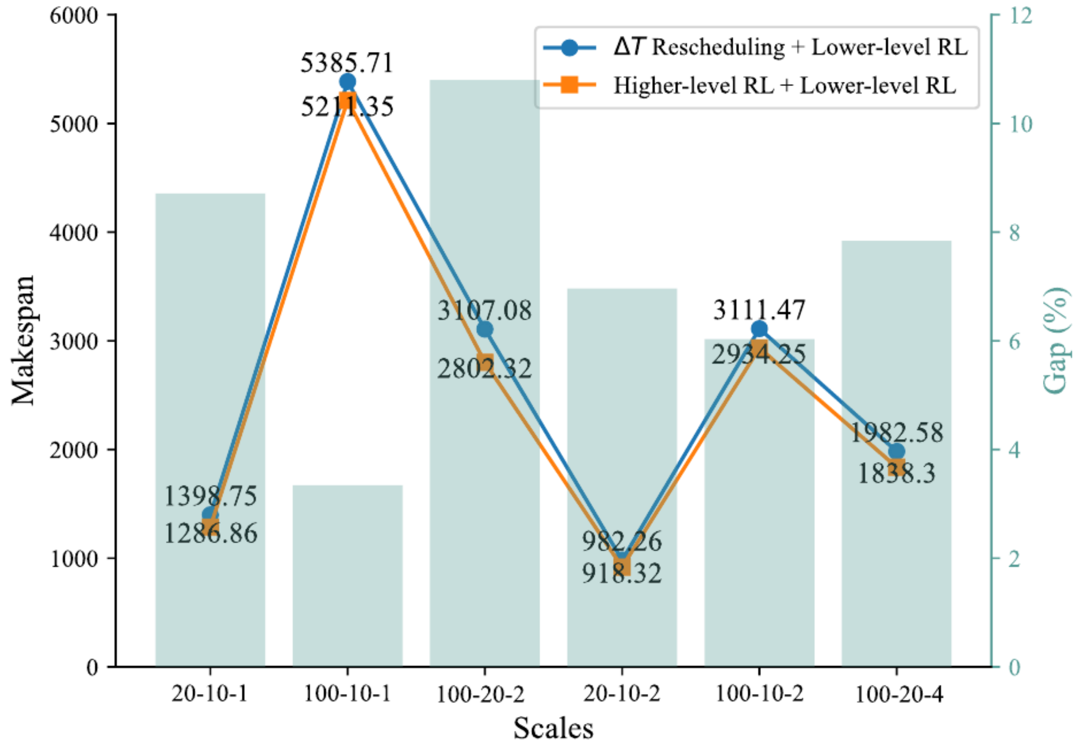


Fig. 6. Effectiveness of the higher-level agent.

Ⅲ. Computational Experiment

- Ablation 3) Are Lower-level agents needed to solve the static FJSP efficiently?

TABLE VI
EFFECTIVENESS OF LOWER-LEVEL AGENTS ON STATIC FJSP

Size		MIP	Dispatching Rules		Ours
			Best	Worst	
20 × 10	Obj.	391.41	566.32	1815.82	**454.85**
	Gap	0.00%	44.69%	363.92%	**16.21%**
	Time (s)	3600	**0.21**	0.23	0.34
20 × 20	Obj.	322.54	430.79	2762.01	**361.75**
	Gap	0.00%	33.56%	756.33%	**12.16%**
	Time (s)	3600	0.46	**0.44**	1.08
30 × 20	Obj.	-	525.08	3462.27	**433.42**
	Gap		21.15%	698.83%	**0.00%**
	Time (s)		0.86	**0.82**	1.97

Note: "Best" and "Worst" denote the best and worst results of eight dispatching rules for each scale instance.

Ⅲ. Computational Experiment

- Ablation 4) Can Lower-level agents generalize to public benchmarks?

  → Efficient applications

TABLE VII
RESULTS ON PUBLIC BENCHMARKS

Instance	Dispatching rules	Meta-heuristics		Ours	UB
	Best	ACO	2SGA [25]		
10 × 5	613.4 (20.98%)	**537.4 (6.04%)**	**510.4 (0.67%)**	547.8 (7.85%)	507.0
15 × 5	842.8 (6.14%)	**821.2 (3.42%)**	**795.0 (0.13%)**	820.0 (3.27%)	794.0
20 × 5	1090.0 (4.73%)	**1068.2 (2.59%)**	**1041.2 (0.04%)**	1060.2 (1.86%)	1040.8
10 × 10	716.4 (5.38%)	**694.8 (2.28%)**	**680.2 (0.06%)**	681.8 (0.29%)	679.8
15 × 10	894.4 (15.08%)	**852.0 (9.65%)**	**796.6 (2.50%)**	860.8 (10.75%)	777.2
20 × 10	1119.8 (6.22%)	**1126.4 (6.87%)**	**1067.2 (1.23%)**	1096.8 (4.04%)	1054.2
30 × 10	1611.2 (3.80%)	**1597.2 (2.89%)**	**1557.2 (0.36%)**	1584.2 (2.06%)	1552.2
15 × 15	1002.4 (5.43%)	**993.0 (4.44%)**	**953.0 (0.23%)**	981.6 (3.23%)	950.8
Ave. Gap	8.54%	**4.77%**	**0.80%**	4.20%	0.00%

Note: "UB" column is the best result from the literature. We run the ACO for one time and 200 iterations to get the results.