SCIENCE CHINA
Information Sciences

• Supplementary File •

Deep multi-agent reinforcement learning for dynamic energy-efficient cascaded dual-shop collaborative scheduling with mating operation

Haizhu BAO[1,2], Quanke PAN[1*], Chee-Meng CHEW[2], Ling WANG[3] & Liang GAO[4,5]

[1]*School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China*
[2]*Department of Mechanical Engineering, National University of Singapore, Singapore 117575, Singapore*
[3]*Department of Automation, Tsinghua University, Beijing 100084, China*
[4]*National Center of Technology Innovation for Intelligent Design and Numerical Control, Wuhan 430074, China*
[5]*State Key Laboratory of Intelligent Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China*

Appendix A Details about problem description

Appendix A.1 Notation description

Notation descriptions are given as follows:

Indexes:

j, j' Index of job, $j,\ j' \in \{0, 1, \cdots, n\}$.

i Index of machine on Phase 1, $i \in \{0, 1, \cdots, m\}$.

f Index of factory on Phase 1, $f \in \{1, 2, \cdots, \delta\}$.

k Index of stage on Phase 2, $k \in \{0, 1, \cdots, s\}$.

l Index of parallel machine on Phase 2, $l \in \{0, 1, \cdots, r_k\}$.

Sets:

JM Main order set.

JS Sub order set.

J Order set, $J = JM \cup JS = \{J_1, J_2, \cdots, J_n\}$.

F Factory set on Phase 1, $F = \{F_1, F_2, \cdots, F_\delta\}$.

M_f Machine set in F_f on Phase 1, $M_f = \{M_{f,1}, M_{f,2}, \cdots, M_{f,m}\}$.

K Stage set on Phase 2, $K = \{K_1, K_2, \cdots, K_s\}$.

M_k^p Parallel machine set at K_k on Phase 2, $M_k^p = \{M_{k,1}^p, M_{k,2}^p, \cdots, M_{k,r_k}^p\}$.

O Operation set, $O = \{O_{1,1}, O_{1,2}, \cdots, O_{n,m+s}\}$.

Ψ Subset of J which has at least two jobs.

$Y_{j,j',k}$ Mating operation set.

Parameters:

n Number of jobs.

m Number of machines on Phase 1.

δ Number of factories on Phase 1.

s Number of stages on Phase 2.

r_k Number of machines at K_k on Phase 2.

h A sufficiently large positive number.

AT_j Arrival time of J_j.

ρ_j Tightness factor of J_j.

* Corresponding author (email: panquanke@shu.edu.cn)

D_j	Due date of J_j, $D_j = AT_j + \rho_j \times (\sum_{i=1}^{m} p_{j,i}^1 + \sum_{k=1}^{s} p_{j,k}^2)$.
$p_{j,i}^1$	Processing time of $O_{j,i}$ on Phase 1.
$p_{j,k}^2$	Standard processing time of $O_{j,k}$ at K_k on Phase 2.
$st_{j,j',i}^1$	Setup time from $O_{j,i}$ to $O_{j',i}$ on Phase 1.
$st_{j,j',l,k}^2$	Setup time from $O_{j,k}$ to $O_{j',k}$ at $M_{k,l}^p$ on Phase 2.
$tp_{j,f}$	Transportation time of J_j from $M_{f,m}$ to Phase 2.
β_i^1	Energy consumption (EC) per unit time of $M_{f,i}$ at processing mode on Phase 1.
$\beta_{k,l}^2$	EC per unit time of $M_{k,l}^p$ at processing mode on Phase 2.
α	EC per unit time at setup mode.
θ	EC per unit time at idle mode.
μ	EC per unit time at transportation phase.

Variables:

λ_j^a	Binary auxiliary variable, $\lambda_j^a \in \{0,1\}, a \in \{1,2,3,4\}$.
$x_{j,f}$	Binary variable that takes 1 if J_j is assigned in F_f on Phase 1; and 0 otherwise.
$z_{j,j',f}$	Binary variable that takes 1 if J_j is the immediate predecessor of $J_{j'}$ in F_f on Phase 1; and 0 otherwise.
$y_{j,l,k}$	Binary variable that takes 1 if J_j is assigned to $M_{k,l}^p$ on Phase 2; and 0 otherwise.
$w_{j,j',l,k}$	Binary variable that takes 1 if J_j is the immediate predecessor of $J_{j'}$ at $M_{k,l}^p$ on Phase 2; and 0 otherwise.
t_j	Auxiliary variable.
$d_{j,i}^1$	Completion time of J_j at $M_{f,i}$ on Phase 1.
$d_{j,k}^2$	Completion time of J_j at K_k on Phase 2.
T_j	Tardiness of $J_j, T_j = \max(0, d_{j,s}^2 - D_j)$.
TTD	Total tardiness, $TTD = \sum_{j=1}^{n} T_j$.
$PEC_{j,f,i}^1$	Continuous variable for EC when J_j is processed on $M_{f,i}$ at processing mode on Phase 1.
$PEC_{j,l,k}^2$	Continuous variable for EC when J_j is processed on $M_{k,l}$ at processing mode on Phase 2.
$SEC_{j,f,i}^1$	Continuous variable for EC when J_j is processed on $M_{f,i}$ at setup mode on Phase 1.
$SEC_{j,l,k}^2$	Continuous variable for EC when J_j is processed on $M_{k,l}^p$ at setup mode on Phase 2.
$IEC_{j,f,i}^1$	Continuous variable for EC when J_j is processed on $M_{f,i}$ at idle mode on Phase 1.
$IEC_{j,l,k}^2$	Continuous variable for EC when J_j is processed on $M_{k,l}$ at idle mode on Phase 2.
$REC_{j,l}$	Continuous variable for EC when J_j is transported to $M_{1,l}^p$ on transportation phase.
TEC	Continuous variable for EC of a schedule.

Appendix A.2 MILP model

The MILP model for DECDS-M is established as follows.

Objectives:

$$Minimize \begin{cases} f_1 = TTD \\ f_2 = TEC \end{cases} \tag{A1}$$

Subject to:

$$x_{0,f} = 1, \forall f \tag{A2}$$

$$\sum_{f=1}^{\delta} x_{j,f} = 1, \forall j \notin \{0\} \tag{A3}$$

$$z_{j,j',f} \leqslant x_{j,f}, \forall j', j \notin \{0\}, j \neq j', \forall f \tag{A4}$$

$$z_{j,j',f} \leqslant x_{j',f}, \forall j', j \notin \{0\}, j \neq j', \forall f \tag{A5}$$

$$\sum_{j=0, j\neq j'}^{n} z_{j,j',f} = x_{j',f}, \forall j', f \tag{A6}$$

$$\sum_{j'=0, j'\neq j}^{n} z_{j,j',f} = x_{j,f}, \forall j, f \tag{A7}$$

$$\sum_{J_j \in \Psi} \sum_{J_{j'} \in \Psi} z_{j,j',f} \leqslant |\Psi| - 1, \forall \Psi \subseteq J, j \neq j' \tag{A8}$$

$$d^1_{j,i} \geqslant AT_j, \forall i, j \tag{A9}$$

$$d^1_{j,i} \geqslant d^1_{j,i-1} + p^1_{j,i}, \forall i, j \notin \{0\} \tag{A10}$$

$$d^1_{j',i} \geqslant d^1_{j,i} + p^1_{j',i} + st^1_{j,j',i} + (z_{j,j',f} - 1) \cdot h, \forall i, j, j' \notin \{0\}, j \neq j', f \tag{A11}$$

$$d^1_{j,i} \geqslant st^1_{0,j,i} + p^1_{j,i} + (z_{0,j,f} - 1) \cdot h, \forall i, j, f \tag{A12}$$

$$\sum_{l=1}^{r_k} y_{j,l,k} = 1, \forall j, k \tag{A13}$$

$$y_{0,l,k} = 1, \forall l, k \tag{A14}$$

$$w_{j,j',l,k} \leqslant y_{j,l,k}, \forall j, j', k \notin \{0\}, j \neq j', l \tag{A15}$$

$$\sum_{j=0, j\neq j'}^{n} w_{j,j',l,k} = y_{j',l,k}, \forall j', k \notin \{0\}, l \tag{A16}$$

$$\sum_{j'=0, j\neq j'}^{n} w_{j,j',l,k} = y_{j',l,k}, \forall j, k \notin \{0\}, l \tag{A17}$$

$$\sum_{J_j \in \Psi} \sum_{J_{j'} \in \Psi} w_{j,j',l,k} \leqslant |\Psi| - 1, \forall \Psi \subseteq J, j, j', k \notin \{0\}, j \neq j', l \tag{A18}$$

$$d^2_{j,k} \geqslant 0, \forall j, k \tag{A19}$$

$$d^2_{j,0} \geqslant d^1_{j,m} + tp_{j,f} + (x_{j,f} - 1) \cdot h, \forall j \notin \{0\}, f \tag{A20}$$

$$d^2_{j,k} \geqslant d^2_{j,k-1} + \frac{p^2_{j,k}}{v_{k,l}} + (y_{j,l,k} - 1) \cdot h, \forall j, k \notin \{0\}, l \tag{A21}$$

$$t_j \geqslant \begin{cases} d^2_{j,k-1} + \frac{p^2_{j',k}}{v_{k,l}} + \frac{p^2_{j,k}}{v_{k,l}} + (y_{j,l,k} - 1) \cdot h, \\ d^2_{j',k-1} + \frac{p^2_{j',k}}{v_{k,l}} + \frac{p^2_{j,k}}{v_{k,l}} + (y_{j',l,k} - 1) \cdot h, \end{cases} \quad \forall O_{j,k}, O_{j',k} \in Y_{j,j',k}, l \tag{A22}$$

$$t_j \leqslant \begin{cases} d^2_{j,k-1} + \frac{p^2_{j',k}}{v_{k,l}} + \frac{p^2_{j,k}}{v_{k,l}} + \left(2 - y_{j,l,k} - \lambda^1_j\right) \cdot h, \\ d^2_{j',k-1} + \frac{p^2_{j',k}}{v^2_{k,l}} + \frac{p^2_{j,k}}{v_{k,l}} + \left(2 - y_{j',l,k} - \lambda^1_{j'}\right) \cdot h, \end{cases} \quad \forall O_{j,k}, O_{j',k} \in Y_{j,j',k}, l \tag{A23}$$

$$\lambda^1_j + \lambda^1_{j'} \geqslant 1, \forall O_{j,k}, O_{j',k} \in Y_{j,j',k}, l \tag{A24}$$

$$d^2_{j,k} \geqslant t_j, \forall O_{j,k} \in Y_{j,j',k} \tag{A25}$$

$$d^2_{j',k} \geqslant t_j, \forall O_{j',k} \in Y_{j,j',k} \tag{A26}$$

$$d^2_{j',k} \geqslant d^2_{j,k} + \frac{p^2_{j',k}}{v_{k,l}} + st^2_{j,j',l,k} + (w_{j,j',l,k} - 1) \cdot h, \forall j, j', k \notin \{0\}, j \neq j', l \tag{A27}$$

$$T_j \geqslant 0, \forall j \tag{A28}$$

$$T_j \geqslant d^2_{j,s} - D_j, \forall j \tag{A29}$$

$$T_j \leqslant \left(1 - \lambda^2_j\right) \cdot h, \forall j \tag{A30}$$

$$T_j \leqslant d^2_{j,s} - D_j + \left(1 - \lambda^3_j\right) \cdot h, \forall j \tag{A31}$$

$$\lambda^2_j + \lambda^3_j \geqslant 1, \forall j \tag{A32}$$

$$PEC^1_{j,f,i} \geqslant \beta^1_i \cdot p^1_{j,i} + (x_{j,f} - 1) \cdot h, \forall j, f, i \tag{A33}$$

$$SEC^1_{j,f,i} \geqslant \alpha \cdot st^1_{j,j',i} + (z_{j,j',f} - 1) \cdot h, \forall j, j' \notin \{0\}, j \neq j', f, i \tag{A34}$$

$$IEC^1_{j,f,i} \geqslant \theta \cdot \left(d^1_{j,i-1} - d^1_{j',i} - st^1_{j,j',i}\right) + (z_{j,j',f} - 1) \cdot h, \forall j, j' \notin \{0\}, j \neq j', f, i \tag{A35}$$

$$PEC^2_{j,l,k} \geqslant \beta^2_{k,l} \cdot \frac{p^2_{j,k}}{v_{k,l}} + (y_{j,l,k} - 1) \cdot h, \forall j, l, k \tag{A36}$$

$$SEC^2_{j,l,k} \geqslant \alpha \cdot st^2_{j,j',l,k} + (w_{j,j',l,k} - 1) \cdot h, \forall j, j' \notin \{0\}, j \neq j', l, k \tag{A37}$$

$$IEC^2_{j,l,k} \geqslant \theta \cdot \left(d^2_{j',k-1} - d^2_{j,k} - st^2_{j,j',l,k}\right) + (w_{j,j',l,k} - 1) \cdot h, \forall j, j' \notin \{0\}, j \neq j', l, k \tag{A38}$$

$$REC_{j,l} \geqslant \mu \cdot t_{f,l} + (x_{j,f} + y_{j,l,1} - 2) \cdot h, \forall j, l \tag{A39}$$

$$TEC \geqslant \sum_{f=1}^{\delta} \sum_{i=1}^{m} \sum_{j=0}^{n} \left(PEC^1_{j,f,i} + SEC^1_{j,f,i} + IEC^1_{j,f,i}\right) + \sum_{k=1}^{s} \sum_{l=1}^{r_k} \sum_{j=0}^{n} \left(PEC^2_{j,l,k} + SEC^2_{j,l,k} + IEC^2_{j,l,k}\right)$$

$$+\sum_{l=1}^{r_1}\sum_{j=1}^{n} REC_{j,l} \tag{A40}$$

$$TTD \geqslant \sum_{j=1}^{n} T_j. \tag{A41}$$

Equation (A1) demonstrates the objective function. Constraint (A2) ensures that each factory includes a dummy job. Constraint (A3) guarantees that each job must be assigned to only one factory in Phase 1. Constraints (A4) and (A5) ensure that if the job is not assigned to a particular factory, there is no precursor and successor of the job in the factory. Constraints (A6)-(A8) guarantee that the sequence of jobs assigned to any factory cannot form a subring. Constraint (A9) indicates that a job can only be processed only after it has arrived. Constraint (A10) represents the precedence constraint between consecutive operations of the same job. Constraint (A11) describes the relationship between the completion times of two adjacent jobs in the same factory. Constraint (A12) stipulates the completion time of the initial job in each factory. Constraint (A13) ensures that in Phase 2, each operation can only be assigned to one machine among multiple parallel machines for processing. Constraint (A14) guarantees that each stage includes a dummy job in Phase 2. Constraint (A15) ensures that if the job is not assigned to a particular machine in Phase 2, there is no precursor and successor of the job on the machine. Constraints (A16)-(A18) prevent the sequence of jobs assigned to any machine cannot form a subring. Constraint (A19) specifies that the completion time of J_j is non-negative in Phase 2. Constraint (A20) ensures that all jobs are promptly transferred to the transportation phase after completing processing in Phase 1. Constraint (A21) enforces that the operation of a job cannot begin until the previous stage has completed. Constraints (A22)-(A26) require that the mating operations of the main order and the corresponding sub-order can only begin after their respective predecessor operations are complete. Constraint (A27) describes the relationship between the completion times of two adjacent operations on any machine. Constraints (A28)-(A32) define the tardiness of the jobs and serve as the linearized representation of T_j. The EC of the machines in Phase 1 for processing mode, setup mode, and idle mode is defined by constraints (A33)-(A35), respectively. In Phase 2, constraints (A36)-(A38) define the EC in these modes. Constraint (A39) represents the EC during the transportation phase. Equations (A40) and (A41) define the TEC and TTD, respectively. The code of the MILP model is published on GitHub (https://github.com/banian2314/DECDS-M.git).

Appendix A.3 Numerical example

The standard processing times and SDSTs of all jobs, along with the processing speeds and power consumption in the processing mode of all machines are listed in Table A1, Table A2, and Table A3. Besides, the EC of all machines per unit of time is set as Table A4.

Table A1 Example of the DECDS-M.

Phase		Standard processing time ($p^1_{j,i}$ or $p^2_{j,l}$)						$\left(v_{k,l}, \beta^2_{k,l}\right)$				
		J_1	J_2	J_3	J_4	J_5	J_6	$M^p_{1,1}$	$M^p_{1,2}$	$M^p_{2,1}$	$M^p_{2,2}$	$M^p_{2,3}$
1	$i=1$	3	3	3	2	2	3	/	/	/	/	/
	$i=2$	2	3	2	2	2	1	/	/	/	/	/
2	$k=1$	2.4	3.6	3	2	3	/	$(1, 6)$	$(1.2, 8)$	/	/	/
	$k=2$	4	1	3.6	2	2	1.2	/	/	$(1.2, 7)$	$(1, 6)$	$(1, 5)$
AT_j		0	0	3	0	5	5	/	/	/	/	/
ρ_j		1.5	1.5	1	1.5	1	2	/	/	/	/	/
D_j		17.1	20.4	17.6	12	14	21.4	/	/	/	/	/

Table A2 Sequence-dependent setup times in Phase 1 $\left(st^1_{j,j',1}, st^1_{j,j',2}\right)$.

	J_1	J_2	J_3	J_4	J_5	J_6
J_0	$(1, 2)$	$(1, 2)$	$(2, 1)$	$(2, 3)$	$(1, 2)$	$(2, 1)$
J_1	/	$(2, 2)$	$(2, /)$	$(1, 1)$	$(2, 1)$	$(1, 1)$
J_2	$(1, 2)$	/	$(2, 1)$	$(2, 1)$	$(1, 1)$	$(1, 1)$
J_3	$(1, 1)$	$(2, 2)$	/	$(1, 1)$	$(1, 2)$	$(2, 1)$
J_4	$(1, 2)$	$(2, 2)$	$(1, 1)$	/	$(2, 2)$	$(1, 2)$
J_5	$(1, 1)$	$(1, 2)$	$(2, 2)$	$(1, 1)$	/	$(1, 2)$
J_6	$(2, 1)$	$(2, 1)$	$(2, 1)$	$(2, 1)$	$(1, 1)$	/

Table A3 Sequence-dependent setup times in Phase 2.

	($st^2_{j,j',1,1}, st^2_{j,j',2,1}$)						($st^2_{j,j',1,2}, st^2_{j,j',2,2}, st^2_{j,j',3,2}$)					
	J_1	J_2	J_3	J_4	J_5	J_6	J_1	J_2	J_3	J_4	J_5	J_6
J_0	(1, 1)	(1, 1)	(2, 2)	(2, 2)	(1, 1)	(1, 1)	(2,2,2)	(2,2,2)	(1,1,1)	(3,3,3)	(2,2,2)	(2,2,2)
J_1	/	(2, 2)	(2, 2)	(1, 1)	(2, 2)	(2, 2)	/	/	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
J_2	(1, 2)	/	(2, 1)	(2, 1)	(1, 1)	(1, 1)	(2,2,2)	/	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)
J_3	(1, 1)	(2, 2)	/	(1, 1)	(1, 1)	(1, 1)	(1,1,1)	(1,1,1)	/	(1,1,1)	(2,2,2)	(1,1,1)
J_4	(1, 1)	(2, 2)	(1, 1)	/	(2, 2)	(2, 2)	(1,1,1)	(1,1,1)	(1,1,1)	/	(2,2,2)	(2,2,2)
J_5	(2, 2)	(2, 2)	(2, 2)	(2, 2)	/	/	(2,2,2)	(2,2,2)	(2,2,2)	(1,1,1)	/	(2,2,2)
J_6	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(1, 1)	/	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	(1,1,1)	/

Table A4 The EC of all machines per unit of time.

	β^1_i and $\beta^2_{k,l}$		α	θ	μ
Phase 1	$i = 1$	$\beta^1_1 = 6$			
	$i = 2$	$\beta^1_2 = 7$			
Phase 2	$k = 1, l = 1$	$\beta^2_{1,1} = 6$			
	$k = 1, l = 2$	$\beta^2_{1,2} = 8$	1	0.5	3
	$k = 2, l = 1$	$\beta^2_{2,1} = 7$			
	$k = 2, l = 2$	$\beta^2_{2,2} = 6$			
	$k = 2, l = 3$	$\beta^2_{2,3} = 5$			

Appendix B Details of the reward function

The reward function guides agents to perceive feedback from their actions, driving the improvement of the learning policy's performance. In DECDS-M, the optimization goal is to minimize the TTD and TEC. To achieve this, we design the reward based on the difference in objective values between two consecutive states, normalized as shown in Eq. (B1).

$$r\left(s_t, a_t, s_{t+1}\right) = \frac{w\left(TTD\left(s_t\right) - TTD\left(s_{t+1}\right)\right)}{TTD\left(s_0\right)} + \frac{(1-w) \times \left(TEC\left(s_t\right) - TEC\left(s_{t+1}\right)\right)}{TEC\left(s_0\right)}. \tag{B1}$$

It is important to note that since there are unscheduled operations in both states s_t and s_{t+1}, we estimate the two objectives at each state. When the discount factor γ is set to 1, the cumulative reward can be derived as shown in Eq. (B2).

$$\begin{aligned} G = \sum_{t=0}^{|O|} r\left(s_t, a_t, s_{t+1}\right) = & \frac{w\left(TTD\left(s_0\right) - TTD\left(s_1\right) + TTD\left(s_1\right) - TTD\left(s_2\right) + \cdots + TTD\left(s_{|O|-1}\right) - TTD\left(s_{|O|}\right)\right)}{TTD\left(s_0\right)} + \\ & \frac{(1-w) \cdot \left(TEC\left(s_0\right) - TEC\left(s_1\right) + TEC\left(s_1\right) - TEC\left(s_2\right) + \cdots + TEC\left(s_{|O|-1}\right) - TEC\left(s_{|O|}\right)\right)}{TEC\left(s_0\right)} = \\ & \frac{w \cdot \left(TTD\left(s_0\right) - TTD\left(s_{|O|}\right)\right)}{TTD\left(s_0\right)} + \frac{(1-w) \cdot \left(TEC\left(s_0\right) - TEC\left(s_{|O|}\right)\right)}{TEC\left(s_0\right)} \\ = & \frac{w \cdot \left(TTD\left(s_0\right) - TTD\right)}{TTD\left(s_0\right)} + \frac{(1-w) \cdot \left(TEC\left(s_0\right) - TEC\right)}{TEC\left(s_0\right)} \end{aligned} \tag{B2}$$

Appendix C More details about experiments

Appendix C.1 Training and testing instances

DMRLIG was trained and evaluated on synthetic instances of varying scales. The instance generation process was consistent with the methodologies described in the literature [1, 2]. The parameters of the instances are listed in Table C1. In the computational experiments, each instance size is denoted by a triplet (n, m, s), where $n \in \{15, 20, 40, 50\}$ represents the number of jobs, $m \in \{5, 10\}$ denotes the number of machines in Phase 1, and $s \in \{5, 10\}$ indicates the number of stages in Phase 2. For example, $(15, 5, 5)$ represents an instance size with 15 jobs, 5 machines, and 5 stages. For each instance size, 10 independent instances were randomly generated.

For each instance, 10 initial jobs were assumed to exist in the DECDS-M. The arrival of remaining jobs follows a Poisson distribution, with the interval between consecutive job arrivals following an exponential distribution. DMRLIG was trained on smaller instances ($n = 15$) and evaluated on larger, unseen synthetic instances ($n = 20, 40, 50$) to test its generalization capability across different problem scales. In total, 100 training instances were generated, together with 100 validation instances of the same scale as the training set and 100 test instances for evaluation.

Table C1 The configurations of benchmarks.

Parameters	Range of values
Number of jobs.	$n \in \{15, 20, 40, 50\}$
Number of factories on Phase 1.	$\delta \in \{2, 3, 4, 5\}$
Number of machines on Phase 1.	$m \in \{5, 10\}$
Number of stages on Phase 2.	$s \in \{5, 10\}$
Number of machines at K_k on Phase 2.	$r_k \sim U[2, 5]$
Processing time of $O_{j,i}$ on Phase 1.	$p^1_{j,i} \sim U[10, 30]$
Standard processing time of $O_{j,k}$ at K_k on Phase 2.	$p^2_{j,k} \sim U[10, 50]$
Setup time for changeover from $O_{j,i}$ to $O_{j',i}$ on Phase 1.	$st^1_{j,j',i} \sim U[10, 20]$
Setup time for changeover from $O_{j,k}$ to $O_{j',k}$ at $M^p_{k,l}$ on Phase 2.	$st^2_{j,j',l,k} \sim U[10, 20]$
Transportation time of the J_j from $M_{f,m}$ on Phase 1 to Phase 2.	$tp_{j,f} \sim U[2, 5]$
Speed of $M^p_{k,l}$.	$v_{k,l} \in \{1, 1.1, 1.2, 1.3, 1.4\}$
EC per unit time of $M_{f,i}$ at processing mode on Phase 1.	$\beta^1_i \sim U[4, 8]$
EC per unit time of $M^p_{k,l}$ at processing mode on Phase 2.	$\beta^2_{k,l} \sim U[5, 10]$
EC per unit time at setup mode.	$\alpha = 1$
EC per unit time at idle mode.	$\theta = 0.5$
EC per unit time at transportation phase.	$\mu = 3$
Number of main orders.	$\|JM\| = \lceil 0.75 \times n \rceil$
Number of sub orders.	$\|JS\| = n - \|JM\|$
Tightness factor of J_j.	$\alpha_j \in \{1, 1.5, 2\}$

Appendix C.2 Hyperparameter settings

The hyperparameter configurations for the DMRLIG are summarized in Table C2. Specifically, the total number of training epochs is set to $\mathcal{L} = 1000$, and each iteration concurrently processes 20 instances ($B = 20$). Validation is conducted every 10 iterations using a separate set of validation cases, and new training instances are samples to mitigate the risk of model overfitting.

Table C2 Hyperparameter configurations of the DMRLIG.

Parameters	Value
Number of IG iterations IT	$IT = 10$
Number of HGNN iterations L	$L = 2$
Dimensions of embeddings d	$d = 10$
Hidden dimensions of MLPs within representation learning	$d_\vartheta = d_\eta = 128$
Hidden dimensions of MLPs within PPO	$d_\phi = d_\omega = 64$
Number of training iterations $\mathcal{L}$	$\mathcal{L} = 1000$
Instance batch size B	$B = 20$
Weights of reward function w	$w = 0.8$
Clipping ratio ϵ	$\epsilon = 0.2$
Discount factor γ	$\gamma = 1.0$
Optimizer	Adam
Number of episodes per update R	$R = 3$
Learning rate l_r	$l_r = 2 \times 10^{-4}$

To assess the robustness of DMRLIG, we conducted sensitivity experiments on four key hyperparameters: the reward weight w, discount factor γ, clipping ratio ϵ, and the number of HGNN iterations L. For each parameter, several values were tested while keeping others fixed, and results were averaged over five runs. Table C3 reports the HV results. As shown in Fig. C1, DMRLIG is robust to variations in the major hyperparameters. The reward weight w produces only minor differences in performance, as the IG-based reward shaping compensates for imbalances between TTD and TEC. The discount factor γ influences convergence dynamics but has limited effect on the final Pareto front quality. Similarly, the exploration parameter ϵ mainly affects early-stage stability, with moderate values ($\epsilon = 0.10$) yielding the most consistent results. For the number of HGNN iterations L, the performance improves when moving from 1 to 3 layers, but further increase brings no significant gain while incurring higher computational costs. Overall, these findings confirm that DMRLIG does not rely on precise hyperparameter tuning, and its performance remains stable across a wide range of configurations.

Table C3 Sensitivity analysis of DMRLIG with respect to key hyperparameters.

Parameter	Values tested	Best HV	Worst HV	Mean ± Std	Observation
w	0.6, 0.8, 0.9	0.711	0.706	0.709 ± 0.003	Robust due to IG adaptation
γ	0.95, 0.99, 1.0	0.711	0.708	0.710 ± 0.002	Minor effect on convergence trends
ϵ	0.05, 0.10, 0.20	0.712	0.707	0.710 ± 0.004	Larger ϵ slows early convergence
L	1, 2, 3, 4	0.713	0.704	0.709 ± 0.004	2–3 layers offer best trade-off

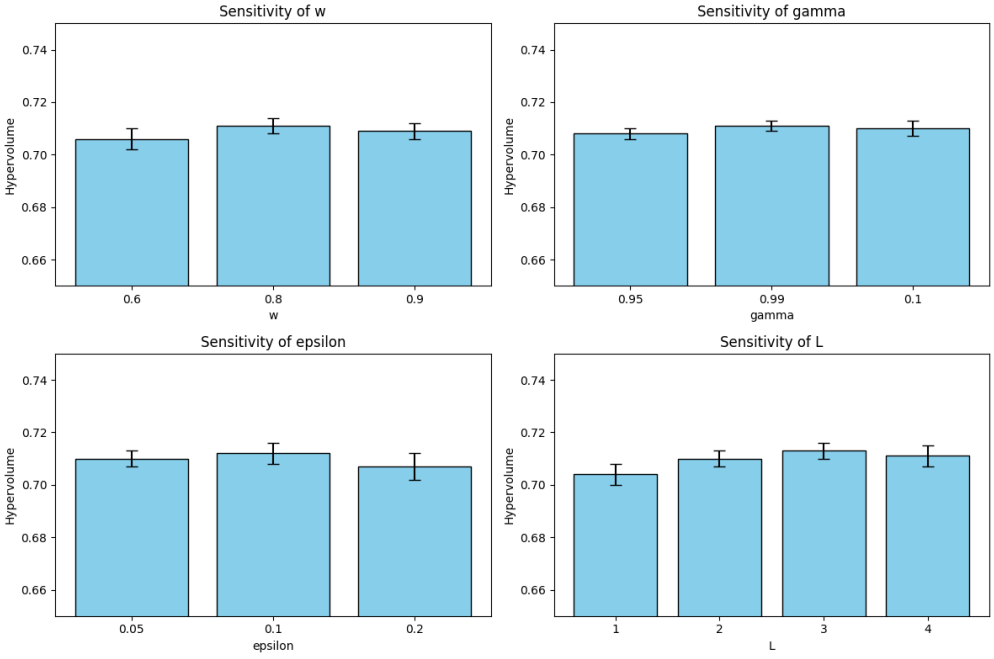

Figure C1 (Color online) Sensitivity analysis of DMRLIG with respect to four hyperparameters.

Appendix C.3 Details of the compared composite rules

In Section 5.1 of the main manuscript, a comparison is made between the proposed DMRLIG and 6 Priority Dispatch Rules (PDRs). Since no existing PDRs have been specifically designed for DECDS-M, two assignment rules and three sequencing rules are applied to form the Composite Rules (CRs) for DECDS-M. The 6 CRs are shown in Table C4. The definition of the assigning rules and the sequencing rules can be found in Table C5.

To ensure a fair comparison, both DDQN and DDPG were re-implemented using the same HGNN-based state representations introduced in Section 4.2 and the same hierarchical action structure (job–factory assignment in Phase 1 and operation–machine assignment in Phase 2).

For DDQN, a two-layer Q-network with ReLU activations was trained with experience replay and a target network, and its output layer corresponded to the Q-values of the feasible discrete actions after action masking.

For DDPG, an actor–critic architecture was used. The actor produced continuous proto–actions that were subsequently projected onto the discrete feasible action set through a softmax-based discretization step, while the critic approximated action-value functions. As noted in recent DRL studies, such projection-based operators inherently place DDPG at a disadvantage in fully discrete action spaces, especially when the feasible set varies across states. We therefore include DDPG as a reference baseline for completeness, but acknowledge this structural limitation.

Both baselines were tuned following standard practice while adapting their learning rates, exploration strategies, and projection mechanisms to the DECDS-M environment. Action masking was consistently applied to ensure constraint satisfaction and to maintain comparability with DMRLIG.

Appendix C.4 Results of ablation studies

The distinctive features of DMRLIG, which set it apart from other baseline models, are its hierarchical multi-action mechanism and the evolutionary operation based on IG. To assess the significance of these components, we conduct an evaluation in which both are deactivated. Instead, we adopt a classical action execution scheme [3], leading to the development of two algorithms: DMRLIG-H and DMRL. These algorithms are used to evaluate the effectiveness of the above two mechanisms respectively. The experimental phase involved evaluating instances within dynamic environments, considering two scales: 20 and 40 jobs. The results of these evaluations are summarized in Table C6.

Table C4 Compared composite rules (CRs).

Abbreviations	Rules
CR1	SNJ + EDD
CR2	SNJ + SPT
CR3	SNJ + CR
CR4	MTP + EDD
CR5	MTP + SPT
CR6	MTP + CR

Table C5 The assigning rules and sequencing rules used in the CRs.

Category	Abbreviations	Description
Assigning rules	SNJ	Assign the job to the factory in Phase 1 (machine in Phase 2) with the smallest number of jobs.
	MTP	Assign the job to the factory in Phase 1 (machine in Phase 2) with the minimum total processing time.
Sequencing rules	EDD	Select the job with the earliest due date.
	SPT	Select the job with the shortest processing time.
	CR	Select the job with the minimum earliest critical ratio.

Table C6 Results of ablation studies (HV, Gap, p-values).

Instances		DMRLIG	DMRLIG-H	DMRL
(20,5,5)	HV	**0.162**	0.147	0.140
	Gap	**0.00%**	-9.37%	-13.71%
	p-values	/	2.32E-02	5.19E-04
(20,5,10)	HV	**0.181**	0.158	0.156
	Gap	**0.00%**	-12.91%	-14.06%
	p-values	/	6.10E-03	2.88E-03
(20,10,5)	HV	**0.159**	0.147	0.145
	Gap	**0.00%**	-7.73%	-8.92%
	p-values	/	4.75E-02	2.63E-02
(20,10,10)	HV	**0.205**	0.192	0.186
	Gap	**0.00%**	-6.34%	-9.43%
	p-values	/	1.42E-01	1.57E-02
(40,5,5)	HV	**0.321**	0.282	0.249
	Gap	**0.00%**	-12.15%	-22.43%
	p-values	/	2.93E-03	2.90E-08
(40,5,10)	HV	**0.194**	0.162	0.146
	Gap	**0.00%**	-16.85%	-25.02%
	p-values	/	9.46E-04	4.68E-06
(40,10,5)	HV	**0.348**	0.292	0.268
	Gap	**0.00%**	-16.04%	-22.94%
	p-values	/	1.20E-08	5.11E-09
(40,10,10)	HV	**0.258**	0.215	0.228
	Gap	**0.00%**	-11.75%	-16.63%
	p-values	/	5.12E-03	5.51E-05

Appendix C.5 Computational complexity analysis

We analyze the computational complexity of DMRLIG by distinguishing the training and inference phases. Let n be the number of jobs, m the number of machines, δ the number of factories, E the number of edges in the heterogeneous graph, d the dimension of node features, L the number of HGNN layers, B the batch size, H the trajectory length, T the total training iterations, and I_{IG} the number of IG local search iterations.

1) Training Phase. The training process involves HGNN-based state encoding, PPO policy/value updates, and IG-based reward shaping.

- **HGNN encoding:** Each message-passing layer aggregates information over E edges with feature dimension d. With L layers, the complexity per step is:

$$O(L \cdot E \cdot d).$$

- **Action sampling and masking:** Each job selects among δ factories and m machines, leading to:

$$O(n \cdot \delta + n \cdot m).$$

- **PPO updates (actor + critic):** With B trajectories of length H, the cost per iteration is:

$$O\big(B \cdot H \cdot (L \cdot E \cdot d + n \cdot \delta + n \cdot m)\big).$$

- **IG reward adaptation:** Each IG iteration reconstructs a subset of operations in $O(n)$, repeated I_{IG} times:

$$O(I_{IG} \cdot n).$$

Thus, the overall training complexity per iteration is:

$$O\Big(B \cdot H \cdot (L \cdot E \cdot d + n \cdot \delta + n \cdot m) + I_{IG} \cdot n\Big).$$

Over T training iterations, the total training cost becomes:

$$O\Big(T \cdot (B \cdot H \cdot (L \cdot E \cdot d + n \cdot \delta + n \cdot m) + I_{IG} \cdot n)\Big).$$

2) Inference Phase. During execution, no gradient updates are required; only forward passes and optional IG adjustments are performed:

$$O(L \cdot E \cdot d + n \cdot \delta + n \cdot m + I_{IG} \cdot n).$$

This shows that inference grows linearly with n, m, and δ, ensuring scalability.

3) Discussion. The training phase is computationally intensive, as expected for deep reinforcement learning, but it can be performed offline. The inference phase is lightweight and suitable for online deployment, with runtimes (e.g., ≈ 13 seconds for $n = 50$) well within practical decision-making windows in industrial production.

Appendix C.6 Details of case study

Table C7 The speed and EC of all machines.

	Machines	$v_{k,l}$	β_i^1 and $\beta_{k,l}^2$	α	θ	μ
Phase 1	$i = 1$ lathe ($M_{f,1}$)	/	$\beta_1^1 = 6$	1	0.5	3
	$i = 2$ miller ($M_{f,2}$)	/	$\beta_2^1 = 7$	1.2	0.7	
	$i = 3$ CNC lathe ($M_{f,3}$)	/	$\beta_2^1 = 8$	1	0.6	
	$k = 1, l = 1$ polishing devices ($M_{1,1}^p$)	1	$\beta_{1,1}^2 = 6$	1.1	0.5	
	$k = 1, l = 2$ polishing devices ($M_{1,2}^p$)	1.2	$\beta_{1,2}^2 = 8$	1.2	0.5	
	$k = 1, l = 3$ polishing devices ($M_{1,3}^p$)	1	$\beta_{1,2}^2 = 5$	1	0.5	
Phase 2	$k = 2, l = 1$ electrical processing machines ($M_{2,1}^p$)	1.2	$\beta_{2,1}^2 = 7$	1	0.8	
	$k = 2, l = 2$ electrical processing machines ($M_{2,2}^p$)	1	$\beta_{2,2}^2 = 6$	1	0.8	
	$k = 2, l = 3$ electrical processing machines ($M_{2,3}^p$)	1	$\beta_{2,2}^2 = 5$	1	0.8	
	$k = 2, l = 4$ electrical processing machines ($M_{2,4}^p$)	1.2	$\beta_{2,3}^2 = 6$	1	0.8	

$v_{f,k,i}$: Speed of $M_{k,l}^p$; β_i^1: EC per unit time of $M_{f,i}$ at processing mode on Phase 1;
$\beta_{k,l}^2$: EC per unit time of $M_{k,l}^p$ at processing mode on Phase 2; α : EC per unit time at setup mode;
θ : EC per unit time at idle mode. μ : EC per unit time at transportation phase.

This empirical validation for this study was conducted in an electronic product manufacturing workshop located in Shandong Province, China. The workshop operates two production lines (F_1 and F_2), each equipped with a lathe ($M_{f,1}$), miller ($M_{f,2}$), CNC lathe ($M_{f,3}$), three polishing devices ($M_{1,1}^p$ $M_{1,3}^p$), and four electrical processing machines ($M_{2,1}^p$ $M_{2,4}^p$). Parts processed in both factories are transported to polishing devices for further processing, with machining times differing across machines for both equipment types. This case study selected four components for experimentation: the casing of TWS (P_1), the acoustic chamber of TWS (P_2), the lens support frame for VR and AR devices (P_3), and the fan hub (P_4). Information on all equipment is outlined in Table. C7. Routine production is characterized by high flexibility in issuing part orders. This flexibility enables the scenario to be modeled as DECDS-M. At the initial production moment (time 0), the workshop system contains five jobs, and a total of 20 jobs will be processed in the subsequent periods, with arrival times following a Poisson distribution. For additional details regarding all jobs, please refer to Table. C8. The DECDS-M scheduling scheme, derived from the production data, is shown in Fig. C2, yielding TTD and TEC values of 740.67 and 9411, respectively.

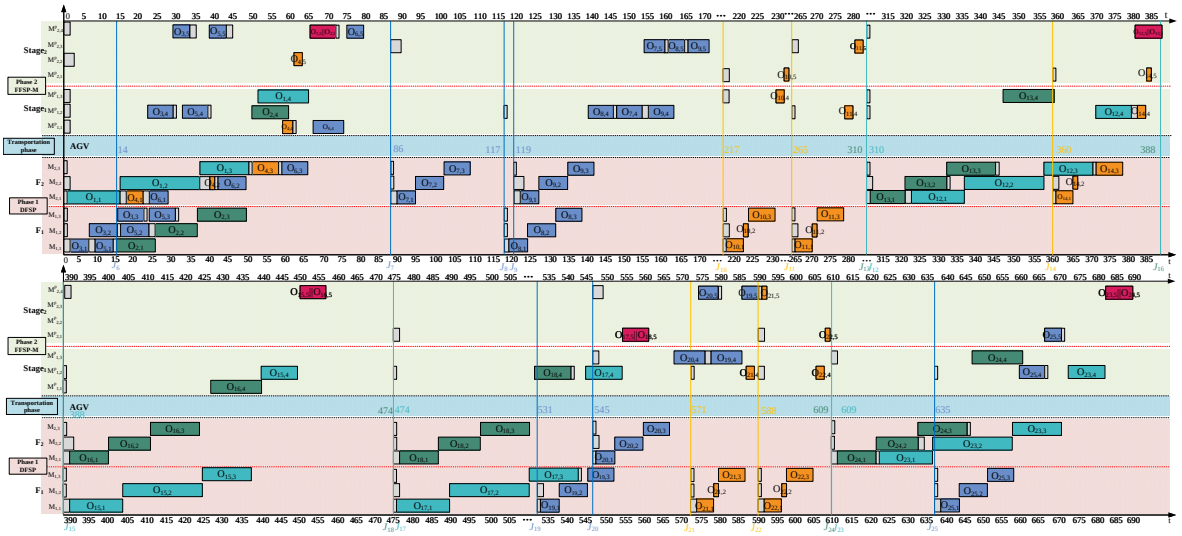

Figure C2 (Color online) Gantt chart of the test case.

Table C8 Processing information of jobs in the case study.

Jobs	Arrival times	Tightness factor	Due date	Processing times in Phase 1/ mins	Processing times in Phase 2/ mins	Order type	Mating operation
J_1 (P_1)	0	1	69	(14, 21, 13)	(13, 8)	Main order	$O_{j,5}$
J_2 (P_2)	0	1	55	(10, 11, 13)	(13, 8)	Suber order	$O_{j,5}$
J_3 (P_3)	0	1.5	49.5	(5, 8, 7)	(8, 5)	Main order	/
J_4 (P_4)	0	1.5	25.5	(4.5, 1,7)	(2.5, 2)	Main order	/
J_5 (P_3)	0	2	66	(5, 8, 7)	(8, 5)	Main order	/
J_6 (P_3)	14	1	47	(5, 8, 7)	(8, 5)	Main order	/
J_7 (P_3)	86	2	152	(5, 8, 7)	(8, 5)	Main order	/
J_8 (P_3)	117	1.5	166.5	(5, 8, 7)	(8, 5)	Main order	/
J_9 (P_3)	119	2	185	(5, 8, 7)	(8, 5)	Main order	/
J_{10} (P_4)	217	1	234	(4.5, 1,7)	(2.5, 2)	Main order	/
J_{11} (P_4)	265	2	299	(4.5, 1,7)	(2.5, 2)	Main order	/
J_{12} (P_1)	310	1.5	413.5	(14, 21, 13)	(13, 8)	Main order	$O_{j,5}$
J_{13} (P_2)	310	1.5	392.5	(10, 11, 13)	(13, 8)	Suber order	$O_{j,5}$
J_{14} (P_4)	360	1.5	385.5	(4.5, 1,7)	(2.5, 2)	Main order	/
J_{15} (P_1)	388	1.5	491.5	(14, 21, 13)	(13, 8)	Main order	$O_{j,5}$
J_{16} (P_2)	388	2	498	(10, 11, 13)	(13, 8)	Suber order	$O_{j,5}$
J_{17} (P_1)	474	1	543	(14, 21, 13)	(13, 8)	Main order	$O_{j,5}$
J_{18} (P_2)	474	2	584	(10, 11, 13)	(13, 8)	Suber order	$O_{j,5}$
J_{19} (P_3)	531	1.5	634.5	(5, 8, 7)	(8, 5)	Main order	/
J_{20} (P_3)	545	2	611	(5, 8, 7)	(8, 5)	Main order	/
J_{21} (P_4)	571	1	588	(4.5, 1,7)	(2.5, 2)	Main order	/
J_{22} (P_4)	588	2	622	(4.5, 1,7)	(2.5, 2)	Main order	/
J_{23} (P_1)	609	1.5	712.5	(14, 21, 13)	(13, 8)	Main order	$O_{j,5}$
J_{24} (P_2)	609	2	719	(10, 11, 13)	(13, 8)	Suber order	$O_{j,5}$
J_{25} (P_3)	635	2	701	(5, 8, 7)	(8, 5)	Main order	/

Appendix D Detailed proofs of Lemma 2, Lemma 3, and Theorem

Appendix D.1 Notation and assumptions.

We analyze DMRLIG under the following notation and assumptions. We consider two objectives $b \in \{1, 2\}$ (total tardiness and total energy consumption). The actor (policy) parameters are ω. For objective b, the true return is

$$J^b(\omega) = \mathbb{E}\Big[\sum_{t=0}^{H-1} \gamma_b^t r_t^b\Big],$$

where H is the (finite) episode horizon used in practice. The clipped PPO surrogate for objective b is

$$L^b_{\mathrm{PPO}}(\omega) = \mathbb{E}_t\Big[\min\big(r_t(\omega)\,\hat{A}^b_t,\ \mathrm{clip}(r_t(\omega), 1-\epsilon_c, 1+\epsilon_c)\,\hat{A}^b_t\big)\Big],$$

with $r_t(\omega) = \dfrac{\pi_\omega(a_t|s_t)}{\pi_{\omega_{\mathrm{old}}}(a_t|s_t)}$ and $\hat{A}^b_t$ the empirical advantage (e.g., GAE).

Assumption 1 (Bounded rewards) For any agent $g \in \mathcal{N}$ and objective $b \in \{1, 2\}$, where $\mathcal{N} = \mathcal{O} \cup \mathcal{F} \cup \mathcal{M}$ represents the agent set, there exists a constant $R > 1$ such that the instantaneous reward r^b_t at time $t \geqslant 0$ satisfies $|r^b_t| \leqslant R$.

Assumption 2 (Exploration / coverage) The policy π_ω induces state-action visitation distributions with sufficient coverage under the on-policy sampling regime used in training. The discounted state-action visitation distribution $\xi^b_{\omega,\rho}(s, a)$ satisfies that

$$\inf_{\omega} \min_{(s,a)\in S\times A} \xi^b_{\omega,\rho}(s, a) > 0.$$

Assumption 3 (Critic approximation) For each objective b, the value-function approximator $V^b(s; \phi)$ (the critic) satisfies a uniform approximation error bound

$$\sup_{s\in\mathcal{S}} \left|V^b(s; \phi) - V^b_\pi(s)\right| \leqslant \varepsilon^{critic},\ \forall b \in \{1, 2\}.$$

where V^b_π is the true value under policy π.

Assumption 4 (Smoothness / Lipschitz). Policy and value networks are sufficiently smooth; in particular log-probability gradients $\nabla_\omega \log \pi_\omega(a|s)$ are uniformly bounded and Lipschitz in ω. We denote a uniform bound $L_{\log \pi}$ on $\|\nabla_\omega \log \pi_\omega(a|s)\|$.

Assumption 5 (Bounded per-iteration parameter change). There exists $\Delta > 0$ such that for all iterations t,

$$\|\omega_{t+1} - \omega_t\| \leqslant \Delta.$$

Appendix D.2 Lemma 1 (Policy gradient)

For any joint policy parameter ω, the gradient of the b-th objective is

$$\nabla_\omega J^b(\omega) = \frac{1}{1-\gamma_b} \mathbb{E}_{s\sim d^b_{\omega,\rho}, a\sim\pi_\omega}\left[\nabla_\omega \log \pi_\omega(a|s)\, A^b_\pi(s, a)\right].$$

Appendix D.3 Lemma 2 (Surrogate gradient approximation in PPO)

Under Assumptions A1–A5, let $\hat{g}^b(\omega) = \nabla_\omega L^b_{\mathrm{PPO}}(\omega)$ denote the population surrogate gradient and $g^b(\omega) = \nabla_\omega J^b(\omega)$ the true policy gradient for objective b. Then for per-update batch size B there exist constants $C_1, C_2, C_3 > 0$ (depending on $R, \epsilon_c, L_{\log\pi}, L_J, H$) such that,

$$\mathbb{E}\big[\|\hat{g}^b(\omega) - g^b(\omega)\|_2^2\big] \leqslant C_1\, \varepsilon^{\mathrm{critic}} + \frac{C_2}{B} + C_3\Delta^2.$$

Here the expectation is taken over the sampling randomness used to estimate gradients (trajectories of length H per rollout and batch aggregation).

Proof of Lemma 2

Recall notation. Per sample (state–action) we write the per-sample policy-gradient-like vectors

$$g^b(s, a) \triangleq \nabla_\omega \log \pi_\omega(a|s)\, A^b_\pi(s, a),$$

and (unclipped surrogate with estimated advantage)

$$\bar{g}^b(s, a) \triangleq \nabla_\omega \log \pi_\omega(a|s)\, \hat{A}^b(s, a),$$

where $\hat{A}^b$ is the estimated advantage (computed by GAE or TD bootstrap) and the clipped-per-sample gradient (population) induced by the PPO surrogate is denoted by $\tilde{g}^b(s, a)$. The population quantities satisfy

$$g^b(\omega) = \mathbb{E}_{s,a}[g^b(s, a)], \qquad \hat{g}^b(\omega) = \mathbb{E}_{s,a}[\tilde{g}^b(s, a)].$$

Proof.

The proof decomposes the gradient discrepancy into three sources: (i) critic-induced advantage estimation bias; (ii) clipping-induced bias from PPO; (iii) finite-sample (Monte Carlo) variance. We bound each term and combine.

Decomposition. We decompose the population-level difference as

$$\hat{g}^b(\omega) - g^b(\omega) = \big(\hat{g}^b(\omega) - \bar{g}^b(\omega)\big) + \big(\bar{g}^b(\omega) - g^b(\omega)\big),$$

i.e. into a *clipping correction* and an *advantage (critic) bias* term. We will bound the mean-square norm of each term and then combine them, adding the finite-sample variance contribution at the end.

Step 1. Bound for the critic / advantage bias term $\bar{g}^b(\omega) - g^b(\omega)$.

Pointwise,

$$\bar{g}^b(s,a) - g^b(s,a) = \nabla_\omega \log \pi_\omega(a|s)\,\big(\hat{A}^b(s,a) - A^b_\pi(s,a)\big).$$

By Assumption A4 we have a uniform bound $\|\nabla_\omega \log \pi_\omega(a|s)\| \leqslant L_{\log\pi}$. Assumption A3 (shared-critic + GAE/TD decomposition) gives that the advantage estimation error $\Delta^b(s,a) := \hat{A}^b(s,a) - A^b_\pi(s,a)$ satisfies $\mathbb{E}_{s,a}[|\Delta^b(s,a)|^2] \leqslant \varepsilon^{\text{critic}}$. Hence, taking expectation and using Cauchy–Schwarz / Jensen,

$$\mathbb{E}\big[\|\bar{g}^b(\omega) - g^b(\omega)\|_2^2\big] \leqslant L^2_{\log\pi}\,\mathbb{E}\big[|\Delta^b(s,a)|^2\big] \leqslant L^2_{\log\pi}\,\varepsilon^{\text{critic}}.$$

Thus the critic-induced contribution to the MSE is $O(\varepsilon^{\text{critic}})$. We may set

$$C_1' := L^2_{\log\pi}, \qquad \text{so} \quad \mathbb{E}\big[\|\bar{g}^b - g^b\|_2^2\big] \leqslant C_1'\varepsilon^{\text{critic}}.$$

Step 2. Bound for the clipping correction term $\hat{g}^b(\omega) - \bar{g}^b(\omega)$.

By definition, for each (s,a) the clipped-per-sample gradient $\tilde{g}^b(s,a)$ equals the unclipped per-sample gradient $\bar{g}^b(s,a)$ except on the event $\mathcal{E}_{s,a}$ where the importance ratio $r_\omega(s,a) = \dfrac{\pi_\omega(a|s)}{\pi_{\omega_{\text{old}}}(a|s)}$ deviates outside $[1-\epsilon_c, 1+\epsilon_c]$. Thus

$$\tilde{g}^b(s,a) - \bar{g}^b(s,a) = \mathbf{1}_{\mathcal{E}_{s,a}} \cdot \Delta^{\text{clip}}(s,a),$$

for some vector $\Delta^{\text{clip}}(s,a)$ whose norm is bounded pointwise by $|\hat{A}^b(s,a)| \cdot \big\|\nabla_\omega \log \pi_\omega(a|s)\big\| + (\text{terms involving } \nabla_\omega r_\omega\big)$. Under A1 (bounded rewards) and standard GAE stability, $\hat{A}^b(s,a)$ is bounded in second moment; combined with A4 this gives a uniform per-sample bound

$$\|\Delta^{\text{clip}}(s,a)\| \leqslant G_{\text{clip}},$$

where G_{clip} depends on $R, \epsilon_c, L_{\log\pi}, H$.

Consequently,

$$\|\hat{g}^b(\omega) - \bar{g}^b(\omega)\|_2 = \big\|\mathbb{E}_{s,a}[\tilde{g}^b(s,a) - \bar{g}^b(s,a)]\big\| \leqslant \mathbb{E}_{s,a}\big[\mathbf{1}_{\mathcal{E}_{s,a}}\|\Delta^{\text{clip}}(s,a)\|\big] \leqslant G_{\text{clip}} \cdot \Pr(\mathcal{E}),$$

where $\Pr(\mathcal{E})$ is the population probability that r_ω lies outside the clip interval.

To bound $\Pr(\mathcal{E})$ we use Assumption A5 (controlled per-step update): $\|\omega - \omega_{\text{old}}\| \leqslant \Delta$. Under smoothness of $\log \pi_\omega(a|s)$ (A4) and a first-order Taylor expansion of r_ω around ω_{old},

$$|r_\omega(s,a) - 1| \leqslant L_r\|\omega - \omega_{\text{old}}\| \leqslant L_r\Delta,$$

for some Lipschitz constant L_r depending on $L_{\log\pi}$ and the policy parameterization. Thus if $L_r\Delta \leqslant \epsilon_c$ then $\Pr(\mathcal{E})$ is (approximately) zero; more generally $\Pr(\mathcal{E})$ scales linearly with Δ. Therefore there exists a constant C_{clip} (depending on $G_{\text{clip}}, L_r, \epsilon_c$) such that

$$\|\hat{g}^b(\omega) - \bar{g}^b(\omega)\|_2 \leqslant C_{\text{clip}}\,\Delta.$$

Squaring and taking expectation yields

$$\mathbb{E}\big[\|\hat{g}^b(\omega) - \bar{g}^b(\omega)\|_2^2\big] \leqslant C^2_{\text{clip}}\,\Delta^2.$$

Set $C_3 := C^2_{\text{clip}}$ to match the statement.

Step 3. Finite-sample (Monte Carlo) variance term.

Thus far the bounds are population-level. In practice we compute empirical gradients from a finite batch of B (independent or weakly dependent) trajectories. Let $\widehat{\hat{g}}^b$ be the empirical estimator of the surrogate gradient and $\widehat{g}^b$ that of the true gradient. Standard concentration / variance bounds for i.i.d. samples with bounded second moments give

$$\mathbb{E}\big[\|\widehat{\hat{g}}^b - \hat{g}^b\|_2^2\big] \leqslant \frac{\sigma^2_{\hat{g}}}{B}, \qquad \mathbb{E}\big[\|\widehat{g}^b - g^b\|_2^2\big] \leqslant \frac{\sigma^2_g}{B},$$

for variance constants $\sigma^2_{\hat{g}}, \sigma^2_g$ that depend on the per-sample second moments of the per-sample gradients. Combining the two and noting that we compare empirical surrogate gradient to the population true gradient, the finite-sample contribution can be bounded by $\dfrac{C_2}{B}$ for an appropriate C_2.

Step 4. Combine the three contributions.

Using the decomposition and triangle / Minkowski inequalities,

$$\begin{aligned}\mathbb{E}\big[\|\hat{g}^b(\omega) - g^b(\omega)\|_2^2\big] &\leqslant 3\Big(\mathbb{E}\big[\|\hat{g}^b(\omega) - \bar{g}^b(\omega)\|_2^2\big] + \mathbb{E}\big[\|\bar{g}^b(\omega) - g^b(\omega)\|_2^2\big] + \mathbb{E}\big[\|\widehat{\hat{g}^b} - \hat{g}^b\|_2^2\big]\Big) \\ &\leqslant 3\big(C_1'\varepsilon^{\text{critic}} + C_3\Delta^2 + C_2/B\big),\end{aligned}$$

where constants have been absorbed (the factor 3 is from the elementary inequality $\|x+y+z\|^2 \leqslant 3(\|x\|^2 + \|y\|^2 + \|z\|^2)$). Renaming $3C_1' \to C_1$ yields the claimed bound:

$$\mathbb{E}\big[\|\hat{g}^b(\omega) - g^b(\omega)\|_2^2\big] \leqslant C_1\,\varepsilon^{\text{critic}} + \frac{C_2}{B} + C_3\Delta^2.$$

This completes the proof.

Appendix D.4 Lemma 3 (MGDA-based multi-objective descent)

Let $g_t^b = \nabla_\omega L_{\text{PPO}}^b(\omega_t)$ denote the surrogate gradients at iteration t. MGDA computes:

$$\lambda_t = \arg\min_{\lambda \geqslant 0,\ \|\lambda\|_1 = 1} \left\| \sum_{b\in\{1,2\}} \lambda_b g_t^b \right\|_2^2.$$

Properties of the MGDA Combination.

At iteration t, let $\hat{g}_t^b$ be the PPO surrogate gradient for objective $b \in \{1,2\}$. The multi-objective weight vector λ_t is obtained by

$$\lambda_t = \arg\min_{\lambda\in\Delta_2} \left\| \sum_{b=1}^{2} \lambda_b\, \hat{g}_t^b \right\|_2^2, \qquad \Delta_2 = \{\lambda \in \mathbb{R}_+^2 : \lambda_1 + \lambda_2 = 1\}.$$

Then λ_t satisfies the following properties:

1. **Convexity and Feasibility:** $\lambda_t \in \Delta_2$, and hence $\sum_b \lambda_{t,b} = 1$ and $\lambda_{t,b} \geqslant 0$.

2. **Minimal-Norm Combination:** λ_t gives the minimum-norm convex combination of the surrogate gradients, i.e.,

$$\Big\| \sum_{b=1}^{2} \lambda_{t,b}\hat{g}_t^b \Big\|_2^2 \ \leqslant\ \Big\| \sum_{b=1}^{2} \lambda_b \hat{g}_t^b \Big\|_2^2, \qquad \forall \lambda \in \Delta_2.$$

3. **Alignment with the True Multi-Objective Descent Direction:** Let $g_t^b = \nabla_\omega J^b(\omega_t)$ be the true gradients. Then,

$$\left\langle \sum_{b=1}^{2} \lambda_{t,b} g_t^b,\ \sum_{b=1}^{2} \lambda_{t,b} \hat{g}_t^b \right\rangle \geqslant 0.$$

   That is, the surrogate direction is a descent direction for the true multi-objective gradient in expectation.

These properties ensure that the update direction chosen by MGDA is a valid first-order descent direction within the feasible convex hull of the per-objective gradients.

Appendix D.5 Remark (Convergence of Adaptive IG Weights w_t)

Consider the adaptive weight sequence $\{w_t\}$ produced by the IG-based reward shaping controller. Suppose the IG adaptation rule satisfies a diminishing-adaptation condition (i.e., updates to w_t shrink over time and are bounded) and the improvement signal used for adaptation is bounded. Then the sequence $\{w_t\}$ is bounded and convergent: there exists w^* with $\lim_{t\to\infty} w_t = w^*$.

Proof. The proof uses standard arguments for bounded monotone sequences under diminishing step sizes.

Let the IG update be written abstractly as

$$w_{t+1} = w_t + \alpha_t\, u_t,$$

where u_t is the normalized improvement signal (bounded) and $\alpha_t > 0$ is the adaptation step size satisfying $\sum_t \alpha_t < \infty$ and $\alpha_t \to 0$. The boundedness of u_t follows from bounded rewards and bounded advantage estimates. Then the series $\sum_t \alpha_t u_t$ converges absolutely, implying that $\{w\}$ is Cauchy and thus converges to a finite limit w^*. Additionally, if the adaptation enforces simplex constraints (nonnegative weights summing to one), projection ensures boundedness at each step. Hence the claim.

Appendix D.6 Theorem (Pareto-Stationary Convergence of PPO-based DMRLIG)

Since the policy is restricted by feasibility-preserving masking, the optimization dynamics take place entirely within the feasible action manifold. Therefore, Pareto-stationarity is understood with respect to the constrained feasible set.

At iteration t, DMRLIG computes a convex combination $\lambda_t \in \Delta_2$ via

$$\lambda_t = \arg\min_{\lambda\in\Delta_2} \Big\| \sum_{b=1}^{2} \lambda_b \, \hat{g}_t^b \Big\|_2^2,$$

and updates the actor parameters by

$$\omega_{t+1} = \omega_t - \eta_t \sum_{b=1}^{2} \lambda_{t,b} \, \hat{g}_t^b.$$

Under Assumptions A1–A5 and a diminishing step-size sequence $\{\eta_t\}$ satisfying $\sum_t \eta_t = \infty,\ \sum_t \eta_t^2 < \infty$, there exists a universal constant $C > 0$ such that, for an index $\hat{T}$ sampled uniformly from $\{1, \ldots, T\}$,

$$\mathbb{E}\left[\min_{\lambda\in\Delta_2} \Big\| \sum_{b=1}^{2} \lambda_b \, \nabla_\omega J^b(\omega_{\hat{T}}) \Big\|_2^2 \right] \leqslant \frac{C}{T} + C_1 \varepsilon^{\text{critic}} + \frac{C_2}{B} + C_3 \Delta^2.$$

Hence DMRLIG converges to an ε-Pareto-stationary point, with

$$\varepsilon = C_1 \varepsilon^{\text{critic}} + \frac{C_2}{B} + C_3 \Delta^2,$$

which can be made arbitrarily small by increasing critic accuracy, batch size, and by reducing the clipping bias parameter Δ.

Proof of Theorem

Our Pareto-stationary convergence proof follows the stochastic approximation framework commonly used in distributed neural policy gradient algorithms [4]. The proof follows a standard template for stochastic multi-objective optimization based on (i) per-objective gradient approximation (Lemma 2), (ii) bounded and convergent weight selection (Remark), and (iii) diminishing step-size stochastic approximation arguments.

Step 1. Multi-objective update direction and MGDA alignment. Recall that

$$d_t = \sum_{b=1}^{2} \lambda_{t,b} \, \hat{g}_t^b \quad \text{and} \quad g_t^b = \nabla_\omega J^b(\omega_t).$$

By Lemma 3 (properties of MGDA), the chosen $\lambda_t \in \Delta_2$ satisfies

$$\Big\| \sum_{b=1}^{2} \lambda_{t,b} \, \hat{g}_t^b \Big\|_2^2 \leqslant \Big\| \sum_{b=1}^{2} \lambda_b \, \hat{g}_t^b \Big\|_2^2 \quad \forall \lambda \in \Delta_2, \tag{D1}$$

and moreover,

$$\Big\langle \sum_{b=1}^{2} \lambda_{t,b} \, g_t^b, \ \sum_{b=1}^{2} \lambda_{t,b} \, \hat{g}_t^b \Big\rangle \geqslant 0. \tag{D2}$$

The inequality (D2) expresses that the surrogate MGDA direction is aligned (in expectation) with the true MGDA direction.

Step 2. Decomposition of the surrogate gradient error. Lemma 2 (surrogate gradient approximation) gives

$$\mathbb{E}\left[\|\hat{g}_t^b - g_t^b\|_2^2 \right] \leqslant C_1 \varepsilon^{\text{critic}} + \frac{C_2}{B} + C_3 \Delta^2,$$

which holds uniformly over objectives b and over t.

Define the total surrogate gradient error $\xi_t^b = \hat{g}_t^b - g_t^b$. Then

$$\mathbb{E}\|\xi_t^b\|_2^2 \leqslant C_1 \varepsilon^{\text{critic}} + \frac{C_2}{B} + C_3 \Delta^2.$$

By convexity of the squared norm and $\lambda_t \in \Delta_2$, we also have

$$\mathbb{E}\left[\Big\| \sum_b \lambda_{t,b} \xi_t^b \Big\|_2^2 \right] \leqslant C_1 \varepsilon^{\text{critic}} + \frac{C_2}{B} + C_3 \Delta^2.$$

Step 3. Expected descent using smoothness of each objective. By L_J-smoothness of $J^b(\omega)$ (Assumption A4),

$$J^b(\omega_{t+1}) \leqslant J^b(\omega_t) + \langle g_t^b, \omega_{t+1} - \omega_t \rangle + \frac{L_J}{2} \|\omega_{t+1} - \omega_t\|_2^2.$$

Summing over b weighted by $\lambda_{t,b}$ gives

$$\sum_b \lambda_{t,b} J^b(\omega_{t+1}) \leqslant \sum_b \lambda_{t,b} J^b(\omega_t) - \eta_t \Big\langle \sum_b \lambda_{t,b} g_t^b,\ \sum_b \lambda_{t,b} \hat{g}_t^b \Big\rangle + \frac{L_J \eta_t^2}{2} \Big\| \sum_b \lambda_{t,b}\, \hat{g}_t^b \Big\|_2^2.$$

Taking expectation and using (D2) yields

$$\mathbb{E}\left[\sum_b \lambda_{t,b} J^b(\omega_{t+1})\right] \leqslant \mathbb{E}\left[\sum_b \lambda_{t,b} J^b(\omega_t)\right] - \eta_t\, \mathbb{E}\left[\Big\| \sum_b \lambda_{t,b}\, g_t^b \Big\|_2^2\right] + \eta_t\, \mathbb{E}\left[\Big\| \sum_b \lambda_{t,b} \xi_t^b \Big\|_2^2\right] + L_J \eta_t^2\, \mathbb{E}\left[\Big\| \sum_b \lambda_{t,b} \hat{g}_t^b \Big\|_2^2\right].$$

Using Lemma 3 again and the uniform bound $\|\hat{g}_t^b\| \leqslant G$ from Assumption A5,

$$\mathbb{E}\left[\Big\| \sum_b \lambda_{t,b} \hat{g}_t^b \Big\|_2^2\right] \leqslant G^2.$$

Thus,

$$\eta_t\, \mathbb{E}\left[\Big\| \sum_b \lambda_{t,b}\, g_t^b \Big\|_2^2\right] \leqslant \mathbb{E}\Big[\sum_b \lambda_{t,b} J^b(\omega_t) - \sum_b \lambda_{t,b} J^b(\omega_{t+1})\Big] + \eta_t \left(C_1 \varepsilon^{\text{critic}} + \frac{C_2}{B} + C_3 \Delta^2\right) + L_J \eta_t^2 G^2.$$

Step 4. Summing over $t = 1, \ldots, T$ and normalization. Summing yields

$$\sum_{t=1}^T \eta_t\, \mathbb{E}\left[\Big\| \sum_b \lambda_{t,b}\, g_t^b \Big\|_2^2\right] \leqslant \sum_{t=1}^T \mathbb{E}\Big[\sum_b \lambda_{t,b} J^b(\omega_t) - \sum_b \lambda_{t,b} J^b(\omega_{t+1})\Big] + \left(C_1 \varepsilon^{\text{critic}} + \frac{C_2}{B} + C_3 \Delta^2\right) \sum_{t=1}^T \eta_t + L_J G^2 \sum_{t=1}^T \eta_t^2.$$

The telescoping sum on the right is bounded because $J^b(\omega)$ is bounded (Assumption A1), and $\sum_t \eta_t^2 < \infty$. Let C' denote the resulting constant bound. Then,

$$\sum_{t=1}^T \eta_t\, \mathbb{E}\left[\Big\| \sum_b \lambda_{t,b}\, g_t^b \Big\|_2^2\right] \leqslant C' + \left(C_1 \varepsilon^{\text{critic}} + \frac{C_2}{B} + C_3 \Delta^2\right) \sum_{t=1}^T \eta_t.$$

Since $\sum_{t=1}^T \eta_t \asymp T$ for standard diminishing step sizes, dividing both sides by $\sum_{t=1}^T \eta_t$ yields

$$\mathbb{E}\Big[\frac{\sum_{t=1}^T \eta_t \Big\| \sum_b \lambda_{t,b}\, g_t^b \Big\|_2^2}{\sum_{t=1}^T \eta_t}\Big] \leqslant \frac{C'}{T} + C_1 \varepsilon^{\text{critic}} + \frac{C_2}{B} + C_3 \Delta^2.$$

Finally, choosing $\hat{T}$ uniformly in $\{1, \ldots, T\}$ gives

$$\mathbb{E}\left[\min_{\lambda \in \Delta_2} \Big\| \sum_{b=1}^2 \lambda_b\, \nabla_\omega J^b(\omega_{\hat{T}}) \Big\|_2^2\right] \ \leqslant\ \frac{C}{T} + C_1 \varepsilon^{\text{critic}} + \frac{C_2}{B} + C_3 \Delta^2,$$

as claimed.

References

1 Du Y, Li J, Li C, et al. A reinforcement learning approach for flexible job shop scheduling problem with crane transportation and setup times. IEEE Transactions on Neural Networks and Learning Systems, 2022, 35(4): 5695-5709

2 Huang J, Gao L, Li X. A hierarchical multi-action deep reinforcement learning method for dynamic distributed job-shop scheduling problem with job arrivals. IEEE Transactions on Automation Science and Engineering, 2025, 22: 2501-2513

3 Song W, Chen X, Li Q, et al. Flexible job-shop scheduling via graph neural network and deep reinforcement learning. IEEE Transactions on Industrial Informatics, 2022, 19(2): 1600-1610

4 Dai P, Mo Y., Yu W., et al. Distributed neural policy gradient algorithm for global convergence of networked multi-agent reinforcement learning. 2025. ArXiv:2505, 24113