Supplementary File for Paper: Balancing Disassembly Line by Considering Human Learning Effect

Shujin Qin, *Member, IEEE,* JianPing Wang, MengChu Zhou, *Fellow, IEEE*, Jiacun Wang, *Senior Member, IEEE,* XiWang Guo, *Senior Member, IEEE,* Liang Qi, *Member, IEEE*

This Supplementary File includes the following two sections:

Section I provides two tables, respectively corresponding to the full forms of acronyms and the symbols explanation. Section II is dedicated to a series of illustrative figures that provide insights into the intricacies of the improved carnivorous plant algorithm (ICPA). Section III showcases a comprehensive array of tables and graphical representations that are associated with the obtained experimental findings.

I. Symbols and Acronyms

Table S1 provides the expanded versions of all acronyms used.

Table S2 catalogues all symbols featured in the paper, offering detailed explanations for each individual symbol.

TABLE S1
GLOSSARY OF ACRONYMS

Acronym	Full name
CDP-L	circular layout disassembly line balacing problem considering the learning effects of the workers
CPA	carnivorous plant algorithm
DAOG	disassembly AND/OR graph
DLBP	disassembly line balancing problem
DMBO	discrete migrating birds optimization
DOA	dingo optimization algorithm
FOA	fruit fly optimization algorithm
ICPA	improved carnivorous plant algorithm
S.F.	supplementary file
WOA	whale optimization algorithm

II. Illustrative Diagrams of ICPA

This section provides diagrams of the encoding process and key functional operators for the ICPA.

A. Encoding process

As shown in Fig. S1, taking the treadmill example, We can easily obtain the task sequence corresponding to σ_1 as 1, 4, 10, 15, 12, 16. Tasks 1 and 15 are assigned to workstation 1, tasks 4 and 12 are assigned to workstation 2, and tasks 10 and 16 are assigned to workstation 3. σ_2 represents the mapping relationship between tasks and workstations. It can be represented as:

TABLE S2
SYMBOLS EXPLANATION REFERENCE

Symbol	Description
(symbols in the formulation section)	
$\mathbb{W}$	index set of workstations
$\mathbb{S}$	index set of skills
$\mathbb{R}$	index set of workers
$\mathbb{P}$	index set of products
$\mathbb{K}$	index set of positions on each workstation
$\mathbb{L}$	index set of skill level index
$\mathbb{J}$	subassembly set of product
$\mathbb{I}$	task set of product
$\mathbb{I}_{pi}^{\mathrm{con}}$	task set that conflicts with task *i* of product *p*
$\mathbb{I}_{pi}^{\mathrm{pre}}$	immediately preceding task set of task *i* of product *p*
T_{pirl}	the time needed for performing task *i* on product *p* by the *r*-th worker with skill level *l*
E_{rs}^{0}	the initial experience of *r-th* worker's skill *s*
E_{sl}^{lb}	the lower bound of experience required for skill *s* at level *l*
C_w	startup cost of workstation *w*
V_{pj}	value of subassembly *j* of product *p*
α_s	learning effect coefficient of skill *s*
C_T	the cost incurred per unit of time for disassembly activities
M	a sufficiently large number
x_{piwk}	If task *i* of product *p* is executed at the *k*-th position on workstation *w*, x_{piwk} = 1; otherwise, x_{piwk} = 0
z_{rw}	If the *r*-th worker is assigned to the *w*-th workstation, z_{rw}=1; otherwise, z_{rw} = 0
u_w	If the *w*-th workstation is started,u_w = 1; otherwise, u_w= 0
y_{piwk}^{rsl}	if task *i* of product *p* is executed by *r*-th worker at the *k*-th position on workstation *w* and the level of skill *s* is *l*, then y_{piwk}^{rsl} = 1; otherwise, y_{piwk}^{rsl} = 0
e_{wks}	experience corresponding to the skill *s* of the worker at workstation *w* at position *k*
s_{piwk}	start time of task *i* of product *p* at position *k* on workstation *w*
(symbols in the algorithm section)	
$\widehat{c}$	the number of carnivorous plants
$\widetilde{c}$	the number of preys
σ	the three-segment encoding which represents a solution
σ_1	the 1st segment encoding of σ which indicates a disassembly task sequence
σ_2	the 2nd segment encoding of σ which implies the matching scheme between tasks and workstations
σ_3	the 3rd segment encoding of σ which corresponds to the worker-to-workstation assignment scheme

$\sigma_2 = \{(1,1),(15,1),(4,2),(12,2),(10,3),(16,3)\}$. Workers 4, 2, and 3 are assigned to workstations 1, 2, and 3, respectively. σ_3 represents the mapping relationship between workers and workstations and it can be represented as: $\sigma_3 = \{(4,1),(2,2),(3,3)\}$.

B. Processes of predation and growth

Fig. S2 illustrates the process of predation operator. Fig. S3-S6 demonstrate the processes of four growth operators.

As shown in Fig. S3, task 12 is randomly selected. we find the immediately preceding task 3 and the immediately following task 21 of task 12. Then the positions between tasks 3 and 21 are all the positions that task 12 can choose. Then we randomly insert the randomly selected task 12 into a new position.

As shown in Fig. S4, task 5 is randomly selected and then deleted with its successor tasks. According to the AND/OR figure, we find out the components disassembled by task 5 and select another disassembly method for this component to obtain disassembly tasks 6, 16, 18, and 20. Finally, these disassembly tasks are added to the original disassembly sequence.

As shown in Fig. S5, workstation 1 is a randomly selected one. Then we cancel the assignment of the original workers in this workstation and re-select an unassigned worker to assign to this workstation.

As shown in Fig. S6, we select the workstation with the minimum allocation of tasks and randomly assign the tasks of the workstation to another workstation.

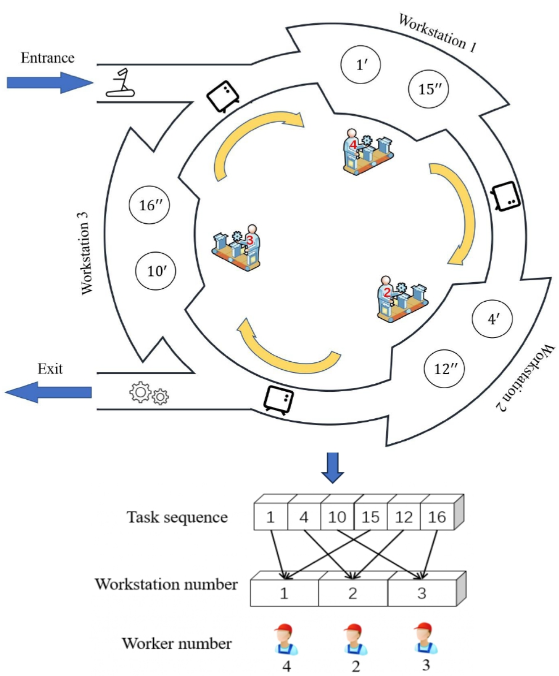

Fig. S1. Schematic diagram of coding structure.

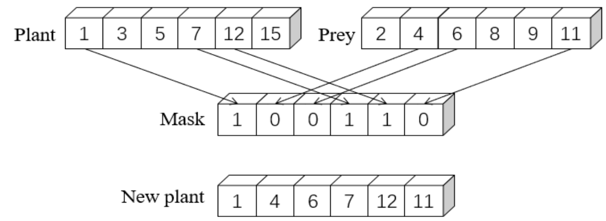

Fig. S2. Predation process.

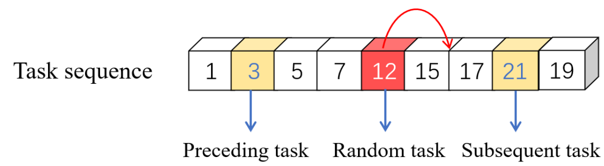

Fig. S3. Sequential variation.

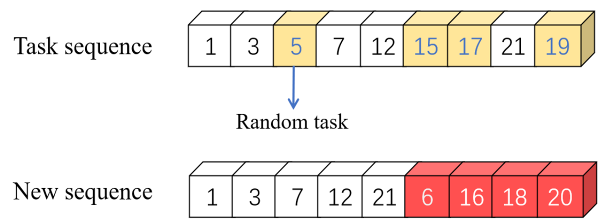

Fig. S4. Task variation.

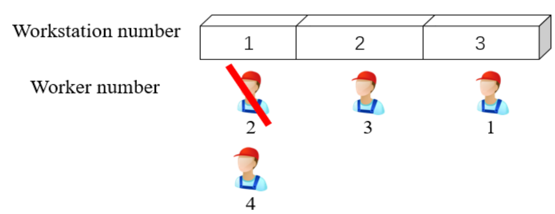

Fig. S5. Redistribution of workers.

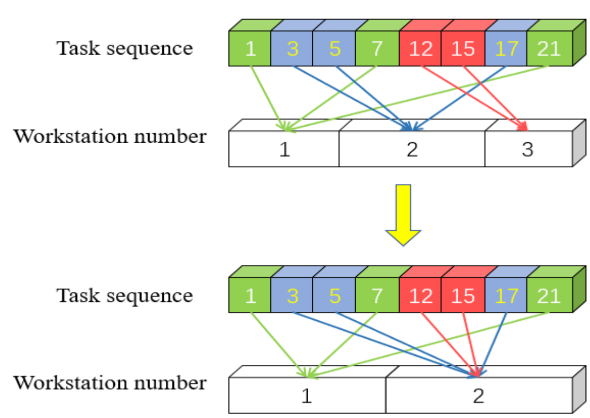

Fig. S6. Reassignment between workstations and tasks.

TABLE S3
CPLEX SOLUTIONS

Case #	Disassembly sequence	Profit (Best Bound)	Solution status	Gap	Computing time(s)
1	$(10, 6) \rightarrow 6, (7) \rightarrow 1, (1, 9, 3) \rightarrow 2$	595 (595)	optimal	0.00%	87.9
2	$(19, 9) \rightarrow 2, (11, 13, 16, 17, 1, 3, 20, 10, 6, 7) \rightarrow 6$	1195 (1528)	feasible	27.93%	14400.0
3	$(1, 18, 12, 5, 22, 11, 28, 29, 33) \rightarrow 5, (16, 34, 17) \rightarrow 3$	1502 (1714)	feasible	14.11%	14400.0
4-6	—a	—	—	—	—

For Cases 4-6, CPLEX can't find a feasible solution within the set 4 hours.

TABLE S4
ICPA SOLUTIONS

Case #	Iterations / Population	Disassembly sequence	Profit	Computing time(s)
1	100 / 50	$(7) \rightarrow 1, (1, 9, 3) \rightarrow 2, (10, 6) \rightarrow 6$	595	0.07
2	200 / 50	$(17, 7) \rightarrow 4, (1, 9, 11, 19, 3, 13) \rightarrow 2, (10, 20, 16, 6) \rightarrow 3$	1266	0.15
3	200 / 50	$(18, 29, 22, 28, 1, 5, 11, 12) \rightarrow 5, (34, 16, 17, 33) \rightarrow 3$	1525	0.54
4	200 / 50	$(1, 4, 12, 21, 31, 11, 24, 33) \rightarrow 4, (2, 14, 19, 29, 41, 45, 28) \rightarrow 2, (7, 36, 18, 33, 23, 44) \rightarrow 3$	2087	0.88
5	800 / 100	$(1, 4, 12, 21, 31, 11, 24, 33) \rightarrow 4, (47, 58, 2, 51, 57, 19, 29, 41, 45, 14, 28) \rightarrow 5, (62, 63, 7, 18, 36, 39, 44, 23) \rightarrow 3$	3025	2.83
6	800 / 100	$(88, 100, 1, 4, 12, 11, 59, 21, 31, 24, 33, 66, 116, 53) \rightarrow 4,$ $(97, 2, 70, 19, 74, 80.29.28, 113, 14, 81, 47, 55, 49, 41, 45) \rightarrow 5,$ $(106, 7, 110, 18, 36, 115, 95, 57, 63, 68, 39, 85, 52, 86, 44, 23, 56) \rightarrow 6$	5889	6.67

TABLE S5
ALGORITHM PERFORMANCE COMPARISON

Case #	Optimal profit						Excellent rate						Computing time(s)					
	ICPA	CPA	FOA	DMBO	DOA	WOA	ICPA	CPA	FOA	DMBO	DOA	WOA	ICPA	CPA	FOA	DMBO	DOA	WOA
1	**595**	**595**	**595**	**595**	**595**	**595**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	0.12	0.14	0.17	0.14	0.09	0.04
2	**1266**	**1266**	**1266**	**1266**	**1266**	**1266**	**100%**	**100%**	**100%**	50%	80%	35%	0.57	0.56	0.55	0.43	0.52	0.06
3	**1525**	**1525**	**1525**	**1525**	**1525**	**1525**	**100%**	**100%**	90%	85%	90%	75%	0.74	0.76	0.78	0.75	0.70	0.12
4	**2087**	**2087**	**2087**	**2087**	**2087**	**2087**	**100%**	90%	95%	55%	35%	5%	0.93	0.90	0.96	0.95	0.80	0.15
5	**3025**	**3025**	**3025**	3023	**3025**	3020	15%	20%	**30%**	0%	10%	0%	3.64	3.54	6.44	5.42	3.04	0.51
6	**5889**	5888	5884	5881	5880	5869	**5%**	0%	0%	0%	0%	0%	7.89	7.74	8.04	9.74	6.52	1.30

III. EXPERIMENTAL RESULTS

This section consist of three tables and four figures, specifically crafted to illustrate the experimental outcomes.

A. CPLEX vs ICPA

Table S3 and S4 sequentially display the experimental outcomes obtained from solving cases 1-6 via CPLEX and ICPA, respectively.

B. Comparison of different configurations

Fig. S7 visually compares the variations in task allocation in relation to differing workstation costs.

Fig. S8 juxtaposes task allocation in circular versus linear layouts, elucidating the benefits of the circular configuration.

C. Comparison with peer algorithms

Fig. S9 illustrates the superior performance of the ICPA algorithm in seeking optimal solutions compared to the other five algorithms, Carnivorous Plant Algorithm(CPA), Discrete Migrating Birds Optimization(DMBO), Dingo Optimization Algorithm(DOA), Whale Optimization Algorithm(WOA), and Fruit fly Optimization Algorithm(FOA).

Table S5 and Fig. S10 collectively highlight ICPA's advantage in terms of stability over the other five algorithms.

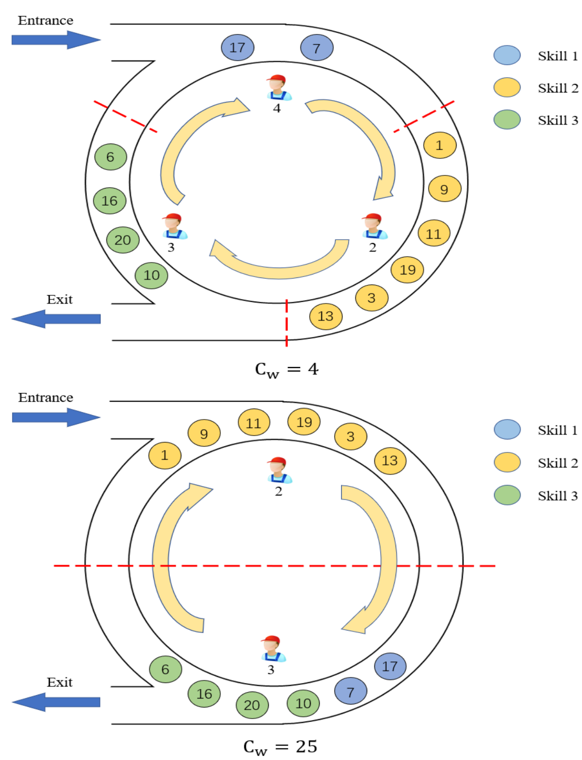

Fig. S7. Task assignments of Case 2 under different workstation costs.

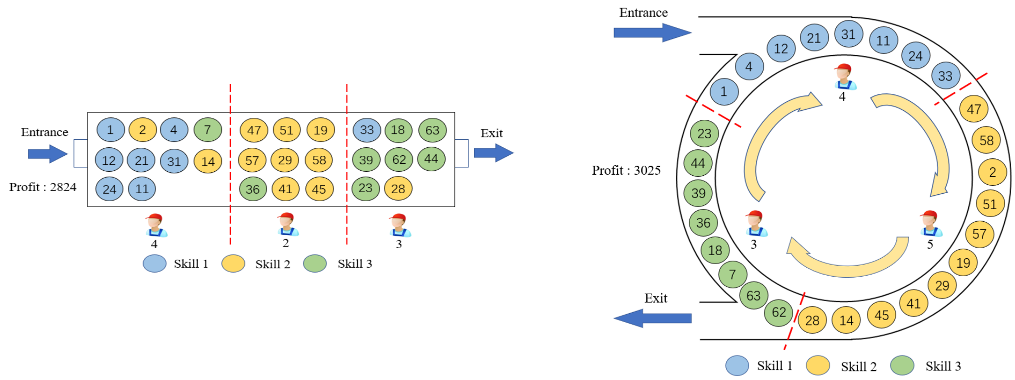

Fig. S8. Task assignments under the linear and circular layout of Case 5.

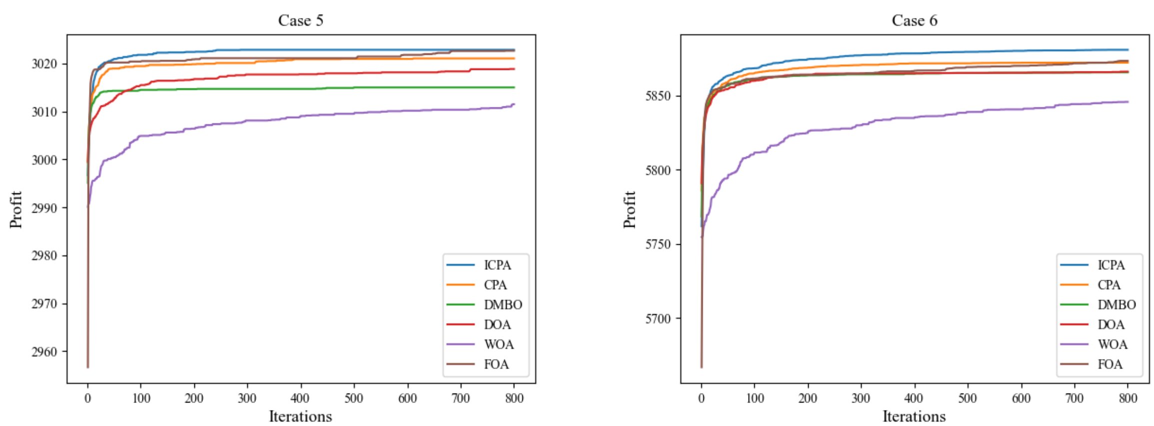

Fig. S9. Iterative comparison.

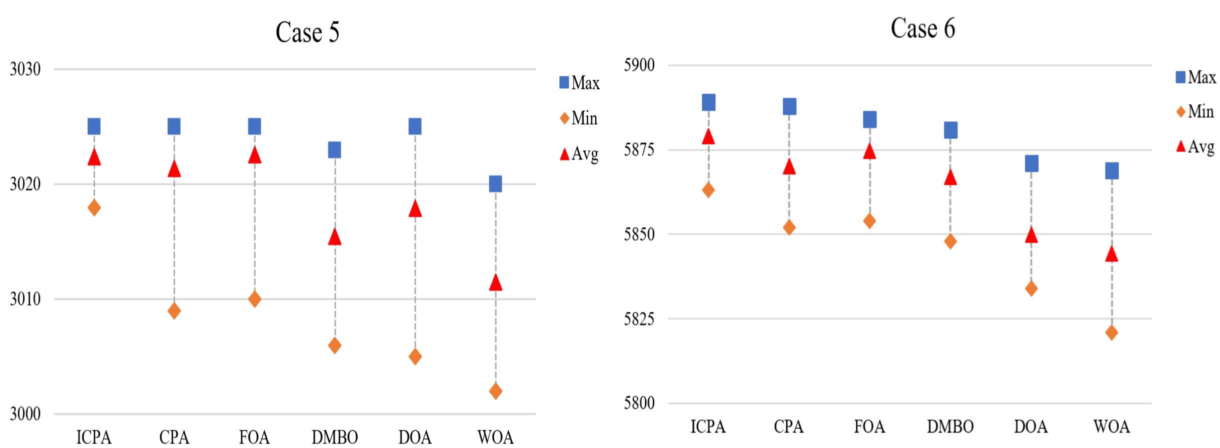

Fig. S10. Stability test.