Improved Adaptive Integer form of Population-Based Incremental Learning and Reactive Tabu Search for an Integrated Energy Supply and Demand Optimization Framework in Factories

Shuhei Kawaguchi
Corporate R&D Headquarters in Mitsubishi Electric Co., Ltd and Meiji University
Kanagawa, Japan
Kawaguchi.Shuhei@ds.MitsubishiElectric.co.jp

Yoshikazu Fukuyama
Graduate School of Advanced Mathematical Sciences Meiji University
Tokyo, Japan
yfukuyama@meiji.ac.jp

***Abstract*— This paper proposes improved adaptive integer form of population-based incremental learning and reactive tabu search (IAIPBIL-RTS) for an integrated energy supply and demand optimization framework in factories. The proposed IAIPBIL-RTS is applied to the production scheduling optimization problem, namely the primary problem of the demand side in factories. As a sub-problem, the optimal operational planning problem of energy plants of the supply side in factories is solved simultaneously. It is confirmed that production costs of a whole factory can be minimized, and the high-quality solution can be obtained when the proposed IAIPBIL-RTS based method is applied.**

Keywords*— Optimal production scheduling, optimal operational planning of energy plants, carbon neutrality, hierarchical combinatorial optimization problem

I. Introduction

The Sustainable Development Goals, particularly Goals 7 and 13, have driven global efforts towards achieving carbon neutrality [1]. In order to achieve carbon neutrality, it is necessary to reduce electricity consumption, which is one of the main sources of CO_2 emissions. In 2020 in Japan, energy consumption in the manufacturing industry accounted for about 42.2% of total energies in all sectors [2]. It represents the highest among all sectors. Therefore, energy reduction efforts in the manufacturing industry are crucial for achieving carbon neutrality. Furthermore, from a managerial perspective, it is crucial to reduce labor costs and energy purchase costs associated with operating factory production sites.

Some medium or large-scale factories have energy plants consisting of turbo refrigerators, boilers, a heat storage tank, and so on. The energy plants efficiently supply electric power, steam, heat, and compressed air (namely tertiary energies) required for production facilities using electric power and gas (namely secondary energies) purchased from electricity and gas utilities.

In factories, the energy plants convert secondary energies into tertiary energies and supplies them to the production facilities (see upper part of Fig. 1 [3]). In terms of determining energy purchase costs, hourly tertiary energy consumption, which is required to realize the determined production schedule, should be determined. Subsequently, the energy plants are operated for supplying the tertiary energies. Finally, the required hourly secondary energies are determined and the secondary energy purchase costs are determined (see the lower part of Fig. 1). In other words, the hourly tertiary energies supplied by energy plants are determined by changing a production schedule.

At the demand side of factories, optimization of production schedule is important to reduce labor costs, energy purchase costs, and CO_2 emissions. Studies on the production scheduling optimization considering energy consumption or CO_2 emissions have been conducted [4-6]. However, these studies utilized only ideal formulations such as job-shop and flow-shop scheduling problems. Therefore, these studies cannot be applied to actual factories. At the supply side of factories, studies on the optimal operational planning of energy plants have been conducted [7-9]. However, these studies have utilized fixed energy demands such as electric power, heat energy, and steam energy. In other

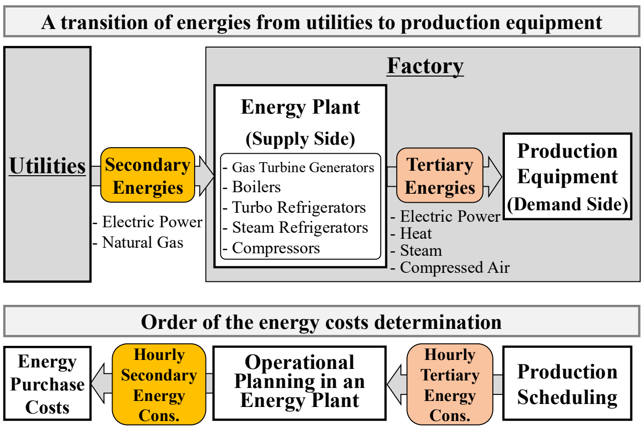


Fig. 1 A transition of energies from utilities to production equipment and the order of the energy costs determination in a factory [3].

words, these studies do not consider variations of energy demands by changing a production schedule.

Based on these backgrounds, the authors' previous study proposed the integrated supply and demand optimization framework for the production scheduling optimization problem (PSOP) and the optimal operational planning problem of energy plants (OOPPEPs), which simulate the actual supply-demand system [10]. The framework is hierarchical. PSOP is the primary problem and OOPPEPs is the sub-problem. The framework utilizes a production simulator [11], which can accurately reproduce PSOP. Therefore, it can be applied to actual factories.

In the primary problem of the framework, the authors' previous research results have shown that the decision variables are "the order of production", "the ratio of the number of produced products per line", and "the production start time" [12]. Since these decision variables can be treated as discrete values, the primary problem can be treated as a combinatorial optimization problem. There are two effective solution methods for combinatorial optimization problems: exact solution and approximate solution techniques. Although exact solution techniques require large amount of computation time, it obtains an exact optimal solution. On the other hand, approximate solution techniques obtain a solution close to the optimal solution and requires less computation time. The primary problem requires a high-quality solution to be obtained in a short time considering actual factory operation needs. Therefore, the paper applies meta-heuristics, a type of approximate solution techniques. Previous researches have confirmed that the Integer form of Population Based Incremental Learning (IPBIL), which is one of meta-heuristics, has been effective for various combinatorial optimization problems [13,14]. The authors propose Improved Adaptive Integer form of Population Based Incremental Learning (IAIPBIL) method and applied IAIPBIL for the PSOP where the order of production is utilized as decision variables [10]. IAIPBIL has an improved adaptive function and it varies the learning rate during the search procedures. The simulation results confirmed that the IAIPBIL based method outperforms the comparative methods and reduces standard deviations of the solutions [10].

As stated above, the optimal production schedule should be obtained in a short time. Actually, the factory production manager needs to generate the optimal production schedule several times during working hours. For example, the maximum time that can be spent on the optimal production scheduling per time is around four hours, namely, 14,400 seconds. In the PSOP targeted in this study, it is required to execute a production simulator in order to evaluate the objective function values. Since executing the production simulator takes around 10 seconds, the maximum number of objective function evaluation is limited to around 1,500. Therefore, in this study, the maximum number of iterations and the number of individuals are set extremely smaller than those generally set in various IPBIL based applications. The feature leads the conventional IAIPBIL based method to weak intensification of the search.

In the field of optimization, it is widely acknowledged that high-quality solutions often exist in the neighborhood of other high-quality solutions. The principle is called "Proximate Optimality Principle (POP)" [15]. Furthermore, to achieve efficient search, optimization methods generally require a balance between diversification and intensification. On the other hand, the intensification function of PBIL-based methods is not sufficient under conditions where the number of the objective function value evaluations is small. Therefore, for problems with a small number of these evaluations, IAIPBIL had room for improvement in the quality of the solutions.

Generally, evolutionary computation techniques can efficiently obtain high-quality solutions by promoting diversification in the first half of the search and promoting intensification in the second half. To solve the problem of the comparative IAIPBIL method, it is effective to promote the diversification function using IAIPBIL in the first half of the search and to promote the intensification in the second half of the search. In order to promote the intensification of the search, neighboring search from high-quality solutions is effective based on the theory of POPs. Using the optimization methods, users can reduce the man-hours required for parameter tuning with the fewest number of parameters to be set in advance. There are several metaheuristics for combinatorial optimization problems. Among them, Reactive Tabu Search (RTS) has fewer parameters than other methods [16]. Moreover, RTS is one of neighboring search methods. Therefore, RTS is one of the appropriate candidates for promoting intensification.

Considering the above issues, this paper proposes IAIPBIL-RTS for PSOPs. The proposed method solves the challenge of improving quality of solutions by IAIPBIL based method [10] because it enhances intensification in the second half of the search. In this paper, the proposed method is applied to a PSOP, and effectiveness of the proposed method is verified with a PSOP of the actual machining process in an assembly and fabrication factory and a OOPPEPs [17]. The decision variable on the demand side is the order of production of products. The effectiveness of the proposed IAIPBIL-RTS based method is confirmed by comparing with the comparative RTS based method and IAIPBIL based method.

II. The Integrated Enrgy Supply and Demand Optimization Framework

An overview of the integrated supply and demand optimization framework [10] is shown in Fig. 2. The algorithmic procedure of this framework is also shown below.

Step 1 In the primary problem, the optimal production scheduling sends the production parameters obtained by random numbers to the production simulator ((1) in Fig. 2).

Step 2 In the primary problem, the production simulator obtains a production schedule based on the production parameters and calculate the tertiary energy load per hour using the obtained production schedule. Then, the simulator sends it to the sub-problem ((2) in Fig. 2).

Step 3 In the sub-problem, the optimal operational planning of energy plants calculates the optimal secondary energy purchase costs of energy plants based on the tertiary energy load and sends it to the production simulator ((3) in Fig. 2).

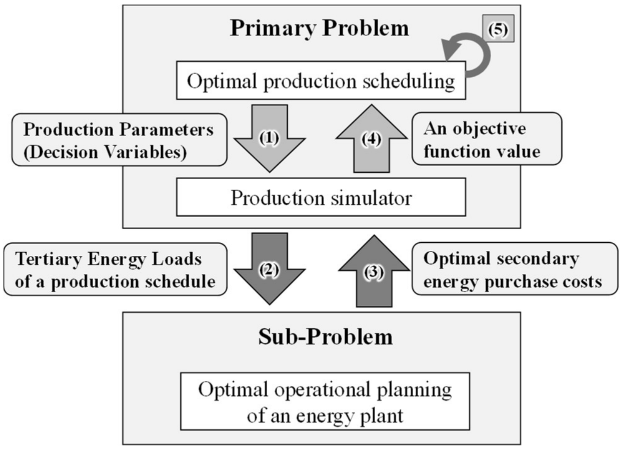


Fig. 2 An overview of the integrated demand and supply optimization framework [10].

Step 4 In the primary problem, the production simulator calculates the objective function value based on the optimal secondary energy purchase costs of energy plants and the production schedule. Then, the simulator sends it to the optimal production scheduling ((4) in Fig. 2).

Step 5 In the primary problem, the optimal production scheduling updates the production parameters based on the objective function values ((5) in Fig. 2).

Step 6 When the number of iterations reaches the maximum number of iterations, the procedure terminates. Otherwise, return to Step 2.

As shown above, the framework solves the primary problem and the sub-problem alternately and obtains a high-quality solution considering PSOP and OOPPEPs.

A. The Primary Problem

In the primary problem, the actual production system of an assembly and processing plant with multiple production facilities and workers is modeled using a production simulator. The primary problem aims to generate a production schedule that minimizes the production cost index. This index is calculated based on factors such as labor cost obtained from labor man-hours, electric power purchase costs obtained from electric power consumption, and CO2 emission costs obtained from the purchased and generated electric power.

(1) Decision variable

The decision variable is the order of production of the yth product at the xth production line.

$$P_{xy}\ (P_{xy} \in \mathbb{N} :\ x = 1, \ldots, NL, y = 1, \ldots, NP_x)$$

where P_{xy} is the order of production of the yth product at the xth production line, $\mathbb{N}$ is a natural number. NL is the number of production lines. NP_x is the number of produced products at the xth production line.

(2) Objective function

A production cost index is calculated in the objective function with three indices (KPIs): productivity, energy efficiency, and environmental loads using the following equation:

$$\min\ \alpha_1 \times P(P_{xy}) + \alpha_2 \times E(P_{xy}) + \alpha_3 \times C(P_{xy}) + P_{odr} + P_{epc} \quad (1)$$

$$P(P_{xy}) = \frac{\sum(MC(P_{xy}) + LC(P_{xy})) \times LT}{NP} \quad (2)$$

$$E(P_{xy}) = \frac{\sum\{EC(P_{xy}) + G\ \ (P_{xy})\} \times LT}{NP} \quad (3)$$

$$C(P_{xy}) = \frac{\sum(CC(P_{xy})) \times LT}{NP} \quad (4)$$

$$LT_{xy} = OT_{xy} - ST_{xy} \quad (5)$$

$$P_{odr} = (100 - DR)^2 \quad (6)$$

$$P_{epc} = \left(EC(P_{xy}) - FT\right) \times 100 \quad (EC(P_{xy}) > FT) \quad (7)$$

where $P(P_{xy})$ is the productivity KPI value calculated with P_{xy}, $E(P_{xy})$ is the energy efficiency KPI value calculated with P_{xy}, $C(P_{xy})$ is the environmental load KPI value calculated with P_{xy}, α_1, α_2, and α_3 are weighting coefficients ($\alpha_1 + \alpha_2 + \alpha_3 = 1$), P_{odr} is a penalty value obtained by squaring the percentage of violations of on-time delivery rate constraints in the production schedule, P_{epc} is a penalty value calculated from electric power purchase costs and their thresholds, $MC(P_{xy})$ are material costs calculated with P_{xy}, $LC(P_{xy})$ are labor costs calculated with P_{xy}, LT_{xy} is a load time interval of the yth product at the xth line, NP is the number of products, $EC(P_{xy})$ are electric power purchase costs calculated with P_{xy}, $GC(P_{xy})$ are gas purchase costs calculated with P_{xy}, $CC(P_{xy})$ are CO2 emissions costs calculated with P_{xy}, OT_{xy} is an operating time interval of the yth product at the xth line, ST_{xy} is a planned shutdown time interval of the yth product at the xth line, DR is on-time delivery rate in the production schedule, and FT is an electric power purchase cost threshold.

The weighting coefficients can be set by users considering each production situation.

(3) Constraints

The constraints of the practical factory production system are modeled inside the production simulator, which is a black box. Therefore, they cannot be expressed with mathematical formulations.

B. The Sub-problem

Target energy plants using in the sub-problem is a model based on actual energy plants [17]. The energy plants supply electric power to an electric power load through gas turbine generators (GTGs) using purchased electric power and natural gas. The energy plants also supply steam to a steam load by generating steam using boilers and waste heat boilers of the GTGs. Furthermore, these energy plants supply heat to a heat load by supplying chilled water from steam absorption refrigerators (SRs) and turbo refrigerators (TRs) through a heat storage tank (HST).

(1) Decision variables

The decision variables are the amount of hourly heat output of TR, the amount of hourly heat output of SR, and the amount of hourly gas consumed by GTG per day. The details of decision variables are shown below.

$Q_{xy}^{tr}\ (Q_{xy}^{tr} \in \mathbb{R}, x = 1, \cdots, X, y = 1, \cdots, E_t)$

where Q_{xy}^{tr} is an amount of TR heat output of the yth unit at time x, $\mathbb{R}$ is a real number, X is target hours ($= 24$), and E_t is the number of TRs.

$Q_{xy}^{st}\ (Q_{xy}^{st} \in \mathbb{R}, x = 1, \cdots, X, y = 1, \cdots, E_s)$

where Q_{xy}^{st} is an amount of SRs heat output of the yth unit at time x, E_s is the number of SRs.

$Q_{xy}^{gt}\ (Q_{xy}^{gt} \in \mathbb{R}, x = 1, \cdots, X, y = 1, \cdots, E_{gt})$

where Q_{xy}^{gt} is an amount of GTG gas output of the yth unit at time x, E_{gt} is the number of GTGs.

(2) Objective function

The objective function value is calculated by minimization of the secondary energy purchase costs calculated from the amount of electric power and natural gas purchased per day.

$$min \quad \sum_{x=1}^{X} \left\{ up_x^{Er} e_x + up_x^{Gr} \left(\sum_{y=1}^{E_{gt}} Q_{xy}^{gt} + \sum_{y=1}^{E_b} Q_{xy}^{b} \right) \right\} \quad (8)$$

where up_x^{Er} is an electricity unit price at time x, e_x is an amount of electricity purchased at time x, up_x^{Gr} is a natural gas unit price at time x, E_b is the number of boilers, Q_{xy}^{b} is an amount of boiler gas output of the yth unit at time x.

In the objective function, e_x and Q_{xy}^{b} are dependent variables. Namely, e_x can be calculated by an electric power load plus the amount of electric power input of TRs minus the amount of electric power output of GTGs. Q_{xy}^{b} can be calculated by a steam load plus the amount of steam input of SRs minus the amount of steam output of waste heat boilers of GTGs.

(3) Constraints

For simplicity, only the demand and supply balance and machine characteristic constraint items are listed below. For more details on each constraint, see Ref. [17].

a) *Demand and supply balances*:

- Electric power balance - Steam balance - Heat balance

b) *machine characteristics*:

- TR - SR - GT - Boiler - Thermal storage

III. The Proposed Improved Adaptive Integer form of Population Based Incremental Learning-Reactive Tabu Search and its Application to the Primary Problem

The proposed IAIPBIL-RTS method, which combines IAIPBIL [10] and RTS [16] methods, applies to IAIPBIL search during the initial phase and RTS search during the latter phase of the optimization process. By adopting the method, a well-balanced search strategy including both diversification and intensification is achieved. The PSOP in this paper utilizes the production order of products as decision variables. Decision variables are combinations of different integer values ranging from 1 to the number of products. In other words, each integer value is appeared only once in the decision variables. Therefore, this paper utilizes an unused list of integer values in decision variables at line z (L_{uuz}) and an used list of integer values in decision variables at line z (L_{uz}). Already selected integer values as decision variables are added in L_{uz} and removed from L_{uuz}, and a next decision variable is selected in L_{uuz}. The algorithm of applying the proposed IAIPBIL-RTS to the primary problem is shown below in three parts. Each part is performed in this order and the final solution is obtained.

(1) Initialization

The algorithm for initialization is shown below.

Step 1 Set the iteration number of IAIPBIL $iter_1$ to 1 and the iteration number of RTS $iter_2$ to 1.

Step 2 Set $z = 1$, $L_{uz} = \{\}$, $L_{uuz} = \{1, \dots, p_z\}$ where p_z is the number of products in line z.

Step 3 Initialize each element of the probability matrix with (9).

$$P_{xyz}^{iter_1} = \frac{1}{p_z}$$
$$(x = 1, \cdots, p_z, y = 1, \cdots, p_z, z = 1, \cdots, L) \quad (9)$$

where $P_{xyz}^{iter_1}$ is probability value of the yth possible order of production of product x at line z at $iter_1$, L is total number of lines.

Step 4 Set the individual number in to 1 and the order of production y to 1.

(2) IAIPBIL part

The algorithm of IAIPBIL part is shown below.

Step 1 Determine the product x for the order of production y at line z using the probability matrix $P_{xyz}^{iter_1} (x \in L_{uuz})$.

Step 2 Store the product x of line z determined in step 1 into L_{uz} and remove the product x of line z determined in step.1 from L_{uuz}.

Step 3 If the order of production y equals to p_z, go to Step 4. Otherwise, return to Step 1 and the order of production y is set to $y + 1$.

Step 4 If the individual number in equals to N_{ind}, go to Step 5. Otherwise, the individual number in is changed to $in + 1$ and return to Step 1.

Step 5 Update the learning rate using (10) and (11).

$$\varepsilon^{iter_1} = \begin{cases} \frac{\log(1/\varepsilon_{ini}) \times E_c}{(im_{iaipbil})^2} \times iter_1 + \varepsilon_{ini} & (iter_1 < i_t) \\ -\frac{\log(1/\varepsilon_{ini}) \times E_c}{2 \times (im_{iaipbil})^2} \times (iter_1 - i_t) & \\ + \frac{\log(1/\varepsilon_{ini}) \times E_c}{(im_{iaipbil})^2} \times i_t + \varepsilon_{in} & (iter_1 \geq i_t) \end{cases} \quad (10)$$

$$i_t = RU(im_{iaipbil} \times \beta) \quad (11)$$

where ε^{iter_1} is a learning rate when iteration is $iter_1$, ε_{ini} is an initial learning rate, $im_{iaipbil}$ is the maximum iteration number, E_c is the number of column elements in the probability matrix, i_t is a preset iteration when the learning rate begins to decrease, $RU()$ is the round-up function, and β is a constant value.

Step 6 Calculate an objective function value for each individual using the order of production obtained from Step 1 to Step 5, and the individual with the best value is selected as the best individual (BI).

Step 7 Update the probability matrix using the learning rate updated in Step 5 and the decision variable of the best individual selected in Step 6 with (12).

$$P_{xyz}^{iter_1+1} = P_{xyz}^{iter_1} + \varepsilon^{iter_1}$$
$$(x = 1, \cdots, p_z, y = BI, z = 1, \cdots, L) \quad (12)$$

Step 8 Mutate the probability matrix according to the mutation probability (MUT_P) using (13).

$$P_{xyz}^{iter_1+1} = (1.0 - S) \times P_{xyz}^{iter_1+1} + R(0\ or\ 1) \times S$$
$$(x = 1, \cdots, p_z, y = 1, \cdots, p_z, z = 1, \cdots, L) \quad (13)$$

Step 9 Normalize the probability matrix using (14).

$$P_{xyz}^{iter_1+1} = \frac{P_{xyz}^{iter_1+1}}{\sum_{y=1}^{p_z} P_{xyz}^{iter_1+1}}$$
$$(x = 1, \cdots, p_z,\ y = 1, \cdots, p_z, z = 1, \cdots, L) \quad (14)$$

Step 10 If the line number z equals to L, go to Step 11. Otherwise, line number z is changed to $z + 1$ and return to Step 1.

Step 11 If $iter_1$ equals to $im_{iiapbil}$, the best order of production in the search is saved as $IAIPBIL_{best}$ and go to the RTS part. Otherwise, $iter_1$ is changed to $iter_1 + 1$ and return to Step 1.

(3) RTS part

The algorithm of RTS part is shown below.

Step 1 Set an initial solution as $IAIPBIL_{best}$.

Step 2 Generate $N_{neighbor}$ candidates order of production of the current solution by generating neighboring order of production.

Step 3 Evaluate the production cost of each candidate order of production.

Step 4 Determine whether the order of production violates tabu conditions or not.

Step 5 Select the best order of production that does not violate tabu conditions.

Step 6 If the selected order of production is searched again within a fixed iteration, or the selected solution is already searched, expand the length of the tabu list.

Step 7 If the length of the tabu list is not adjusted for a longer iteration than a moving average of the iteration until the searched order of production are appeared again, shorten the length of the tabu list.

Step 8 When $iter_2$ equals to im_{rts}, the best order of production and the best schedule in the search is output and the search is terminated. Otherwise, set $iter_2$ to $iter_2 + 1$ and return to Step 2.

IV. Simulations

A. Simulation Conditions

The primary problem models some production processes in an actual assembly and processing factory. Specifically, the problem is to determine the processing order of product models (16 variables) for eight types of products in two lines. The sub-problem is the optimization benchmark problem of OOPPEPs [17]. Since the sub-problem can be formulated as a linear programming problem, the linear programming package is utilized. Using the comparative IAIPBIL method (the comparative method 1 [10]), RTS method (the comparative method 2), and the proposed IAIPBIL-RTS method (the proposed method), for the primary problem, the production costs are compared.

The common parameters, parameters of the proposed and the comparative methods are shown below. These parameters are determined by pre-simulations. The number of objective function evaluations for the comparative method and the proposed method is set to be the same for fair comparison.

1) Common parameters:
The number of trials: 30, $\alpha_1, \alpha_2, \alpha_3$: 0.333 each, p_z: 8, L: 2

2) A parameter of the proposed method and the comparative method 1:
elements of the initial probability matrix: 1/6 each, ε_{ini}: 0.1, MUT_P: 0.02, S: 0.02, β: 0.8, $N_{neighbor}$: 50

3) A parameter of the comparative method 1:
$im_{iaipbil}$: 30

4) A parameter of the comparative method 2:
im_{rts}: 30, N_{ind}: 50

5) Parameters of the proposed method:
$im_{iaipbil}$: 10, 15, 20, 24, im_{rts}: 20, 15, 10, 6

Simulation softwares are developed using C language (Microsoft Visual studio 2019 Visual C++), C# (Microsoft Visual studio 2019 Visual C#), Microsoft SQL Server 2016, and GLPK (VS2012 ARM Cross Tools Command Prompt) on an Intel core i9-10980XE (3.00GHz) PC.

B. Simulation Results

Table 1 shows comparison of average and standard deviation values of objective function and the number of minimum solution obtained through 30 trials by the proposed method and the comparative methods, and a result of a statistical test with a p-value using the Friedman test. Normality of the simulation results is not confirmed by D'Agostino and Pearson, and Anderson and Darling tests. Therefore, the Friedman test is applied. The average value of the objective function value by the proposed method ($im_{iaipbil} = 15, 20$) is lower than the value by the comparative methods. The standard deviation value of the objective function value by the proposed method ($im_{iaipbil} = 15$) is lower than those values by the comparative methods. In the number of the minimum solution obtained, the number by the comparative method 1 is two, and the number by the

TABLE I. AVERAGE AND STANDARD DEVIATION VALUES OF OBJECTIVE FUNCTION, THE NUMBER OF MINIMUM SOLUTION OBTAINED BY THE PROPOSED AND THE COMPARATIVE METHODS, AND A RESULT OF A STATISTICAL TEST WITH A P-VALUE USING THE FRIEDMAN TEST.

Method	$im_{iaipbil}$	*Ave.*	*Std.*	*Num. of Min. obtained*	*p-value*
The proposed method	10	2536.014	30.344	3	4.91E-05
	15	2521.930	**27.915**	4	
	20	**2519.220**	30.202	**6**	
The comparative method 1 [10]	30	2522.179	30.194	2	
The comparative method 2	0	2575.742	57.690	0	

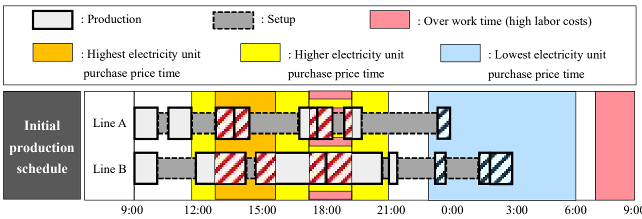
comparative method 2 is zero. Namely, the comparative method 2 cannot obtained the minimum solution. On the other hand, the proposed method ($im_{iaipbil} = 20$) can obtained it 6 times. It indicates that the neighborhood search by RTS in the proposed method can obtain more minimum solutions than the comparative method 1. There is a significant difference among the three methods with 0.05 significant level because p-value is 4.91E-05 as shown in the table. As a post hoc test, the Wilcoxon's signed rank tests between two methods are performed with correction for p-values by Holm method. From results of the test, the proposed method is confirmed to be more significant than the comparative methods.

Figure 3 shows comparison of an initial production schedule and the optimal production schedule by the proposed method. In the initial production schedule, many products are produced at the over work time intervals and higher electricity unit purchase price time intervals. Therefore, labor costs and electric power purchase costs become high. On the other hand, in the best production schedule by the proposed method, production time at over time work intervals and higher electricity unit purchase price time intervals is shorter than the time of the initial production schedule. Therefore, labor costs and electric power purchase costs can be reduced effectively.

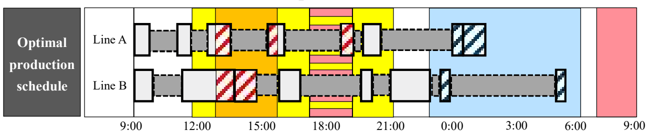
V. CONCLUSIONS

This paper proposes IAIPBIL-RTS for a PSOP. The proposed method solves the challenge of improving quality of solutions by IAIPBIL based method because it can enhance intensification in the second half of the search. Effectiveness of the proposed method is verified with a PSOP of the actual machining process in an assembly and fabrication factory and OOPPEPs. It is confirmed that the proposed method can generate high-quality solutions and can reduce the production costs.

As future works, more effective metaheuristic and evolutionary computation methods are investigated for PSOPs of various discrete manufacturing industries.



(a) An initial production schedule.



(b) The optimal production schedule.

Fig. 3. Comparison of an initial production schedule and the optimal production schedule by the proposed method.

REFERENCES

[1] Carbon Neutrality by United Nations, https://sdgs.un.org/partnerships/carbon-neutrality.

[2] Agency for Natural Resources and Energy, Ministry of Economy, Trade and Industry, "Annual Report on Energy 2021 (Energy White Paper 2022)," Part 2, Chapter 1, June 2022 (in Japanese). https://www.enecho.meti.go.jp/about/whitepaper/2022/pdf/

[3] S. Kawaguchi, et al., "Improved Parallel Reactive Hybrid Particle Swarm Optimization using an Improved Neighborhood Schedule Generation Method for the Integrated Framework of Optimal Production Scheduling and Optimal Operational Planning of an Energy Plant in a Factory," IEEJ Transaction on Industrial Applications, Vol. 139, No. 12, pp.1005-1014, 2019 (in Japanese).

[4] R. Zhang, et al., "Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption," Journal of Cleaner Production, Vol. 112, Part 4, pp. 3361-3375, 2016.

[5] L. Yin, et al., "A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem," Journal of Sustaineble Computing: Informatics and Systems, Vol. 13, pp.15-30, 2017.

[6] H. Zhang, et al., "A novel heuristic method for the energy-efficient flexible job-shop scheduling problem with sequence-dependent set-up and transportation time," Journal of Engineering Optimization, 2022.

[7] K. F. Fong, et al., "HVAC system optimization for energy management by evolutionary programming," Journal of Energy and Building, Vol.38, Issue 3, pp.220-231, 2006.

[8] S. Wang, et al., "Supervisory and Optimal Control of Building HVAC Systems: A Review," HVAC&R Research, Vol. 14, pp. 3-32, 2011.

[9] K. Arai, et al., "Optimal Operational Planning of Energy Plants Considering Uncertainty of Renewable Energy Outputs by Global-best Brain Storm Optimization," International Federation of Automatic Control, Vol. 52, Issue 4, pp.153-158, 2019.

[10] S. Kawaguchi, et al., "Integrated Supply and Demand Optimization Framework in Factories using an Improved Integer form of Adaptive Population-Based Incremental Learning and a Production Simulator," IEEJ Transaction on Industrial Applications, Vol. 143, No. 11, 2023 (in Japanese, to be appeared).

[11] K. Takahashi, et al., "Optimal Production Scheduling using a Production Simulator and Multi-population Global-best Modified Brain Storm Optimization," Proc. of IEEE WCCI, 2022.

[12] R. Katagiri, et al., "Optimal Production Scheduling by Integer Form of Population-Based Incremental Learning with Initial Probability Matrix Setting Methods and a Practical Production Simulator," Proc. of IEEE SSCI, 2023 (to be appeared).

[13] Z. He, et al., "A New Population-based Incremental Learning Method for the Traveling Salesman Problem," Proc. of IEEE CEC1999, 1999.

[14] Y. Kobayashi, et al., "Transposition Point Optimization for Voltage Unbalance Mitigation in Electric Power Distribution Systems by Integer Form of Population-Based Incremental Learning," Proc. of IEEE CEC2023, 2023.

[15] T.Kanazawa, et al., "Proximate optimality principle based Tabu Search," Proc. of IEEE SMC, 2003.

[16] R. Battiti, et al., "The Reactive Tabu Search," ORSA Journal on Computing, Vol. 6, No. 2, pp.1-26, 1994.

[17] Committee of new exploration and industrial applications of computational intelligent system, "Optimization benchmark problems for industrial applications," Technical report of IEE of Japan, No. 1287, pp. 9-12, 2013 (in Japanese).