

Study on Job Shop Scheduling for Keeping the Requested Shipping Sequence by Production System Modeling and Backward Simulation

Okubo, Yui
Yokohama National University

Mitsuyuki, Taiga
Yokohama National University

<https://hdl.handle.net/2324/7399680>

出版情報 : Advances in Transdisciplinary Engineering. 16, pp.203-212, 2021. IOS Press
バージョン :
権利関係 : © 2021 The authors and IOS Press.



Study on Job Shop Scheduling for Keeping the Requested Shipping Sequence by Production System Modeling and Backward Simulation

Yui OKUBO¹ and Taiga MITSUYUKI

Yokohama National University, Japan

Abstract. In the field of high-mix low-volume production, to keep shipping deadline of each order is one of key factors due to the reason for make-to-order production. However, this type of production has a enormous number of parts to be handled, complicated work procedures, and many restrictions on production resources. Therefore, it is difficult to make an optimal production plan considering shipping deadline and efficient usage of production resources. In this paper, the objective is to create work plans which meet the due date with short lead times as much as possible in a job shop type production line. First, we propose the method how to model and represent the complex data of an entire factory structure. Then, based on this model, it was proved that we can obtain the production plan with short lead times that emphasizes compliance with the sequence of product shipping by backward discrete event simulation. Furthermore, the effectiveness of the proposed method and the validity of the obtained production plan are confirmed by using actual factory processes and real data.

Keywords. Job Shop Scheduling, Backward Simulation, Shipping Sequence, Production Modeling

Background

In high-mix low-volume production, to keep shipping deadline of each order is one of the major issues in production management. For solving this issue, it is important to create a realistic and optimal production schedule considering shipping deadline of each order and efficient usage of production resources. Such job-shop type factories have huge complex project architecture due to the large number and variety of parts and processes. For example, each product has a unique procedure and workload for manufacturing. In addition, production resources including workers and robots have many restrictions and differences in their capabilities. For these reasons, it is difficult to create a realistic and optimal production schedule considering the requested shipping sequence and shipping deadline of each order and efficient usage of production resources.

There have been many studies on scheduling problems focusing on shipping deadline. Morinaga et al. proposed a production scheduling method using a genetic algorithm for

¹ Corresponding Author, Mail: okubo-yui-pb@ynu.jp.

multi-objective optimization of weighted shipping delays and equalization of the workload of set-up personnel [1]. Camino R. Vela et al. combined a tabu search procedure using nearest neighbor search with a genetic algorithm to maximize due date satisfaction under uncertain task durations and flexible shipping deadline [2]. J.Lohmer et al. presented a model formulation of the distributed job-shop scheduling problem with due date consideration and proposed adapted greedy heuristics as well as a genetic algorithm to solve large problem instances [3]. These studies propose an optimization algorithm of job-shop scheduling problems with shipping deadline added to the objective function.

In a situation where the shipping deadline has been determined, establishing the production plan in reverse is often adopted. This method is commonly referred to as backward simulations. Lynch and Vaandrager conducted a study on schedule planning using forward and backward simulation [4]. They established the production plan through backward simulations on plans involving several processes, and equipment allocation and verification were performed through forward simulation. Musselman et al. defined the schedule planning system and information flows through forward and backward simulations of APS (the Advanced Planning and Scheduling System) in an overall enterprise information system [5]. In addition, studies have been conducted on applying backward simulation-based production planning to semiconductor processes, which have batch production and Make-to-Stock (MTS) characteristics [6] [7]. Zhu et al used backward simulation to establish production plans for steelworks, which include several processes in their production flow [8]. Most existing studies performed backward simulations that focused on industries with fixed processes and facilities. In production simulations with fixed facilities, a single simulation model that focuses on the facilities is used. However, in high-mix low-volume production, all designs and characteristics of the products to be produced are different, and they undergo different processes and require different facilities. It is difficult for establishing the production plan in high-mix low-volume production to apply these studies.

On the other hand, the Resource-Constrained Project Scheduling Problem (RCPSP) [9], one of the most challenging combinatorial optimization scheduling problems, has been the focus of a great deal of research. This involves the scheduling of project activities subject to precedence and resource constraints to meet the objectives in the best possible way. Nouri N. et al. proposed an efficient nature-inspired ABC algorithm for solving the RCPSP by using three types of honey bees [10]. Ripon K. et al. considered RCPSPs with known deterministic renewable resource requirements but uncertain activity durations, to approach based on the robust optimization concept [11]. However, these studies deal with very simple problems compared to the real problems, because the optimizing calculation cost is too heavy for utilizing these methods in the real field.

With the optimization approach, only the production planners can understand it, but not the production managers and factory workers, and they are not able to respond well to sudden situations. From the perspective of production managers, it is necessary to have a scheduling method that makes it easy to make plans again when some troubles occur in the field. Optimization methods that take a lot of computational time may sometimes be inappropriate because troubles frequently occur in the production field. And, solutions obtained by optimization methods are not always satisfactory for the field. From factory workers' point of view, they would prefer to adopt easy-to-understand working rules rather than having to check a work plan every time. The assignment solution obtained by the optimization method does not have clear rules such as dispatching rules, which are often unacceptable to workers. Therefore, it is necessary to

have a transdisciplinary scheme that allows information sharing among production planners, production managers, and factory workers, who have their own individual disciplines for operational reasons. In this study, we focus on a factory where products are produced in all different processes and facilities. Then we approach the scheduling planning to ship the products in the required sequence. We propose the factory modeling method using nodes and links, and backward simulation that can be achieved by simply reversing the links to satisfy the requested sequence.

1. Proposed Method

1.1. *System modeling of target production project*

All data reproducing a factory consists of four models: product model, workflow model, workplace model, and team model. These models will be the input information for the discrete event simulation described below.

The product model represents information related to products to be manufactured in the target production project. Each product has its dependent relationship with a workflow described below.

The workflow model represents a workflow for expressing a unique work procedure required to produce a target product. First, since this workflow is composed of multiple tasks. All tasks can be created from the information of the production model and target domain knowledge. Each task has the attribute of work amount and a dependency on others.

The workplace model represents information about a workplace and facilities in the workplace. A workplace is a variable used to express the limit of the area where products and facilities can be placed. First, define each workplace, the number of products that can be placed in this workplace, tasks that can be handled in this workplace, and facilities that this workplace has. The number of workers in a workplace can be constrained by dependency between the workplace and facilities. Next, define each facility and skill value for each task handled by it. If the skill value is set to 0, it means that the facility is not able to perform the corresponding task.

The team model represents information about an organizational team and workers for a target production project. First, define each team, tasks that the team corresponds to, and workers who belong to the team. Next, define each worker, skill value for each task, and facility that can be handled.

1.2. *Simulation system*

Next, we propose the discrete event simulation system to create a work plan using these models in the previous section. The proposed simulation system is developed based on the previous study[12]. The method for performing a task in this simulation is that the remaining work amount of each task is reduced for each step based on skill values of assigned facilities and workers, and status is updated. (NONE: cannot start, READY: can start, WORKING: work in progress, FINISHED: work completed). The specific procedure of simulation is following:

1. Initialize the simulation setting.
2. Extract of tasks which states are READY or WORKING.

3. Place products that are dependent on the extracted tasks in corresponding workplaces.
4. Extract available workers and facilities.
5. Allocate workers and facilities based on the task priority rule.
6. Perform the task
7. Determine the end of the simulation

As an initial setting to start the simulation, set the state of head tasks to READY and other tasks to NONE. In step 2, extract tasks in the READY or WORKING state. In step 3, the products depending on the READY tasks are placed in the workplace according to the workplace model. If the corresponding workplace is not available, the target product will stay at the same workplace as at time $t-1$. The products depending on WORKING tasks are placed in the same workplace as at time $t-1$. In step 4, available workers and facilities that are not allocated or working at the current time are extracted. In step 5, according to the task priority rule, workers and facilities with skill values for the corresponding task are allocated. The task priority rule is adopted to a dispatching rule called TSLACK (Total Slack Time) [13]. This rule performs the task included in a workflow that is on the critical path preferentially. Specifically, the smaller SLACK of the task calculated by the Critical Path Method [14], the higher its priority. Slack(t) of task j at time t is calculated as in equation (1) using the latest start time $LST_j(t)$ and the earliest start time $EST_j(t)$.

$$Slack(t) = LST_j(t) - EST_j(t) \quad (1)$$

If a resource is allocated to the task, the state of this task becomes WORKING and updates the remaining work amount of the task based on the skill value of the resource. If the remaining work amount is 0 in this update, the task state is set to FINISHED. If all preceding tasks are FINISHED, the state of a task becomes READY. When the states of all tasks are FINISHED, the simulation is terminated. If not, advance time t by one step and go back to step 2. The simulation result outputs the end time of the simulation, the state of each model at each time, and the tasks that each resource has assigned.

As an exception to the above simulation flow, auto task can be defined. Auto task allows automatic task performance without the need to assign facilities or workers. For example, it can be used to adjust the time interval between the input time of one product to the production line and the input time of the next product.

1.3. Backward simulation

Traditionally, forward scheduling methods have been used mainly for solving production scheduling problems by using simulation techniques. These methods allocate jobs in advance according to the process steps along with the time flow. However, it is not suitable for creating a schedule that aims to improve shipping sequence compliance. Because forward scheduling methods, which use a discrete event simulation following a time axis, cannot control the end time of work for each product. Therefore, this paper adopts the backward simulation as a scheduling method. In this method, the due date of each targeted product is starting time of the process and simulate in opposite direction

from the real-time flow. To perform backward simulation using the simulation system described in section 1.2, it is necessary to modify the workflow model. Figure 1 describes how to model the backward simulation method by comparing it with the forward simulation method. This figure considers a work plan in which N products are shipped in the requested sequence.

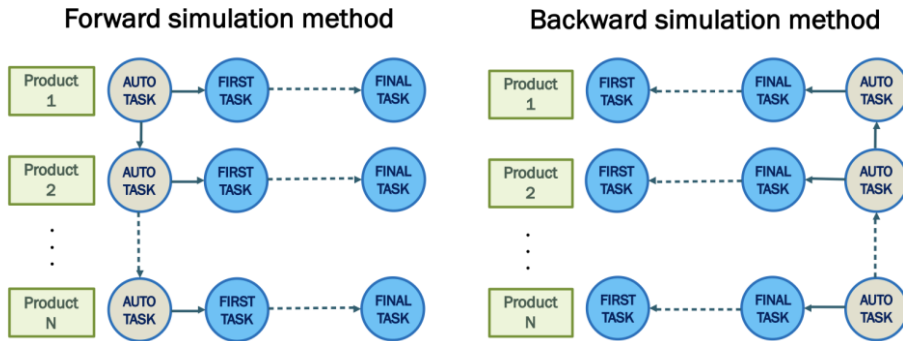


Figure 1. Comparison of the two methods.

First, as for the forward simulation method, simulation start time is when Product 1 (the product with the highest requirement sequence) is put into the production line. Next, define auto task and set this work amount taking into account the relationship between input time intervals for each product. Then, add auto task before the first task, and create a workflow by linking tasks with each other in the direction of the actual procedure. On the other hand, as for the backward simulation method, simulation start time is when Product N (the product with the lowest requirement sequence) is shipped. Next, define auto task to adjust shipment timing between products. Then, add auto task after the final task, and create a workflow in the opposite direction of the forward simulation method.

2. Case Study

In order to verify the validity of the proposed method, a case study is conducted using the structure and product data of a real construction machinery production project. This case study assumed that this factory should manufacture 20 products with a different deadline considering the requested shipping sequence on a certain day. Figure 2 shows the overview of the target production project.

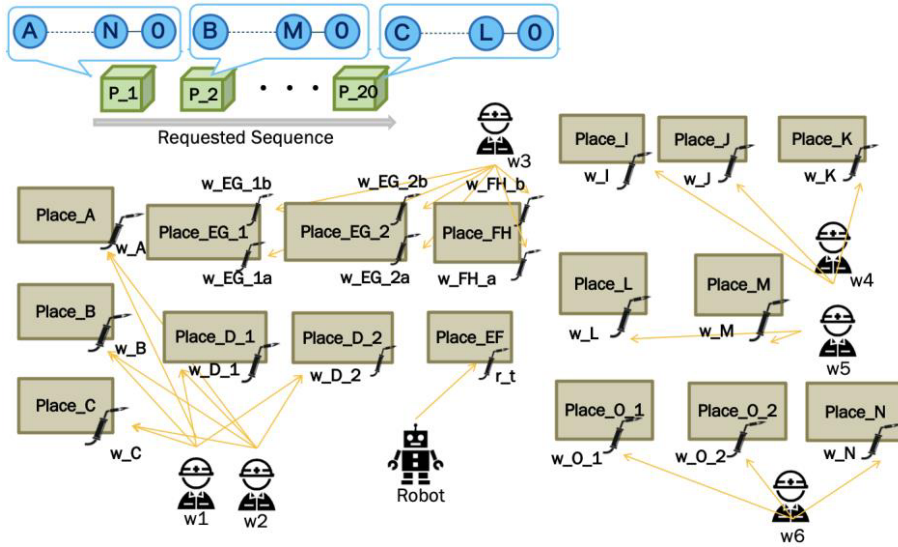


Figure 2. Overview of target production project.

First, define each product according to the product model by giving it a unique name for example “P_1”. The requested shipping sequence is determined according to the number “1” through “20” at the end of the product name in this case study. Since each of the 20 products has a different completion process, they have their own workflows as shown in Figure 2. Next, define 15 different tasks from “A” to “O” in this factory. For each product, create a workflow that prioritizes the tasks required to ship it. The work amount for each task is determined independently. In order to keep only the requested shipping sequence, the work amount of auto task for each product is set 0.

Each product is placed on a jig tool and workers use a welding machine on it to manufacture. In this factory, there is a workplace dedicated to robots where humans cannot enter. A jig tool is modeled as a workplace and a welding machine as a facility. Based on the floor map of the factory, define all workplaces and facilities and their dependencies. In this case study, a robot is modeled as a worker like a human, not as a facility and the robot's workplace has a facility dedicated to the robot. For example, the workplace “Place_FH” has two facilities, “w_FH_a” and “w_FH_b”. We can constrain the number of workers who can work in the same workplace by the number of welding machines.

Next, the task name following a workplace name “Place_” shows the dependency between them on which task can be performed in each workplace. For example, the workplace “Place_FH” can deal with the task “F” and “H”. The dependency between a facility and a task is the same as the dependency between the workplace to which the facility belongs and the task. Skill values of each facility are set to 1 for the task related by the arrow, and 0 for the others. For example, the facility “w_FH_a” has a skill value of 1 for the task “F” and “H”, and a skill value of 0 for others, which means “w_FH_a” can deal with “F” and “H”.

This factory is operated by six human workers and one robot worker. As the feature of this factory, for the task “E” and “F”, it is possible to choose to be performed either by human workers or robot worker. 7 workers named “w1” to “w6” and “Robo” are defined in Table 1. The dependency between the team and tasks is defined as this team

is targeted to work from the task “A” to “O”. Since the worker “Robo” can work at 2.5 times the human work efficiency, the skill value is set to 2.5.

Table 1. Worker Skill info.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Robo	0	0	0	0	2.5	2.5	0	0	0	0	0	0	0	0	0
w1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
w2	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
w3	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
w4	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
w5	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
w6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Finally, arrows connecting a worker and a facility in Figure 2 show the dependency between them regarding which facility each worker can handle. Since both the worker “w3” and “Robo” have skill of the task “E” and “F”, they may be assigned to the same workplace if only modeling in Table1. However, human workers can work together in the same workplace, but the robot cannot work with other workers. This constraint can be modeled by specifying the facility that can be handled by “Robo” and “w3”.

The above resource constraints and process characteristics of the factory are modeled and simulated by using a python discrete event simulation package called pDESy [15] that we have developed. Figure 3 shows a Gantt chart for expressing the Gantt chart of each product by executing the backward simulation. The orange periods indicate that work in progress and the gray periods express the time when the product is stagnant in the workplace. From Figure 3, it can be said that the 20 products are completed in the required sequence. If each product is manufactured based on this result, it can be shipped in the required shipping sequence.

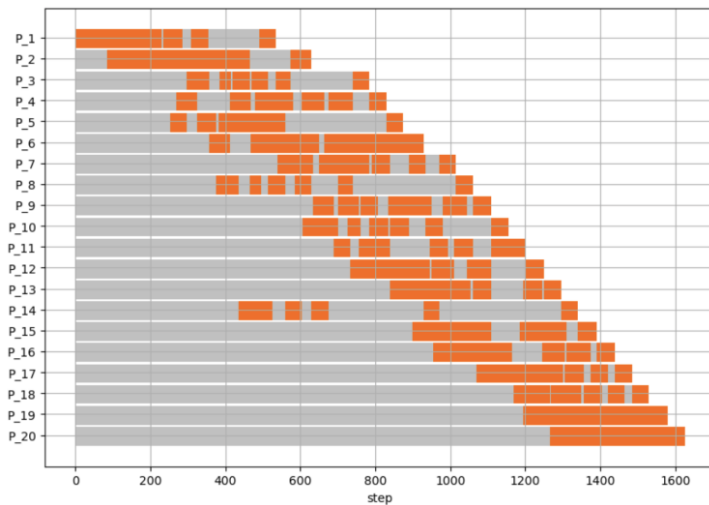


Figure 3. Gantt Chart for Case Study.

3. Discussion

This section compares the commonly used forward scheduling methods with the proposed backward simulation method for the same case to validate the production plan obtained with our proposed method. The forward simulation results performed by using pDESy are shown in the Gantt chart of Figure 4.

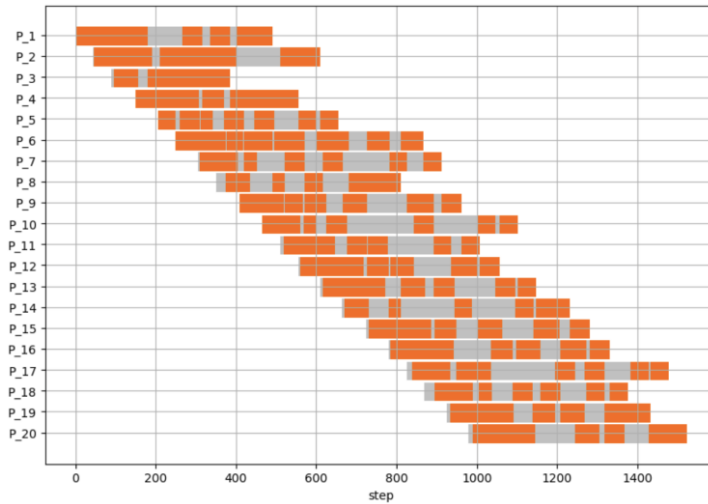


Figure 4. Gantt Chart for forward scheduling.

For comparing the results of forward and backward simulation, the following four KPIs are used.

- Lead time (LT)
- Total waiting time of workers ($Total\ FT$)
- Total stagnating amount of products ($Total\ ST$)
- Compliance value (CV)

In order to calculate the KPI correctly, it is necessary to evaluate the time period when products exist on all tasks. The lead time (LT) is defined as the period from the time when the first product in the completion sequence is shipped to the time when the 10th product in the completion sequence is shipped in this paper. Next, $Total\ FT$ is the total amount of time that each worker in a team waits between completing a task and starting the next task. Then, $Total\ ST$ is the total amount of time that each product stagnates in the workplace between the end of the task in progress and the start of the next task. Finally, the compliance value (CV) indicates how well the 10 products shipped comply with the requested shipping sequence. CV for a product (p_x) is calculated as in equation (2) using the requested shipping ranking of p_x ($Requested\ S_{p_x}$) and completion ranking of p_x by simulation output results ($Shipping\ S_{p_x}$). In other words, a smaller value indicates that products are completed in the requested sequence.

$$CV = \sum_{x=1,2,\dots,10} |Requested S_{p,x} - Shipping S_{p,x}| \quad (2)$$

Table 2 shows the results of comparing the same case study between the backward and forward types using KPIs defined above.

Table 2. Backward and Forward Results.

	LT [min]	Total FT [min]	Total ST [min]	CV
Backward	621	805	2230	0
Forward	622	554	2129	11

From the comparison, the production plan obtained from the backward method has a very good compliance value, even though the lead time is almost the same as obtained from the forward simulation method. However, the waiting time of workers and the stagnating amount of products are higher. In the forward simulation, tasks flowing to the production line are performed one after another by free workers without considering the future which results in an incorrect shipping sequence. When the backward simulation results are viewed in the direction of the time axis, it appears that the workers and products are made to have free time. Therefore, it is thought that products can be completed in the required sequence by providing a buffer and making adjustments, such as intentionally stagnating tasks in the backward simulation or forcibly giving workers free time.

In order to show that our method can be applied to other cases, we also investigated other 5 days' cases. The results are summarized in Table 3. From Case 2 to Case 5, the results were generally the same as the one in the case study. In Case 6, the compliance value of forward simulation result was 0 because the same type of products were produced continuously on that day. From these results, it can be said that the proposed method can be applied to other cases in this factory .

Table 3. Results of other 5 days' cases.

	LT		Total FT		Total ST		CV	
	FS	BS	FS	BS	FS	BS	FS	BS
Case 2	557	596	402	749	2158	2673	8	0
Case 3	521	556	496	798	1552	1984	5	0
Case 4	458	501	449	702	1387	2043	8	0
Case 5	553	598	734	849	1626	2239	8	0
Case 6	505	516	334	578	1111	1611	0	0

4. Conclusion

This paper proposed a method of job-shop scheduling considering keeping the requested shipping sequence of each product by system modeling of target production project and executing backward process simulation. Firstly, we modeled complex production project

by using the product model, workflow model, workplace model, and team model. From these models, proposed method can create a production plan for keeping requested shipping sequence by backward simulation. The proposed method was applied to the case study of the actual factory process of manufacturing multiple types of products. Results show that the backward simulation result is superior to the forward simulation result considering the comparison of lead time, total waiting time of workers, total stagnating amount of products and compliance value. From this results, it can be said that proposed method is effective as a production planning method for keeping the requested shipping sequence in a job-shop type production line.

Acknowledgement

We would like to express our gratitude to Mr. Rintaro Matsui and Mr. Hiroaki Nagai of Sumitomo Heavy Industries, Ltd. for their cooperation in providing us with field data and videos of operation at their construction machinery factory. And also, this work was supported by JSPS KAKENHI Grant Number 20K14962.

References

- [1] Morinaga, M. Nagao and M. Sano, Optimization of flexible job-shop scheduling with weighted tardiness and setup-worker load balance in make-to-order manufacturing, *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, Kitakyushu, Japan, 2014, pp. 87-94.
- [2] Camino R. Vela, Sezin Afsar, Juan José Palacios: Evolutionary tabu search for flexible due-date satisfaction in fuzzy job shop scheduling, *Computers and Operations Research*, 2020, Vol.119, pp.1-19.
- [3] J. Lohmer, D. Spengler and R. Lasch, Multi-factory Job Shop Scheduling with Due Date Objective, *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, Singapore, 2020, pp. 79-84.
- [4] Lynch, N., Vaandrager, F., Forward and backward simulations. *Inf. Comput.* 1995, 121 (2), pp. 214-233.
- [5] Musselman, K., O'Reilly, J., Duket, S., Supply chain planning: the role of simulation in advanced planning and scheduling. *Proc. 2002 Winter Simul. Conf.*, 2002, pp.1825-1830.
- [6] Park, B.C., Park, E.S., Choi, B.K., Kim, B.H., Lee, J.H., Simulation based planning and scheduling system for TFT-LCD fab. *Proc. 2002 Winter Simul. Conf.*, 2008, pp.2271-2276.
- [7] Seo, J.C., Chung, Y.H., Kim, B.H., Park, S.C., Backward capacity-filtering for electronics Fabs. *Prod. Plann. Contr.*, 2016, 27 (11), pp. 925-933.
- [8] Zhu, D.F., Zheng, Z., Gao, X.Q.. Intelligent optimization-based production planning and simulation analysis for steelmaking and continuous casting process. *Int. J. Iron Steel Res.*, 2010, 17 (9), pp.19-30.
- [9] W. Herroelen, E. Demeulemeester, B. De Reyck, Resource-constrained project scheduling: A survey of recent developments, *Computers & Operations Research*, 1998, Vol.25, Issue.4, pp. 279-302.
- [10] N. Nouri, S. Krichen, T. Ladhari, and M. Labidi, Solving the Resource-Constrained Project Scheduling Problem Using ABC Algorithm, *The 5th International conference on modeling, simulation and applied optimization*, 2013.
- [11] R.K. Chakraborty, A.Sarker, D.L. Essam: Resource constrained project scheduling with uncertain activity durations, *Computers & Industrial Engineering*, 2017, Vol. 112, pp 537-550.
- [12] T. Mitsuyuki, K. Hiekata, G. Takuya, Evaluation of Project Architecture in Software Development Mixing Waterfall and Agile by Using Process Simulation, *Journal of Industrial Integration and Management*, 2017, Vol. 2, No. 2, 1750007.
- [13] T. Mitsuyuki, K. Hiekata, H. Yamato, Design of production strategy considering the cutting peak demand of electricity in the shipbuilding industry, *Journal of Marine Science and Technology*, 2014, Vol.19, No.4, pp.425-437.
- [14] A.G. Mercier, *The Critical Path Method : its fundamentals*, Institutional Archive of Naval Postgraduate School, 1965.
- [15] pDESy, <https://github.com/pDESy/pDESy>