Personnel Scheduling for Logistics Warehouses Based on the Theory of Constraints

Naoya Nishio, Masaharu Kondo, and Ryota Kamoshida
Center for Technology Innovation-Artificial Intelligence, Research & Development Group, Hitachi, Tokyo, Japan
Email: naoya.nishio.jv@hitachi.com, {masaharu.kondo.bw, ryota.kamoshida.vw}@ hitachi.com

***Abstract*—We propose an efficient personnel scheduling method for logistics warehouse operation. In conventional methods, applying genetic algorithms or integer programming causes combinatorial explosions when conducting multiple processes concurrently and reallocating laborers to processes during processing. To address this problem, we introduce the idea of "effective productivity" to the theory of constraints (TOC) and propose four types of scheduling procedures to decide which process is given priority by considering productivity and operation completion time. We evaluate the proposed method by comparing with a method that imitate line managers. The results show that the operation completion time of schedules created by the proposed method are shorter than the imitated manual method. Moreover, our method's scheduling is proved to be faster than the imitated manual method, therefore, the proposed method is considered to be more suitable for use in real environments.**

***Index Terms*—personnel scheduling, logistics warehouse, TOC**

I. Introduction

Operational efficiency is crucial in various industries. This is especially urgent in the logistics industry, since the workloads of warehouse operations have been increasing with the growth of the e-commerce market. Additionally, the decline in the working population is a serious issue in Japan, and the lack of manpower has been rapidly worsening. Therefore, the best use of a reduced labor force is needed to maintain operational efficiency.

Warehouse operations consist of many processes, for example picking, inspection, and packing. For personnel scheduling and assigning the necessary amount of personnel to each process, there are various personnel scheduling problems, such as shift scheduling problems and resource constrained project scheduling problems (RCPSPs) [1]. A shift scheduling problem involves each worker being assigned to a single process at a fixed time interval such as an hour, a half day, or a day. Nurse scheduling is one example of scheduling problems where staff must be assigned to a day shift or a night shift, taking into consideration the required number of nurses, each nurses requirements and suitable skill levels. Whereas the purpose of RCPSP is to minimize operation completion time or tardiness. A fixed amount of resources is allocated to multiple processes that have anteroposterior relation, like production processes.

For solving these problems, there are two major approaches: a metaheuristic and integer programming. A metaheuristic is a heuristic approach to finding an approximate optimal solution for combinatorial optimization problems. A representative example is a genetic algorithm. While the metaheuristic is independent of specific computational problems, it requires a lot of computational time to get a solution. As for integer programming, it is impossible to solve large-scale optimization problems in a practical amount of time, and it is also difficult to solve nonlinear problems. On the other hand, at real production sites, real-time properties are required for rescheduling when a gap between a schedule prepared in advance and a performance occurs, and so a metaheuristic and integer programming are virtually unavailable.

Currently, personnel scheduling is conducted manually by site managers. They manage to monitor changes to all processes and allocate laborers to appropriate processes that can be started concurrently. It is burdensome for the managers because multiple processes operate at the same time. Moreover, there are three considerations for an optimal personnel scheduling in real time:

1. It must be possible to start a posterior process before the anterior process is complete.
2. The completion time of a process depends on the number of assigned laborers.
3. It must be possible to reassign labor to different processes.

Manual personnel scheduling is a reactive approach to actual situations, so that it is really hard for managers to achieve optimal assignment.

This study proposes a new scheduling method capable of creating a schedule in real time to deal with the ever-changing situation in logistics operation. We apply the concepts of a well-known production line management method called the "theory of constraints" (TOC) [2] to personnel scheduling and introduce an idea of "effective productivity" i.e., rate-limiting productivity that considers anterior processes. The principle of the TOC is that the process with the lowest productivity (the bottleneck process) dominates the productivity of the whole operation. The proposed method creates a schedule that maximizes the productivity of the whole operation by doing personnel scheduling to increase the productivity of

Manuscript received January 11, 2022; revised August 28, 2022.

doi: 10.18178/joams.10.3.69-76

the bottleneck process. This method could also quickly solve complex problems because this is a rule-based scheduling method based on the TOC.

The rest of the paper is organized as follows. We explain problem settings of logistics warehouse operations, related works, and issues of these applications in the next section. In Section 3, the proposed method for solving a target problem is described in detail. In Section 4, an evaluation of scheduling performance is shown by comparing the proposed method with an imitated manual scheduling method using test data assumed from real logistics warehouse operation. Conclusions are drawn at the end of the paper.

II. Background

A. Problem Setting

In logistics warehouse operations, the number of items to be processed in a day is determined, and the time in which the items are to be processed is the "operation completion time." Operations are composed of multiple processes, such as picking, inspection, and packing, and there are anteroposterior relations between processes. For example, the picking process of taking specific items out of a warehouse is required before an inspection process is conducted. An anteroposterior relation like this is not only a one-to-one but also a one-to-many and a many-to-many relationship. In an operation, the processing unit determines what a piece is, and items are processed as individual pieces. A process is completed when all the pieces are processed. Pieces processed in an anterior process are stored in the buffer set between processes. A posterior process can be started when the number of stored pieces reaches a predetermined number. An anterior process need not stop at that time and can be allowed to continue concurrently with a posterior process. Process completion time is increased or decreased by the number of assigned laborers because productivity per labor per hour is determined in each process. Each time the number of allocated laborers is changed, the process completion time is changed because the assignment number can be changed dynamically before completion of the process. Moreover, sometimes each process has a maximum or a minimum number of laborers required to process.

B. Related Works

Multi-mode resource constrained project scheduling problem with generalized precedence relations (MRCPSP-GPRs) is when there are scheduling problems in planning the start time and resource allocation of a process with an operation composed of multiple processes and anteroposterior relations with permission to conduct anterior processes concurrently with posterior processes [3].

In RCPSP, the maximum number of resources is determined at the same time. MRCPSP is an extended problem of RCPSP and has multiple processing "modes." The number of resources, the kind of resources, and the processing time are determined for each mode. Once a processing mode is determined, this processing mode must continue until completion. There are two types of generalized precedence relations (GPRs): time GPRs and work GPRs. Time GPRs are relations in which a posterior process must start by p hour after an anterior process started. Work GPRs are relations in which p_j % of a posterior process is completed at the time of p_i % of an anterior process completion. In our target problem, the relation between the number of laborers allocated to process and the processing time could be considered as processing modes. Therefore, that processing modes can be changed during processing is an extended problem of MRCPSP-work GPRs.

Although optimization methods with genetic algorithm approaches [4][5] for MRCPSP are proposed, these approaches are not suitable for our target problem due to not considering GPRs. A genetic algorithm approach [3], a mixed integer linear programming approach [6] and a satisfiability modulo theories approach [7] are proposed optimization methods for MRCPSP-GPRs. However, [5],[6] and [7] are not suitable for our target problem because preemption and mode switching are not allowed in their target problems. In [8], a Simulated Annealing approach for MRCPSP with preemption and mode switching is proposed, but it is not applicable to our target problem because GPRs are not considered. There are very few studies in which the problem setting includes all of GPR, preemption, and mode switching. Although [3] is considered GPRs, preemption and mode switching, it is not suitable for our target problem. The reason why there are restrictions that only one person can be assigned to a process at once and an upper bound on the mode switching for any process. If there were not these restrictions, the computational complexity becomes very large. This is explained below (shown with a simple example following Fig. 1).

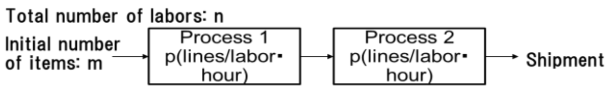


Figure 1. An example of operation.

- An operation is composed of Process 1 and Process 2.
- The number of laborers is n[laborers].
- The initial number of items is m[lines].
- Productivity of each process is p[lines/laborer · h].
- Process 2 can be started when one line is completed at Process 1.

The number of assignments from 0[h] to $\frac{1}{p-n}$ [h] is 1. After $\frac{1}{p-n}$ [h], assuming that the number of assignments for Process 1 is n_1 [laborers], each time $\frac{1}{p-n_1}$ [h] passes, $\frac{n(n+1)}{2}-1$ assignment patterns occur. At this time, at least one laborer should be assigned to each process. This occurs m-1 times until all items are processed in Process 1. Therefore, the total number of patterns is $\left\{\frac{n(n+1)}{2}-1\right\}^{(m-1)}$. For this reason, the order of this problem is $O(n^m)$. Even if the problem is very small scale like two laborers and 50 lines, computational complexity becomes very large: $O(2^{50})$. Therefore, this problem cannot be expected to be solved quickly with previous approaches.

To improve productivity, the TOC method developed by Goldratt is repeatedly applied to processes. The procedures are:

Step 1. Identifying the system's constraints

Step 2. Determining how to exploit the system's constraints

Step 3. Subordinating everything else to the above decision

Step 4. Elevating the system's constraints

Step 5. Going back to Step 1 if a constraint has been broken in the previous steps

The TOC's advantages are that it does not require complex procedures for application and it can be executed quickly because it is rule-based. Trojanowska and Dostatni (2017) applied the TOC to project management [9] and Wang et al. (2010) applied to permutation flow shop scheduling [10]. In Wang et al. (2010), after the processing order at a fixed bottleneck machine is determined by the ant colony algorithm, a schedule is created by applying the TOC to all machines except bottleneck machines. In order to solve our target problem, applying a metaheuristic to a bottleneck process is impossible because productivity of a process is changed by the number of allocated laborers, and a bottleneck process is not fixed. Moreover, as mentioned above, a metaheuristic is also not suitable because the computational complexity is high.

III. Method

A. Approach Based on the TOC

In this paper, "productivity" means productivity of a process, which is calculated by the number of lines per hour at the process. According to the TOC, improvement of total productivity for a whole production line could be realized by finding out a "bottleneck process" and resolving it. Here, a "bottleneck process" is a rate-limiting process of a whole production line. Applying the TOC to the target problem makes the process that has the least effective productivity the bottleneck at the time of labor allocation. So, we define "effective productivity" as the rate-limiting productivity of posterior processes considering anterior processes. Additionally, in this research, the process that has the longest estimated processing time is also considered the bottleneck. Because if all items are not processed during a process, all items are not processed at posterior processes of that process.

B. Scheduling Procedures

We propose three scheduling procedures considered a bottleneck in terms of productivity and processing time hereafter, along with one simple scheduling procedure

[Proc.1] Upper process priority procedure

This procedure prioritizes upper processes when laborers are allocated.

Step 1. Select the uppermost process among the processes that are in operation and the number of allocated laborers does not reach the maximum allocation number of laborers.

Step 2. If the sum of the effective productivity of processes is larger than before allocation, allocate one laborer to that process. Otherwise, select the second uppermost process and go back to the beginning of this step.

[Proc.2] Lowest productivity process priority procedure

This procedure prioritizes lower productivity processes when laborers are allocated.

Step 1. Select the lowest productivity process among the processes that are in operation and the number of allocated laborers does not reach the maximum allocation number of laborers.

Step 2. If the sum of the effective productivity of the processes is larger than before allocation, allocate one laborer to that process. Otherwise, select the second lowest productivity process and go back to the beginning of this step.

[Proc.3] Longer processing time of the unprocessed items process priority procedure

This procedure prioritizes longer processing times of unprocessed item processes when laborers are allocated.

Step 1. Select the process that has the largest value of "the number of unprocessed items divided by the effective productivity of the process" among the processes that are in operation and the number of allocated laborers does not reach the maximum allocation number of laborers. Hereinafter, the value called "X" is the estimated time required for processing unprocessed items. If there is a process or processes in which no laborers are allocated, select the process that has largest value of "the number of unprocessed items divided by the effective productivity of allocating one laborer" among them. Hereinafter, the value is called X'.

Step 2. If the sum of the effective productivity of the processes is larger than before allocation, allocate one laborer to that process. Otherwise, select the second largest X or X' process and go back to the beginning of this step.

[Proc.4] Longer processing time of initial items process priority procedure

This procedure prioritizes longer processing times of initial item processes when laborers are allocated.

Step 1. Select the process that has largest value of "the number of initial items divided by the effective productivity of the process" among the processes that are in operation and the number of allocated laborers does not reach the maximum allocation number of laborers. Hereinafter, the value X means the estimated time required for processing all the items. If there is a process or are processes that no laborers are allocated, select the process that has largest value of "the number of unprocessed items divided by the effective productivity of allocating one laborer" among them. Hereinafter, the value is called X'.

Step 2. If the sum of the effective productivity of processes is larger than before allocation, allocate one laborer to that process. Otherwise, select the second largest X or X' process and go back to the beginning of this step.

Laborers are reallocated to processes in all procedures each time the following events occur.

1. the time when a process is ready to start
2. the time when a process is finished
3. the time when a process cannot continue

Hereinafter, these events are called "allocation timing."

IV. Evaluation

In this section, we evaluate the effectiveness of our method by comparing the four scheduling procedures (Proc.1-4, mentioned above) with an imitated manual scheduling method.

A. The Imitated Manual Scheduling Method

At real production sites, site managers conduct personnel scheduling depending on the situation through trial and error. It is hard to get a grasp of all the processes exactly, so that they tend to give priority to upper process simply. Therefore, the procedures of imitated manual scheduling are:

Step 1. Allocate one laborer to an allocatable process for which no laborers are allocated.

Step 2. If at least one laborer is allocated to all the processes, the upper processes are prioritized in the allocation of labor.

Step 3. If the number of processes for which no laborers are allocated is more than the number of unallocated laborers, the upper processes are prioritized in the allocation of labor.

Step 4. If there is a process or processes whose buffer has more than half the number of the initial number of items, one laborer is moved from the most laborers allocated process to them.

Step 5. If there are multiple first processes, laborers are allocated by the ratio of the initial number of items.

Step 6. Laborers are allocated as evenly as possible to processes that have the same number of anterior processes.

Scheduling methods are evaluated with both of the elapsed time in creating schedules and the operation completion time of the created schedules on computing environment using an Intel Core i7-6700K 4.00 GHz processor. The time for managers to allocate labor to their respective processes is assumed to be around five seconds. So, the elapsed scheduling time in the imitated manual scheduling is set to the allocation timing multiplied by 5 seconds.

B. Evaluation Conditions

In the evaluation, five types of operation are assumed as logistics warehouse operations as follows:

As shown in Fig. 2, the operation of Test 1 is composed of picking, inspection, and packing. In picking, items are collected for shipping. In inspection, the picked items are checked that they match orders. In packing, the inspected items are packed into boxes, etc.

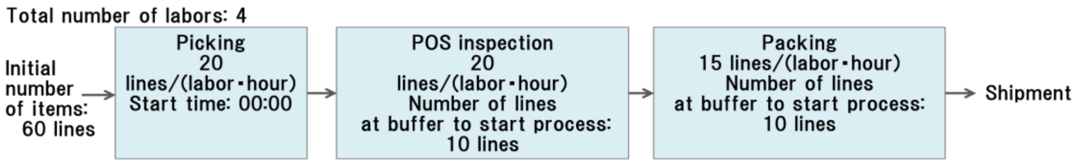


Figure 2. Test 1: operation including three sequential processes.

As shown in Fig. 3, test 2 is an extended Test 1 in that the maximum allocation number of laborers in the inspection process is restricted to 1.

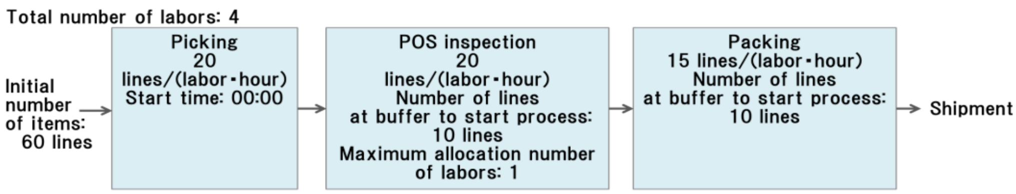


Figure 3. Test 2: operation including three sequential processes with limitation for a number of labor allocation.

As shown in Fig. 4, the operation of Test 3 is composed of automated guided vehicle (AGV) picking, multi-picking, relay picking, point of sales (POS) inspection, and packing, and it contains process branching and joining. In multi-picking, items required by multiple orders are picked all at once. In relay picking, working areas are divided into several blocks, and items picked by block are put into boxes that are on conveyors. In process branching, the distribution ratio to two posterior processes is 1:1.

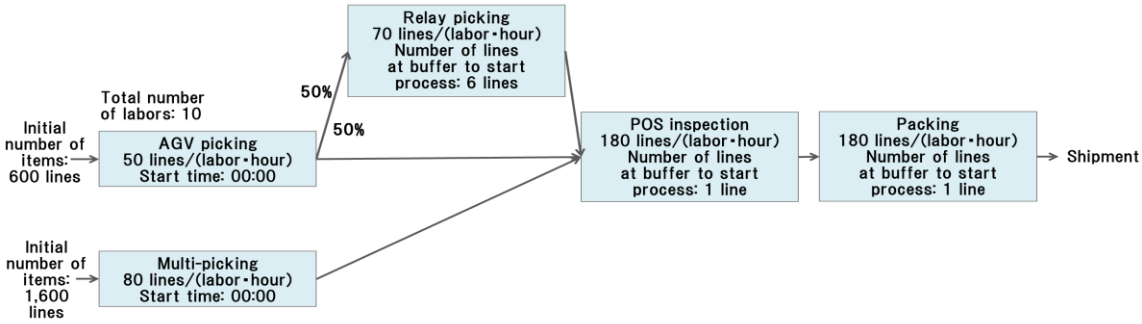


Figure 4. Test 3: operation including three processes joining into one process.

As shown in Fig. 5, test 4 is an extended Test 3 in that the maximum allocation number of laborers of the AGV picking process, the POS inspection process, and the packing process is restricted to 7, 7, and 3, respectively.

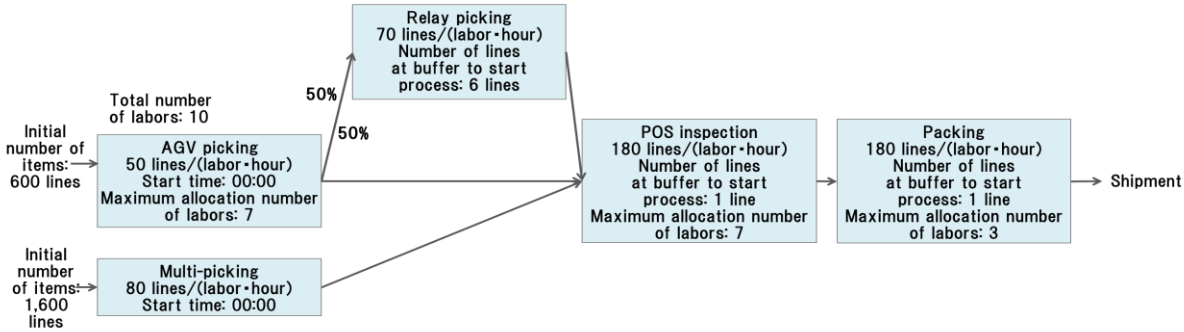


Figure 5. Test 4: operation including three processes joining into one process with limitation for a number of labor allocation.

As shown in Fig. 6, test 5 is composed of ten processes and contains process branching and joining. In process branching, the distribution ratio to two posterior processes is 1:1.

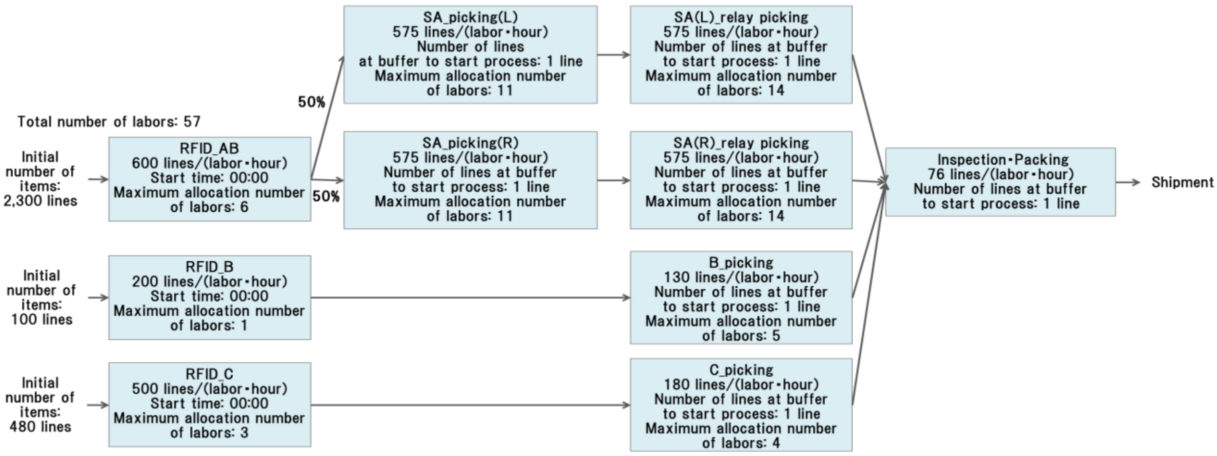


Figure 6. Test 5: operation including four processes joining into one process with limitation for a number of labor allocation.

C. Results

The results of the labor allocations, by which the proposed method and the imitated manual method are respectively conducted for the five test conditions, are shown in Fig. 7 – Fig. 11 and Fig. 12 – Fig. 16. Where, the numbers in the bar charts indicate the number of allocated laborers, and the horizontal line represents the elapsed time. Note that, the proposed method adopts the procedure which is the best time among the four procedures.

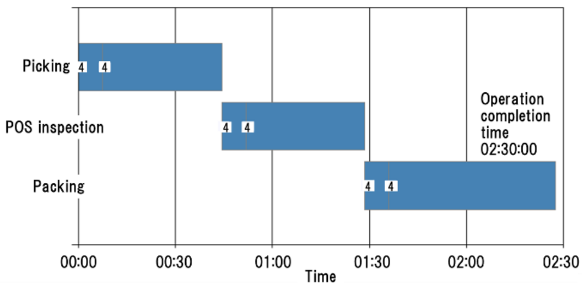


Figure 7. Proposed method schedule for Test 1.

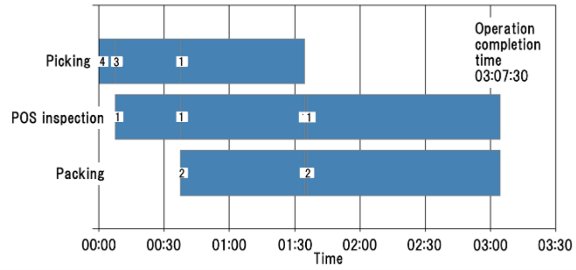


Figure 8. Proposed method schedule for Test 2.

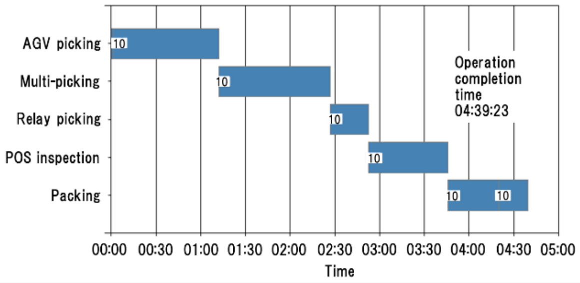


Figure 9. Proposed method schedule for Test 3.

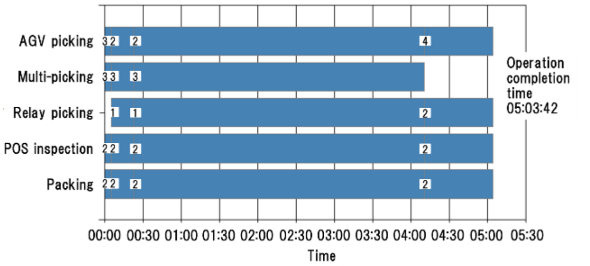


Figure 10. Proposed method schedule for Test 4.

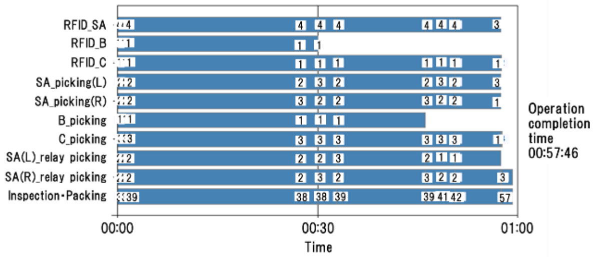


Figure 11. Proposed method schedule for Test 5.

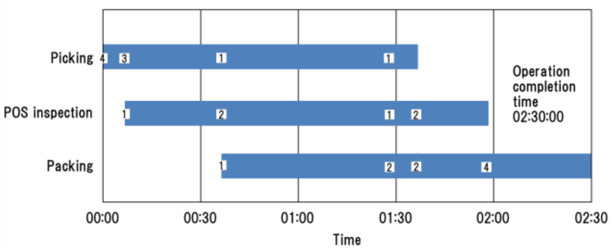


Figure 12. Imitated manual method schedule for Test 1.

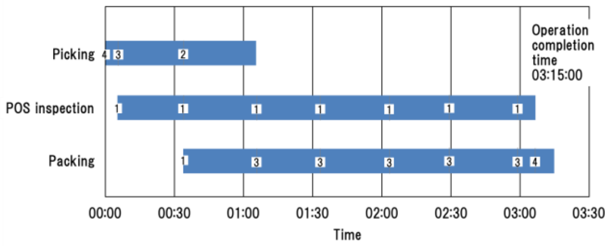


Figure 13. Imitated manual method schedule for Test Data 2.

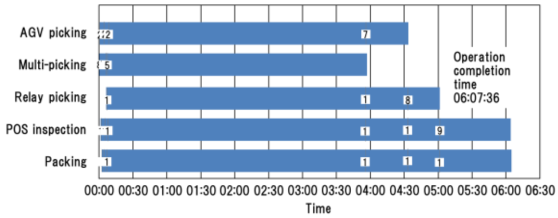


Figure 14. Imitated manual method schedule for Test 3.

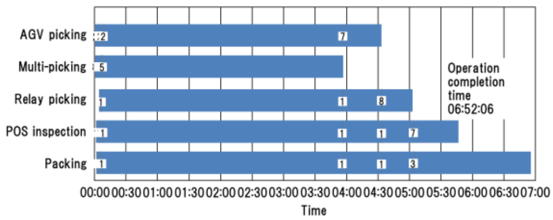


Figure 15. Imitated manual method schedule for Test 4.

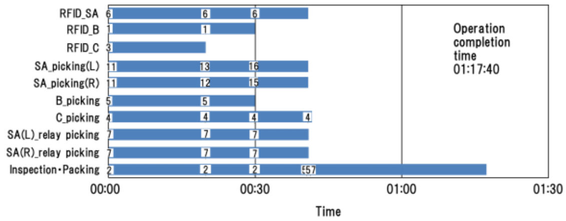


Figure 16. Imitated manual method schedule for Test 5.

The operation completion times are summarized in Table I from Fig. 7 – Fig. 11 and Fig. 12 – Fig. 16. The comparison shows that schedules created by the proposed method have shorter operation completion times than the imitated manual method. However, the best procedure must be selected by creating the schedules and comparing all of them because the best procedure was different for all the test conditions.

Next, the elapsed time of Test 5 in creating schedules is shown in Table II. It is the longest because the total number of laborers and the allocation timing were the greatest in all the test data. Table 2 shows the average elapsed time is 0.295 seconds. In contrast, the allocation timing of the imitated manual method occurs five times, so the elapsed time of this method is 25 seconds. Even when all the proposed scheduling procedures were conducted, the total elapsed time was several seconds. Thus, our methods can create schedules faster than manual personnel scheduling.

Scheduling using Proc.1 (the upper process priority procedure) had the earliest operation completion time of Test 1 and 3. Scheduling using Proc.3 (the longer processing time of the unprocessed items process priority procedure) had the earliest operation completion time of Test 2, 4, and 5 because the maximum allocation number of laborers of all the processes except the first and last processes was lower than the total number of laborers. When Proc.1 was applied to this test condition, the processes with the maximum allocation number of laborers became a bottleneck and the productivity of posterior processes decreased. In the TOC principle, productivity of a bottleneck process must be increased. However, it is impossible due to setting the maximum allocation number of laborers, so Proc.1 is not suitable for Test 2, 4, and 5. In other words, if an operation has no maximum allocation number of laborers like Test 1 and 3, Proc.1 is suitable. Note that, in this case, the number of items at the buffers greatly increases.

The time required for process completion is a suitable criterion for judging whether a process is bottlenecked because Proc.3 is suitable for Test 2, 4, and 5. The reason is because processes with long processing times affect the process completion times of all the posterior processes. Low productivity processes with a low number of items to process barely affects posterior processes and operation completion time.

TABLE I. COMPARISON OF OPERATION COMPLETION TIME BETWEEN PROPOSED METHODS AND IMITATED MANUAL METHOD

Method / Condition	Imitated Manual Method	Proposed Method	Adopted procedures for the proposed method
Test 1	02:30:00	02:30:00	Proc.1
Test 2	03:15:00	03:07:30	Proc.2, Proc.3, Proc.4
Test 3	06:07:38	04:39:23	Proc.1
Test 4	06:52:06	05:03:42	Proc.3, Proc.4
Test 5	01:17:40	00:57:46	Proc.3

V. DISCUSSION

Experimental results show that our proposed method is sufficiently fast and capable of producing short makespan schedules compared to imitated manual method, in a problem setting that is closer to reality than previous studies [3-7].

Of the four proposed scheduling procedures we proposed, we found that upper process priority procedure is suitable for operations where there is no limit on the maximum allocation number of laborers for any process, and longer processing time of unprocessed items process priority procedure is suitable for operations where there is a limit on the maximum allocation number of laborers for one or more processes. Our proposed method cannot always generate global optimal solution, while calculation time is very short. However, methods such as GA used in previous studies have similar drawbacks. In addition, since real-time scheduling is crucial in the real site, the proposed method in our study, which is faster than methods such as GA, is considered to be more suitable for use in real environments. Although a genetic algorithm approach is adopted for solving MRCPSP-GPRs with preemption and mode switching in a previous study [3], the previous study has restrictions that only one person can be assigned to a process at once and an upper bound on the mode switching for any process. Without these restriction, computational complexity would be enormous and it would not be possible to meet the demands of site managers for real-time scheduling. By contrast, since our proposed method is a rule-based scheduling method, it can create schedules very fast even for problems that do not have these restrictions and meet the demands of site managers.

TABLE II. ELAPSED TIME CREATING A SCHEDULE WITH TEST 5

Trial	1	2	3	4	5	6	7	8	9	10	Average
Elapsed Time (sec.)	0.342	0.275	0.313	0.301	0.304	0.311	0.282	0.295	0.265	0.267	0.295

VI. CONCLUSION

In this paper, we proposed a personnel scheduling method for logistics warehouse operation.

Generally, since multiple processes are conducted concurrently in warehouse, its completion time could be improved by timely reallocation of appropriate labor according to progress of each process. However, conventional methods like a metaheuristic and integer programming are not suitable for real-time scheduling due to combinatorial explosions.

In order to address to this problem, the idea of effective productivity has been introduced to the “theory of constraints” (TOC) management method for production lines composed of multiple processes. For the elimination of bottlenecks, which is the essential purpose of the TOC, four types of scheduling procedures are proposed to decide which process is given priority by considering productivity and operation completion time, i.e., lowest productivity process, longer processing time process for both of the unprocessed items and initial items, and lastly upper process.

To evaluate the proposed method, we compared it to an imitated manual scheduling method. Experimental results show that the proposed methods can create schedules faster than manual methods and, in addition, operation completion times of these schedules were shorter.

For future work, we plan to improve our scheduling method that can handle operations including different deadline for each process and develop a method to automatically search for the optimal combination of the four procedures according to the warehouse operations in the real site.

Conflict of Interest

The authors have no conflicts of interest directly relevant to the content of this article.

Author Contributions

Naoya Nishio is involved in design of methodology, software development, verification and writing. Masaharu Kondo is involved in provision of computing resources and reviewing. Ryota Kamoshida is involved in project administration and reviewing. All authors had approved the final version.

References

[1] C. Artigues, S. Demassey, and E. Néron, *Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Application*, Wiley, 2008.

[2] E. M. Goldratt, *Theory of Constraints*, Croton-on-Hudson: North River, 1990.

[3] S. Quintanilla, Á. Pérez, P. Lino, and V. Valls, "Time and work generalised precedence relationships in project scheduling with pre-emption: An application to the management of Service Centres," *European Journal of Operational Research*, vol. 219, no. 1, pp. 59–72, May 2012.

[4] M. R. Afshar, V. Shahhosseini, and M. H. Sebt, "A genetic algorithm with a new local search method for solving the multimode resource-constrained project scheduling problem," *International Journal of Construction Management*, pp. 1–9, 2019.

[5] M. Khalilzadeh, S. Hosseini, and R. Ghaeli, "A genetic algorithm-based method for solving multi-mode resource-constrained project scheduling problem in uncertain environment," *Journal of Project Management*, vol. 5, no. 2, pp. 79–86, 2020.

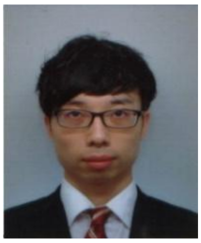
[6] M. Sabzehparvar, H. S. Seyed, and S. Nouri, "A mathematical model for the multimode resource investment problem," *Journal of Industrial Engineering International*, vol. 4, no. 7, pp. 25–32, 2008.

[7] M. Bofill, J. Coll, J. Suy, and M. Villaret, "An efficient SMT approach to solve MRCPSP/max instances with tight constraints on resources," *International Conference on Principles and Practice of Constraint Programming*, pp. 71-79, 2017.

[8] B. Afshar-Nadjafi, "A solution procedure for preemptive multi-mode project scheduling problem with mode changeability to resumption," *Applied Computing and Informatics*, vol. 14, no. 2, pp. 192-201, 2018.

[9] J. Trojanowska, and E. Dostatni, "Application of the theory of constraints for project management," *Management and Production Engineering Review*, vol. 8, 2017.

[10] J. Q. Wang, J. B. Yang, J. Chen, S. F. Zhang, and S. D. Sun, "Multi-bottleneck permutation flow-shop scheduling driven by bottleneck," presented at the 2010 IEEE International Conference on Industrial Engineering and Engineering Management, CN, December 7–10, 2010.



Naoya Nishio obtained master's degree in informatics from Kyoto University, Japan. He is working as a researcher at the Center for Technology Innovation - Artificial Intelligence, Hitachi, Ltd. Research & Development Group. His research interests include Operations Research, Machine Learning, Database Management System, Cognitive Science and Multi-agent Simulation.

Masaharu Kondo obtained a master's degree in science from Hokkaido University, Japan, and doctor's degree in engineering from Yokohama National University, Japan. He is working as a researcher at Data Science Laboratory, Hitachi. Ltd. Research & Development Group. His research interests include Coding Theory, Operations Research and Machine Learning.

Ryota Kamoshida obtained a master's degree in engineering from Waseda University, Japan. He is now working as a senior researcher at the Center for Technology Innovation - Artificial Intelligence, Hitachi, Ltd. Research & Development Group. His research interests include Manufacturing Systems, Operations Research, Combinatorial Optimization and Machine Learning.