Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

SEQUENTIAL DECISION-MAKING FRAMEWORK FOR ROBOTIC MOBILE FULFILLMENT SYSTEM-BASED AUTOMATED KITTING SYSTEM

Jaeung Lee
Sungwook Jang
Young Jae Jang

Department of Industrial & Systems Engineering
KAIST
291, Daehak-ro, Yuseong-gu
Daejeon, 34141, REPUBLIC OF KOREA

Yooeui Jin
Il Kyu Lim
Seungmin Jeong
Eoksu Sim

Global Technology Research
Samsung Electronics
129, Samsung-ro, Yeongtong-gu
Suwon, 16677, REPUBLIC OF KOREA

ABSTRACT

In a flexible production line capable of producing various product types within a single assembly line, an efficient parts supply is critical. The kitting feeding policy, implemented in the flexible production line, aims to kit and supply the necessary parts to the production line without delay. This study investigates the kitting feeding operation for Samsung Electronics' surface-mount device production line. To facilitate the timely supply of parts required for surface-mount device production, Samsung Electronics introduced a robotic mobile fulfillment system-based automated kitting system. This research proposes a sequential decision-making framework to address the kitting operation optimization problem, as well as a kitting scheduling algorithm within the proposed framework. A simulation environment has been implemented to verify the performance of the proposed framework and algorithm through a series of experiments. The experimental results indicate that the proposed framework enhances operational performance and maintains stability, even as the problem size expands.

1 INTRODUCTION

In modern manufacturing systems, flexible production lines capable of producing various product types within a single assembly line have gained prominence. Ensuring the timely supply of materials or parts required at each assembly stage is critical for maintaining the stable operation of such lines. For lines producing a limited variety of products, the *line side stocking policy* is commonly employed, supplying the parts required for each step in units of pallets. However, as the product variety increases, the space occupied by pallets for part supply expands, and workers spend more time searching for the parts they need. To address these challenges, researchers have proposed the *kitting feeding policy* as an efficient part supply policy for flexible production lines. This policy involves supplying the required parts at each stage of the line in a kit, in a predetermined quantity, in a separate working area (i.e., kitting area)(Kilic and Durmusoglu 2015).

The implementation of the kitting feeding policy involves two main stages in supplying parts to each line: preparing the required parts for each line in the form of kits and transporting the prepared kits to the necessary locations in each line. The timely delivery of kits to each line is influenced by the speed of kit preparation in the first stage. Consequently, the efficient operation of the kitting area that prepares the kits is critical to the success of the kitting feeding policy. This study aims to investigate the operation of the kitting area during kitting supply operations to enhance the efficiency of the kit preparation process.

979-8-3503-6966-3/23/$31.00 ©2023 IEEE

The kit preparation process can be considered a type of order-picking process (Hanson et al. 2015). The order-picking process is primarily conducted in a picker-to-parts manner, wherein workers in the warehouse system move to the storage area, select the parts requested by the production line, and deliver them to the kitting area manually (Kulak et al. 2012; Chen et al. 2016; Menendez et al. 2017; Weidinger et al. 2019). However, traditional picker-to-parts-based warehouse systems may experience inefficiencies due to various human factors, such as physical, mental, and psychological strain during order processing (Neumann and Dul 2010).

To address the inefficiencies of the traditional picker-to-parts-based order-picking process, researchers have proposed the parts-to-picker-based order-picking process. In this approach, parts in the warehouse are transported directly to the kitting area, eliminating the need for workers to move to the storage area. Parts-to-picker-based order-processing systems incorporate automated material handling equipment, such as robotic mobile fulfillment systems (RMFSs), carousels, and modular vertical lifts (Boysen et al. 2017).

Among the parts-to-picker systems, RMFS, in which robots deliver movable racks to workstations, is mainly used. Boysen et al. (2017) proposed an algorithm that determines the sequence in which customer requests are processed and the interdependent rack call sequence in a single workstation. Recently, various studies are being conducted to solve the same decision problem in multiple workstations using mathematical optimization or metaheuristic algorithms(Valle and Beasley 2021; Yang et al. 2021; Wang et al. 2022; Zhuang et al. 2022).

Despite the benefits of parts-to-picker-based order-picking systems, potential inefficiencies may arise from manual picking by workers. However, studies on fully automated systems that include the entire process, including picking, are scarce.

In this study, we present a fully automated kitting system that employs manipulator robots as pickers and an RMFS-based order-picking process. Furthermore, we propose a decision-making framework for kitting operations to manage part supply requests in this system. We conducted simulation-based experiments to evaluate the performance of the kitting operation using real-world data when operated according to the derived scheduling.

This paper is structured as follows. Section 2 presents the kitting operation problem concerning the parts supply in Samsung Electronics' surface-mount device (SMD) production line and delineates the kitting operation optimization problem. In Section 3, a practical framework employing a sequential decision-making approach for addressing the kitting operation optimization problem is proposed. Section 4 introduces a heuristic algorithm for kitting scheduling, which constitutes one of the sequential decision-making stages. Section 5 reports the results of simulation experiments conducted using actual parts supply request data from the SMD production line. Lastly, Section 6 offers the conclusion of this paper.

2 PROBLEM DESCRIPTION

We examine the kitting feeding operation of Samsung Electronics' SMD production line, which involves the assembly of small electronic components known as surface-mount components (SMCs) on a printed circuit board (PCB). SMCs are essential for the production of various electronic products, such as smartphones and tablets. These components are stored on reel tapes, with each component attached to the tape individually. Previously, SMC reels were stored in fixed racks within the SMC warehouse, and workers would manually kit the reels onto carts for each line when SMC supply was requested (Sung et al. 2021). With the growing scale of electronic product manufacturing, the demand for SMCs needed for SMD production lines has been increasing, leading to a gradual increase in the kitting operation's capacity required to supply numerous SMCs to each line promptly. In response to this demand, Samsung Electronics proposed an RMFS-based automated kitting system.

Figure 1 illustrates the schematic diagram of the RMFS-based automated kitting system proposed by Samsung Electronics, which is composed of several components. The *inventory* is an area where various parts (i.e. SMC reels) required for production are stored. Each part is stored in a *rackbin*, and several rackbins are placed in the *rack*, which is stored in the inventory. The *kitting box buffer* is an area where

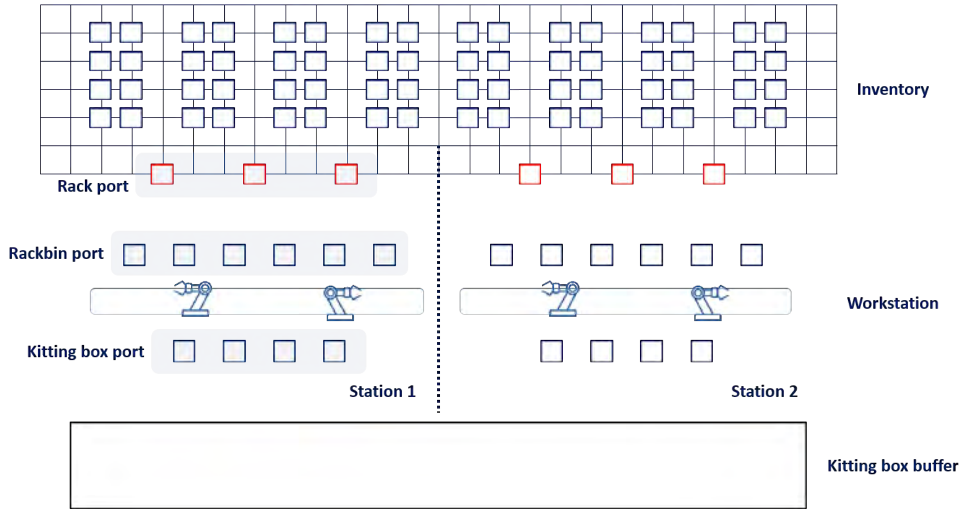


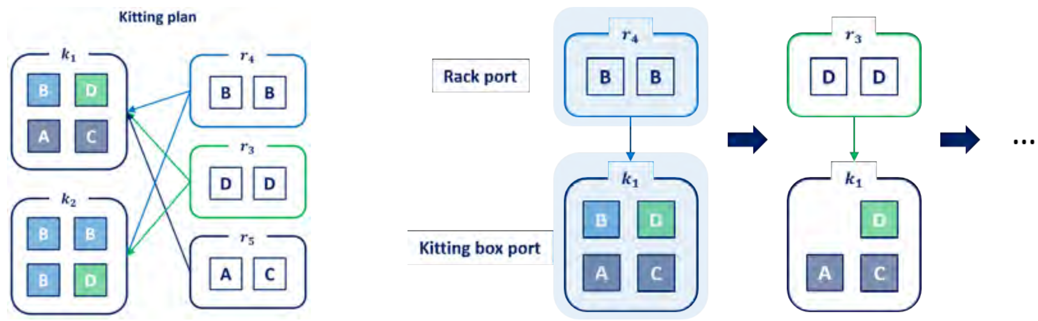
Figure 1: RMFS-based automated kitting system.

the *kitting box* to be supplied to each line is stored. The *kitting workstation* is a space where the picking and kitting work are performed and is divided into several independent areas. Each kitting workstation consists of *rack ports* where the rack is transported from the inventory, *rackbin ports* where the rackbin is taken out of the rack port and located, and *kitting box ports* where the kitting box is transported and located. Finally, the system has *material-handling robots* in charge of transporting between each area, and *kitting robots* responsible for picking and kitting.

The kitting operation in this system is executed through a decision-making process consisting of several steps. Initially, the configuration of the kitting box and the selection of the corresponding rackbins for processing the part supply requests must be determined. This process is referred to as *kitting planning*. A *kitting plan* is then developed, which is a request processing plan that assigns kitting boxes, racks, and rackbins to each part supply request. To execute the kitting work according to the kitting plan, the kitting box must be called to the kitting area, and the rack mapped to the kitting box must be called to retrieve the necessary rackbin into the rackbin port and transport the parts as required.

To process all the kitting plans, it is necessary to decide on the order and manner in which the kitting boxes, racks, and rackbins are called, and the parts are removed and transported in each kitting area. This process is called *kitting scheduling*. Finally, when processing transport requests between areas and ports, multiple material-handling robots must determine the order and routes for processing the transport requests. This process is referred to as *material-handling robot scheduling*. Figure 2 illustrates an exemplary scenario depicting a kitting plan and kitting schedule involving two kitting box ports, three rack ports, and four distinct types of parts.

To perform efficient kitting operations, a decision-making methodology that considers the interactions among different decisions is necessary. This study addresses the *kitting operation optimization problem* for the RMFS-based automated kitting system, which involves finding solutions for kitting planning and kitting scheduling, while excluding material handling robot scheduling. The problem is characterized by batch request processing and plan-ahead decision-making, and assumes that only part supply requests from the same machine can be kitted in one kitting box during kitting planning. The kitting operation optimization problem involves several parameters and inputs, including part supply requests, inventory information, layout parameters, and scheduling parameters. The objective is to minimize the total time required to kit all part supply requests, which is referred to as *makespan*. Table 1 provides the definitions of the notations required for the problem formulation.



(a) (b)

Figure 2: Kitting plan and kitting schedule example.

Table 1: Notations and definitions.

Notation	Definition
I	Number of workstations
w_i	Workstation i; $i \in \{1, 2, ..., I\}$
J	Number of racks
r_j	Rack j; $j \in \{1, 2, ..., J\}$
L	Number of rackbins in each rack
$b_{j,l}$	Rackbin l in rack j; $j \in \{1, 2, ..., J\}$, $l \in \{1, 2, ..., L\}$
M	Number of kitting boxes used in kitting plan
k_m	Kitting box m; $m \in \{1, 2, ..., M\}$
P_{rack}	Number of rack ports in each workstation
$P_{rackbin}$	Number of rackbin ports in each workstation
$P_{kittingbox}$	Number of kitting box ports in each workstation

3 APPROACH

Making optimal decisions for kitting planning and kitting scheduling simultaneously while considering their interaction can be challenging due to a large number of possible solutions. To tackle this issue, this study proposes a sequential decision-making framework that divides the kitting operation optimization problem into three sub-problems. The first sub-problem involves kitting planning, which is further divided into two stages: *rack mapping* and *kitting box mapping*. The resulting kitting plan is then used as input to the second sub-problem, which is kitting scheduling. By reducing the solution space of the kitting scheduling problem in this way, the proposed framework can derive a good solution within a reasonable time.

3.1 Rack Mapping

The first stage of the proposed sequential decision-making framework is the rack mapping step, which aims to derive a list of racks that can cover all part supply requests. This is accomplished by utilizing inventory information and part supply requests that need to be processed. The racks and rackbins are then allocated to each part supply request. The objective of the rack mapping step is to minimize the number of racks used. This is because by reducing the number of racks used, the alternative racks that can be selected during the kitting scheduling step are also reduced, which leads to quick decision-making. The problem of determining a minimal set of racks that can cover all part supply requests is similar to the well-known *set covering problem*. Li et al. (2017) defined a rack selection problem that extends the set covering problem and proposed a hybrid heuristic algorithm. Additionally, if each rack has an adequate number of parts for all types, various algorithms of the set covering problem can be applied, such as algorithms proposed by Chvatal (1979) and Caprara et al. (2000).

3.2 Kitting Box Mapping

The kitting box mapping step involves assigning kitting boxes to part supply requests with racks and rackbins, with the aim of minimizing the number of kitting boxes used and reducing the number of racks required to kit each kitting box. An excessive number of kitting boxes increases the cost of operating the system and necessitates a larger kitting box buffer. Therefore, there is a need to reduce the number of kitting boxes in practical settings. Furthermore, in the kitting scheduling step, similar to the previous step of rack mapping, reducing the number of available kitting boxes allows for efficient decision-making in determining the call order of kitting boxes. Minimizing the required number of racks to kit each kitting box is essential to ensure efficient kitting when calling the racks in the same order. As shown in Figure 3, even if the same number of parts are requested from the same production line, the number of racks required varies depending on the kitting box. This problem is similar to the classical *bin packing problem*, where the goal is to process all part supply requests using minimal kitting boxes. However, in this study, there are constraints on assigning part supply requests to a kitting box, such as only requests from the same machine can be assigned to the same kitting box. Nevertheless, this problem can be transformed into the bin packing problem by classifying the requests that can be assigned together and assigning them to the corresponding kitting box. Therefore, established bin packing algorithms, such as those presented by Johnson (1973) and Korf (2002), can be utilized in this case.

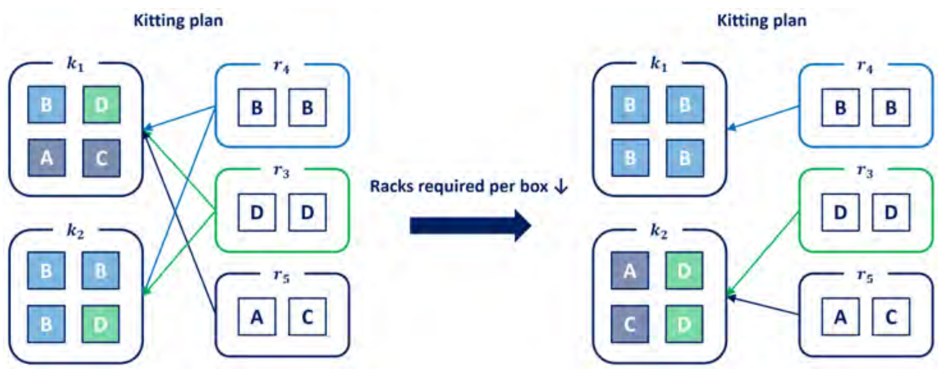


Figure 3: Kitting box mapping comparison.

3.3 Kitting Scheduling

In the context of kitting operations, the kitting scheduling step is responsible for determining the order in which the kitting plan, generated by the kitting box mapping step, will be executed. The kitting schedule is composed of three schedules: *rack scheduling*, *rackbin scheduling*, and *kitting box scheduling*. The relationship between each schedule is interdependent, as the rackbin schedule is constrained by the rack schedule since the rackbin is stored on the rack. Improper scheduling in any of these steps may lead to the unavailability of kitting work at certain points in time. Improper scheduling can result in inefficiencies, exemplified in Figure 4, wherein racks are called despite their inability to process any kitting work. Therefore, it is crucial to ensure proper scheduling in each step to avoid any inefficiencies or delays in the kitting operation.

To ensure that kitting work can be performed at all times, the kitting scheduling process needs to take into account the possibility of *preemption* when calling racks, rackbins, and kitting boxes. Allowing preemption means that once a rack, rackbin, or kitting box enters the port, it is not obligated to complete all the work at once and permits multiple visits to the port. Figure 5 illustrates an example of a kitting work sequence that allows preemption. Even with the reduced solution space resulting from the previous

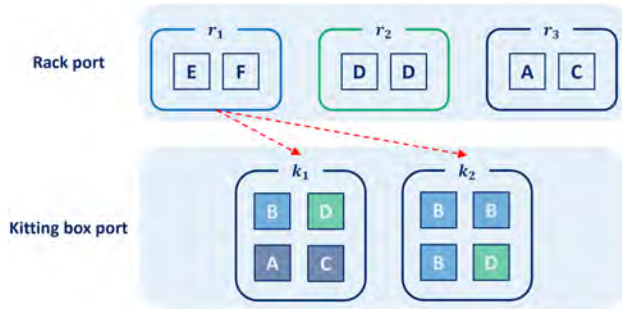


Figure 4: Improper rack call example where $P_{rack} = 3$ and $P_{kittingbox} = 2$.

steps of rack mapping and kitting box mapping, there are still numerous potential solutions for kitting scheduling due to the preemption allowed in each call to a rack, rackbin, or kitting box.

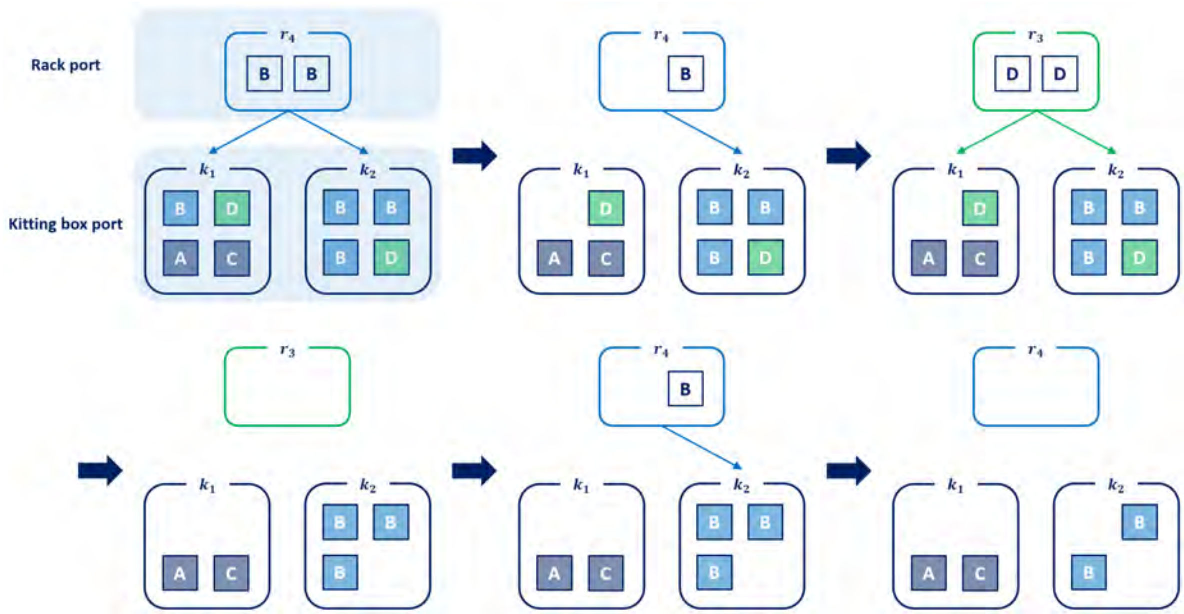


Figure 5: Preemption-allowed kitting work sequence example for the kitting plan in Figure 2 where $P_{rack} = 1$ and $P_{kittingbox} = 2$.

Each rack, rackbin, and kitting box can enter kitting area multiple times. This requires determining the call order and the number of parts to be kitted on each call. This study reduced the solution space of kitting scheduling by limiting preemption as follows, making it easier to explore scheduling solutions. First, The preemption for the rack is restricted so that parts are kitted to all kitting boxes that require the rack for a single rack call. This simplifies the problem to only determining the order to call all racks needed by the kitting plan when scheduling racks. The reason for this restriction is that, in general, the time to transport the rack from the inventory to the kitting area is quite large when a rack is called in RMFS-based warehouse systems. For this reason, if the number of rack calls increases, the rack will not arrive on time due to transport time and the kitting area is likely to be idle. Similarly, preemption for the rackbin is restricted, so parts are kitted to all kitting boxes that require the rackbin for one call, and the problem is simplified to determine the order in which to call rackbins from the rack currently in the rack port. In contrast, preemption for the kitting box is not restricted to avoid the possibility of system deadlock. The kitting schedule derived when the preemption is restricted as described above has the following characteristics. First, the number of rack calls is fixed and equal to the number of racks used in the kitting plan. The number of rackbin calls is fixed and equal to the number of rackbins used in the kitting plan. This allows the kitting scheduling problem to be redefined as rack scheduling, rackbin scheduling, and kitting box scheduling problems that minimize the number of kitting box calls. The reason for minimizing the number of kitting box calls is similar to the reason for restricting the preemption for racks. It is because under the same number of rack calls and rackbin calls, as the number of kitting box calls increases, the possibility that the kitting box will not arrive on time and the kitting area will become idle increases due to the transportation time. In other words, it is because reducing the number of kitting box calls may reduce the makespan.

4 KITTING SCHEDULING ALGORITHM

The kitting scheduling problem, which aims to minimize the number of kitting box calls, bears resemblance to the order-picking problem proposed by Boysen et al. (2017) and Zhuang et al. (2022). In this study, we represent the initial state of kitting area for kitting scheduling as the following example. The notations used in this section conform to the symbols presented in Table 1. Suppose $I = 2$, $P_{rack} = 3$, $P_{rackbin} = 8$, and $P_{kittingbox} = 10$.

- w_1
  - Rack port: ϕ
  - Rackbin port: ϕ
  - Kitting box port: ϕ
- w_2
  - Rack port: ϕ
  - Rackbin port: ϕ
  - Kitting box port: ϕ

The state of each kitting area comprises three parts: the rack port state, the rackbin port state, and the kitting box state. Each state is represented as a set, and the maximum cardinality of the set is determined by the given parameters. The elements of each set are racks, rackbins, and kitting boxes. During kitting scheduling, the state of each set can be altered in the following ways.

- w_1
  - Rack port: $\{r_1,\ r_2,\ r_3\}$
  - Rackbin port: $\{b_{1,2},\ b_{1,3},\ b_{1,7},\ b_{2,4},\ b_{2,8},\ b_{2,10},\ b_{3,1},\ b_{3,2}\}$
  - Kitting box port: $\{k_1,\ k_2,\ k_3,\ k_4,\ k_5,\ k_6,\ k_7,\ k_8,\ k_9,\ k_{10}\}$

Kitting scheduling starts from the initial kitting area state and is performed sequentially by rack scheduling, rackbin scheduling, and kitting box scheduling. Rack scheduling consists of two steps. The first step in rack scheduling is to remove the rack from the rack port if there is a rack to be removed. A rack to be removed is defined as a rack that no longer has parts to be kitted in current kitting boxes in kitting box ports and no rackbin taken out of itself in rackbin ports. The second step in rack scheduling is to add a new rack to the rack port state of kitting scheduling if there is an empty rack port (the cardinality of the rack port state is less than P_{rack}). The rack to be added is determined by *the minimum box conflict rule* (MCFR).

MCFR has been proposed as a means to address idle time that may arise from the simultaneous calling of a single kitting box by multiple workstations. When multiple workstations attempt to utilize the same kitting box simultaneously, certain workstations may be unable to proceed with kitting works until the kitting box has been removed from another workstation, leading to idle time. MCFR is applied in two distinct cases. Firstly, in situations where all rack ports in a non-target workstation are currently empty, a rack with the largest number of kitting boxes to be kitted is selected and added to the schedule. Secondly, in cases where rack ports in other workstations are not empty, an alternative rack is selected based on the smallest intersection of the list of kitting boxes to be kitted by the rack with the list of kitting boxes to be kitted by the racks in the rack ports of other workstations. Through the implementation of MCFR, the likelihood of multiple workstations calling the same kitting box can be minimized in advance, thus reducing the incidence of idle time during kitting tasks.

After completing the rack scheduling phase in kitting scheduling, the subsequent step is rackbin scheduling, which is also performed in two steps. The first step involves removing the rackbin from the rackbin port if there exists a rackbin that no longer contains any parts to be kitted in the current kitting boxes in the kitting box ports. The second step in rackbin scheduling is adding a new rackbin to the rackbin port state of the kitting scheduling if the cardinality of the rackbin port state is less than the prescribed limit of $P_{rackbin}$. The choice of the rackbin to be added is governed by the *largest quality rackbin first rule* (LQRFR), which selects and adds the rackbin containing the largest quantity of parts to be kitted.

Following the completion of rackbin scheduling, the next step is kitting box scheduling, which consists of two stages. Firstly, if there exists a kitting box that has no further parts to be kitted from the rackbins in the rackbin ports, it is removed from the kitting box port. Secondly, if the cardinality of the kitting box port state is less than $P_{kittingbox}$, a new kitting box is added to the kitting scheduling's kitting box port state. The kitting box to be added is selected using the *largest quantity kitting box first rule* (LQBFR), which selects and adds the kitting box with the largest quantity of parts to be kitted from the rackbins in the rackbin ports in the current workstation.

Following the completion of kitting box scheduling, the kitting area state is updated. To update the kitting area state, the rackbin to carry out the unit kitting work must first be selected. The target rackbin containing the parts that must be kitted to the kitting boxes in the current kitting box ports from the rackbins in the rackbin ports is chosen. In the event that there are several rackbins containing parts to be kitted, the first rackbin that has entered the rackbin port is chosen.

After updating all kitting box schedules with the parts to be kitted, we return to the rack scheduling stage and repeat the process until all kitting work has been completed.

5 EXPERIMENTS

This section presents the experimental evaluation of the proposed sequential decision-making framework, and the obtained results are analyzed. To assess the performance of the heuristic algorithm, we conducted two experiments.

5.1 Simulation Environment

In this study, we developed a simulation environment using the Python programming language to evaluate the performance of the kitting scheduling solution obtained from the proposed sequential decision-making framework. The simulation environment is implemented using the Simpy package, which is a discrete-event simulation framework based on a process-oriented approach. To facilitate rapid performance verification under various environmental conditions, we simplified the movement of logistics robots and manipulators during kitting work. Table 2 presents the key parameters used in the simulation.

The simulation was developed to execute the kitting work by calling racks, rackbins, and kitting boxes to each workstation based on the kitting plan and scheduling solution in the designated order. The simulation considers the availability of logistics robots and the location of the called rack or kitting box. If the required resource is not available or is present in another workstation, the kitting workstation waits for the resource to become available, replicating the real-world scenario.

The simulation environment utilized in this experiment is established with the following configuration parameters: I = 2, P_{rack} = 5, $P_{rackbin}$ = 7, and $P_{kittingbox}$ = 15. R_{rack} is assumed to be large enough so that

Table 2: Simulation parameters.

Notation	Definition
t_{rack}	Average transport time between inventory and rack port
$t_{rackbin}$	Average transport time between rack port and rackbin port
$t_{kittingbox}$	Average transport time between kitting box buffer and kitting box port
$t_{kitting}$	Average kitting time per part by a manipulator
R_{rack}	Number of material handling robots between inventory and rack port
$R_{rackbin}$	Number of material handling robots between rack port and rackbin port in each workstation
$R_{kittingbox}$	Number of material handling robots between kitting box buffer and kitting box port

there is no delay due to the lack of robots between inventory and rack port. The remaining simulation parameters are outlined below.

- t_{rack}: 120.15 sec
- $t_{rackbin}$: 5.65 sec
- $t_{kittingbox}$: 14.08 sec
- $t_{kitting}$: 9.5 sec
- $R_{rackbin}$: 1
- $R_{kittingbox}$: 7

5.2 Experimental Data

The present study employed a dataset comprising 10 scenarios of part supply requests for a day obtained from Samsung Electronics' mobile phone manufacturing plant. Each scenario's part supply requests consist of thousands of parts in total, encompassing hundreds of different types of parts, which are requested from multiple production lines on a daily basis. For each part supply request, we derived a kitting plan by following the sequential decision-making framework proposed in this study, which involved the steps of rack mapping and kitting box mapping. Rack mapping was performed using an existing algorithm employed within Samsung Electronics and kitting box mapping was carried out using a heuristic algorithm based on the Next-fit algorithm (Johnson 1973), a widely used heuristic for the bin packing problem. Finally, the kitting plans employed to compare scheduling performance encompassed approximately 200 racks and 300 kitting boxes.

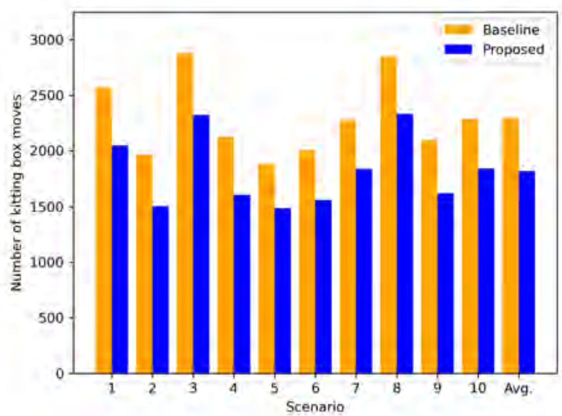
5.3 Results

For each scenario, we compared the performance of the scheduling rule used in the field, referred to as the baseline algorithm, with the proposed scheduling heuristic algorithm. The baseline algorithm randomly assigns racks to each workstation and then determines the order of rack and kitting box calls according to their indices.

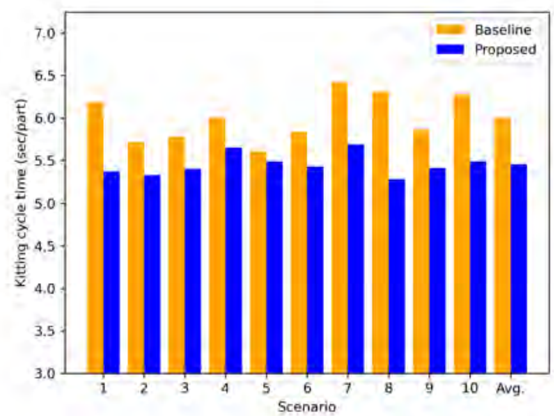
To evaluate the performance of the two algorithms, we use key performance indicators (KPIs) including the number of kitting box moves and kitting cycle time. The number of kitting box moves is the sum of the number of kitting box calls for each workstation based on the kitting schedule. Kitting cycle time refers to the average time needed to process a part supply request, obtained by dividing the makespan by the total number of part supply requests when the kitting work is performed according to the kitting schedule. The simulation was conducted on X64, Intel® Core™ i5-10400 CPU @ 2.90GHz and 16GB RAM.

As shown in Figure 6, the proposed algorithm reduces the number of kitting box moves by 21% on average, with a minimum reduction of 18% and a maximum of 25%, compared to the baseline algorithm.

Additionally, the kitting cycle time for the proposed algorithm decreases by 9% on average, with a minimum decrease of 2% and a maximum of 16%, compared to the baseline algorithm.



(a)



(b)

Figure 6: The number of kitting box calls and kitting cycle time comparison among different scenarios.

In the subsequent analysis, we evaluate the performance of the proposed scheduling algorithm as a function of part supply request size. To this end, we create a reference part supply request with an average part request quantity of 10 scenarios based on the data from the first experiment. We then experiment with data generated by scaling the reference part supply request. The key performance indicators for this experiment are the execution time of the scheduling heuristic algorithm and kitting cycle time.

As depicted in Figure 7, the experimental results demonstrate that the execution time of the proposed algorithm exhibits a quadratic increase with the size of the part supply request. However, despite the increase in the size of the part supply requests, the kitting cycle time for the proposed scheduling algorithm remains stable.

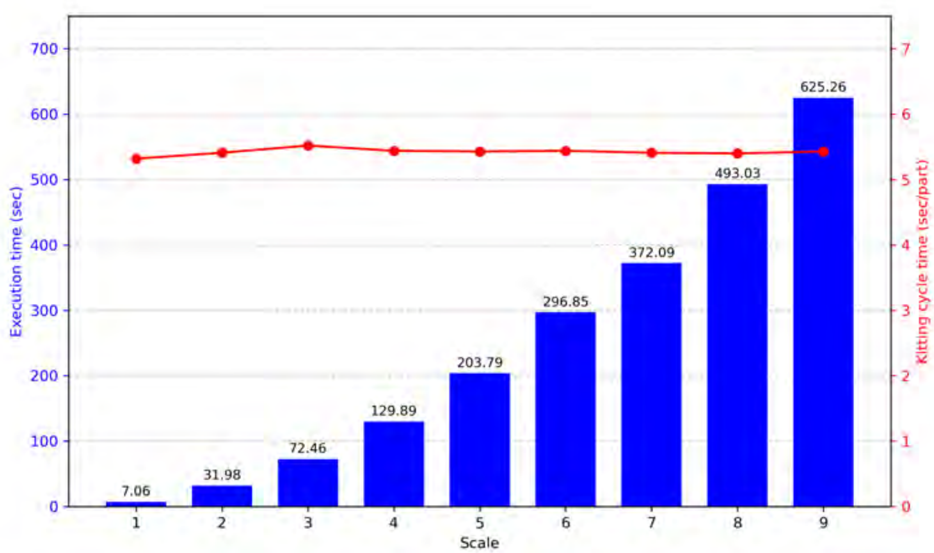


Figure 7: Execution time and kitting cycle time by scale.

6 CONCLUSION

This research presents a decision-making framework designed to optimize the kitting operation for RMFS-based automated kitting systems. This framework breaks down the kitting process into three distinct sub-problems, enabling a more streamlined approach to improve efficiency and performance.

The first two sub-problems are tackled using heuristic algorithms to generate kitting plans through rack and kitting box mapping. The third sub-problem focuses on kitting scheduling, which itself is divided into three smaller components, each with a proposed algorithm. Experimental results have been promising, with the new algorithm outperforming the baseline algorithm by an average of 15%. This improvement leads to better kitting operation performance and reduces the workload for material-handling robots. Additionally, the kitting cycle time remains consistent even as part supply request sizes increase.

Despite these promising results, the study acknowledges that there is room for improvement in the rack and kitting box mapping problems. The current objectives of minimizing the number of racks and kitting boxes used may not directly correlate with maximizing kitting operation performance. Future studies could explore alternative rack and kitting box configurations to further enhance overall efficiency.

The study also allowed for the preemption of kitting boxes while disallowing the preemption of racks and rackbins. This decision was based on the time it takes for a rack to arrive at the rack port after being called. However, allowing preemption of racks and rackbins could lead to shorter scheduling makespans, and researchers recommend exploring this possibility in future studies.

Lastly, the study highlights that kitting operation performance may vary significantly depending on the composition of racks in the inventory. To address this, we suggest examining the storage assignment problem to determine which parts should be stored on each rack and where to place the racks. This research could contribute to finding the optimal rack configuration that maximizes kitting operation performance.

In conclusion, this decision-making framework shows promise in improving the efficiency of flexible automated kitting systems. With further research and optimization, the potential for even greater gains in kitting operation performance is on the horizon.

ACKNOWLEDGMENTS

This work was supported by Global Technology Research at Samsung Electronics, Ministry of Education of the Republic of Korea, and National Research Foundation of Korea (NRF-2021R1A2C3008172).

REFERENCES

Boysen, N., D. Briskorn, and S. Emde. 2017. “Parts-to-Picker Based Order Processing in a Rack-Moving Mobile Robots Environment”. *European Journal of Operational Research* 262(2):550–562.

Caprara, A., P. Toth, and M. Fischetti. 2000. “Algorithms for the Set Covering Problem”. *Annals of Operations Research* 98(1-4):353–371.

Chen, F., H. Wang, Y. Xie, and C. Qi. 2016. “An ACO-Based Online Routing Method for Multiple Order Pickers with Congestion Consideration in Warehouse”. *Journal of Intelligent Manufacturing* 27:389–408.

Chvatal, V. 1979. “A Greedy Heuristic for the Set-Covering Problem”. *Mathematics of Operations Research* 4(3):233–235.

Hanson, R., L. Medbo, and M. I. Johansson. 2015. “Order Batching and Time Efficiency in Kit Preparation”. *Assembly Automation* 35(1):143–148.

Johnson, D. S. 1973. *Near-Optimal Bin Packing Algorithms*. Ph. D. thesis, Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Kilic, H. S., and M. B. Durmusoglu. 2015. “Advances in Assembly Line Parts Feeding Policies: A Literature Review”. *Assembly Automation* 35(1):57–68.

Korf, R. E. 2002. “A New Algorithm for Optimal Bin Packing”. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada*, edited by R. Dechter, M. J. Kearns, and R. S. Sutton, 731–736.

Kulak, O., Y. Sahin, and M. E. Taner. 2012. “Joint Order Batching and Picker Routing in Single and Multiple-Cross-Aisle Warehouses Using Cluster-Based Tabu Search Algorithms”. *Flexible Services and Manufacturing Journal* 24:52–80.

Li, Z., J. Zhang, H. Zhang, and G. Hua. 2017. “Optimal Selection of Movable Shelves under Cargo-to-Person Picking Mode”. *International Journal of Simulation Modelling* 16(1):145–156.

Menendez, B., E. G. Pardo, J. Sánchez-Oro, and A. Duarte. 2017. "Parallel Variable Neighborhood Search for the Min–Max Order Batching Problem". *International Transactions in Operational Research* 24(3):635–662.

Neumann, W. P., and J. Dul. 2010. "Human Factors: Spanning the Gap between OM and HRM". *International Journal of Operations & Production Management* 30(9):923–950.

Sung, S. W., S. Jeong, S. Park, E. Sim, and C. K. Kim. 2021. "Warehouse Storage Assignment for Surface Mount Components Using an Improved Genetic Algorithm". In *Proceedings of the 2021 Winter Simulation Conference*, edited by S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, 1–10. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Valle, C. A., and J. E. Beasley. 2021. "Order Allocation, Rack Allocation and Rack Sequencing for Pickers in a Mobile Rack Environment". *Computers & Operations Research* 125:105090.

Wang, B., X. Yang, and M. Qi. 2022. "Order and Rack Sequencing in a Robotic Mobile Fulfillment System with Multiple Picking Stations". *Flexible Services and Manufacturing Journal*:1–39.

Weidinger, F., N. Boysen, and M. Schneider. 2019. "Picker Routing in the Mixed-Shelves Warehouses of E-commerce Retailers". *European Journal of Operational Research* 274(2):501–515.

Yang, X., G. Hua, L. Hu, T. Cheng, and A. Huang. 2021. "Joint Optimization of Order Sequencing and Rack Scheduling in the Robotic Mobile Fulfilment System". *Computers & Operations Research* 135:105467.

Zhuang, Y., Y. Zhou, Y. Yuan, X. Hu, and E. Hassini. 2022. "Order Picking Optimization with Rack-Moving Mobile Robots and Multiple Workstations". *European Journal of Operational Research* 300(2):527–544.

AUTHOR BIOGRAPHIES

JAEUNG LEE received a B.S. degree in the Department of System Management Engineering from Sungkyunkwan University in 2018, and an M.S. degree in the Department of Industrial & Systems Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2020, where he is currently pursuing his Ph.D. in the same department. His research interests include automated material handling systems (AMHSs), traffic management, and reinforcement learning. His email address is ira.hyper@kaist.ac.kr.

SUNGWOOK JANG received a B.S. degree in the Department of Mathematical Sciences from Korea Advanced Institute of Science and Technology (KAIST) in 2019 and an M.S. degree in the Department of Industrial & Systems Engineering from KAIST in 2021. He is currently pursuing his Ph.D. degree in the Department of Industrial & Systems Engineering at KAIST. His research interests include automated material handling systems (AMHSs), storage assignment problems, and vehicle dispatching algorithms. His email address is jedi829@kaist.ac.kr.

YOUNG JAE JANG received his B.S. degree in Aerospace Engineering from Boston University in 1997, and double M.S. degrees in Mechanical Engineering and Operations Research, and a Ph.D. in Mechanical Engineering from Massachusetts Institute of Technology (MIT) in 2001 and 2007, respectively. He is currently an associate professor in the Department of Industrial & Engineering at Korea Advanced Institute of Science and Technology (KAIST). His email address is yjang@kaist.ac.kr.

YOOEUI JIN is a principal engineer in Global Technology Research at Samsung Electronics. She earned her master's degree in the Department of System Management Engineering at Sungkyunkwan University. Her research interests include production simulation, digital twin, and material control systems. Her e-mail address is yooeui.jin@samsung.com.

IL KYU LIM is a principal engineer in Global Technology Research at Samsung Electronics. His research interests include optimization methodology for material supply systems and logistics hardware. He is the leader of this project and planned and designed logistics automation. His e-mail address is iklim77@samsung.com.

SEUNGMIN JEONG is a staff engineer in Global Technology Research at Samsung Electronics. He earned his Ph.D. degree in the Department of Industrial and Systems Engineering at Korea Advanced Institute of Science and Technology (KAIST). His research interests include the mathematical modeling of complex systems, optimization methodologies, networks, and logistics systems. He has investigated the optimization of the manufacturing logistics design and operation. His e-mail address is sm.jeong@samsung.com.

EOKSU SIM is a principal professional and the head of the Manufacturing Logistics Solution Group in Global Technology Research at Samsung Electronics. He earned his Ph.D. in the Department of Industrial Engineering at Seoul National University. His research interests include smart factories, logistics, supply chain management (SCM), and optimization. His email address is eoksu.sim@samsung.com.