Sequential batching with minimum quantity commitment in N-level non-exclusive agglomerative hierarchical clustering structures

Seung-Kil Lim[a*]

[a]*Department of Industrial and Management Engineering, Sungkyul University, 400-10 Anyang-dong, Manan-gu, Anyang-city, Gyeonggi-do 14097, South Korea*

CHRONICLE	ABSTRACT
	This study considers a sequential batching problem with a minimum quantity commitment (MQC) constraint in N-level non-exclusive agglomerative hierarchical clustering structures (AHCSs). In this problem, batches are created for item types included in clusters according to the sequence of the levels in a given AHCS such that the MQC constraint as well as the maximum and minimum batch size requirements are satisfied, simultaneously. The MQC constraint ensures that more items than a committed minimum quantity must be batched at a level before items not batched at the level are sent to the next level. We apply the MQC constraint to control effectively the degree of heterogeneity (DoH) in the batching results. We developed a sequential batching algorithm for minimizing the total processing cost of items using properties identified to find better solutions of large-sized practical problems. Results of computational experiments showed that the developed algorithm found very good solutions quickly and the heuristic algorithm could be used in various practical sequential batching problems with the MQC constraint such as input lot formations in semiconductor wafer fabrication facilities, determination of truckloads in delivery service industry, etc. Also, we found some meaningful insights that dense cluster, small batch size, and tight MQC constraint are effective in reducing the total processing cost. Additionally, small batch size with loose MQC constraint seem to be helpful to reduce the DoH in the batching results. Finally, we suggested that the density of cluster, batch size, and MQC tightness should be determined simultaneously because of interactions among these factors.

1. Introduction

Clustering is the classification of a set of item types into clusters so that item types in the same cluster have more similar attributes to each other than those in other clusters. Examples of clustering include segmenting various customers, classifying product types into product lines or families, and grouping delivery destinations into several zones. We can group item types into clusters only once; however, sometimes, clusters that were already made are grouped again into larger clusters or divided again into smaller clusters. This kind of clustering is called hierarchical clustering or multi-level clustering. Item types in the same cluster become more homogeneous in the divisive hierarchical clustering structure (DHCS), whereas they become more heterogeneous in the agglomerative hierarchical clustering structure (AHCS) as the level of hierarchy increases. In the AHCS, a cluster includes more and more item types

* Corresponding author
E-mail: seungkil@sungkyul.ac.kr (S-K., Lim)

as the level of the hierarchy increases. Also, AHCS can be configured so that an item type included in a cluster at a level belongs to only one cluster at the next level, or it has a configuration that an item type included in a cluster is included in more than one cluster at the next level. We call the former the exclusive AHCS and the latter the non-exclusive AHCS.

Many studies have been conducted on clustering in areas of manufacturing, biotechnology (BT), information technology (IT), and logistics. Refer to Murtagh and Contreras (2012), Lim et al. (2017), and Delgoshaie et al. (2019) for previous studies on hierarchical clustering algorithms. In manufacturing, there are various studies on clustering techniques for cell formation of cellular manufacturing systems. Vakharia and Wemmerlöv (1995) compared the performance of seven techniques for hierarchical clustering with ten performance measures and identified that selecting suitable clustering technique is more critical than the choice of dissimilarity measure for better cell formation. Recently, Delgoshaei et al. (2019) conducted a literature review on clustering methods successfully used for the cell formation to explain drawbacks and shortcomings of cell forming and scheduling the formed cells. Also, clustering techniques are applied to scheduling and ordering problems. Chen et al. (2011) transformed a single machine batch scheduling into a special type of clustering problem and developed a solution algorithm using clustering techniques. Nagasawa et al. (2012) developed a method using canonical correlation and hierarchical cluster analysis (HCA) for classifying items according to shipping trends to determine a suitable ordering policy. Clustering is also actively being used in IT field. Arifin and Asano (2006) used HCA to develop a method for image segmentation by a histogram thresholding. Costa et al. (2013) developed a hierarchical method to classify structurally homogeneous clusters of XML documents considering multiple forms of structural components. Nunez-Iglesias et al. (2013) suggested an active learning paradigm for image segmentation to divide an image into meaningful parts using agglomerative hierarchical clustering. Bouguettaya et al. (2015) combined agglomerative hierarchical clustering and partitional clustering to reduce computational costs without losing the clustering performance while handling huge data collections. Zaitoun and Aqel (2015) compared clustering-based image segmentation techniques and recommended a hybrid solution that combines multiple methods for image segmentation because many factors influence the results of image segmentation. Huang and Ma (2019) suggested a new hybrid clustering method, which combines hierarchical clustering and K-means for image categorization.

Clustering techniques are widely used in BT field for detecting clusters in genomic data (Langfelder et al. 2008, Cameron et al. 2012), expressing gene (Jun et al. 2010), classifying protein families (Saunders et al. 2012), and finding the hierarchical clustering structure (HCS) for the DNA sequence (Cheng et al. 2013). Andreopoulos et al. (2009) surveyed important clustering applications in biomedicine to find strengths and weaknesses of existing clustering algorithms and provide guidelines for matching them to biomedical applications. In the logistics field, clustering techniques are used for clustering demand nodes to solve vehicle routing problems (Özdamar & Demir, 2012; Coral et al., 2017, Comert et al., 2018; Hintsch & Irnich 2020), flow mapping, and clustering to effectively discover major flow patterns from large point-to-point spatial flow data (Zhu & Guo, 2014) and evaluating intercity ground transportation infrastructure and services to support the integration and development of urban agglomerations (Yue et al., 2019).

This study considers a sequential batching problem (SBP), wherein batching priorities are assigned based on AHCSs. The sequential batching arises in many areas including input lot formations in semiconductor wafer fabrication facilities, known as wafer fabs, bulk mail presorts for discounted mailing fees, and truck or container loads determinations in vendor managed inventory (VMI) environments. Unlike traditional batching problems, in this problem, batches are created sequentially from the first level of the AHCS given the highest priority to the last level given the lowest batching priority. For example, in a wafer fab, we can create an AHCS for batching priorities in the form of product hierarchy, which represents the hierarchical relationship among products. Product types are placed at the first level of the

product hierarchy, whereas product lines, which are groups of product types with similar production processes, are placed at the second level. Product families, which are groups of product lines with a certain functional coherence, can be placed at the next level. The number of levels in an AHCS varies depending on the industry or company. In the postal service industry, the number of presort levels becomes that of the levels in the AHCS. For example, the Korea Post and United States Postal Service (USPS) have three and four presort levels for commercial bulk mail, respectively. Typically, many companies, including semiconductor manufacturers in other industries, have more than three hierarchical levels.

In a wafer fab, wafers are made up of lots and then released into the wafer fab, where the maximum number of wafers in a lot (i.e., lot size) is 25 in general. Also, a lot can be made up of wafers for single type of product or multiple types of products. We call the former homogeneous wafer lot and the latter heterogeneous wafer lot, which is split into several lots for multiple products during the wafer fabrication process. Because quantities to be released for product types are not a multiple of the lot size, the total number of lots (including the number of lots with a few wafers) increases significantly, which influences the productivity of the wafer fabrication if only homogeneous wafer lots are created. Therefore, it is required to create a certain number of heterogeneous wafer lots. However, it is important to create homogeneous wafer lots for each product type adequately before forming heterogeneous wafer lots for multiple product types, because excessive lot-split can lead to higher production costs. Here, an AHCS (based on product hierarchy) for product types is used to set priorities for sequential batching, wherein lots are formed sequentially from the first level of the AHCS to the last level. Particularly, the input lot formation by the sequential batching is very important for the customized small-volume production of semiconductor chips, such as application specific integrated circuits (ASIC) because hundreds of different kinds of products are fabricated in a typical wafer fab for ASIC chips of Korean global semiconductor manufacturers. See Kim and Lim (2012), Bang et al. (2012) and Lim et al. (2014) for the release of input lots and lot-split during the wafer fabrication process.

In the sequential batching process for the input lot formation, to avoid excessive lot-split, a certain number of homogeneous wafer lots is first formed for each product type before heterogeneous wafer lots for multiple product types are created. We refer to this number of (homogeneous) wafer lots as the committed minimum quantity at the first level of the AHCS, which should be appropriately set to control the heterogeneity degree more effectively, because the committed minimum quantity influences the number of lots with a few wafers and the total number of lots in the wafer fab, as mentioned above. Here, the degree of heterogeneity (DoH) can be defined as the ratio of production orders for which production quantities are satisfied by split lots from heterogeneous wafer lots. In this study, we include a special constraint, called the minimum quantity commitment (MQC) constraint, to effectively control the DoH in the batching results. The MQC constraint ensures that more wafers than the committed minimum quantity are formed as lots at a level before wafer lots are made at the next level of the AHCS. As high heterogeneous wafer lots lead to excessive lot-split in the wafer fabrication process, multiple deliveries for destinations occur from heterogeneous truckloads that include items for several destinations. Excessive multiple deliveries for destinations increase the total delivery costs and reduce customer satisfaction. The MQC constraint enforces that more items than the committed minimum quantity are formed as truckloads for each destination to avoid excessive multiple deliveries before truckloads for multiple destinations are formed.

As a study on the SBP for an exclusive AHCS, Lim et al. (2015) studied a presort loading problem (PLP), where three-level AHCS is defined by a discounted structure of mailing fees for presorted commercial bulk mail. Batches of mail pieces to be loaded onto mail trays are created for zip codes included in clusters according to the sequence of the levels in a given AHCS such that the maximum capacity of the mail tray and the minimum requirement for discounted mailing fees are satisfied. Lim et al. (2015) proved that PLP is a special case of transportation problem (TP) with the MQC constraint, where suppliers are divided into three levels, transportation costs from suppliers in the same level to any customers are not

different from each other while transportation costs increase as the level of supplier increases and the increasing rate is gradually decreased by the tapering principle. They developed an exact solution method for the special case although the TP with the MQC constraint is NP-hard (Lim & Xu, 2006).

Bang et al. (2016) developed a multi-dimensional dynamic programming (DP) algorithm for an input-lot formation in a semiconductor wafer fab, which is a kind of the SBP without the MQC constraint under *N*-level exclusive AHCS. As an extended version of Lim et al. (2015), Bang et al. (2016) extended the level of the AHCS from 3 to *N* and considered more general cost structure that does not follow the tapering principle. Lim et al. (2017) reduced the number of states in the DP algorithm developed by Bang et al. (2016). The DP algorithm, however, is not suitable to solve practical problems due to high computational complexity and very long computation time. Also, they did not consider non-exclusiveness of the AHCS and the MQC constraint for effective control over the DoH in the batching results, although they frequently appear in many application areas. See Lim et al. (2015) and Han et al. (2019) for a review of some studies on the MQC constraint. Studies on the batching problem have been conducted since a long time. However, there is no previous research on the SBP with the MQC constraint under the non-exclusive AHCS, although the MQC concept has been applied to supply–demand contracts and logistics-related decision-making. In this study, we consider the SBP with the MQC constraint in *N*-level non-exclusive AHCSs (shortly SBPM-NA). In this problem, batches are created for item types included in clusters according to the sequence of the levels in a given AHCS such that the MQC constraint as well as the maximum and minimum batch size requirements are satisfied simultaneously. Item types and their quantities to be batched are given at the first level of the AHCS, and those not batched at one level of the AHCS are collected and batched again at the next level. Note that batching at each level is not independent of batching at other levels, although the batching process is carried out sequentially from the first level until to the last level, because the results of batching at one level affect batch results at subsequent levels and batch results at a level also are influenced by batch results at previous levels. It costs more to process batches made at higher levels of the AHCS, because items with more heterogeneous item types are batched as the level of the AHCS increases. The MQC constraint ensures that more items than a committed minimum quantity must be batched at a level before items not batched at the level are sent to the next level. We apply the MQC constraint to control effectively the degree of heterogeneity (DoH) in the batching results. We developed a sequential batching algorithm for minimizing the total processing cost of items using properties identified to find better solutions of large-sized practical problems.

We explain the SBPM-NA using an example truck-loading problem in section 2. Also, we identify some solution properties for the SBPM-NA and describe a sequential batching algorithm. Then, results of computational experiments and concluding remarks are given.

2. Problem Description

Fig. 1 shows a four level non-exclusive AHCS for an example truck-loading problem modeled as the SBPM-NA. In the truck-loading example, item types, their quantities (to be batched), batches of items, batch processing costs, and unit processing costs correspond to destinations, quantities to be transported to destinations, truckloads, and their freight charges, respectively. A cluster consists of destinations for trucks to trip. Assume that each item has equal weight and volume and that items are transported to nine destinations using the same type of trucks. Also, assume that the quantities of items to transport for each of the nine destinations are known. There exists a requirement on the minimum quantity of items that must be loaded (i.e., the minimum batch size requirement, shortly MIBR) and a limit on the maximum quantity of items that can be loaded (i.e., the maximum batch size limit, shortly MABL) onto each truck. That is, each truck has a minimum load requirement and a maximum load capacity. In addition, if a truck trips multiple destinations, the route and schedule for destinations are fixed and known.
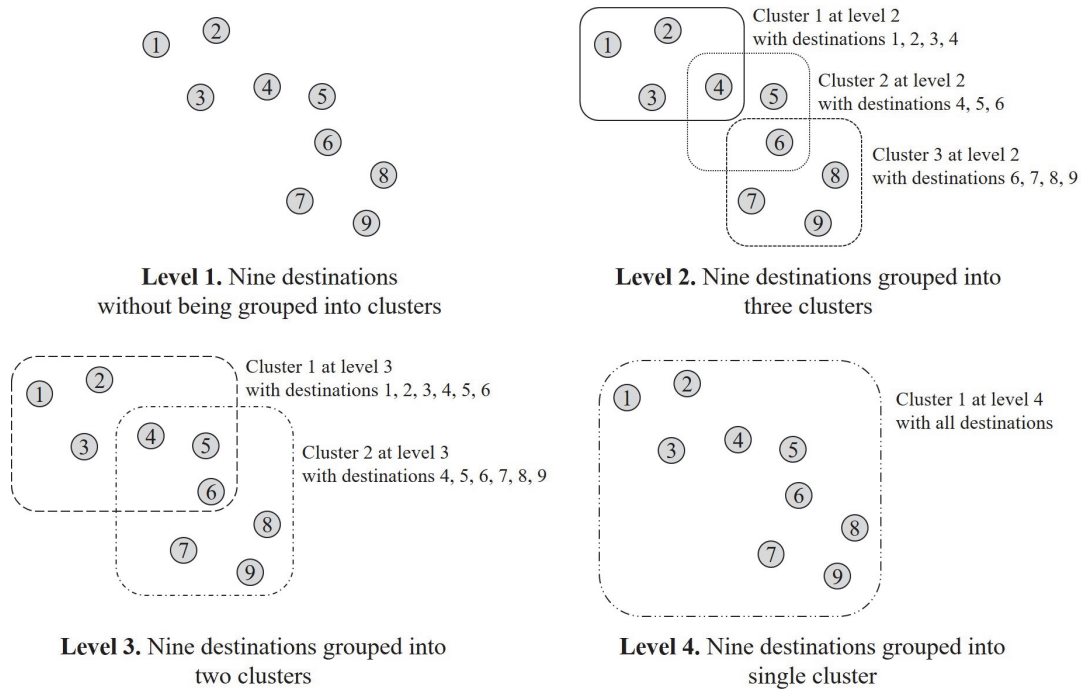
Level 1. Nine destinations without being grouped into clusters

Level 2. Nine destinations grouped into three clusters

Level 3. Nine destinations grouped into two clusters

Level 4. Nine destinations grouped into single cluster

Fig. 1. 4-level Non-exclusive AHCS for an example truck loading problem

Level 1 in Fig. 1 shows the case that all destinations are not grouped into any clusters. In this case, batches are made by each item type (i.e., each destination) and, therefore, items loaded onto the same truck at Level 1 have the same destination. Level 2 has three clusters, Clusters 1–3, with nine destinations divided into these three clusters. Note that Destination 4 (Destination 6) is included in both Clusters 1 and 2 (Clusters 2 and 3). Items loaded onto each truck at Level 2 will be sent to destinations included in the same cluster. Likewise, Level 3 has two clusters, and items loaded onto each truck at this level will be transported to destinations included in the same cluster of this level. Finally, all destinations are grouped into a single cluster at Level 4, and items loaded onto each truck at this level have them as their destinations. As the level increases, more destinations are included in a cluster. Furthermore, note that Destination 4 (Destination 6) is included in both Clusters 1 and 2 (Clusters 2 and 3) at Level 2, whereas Cluster 2 at Level 2 is included in both Clusters 1 and 2 at Level 3, resulting in a non-exclusive AHCS. In this example, batching items corresponds to making truckloads such that two loading constraints, the minimum requirement and the maximum limit, are satisfied. Also, the total number of batches corresponds to the total number of trucks used to transport items to their destinations. Although the total transportation cost becomes the minimum when each of the truckloads has only one destination, items for different destinations may be loaded onto some trucks to satisfy the two loading constraints. When we make a truckload with several destinations, the truckload should have destinations included in the same cluster. Note that transportation costs for truckloads at higher levels are higher than those at lower levels, because the truckloads at higher levels have more destinations to visit. Therefore, the truck-loading decision should be made through Level 1 to Level 4 sequentially for minimizing the total transportation cost. First, items are batched to make truckloads at Level 1, and then items not batched at this level are collected by the cluster at Level 2 and batched again to make truckloads for Clusters 1–3 at Level 2. Similarly, items not batched at Level 2 are collected by the cluster at Level 3 and batched again. This batching process continues until Level 3 and all remnants not batched until Level 3 are sent to Level 4, where all remnants are collected and processed individually. The MQC constraint ensures that items more than a committed minimum quantity must be batched and be made as truckloads at a level before items not batched at the level are sent to the next level. Table 1 gives a problem data of the truck-loading example. More than 15 items must be loaded for the truckload (TL) freight charge, and up to 20 items can be loaded on a truck. Freight charges for truckloads made at Levels 1, 2,

and 3 are 100, 120 (or 130), and 150, respectively. Items not made as truckloads until Level 3 incur Less-than-truck-load (LTL) freight charge (12 per item). Also, the TL and LTL freight charges become the batch processing and unit processing costs, respectively. Items are loaded onto several trucks from Level 1 to Level 4 sequentially, such that the truckloads satisfy the minimum truckload requirement and the maximum truck capacity.

Table 1
Data for the example 4-level truck loading problem

Delivery destination	Quantities to transport	Minimum requirement for truck-load	Maximum capacity of truck	Freight charge for truck-load			Freight charge for less-than-truck load
				Level 1	Level 2	Level 3	
1	29						
2	57				Cluster 1　130		
3	43						
4	83						12 (per unit)
5	13	16	20	100	Cluster 2　120	150	
6	10						
7	29						
8	71				Cluster 3　130		
9	8						

Table 2
A truck-loading decision with 18 trucks for the example problem

	Level 1			Level 2			Level 3			Level 4	
Destin-ation	TL-ID (# of items loaded)	Not loaded	Cluster [destin-ations]	TL-ID (# of items loaded)	Not loaded	Cluster [destin-ations]	TL-ID (# of items loaded)	Not loaded	Cluster [destin-ations]	LTL-ID (# of items loaded)
1	TL-1 (20)	9									
2	TL-2 (19) TL-3 (19) TL-4 (19)	0	1	TL-16 (17)		1					
3	**TL-5 (19)** **TL-6 (19)**	5	[1, 2, 3, 4]	(9, 0, 5, 3)	0	[1, 2, 3, 4, 5, 6]	-	-		
4	TL-7 (20) TL-8 (20) TL-9 (20) TL-10 (20)	3	2	TL-17 (20)					1	
5	-	13	[4, 5, 6]	(0, 13, 7)	0				[1, 2, 3, 4, 5, 6, 7, 8, 9]	-
6	-	10				2				
7	TL-11 (20)	9	3	TL-18 (20)						
8	**TL-12 (18)** **TL-13 (18)** **TL-14 (18)** **TL-15 (17)**	0	[6, 7, 8, 9]	(3, 9, 0, 8)	0	[4, 5, 6, 7, 8, 9]	-	-		
9	-	8								

Table 2 and Fig. 2 show a truck-loading decision by the sequential batching process and its graphical representation. Fifteen truckloads (from TL-1 to TL-15) are formed at Level 1, while each truckload contains only items for one destination. We call them homogeneous truckloads. For items not loaded at Level 1, three truckloads (TL-16, TL-17, and TL-18) are created for three clusters at Level 2. The truckload TL-16 passes through destinations 4, 3, and 1, whereas TL-17 (TL-18) contains items for destinations 6 and 5 (7, 9, and 6). We call them heterogeneous truckloads. Items for destination 6 are divided and loaded onto two truckloads, TL-17 and TL-18. The sequential batching process for all items is completed at Level 2 without any items charged by LTL freight rate. This is because the AHCS for the example problem is non-exclusive, where destination 6 is included in both Clusters 2 and 3 at Level 2. If the AHCS is exclusive and nine destinations are divided into three clusters, such as [1, 2, 3], [4, 5, 6],

and [7, 8, 9], the batching process cannot be completed at Level 2. Instead, some of the items will be sent to Level 3 or Level 4, increasing the possibility that items for several destinations are loaded onto the same truck. Table 3 and Fig. 3 show another truck-loading decision and its graphical representation for the example problem. Here, we create truckloads by filling items up to the maximum truck capacity. We can observe an important difference between the item quantities of TL-5 and TL-6 for destination 3 of the first truck-loading decision in Table 2 and those of the second decision in Table 3. In the first decision, the number of items loaded in each truck is determined as the value between the minimum requirement for TL freight charge and the maximum capacity of the truck, instead of making full truckloads, as in the second decision. We can also find the same difference for destination 8. The total costs for the first and second decisions are 1,880 ($(100{\times}15){+}(120{\times}1){+}(130{\times}2)$) and 1,896 ($(100{\times}14){+}(120{\times}1){+}(130{\times}1){+}(150{\times}1){+}(12{\times}8)$), respectively.

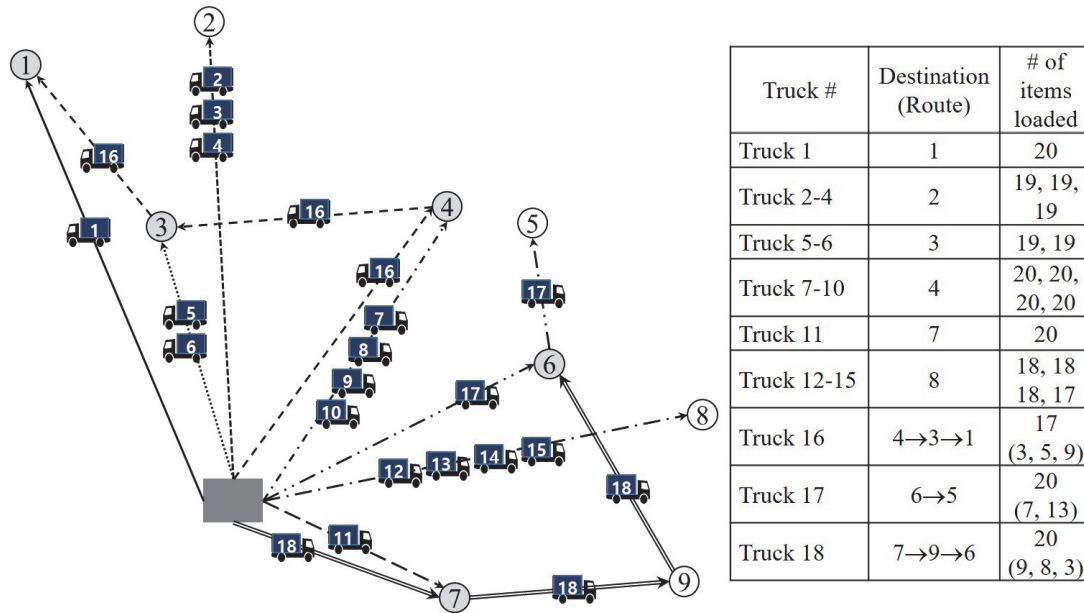| Truck #	Destination (Route)	# of items loaded
Truck 1	1	20
Truck 2-4	2	19, 19, 19
Truck 5-6	3	19, 19
Truck 7-10	4	20, 20, 20, 20
Truck 11	7	20
Truck 12-15	8	18, 18, 18, 17
Truck 16	4→3→1	17 (3, 5, 9)
Truck 17	6→5	20 (7, 13)
Truck 18	7→9→6	20 (9, 8, 3)

Fig. 2. Graphical representation of sequential batching for results shown in Table 2

Multiple deliveries and receipts occur for five destinations (i.e., destinations 1, 3, 4, 6, and 7) in the first decision, as depicted in Fig. 2, whereas multiple receipts for more destinations (i.e., 1, 3, 4, 6, 7, and 8) occur in the second decision, as depicted in Fig. 3. Multiple deliveries from heterogeneous truckloads occur inevitably because there exist some remnants (not loaded at Level 1) for several destinations due to the minimum load requirement and the truck capacity. A similar situation to that of multiple deliveries also exists when forming input release lots in semiconductor wafer fabrication, wherein a cluster consists of product types. As mentioned earlier, it is necessary to make a certain number of heterogeneous wafer lots owing to the productivity issue and the lot size requirements. These heterogeneous wafer lots cause lot-splits during the wafer fabrication process. It is important to effectively control the DoH determined by heterogeneous truckloads or heterogeneous wafer lots that causes multiple deliveries or lot-splits, because they are directly related to the total processing cost of items. As shown from the two decisions for the truck-loading example, the number of destinations where multiple deliveries occur can be decreased when the number of items loaded in each truck is determined as the value between the minimum requirement and the maximum truck capacity, instead of filling items up to the maximum truck capacity. We use the MQC constraint for effective control over the DoH. We explain the MQC constraint in more detail in the following section. Also, the non-exclusiveness of the AHCS should be considered to complete the sequential batching process at a lower level.

Table 3
Another truck-loading decision with 18 trucks for the example problem

	Level 1			Level 2			Level 3			Level 4	
Destin-ation	TL-ID (# of items loaded)	Not loaded	Cluster [destin-ations]	TL-ID (# of items loaded)	Not loaded	Cluster [destin-ations]	TL-ID (# of items loaded)	Not loaded	Cluster [destin-ations]	LTL-ID (# of items loaded)
1	TL-1 (20)	9								
2	TL-2 (20) TL-3 (20) TL-4 (17)	0	1							
3	TL-5 (20) TL-6 (20)	3	[1, 2, 3, 4]	-	12 (9, 0, 3, 0)	1	TL-17	0		
						[1, 2, 3, 4, 5, 6]	(9, 0, 3, 3, 0, 3)		1	LTL-18
4	TL-7 (20) TL-8 (20) TL-9 (20) TL-10 (20)	3	2	TL-15	6 (3, 0, 3)				[1, 2, 3, 4, 5, 6, 7, 8, 9]	(0, 0, 0, 0, 0, 0, 0, 0, 8)
5	-	13	[4, 5, 6]	(0, 13, 7)						
6	-	10								
7	TL-11 (20)	9	3	TL-16	8 (0, 0, 0, 8)	2	-	8 (0, 0, 0, 0, 0, 8)		
8	TL-12 (20) TL-13 (20) TL-14 (20)	11	[6, 7, 8, 9]	(0, 9, 11, 0)		[4, 5, 6, 7, 8, 9]				
9	-	8								

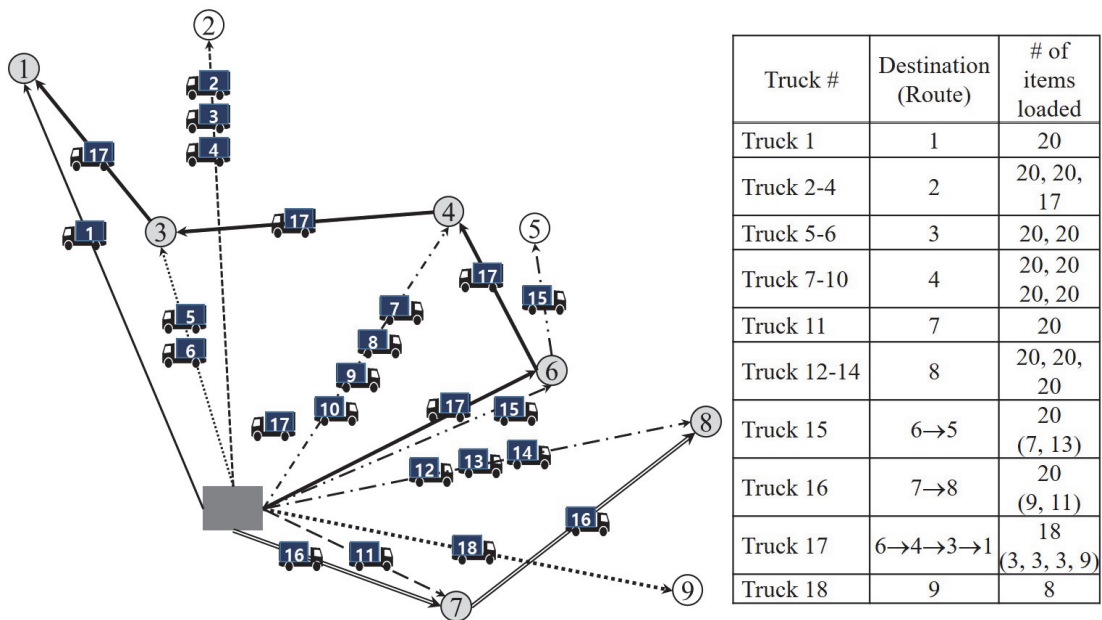| Truck #	Destination (Route)	# of items loaded
Truck 1	1	20
Truck 2-4	2	20, 20, 17
Truck 5-6	3	20, 20
Truck 7-10	4	20, 20, 20, 20
Truck 11	7	20
Truck 12-14	8	20, 20, 20
Truck 15	6→5	20 (7, 13)
Truck 16	7→8	20 (9, 11)
Truck 17	6→4→3→1	18 (3, 3, 3, 9)
Truck 18	9	8

Fig. 3. Graphical representation of sequential batching for results shown in Table 3

Notations to describe mathematically the SBPM-NA and the sequential batching algorithm are as follows.

Parameters

n index of levels ($n = 1, 2, …, N$)

$i^{(n)}$ index of item types (for $n=1$) or clusters (for $n \geq 2$) at level n (Note that batches are made by each item type at Level 1. Also, note that $i=1$ at Level 1, that is $1^{(1)}$, is different from $i=1$ at Level 2, that is $1^{(2)}$, although we use the same index i for any level n for notational simplicity.)

$\Lambda^{(n)}$ set of all item types (for $n=1$) or all clusters (for $n \geq 2$) at level n

$\Delta_{i^{(n)}}$ set of clusters at level $n+1$ that includes an item type $i^{(n)}$ (for $n=1$) or a cluster $i^{(n)}$ (for $n\geq2$) at level n

$\Upsilon_{i^{(n)}}$ set of item types (for $n=2$) or clusters (for $n\geq3$) at level $n-1$ that are included in $i^{(n)}$ at level n

$H_{i^{(n)}}$ the MABL of $i^{(n)}$

$K_{i^{(n)}}$ the MIBR of $i^{(n)}$ (Note that $K_{i^{(N)}} = 0$ for all $i^{(N)}$, that is, the MIBR is not applied to level N. Instead, all remained items not batched until level $N-1$ are processed individually at the last level.)

$C_{i^{(n)}}^{B}$ processing cost of items batched for $i^{(n)}$ (We assume that $C_{i^{(n)}}^{B} < C_{i^{(n+1)}}^{B}$ for all $n\leq N-1$ and $i^{(n)}$. That is, it costs more to process batches made at higher levels of the AHCS.)

$C_{i^{(N)}}^{U}$ processing cost of items at the last level N (We assume that $(H_{i^{(n)}} \cdot C_{i^{(N)}}^{U}) > C_{i^{(n)}}^{B}$ for all $n\leq N-1$ and $i^{(n)}$, that is $\left(C_{i^{(N)}}^{U} > (C_{i^{(n)}}^{B}/H_{i^{(n)}})\right)$. That is, it costs less to process items batched than not batched ones.)

Decision Variables

$X_{i^{(n)}}$ quantity of items to be batched (shortly QTB) for $i^{(n)}$ (Here, note that $X_{i^{(1)}}$ is the QTB and it is given in advance for all $i^{(1)}$. Also, note that $X_{i^{(N)}}$ is the quantity of items not batched (shortly QNB) until level $N-1$ and they are charged by unit processing cost $(C_{i^{(N)}}^{U})$ at the last level N.)

$Y_{i^{(n)}}$ quantity of items batched (shortly QB) for $i^{(n)}$ (Here, $1\leq n\leq N-1$.)

$Z_{i^{(n)}}$ the minimum number of batches (shortly MNB) needed to make batches with $Y_{i^{(n)}}$ items satisfying the MQC constraints as well as the MIBR ($K_{i^{(n)}}$) and MABL ($H_{i^{(n)}}$) on the batch size for $i^{(n)}$ (Here, $1\leq n\leq N-1$.)

$R_{i^{(n)}}^{i^{(n+1)}}$ quantity of items not batched for $i^{(n)}$ at level n and sent to the next level $n+1$ to be batched for $i^{(n+1)}$ (Here, $1\leq n\leq N-1$.)

Note that $X_{i^{(n)}} = \sum_{\forall\, i^{(n-1)}\in\Upsilon_{i^{(n)}}} R_{i^{(n-1)}}^{i^{(n)}}$ (for all $n\geq2$ and $i^{(n)}$) and $X_{i^{(n)}} - Y_{i^{(n)}} = \sum_{\forall\, i^{(n+1)}\in\Delta_{i^{(n)}}} R_{i^{(n)}}^{i^{(n+1)}}$ (for all $n\leq N-1$ and $i^{(n)}$). The total processing cost of items is $\sum_{1\leq n\leq N-1}\sum_{\forall\, i^{(n)}} C_{i^{(n)}}^{B} Z_{i^{(n)}} + \sum_{\forall\, i^{(N)}} C_{i^{(N)}}^{U} X_{i^{(N)}}$, where the first term is the processing cost for items batched whereas the second term is the processing cost for items not batched until level $N-1$.

3. Sequential Batching Algorithm for SBPM-NA

Let $\breve{B}_{i^{(n)}}$ be the MNB needed to make batches with $X_{i^{(n)}}$ items satisfying the MIBR ($K_{i^{(n)}}$) and MABL ($H_{i^{(n)}}$). Also, let $\breve{R}_{i^{(n)}}$ and $\hat{R}_{i^{(n)}}$ be the minimum and maximum numbers of remnants, respectively, for $i^{(n)}$ when making $\breve{B}_{i^{(n)}}$ batches with $X_{i^{(n)}}$ items to satisfy the MIBR and MABL. The following Property 1 to determine $\breve{B}_{i^{(n)}}$, $\breve{R}_{i^{(n)}}$ and $\hat{R}_{i^{(n)}}$ for given $X_{i^{(n)}}$ is derived from the solution property identified by Lim *et al.* (2015) for the SBP without the MQC constraint under three level exclusive AHCS.

Property 1. For a given $i^{(n)}$, $\breve{B}_{i^{(n)}}$, $\breve{R}_{i^{(n)}}$ and $\hat{R}_{i^{(n)}}$ are as follows:

(a) When $X_{i^{(n)}}/H_{i^{(n)}}$ is an integer, $\breve{B}_{i^{(n)}} = (X_{i^{(n)}}/H_{i^{(n)}})$, $\breve{R}_{i^{(n)}} = 0$ and $\hat{R}_{i^{(n)}} = (X_{i^{(n)}} - (X_{i^{(n)}}/H_{i^{(n)}}) \cdot K_{i^{(n)}})$

(b) When $X_{i^{(n)}}/H_{i^{(n)}}$ is not an integer and $(X_{i^{(n)}}/\lceil X_{i^{(n)}}/H_{i^{(n)}}\rceil) \geq K_{i^{(n)}}$, $\breve{B}_{i^{(n)}} = \lceil X_{i^{(n)}}/H_{i^{(n)}}\rceil$, $\breve{R}_{i^{(n)}} = 0$ and $\hat{R}_{i^{(n)}} = (X_{i^{(n)}} - (\lceil X_{i^{(n)}}/H_{i^{(n)}}\rceil \cdot K_{i^{(n)}}))$

(c) When $X_{i^{(n)}}/H_{i^{(n)}}$ is not an integer and $(X_{i^{(n)}}/\lceil X_{i^{(n)}}/H_{i^{(n)}}\rceil) < K_{i^{(n)}}$, $\breve{B}_{i^{(n)}} = \lfloor X_{i^{(n)}}/H_{i^{(n)}}\rfloor$, $\breve{R}_{i^{(n)}} = (X_{i^{(n)}} - (\lfloor X_{i^{(n)}}/H_{i^{(n)}}\rfloor \cdot H_{i^{(n)}}))$ and $\hat{R}_{i^{(n)}} = (X_{i^{(n)}} - (\lfloor X_{i^{(n)}}/H_{i^{(n)}}\rfloor \cdot K_{i^{(n)}}))$

Proof. Let β be the number of items included in each batch. When $X_{i(n)}/H_{i(n)}$ is an integer, $\check{B}_{i(n)} = (X_{i(n)}/H_{i(n)})$ because we can make $(X_{i(n)}/H_{i(n)})$ batches if each batch contains $H_{i(n)}$ items, that is, $β = H_{i(n)}$. Then, $\check{R}_{i(n)} = 0$ since there is no remnant. Also, $\hat{R}_{i(n)} = (X_{i(n)} - (X_{i(n)}/H_{i(n)}) \cdot K_{i(n)})$ if each batch contains $K_{i(n)}$ items, that is $β = K_{i(n)}$. If $X_{i(n)}/H_{i(n)}$ is not an integer and $(X_{i(n)}/\lceil X_{i(n)}/H_{i(n)} \rceil) \geq K_{i(n)}$, $\check{B}_{i(n)} = \lceil X_{i(n)}/H_{i(n)} \rceil$ because we can make $\lceil X_{i(n)}/H_{i(n)} \rceil$ batches if $K_{i(n)} \leq β < H_{i(n)}$. Then, $\check{R}_{i(n)} = 0$. Also, $\hat{R}_{i(n)} = (X_{i(n)} - (\lceil X_{i(n)}/H_{i(n)} \rceil \cdot K_{i(n)}))$ if each batch contains $K_{i(n)}$ items, that is, $β = K_{i(n)}$. When $X_{i(n)}/H_{i(n)}$ is not an integer and $(X_{i(n)}/\lceil X_{i(n)}/H_{i(n)} \rceil) < K_{i(n)}$, $\check{B}_{i(n)} = \lfloor X_{i(n)}/H_{i(n)} \rfloor$ because we can make $\lfloor X_{i(n)}/H_{i(n)} \rfloor$ batches if each batch contains $H_{i(n)}$ items, that is, $β = H_{i(n)}$. Then, $\check{R}_{i(n)} = X_{i(n)} - (\lfloor X_{i(n)}/H_{i(n)} \rfloor \cdot H_{i(n)})$. Also, $\hat{R}_{i(n)} = (X_{i(n)} - (\lfloor X_{i(n)}/H_{i(n)} \rfloor \cdot K_{i(n)}))$ if each batch contains $K_{i(n)}$ items, that is, $β = K_{i(n)}$. ■

In this study, the committed minimum quantity and the MQC constraint are set to $\check{B}_{i(n)} \cdot K_{i(n)}$ and $Y_{i(n)} \geq \check{B}_{i(n)} \cdot K_{i(n)}$, for each of all $i^{(n)}$s, respectively. By setting it like this, the number of items included in each batch for $i^{(n)}$ is determined as a value between $K_{i(n)}$ and $H_{i(n)}$ while minimizing the number of batches needed for $X_{i(n)}$ items, as shown in Property 1. Also $Y_{i(n)}$ satisfies the MQC constraint if $Y_{i(n)}$ is determined as a value between $(X_{i(n)} - \hat{R}_{i(n)})$ and $(X_{i(n)} - \check{R}_{i(n)})$ as shown by Property 2.

Property 2. If $Y_{i(n)}$ is determined as a value between $(X_{i(n)} - \hat{R}_{i(n)})$ and $(X_{i(n)} - \check{R}_{i(n)})$, $Y_{i(n)}$ satisfies the MQC constraint, $Y_{i(n)} \geq (\check{B}_{i(n)} \cdot K_{i(n)})$, and $Z_{i(n)} = \check{B}_{i(n)}$.

Proof. For (a) of Property 1, $X_{i(n)} - ((X_{i(n)} - (X_{i(n)}/H_{i(n)}) \cdot K_{i(n)})) \leq Y_{i(n)} \leq X_{i(n)}$ since $\hat{R}_{i(n)} = (X_{i(n)} - (X_{i(n)}/H_{i(n)}) \cdot K_{i(n)})$ and $\check{R}_{i(n)} = 0$. As a result, $X_{i(n)} - (X_{i(n)} - (\check{B}_{i(n)} \cdot K_{i(n)})) \leq Y_{i(n)} \leq X_{i(n)}$ since $\check{B}_{i(n)} = (X_{i(n)}/H_{i(n)})$ in this case. That is, $(\check{B}_{i(n)} \cdot K_{i(n)}) \leq Y_{i(n)} \leq X_{i(n)}$. Also, $Z_{i(n)} = \check{B}_{i(n)}$ according to Property 1. For (b) and (c) of Property 1, it can be proved similarly. ■

In the truck-loading example, the number of destinations where multiple deliveries occur can be decreased and the sequential batching process can be completed at an earlier level when the number of items loaded onto each truck is determined as a value between the minimum requirement for TL freight charge and the maximum truck capacity. Additionally, setting the MQC constraint $Y_{i(n)} \geq \check{B}_{i(n)} \cdot K_{i(n)}$ enables more effective control over the DoH in the batching results because we can change the completed level of the sequential batching process and the number of batches formed at each level by reducing or increasing the MIBR and MABL ($K_{i(n)}$ and $H_{i(n)}$), while minimizing the number of batches (i.e., $Z_{i(n)} = \check{B}_{i(n)}$). In the truck-lading example, the DoH can be defined by the ratio of destinations where multiple deliveries occur. As mentioned earlier, batches are created for item types included in clusters according to the sequence of the levels in a given AHCS. Therefore, the committed minimum quantity ($\check{B}_{i(n)} \cdot K_{i(n)}$) and batching results ($X_{i(n)}$, $Y_{i(n)}$, $Z_{i(n)}$ and $R_{i(n)}^{i^{(n+1)}}$) are not determined for all levels at once but is determined sequentially for only the level to which batching is being performed from the first level. Because of this distinct feature, it is called sequential batching with the MQC constraint. Batching at each level is not independent of batching at other levels, although the batching process is carried out sequentially from the first level until to the last level, because the results of batching at one level affect batch results at subsequent levels and batch results at a level also are influenced by batch results at previous levels. Recursive equations of the dynamic programming (DP) can be formulated for the sequential batching problem. However, the DP approach is not suitable to solve practical problems due to high computational complexity and very long computation time as mentioned by Bang et al. (2016) and Lim et al. (2017). In this study, we develop a sequential batching algorithm that makes batches for

two successive levels sequentially from the first level using properties identified to find better solutions quickly.

Table 4
A mathematical solution for the example truck-loading problem

Level (n)	$i^{(n)}$	Destination	$X_{i^{(n)}}$	$\breve{B}_{i^{(n)}}$	$\breve{R}_{i^{(n)}}$	$\hat{R}_{i^{(n)}}$	$Y_{i^{(n)}}$ (MQC)	$\Delta_{i^{(n)}}$	$R_{i^{(n)}}^{i^{(n+1)}}$
(1)	$1^{(1)}$	1	$X_{1^{(1)}} = 29$	1	9	13	$Y_{1^{(1)}} = 20$ $(Y_{1^{(1)}} \geq 16)$	$\Delta_{1^{(1)}} = \{1^{(2)}\}$	$R_{1^{(1)}}^{1^{(2)}} = 9$
	$2^{(1)}$	2	$X_{2^{(1)}} = 57$	3	0	9	$Y_{2^{(1)}} = 57$ $(Y_{2^{(1)}} \geq 48)$	$\Delta_{2^{(1)}} = \{1^{(2)}\}$	$R_{2^{(1)}}^{1^{(2)}} = 0$
	$3^{(1)}$	3	$X_{3^{(1)}} = 43$	2	3	11	$Y_{3^{(1)}} = 38$ $(Y_{3^{(1)}} \geq 32)$	$\Delta_{3^{(1)}} = \{1^{(2)}\}$	$R_{3^{(1)}}^{1^{(2)}} = 5$
	$4^{(1)}$	4	$X_{4^{(1)}} = 83$	4	3	19	$Y_{4^{(1)}} = 80$ $(Y_{4^{(1)}} \geq 64)$	$\Delta_{4^{(1)}} = \{1^{(2)}, 2^{(2)}\}$	$R_{4^{(1)}}^{1^{(2)}} = 3$ $R_{4^{(1)}}^{2^{(2)}} = 0$
	$5^{(1)}$	5	$X_{5^{(1)}} = 13$	0	13	13	$Y_{5^{(1)}} = 0$ $(Y_{5^{(1)}} \geq 0)$	$\Delta_{5^{(1)}} = \{2^{(2)}\}$	$R_{5^{(1)}}^{2^{(2)}} = 13$
	$6^{(1)}$	6	$X_{6^{(1)}} = 10$	0	10	10	$Y_{6^{(1)}} = 0$ $(Y_{6^{(1)}} \geq 0)$	$\Delta_{6^{(1)}} = \{2^{(2)}, 3^{(2)}\}$	$R_{6^{(1)}}^{2^{(2)}} = 7$ $R_{6^{(1)}}^{3^{(2)}} = 3$
	$7^{(1)}$	7	$X_{7^{(1)}} = 29$	1	9	13	$Y_{7^{(1)}} = 20$ $(Y_{7^{(1)}} \geq 16)$	$\Delta_{7^{(1)}} = \{3^{(2)}\}$	$R_{7^{(1)}}^{3^{(2)}} = 9$
	$8^{(1)}$	8	$X_{8^{(1)}} = 71$	4	0	7	$Y_{8^{(1)}} = 71$ $(Y_{8^{(1)}} \geq 64)$	$\Delta_{8^{(1)}} = \{3^{(2)}\}$	$R_{8^{(1)}}^{3^{(2)}} = 0$
	$9^{(1)}$	9	$X_{9^{(1)}} = 8$	0	8	8	$Y_{9^{(1)}} = 0$ $(Y_{9^{(1)}} \geq 0)$	$\Delta_{9^{(1)}} = \{3^{(2)}\}$	$R_{9^{(1)}}^{3^{(2)}} = 8$
(2)	$1^{(2)}$	1, 2, 3, 4	$X_{1^{(2)}} = R_{1^{(1)}}^{1^{(2)}} + R_{2^{(1)}}^{1^{(2)}} + R_{3^{(1)}}^{1^{(2)}}$ $+R_{4^{(1)}}^{1^{(2)}} = 9 + 0 + 5 + 3 = 17$	1	0	1	$Y_{1^{(2)}} = 17$ $(Y_{1^{(2)}} \geq 16)$	$\Delta_{1^{(2)}} = \{1^{(3)}\}$	$R_{1^{(2)}}^{1^{(3)}} = 0$
	$2^{(2)}$	4, 5, 6	$X_{2^{(2)}} = R_{4^{(1)}}^{2^{(2)}} + R_{5^{(1)}}^{2^{(2)}} +$ $R_{6^{(1)}}^{2^{(2)}} = 0 + 13 + 7 = 20$	1	0	4	$Y_{2^{(2)}} = 20$ $(Y_{2^{(2)}} \geq 16)$	$\Delta_{2^{(2)}} = \{1^{(3)}, 2^{(3)}\}$	$R_{2^{(2)}}^{1^{(3)}} = 0$ $R_{2^{(2)}}^{2^{(3)}} = 0$
	$3^{(2)}$	6, 7, 8, 9	$X_{3^{(2)}} = R_{6^{(1)}}^{3^{(2)}} + R_{7^{(1)}}^{3^{(2)}} + R_{8^{(1)}}^{3^{(2)}}$ $+R_{9^{(1)}}^{3^{(2)}} = 3 + 9 + 0 + 8 = 20$	1	0	4	$Y_{3^{(2)}} = 20$ $(Y_{3^{(2)}} \geq 16)$	$\Delta_{3^{(2)}} = \{2^{(3)}\}$	$R_{3^{(2)}}^{2^{(3)}} = 0$

According to Properties 1 and 2, we can obtain a mathematical solution of the example truck-loading problem. Table 4 presents the solution of the example truck-loading problem given in Table 2. Here, $N = 4$ and $X_{i^{(1)}}$ is the quantity to be transported to destination i. Additionally, $C_{i^{(1)}}^B = 100$ for all $i^{(1)}$, $C_{1^{(2)}}^B = C_{3^{(2)}}^B = 130$, $C_{2^{(2)}}^B = 120$, $C_{i^{(3)}}^B = 150$ for all $i^{(3)}$, $C_{i^{(4)}}^U = 12$ for all $i^{(4)}$, $K_{i^{(n)}} = 16$ for all n and $i^{(n)}$, and $H_{i^{(n)}} = 20$ for all n and $i^{(n)}$, as given in Table 1. The total processing cost is the sum of the total TL and LTL freight charges. Moreover, $\breve{B}_{i^{(n)}}$ is the number of truckloads for $i^{(n)}$, and $Y_{i^{(n)}}$ items are divided and loaded onto the $\breve{B}_{i^{(n)}}$ trucks. Following properties 3 and 4 indicate whether we can find an optimal solution of the SBPM-NA at Level 1 and Level 2, respectively.

Property 3. If $\breve{R}_{i^{(1)}} = 0$ for all $i^{(1)}$s, $Y_{i^{(1)}} = X_{i^{(1)}}$ and $Z_{i^{(1)}} = \breve{B}_{i^{(1)}}$ are optimal solutions of the SBPM-NA.

Proof. It is obvious since all items can be batched satisfying the MQC constraint at Level 1 without any remnants with the MNB. ■

Property 4. For N≥3, if there exist some $i^{(1)}$s with $\check{R}_{i^{(1)}} > 0$ but $\check{R}_{i^{(2)}} = 0$ for all $i^{(2)}$s, then $Y_{i^{(1)}} = X_{i^{(1)}} - \check{R}_{i^{(1)}}$, $Y_{i^{(2)}} = X_{i^{(2)}}$, $Z_{i^{(1)}} = \check{B}_{i^{(1)}}$ and $Z_{i^{(2)}} = \check{B}_{i^{(2)}}$ are optimal solutions of the SBPM-NA. Here, $X_{i^{(2)}} = \sum_{i^{(1)} \in Y_{i^{(2)}}} R_{i^{(1)}}^{i^{(2)}}$, where $R_{i^{(1)}}^{\tilde{l}^{(2)}} = \check{R}_{i^{(1)}}$ for $\tilde{l}^{(2)} = \underset{i^{(2)} \in \Delta_{i^{(1)}}}{\text{argmin}} \, C_{i^{(2)}}^B$, but $R_{i^{(1)}}^{i^{(2)}} = 0$ for all other $i^{(2)} \in \Delta_{i^{(1)}}$.

Proof. Note that $\tilde{l}^{(2)}$ is the cluster with the minimum batch processing cost among clusters in $\Delta_{i^{(1)}}$. The QB at Level 1 becomes the maximum and satisfies the MQC constraint by Property 2, while the number of batches for them is also the minimum by Property 1. Also, all remnants not batched at Level 1 are sent to the cluster of Level 2 with the minimum batch processing cost, and they are batched for the cluster without any remnants. Therefore, total batch processing cost becomes the minimum. ∎

Note that it is impossible to determine whether the solution in Table 4 is optimal based on Property 4 because remnants not batched at Level 1 are not sent to the cluster with the minimum batch processing cost at Level 2 (i.e., $R_{4^{(1)}}^{1^{(2)}} = 3$, $R_{4^{(1)}}^{2^{(2)}} = 0$, $R_{6^{(1)}}^{3^{(2)}} = 3$ and $R_{6^{(1)}}^{2^{(2)}} = 7$), even if the sequential batching process is completed at this level. The sequential batching process must be completed when $R_{4^{(1)}}^{1^{(2)}} = 0$, $R_{4^{(1)}}^{2^{(2)}} = 3$, $R_{6^{(1)}}^{2^{(2)}} = 10$ and $R_{6^{(1)}}^{3^{(2)}} = 0$ if the solution in Table 4 is to be an optimal solution according to Property 4, because cluster $2^{(2)}$ has the minimum batch processing cost among all clusters at Level 2.

The following Property 5 provides conditions for termination of the sequential batch process after Level 2. Note that Property 5 does not ensure an optimal solution of the SBPM-NA.

Property 5. For given $X_{i^{(n)}}$, $\hat{R}_{i^{(n)}}$ and $\check{R}_{i^{(n)}}$ at any level n where $2 \leq n \leq N-2$, the sequential batching process is completed at level $n+1$ if there exist $R_{i^{(n)}}^{i^{(n+1)}}$s and $X_{i^{(n+1)}}$s satisfying the following conditions.

$$\check{R}_{i^{(n)}} \leq \sum_{\forall i^{(n+1)} \in \Delta_{i^{(n)}}} R_{i^{(n)}}^{i^{(n+1)}} \leq \hat{R}_{i^{(n)}} \qquad \text{for all } i^{(n)} \tag{1}$$

$$\sum_{i^{(n)} \in Y_{i^{(n+1)}}} R_{i^{(n)}}^{i^{(n+1)}} = X_{i^{(n+1)}} \qquad \text{for all } i^{(n+1)} \tag{2}$$

$$\check{R}_{i^{(n+1)}} = 0 \qquad \text{for all } i^{(n+1)} \tag{3}$$

Proof. Since $\sum_{\forall i^{(n+1)} \in \Delta_{i^{(n)}}} R_{i^{(n)}}^{i^{(n+1)}} = X_{i^{(n)}} - Y_{i^{(n)}}$, $X_{i^{(n)}} - \hat{R}_{i^{(n)}} \leq Y_{i^{(n)}} \leq X_{i^{(n)}} - \check{R}_{i^{(n)}}$ from the condition (1). Therefore, $Y_{i^{(n)}}$ satisfies the MQC constraint by the Property 2. Also, the conditions (2) and (3) ensure that $Y_{i^{(n+1)}} = X_{i^{(n+1)}}$ and it also satisfies the MQC constraint. ∎

The following Property 6 shows how to obtain $Y_{i^{(N-1)}}$, $Z_{i^{(N-1)}}$ and $R_{i^{(N-1)}}^{i^{(N)}}$ that minimize the sum of batch processing costs at level $N-1$ and unit processing costs at the last level N.

Property 6. For given $X_{i^{(N-1)}}$ at level $N-1$, the following $Y_{i^{(N-1)}}$, $Z_{i^{(N-1)}}$ and $R_{i^{(N-1)}}^{i^{(N)}}$ minimize the sum of batch processing costs at level $N-1$ and unit processing costs at the last level, where $\tilde{l}^{(N)} = \underset{i^{(N)} \in \Delta_{i^{(N-1)}}}{\text{argmin}} \, C_{i^{(N)}}^U$.

   (a) $Y_{i^{(N-1)}} = X_{i^{(N-1)}} - \check{R}_{i^{(N-1)}}$ and $Z_{i^{(N-1)}} = \check{B}_{i^{(N-1)}}$

   (b) $R_{i^{(N-1)}}^{i^{(N)}} = \check{R}_{i^{(N-1)}}$ if $\left| \Delta_{i^{(N-1)}} \right| = 1$

   (c) $R_{i^{(N-1)}}^{\tilde{l}^{(N)}} = \check{R}_{i^{(N-1)}}$ and $R_{i^{(N-1)}}^{i^{(N)}} = 0$ for all other $i^{(N)} \in \Delta_{i^{(N-1)}}$, if $\left| \Delta_{i^{(N-1)}} \right| > 1$

Proof. $Y_{i^{(N-1)}}$ satisfies the MQC constraint by the Property 2 and it is the maximum QB with the MNB since $\check{R}_{i^{(N-1)}}$ is the minimum quantity of remnants after batching by Properties 1 and 2. Also, $\check{R}_{i^{(N-1)}}$ is sent to the cluster where the minimum unit processing cost occurs. ∎

The following Property 7 shows how to obtain a lower bound for optimal total processing cost of the SBPM-NA. Here, Here, $S_1^{(1)}$, $S_2^{(1)}$ and $S_3^{(1)}$ are:

$$S_1^{(1)} = \{i^{(1)}\,|\,X_{i^{(1)}}/H_{i^{(1)}} \text{ is an integer}\},$$
$$S_2^{(1)} = \{\,i^{(1)}\,|\,X_{i^{(1)}}/H_{i^{(1)}} \text{ is not integer and } (X_{i^{(1)}}/\lceil X_{i^{(1)}}/H_{i^{(1)}}\rceil) \geq K_{i^{(1)}}\} \text{ and}$$
$$S_3^{(1)} = \{\,i^{(1)}\,|\,X_{i^{(1)}}/H_{i^{(1)}} \text{ is not integer and } (X_{i^{(1)}}/\lceil X_{i^{(1)}}/H_{i^{(1)}}\rceil) < K_{i^{(1)}}\}.$$

Property 7. A lower bound for optimal total processing cost for the SBPM-NA, called LB, is

$$\sum_{\forall\, i^{(1)}\in S_1^{(1)}} C_{i^{(1)}}^B(X_{i^{(1)}}/H_{i^{(1)}}) + \sum_{\forall\, i^{(1)}\in S_2^{(1)}} C_{i^{(1)}}^B(\lceil X_{i^{(1)}}/H_{i^{(1)}}\rceil) + \sum_{\forall\, i^{(1)}\in S_3^{(1)}} C_{i^{(1)}}^B(\lfloor X_{i^{(1)}}/H_{i^{(1)}}\rfloor)$$

$$+C_{\tilde{l}^{(2)}}^B\left(\sum_{\forall\, i^{(1)}\in S_3^{(1)}} \check{R}_{i^{(1)}}\right)/H_{\hat{i}^{(2)}}, \text{ where } \tilde{l}^{(2)} = \operatorname*{argmin}_{i^{(2)}} C_{i^{(2)}}^B \text{ and } \hat{i}^{(2)} = \operatorname*{argmax}_{i^{(2)}} H_{i^{(2)}}.$$

Proof. By Property 1 through Property 3, the optimal total processing cost becomes $\sum_{\forall\, i^{(1)}\in S_1^{(1)}} C_{i^{(1)}}^B(X_{i^{(1)}}/H_{i^{(1)}}) + \sum_{\forall\, i^{(1)}\in S_2^{(1)}} C_{i^{(1)}}^B(\lceil X_{i^{(1)}}/H_{i^{(1)}}\rceil)$ when the sequential batching process is completed at Level 1. In this case, $S_3^{(1)} = \emptyset$ and LB equals to the optimal total processing cost. On the other hand, the optimal total processing cost becomes $\sum_{\forall\, i^{(1)}\in S_1^{(1)}} C_{i^{(1)}}^B(X_{i^{(1)}}/H_{i^{(1)}}) + \sum_{\forall\, i^{(1)}\in S_2^{(1)}} C_{i^{(1)}}^B(\lceil X_{i^{(1)}}/H_{i^{(1)}}\rceil) + \sum_{\forall\, i^{(1)}\in S_3^{(1)}} C_{i^{(1)}}^B(\lfloor X_{i^{(1)}}/H_{i^{(1)}}\rfloor) + \sum_{\forall\, i^{(2)}\in S_2^{(2)}} C_{i^{(2)}}^B(X_{i^{(2)}}^*/H_{i^{(2)}}) + \sum_{\forall\, i^{(2)}\in S_2^{(2)}} C_{i^{(2)}}^B(\lceil X_{i^{(2)}}^*/H_{i^{(2)}}\rceil)$ when the sequential batching process is completed at Level 2. Here, $X_{i^{(2)}}^*$ is an optimal solution at Level 2. In this case, $S_3^{(1)} \neq \emptyset$ but $S_3^{(2)} = \emptyset$. Since $\{\sum_{\forall\, i^{(2)}\in S_2^{(2)}} C_{i^{(2)}}^B(X_{i^{(2)}}^*/H_{i^{(2)}}) + \sum_{\forall\, i^{(2)}\in S_2^{(2)}} C_{i^{(2)}}^B(\lceil X_{i^{(2)}}^*/H_{i^{(2)}}\rceil)\} \geq \sum_{\forall\, i^{(2)}} C_{i^{(2)}}^B(X_{i^{(2)}}^*/H_{i^{(2)}}) \geq \left(\sum_{\forall\, i^{(2)}} C_{i^{(2)}}^B X_{i^{(2)}}^*\right)/H_{\hat{i}^{(2)}} \geq C_{\tilde{l}^{(2)}}^B\{\left(\sum_{\forall\, i^{(2)}} X_{i^{(2)}}^*\right)/H_{\hat{i}^{(2)}}\} \geq C_{\tilde{l}^{(2)}}^B\{\left(\sum_{\forall\, i^{(1)}\in S_3^{(1)}} \check{R}_{i^{(1)}}\right)/H_{\hat{i}^{(2)}}\}$ due to $\sum_{\forall\, i^{(2)}} X_{i^{(2)}}^* \geq \sum_{\forall\, i^{(1)}\in S_3^{(1)}} \check{R}_{i^{(1)}}$, $H_{\hat{i}^{(2)}} \geq H_{i^{(2)}}$ and $C_{\tilde{l}^{(2)}}^B \leq C_{i^{(2)}}^B$ for all $i^{(2)}$, $\sum_{\forall\, i^{(1)}\in S_1^{(1)}} C_{i^{(1)}}^B(X_{i^{(1)}}/H_{i^{(1)}}) + \sum_{\forall\, i^{(1)}\in S_2^{(1)}} C_{i^{(1)}}^B(\lceil X_{i^{(1)}}/H_{i^{(1)}}\rceil) + \sum_{\forall\, i^{(1)}\in S_3^{(1)}} C_{i^{(1)}}^B(\lfloor X_{i^{(1)}}/H_{i^{(1)}}\rfloor) + C_{\tilde{l}^{(2)}}^B\left(\sum_{\forall\, i^{(1)}\in S_3^{(1)}} \check{R}_{i^{(1)}}\right)/H_{\hat{i}^{(2)}}$ becomes a lower bound for the SBPM-NA. ∎

Using the properties identified above, we developed a sequential batching algorithm that makes batches for two successive levels from the first level. In this algorithm, for two successive levels n and $n+1$, we first determine both the QB for all $i^{(n)}$s at level n (i.e., $Y_{i^{(n)}}$) and the quantity of items not batched for $i^{(n)}$ at level n that are sent to the next level $n+1$ (i.e., $R_{i^{(n)}}^{i^{(n+1)}}$ and $X_{i^{(n+1)}}$). For all $i^{(n)}$s, we tentatively set $Y_{i^{(n)}}$ to $X_{i^{(n)}} - \check{R}_{i^{(n)}}$ (i.e., the maximum quantity of items that can be batched.). As a result, $\check{R}_{i^{(n)}}$ items can be sent to clusters included in $\Delta_{i^{(n)}}$ at level $n+1$. In this algorithm, we send all $\check{R}_{i^{(n)}}$ items to only one cluster at level $n+1$ where the minimum batch processing cost occurs. That means we tentatively set $R_{i^{(n)}}^{\tilde{l}^{(n+1)}} = \check{R}_{i^{(n)}}$ for all $i^{(n)}$s, where $\tilde{l}^{(n+1)} = \operatorname*{argmin}_{i^{(n+1)}\in\Delta_{i^{(n)}}} C_{i^{(n+1)}}^B$. Then, we can compute $X_{i^{(n+1)}}$ for all $i^{(n+1)}$s by $X_{i^{(n+1)}} = \sum_{\forall\, i^{(n)}\in Y_{i^{(n+1)}}} R_{i^{(n)}}^{i^{(n+1)}}$. If all $X_{i^{(n+1)}}$ items can be batched for all $i^{(n+1)}$s, the sequential batching process is completed at level $n+1$. However, if there exist $i^{(n+1)}$ such that $\check{R}_{i^{(n+1)}} > 0$, we check whether some items among tentatively batched at level n can be sent additionally so that $\check{R}_{i^{(n+1)}} = 0$ without violating the MQC constraint. If that is possible, we send additional items to the clusters so that $\check{R}_{i^{(n+1)}}$ becomes zero with the additional items from level n. Otherwise, we do not send the items batched at level n additionally to the next level and continue the sequential batching process. Now, we describe the sequential batching algorithm in detail.

Procedure 1 (*Sequential Batching Algorithm*)

Step 1. Compute $\check{R}_{i^{(1)}}$ and $\hat{R}_{i^{(1)}}$ for each of all $i^{(1)}$s according to Property 1. If $\check{R}_{i^{(1)}} = 0$ for all $i^{(1)}$s, make $\check{B}_{i^{(1)}}$ batches with $X_{i^{(1)}}$ for each of all $i^{(1)}$s according to Property 1 and terminate by Property 3.

Step 2. For each of all $i^{(1)}$s such that $\check{R}_{i^{(1)}} > 0$, set $R_{i^{(1)}}^{\tilde{l}^{(2)}} = \check{R}_{i^{(1)}}$ where $\tilde{l}^{(2)} = \underset{i^{(2)} \in \Delta_{i^{(1)}}}{\operatorname{argmin}} C_{i^{(2)}}^{B}$, and set

$R_{i^{(1)}}^{i^{(2)}} = 0$ for all other $i^{(2)}$s $\in \Delta_{i^{(1)}}$. Then, for each of all $i^{(2)}$s, compute $X_{i^{(2)}} = \sum_{i^{(1)} \in Y_{i^{(2)}}} R_{i^{(1)}}^{i^{(2)}}$, and compute $\check{R}_{i^{(2)}}$ and $\hat{R}_{i^{(2)}}$ according to Property 1. If $\check{R}_{i^{(2)}} = 0$ for all $i^{(2)}$s, set $Y_{i^{(1)}} = X_{i^{(1)}} - \check{R}_{i^{(1)}}$ for each of all $i^{(1)}$s and $Y_{i^{(2)}} = X_{i^{(2)}}$ for each of all $i^{(2)}$s. Make $\check{B}_{i^{(1)}}$ batches with $Y_{i^{(1)}}$ and $\check{B}_{i^{(2)}}$ batches with $Y_{i^{(2)}}$ for each of all $i^{(1)}$s and $i^{(2)}$s according to Property 2, and terminate by Property 4. If there exist some $i^{(2)}$s such that $\check{R}_{i^{(2)}} > 0$, set $n{=}1$ and go to Step 3.

Step 3. For each of all $i^{(n)}$s, compute $Y_{i^{(n)}} = X_{i^{(n)}} - \check{R}_{i^{(n)}}$ and make $\check{B}_{i^{(n)}}$ batches with $Y_{i^{(n)}}$ according to Property 2. Also, for each of all $i^{(n)}$s, set $R_{i^{(n)}}^{\tilde{l}^{(n+1)}} = \check{R}_{i^{(n)}}$ and $R_{i^{(n)}}^{i^{(n+1)}} = 0$ for all other $i^{(n+1)}$s $\in \Delta_{i^{(n)}}$, where $\tilde{l}^{(n+1)} = \underset{i^{(n+1)} \in \Delta_{i^{(n)}}}{\operatorname{argmin}} C_{i^{(n+1)}}^{B}$.

Step 4. For each of all $i^{(n+1)}$s, compute $X_{i^{(n+1)}} = \sum_{\forall i^{(n)} \in Y_{i^{(n+1)}}} R_{i^{(n)}}^{i^{(n+1)}}$ and set $\tilde{X}_{i^{(n+1)}} = 0$ if $X_{i^{(n+1)}}$ satisfies (a) or (b) of Property 1, otherwise, set $\tilde{X}_{i^{(n+1)}} = \left(\lceil X_{i^{(n+1)}}/H_{i^{(n+1)}} \rceil \cdot K_{i^{(n+1)}}\right) - X_{i^{(n+1)}}$. Solve the following transportation problem (TP) to check termination conditions of Property 5. In the objective function, M is a very large constant value.

(TP) Minimize $\sum_{\forall i^{(n)} \in Y_{i^{(n+1)}}} \sum_{\forall i^{(n+1)} \in \Delta_{i^{(n)}}} C_{i^{(n+1)}}^{B} U_{i^{(n)}}^{i^{(n+1)}} + \sum_{\forall i^{(n+1)}} M Q_{i^{(n+1)}}$ (4)

s.t. $\sum_{\forall i^{(n+1)} \in \Delta_{i^{(n)}}} U_{i^{(n)}}^{i^{(n+1)}} \leq \hat{R}_{i^{(n)}} - \check{R}_{i^{(n)}}$ for all $i^{(n)}$ (5)

$\sum_{\forall i^{(n)} \in Y_{i^{(n+1)}}} U_{i^{(n)}}^{i^{(n+1)}} + Q_{i^{(n+1)}} = \tilde{X}_{i^{(n+1)}}$ for all $i^{(n+1)}$ (6)

$U_{i^{(n)}}^{i^{(n+1)}} \geq 0$ and integer for all $i^{(n)}$ and $i^{(n+1)}$ (7)

$Q_{i^{(n+1)}} \geq 0$ and integer for all $i^{(n+1)}$ (8)

If $\sum_{\forall i^{(n+1)}} \tilde{Q}_{l^{(s+1)}} = 0$, compute $Y_{i^{(n)}} = Y_{i^{(n)}} - \sum_{\forall i^{(n+1)} \in \Delta_{i^{(n)}}} \tilde{U}_{i^{(n)}}^{i^{(n+1)}}$ for each of all $i^{(n)}$s and $R_{i^{(n)}}^{i^{(n+1)}} = R_{i^{(n)}}^{i^{(n+1)}} + \sum_{\forall i^{(n)} \in Y_{i^{(n+1)}}} \tilde{U}_{i^{(n)}}^{i^{(n+1)}}$ for all $i^{(n)}$s and $i^{(n+1)}$s. Then, for each of all $i^{(n+1)}$s, compute $X_{i^{(n+1)}} = \sum_{\forall i^{(n)} \in Y_{i^{(n+1)}}} R_{i^{(n)}}^{i^{(n+1)}}$. Here, $\tilde{Q}_{i^{(n+1)}}$ and $\tilde{U}_{i^{(n)}}^{i^{(n+1)}}$ are optimal solutions of (TP). Make $\check{B}_{i^{(n)}}$ and $\check{B}_{i^{(n+1)}}$ batches with changed $Y_{i^{(n)}}$ and $X_{i^{(n+1)}}$ for all $i^{(n)}$s and $i^{(n+1)}$s according to Properties 1 and 2, then terminate by Property 5. If $\sum_{i^{(n+1)}} \tilde{Q}_{i^{(n+1)}} > 0$, go to Step 5.

Step 5. For each of all $i^{(n)}$s and all $i^{(n+1)}$s $\in \Delta_{i^{(n)}}$, set $R_{i^{(n)}}^{i^{(n+1)}} = R_{i^{(n)}}^{i^{(n+1)}} + \tilde{U}_{i^{(n)}}^{i^{(n+1)}}$ if $Q_{i^{(n+1)}} = 0$. Here, $Q_{i^{(n+1)}}$ and $\tilde{U}_{i^{(n)}}^{i^{(n+1)}}$ are optimal solutions of (TP) solved in Step 4. Set $Y_{i^{(n)}} = Y_{i^{(n)}} - \sum_{\forall i^{(n+1)} \in \Delta_{i^{(n)}}} R_{i^{(n)}}^{i^{(n+1)}}$ for each of all $i^{(n)}$s. Make $\check{B}_{i^{(n)}}$ batches with changed $Y_{i^{(n)}}$ for each of all $i^{(n)}$s according to Property 2 and go to Step 6.

Step 6. If $n{=}N{-}2$, set $n{=}n{+}1$ and go to Step 7. If $n{<}N{-}2$, set $n{=}n{+}1$ and compute $X_{i^{(n)}} = \sum_{\forall i^{(n-1)} \in Y_{i^{(n)}}} R_{i^{(n-1)}}^{i^{(n)}}, \check{R}_{i^{(n)}}$ and $\hat{R}_{i^{(n)}}$ for each of all $i^{(n)}$s according to Property 1. If $\check{R}_{i^{(n)}} = 0$ for all $i^{(n)}$s, make $\check{B}_{i^{(n)}}$ batches with $X_{i^{(n)}}$ for each of all $i^{(n)}$s according to Property 1 and terminate. If there exist some $i^{(n)}$s such that $\check{R}_{i^{(n)}} > 0$, go to Step 3.

Step 7. For each of all $i^{(n)}$s, compute $X_{i^{(n)}} = \sum_{\forall i^{(n-1)} \in Y_{i^{(n)}}} R_{i^{(n-1)}}^{i^{(n)}}$ and $\check{R}_{i^{(n)}}$. Then, according to Property 6, set $Y_{i^{(n)}} = X_{i^{(n)}} - \check{R}_{i^{(n)}}$ and $R_{i^{(n)}}^{\tilde{l}^{(n+1)}} = \check{R}_{i^{(n)}}$ if $|\Delta_{i^{(n)}}| = 1$. Set $R_{i^{(n)}}^{\tilde{l}^{(n+1)}} = \check{R}_{i^{(n)}}$ and $R_{i^{(n)}}^{i^{(n+1)}} =$

0 for all other $i^{(n+1)} \in \Delta_{i^{(n)}}$ if $|\Delta_{i^{(n)}}| > 1$, where $\tilde{l}^{(n+1)} = \underset{i^{(n+1)} \in \Delta_{i^{(n)}}}{\operatorname{argmin}} C_{i^{(n+1)}}^U$. Make $\check{B}_{i^{(n)}}$ batches

with $Y_{i^{(n)}}$ according to Property 2 and set $X_{i^{(n+1)}} = \sum_{\forall i^{(n)} \in Y_{i^{(n+1)}}} R_{i^{(n)}}^{i^{(n+1)}}$. Compute the lower bound according to Property 7 and compute the total processing cost. Terminate.

4. Computational Experiments

We performed computational experiments for evaluating the sequential batching algorithm. Problem instances are generated as follows.

(1) Three cases for the number of item types: 400, 600 and 800.
(2) Three cases for the mean of the QTB for each item type (μ): 400, 600 and 800.
(3) Three cases for the variation of the QTB ($\omega=(\sigma \div \mu)$): Low ($\omega=1.0$), Medium ($\omega=1.5$) and High ($\omega=2.0$), where μ and σ are two parameters of the normal distribution $N(\mu, \sigma^2)$. Here, the size of variation becomes larger as ω increases. Also, we excluded negative random values generated from the normal distribution.
(4) Three cases for the number of levels (N): 3-Level, 4-Level and 5-Level.
(5) Three cases for the density of clusters (δ): High ($\delta=5$), Medium ($\delta=4$) and Low ($\delta=3$). Here, δ is the average number of item types (for $n=1$) or the average number of clusters (for $n \geq 2$) at level n, that is included in each cluster at level $n+1$. The larger δ, each cluster contains more item types.
(6) Three cases for the MABL ($\beta=H_{i^{(n)}}$): Small ($\beta=50$), Medium ($\beta=100$) and Large ($\beta=200$).
(7) Three cases for the MQC tightness ($\tau=K_{i^{(n)}}$): Tight ($\tau=\beta\times0.96$), Ordinary ($\tau=\beta\times0.90$) and Loose ($\tau=\beta\times0.80$). Here, remember that we set the committed minimum quantity to ($\check{B}_{i^{(n)}} \times K_{i^{(n)}}$), where $\check{B}_{i^{(n)}}$ is the MNB by Property 1 and $K_{i^{(n)}}$ is the MIBR. That is, the bigger τ, the larger $K_{i^{(n)}}$. Therefore, the MQC constraint becomes tighter.

The first three parameters (namely, the number of item types and the mean and variation of the QTB for each item type) define demand scenarios. We constructed 27 demand scenarios with the three parameters of three levels. The other four parameters define the clustering structure (the number of levels and density of clusters) and batch configuration (the MABL and MQC tightness). For each of the 27 demand scenarios, we tested 81 combinations for the four parameters of clustering structure and batch configuration. For each of the 81 combinations, we generated five problem instances and solved them using the sequential batching algorithm. Note that the number of item types, QTB, and batch and unit processing costs are the same for all the five problem instances of the same demand scenario, whereas the clustering structure and batch configuration are different. The batch processing costs were generated from DU(100, 120) for Level 1, DU(150, 200) for Level 2, DU(250, 300) for Level 3, and DU(350, 400) for Level 4 when the MABL is small. Here, DU(a, b) represents the discrete uniform distribution with range parameters a and b. When the MABLs are medium and large, the batch processing costs for them were set to doubled and quadrupled, respectively, of those for the small MABL. The unit processing costs at the last level were set to satisfy $C_{i^{(N)}}^U > (C_{i^{(n)}}^B / H_{i^{(n)}})$ for all $i^{(n)}$s. The sequential batching algorithm was coded in C programming language and run on a PC with an Intel Core i7 processor to solve the problem instances. The algorithm solved all the problem instances very quickly within seconds. The average percentage gap is 2.07%, which means that the algorithm finds very good lower and upper bounds. Here, the percentage gap is ((UB−LB)÷LB)×100, where UB is the total processing cost obtained by the algorithm and LB is the lower bound by Property 7. For each demand scenario, we performed analysis of variance (ANOVA) to know whether factors for the clustering structure and batch configuration have statistically significant effect on the total processing cost. Table 5 gives results of ANOVA. The MABL is a statistically significant factor that affects the total processing cost for all 27 demand scenarios. However, the number of levels is not statistically significant factor on the total processing cost for all demand scenarios. As shown in the results for demand scenarios S-1, S-2, S-3, S-

10, S-11, S-12, S-19, S-20, and S-21, the MQC tightness affects the total processing cost when the average demand is small, regardless of the number of item types and size of variation. The MQC tightness also affects the total processing cost for all demand scenarios except S-27, when the number of item types is large, as shown in the results for demand scenarios from S-19 through S-26. When the number of item types is small, the density of clusters affects the total processing cost when both the average demand and size of variation are small, as shown in the results for demand scenario S-1. As the number of item types increases, the density of clusters affects the total processing cost in cases where the average demand or size of variation increases, as shown in the results for demand scenarios S-11, S-13, S-20, and S-22. When both the average demand and size of variation are small, interactions among the density of clusters, MABL, and the MQC tightness affect the total processing cost, as shown in the results for demand scenarios S-1, S-10, and S-19. Also, when the number of item types increases, the interaction between the MABL and MQC tightness affects the total processing cost for larger average demand or larger variation, as shown in the results for demand scenarios S-13, S-20, S-22, S-23, and S-25.

Table 5
Results of ANOVA for N-level sequential batching

demand scenario				statistical significance			
				clustering structure		Batch configuration	
scenario ID	number of item types	mean of the QTB (μ)	variation of the QTB (ω)	number of levels (N)	density of cluster (δ)	MABL (β)	MQC tightness (τ)
S-1			Low		*	*	*
S-2		400	Medium			*	*
S-3			High			*	*
S-4	400	600	Low			*	*
S-5			Medium		-	*	-
S-6			High			*	
S-7		800	Low			*	*
S-8			Medium			*	-
S-9			High			*	*
S-10		400	Low		*	*	*
S-11			Medium		*	*	*
S-12			High		-	*	*
S-13	600	600	Low		*	*	*
S-14			Medium	-		*	*
S-15			High			*	-
S-16		800	Low		-	*	*
S-17			Medium			*	
S-18			High			*	-
S-19		400	Low		*	*	*
S-20			Medium		*	*	*
S-21			High		-	*	*
S-22	800	600	Low		*	*	*
S-23			Medium			*	*
S-24			High			*	*
S-25		800	Low		-	*	*
S-26			Medium			*	*
S-27			High			*	-

* It is a statistically significant factor at the significance level of 0.05. Gray cells represent there exist statistically significant interaction effects.

From the results of ANOVA, we now give some meaningful insights to help determine optimal parameters for the clustering structure and batch configuration from detailed results for some demand scenarios. Tables 6–8 show test results for demand scenario S-1, where the number of item types, average demand, and size of variation are small. Results for the 3-level case are shown in Table 6, whereas those for 4-level and 5-level cases are shown in Tables 7 and 8, respectively. All values in the fifth through the last column are the average values of test results for the five problem instances solved. In the tables, the first column gives the variation in the QTB. The QTB for each item type were generated from $N(400, 400^2)$ in demand scenario S-1.

Table 6
Results for demand scenario S-1 with 3-Level non-exclusive AHCS

ω	Clustering Structure (δ)	β	τ	Total Processing Cost	GAP (%)	Fill Rate	Batched	Not Batched	Batched at L1	Batched at L2	Completed Level
N(400, 400²), (ω =1.0)	3-400-80-16 (High)	50	48	471248.00	1.79	0.97	1.0000	0.0000	0.9728	0.0272	2.2
			45	474901.81	0.88	0.93	1.0000	0.0000	0.9874	0.0126	2
			40	475895.00	0.39	0.88	1.0000	0.0000	0.9949	0.0051	2
		100	96	485199.19	3.81	0.97	0.9995	0.0005	0.9256	0.0739	3
			90	490660.41	3.04	0.93	1.0000	0.0000	0.9578	0.0422	2
			80	496520.81	1.62	0.87	1.0000	0.0000	0.9802	0.0198	2
		200	192	514691.59	7.02	0.98	0.9924	0.0076	0.8394	0.1531	3
			180	517580.81	7.61	0.93	0.9998	0.0002	0.8751	0.1247	2.8
			160	531167.19	5.69	0.86	1.0000	0.0000	0.9310	0.0690	2
	3- 400-100-25 (Medium)	50	48	472973.81	2.17	0.97	1.0000	0.0000	0.9705	0.0295	2.8
			45	475194.59	0.95	0.93	1.0000	0.0000	0.9864	0.0136	2
			40	475877.81	0.39	0.88	1.0000	0.0000	0.9950	0.0050	2
		100	96	487284.00	4.27	0.97	0.9986	0.0014	0.9240	0.0746	3
			90	493931.19	3.74	0.93	0.9999	0.0001	0.9535	0.0464	2.4
			80	497250.81	1.77	0.87	1.0000	0.0000	0.9785	0.0215	2
		200	192	522097.19	8.61	0.99	0.9846	0.0154	0.8422	0.1424	3
			180	521969.59	8.56	0.93	0.9988	0.0012	0.8705	0.1282	3
			160	537558.00	6.98	0.85	0.9999	0.0001	0.9240	0.0759	2.4
	3-400-133-44 (Low)	50	48	475174.59	2.65	0.97	0.9991	0.0009	0.9690	0.0301	3
			45	475768.19	1.07	0.93	1.0000	0.0000	0.9853	0.0147	2.4
			40	475971.19	0.41	0.87	1.0000	0.0000	0.9947	0.0053	2
		100	96	491516.41	5.18	0.98	0.9932	0.0068	0.9281	0.0651	3
			90	497792.81	4.55	0.93	0.9989	0.0011	0.9500	0.0489	3
			80	498501.19	2.03	0.86	0.9999	0.0001	0.9771	0.0227	2.6
		200	192	542375.63	12.82	0.99	0.9685	0.0315	0.8446	0.1239	3
			180	528476.00	9.92	0.94	0.9927	0.0073	0.8719	0.1208	3
			160	543475.63	8.16	0.85	0.9980	0.0020	0.9186	0.0794	3

Table 7
Results for demand scenario S-1 with 4-Level non-exclusive AHCS

ω	Clustering Structure (δ)	β	τ	Total Processing Cost	GAP (%)	Fill Rate	Batched	Not Batched	Batched at L1	Batched at L2	Batched at L3	Completed Level
N(400, 400²), (ω =1.0)	4-400-80-16-3 (High)	50	48	471100.00	1.76	0.97	1.0000	0.0000	0.9729	0.0271	0.0000	3.2
			45	475163.00	0.94	0.93	1.0000	0.0000	0.9872	0.0128	0.0000	2
			40	475867.59	0.39	0.88	1.0000	0.0000	0.9949	0.0051	0.0000	2
		100	96	484948.41	3.75	0.97	0.9997	0.0003	0.9259	0.0738	0.0000	4
			90	490399.59	2.98	0.93	1.0000	0.0000	0.9583	0.0417	0.0000	2
			80	496634.00	1.64	0.87	1.0000	0.0000	0.9802	0.0198	0.0000	2
		200	192	512862.81	6.63	0.98	0.9959	0.0041	0.8380	0.1542	0.0036	4
			180	517064.81	7.51	0.93	0.9998	0.0002	0.8754	0.1244	0.0000	3.2
			160	530744.00	5.61	0.86	1.0000	0.0000	0.9322	0.0678	0.0000	2
	4-400-100-25-6 (Medium)	50	48	473074.00	2.19	0.97	0.9999	0.0001	0.9704	0.0296	0.0000	3.6
			45	475497.59	1.01	0.93	1.0000	0.0000	0.9861	0.0139	0.0000	2
			40	476010.00	0.42	0.87	1.0000	0.0000	0.9949	0.0051	0.0000	2
		100	96	486737.19	4.16	0.97	0.9986	0.0014	0.9249	0.0735	0.0002	4
			90	494350.41	3.83	0.93	0.9999	0.0001	0.9532	0.0467	0.0000	2.8
			80	496846.41	1.69	0.86	1.0000	0.0000	0.9782	0.0218	0.0000	2
		200	192	516407.59	7.42	0.99	0.9918	0.0082	0.8418	0.1417	0.0083	4
			180	521402.41	8.45	0.93	0.9985	0.0015	0.8718	0.1267	0.0000	4
			160	537000.00	6.87	0.85	0.9999	0.0001	0.9239	0.0761	0.0000	2.8
	4-400-133-44-15 (Low)	50	48	475146.00	2.64	0.97	0.9992	0.0008	0.9687	0.0305	0.0000	4
			45	475704.00	1.05	0.93	0.9999	0.0001	0.9853	0.0146	0.0000	2.8
			40	475975.59	0.41	0.88	1.0000	0.0000	0.9947	0.0053	0.0000	2
		100	96	490026.81	4.86	0.98	0.9950	0.0050	0.9275	0.0663	0.0012	4
			90	498044.41	4.60	0.93	0.9988	0.0012	0.9501	0.0487	0.0000	4
			80	498369.19	2.00	0.86	0.9999	0.0001	0.9770	0.0229	0.0000	3.2
		200	192	527537.19	9.74	0.99	0.9844	0.0156	0.8445	0.1244	0.0156	4
			180	528739.19	9.98	0.94	0.9927	0.0073	0.8725	0.1182	0.0020	4
			160	541727.63	7.81	0.86	0.9981	0.0019	0.9202	0.0779	0.0000	4

Table 8
Results for demand scenario S-1 with 5-Level non-exclusive AHCS

ω	Clustering Structure (δ)	β	τ	Total Processing Cost	GAP (%)	Fill Rate	Batched	Not Batched	Batched at L1	Batched at L2	Batched at L3	Batched at L4	Completed Level
N(400, 400²), (ω = 1.0)	5-400-80-16-3-1 (High)	50	48	471228.00	1.78	0.97	1.0000	0.0000	0.9728	0.0272	0.0000	0.0000	3.2
		50	45	474867.19	0.87	0.93	1.0000	0.0000	0.9873	0.0127	0.0000	0.0000	2
			40	476091.59	0.44	0.87	1.0000	0.0000	0.9949	0.0051	0.0000	0.0000	2
		100	96	485504.81	3.87	0.97	0.9998	0.0002	0.9248	0.0750	0.0000	0.0000	5
			90	490297.19	2.96	0.93	1.0000	0.0000	0.9589	0.0411	0.0000	0.0000	2
			80	496374.41	1.59	0.87	1.0000	0.0000	0.9802	0.0198	0.0000	0.0000	2
		200	192	510568.81	6.15	0.98	0.9986	0.0014	0.8394	0.1514	0.0055	0.0023	5
			180	518151.59	7.74	0.93	0.9999	0.0001	0.8746	0.1253	0.0000	0.0000	3.2
			160	531429.63	5.74	0.86	1.0000	0.0000	0.9311	0.0689	0.0000	0.0000	2
	5-400-100-25-6-1 (Medium)	50	48	472950.59	2.16	0.97	0.9999	0.0001	0.9705	0.0294	0.0000	0.0000	5
			45	475486.00	1.01	0.93	1.0000	0.0000	0.9860	0.0140	0.0000	0.0000	2
			40	476089.19	0.44	0.87	1.0000	0.0000	0.9948	0.0052	0.0000	0.0000	2
		100	96	486886.41	4.19	0.97	0.9989	0.0011	0.9243	0.0742	0.0002	0.0002	5
			90	494037.19	3.76	0.93	0.9999	0.0001	0.9530	0.0469	0.0000	0.0000	3.2
			80	497346.41	1.79	0.86	1.0000	0.0000	0.9785	0.0215	0.0000	0.0000	2
		200	192	513051.59	6.73	0.99	0.9982	0.0018	0.8415	0.1429	0.0073	0.0065	5
			180	522036.81	8.58	0.93	0.9991	0.0009	0.8705	0.1281	0.0003	0.0002	5
			160	536344.00	6.73	0.85	0.9999	0.0001	0.9240	0.0759	0.0000	0.0000	3.2
	5-400-133-44-15-5 (Low)	50	48	474688.00	2.54	0.98	0.9992	0.0008	0.9693	0.0297	0.0000	0.0002	5
			45	475762.19	1.07	0.93	1.0000	0.0000	0.9853	0.0147	0.0000	0.0000	3.2
			40	476071.41	0.43	0.88	1.0000	0.0000	0.9946	0.0054	0.0000	0.0000	2
		100	96	489819.19	4.81	0.99	0.9970	0.0030	0.9276	0.0651	0.0012	0.0031	5
			90	497610.41	4.51	0.93	0.9991	0.0009	0.9504	0.0484	0.0000	0.0002	5
			80	498166.00	1.96	0.86	0.9999	0.0001	0.9772	0.0228	0.0000	0.0000	3.8
		200	192	523082.00	8.81	0.99	0.9943	0.0057	0.8443	0.1234	0.0159	0.0106	5
			180	526698.38	9.55	0.95	0.9959	0.0041	0.8724	0.1194	0.0017	0.0025	5
			160	542236.81	7.91	0.85	0.9986	0.0014	0.9187	0.0797	0.0002	0.0000	5

The second column gives the clustering structure, where the first value represents the number of levels and the remaining values represent the numbers of item types at level 1, clusters at level 2, and clusters at level 3, respectively. For example, the clustering structure 3-400-80-16 has three levels, 400 item types in the first level, 80 clusters in the second level, and 16 clusters in the third level. We set the portion of clusters (or product types) at a level that are included in two or more clusters at the next level to be between 15% and 45% to reflect non-exclusiveness of the AHCS. The fifth and sixth columns give the total processing cost and the percentage gap, respectively. The seventh column gives the average fill rate computed by dividing the average number of items included per batch by the MABL. For example, 48.5 items are included per batch on average when the fill rate is 0.98 and the MABL is 50. The eighth, ninth, tenth and eleventh columns represent the portions of the total QB, total QNB until the last level, and the QB at Levels 1 and 2, respectively. The last column represents the average level where the sequential batching process is completed. Results of the computational experiments for demand scenario S-1 show that dense cluster, small MABL, and tight MQC constraint are effective in reducing the total processing cost. Note that we assume the batch processing cost increases proportional to the MABL. If economies of scale at the batch processing cost occur as the MABL grows, a larger MABL may reduce the total processing cost. Furthermore, the larger MABL reduces the number of batches and may result in reduced material handling costs. However, the heterogeneity of each cluster becomes higher, and item types included in batches may also increase when we decrease the number of clusters to make the density of clusters denser. This may increase the batch processing cost and total processing cost. Test results for 4-level and 5-level cases compared with results for the 3-level case present some new findings. The small MABL and tight MQC constraint help reduce the total processing cost further when we increase the number of levels from three to four or five if the variation in the QTB is not large, although the number of levels is not statistically significant factor on the total processing cost. Also, the QB increases slightly when we increase the number of levels. Tables 9–11 show test results for demand scenario S-14, where the number of item types, average demand, and size of variation are all medium levels. Tables 9, 10 and 11 show results for the 3-level, 4-level and 5-level cases, respectively. For this demand scenario, dense

cluster, small MABL, and tight MQC constraint are still effective to reduce the total processing cost, although the density of cluster is not statistically significant. However, increasing the number of levels does not help reduce the total processing cost any more in this demand scenario unlike in the S-1 demand scenario where the number of item types, average demand, and size of variation are small.

Table 9
Results for demand scenario S-14 with 3-Level non-exclusive AHCS

ω	Clustering Structure (δ)	β	τ	Total Processing Cost	GAP (%)	Fill Rate	Batched	Not Batched	Batched at L1	Batched at L2	Completed Level
N(600, 900²), (ω =1.5)	3-600-120-24 (High)	50	48	1339589.75	0.61	0.97	1.0000	0.0000	0.9904	0.0096	2.4
			45	1342580.00	0.29	0.94	1.0000	0.0000	0.9954	0.0046	2
			40	1343333.25	0.14	0.88	1.0000	0.0000	0.9982	0.0018	2
		100	96	1361730.75	1.87	0.97	1.0000	0.0000	0.9709	0.0291	2.2
			90	1372576.00	1.04	0.93	1.0000	0.0000	0.9850	0.0150	2.2
			80	1376889.25	0.60	0.88	1.0000	0.0000	0.9925	0.0075	2
		200	192	1404139.63	4.12	0.97	0.9994	0.0006	0.9226	0.0768	3
			180	1420828.38	3.34	0.93	0.9999	0.0001	0.9539	0.0460	2.6
			160	1438384.75	2.01	0.87	1.0000	0.0000	0.9757	0.0243	2
	3-600-150-38 (Medium)	50	48	1340789.25	0.70	0.97	1.0000	0.0000	0.9896	0.0104	2.6
			45	1343157.38	0.34	0.94	1.0000	0.0000	0.9952	0.0048	2
			40	1343440.63	0.15	0.88	1.0000	0.0000	0.9982	0.0018	2
		100	96	1365279.25	2.14	0.97	0.9999	0.0001	0.9693	0.0306	3
			90	1373565.25	1.12	0.93	1.0000	0.0000	0.9838	0.0162	2.2
			80	1377421.25	0.64	0.87	1.0000	0.0000	0.9922	0.0078	2
		200	192	1410282.75	4.58	0.97	0.9978	0.0022	0.9223	0.0755	3
			180	1426412.38	3.74	0.93	0.9998	0.0002	0.9507	0.0492	3
			160	1442067.25	2.27	0.87	1.0000	0.0000	0.9725	0.0275	2
	3-600-200-66 (Low)	50	48	1342841.25	0.86	0.97	0.9998	0.0002	0.9885	0.0113	3
			45	1343426.25	0.36	0.94	1.0000	0.0000	0.9948	0.0052	2.2
			40	1343480.25	0.15	0.88	1.0000	0.0000	0.9981	0.0019	2.2
		100	96	1372966.38	2.71	0.97	0.9987	0.0013	0.9681	0.0305	3
			90	1377372.00	1.40	0.93	0.9998	0.0002	0.9820	0.0178	3
			80	1378407.25	0.71	0.87	1.0000	0.0000	0.9918	0.0082	2.2
		200	192	1425495.63	5.71	0.98	0.9916	0.0084	0.9266	0.0650	3
			180	1440699.63	4.78	0.93	0.9984	0.0016	0.9466	0.0518	3
			160	1448087.25	2.70	0.86	0.9999	0.0001	0.9705	0.0293	3

Table 10
Results for demand scenario S-14 with 4-Level non-exclusive AHCS

ω	Clustering Structure (δ)	β	τ	Total Processing Cost	GAP (%)	Fill Rate	Batched	Not Batched	Batched at L1	Batched at L2	Batched at L3	Completed Level
N(600, 900²), (ω =1.5)	4-600-120-24-5 (High)	50	48	1339701.25	0.62	0.97	1.0000	0.0000	0.9903	0.0097	0.0000	2.4
			45	1343022.00	0.33	0.94	1.0000	0.0000	0.9953	0.0047	0.0000	2
			40	1343388.75	0.15	0.88	1.0000	0.0000	0.9982	0.0018	0.0000	2
		100	96	1362552.00	1.93	0.97	1.0000	0.0000	0.9706	0.0294	0.0000	2.8
			90	1372762.75	1.06	0.93	1.0000	0.0000	0.9847	0.0153	0.0000	2
			80	1376516.38	0.57	0.88	1.0000	0.0000	0.9926	0.0074	0.0000	2
		200	192	1402072.75	3.97	0.97	0.9996	0.0004	0.9233	0.0763	0.0000	4
			180	1421088.00	3.36	0.93	1.0000	0.0000	0.9535	0.0464	0.0000	3.2
			160	1438775.25	2.04	0.87	1.0000	0.0000	0.9752	0.0248	0.0000	2
	4-600-150-38-9 (Medium)	50	48	1340984.75	0.72	0.97	1.0000	0.0000	0.9896	0.0104	0.0000	3.2
			45	1342702.38	0.30	0.94	1.0000	0.0000	0.9952	0.0048	0.0000	2.4
			40	1343361.25	0.15	0.88	1.0000	0.0000	0.9981	0.0019	0.0000	2
		100	96	1365554.38	2.16	0.97	0.9999	0.0001	0.9691	0.0308	0.0000	4
			90	1373701.25	1.13	0.93	1.0000	0.0000	0.9837	0.0162	0.0000	2.4
			80	1377551.25	0.65	0.87	1.0000	0.0000	0.9921	0.0079	0.0000	2
		200	192	1409527.25	4.52	0.98	0.9981	0.0019	0.9223	0.0756	0.0002	4
			180	1425240.38	3.66	0.93	0.9997	0.0003	0.9514	0.0484	0.0000	4
			160	1439926.38	2.12	0.87	1.0000	0.0000	0.9729	0.0271	0.0000	2
	4-600-200-66-22 (Low)	50	48	1342957.75	0.87	0.97	0.9998	0.0002	0.9886	0.0113	0.0000	4
			45	1342917.38	0.32	0.94	1.0000	0.0000	0.9949	0.0051	0.0000	2.8
			40	1343409.38	0.15	0.88	1.0000	0.0000	0.9981	0.0019	0.0000	2
		100	96	1372710.00	2.70	0.97	0.9986	0.0014	0.9683	0.0303	0.0000	4
			90	1377457.25	1.40	0.93	0.9998	0.0002	0.9822	0.0176	0.0000	4
			80	1378440.00	0.72	0.87	1.0000	0.0000	0.9919	0.0081	0.0000	2.4
		200	192	1421137.25	5.39	0.98	0.9932	0.0068	0.9267	0.0648	0.0017	4
			180	1443046.75	4.95	0.93	0.9984	0.0016	0.9459	0.0523	0.0001	4
			160	1447262.38	2.64	0.86	0.9998	0.0002	0.9708	0.0291	0.0000	4

Table 11
Results for demand scenario S-14 with 5-Level non-exclusive AHCS

ω	Clustering Structure (δ)	β	τ	Total Processing Cost	GAP (%)	Fill Rate	Batched	Not Batched	Batched at L1	Batched at L2	Batched at L3	Batched at L4	Completed Level
$N(600, 900^2)$, (ω=1.5)	5-600-120-24-5-1 (High)	50	48	1340014.63	0.65	0.97	1.0000	0.0000	0.9902	0.0098	0.0000	0.0000	2.6
			45	1342772.63	0.31	0.94	1.0000	0.0000	0.9954	0.0046	0.0000	0.0000	2
			40	1343431.25	0.15	0.88	1.0000	0.0000	0.9982	0.0018	0.0000	0.0000	2
		100	96	1362572.00	1.94	0.97	1.0000	0.0000	0.9705	0.0295	0.0000	0.0000	2.6
			90	1372584.75	1.05	0.93	1.0000	0.0000	0.9847	0.0153	0.0000	0.0000	2.6
			80	1377037.25	0.61	0.88	1.0000	0.0000	0.9925	0.0075	0.0000	0.0000	2
		200	192	1404454.75	4.14	0.97	0.9996	0.0004	0.9222	0.0774	0.0000	0.0000	5
			180	1420816.75	3.34	0.93	0.9999	0.0001	0.9537	0.0463	0.0000	0.0000	3.2
			160	1438524.75	2.02	0.87	1.0000	0.0000	0.9749	0.0251	0.0000	0.0000	2
	5-600-150-38-9-2 (Medium)	50	48	1340714.75	0.70	0.97	1.0000	0.0000	0.9897	0.0103	0.0000	0.0000	2.6
			45	1342670.00	0.30	0.94	1.0000	0.0000	0.9952	0.0048	0.0000	0.0000	2
			40	1343363.25	0.15	0.88	1.0000	0.0000	0.9982	0.0018	0.0000	0.0000	2
		100	96	1365237.25	2.14	0.97	0.9999	0.0001	0.9692	0.0307	0.0000	0.0000	4.4
			90	1373660.00	1.12	0.93	1.0000	0.0000	0.9839	0.0161	0.0000	0.0000	2.6
			80	1377761.63	0.66	0.87	1.0000	0.0000	0.9922	0.0078	0.0000	0.0000	2
		200	192	1409734.00	4.54	0.98	0.9988	0.0012	0.9217	0.0762	0.0003	0.0006	5
			180	1425320.38	3.66	0.93	0.9998	0.0002	0.9516	0.0482	0.0000	0.0000	5
			160	1439274.38	2.07	0.87	1.0000	0.0000	0.9731	0.0269	0.0000	0.0000	2
	5-600-200-66-22-7 (Low)	50	48	1342733.38	0.85	0.97	0.9999	0.0001	0.9886	0.0112	0.0000	0.0000	5
			45	1343190.38	0.34	0.94	1.0000	0.0000	0.9949	0.0051	0.0000	0.0000	3.8
			40	1343442.75	0.15	0.88	1.0000	0.0000	0.9981	0.0019	0.0000	0.0000	2.6
		100	96	1371584.00	2.61	0.98	0.9989	0.0011	0.9686	0.0300	0.0001	0.0002	5
			90	1377318.38	1.39	0.93	0.9998	0.0002	0.9820	0.0178	0.0000	0.0000	5
			80	1378389.63	0.71	0.87	0.9999	0.0001	0.9918	0.0081	0.0000	0.0000	3.8
		200	192	1416501.63	5.04	0.99	0.9969	0.0031	0.9264	0.0650	0.0017	0.0038	5
			180	1440727.25	4.78	0.94	0.9987	0.0013	0.9468	0.0516	0.0000	0.0003	5
			160	1447157.63	2.63	0.86	0.9998	0.0002	0.9704	0.0294	0.0000	0.0000	5

Table 12
Results for demand scenario S-27 with 3-Level non-exclusive AHCS

ω	Clustering Structure (δ)	β	τ	Total Processing Cost	GAP (%)	Fill Rate	Batched	Not Batched	Batched at L1	Batched at L2	Completed Level
$N(800, 1600^2)$, (ω=2.0)	3-800-160-32 (High)	50	48	2838819.00	0.25	0.97	1.0000	0.0000	0.9958	0.0042	2
			45	2840398.75	0.10	0.94	1.0000	0.0000	0.9984	0.0016	2
			40	2841041.25	0.05	0.89	1.0000	0.0000	0.9993	0.0007	2
		100	96	2871278.75	0.90	0.97	1.0000	0.0000	0.9861	0.0139	2.2
			90	2881962.00	0.42	0.94	1.0000	0.0000	0.9937	0.0063	2
			80	2885236.75	0.23	0.88	1.0000	0.0000	0.9974	0.0026	2
		200	192	2927182.50	2.45	0.97	1.0000	0.0000	0.9587	0.0413	2.8
			180	2955904.00	1.49	0.93	1.0000	0.0000	0.9783	0.0217	2
			160	2971963.25	0.84	0.87	1.0000	0.0000	0.9893	0.0107	2
	3-800-200-50 (Medium)	50	48	2839612.50	0.28	0.97	1.0000	0.0000	0.9955	0.0045	2
			45	2840713.75	0.11	0.94	1.0000	0.0000	0.9983	0.0017	2
			40	2840977.00	0.05	0.89	1.0000	0.0000	0.9993	0.0007	2
		100	96	2876512.75	1.08	0.97	1.0000	0.0000	0.9848	0.0152	2.4
			90	2883333.25	0.46	0.94	1.0000	0.0000	0.9932	0.0068	2
			80	2884889.25	0.22	0.88	1.0000	0.0000	0.9974	0.0026	2
		200	192	2938668.00	2.85	0.97	0.9997	0.0003	0.9566	0.0431	3
			180	2962564.75	1.72	0.93	1.0000	0.0000	0.9764	0.0236	2
			160	2973719.25	0.90	0.87	1.0000	0.0000	0.9890	0.0110	2
	3-800-266-89 (Low)	50	48	2840391.25	0.31	0.97	1.0000	0.0000	0.9951	0.0049	2.8
			45	2840771.25	0.11	0.94	1.0000	0.0000	0.9982	0.0018	2
			40	2841030.50	0.05	0.89	1.0000	0.0000	0.9993	0.0007	2
		100	96	2882204.75	1.28	0.97	0.9998	0.0002	0.9835	0.0162	3
			90	2883642.75	0.47	0.94	1.0000	0.0000	0.9929	0.0071	2.4
			80	2885566.75	0.24	0.88	1.0000	0.0000	0.9972	0.0028	2
		200	192	2954669.25	3.42	0.97	0.9977	0.0023	0.9568	0.0409	3
			180	2970834.75	2.00	0.93	0.9998	0.0002	0.9742	0.0256	3
			160	2976879.50	1.00	0.87	1.0000	0.0000	0.9883	0.0117	2.4

Table 13
Results for demand scenario S-27 with 4-Level non-exclusive AHCS

ω	Clustering Structure (δ)	β	τ	Total Processing Cost	GAP (%)	Fill Rate	Batched	Not Batched	Batched at L1	Batched at L2	Batched at L3	Completed Level
N(800, 1600²), (ω =2.0)	4-800-160-32-6 (High)	50	48	2838823.25	0.25	0.97	1.0000	0.0000	0.9958	0.0042	0.0000	2
			45	2840466.00	0.10	0.94	1.0000	0.0000	0.9984	0.0016	0.0000	2
			40	2841012.50	0.05	0.89	1.0000	0.0000	0.9993	0.0007	0.0000	2
		100	96	2872386.00	0.93	0.97	1.0000	0.0000	0.9861	0.0139	0.0000	2
			90	2881861.50	0.41	0.94	1.0000	0.0000	0.9938	0.0062	0.0000	2
			80	2884260.00	0.20	0.88	1.0000	0.0000	0.9974	0.0026	0.0000	2
		200	192	2928018.75	2.49	0.97	1.0000	0.0000	0.9585	0.0415	0.0000	3.6
			180	2956227.25	1.50	0.93	1.0000	0.0000	0.9783	0.0217	0.0000	2
			160	2971997.50	0.84	0.87	1.0000	0.0000	0.9896	0.0104	0.0000	2
	4-800-200-50-12 (Medium)	50	48	2839314.50	0.27	0.97	1.0000	0.0000	0.9955	0.0045	0.0000	2
			45	2840499.50	0.10	0.94	1.0000	0.0000	0.9983	0.0017	0.0000	2
			40	2840946.00	0.05	0.89	1.0000	0.0000	0.9993	0.0007	0.0000	2
		100	96	2877005.50	1.10	0.97	1.0000	0.0000	0.9847	0.0153	0.0000	2.8
			90	2883176.75	0.46	0.94	1.0000	0.0000	0.9931	0.0069	0.0000	2
			80	2884630.00	0.21	0.88	1.0000	0.0000	0.9973	0.0027	0.0000	2
		200	192	2937321.25	2.81	0.97	0.9997	0.0003	0.9569	0.0428	0.0000	4
			180	2961735.25	1.69	0.93	1.0000	0.0000	0.9763	0.0237	0.0000	2
			160	2974236.75	0.91	0.87	1.0000	0.0000	0.9889	0.0111	0.0000	2
	4-800-266-89-30 (Low)	50	48	2840065.25	0.30	0.97	1.0000	0.0000	0.9952	0.0048	0.0000	4
			45	2840902.25	0.12	0.94	1.0000	0.0000	0.9983	0.0017	0.0000	2.4
			40	2841123.00	0.06	0.89	1.0000	0.0000	0.9993	0.0007	0.0000	2
		100	96	2883378.75	1.32	0.97	0.9997	0.0003	0.9837	0.0160	0.0000	4
			90	2884290.75	0.50	0.94	1.0000	0.0000	0.9929	0.0071	0.0000	2.8
			80	2885039.25	0.22	0.88	1.0000	0.0000	0.9973	0.0027	0.0000	2
		200	192	2952715.25	3.35	0.98	0.9978	0.0022	0.9572	0.0402	0.0004	4
			180	2972182.00	2.05	0.93	0.9998	0.0002	0.9741	0.0257	0.0000	4
			160	2977352.75	1.02	0.87	1.0000	0.0000	0.9884	0.0115	0.0000	2.4

Table 14
Results for demand scenario S-27 with 5-Level non-exclusive AHCS

ω	Clustering Structure (δ)	β	τ	Total Processing Cost	GAP (%)	Fill Rate	Batched	Not Batched	Batched at L1	Batched at L2	Batched at L3	Batched at L4	Completed Level
N(800, 1600²), (ω =2.0)	5-800-160-32-6-1 (High)	50	48	2839049.50	0.26	0.97	1.0000	0.0000	0.9957	0.0043	0.0000	0.0000	2
			45	2840605.25	0.10	0.94	1.0000	0.0000	0.9984	0.0016	0.0000	0.0000	2
			40	2840874.00	0.05	0.89	1.0000	0.0000	0.9994	0.0006	0.0000	0.0000	2
		100	96	2871904.50	0.92	0.97	1.0000	0.0000	0.9861	0.0139	0.0000	0.0000	2
			90	2882754.75	0.44	0.94	1.0000	0.0000	0.9937	0.0063	0.0000	0.0000	2
			80	2884920.75	0.22	0.88	1.0000	0.0000	0.9974	0.0026	0.0000	0.0000	2
		200	192	2927610.75	2.47	0.97	1.0000	0.0000	0.9586	0.0414	0.0000	0.0000	3.8
			180	2953368.00	1.40	0.93	1.0000	0.0000	0.9785	0.0215	0.0000	0.0000	2
			160	2972337.50	0.85	0.87	1.0000	0.0000	0.9895	0.0105	0.0000	0.0000	2
	5-800-200-50-12-3 (Medium)	50	48	2839792.50	0.29	0.97	1.0000	0.0000	0.9955	0.0045	0.0000	0.0000	2
			45	2840627.50	0.11	0.94	1.0000	0.0000	0.9983	0.0017	0.0000	0.0000	2
			40	2840930.25	0.05	0.89	1.0000	0.0000	0.9993	0.0007	0.0000	0.0000	2
		100	96	2876507.50	1.08	0.97	1.0000	0.0000	0.9847	0.0153	0.0000	0.0000	2.6
			90	2883204.50	0.46	0.94	1.0000	0.0000	0.9932	0.0068	0.0000	0.0000	2
			80	2885325.25	0.23	0.88	1.0000	0.0000	0.9973	0.0027	0.0000	0.0000	2
		200	192	2937053.50	2.80	0.97	0.9997	0.0003	0.9569	0.0428	0.0000	0.0000	5
			180	2961683.25	1.69	0.93	1.0000	0.0000	0.9765	0.0235	0.0000	0.0000	2
			160	2973215.25	0.88	0.87	1.0000	0.0000	0.9889	0.0111	0.0000	0.0000	2
	5-800-266-89-30-10 (Low)	50	48	2840675.50	0.32	0.97	1.0000	0.0000	0.9952	0.0047	0.0000	0.0000	4.4
			45	2840945.25	0.12	0.94	1.0000	0.0000	0.9983	0.0017	0.0000	0.0000	2
			40	2840979.50	0.05	0.89	1.0000	0.0000	0.9993	0.0007	0.0000	0.0000	2
		100	96	2883023.50	1.31	0.97	0.9997	0.0003	0.9837	0.0160	0.0000	0.0000	5
			90	2885027.25	0.52	0.94	1.0000	0.0000	0.9929	0.0071	0.0000	0.0000	2.6
			80	2885563.50	0.24	0.88	1.0000	0.0000	0.9972	0.0028	0.0000	0.0000	2
		200	192	2949834.75	3.25	0.98	0.9984	0.0016	0.9573	0.0404	0.0003	0.0005	5
			180	2972473.50	2.06	0.93	0.9997	0.0003	0.9743	0.0254	0.0000	0.0000	5
			160	2976890.50	1.01	0.87	1.0000	0.0000	0.9884	0.0116	0.0000	0.0000	3.8

Tables 12–14 show test results for demand scenario S-27 where the number of item types, average demand, and size of variation are large. For this demand scenario, dense cluster, small MABL, and tight MQC constraint are still effective to reduce the total processing cost although the effect of the density of

clusters and MQC tightness on the total processing cost become less. From the results of the analyses so far, we can conclude that the density of cluster, MABL, and MQC tightness should be determined simultaneously, not separately, considering interactions of these factors and economies of scale and scope in batch sizes. Also, it is important to set the number of levels considering the variation in the QTB, MABL, and MQC tightness. Main results for other aspects including the fill rate, the QB and QNB are as follows. First, the QB at lower (i.e., earlier) levels increases as the MQC tightness eases, whereas the total processing cost increases because the fill rate decreases as a result of the loose MQC. Here, notice that the DoH may decrease when the sequential batching process is completed at a lower level because clusters at lower level contain less different item types. Second, the QNB increases, which results in increased total processing costs as the MABL becomes larger. Third, as the density of cluster becomes lower, the total processing cost increases because the QNB increases owing to less alternatives for the batch formation. Fourth, as the density of cluster and MABL become lower and larger, respectively, the sequential batching process is completed at higher (i.e., later) levels, which causes an increase in the QB at higher levels. These tendencies become more apparent when the mean and variation in the QTB are small and low. On the contrary, the degrees of these tendencies are diminished for a larger mean and higher variation in the QTB. Here, notice that the DoH may increase when the sequential batching process is completed at a higher level because clusters at higher level contain more different item types.

The analysis results mentioned above can be interpreted in terms of truck-loading and input-lot formation in the wafer fab as follows: Increasing the density of a cluster by including more destinations to the cluster, setting the minimum quantity that should be loaded onto a truck as large as possible, and utilizing smaller trucks can reduce the overall cost of delivery when making a decision on truck-loading using the sequential batching with the MQC constraint. In particular, trucks with a smaller capacity offer more opportunities for reducing multiple deliveries because the sequential batching process is completed at a lower level in this case. Additionally, as the mean and variation of quantities to transport to destinations become smaller, increasing the density of clusters and utilizing small trucks have an advantage in reducing multiple deliveries to destinations. For the input-lot formation in the wafer fab, including more product types into a product line or family, setting the minimum requirement of the number of wafers included in each lot to be larger, and the maximum limit on the number of wafers in each lot to be smaller can reduce the overall production cost. In particular, a smaller lot-size offers more opportunities for reducing lot-splits during the fabrication process. As the mean and variation in the number of wafers for product types to be released into the wafer fab become smaller, increasing the number of product types included in a product line or family, and small lot-sizes provide an advantage in reducing lot-splits. Finally, small capacity of the truck and small lot size with loose MQC constraint are effective to reduce the DoH that results in less multiple deliveries and lot-splits.

5. Concluding Remarks

In this paper, we studied a sequential batching problem with an MQC constraint in *N*-level non-exclusive AHCS. Here, batching priorities are assigned based on AHCSs and the MQC constraint is applied for more effective control over the DoH in the batching results. The sequential batching arises in many areas including input lot formations in semiconductor wafer fabs, presort loading of commercial bulk mail for discounted mailing fees, and determination of truckloads in VMI environments. We developed a sequential batching algorithm that makes batches for two successive levels sequentially from the first level using properties identified to find better solutions and compute lower bounds. Computational experiments were performed for evaluating the sequential batching algorithm and finding some managerial insights. The algorithm solved all the problem instances quickly enough for solving large-sized practical problems. Furthermore, the average percentage GAP was 2.07%, which means that the algorithm finds very good lower and upper bounds. We found some meaningful insights that can be properly interpreted depending on the application area. First, the MABL is a statistically significant factor

on the total processing cost, whereas the number of levels is not significant. Second, dense cluster, small MABL, and tight MQC constraint are effective to reduce the total processing cost. Third, interactions among the density of clusters, MABL, and the MQC tightness have effect on the total processing cost when both the average demand and size of the variation are small. The interaction between the MABL and MQC tightness affects the total processing cost for larger average demand or larger variation. Fourth, a small MABL and tight MQC help reduce the total processing cost for clustering structures with more (deep) levels when the variation in the QTB is not large. Fifth, small MABL with loose MQC constraint seem to be useful for reducing the DoH in the batching results. From these results, we suggest that the density of cluster, MABL, and MQC tightness should be determined simultaneously, not separately, considering interactions of these factors and economies of scale and scope in batch sizes. Also, it is important to set the number of levels considering the variation in the QTB, MABL, and MQC tightness. This research is a groundwork to study various real-world problems in which item sizes vary and, therefore, the number of items included in each batch may vary. Such problems are commonly encountered in the logistics industry, where heterogeneous items are loaded onto vehicles and transported to several destinations. Furthermore, the cost of batch processing may vary depending on the quantities and types of items included in batches. In flexible manufacturing environments, various items with different processing requirements are batched and processed together. It is necessary to conduct well-designed computational experiments using domain-specific field data to apply the main results of this study to these actual problems in manufacturing, logistics, and so forth. Results of this study can provide guidelines for computational experiments and their analysis. In addition, rather than deciding batch configuration and clustering structure independently, we should determine them together as an integrated perspective. We can combine the sequential batching algorithm with clustering techniques based on machine learning for such integrated problems.

Acknowledgements

References

Andreopoulos, B., An, A., Wang, X., & Schroeder, M. (2009). A roadmap of clustering algorithms: finding a match for a biomedical application. *Briefings in Bioinformatics*, *10*(3), 297-314.

Arifin, A. Z., & Asano, A. (2006). Image segmentation by histogram thresholding using hierarchical cluster analysis. *Pattern Recognition Letters*, *27*(3), 1515-1521.

Bang, J., Kim, Y., & Choi, S. (2012). Multiproduct lot merging-splitting algorithms for semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing*, *25*(2), 200-210.

Bang, J., Lim, S., & Kim, J. (2016). Multi-dimensional dynamic programming algorithm for input lot formation in a semiconductor wafer fabrication facility (in Korean). *Journal of the Society of Korea Industrial and Systems Engineering*, *39*(1), 73-80.

Bouguettaya, A., Yu, Q., Zhou, X., & Song, A. (2015). Efficient agglomerative hierarchical clustering. *Expert Systems with Applications*, *42*(5), 2785-2797.

Cameron, D.A., Middleton, F.A., Chenn, A., & Olson, E.C. (2012). Hierarchical clustering of gene expression patterns in the Eomes + lineage of excitatory neurons during early neocortical development. *BMC Neuroscience*, *13*, 90.

Chen, H., Du, B., & Huang, G.Q. (2011). Scheduling a batch processing machine with non-identical job sizes: a clustering perspective. *International Journal of Production Research*, *49*(19), 5755-5778.

Cheng, L., Connor, T.R., Siren, J., Aanensen, D.M., & Corander, J. (2013). Hierarchical and spatially explicit clustering of DNA sequences with BAPS software. *Molecular Biology and Evolution*, *30*(5), 1224-1228.

Comert, E.C., Yazgan, H.R., Kir, S., & Yener, F. (2018). A cluster first-route second approach for a capacitated vehicle routing problem: a case study. *International Journal of Procurement Management*, *11*(4), 399-419.

Coral, D.B., Santos, M.O., & Toledo, C.F.M. (2017). Clustering-based crossover in an evolutionary algorithm for the vehicle routing problem with time windows. *Proceeding of IEEE 29th International Conference on Tools with Artificial Intelligence* (*ICTAI*), 830-837.

Costa, G., Manco, G., Ortale, R., & Ritacco, E. (2013). Hierarchical clustering of XML documents focused on

structural components. *Data & Knowledge Engineering*, *84*, 26-46.

Delgoshaei, A., Delgoshaei, A., & Ali, A. (2019). Evolution of clustering techniques in designing cellular manufacturing systems: A state-of-art review. *International Journal of Industrial Engineering Computations*, *10*(2), 177-198.

Han, X., Yu, Y., & Hu, G. (2019). A dynamic newsvendor problem with goodwill-dependent demands and minimum commitment. *Omega*, *89*, 242-256.

Hintsch, T., & Irnich, S. (2020). Exact solution of the soft-clustered vehicle-routing problem. *European Journal of Operational Research*, *280*(1), 164-178.

Huang, H., & Ma, Y. (2019). A hybrid clustering approach for bag-of-words image categorization. *Mathematical Problems in Engineering*, *2019*, Article ID 4275720.

Jun, C., Lee, J., & Lee, Y. (2010). A Biclustering Method for Time Series Analysis. *Industrial Engineering and Management Systems*, *9*(2), 131-140.

Kim, J., & Lim, S. (2012). Order-lot pegging for minimizing total tardiness in semiconductor wafer fabrication process. *Journal of the Operational Research Society*, *63*, 1258-1270.

Langfelder, P., Zhang, B., & Horvath, S. (2008). Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. *Bioinformatics*, *24*(5), 719-720.

Lim, A., & Xu, Z. (2006). The bottleneck problem with minimum quantity commitments. *Naval Research Logistics*, *53*(1), 91–100.

Lim, S., Kim, J., & Kim, H. (2014). Simultaneous order-lot pegging and wafer release planning for semiconductor wafer fabrication facilities. *International Journal of Production Research*, *52*(12), 3710-3724.

Lim, S., Kim, J., & Shin, Y. (2015). Optimal three-level presort loading of commercial bulk mail in the postal service industry. *Journal of the Operational Research Society*, *66*(6), 1007-1022.

Lim, S., Bang, J., & Kim, J. (2017). Multi-dimensional dynamic programming algorithm for *N*-level batching with hierarchical clustering structure. *Mathematical Problems in Engineering*, *2017*, Article ID 6021708.

Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*, 86-97.

Nagasawa, K., Irohara, T., Matoba, Y., & Liu, S. (2012). Selecting ordering policy and items classification based on canonical correlation and cluster analysis. *Industrial Engineering and Management Systems*, *11*(2), 134-141.

Nunez-Iglesias, J., Kennedy, R., Parag, T., Shi, J., & Chklovskii, D.B. (2013). Machine learning of hierarchical clustering to segment 2D and 3D images. *PLoS One*, *8*(8), e71715.

Özdamar, L., & Demir, O. (2012). A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. *Transportation Research Part E*, *48*(3), 591-602.

Saunders, D.G.O., Win, J., Cano, L.M., Szabo, L.J., Kamoun1, S., & Raffaele, S. (2012). Using hierarchical clustering of secreted protein families to classify and rank candidate effectors of rust fungi. *PLoS One*, *7*(1), e29847.

Vakharia, A.J., & Wemmerlöv, U. (1995). A comparative investigation of hierarchical clustering techniques and dissimilarity measure applied to the cell formation problem. *Journal of Operations Management*, *13*(2), 117-138.

Yue, H., Guan, Q., Pan, Y., Chen, L., Lv, J., & Yao, Y. (2019). Detecting clusters over intercity transportation networks using K-shortest paths and hierarchical clustering: a case study of mainland China. *International Journal of Geographical Information Science*, *33*(5), 1082-1105

Zaitoun, N.M., & Aqel, M.J. (2015). Survey on image segmentation techniques. *Procedia Computer Science*, *65*, 797-806.

Zhu, X., & Guo, D. (2014). Mapping large spatial flow data with hierarchical clustering. *Transportation in GIS*, *18*(3), 421-435.