Flex Serv Manuf J
DOI 10.1007/s10696-015-9223-5

Order batching in a bucket brigade order picking system considering picker blocking

Soondo Hong[1] · Andrew L. Johnson[2] · Brett A. Peters[3]

Abstract Bucket brigade order picking improves operational productivity by balancing workloads among pickers with a minimal level of managerial planning and oversight. However, due to variability and uncertainty of the pick locations within a particular order or batch, pickers can encounter blocking delays and thus lose productivity. This study formulates a model to quantify blocking delays and develops a control model to reduce blocking in bucket brigade order picking systems. The Indexed Batching Model for Bucket brigades (IBMB) has indexed batching constraints for generating batch alternatives, bucket brigade picker blocking constraints for quantifying blocking delay, and release-time updating constraints for progressively connecting the batching results with blocking quantification. The IBMB minimizes total retrieval time and improves picker utilization from 2 to 9 % across diverse and practical order picking situations while maintaining the static Work-In-Process. We note that modeling the separation of retrieved batches into orders still remains a challenge.

Keywords Bucket brigade order picking · Picker blocking · Mixed Integer Programming · (Dynamic) Control model

✉ Soondo Hong
soondo.hong@pusan.ac.kr

Andrew L. Johnson
ajohnson@tamu.edu

Brett A. Peters
petersba@uwm.edu

[1] Department of Industrial Engineering, Pusan National University, Busan 609-735, South Korea

[2] Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77843, USA

[3] Department of Industrial and Manufacturing Engineering, University of Wisconsin - Milwaukee, Milwaukee, WI 53201, USA

Published online: 30 August 2015

1 Introduction

Small-sized orders in broken-case picking are common in the warehousing industry (De Koster et al. 2007; Frazelle 2002; Napolitano 2013; Tompkins et al. 2003; Yu and De Koster 2009). Generally, manual picking is needed to handle irregular shapes which comprise small-sized orders (Frazelle 2002), and the manual operations affect the operational throughput, particularly the order picking throughput. As human pickers retrieve small-sized orders, throughput drops and management adapts zoning and batching strategies to compensate the loss (Yu and De Koster 2009). Zone order picking improves operational productivity by facilitating pickers familiarity with their operational environments, and a batch picking strategy increases throughput by reducing the number of trips and providing more operational stability.

The bucket brigade concept that originated from general assembly-line operations is a well-known solution to a fixed zone approach. When an assembly-line faces product diversity and demand fluctuation, however, the fixed zone approach has difficulty coordinating the workload across zones, and the worst-performing zone, i.e., the bottleneck zone, will determine the throughput. Employing a bucket brigade order picking strategy relocates the human labor to balance the workload per worker (Bartholdi and Eisenstein 1996a), i.e., each worker independently adjusts his/her workload within his/her skill allowance. The result is a balanced workload distribution regardless of product diversity and demand fluctuation, and increased throughput.

The drawback of this strategy is that it does not allow an upstream worker to pass over a downstream worker while balancing workload, and so the upstream worker remains idle until the downstream worker has left the occupied space (Bartholdi and Eisenstein 1996b). To mitigate the blocking delay, the bucket brigade rule assigns pickers in the slowest to fastest order (Bartholdi and Eisenstein 1996a, b), which maintains efficient operations for both upstream and downstream pickers. The bucket size also plays an important role in reducing blocking (Bartholdi and Eisenstein 1996a), because orders can be combined into a larger-sized bucket, which pools the variation of pick distributions over the order picking line. Bartholdi and Eisenstein (1996a) construed that the smaller variation of pick locations and numbers can result in less blocking delay when the size is appropriately selected.

A combined batch picking and sequencing model reduces blocking delay even further (Hong et al. 2012a). When a relatively large number of orders form a set of batches, management can decide the assignment of orders to batches and the release sequence of the batches to the order picking line. To make the batching and sequencing decisions, management needs to model the order fulfillment process, particularly blocking. Hong et al. (2012a)'s batching model utilizes the traversal routing property of pickers in a narrow-aisle order picking system, where every aisle has a unique entrance direction, and the pickers pass through aisles without a pass allowance due to the aisle width. Their model considers single order or single batch picking such that a picker completes one order or batch without zoning.

To our knowledge, no previous work has directly mitigated blocking delays while maintaining batch formations in bucket brigade order picking. Therefore, this paper proposes an Indexed Batching Model for Bucket brigades (IBMB) for mitigating picker blocking in bucket brigade order picking. We analyze a batching model including the bucket brigade concept and identify the blocking model to be combined with the bucket brigade batching model. We use a Mixed Integer Programming (MIP) to integrate the batching model and the blocking model. The resulting IBMB groups multiple orders and sequences or assigns them to pickers to reduce blocking delay.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature and the issues and Sect. 3 defines bucket brigade order picking and picker blocking delay. Section 4 introduces the IBMB and the proposed control policy. Section 5 summarizes the simulation results. Section 6 concludes this paper and suggests future research.

2 Literature review

Among the researchers (Bartholdi and Eisenstein 1996a, b, 2005; Hong 2014; Hong et al. 2015; Koo 2009) who have investigated blocking in order picking systems and bucket brigades, Bartholdi and Eisenstein (1996a), Bartholdi and Eisenstein (2005), and Koo (2009) addressed specific approaches to reduce blocking delays. Bartholdi and Eisenstein (1996a), who presented the order picking strategy based on the bucket brigades from their assembly-line model, found that there was less blocking delay when pickers were sequenced with the slowest worker in the location most upstream and the fastest worker in the location most downstream, and highlighted that the bucket brigade order picking can mitigate blocking delay by modulating the bucket size between batches. Later, Bartholdi and Eisenstein (2005) introduced a passing method to improve performance; when the downstream worker is busy conducting the current operation, the upstream worker does not have to wait, but instead passes over the downstream worker. Koo (2009) demonstrated how picker blocking and hand-off delay reduce bucket brigade order picking system (OPS) productivity when pickers have the same capability. Each picker's picking area is defined by a downstream boundary where the upstream pickers can leave totes if the downstream picker is not available.

Batch picking aggregates multiple orders in a trip to reduce the operational cost in order picking (Hong et al. 2012a, b; Yu and De Koster 2009). While determining a set of batches, travel distance and picker blocking can be accounted for and operational throughput improved (Hong et al. 2012a, b). A very tightly packed batch set experiences relatively low congestion (Hong et al. 2012b) and a well-sequenced batch set mitigates delays (Hong et al. 2012a). Hong et al. (2012a) developed a narrow-aisle Indexed Batching Model (IBM) which generates batches to control picker blocking in an OPS with multiple narrow-aisles when passing is not possible. A batch index represents the batches release sequence and the IBM assigns orders to indexed batches and determines the retrieval routes for each batch.

There are some important drawbacks to the studies cited above. Bartholdi and Eisenstein (1996a) and Koo (2009) both slow the order picking process, because increasing the batch size and the constrained work zone tends to require more Work-In-Process (WIP). Temporal extensions of WIP that affect sorting and packing worsen the overall order retrieval process, so that warehouse management has to invest more money to improve sorting and packing. The passing method by Bartholdi and Eisenstein (2005) is inappropriate for general order picking configurations due to the additional space required for both pickers and totes. The combined batching and sequencing model (Hong et al. 2012a) limits the application as single batch picking. Finally, the IBM lacks the mechanisms that help to maintain batch formations such as hand-off operations and blocking delays.

3 Problem definition

A bucket brigade order picking strategy is useful when combined with a flow-rack order picking system (OPS). Figure 1a illustrates the flow-rack OPS discussed in Bartholdi and Eisenstein (1996a). In this paper, we consider a linear order-picking process with one loading station and one unloading station adapted from Hong et al. (2015) as shown in Figure 1b. Pickers travel through the passage space, obtain items from the shelves, and place the retrieved items in a bin (or tote) on the conveyor. Each picker along the bucket brigade remains in sequence and each follows the decentralized algorithm.

In bucket brigade order picking, each picker is assigned to only one batch at a time and an order is progressively filled by the picking operations of all pickers. There are two directional picker movements. An upstream picker moves forward while filling items. When the upstream picker meets a downstream picker and the downstream picker has no tote, the upstream picker hands over the tote to the

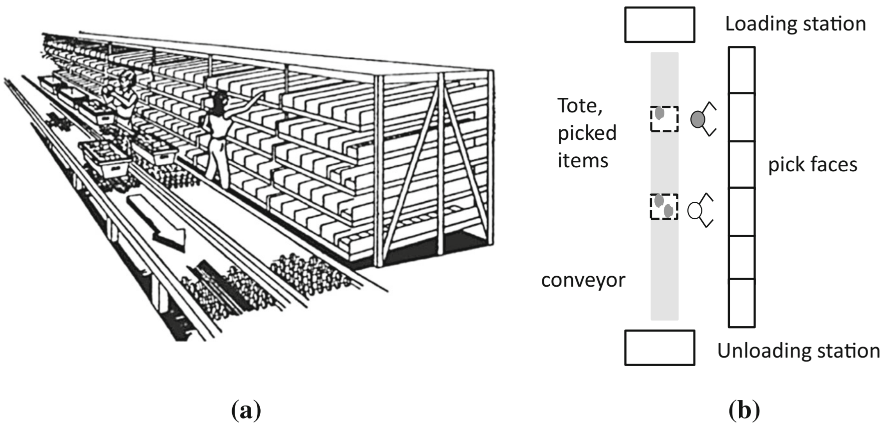

Fig. 1 A flow-rack OPS: **a** physical layout (adapted from Bartholdi and Eisenstein 1996a); and **b** *top view* (adapted from Hong et al. 2015)

downstream picker. Then the now-idle upstream picker, who no longer has a tote, moves backward to take over a new tote from the next picker further upstream. No picker can pass over the other downstream and upstream pickers regardless of the length of the idleness.

Figure 2 shows the bucket brigade order picking operation and hand-overs. Let batches be indexed from the first released to the next released $1, 2, \ldots, i, i+1, \ldots$, pickers be assigned from upstream to downstream $1, \ldots, k, k+1, \ldots$, and the passage space be defined as a set of pick faces. Figure 2a depicts a situation where the upstream picker k takes over a new batch i, begins picking, and places items in the tote. The pick face where the upstream picker k takes over a new batch becomes the hand-over location of the upstream picker k of batch i. The downstream picker $k+1$ continues to fill an order or batch $i-1$ in the same manner. Figure 2a, b show how two adjacent pickers independently handle two batches and progressively extend their operational boundaries. Figure 2c shows that when the downstream picker completes a batch, the downstream picker releases the tote of the batch $i-1$ and moves backward. Upon meeting the upstream picker k, the downstream picker $k+1$ takes over the tote of batch i. At this hand-off pick face, the upstream picker finishes the picks of batch i and the downstream picker resumes picking batch i (Figure 2d). After releasing batch i, the upstream picker moves backward to take over a new batch, and the downstream picker continues to pick batch i and completes the current batch (Figure 2e). The bounded trajectories in Fig. 2 address the relationship between pickers and batches. The first bounded trajectory appears in Fig. 2c. The bounded trajectories in Fig. 2d, e are related by a hand-off operation relevant to batch i.

When a downstream picker is busy picking and an upstream picker still has pick(s) remaining below the downstream picker, picker blocking occurs and productivity drops. Assume that pickers conduct picking and walking operations with a tote and perform identically in walk time and in pick time. Pick requirements are random over pick locations, and the upstream picker often encounters blocking when the downstream picker is busy picking item(s) (Bartholdi and Eisenstein 1996a). Picker blocking occurs when an upstream picker (k) tries to move forward to the next pick face that is occupied by a busy downstream picker ($k+1$). The

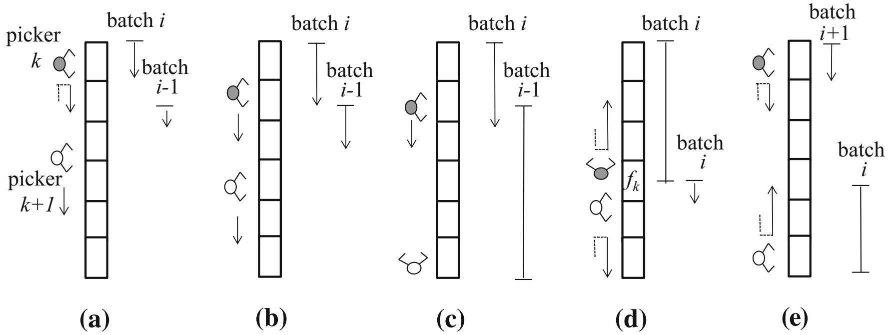

Fig. 2 Bucket brigade order picking and pickers' spatial trajectories over batches

upstream picker cannot hand the current batch to the downstream picker, because the downstream picker is currently picking and the upstream picker cannot pass the downstream picker, because the pickers must maintain the sequence. Bucket brigade order picking also experiences another waiting delay by a downstream picker; hand-off delay occurs when an upstream picker is picking and a downstream picker encountering the upstream picker must wait until the upstream picker completes the pick. According to Koo (2009), the expected hand-off delay per occurrence is very close to the expected pick time at a pick face / 2. Completing a batch requires $PK - 1$ hand-off occurrences (Koo 2009), where PK stands for the number of pickers.

4 Indexed Batching Model for Bucket brigades (IBMB)

This section introduces the proposed conceptual model based on the Indexed Batching Model (IBM) developed in Hong et al. (2012a). We describe the proposed model's three constraints and a mechanism to include hand-off delays and backward walks. Finally, we incorporate them into a single MIP formulation.

4.1 Conceptual model

We number the pickers in the direction of work flow from $1 \ldots PK$. The decision model combines orders into batches and sequences them to reduce picker blocking. We consider a fixed set of O orders that we will partition and sequence into B batches. Our objective is to minimize the total operational time, i.e., the time to retrieve the set of O orders.

The IBMB aims to obtain a batch formation to maximize throughput by optimizing the total retrieval time by the sum of the tote loading and unloading time, pick time, walk time, and delay time of all batches. We assume that the unit loading and unloading times are constant across all batches. We approximate the pick time per batch as the number of picks in a batch times the unit pick time. The walking time is the travel distance times the unit walk time. The unit walk time includes the forward walk time and the backward walk time. The delay time is the waiting time of pickers due to picker blocking. We consider the time to hand off a batch from one picker to another as fixed.

The IBMB has three constraints: indexed batching constraints, picker blocking constraints, and release-time update constraints. To calculate and minimize the total retrieval time given a set of orders, the batching model includes "indexed batching constraints" and "picker blocking constraints." The combined "batching" and "sequencing" model uses a "batch index" to express the batch sequence. When the most upstream picker arrives at the loading station, "release-time update constraints" determine the starting time for the most upstream picker's next trip based on the previous batch's completion time. The abstracted IBMB is:

(Abstracted IBMB)

Min $\sum_{Batches} \begin{pmatrix} \text{Loading/Unloading time} \\ + \text{Pick time} + \text{Walk time} + \text{Time delayed} \end{pmatrix}$

subject to

Indexed batching constraints
Picker blocking constraints
Release-time updating constraints

The batch number in the IBMB represents the release sequence of each batch. Similar to general batching constraints, indexed batching constraints group orders under the capacity limitation. Because the bucket brigade OPS typically is a linear pick line, every batch has a unique route from the loading station to the unloading station. By default, the batch number also becomes the batch's release sequence from the bucket brigade OPS. Obviously, the first batch to arrive is the first to be filled and the first to leave the OPS system. The picker blocking constraints use the batch number to determine the release sequence of batches and the release-time updating constraints identify the previously released batches using the batch number.

Blocking delay represents the total time that a picker is blocked by a downstream picker. Picker k can be blocked by a downstream picker $k+1$ at some pick face f_k if picker $k+1$ is picking an item at pick face $f_{k+1} = f_k + 1$ and picker k's next pick is at some pick face $f_k' \geq f_{k+1}$. Picker blocking constraints measure the time delay by comparing the batch i's expected arrival time (t_i) at the next location with the batch $i-1$'s expected leaving time (t_{i-1}) from the location ($f_k + 1$). If batch i's expected arrival time at the next location is earlier than the batch $i-1$'s expected leaving time from the location, blocking delay lasts as long as the time gap between two batches, i.e., if $t_i < t_{i-1}$ at $f_k + 1$, blocking delay = $t_{i-1} - t_i$; otherwise, blocking delay = 0.

The effects from the hand-off still remain while determining the expected arrival time to the pick faces of the batches. A hand-off may increase the staying time of a batch at every pick face. Each batch experiences the same number of hand-offs repeatedly at the same locations (Bartholdi and Eisenstein 1996a). Thus, all batches will be delayed as long as the amount of the expected hand-off occurrences times the unit hand-off time. We adjust the hand-off delay by adding the expected hand-off time using the release-time constraints. Clearly, the release time of a batch is delayed as long as the hand-off time.

The release time constraints also assign the release time of a newly releasing batch using the completion time of the completed batch. According to the bucket brigade protocol, the number of released batches in a system is equal to the number of pickers. Thus, the completion time of batch i determine the release time of batch $i + PK$ after $PK - 1$ hand-offs. In details, the starting time of the $i + PKth$ batch is determined by the cumulative sum of hand-off delay and backward walk time upon completion of the ith batch, whereas the starting times of the first PK batches are determined by the pickers available times and current locations. We assume that the number of batches in the system is equal to the number of pickers, i is the batch index, and $E[HO]$ is the expected time delay per hand-off. The release time constraints update the release time of batch $i + PK$ using the following logic:

The starting time of batch $i + PK$ at loading station = the completion time of the i^{th} completed batch at unloading station + the expected backward travel time by picker PK for batch $i+1$ + the expected hand-off delay by picker PK for batch $i+1$ ⋮ + the expected backward travel time by picker 1 for batch $i+PK$ + the expected loading time by picker 1 for batch $i+PK$ = the completion time of the i^{th} completed batch at the unloading station + the expected backward travel time by pickers $1, \cdots, PK$ linked by batch i's completion + the expected hand-off delay by pickers $2, \cdots, PK$ + the expected loading time by picker 1 for batch $i+PK$ = the completion time of the i^{th} completed batch at the unloading station + unit backward time$\cdot n$ + $(PK-1)E[HO]$ + the expected loading time by picker 1 for batch $i+PK$

4.2 MIP formulation

The congestion mechanism of a bucket brigade OPS is identical to a circular-aisle abstraction model, which is a primitive model of a no-passing order picking system (Hong 2014; Hong et al. 2015). As a batching model, we consider a bucket brigade system as a special case of a multiple-aisle order picking system under a no-passing situation in a single route case. We formulate the abstracted IBMB using a MIP. The IBMB formulation completes the picker blocking constraints and the release time updating constraints.

The OPS has a linear aisle with n pick faces (the set of pick faces = F) numbered 1 to n. The L/U stations are numbered 0 and $n+1$, respectively, and are 0.5 pick faces away from pick face 1 and n. FT is the forward travel time between neighboring pick faces. BT is the backward travel time between neighboring pick faces. The walk time from 0 to $F+1$ is equal to $n \cdot FT$. The L/U stations are located at the front and rear of the aisle.

We assume that PK pickers are available initially and that management must assign all pickers. The number of batches is not given, but it is equal to or larger than the number of pickers PK. The two batch picking approaches pick-then-sort and sort-while-pick affect cart capacity. When a batch is completed, a new batch enters the system. Its entrance time combines the backward walk time and the expected hand-off delay into the previous batch completion time. There are two primary decisions, batch assignment and release sequence. We use an indexed batch variable (X_{oi}). Orders o's are assigned to batch i and the batch's release order is index i.

Indices and parameters

F, f	the set of pick faces, and its index $f \in F$
O, o	the set of orders, and its index $o \in O$
B, i	the set of batches, and its index $i \in B$
OP_{of}	the number of picks in order o at pick face f
OS_o	the number of picks in order o
ST_i	the starting time of the ith batch
$CAPA$	the capacity of a cart (batch size)
PT	the pick time to pick an item
FT	the forward walk time between two pick faces
BT	the backward walk time between two pick faces
$E[HO]$	the expected hand-off delay per occurrence
PK	the number of pickers

Decision variables

X_{oi}	1 if order o enters the ith batch; 0 otherwise
P_{if}, CP_{if}	the pick time of the ith batch at pick face f and its cumulative pick time
D_{if}, CD_{if}	the time delayed of the ith batch at pick face f and its cumulative time delayed
CW_{if}	the cumulative walk time of the ith batch to pick face f
CT_i	the completion time of the order which finishes at the ith batch

The goal is to minimize total walk time plus total time delayed (1) by batching all orders and sequencing the batches. Because all orders are batches, pick time is constant, and thus is not on the objective function. Walk time is the sum of the travel times of all batches. The travel time of the ith batch is the sum of the forward travel times ($= n \cdot FT$), the backward travel times ($= n \cdot BT$) if $i > PK$, and the hand-off time if $i > PK$. DT is obtained by summing the cumulative delay at the last pick face of all batches.

Formulation

$$\min\left(n \cdot WT \cdot |B| + n \cdot BT \cdot |B| + (PK-1)E[HO] \cdot |B| + \sum_{i \in B} CD_{i|F|}\right) \tag{1}$$

subject to

$$\sum_{i \in B} X_{oi} = 1, \quad \forall o \in O, \tag{2}$$

$$\sum_{o \in O} OS_o \cdot X_{oi} \le CAPA, \quad \forall i \in B, \tag{3}$$

$$P_{if} = PT \cdot \sum_{o \in O} X_{oi} \cdot OP_{of}, \quad \forall i \in B, \forall f \in F \tag{4}$$

$$CW_{if} = \begin{cases} ST_i & \text{if } i \le PK, f = 0 \\ CP_{i-PK,n} + CW_{i-PK,n+1} + CD_{i-PK,n} + n \cdot BT + (PK-1) \cdot E[HO] & \text{if } i > PK, f = 0 \\ CW_{i,f-1} + WT/2 & \text{if } f = 1 \text{ or } n+1 \\ CW_{i,f-1} + WT & \text{otherwise} \end{cases},$$
$$\forall i \in B, \forall f \in F \cup \{0, n+1\}, \tag{5}$$

$$CP_{if} = \begin{cases} P_{if} & \text{if } f = 1 \\ P_{if} + CP_{i,f-1} & \text{otherwise} \end{cases} \quad \forall i \in B, \forall f \in F, \tag{6}$$

$$CD_{if} = \begin{cases} D_{if} & \text{if } f = 0 \\ D_{if} + CD_{i,f-1} & \text{otherwise} \end{cases} \quad \forall i \in B, \forall f \in F \cup \{0\}, \tag{7}$$

$$D_{if} = \begin{cases} \max\left(CP_{i-1,f+1} + CW_{i-1,f+1} + CD_{i-1,f+1} - CW_{if} - WT/2, 0\right) & \text{if } f = 0 \\ \max\left(CP_{i-1,f+1} + CW_{i-1,f+1} + CD_{i-1,f+1} - CP_{if} - CW_{if} - CD_{if} - WT, 0\right) & \text{otherwise} \end{cases},$$
$$\forall i \in B, \forall f \in F \cup \{0\}. \tag{8}$$

An order cannot be separated (2) and a batch should not exceed the capacity (3). Constraints (3) are set for the item-based capacity. When there is order-based capacity, constant 1 replaces OS_o. The related variables are assigned as the release sequence is determined. The pick time vector of batch i at pick face f is updated with batch i's pick time (4). Constraints (5) update CW_{if} at the loading station and pick faces. At the loading station, CW_{if} is determined using the pickers available time (ST_i) or the completion time of the $PKth$ previous trip ($CP_{i-PK,n} + CW_{i-PK,n+1} + CD_{i-PK,n}$) plus the returning time to the entrance to the loading station. The starting time of batch $PK + 1$ is derived from the completion time of the first completed batch, because the first picker for the first batch will be assigned to pick the $PK + 1$ batch. Backward travel time and the expected hand-off delay are added. Constraints (6) and (7) calculate the cumulative pick time and delay time.

Constraint (8) calculates the time delayed (D_{if}) using the leaving time at pick face f. At an $f = 0$, the leaving time of batch i from pick face f is determined by CW_{if}, because there is no pick operation and no previous blocking delay. At a pick face ($f > 0$), the leaving time is assigned with $CP_{if} + CW_{if} + CD_{if}$. The leaving time of batch $i - 1$, i.e., the previously released batch, from next pick face $f + 1$ becomes $CP_{i-1,f+1} + CW_{i-1,f+1} + CD_{i-1,f+1}$. The difference between two leaving times becomes the time delayed D_{if} of batch i at pick face f. The time delayed always is greater than or equals to 0. Constraint (8) includes a logical operation *MAX*, which does not execute on some MIP solvers. Thus, we replace constraint (8) with constraint (8.1) and constraint (8.2).

$$D_{if} \geq \begin{cases} CP_{i-1,f+1} + CW_{i-1,f+1} + CD_{i-1,f+1} - CW_{if} - WT/2 & \text{if } f = 0 \\ CP_{i-1,f+1} + CW_{i-1,f+1} + CD_{i-1,f+1} - CP_{if} - CW_{if} - CD_{if} - WT & \text{otherwise} \end{cases}, \quad \forall i \in B, \forall f \in F \cup \{0\}, \tag{8.1}$$

$$D_{if} \geq 0, \quad \forall i \in B, \forall f \in F \cup \{0\}. \tag{8.2}$$

5 Simulation results

In general, bucket brigade OPSs are used in dynamic order picking situations with a relatively small number of orders available in an order picking time window. We use an order picking profile based on Bartholdi and Eisenstein (1996a) and Koo (2009) to evaluate the proposed procedure. We implement the IBMB to determine the content of batches and the sequence of batches.

5.1 Simulation design

The base case is an order picking operation with 50 pick faces and 5 pickers. A picker performs with pick:forward walk:backward walk times in the ratio 1.0:0.1:0.05. We compare two control cases: RANDOM = sequence orders into batches randomly and release them immediately after construction, and IBMB. The RANDOM case repeats 20 runs and reports the average of the performance measure.

We investigate four scenarios (Table 1) to evaluate both single order picking and batch order picking. The standard scenario uses the walk speed and picking capability configurations defined above. The capability scenario differentiates picking capabilities across pickers; the unit time per pick for 5 pickers is 1.2, 1.1,

Table 1 Summary of order picking simulation environments

Configuration	Values
Scenarios	Standard, capability, fast-walk, small-OPS
Mean of order sizes	2, 6, and 30
Number of items per order	Uniform distribution [min, max] = [mean/2, mean · 3/2]
Pick time	Triangular distribution [min, mode, max] = [0.5, 1.0, 1.5]
Forward walk time	0.1 or 0.05
Backward walk time	0.05 or 0.025
$E[HO]$	0.5
Performance measure	Utilization (%), time blocked (%), and hand-off delay (%)
Runs per instance	20 runs with 1000 orders
Number of batches per one IBMB cycle	5 orders or batches per one IBMB cycle

Table 2 Simulation results for single order picking (size of time window = 5 orders)

Scenarios	Order size	Utilization (%)			Time blocked (%)			Hand-off delay (%)			Run time (s)
		RANDOM	IBMB	Diff (%)	RANDOM	IBMB	Diff (%)	RANDOM	IBMB	Diff (%)	
Standard	4	30.85	32.26	4.59	7.27	2.64	63.69	4.04	4.28	−5.97	0.027
	10	48.98	52.15	6.47	10.24	3.97	61.20	4.14	4.64	−11.86	0.037
	30	67.93	73.81	8.65	12.27	4.47	63.62	2.82	3.21	−13.69	0.049
Capability	4	31.01	32.25	4.01	5.80	1.97	66.02	4.53	4.71	−3.99	0.025
	10	49.64	52.10	4.95	7.90	3.22	59.21	4.66	5.04	−8.13	0.036
	30	69.99	74.26	6.09	8.91	3.28	63.22	3.21	3.50	−9.12	0.050
Fast-walk	40	40.85	43.92	7.51	13.35	6.66	50.14	7.72	8.32	−7.85	0.029
	10	57.80	62.28	7.75	14.33	7.16	50.04	6.31	7.22	−14.40	0.038
	30	73.19	79.99	9.30	14.11	5.83	58.64	3.57	4.15	−16.30	0.051
Small-OPS	4	47.74	52.16	9.28	19.42	11.87	38.90	10.81	11.73	−8.56	0.018
	10	62.95	68.09	8.17	17.51	10.38	40.74	7.85	8.84	−12.56	0.023
	30	75.90	82.80	9.09	15.16	7.35	51.51	4.19	4.65	−11.05	0.031

Table 3 Simulation results for batch order picking from varying batch sizes and batching strategies

Scenarios	Batch size	Utilization (%)			Time blocked (%)			Hand-off delay (%)			Run time (s)
		RANDOM	IBMB	Diff (%)	RANDOM	IBMB	Diff (%)	RANDOM	IBMB	Diff (%)	
Standard	20	66.58	68.94	3.55	4.41	0.97	78.03	3.82	3.87	−1.34	0.088
	40	78.32	80.78	3.13	4.49	1.53	65.95	2.56	2.57	−0.64	0.167
Capability	20	67.13	68.65	2.26	2.83	0.38	86.40	4.18	4.49	−7.50	0.083
	40	79.63	81.16	1.92	2.29	0.40	82.42	2.85	2.91	−2.43	0.129
Fast-walk	20	74.77	78.16	4.53	5.78	1.25	78.44	5.33	5.76	−7.92	0.093
	40	83.91	86.88	3.54	5.00	1.56	68.83	3.25	3.41	−4.88	0.185
Small-OPS	20	79.14	83.38	5.36	7.39	1.94	73.73	5.99	6.77	−13.00	0.095
	40	86.98	89.95	3.42	5.48	2.06	62.35	3.47	3.76	−8.45	0.128

1.0, 0.9, 0.8, where an average picker still performs one pick per unit time. The fast-walk scenario has a pick:forward walk:backward walk time = 1:0.05:0.025 similar to the speeds proposed in Bartholdi and Eisenstein (1996a) and Koo (2009). The small-OPS scenario has only 25 pick faces.

We evaluate all four scenarios by varying the average order sizes among 4, 10, and 30 items for the single order picking strategy, and 2 items per order in the batch picking strategy. We randomly select the size of each order based on a uniform distribution [min, max] = [mean/2, mean·3/2]. Practical workload per picker derives from the literature as 2–4 picks per batch (Koo 2009) and 4 orders per batch Bartholdi and Eisenstein (1996a). Since an order size can vary, but is relatively small, we assume a batch picking policy of 20 items as a regular batch size (4 picks per picker, or equivalently 2 orders per picker) and 40 items for a heavy demand situation (8 picks per picker or equivalently 4 orders per picker). To reflect the random pick times, we draw the value from a triangular distribution of [min, mode, max] = [0.5, 1.0, 1.5]. We assume deterministic forward and backward walk times. $E[HO]$ uses 0.5, which means half of the unit pick time and assumed based on Koo (2009)'s study. Table 1 summarizes the experimental picking environments.

As a performance measure, we compare utilization (%), time blocked (%), and hand-off delay (%). Utilization is the percentage of time spent picking compared to overall operations. Time blocked represents a productivity loss, the percentage of time blocked compared to overall operations. Hand-off delay includes the ratio of hand-off waiting time to the overall time. The column labeled Diff (Tables 2 and 3) shows the comparison between RANDOM and IBMB. The column labeled Run time presents the computation time per cycle, where a cycle has 5 batches.

We use C language and ILOG CPLEX Callable Library C API 12.5 to implement the IBMB formulation. The executable files run on Windows 7 (*i*5 3.40 Ghz CPU, 12 GB memory, 64 bit implementation). We disable both the branch-and-cut option and the heuristic search option to evaluate the exact computational time.

5.2 Single order picking

We consider a short time window of 5 orders, i.e., $OS_o = 1 \, \forall o \in O$ and $CAPA = 1$, which is more common in practice. Using IBMB, we sequence them based on the current shop floor status [Table 2 (Standard)]. Recall that the IBMB only sequences orders because this is a single order picking situation. Compared to batch picking, single order picking produces more picker blocking due to a higher variation in the workloads assigned to pickers. We find that IBMB improves the utilization from 30.85–75.90 to 32.26–82.80 %. When the order sizes are medium or large, picker blocking is more of a concern, but IBMB's picker blocking control is still effective. When the workload is higher and blocking is more serious, IBMB outperforms RANDOM. The run times for the IBMB algorithm are less than 0.05 seconds to determine the release sequence for the 5 pickers.

In the capability scenario, RANDOM produces less picker blocking compared to the standard scenario. Thus, IBMB improvements are smaller, 4.01–6.09 %. Fast-walk and small-OPS situations consistently show increased utilization. We know that there will be more congestion when pickers work faster and the OPS is smaller. When more picker blocking is expected, the benefits of using IBMB increase.

In general, when picker blocking is significantly reduced, there is a small increase in hand-off delay, 3.99–16.30 %. This result explains that benefits in terms of picker blocking can be offset by increases in hand-off delay, but the benefits are still significant.

5.3 Batch order picking

Table 3 summarizes the results of varying the operational scenario and the batch size. Our batching strategy constrains each batch to be less than or equal to the capacity of the cart or picking support vehicle, i.e., a constant number of orders (or items), and each batch is packed as tightly as possible. Typically, the expected number of picks per batch is very close to the cart capacity because an additional batch will decrease the throughput rate. Compared to the single order strategy, this characteristic can produce less variation in the number of picks, and thus reduce picker blocking. IBMB reduces 65.95–78.03 % of picker blocking with, on average, 0.088–0.167 seconds of computational time per cycle (i.e., 5 batches). Utilization improves by 3.13–3.55 % in the standard picking situation. Specifically, the time blocked is 0.97–1.53 % compared to RANDOM values of 4.41–4.49 %. Hand-off delay shows minor increases or decreases. Notably, IBMB shows improvement in the capability scenario, where the unit time per pick for the 5 pickers is not identical and the pickers are optimally sequenced to maximize performance. IBMB shows higher utilization improvements of 3.42–5.36 % for the fast-walk and small-OPS order picking scenarios. The computational time remains low, with an average of 0.095 seconds when the batch size is 20, and 0.185 seconds when the batch size is 40.

6 Conclusions

This paper described the optimization of a bucket brigade OPS. First, a control model of picker blocking was constructed based on the interactions including the impacts of hand-off delay and backward walk speed. Second, a dynamic control method was developed to maximize order picking throughput by judiciously forming batches of orders without variation in the batch size and the WIP size, unlike alternative approaches which rely on increases of WIP to modulate the zones, but reduce overall productivity. The control mechanism integrates the sequencing and batching decisions which changed both the release sequence and the order grouping by managing the batching formation.

Our experiments find that a bucket brigade OPS almost always experienced picker blocking even though WIP is constant. We considered four common situations (i.e., standard, capability, fast-walk, and small-OPS scenarios). In all situations, the IBMB helped to reduce picker blocking. Picker blocking in the bucket brigade OPS partially traded off against hand-off delay; however, the reduction of picker blocking was more than the increase in hand-off delay. In addition, even though aggregating orders in order picking reduces the cost of the order fulfillment process, collecting each order into one bin requires separating and packing each customer order from each batch (i.e., consuming time and resources).

Based on our work, we suggest that future research should consider: (1) reducing the IBMB's computational burden; and (2) applying it to general manufacturing such as the assembly-line described in Bartholdi and Eisenstein (2005), or service systems such as fast food restaurants. Our configuration assumed that warehouse management assigned 5 pickers to manage an order picking line and assumed a 5-batch time window, but some situations will require longer time windows (i.e., when orders arrive more frequently, or when a line includes more pickers). Similar to alternating the Shortest Processing Time (SPT) rule and the Longest Processing Time (LPT) rule, the IBMB may work well in a no-system based operational situation. Our future work will apply the IBMB to more general bucket brigade situations.

Acknowledgments The authors thank the editor and the anonymous reviewers of the manuscript for their valuable suggestions and comments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2014R1A1A2053550).

References

Bartholdi JJ, Eisenstein DD (1996) Bucket brigades: a self-organizing order-picking system for a warehouse. Technical report, Georgia Tech

Bartholdi JJ, Eisenstein DD (1996) A production line that balances itself. Oper Res 44(1):21–34

Bartholdi JJ, Eisenstein DD (2005) Using bucket brigades to migrate from craft manufacturing to assembly lines. Manuf Serv Oper Manag 7(2):121–129

De Koster RBM, Le-Duc T, Roodbergen KJ (2007) Design and control of warehouse order picking: a literature review. Eur J Oper Res 182(2):481–501

Frazelle E (2002) World-class warehousing and material handling, 1st edn. McGraw-Hill, New York

Hong S (2014) Two-worker blocking congestion model with walk speed m in a no-passing circular passage system. Eur J Oper Res 235(3):687–696

Hong S, Johnson AL, Peters BA (2012a) Batch picking in narrow-aisle order picking systems with consideration for picker blocking. Eur J Oper Res 221(3):557–570

Hong S, Johnson AL, Peters BA (2012b) Large-scale order batching in parallel-aisle picking systems. IIE Trans 44(2):88–106

Hong S, Johnson AL, Peters BA (2015) Quantifying picker blocking in a bucket brigade order picking system. Int J Prod Econ. doi:10.1016/j.ijpe.2015.04.012

Koo PH (2009) The use of bucket brigades in zone order picking systems. OR Spectr 31(4):759–774

Napolitano M (2013) 2013 warehouse dc operations survey: multiple paths, same goal. Supply Chain Manag Rev 17(6):48–58

Tompkins JA, Bozer Y, Tanchoco J (2003) Facilities planning, 3rd edn. Wiley, Hoboken

Yu M, De Koster RBM (2009) The impact of order batching and picking area zoning on order picking system performance. Eur J Oper Res 198(2):480–490

Soondo Hong is an Assistant Professor in the Department of Industrial Engineering at Pusan National University, South Korea. He obtained his B.S. and M.S. in Industrial Engineering from POSTECH, South Korea, and his Ph.D. from the Department of Industrial and Systems Engineering at Texas A&M University. He worked on scheduling and material handling at semiconductor and display companies. His research interests include logistics operations and material handling in manufacturing, warehousing,

display, semiconductor, and container terminal industries. He is a member of the INFORMS and the Korean Institute of Industrial Engineers.

Andrew L. Johnson is an Associate Professor and co-director of the Laboratory for Energy-Sustainable Operations in the Department of Industrial and Systems Engineering at Texas A&M University. He obtained his B.S. from the Grado Department of Industrial and Systems Engineering at Virginia Tech and his M.S. and Ph.D. from the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Tech. His research interests include productivity and efficiency measurement, benchmarking, and applied problems. He has worked on research problems in a variety of industries including healthcare, energy, manufacturing and logistics. He is an associate editor of IIE Transactions and a member of the IIE, INFORMS, National Eagle Scout Association, and German Club of Virginia Tech. For more information see his website www.andyjohnson.guru.

Brett A. Peters is the Dean of the College of Engineering & Applied Science and Professor of Industrial and Manufacturing Engineering at the University of Wisconsin-Milwaukee. Before he became Dean in September 2012, he was a Professor of Industrial and Systems Engineering at Texas A&M University, where he served as Department Head from 2002 through 2011. His primary research area is in facility design and material handling systems in manufacturing, warehousing, and distribution applications. He previously served as President of the College-Industry Council on Material Handling Education. He is a Fellow of the Institute of Industrial Engineers.