# Bucket brigades

## The role of storage assignment policies


Adnan Fiaz


BMI paper


VU University
Faculty of Exact Sciences
Business Mathematics and Informatics
De Boelelaan 1081a
1081 HV Amsterdam


September 2007

# Preface

The BMI paper is part of the master programme of Business Mathematics and Informatics (BMI) at the VU University. The purpose of the paper is to perform a research on the basis of a problem which combines at least two of the three aspects of BMI.

The subject of my research is bucket brigades, an order picking system, and the influence of different storage assignment policies on this system. From my experience as a seasonal worker in a warehouse, I sought to find improvements to the order picking process. In my search I came across bucket brigades and found its flexibility an attractive feature. For this reason I chose to research the application of bucket brigades to warehouses with different storage assignment policies.

This paper would not have been possible without the constructive feedback and helpful suggestions of my supervisor, drs. Marco Bijvank. I gratefully thank him and my fellow students for their support during the performance of my research and the completion of my paper.

Adnan Fiaz

Amsterdam, September 2007

# Summary

In warehouses, order picking systems account for a substantial amount of labour and costs. Considerable research has therefore been performed in improving these systems. One of the approaches presently used to optimize the order picking process is bucket brigades. This approach, developed by Bartholdi and Eisenstein (1996a), improves the order picking process by balancing the workload of each individual picker. This maximizes the throughput, the amount of orders completed in a certain time period.

In this paper, the bucket brigade is placed in a warehouse with three different storage assignment policies: random, dedicated and class-based storage. According to Bartholdi and Eisenstein (1996a) the bucket brigade requires a uniform spreading of work content. The storage assignment policies determine this spreading, which therefore may not always be uniform. The objective of this paper is to study the effect the different storage assignment policies have on the performance of the bucket brigade.

This study is performed through a simulation approach. A fixed number of pickers and articles are chosen and orders are generated for each storage assignment policy. The bucket brigade is then implemented under two settings. First, the order pickers are allowed to pass one another. Second, a setting where blocking is enforced is also implemented to reveal the impact of blocking on the results.

The results show that the random storage policy supports the findings of Bartholdi and Eisenstein (1996a) because it complies with their assumption of a uniform spreading of work content. However, the class-based storage and the dedicated storage policy do not achieve a balance of the workload despite their low throughput. The non-uniform spreading of work content causes a considerable amount of variation in the points at which the orders are completed. This prevents the bucket brigade from converging to a balanced system.

In addition, when blocking is enforced and random storage is used the results show that the pickers must be sequenced from slowest to fastest according to their working speed. Otherwise the performance of the bucket brigade deteriorates and a balance of the workload is not achieved. Blocking has less impact on the results for dedicated storage and class-based storage. Under these policies the bucket brigade still does not converge to a balanced system.

Finally, the sensitivity analyses show the conclusions are only sensitive to the variation in the picker speeds. Increasing this variation brings a bucket brigade with class-based storage closer to a balanced system. Reducing the variation results in a slower convergence of the bucket brigade to a balanced system and also reduces the importance of blocking. Further research can focus on further integrating the bucket brigade in a warehouse by also considering routing of pickers. Batching can also be implemented to relief the order picking process from the variation caused by dedicated and class-based storage.

# Contents

# 1 Introduction

The sandwiches and salads restaurant franchise Subway® recommends its franchises to assemble sandwiches by bucket brigade[1]. The ant species *Messor barbarus* also uses a bucket brigade to carry food to the nest. Finally, Mitsubishi Consumer Electronics assembles large-screen televisions and packages cellular phones with the help of bucket brigades. These examples demonstrate the flexibility of the bucket brigade. This flexibility in applications is also why it is the primary subject of this paper.

Bucket brigades belong to the field of warehousing. This field examines the internal and external processes of a warehouse. To understand the purpose of this paper it is necessary to understand these processes. The following paragraphs therefore first explore the concept of a warehouse.

A *warehouse* is a temporary storage location for finished goods, goods-in-process or raw materials between points of origin and points of consumption (De Koster et al. 2006). Warehouses are also employed as a buffer location, where extra products are stored in case of unforeseen events, or as a distribution centre, where products are distributed from the warehouse to other locations. To a company, warehouses form an integral part of the logistics. Not only do they consume a large part of the logistic costs (20% according to a survey performed by ELA/AT Kearney (2004)), they also help to fulfil several company missions. Examples of these missions are achieving transportation economies, achieving production economies, meeting changing market conditions and uncertainties and supporting the firm's customer service policies (Lambert et al. 1998).

The processes performed in a warehouse can be subdivided into the following categories: receiving, storing, order picking, sortation and packaging and finally shipping. Figure 1 depicts these processes in a flow diagram.

[1] http://www.bucketbrigades.com

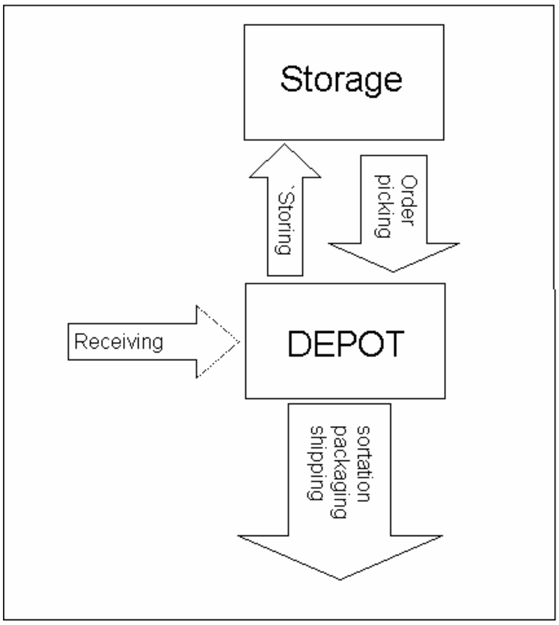

Figure 1: Activities performed in a warehouse (Tompkins et al. 2003, De Koster et al. 2006)

The focus of this paper is on the process of storing and order picking. Storing involves the allocation of incoming products to storage locations. There are several approaches available to perform this process, more commonly known as *storage assignment policies*. The objective of these policies is to minimize the picker's travel time. They can achieve this by for instance transferring the highest demand products closer to the depot. In Chapter 2, I will present more details on these policies but their impact on an order picker's travel time is large.

However, the primary process in a warehouse is order picking. According to several scientific papers (Goetschalckx and Ashayeri 1989, Drury 1988, Tompkins et al. 2003) order picking is the most labour-intensive operation in warehouses with manual systems, and also a capital-intensive operation in warehouses with automated systems. The *order picking process* is nevertheless straightforward. An order, a list of products requested by a customer, is given to a picker. This picker then retrieves the products from their storage location. The complexity arises from the fact that an order picker often has to retrieve several products (a set of orders or customer requests) before he can return to the depot and the retrieval must occur in the shortest time possible.

By minimizing the travel time, more products can be retrieved on a single day. This maximizes the *throughput*, the amount of products that leaves the warehouse within a certain time period. The storage assignment policies are one method to achieve maximum throughput. Other methods or decisions range from the warehouse layout design, order picker routing to the batching of orders. Although these decisions are often studied separately they are all part of an order picking system (OPS). An extensive literature review of the intricacies of an OPS is provided in De Koster et al. (2006). Chapter 2 provides a short review because my focus is on one particular OPS and its relationship with storage assignment policies.

The order picking system under study in this paper is referred to as a *bucket brigade*. It is based on the Toyota Sewn Products Management System (TSS), an assembly line procedure used in the apparel industry. In a TSS, each worker receives a specific amount of work (for order pickers this work is defined by the products mentioned on the order). The last worker, once finished with his task, walks back to the worker behind him and continues with his task. This worker does the same to the picker behind him. Finally, as this process continues, the first worker is reached and he must walk back to the depot to receive a new amount of work.

The difference between a TSS and a bucket brigade is that in a bucket brigade the workers are ordered from slowest to fastest and in a TSS no ordering is imposed. This leads to a system which balances the workload of each worker automatically (Bartholdi et al. 1996a). For the TSS workload balance has to be enforced manually. According to Bartholdi et al. (1996a) this also maximizes throughput, which is the objective of an OPS.

The bucket brigade balances workload and it maximizes throughput when it is applied to an assembly line but it may not always work in warehouses. This is because bucket brigades require work content to be spread continuously and uniformly along the assembly line (Bartholdi et al. 1996a). In warehouses work content consists of products which need to be retrieved from their location. For the assignment of these locations to the products warehouses employ storage assignment policies. The spreading of work content therefore depends on the policy employed.

The purpose of this paper is therefore to study the relationship between bucket brigades and different storage assignment policies. By implementing different policies, I can study the effect a non-uniform spreading of work content, caused by a storage policy, has on the bucket brigade. In addition, the impact of the different policies on the worker ordering (from slowest to fastest) may also be revealed. Perhaps by allocating the products in a certain way one might be able to forgo this requirement.

The remainder of this paper is organized in a literature review (Chapter 2), a definition of the mathematical model (Chapter 3), the results of implementing the model (Chapter 4), the sensitivity analyses (Chapter 5) and the conclusion (Chapter 6).

# 2 Literature

Chapter 1 provided an introduction to the different processes active within a warehouse. In Section 2.1 the literature on warehousing is further explored, specifically the areas on order picking systems (OPS). It is not an outline of all available literature but is background information to show what aspects should be considered in developing and implementing an OPS. For a more comprehensive review the reader is referred to De Koster et al. (2006). Section 2.2 discusses the literature on bucket brigades, while Section 2.3 addresses storage assignment policies.

## 2.1 An order picking system

There are many different aspects to an order picking system. Researchers commonly choose one aspect to study and define assumptions for the other parts. I follow a similar path and start with what is perhaps the most tangible part, the warehouse layout. The layout affects the travelling routes of pickers and therefore the walking distance, the time to completion of an order and finally the throughput. There are two problems involved in determining this layout. First, there is the problem of *facility layout* where the location of the different departments (receiving, storing, shipping) is determined. For example, in Figure 1 it is assumed the receiving and shipping operations are both managed from the depot. However, the receiving process can also be handled at a location which is located above the storage. Kusiak and Heragu (1987), Meller and Gau (1996) and Tompkins et al. (2003) provide more information regarding this problem and give a review of the available literature.

The second problem dealt with by the layout design is the *aisle configuration problem*. The picking or storage area is often divided in several aisles, blocks or racks. The solution to the aisle configuration problem comprises the optimal number of blocks, aisles or racks as well as their length and width. Optimal in this sense is the state in which the travelling routes are minimized. There are several papers available addressing this problem, among them Bassan et al. (1980), Rosenblatt and Roll (1984), Caron et al. (2000), and Le Duc and De Koster (2005b). However, the Erasmus-Logistica website[2] also provides a literature review and has incorporated the knowledge relating to warehouse layout into a tool which can be used interactively to determine the optimal layout.

The different parts of an OPS are all dependent on one another. Therefore, to find the optimal layout design other aspects of the OPS also need to be considered. One of these other aspects is the *order picking method*. This part of order picking systems has been extensively discussed in the literature. Figure 2 presents a summary of the different types of picking methods.

[2] http://www.fbk.eur.nl/OZ/LOGISTICA

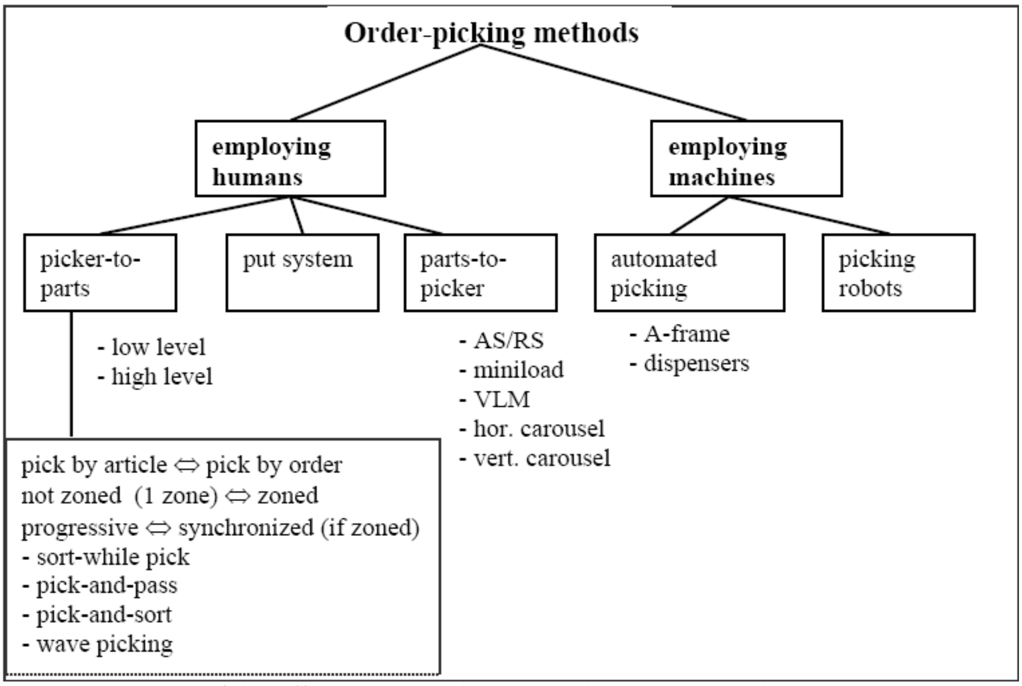

Figure 2: Overview of the different order picking methods (De Koster et al. 2006, De Koster 2004)

Again, the reader is referred to De Koster et al. (2006) for the specifics of each of the picking methods depicted in Figure 2. The picking method employed in bucket brigades is a low-level, picker-to-parts order picking method. In this method, order pickers walk alongside the aisles as they retrieve their articles from the storage racks. This is the most common type used in practice (De Koster 2004). There is no specific literature on order picking methods alone. These methods are almost always studied with other aspects of an OPS, such as routing, zoning or batching.

Routing methods try to find the shortest route for a picker through the warehouse and sequence the items on the order list accordingly. The problem of finding the shortest route for an order picker is related to the *Travelling Salesman Problem* (Lawler 1995). The difference between the order picker and the salesman is that the picker is allowed to visit the same location multiple times and also to return to the starting location before having visited all other locations. This classifies the order picker routing as a *Steiner Travelling Salesman Problem* (De Koster et al. 2006). There is research available on the analytical solution to this problem, mainly by Ratlif and Rosenthal (1983), Cornuéjols et al. (1985) and De Koster and Van der Poort (1998).

However, the difficulty of finding the optimal route is that certain aspects of the problem can not be modelled. For example, if the order pickers find the optimal route illogical or counterintuitive they may deviate from it. Furthermore, the above problem definition does not consider the possibility of aisle congestion. As opposed to the algorithmic methods, heuristic methods are more flexible and therefore more widely applied. These methods include the S-shape method, the return method, the midpoint method and the largest gap method. For a more elaborate discussion of these topics the reader is referred to Hall (1993) and Petersen (1997).

The routing part of an order picking system is dependent on the aspect of layout design. If more aisles are defined, the routing becomes more difficult. *Zoning* is also an aspect that is dependent on the layout. With zoning, the picking or storage area is divided into several sections (zones). Each section is then assigned to a picker, which receives orders (from other pickers or from the depot) for the articles in his section. In this case, the layout controls the maximum amount of zones the picking area can be divided in. De Koster et al. (2006) define the advantages of zoning as: reduced travel distance because of the smaller area a picker has to cover, reduced congestion and the increased familiarity of the picker with his zone which leads to faster picker speeds.

The largest disadvantage of zoning is that orders need to be split and distributed to the zones and combined again before shipment. To overcome this problem a parallel picking approach is used. In this approach pickers start on the same order simultaneously, after which the order is merged. Another approach that does not suffer from the disadvantage of splitting and recombining is the progressive picking approach. This approach forces pickers to pass their order to the picker in the next zone once all articles in their own zone have been picked.

There is little research performed on zoning. The subject does occur in Malmborg (1995), Peterson (2002) and Le Duc and De Koster (2005a). The lack of literature may be because of the complexity of the subject. De Koster et al. (2006) state that "the workload must be equally distributed over the order pickers. […] imbalance can cause serious deterioration of order throughput and order throughput time." Achieving this balance has been attempted by Jane (2000), Jane and Laih (2005) and Le-Duc and De Koster (2005a). The difficulty in practice is however to preserve this balance under changing environmental conditions (changing demand, different workers).

The final two aspects of an order picking system discussed in this section are storing and batching. For storing, Section 2.3 discusses the literature of importance to this paper. *Batching* comprises the division of the entire set of orders into smaller subsets. Normally, an order picker would receive a single order defining his pick route. By implementing batching, the order picker receives multiple orders defining his pick route together. There are two approaches to batching: *proximity batching* and *time window batching*.

For time window batching orders that arrive within a certain time window are automatically batched. This approach to batching is further explored and elaborated in Tang and Chew (1997), Le Duc and De Koster (2003, 2007) and Gibson and Sharp (1992). For proximity batching orders are batched based on the proximity of their locations to other orders. The problem is solved heuristically as it is NP-hard (Gademann et al. 2001, Gademann and Van de Velde 2005). These heuristics comprise of seed and savings algorithms. The seed algorithms first define a starting point (the seed) followed by the order congruency (adding the next order). A selection of papers on this subject comprises Elsayed (1981), Hwang et al. (1988), Pan and Liu (1995) and De Koster et al. (1999b). The savings algorithm originates from Clarke and Wright (1964) and their algorithm for the vehicle routing problem. De Koster et al. (1999b) report that even simple order batching methods lead to significant improvement. They also report certain conditions needed for the seed and the savings algorithms to perform best.

## 2.2 A bucket brigade

Most of the research on bucket brigades is performed by J.J. Bartholdi and D.D. Eisenstein. Bartholdi and Eisenstein (1996a) is the first scientific article to introduce the idea of bucket brigades. As explained in Chapter 1, this idea is based on the Toyota Sewn Management System (TSS). The main difference between the two procedures is the TSS imposes no ordering on the workers and the bucket brigade requires workers to be ordered from slowest to fastest. This ordering was introduced by Bartholdi and Eisenstein to show that with it the flow line becomes a self-balancing system with respect to workload.

A balance is necessary, as stated in Section 2.1, to prevent "deterioration of order throughput and order throughput time." Although this quotation from De Koster et al. (2006) refers to zoning, it is also applicable to bucket brigades. In zoning, the different zones are formed manually to balance the workload and maximize throughput. For bucket brigades the zones are determined dynamically because the system balances itself whenever the environment changes (different workers, changes in demand). Therefore, there is no need for planning or management of the flow line to preserve workload balance.

The ordering of the workers from slowest to fastest is an essential factor to achieve the self-balancing system. Bartholdi and Eisenstein (1996a) even prove that other sequences "can perform much worse but not too much better". The model used by these two authors, defined as the *Normative Model*, relies on the following assumptions and restrictions[3]:

Assumptions:

- *Total ordering of workers by velocity*. Each worker is characterized by a distinct, constant work velocity.
- *Insignificant walking time*. The total time to assemble a product is significantly greater than the time to walk the length of the (assembly) line.
- *Smoothness and predictability of work*. The nominal work content of the product is a constant (which is normalized to 1); and the work content is spread continuously and uniformly along the assembly line.

Restrictions:

- The workers are ordered from slowest to fastest along the flow line.
- The workers are not allowed to pass one another. If a worker is blocked by another worker he must wait until the other worker is finished.

[3] http://www.bucketbrigades.com/

Bartholdi et al. (1999) further explore the ordering of the workers and examine the asymptotic behaviour of bucket brigades with only 2 workers and one with 3 workers. Different orderings of the workers are tested and their effect on the stability of the system and the production rate is studied. The general conclusion is that a slow-to-fast ordering remains optimal regarding throughput but for certain orderings a stable but suboptimal flow line is possible. The stability of a flow line is expressed in terms of fixed hand-over points. A *hand-over point* is a point at which two workers exchange orders. According to Bartholdi et al. (1996a) fixed hand-over points indicate the workload is balanced because the worker performs the same amount of work every time he receives a new order (from the depot or from the worker behind him).

Armbruster and Gel (2002) also examine the ordering assumption and consider the case where the velocity of the workers is not constant. They use an assembly line with two workers to test the performance for both the passing case and the case in which workers are blocked. The bucket brigade proposed by Armbruster and Gel (2002) also converges to a self-balancing system although with more than one fixed hand-over point between the workers.

Bartholdi et al. (2001) focus on the third assumption of the *Normative Model*. Instead of constant, continuously and uniformly spread work content, they assume the work content to be stochastic. The authors show that as the number of workstations on the assembly line increases, the behaviour of the bucket brigades begins to resemble the predictions of the *Normative Model*. The forgoing of the third assumption is also the focus of this paper. This is achieved by implementing different storage assignment policies in a warehouse setting. In this point, the approach of Bartholdi et al. (2001) differs. Bartholdi et al. (2001) apply their bucket brigade to an assembly line with work stations. This setting is difficult to translate to a warehouse setting and it does not guarantee that the bucket brigade converges to the *Normative Model*.

Bratcu and Dolgui (2005) tackle an implicit assumption of the *Normative Model* and assume a finite, constant walk-back speed for each worker. In their results the bucket brigade still converges to a stable system where the hand-over points between the workers are fixed. However, this only occurs under the same restrictions as the *Normative Model*, workers must be ordered by their forward velocity from slowest to fastest. Bartholdi et al. (2003) assign each worker a distinct finite walk-back velocity. Passing between workers is also allowed. This results in extra constraints on the bucket brigade for it to converge to a stable system. The focus is no longer on the ordering of the workers based on their forward velocity alone, the backward velocity is considered as well. Violation of these constraints results in a chaotic system where the hand-over points between workers are random.

In almost all the literature mentioned above the focus is on assembly lines. Bartholdi and Eisenstein (1996b) is an exception. In this article the authors extensively discuss applying bucket brigades to a single aisle order picking system (flow rack). The considerations to be made in this situation are not different from an assembly line. The main problem now comes from the randomness of the orders and how one prevents workers from blocking one another. Bartholdi and Eisenstein (1996b) handle this by varying the bucket size and thus letting the order picker's work on more than one order at a time.

Another exception in the literature is Bartholdi and Eisenstein (2005) where the migration from craft assembly to an assembly line is performed with the help of bucket brigades. It is however beyond the scope of this paper. More examples of applications of bucket brigades can be found on http://www.bucketbrigades.com.

## 2.3 Storage assignment policies

The relationship between storage assignment and bucket brigades is not discussed in any specific scientific paper. Storing does not play a role in assembly lines, which may explain the lack of research in this area. Nevertheless, Bartholdi and Eisenstein (1996a) recognize that when work is "pathologically concentrated", workers block one another more often. This does not define the exact relationship between storage assignment and bucket brigades, although it is indicative for the consequences to the OPS if certain policies "pathologically concentrate" work.

To study the storage assignment policies in more detail, this section discusses the available literature on the subject. De Koster et al. (2006) define five often used policies: random, closest open location storage, dedicated storage, full turnover storage and class-based storage. With *random storage* each product is randomly assigned a location in the warehouse. Every location has equal probability of being selected. Choe and Sharp (1991) show that this method results in a high space utilisation and increased travel distance. Random storage also provides flexibility and is simple to use, it does not need an extensive administration of the characteristics of every product.

In practice, assigning random locations is computerized. Human workers would not be able to choose a random location; instead they would be inclined to choose *the closest open location*, the second policy. The closest open location storage is not an attractive policy because it results in occupied locations close to the depot and empty locations at the back. Space utilization is therefore low. In addition, Hausman et al. (1976) state the performance of this policy is similar to random storage.

The third policy is *dedicated storage*. Dedicated storage assignment allocates a fixed location to every product based on a certain product characteristic and once assigned the location does not change. Although this policy is inflexible and space utilisation is low, it does have the advantage that order pickers become familiar with the locations. The fourth policy is the *full turnover storage*. This policy assigns locations closest to the depot to the items with the highest turnover or demand. A disadvantage of this policy is that demand characteristics change constantly and thus the locations of certain products also change constantly.

Finally, *class-based storage* divides the items into several classes based on for instance their demand frequency. A location area within the warehouse is then assigned to each class. Within this area the items of that particular class are stored randomly. This allows the fast-moving products to be stored close to the depot and the applicability of the low storage space requirements and flexibility of random storage (De Koster et al., 2006).

# 3 Model

The mathematical model of Bartholdi and Eisenstein (1996a) applies an analytic approach to analyze the mechanism of the bucket brigades. This approach requires the assumptions of the *Normative Model* (Section 2.1) and therefore limits the study of certain mechanisms. To study the relationship between bucket brigades and storage assignment policies, I apply a simulation approach. This approach does not need all the assumptions of the *Normative Model*. In addition, it also incorporates the stochastic nature of the demand for the products in a warehouse.

This chapter elaborates on the model underlying the simulation. This model comprises of three parts and each of these parts is discussed separately in the following sections. Section 3.1 first explains the implementation of the different policies. Section 3.2 then explains the generation of orders, which are used in the bucket brigade. The model for the bucket brigade is explained in Section 3.3. Finally, Section 3.4 discusses the input and outputs needed for the simulation.

## 3.1 Storage assignment

As explained in Section 2.3, a storage policy assigns a location to an article. In this model I define the location of an article as the distance to the depot. To model the different storage policies an assumption is needed on the layout of the warehouse. Bartholdi and Eisenstein (1996b) assume a single flow rack (comparable to a single aisle). Although that is not representative of a warehouse in practice, I also incorporate this assumption. The focus of this paper is to study the relationship between bucket brigades and storage assignment policies. The warehouse layout is of secondary concern. In addition, changing both the warehouse layout and the storage policy would make it difficult to distillate the impact of either two on the bucket brigade.

The flow rack consists of as many locations as there are articles. Suppose there are $M$ articles. Each article is stored in a separate location and no locations have the same distance to the depot. The number of a location determines the distance of this location to the depot. The distance of location $m$ ($m<M$) is therefore equal to $m$ units. Before implementation of the storage policies is discussed there is one more item characteristic needed, namely the demand. This demand is commonly derived from empirical research. However, this data is not available. I therefore assume the demand characteristics to be distributed according to *Pareto's principle*[4].

Section 2.3 mentioned five storage assignment policies. Of these five I choose three policies: dedicated storage, class-based storage and random storage. I do not consider

[4] Pareto's principle originally stated that 80% of a country's wealth was distributed among 20% of its population. The principle was also found to apply to other areas, such as inventory in a warehouse. Adapted to this situation, the principle states that 80% of the demand is generated by 20% of the products (De Koster et al. 2006).

closest open location storage because of its similarity with random storage. Full turnover storage is also not considered because I assume demand not to change. It will therefore yield the same results as dedicated storage.

Under the assumptions defined in the previous paragraphs, the implementation of these policies is as follows. First, for the *dedicated storage policy* each item is appointed a fixed location according to its demand characteristics. The item with the highest demand receives the location closest to the depot. The item with the lowest demand receives the location furthest away from the depot. Second, for the *class-based storage policy* I divide the storage area in three separate classes. See Petersen et al. (2004) for a discussion on the optimal division in classes. Again, I assign locations based on demand characteristics. The 33% high demand products are randomly assigned the 33% locations closest to the depot. For the other two classes the locations are also randomly assigned. Finally, the *random storage policy* entails randomly assigning locations to products. No demand characteristics are involved.

## 3.2 Orders

Orders define the work an order picker must complete. They define the articles an order picker must pick and therefore the distance he must walk. Without these orders, a bucket brigade can not be started. This section elaborates on the different properties involved in creating these orders.

Each order comprises a set of articles and their location. The set of articles is the first property of an order. The second is the length of an order or the amount of articles. I assume this second property to have a discrete exponential distribution. This assumption is not supported by other research but I believe it is a reasonable initial assumption.

The length of the order is relatively simple compared with the set of articles to appear on the order. This property is more difficult to model as it depends on the demand characteristics of the articles. I denote the demand of article $j$ as $d_j$ and the total demand of all articles as $D$. Then the probability of appearing on the order for article $j$, denoted by $p_j$, is equal to $d_j/D$. The set of $p_j$'s then define a distribution function $F$ which can be used to draw articles randomly. The process of creating an order thus consists of first determining the demand of the articles and the associated probabilities and then randomly drawing these articles according to the distribution function $F$.

The set of articles by itself is not important, it is their location that controls the distance an order picker must walk. After the order generation, the total distance defined by an order is calculated by choosing the maximum of the individual item locations. It is this property of the order that is passed on to the bucket brigade, discussed in the next section.

## 3.3 Bucket brigades

The model for the bucket brigades underlying the simulation is based on the model as developed by Bartholdi and Eisenstein (1996a) (hereafter referred to as the *BE model*). This model is explained in the next paragraphs. Subsequent to this explanation I present the adjustments I made to the BE model. Bartholdi and Eisenstein (1996a) use the term workers to refer to the persons carrying out the work. In order picking systems these persons are referred to as order pickers. I use these two terms interchangeably.

The state variable in the BE model is the position of the worker. It is expressed as the fraction of work completed by that particular worker. An example of a state with two workers is depicted in Figure 3. The state of the whole system is denoted by the vector $\boldsymbol{x}$ and consists of the individual positions of workers 1,…,$n$. For the example in Figure 3 the state of the system is expressed as $\boldsymbol{x} = (x_1,x_2)$ = (0.15, 0.65). In addition, the BE model enforces blocking. An extra restriction is therefore applied to prevent workers from passing one another. This restriction is expressed by $0 \leq x_1 \leq \ldots \leq x_n$. In Figure 3 this means the position of worker 1 will never be larger than the position of worker 2.


Figure 3: State of the bucket brigade with two workers. Worker one has finished 15% of the work and worker two is at 65% of the total work.

Each worker $i$ has a velocity $v_i$ which is assumed to be constant. This speed only incorporates walking speed, the processing time of orders is considered negligible compared with the walking time. The workers are also ordered from slowest to fastest according to their speed. The speed of the workers controls the state transitions. Such a transition means the mechanism of the bucket brigade is applied and orders are exchanged according to the description presented in Chapter 1. Bartholdi and Eisenstein (1996a) refer to the state transition as the *reset*. In the BE model each reset is initiated when worker $n$ (the last worker) finishes his order. Because the completed work is expressed as a fraction of the total work, the last worker has to complete $(1\text{-}x_n)$% of work to finish his order. It takes this worker $(1\text{-}x_n)/v_n$ time units to fulfil his task. This means that, under the BE model, a reset occurs after every $(1\text{-}x_n)/v_n$ time units.

According to the description of the bucket brigade mechanics in Chapter 1, all workers walk back to the positions of the workers behind them after the reset. The BE model assumes the walk back speed is infinite. The reset therefore does not involve extra time units. Equation (1) presents the worker positions after the reset in a mathematical representation.

$$x_i^{t+1} = \begin{cases} 0 & i = 1 \\ x_{i-1}^t + v_{i-1}\dfrac{(1 - x_n^t)}{v_n} & 1 < i \le n \end{cases} \tag{1}$$

The position of worker $i$ directly after the reset is denoted by $x_i^{t+1}$. As worker $i$ must exchange orders with the worker behind him, worker $i$-1, the position of worker $i$ after the reset is dependent on two properties. First there is the position of worker $i$-1 right after the previous reset, denoted by $x_{i-1}^t$. The second property is the distance walked by worker $i$-1 until the current reset. This distance equals worker $i$-1's velocity $v_{i-1}$ multiplied by the time required by worker $n$ (the last worker) to finish his order. The last worker is important because he initiates the resets.

In the case worker $i$ is the first worker, he must walk back to the depot to start a new order. His position is therefore reset to zero. To preserve the ordering of the workers, the BE model also assumes workers can not pass one another. In this case (1) must be adjusted to include blocking. This changes the formula as follows.

$$x_i^{t+1} = \begin{cases} 0 & i = 1 \\ \min\left\{ x_i^t,\quad x_{i-1}^t + v_{i-1}\dfrac{\left(1 - x_n^t\right)}{v_n} \right\} & 1 < i \le n \end{cases} \tag{2}$$

The difference with (1) is that in (2) worker $i$ may be blocking worker $i$-1. Worker $i$-1 can therefore not pass the position of the worker in front of him. This position is denoted by $x_i^t$. The minimum is therefore taken over the positions of worker $i$ and worker $i$-1 right before the reset. Again, if worker $i$ is the first worker, his position is reset to zero.

The BE model implicitly assumes orders to pass all workers and that the normalized work content is the same for each order. For order pickers in practice this assumption does not always apply. Orders can be finished before ever reaching the last order picker. Incorporating this into the BE model results in a new specification of the state variable. Instead of the fraction of completed work as the state variable ($\boldsymbol{x}$ in the BE model), I use the distance walked by the order pickers denoted by $\boldsymbol{y}$. The total work of an order then equals the total walking distance of this order and is represented by the variable $W$.

With these adjustments, worker $n$ no longer controls the resets. Instead I refer to worker $r$ as the worker who finishes his order first, with $1 \le r \le n$. The position of this worker after the last reset equals $y_r$. After the reset is initiated worker $r$ can only exchange orders with the workers behind him. The workers in front of him are therefore not affected.

$$y_i^{t+1} = \begin{cases} 0 & i = 1 \\ \min\left\{ y_i^t, \quad y_{i-1}^t + v_{i-1} \dfrac{\left(W - y_r^t\right)}{y_r} \right\} & 1 < i \le r \end{cases} \tag{3}$$

The position of worker *i* is still dependent on the same two properties as previously defined. These properties include the position of worker *i*-1 after the last reset and the distance walked by worker *i*-1 until the reset. The difference between (3) and (2) is that in (3) the distance walked by worker *i*-1 is calculated differently. To calculate this distance the velocity of worker *i*-1 is multiplied by the time it takes for worker *r* to finish his order. Analogous to (2), the first worker is reset to zero.

Equation (3) includes the possibility of blockages. However, I will first allow the order pickers to pass one another. This is more realistic because in warehouses most aisles are wide enough for passing to occur. Workers will therefore rarely wait behind another worker when they are able to pass. I will also implement the model with blockages to examine its impact on the results.

## 3.4 Inputs and outputs

In this section, the inputs required for the simulation are presented. This section also presents the outputs the simulation will produce.

<u>Inputs</u>:

- The demand of the products follows a discrete *Pareto* distribution with a shape equal to one and a location equal to 1/25 times the number of articles. This ensures the distribution is in accordance with the Pareto principle as referred to in Section 3.1.

- The length of an order follows a discrete exponential distribution with an average of 10. I find this average is a reasonable assumption.

- Number of order pickers. I perform the simulations with five pickers as I believe that is a reasonable assumption.

- Velocities of the order pickers. I assign the first picker a speed of one, the second picker a speed of two, the third picker a speed of three, etc. This is in line with the assumption that the pickers have constant speeds. It also ensures the faster pickers are faster than the other pickers the entire time.

- Number of articles. I implement the storage assignment policies and the bucket brigade based on 5000 articles. I believe this is a reasonable assumption.

The basis scenario of inputs with which I perform the simulation thus comprises 5000 articles, an average order length of 10 and 5 workers.

Outputs:

- Average time to completion of an order (order throughput time)
- Percentage idle time of a worker caused by blockings
- Distance walked by a worker
- Hand-over points for each worker. These are the points where workers exchange orders
- Number of completed orders for each worker

The first output is the main criteria to assess the performance of the bucket brigade. Bartholdi and Eisenstein (1996a) claim their method minimizes this measure. I therefore use this measure to compare the impact different storage assignment policies have on the bucket brigades. The second output is a secondary measurement of the performance and is employed to examine the impact of blockages. Finally, the third, fourth and fifth outputs examine the balancing property of the bucket brigades.

Bartholdi and Eisenstein (1996a) also claim that next to the minimized throughput time, a bucket brigade adhering to the assumptions of the *Normative Model* also balances workload. According to De Koster at al. (2006) a balance of the workload is necessary to prevent "deterioration of order throughput and order throughput time." The balance can however be measured by several methods. Bartholdi and Eisenstein (1996a) state that when balance is achieved the hand-over points are fixed. The pickers then return to the same point after each reset. The hand-over points therefore function as the first measure of balance. Fixed hand-over points also mean the distance a picker walks is fixed. The distance walked by a picker is thus the second measure of balance. Finally, I define *workload* as the average distance per order divided by the speed of a picker. Workload values which are equal for all pickers are then the third measure of balance.

# 4 Results

The objective of an order picker system is to maximize order throughput or minimize order throughput time. A bucket brigade achieves this objective by imposing a set of rules on the order picking system as explained in Section 2.2 and Section 3.3. One of these rules regards the requirement for a continuous and uniform spreading of work content. Of the three storage assignment policies, the random storage policy is the only policy which complies with this requirement. The other storage policies do not result in a uniform spreading of work content. The extent to which this affects the bucket brigades is the subject of this paper.

In this section, the results are presented of implementing the bucket brigade in combination with three storage assignment policies. This implementation is performed in a simulation environment where article demand, article location and order length are simulated. First, the bucket brigade is implemented in an environment where pickers are allowed to pass one another. The results of this implementation are presented in Section 4.1. Second, an implementation is also performed for the case where pickers are not allowed to pass one another. Section 4.2 presents the results of this case.

## 4.1 Passing of order pickers

Bartholdi and Eisenstein (1996a) did not allow order pickers to pass one another to preserve the ordering of the pickers from slowest to fastest. However, this assumption is not realistic in warehouses where passing is possible and pickers are not likely to wait behind another picker if they can pass one another. This section therefore presents the results for implementing the bucket brigade which allows pickers to pass one another.

However, because processing time is assumed to be zero the time a picker is at work only consist of walking time. The faster pickers can therefore pass the slower pickers but not vice versa. A slower picker can never pass a faster picker because he can never catch up with the faster picker. As the faster pickers pass the slower pickers the bucket brigade will eventually converge to an ordering where the slowest picker is closest to the depot and the fastest picker is furthest away. This means the bucket brigade which allows pickers to pass one another will resemble the ordering of Bartholdi and Eisenstein (1996a). Certain results may therefore resemble the findings of Bartholdi and Eisenstein (1996a).

Table 1 presents the results for the order throughput time of the different storage assignment policies. These are obtained using the basis scenario, as explained in Section 3.4.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 1421 |
| Dedicated | 465 |
| Class-based | 637 |

Table 1: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario

The results of Table 1 show the dedicated storage policy achieves the minimum throughput time. This suggests the bucket brigade performs best under this storage assignment policy. For the random and class-based storage policies the throughput time is higher because these policies randomly assign locations to articles. Articles with a high demand may therefore be assigned a location far away from the depot, increasing the time for an order picker to retrieve this article. This problem does not occur for the dedicated storage policy because high demand products are assigned to locations near the depot. The class-based storage policy also places high demand articles near the depot but does this randomly. Therefore, there is still a chance a high demand article is placed far away from the depot and throughput time is increased.

Aside from the throughput time, I also examine the balancing property of bucket brigades as explained in Section 3.4. Table 2 presents the average distance walked by an order picker per order and the workload this represents. The workload is calculated by dividing the average distance per order by a picker's speed. A distance of, for example, 100 represents a smaller workload for a picker with a high speed compared to a picker with a low speed.

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 284 | 284 | 93 | 93 | 127 | 127 |
| 2 | 572 | 286 | 216 | 108 | 256 | 128 |
| 3 | 865 | 288 | 373 | 124 | 393 | 131 |
| 4 | 1191 | 298 | 615 | 154 | 581 | 145 |
| 5 | 1562 | 312 | 1175 | 235 | 1097 | 219 |

Table 2: Average distance walked per order for each picker and the amount of workload this represents. Picker 1 is the slowest and picker 5 is the fastest

The values for the average distance per order are not surprising. The pickers with a higher speed walk more than the pickers with a lower speed. There is however a difference between the storage policies. For the random storage policy the values are higher because certain articles with a high demand are located further away. This also applies to the class-based storage policy but to a lesser extent. Compared with the dedicated case, the last pickers in the class-based case walk a shorter distance. The first couple of pickers however walk a larger distance on average. The random allocation procedure within the different classes thus has its benefits but also its disadvantages.

The average distance is in favour of the dedicated and class-based storage policies because in both cases the pickers walk a shorter average distance than for random storage. Bartholdi and Eisenstein (1996a) however developed the bucket brigade to also balance the workload of the individual workers. This means the workload values in Table 2 should be equal for all pickers. The random storage policy is the only storage policy which approximates a balanced bucket brigade as the workload values for this storage policy are not far apart.

Compared to the random storage policy, the workload values of the dedicated storage policy are further apart. In this case the last picker shows a value which is almost three times higher than the first picker. For the class-based storage policy the situation is comparable. The differences between the workload values of the last pickers are also larger compared to the random storage policy. The workload values of the first pickers however are close to one another. Nevertheless, for a balanced system all workers must have equal workload.

The values in Table 2 show that the random storage policy approximates a balanced bucket brigade with regard to workload. The reason why the workload values for the other storage policies are not balanced is not evident from Table 2. To further examine the balancing property I therefore use another approach. This approach concerns the hand-over points, the points at which the pickers exchange orders. According to Bartholdi and Eisenstein (1996a), a balanced system results in hand-over points which are fixed. In a balanced bucket brigade all pickers therefore return to the same point after each reset. Furthermore, if hand-over points exist which are frequently visited by a picker, this also means a distance exists which is walked by a picker most often.

In the next paragraphs, the hand-over points are presented in histograms for each storage policy and the middle and the last picker under the basis scenario. The histograms for the other pickers are presented in Appendix 7.1. The first policy discussed is the random storage policy.

**Random storage**

The random storage assignment policy randomly assigns locations to articles. It therefore complies with the third assumption of the *Normative Model*. This assumption states that the spreading of work content should be continuous and uniform. As explained in Chapter 3, work is defined by the total walking distance required by an order.

The compliance of the random storage policy with the third assumption explains why the workload values in Table 2 are almost equal. The balance achieved by applying the random storage policy should therefore also be reflected in the hand-over points. Figure 4 presents the hand-over points of the last picker with the picker behind him and the points at which an order is completed by the last picker.

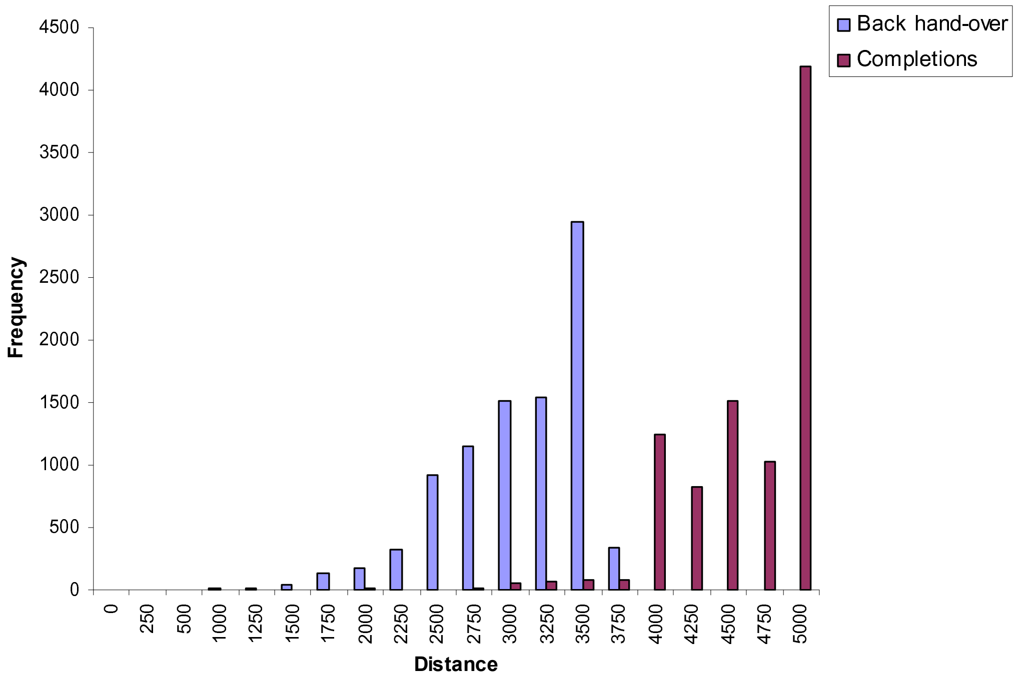


Figure 4: Points at which the last picker exchanges orders with the picker behind him (back hand-over) and points at which the last picker is finished (completions) with the random storage policy

Figure 4 shows there are two distinct points frequently visited by the last picker. This picker exchanges orders with the picker behind him most often around the 3500 mark. The last picker then finishes the orders when the interval 4750-5000 is reached most of the time, as shown by the large number of completions within this interval. The high frequency of the two identified intervals does not guarantee the last picker also walks an approximately fixed distance; a property required for a balanced bucket brigade. For this reason, I also view the distances walked. Figure 5 presents the frequencies of the distances walked by the last picker.

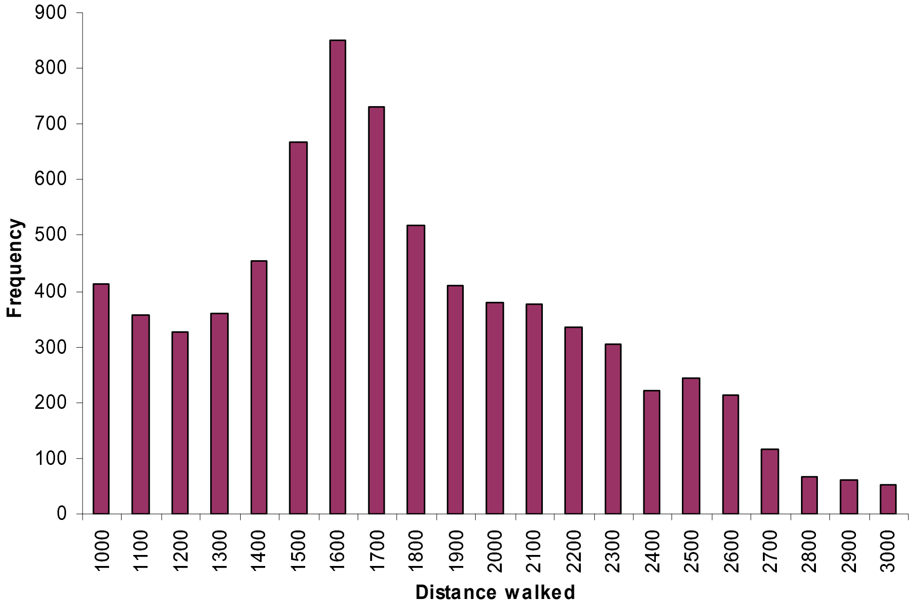


Figure 5: Frequency of the distance walked by the last picker in a bucket brigade with the random storage policy

The histogram in Figure 5 shows a large portion of the distances walked by the last picker is confined to the interval 1500-1700. The variation in these distances is however also large, evidenced by the relatively high frequencies of the other intervals. It is therefore not possible to conclude that the last picker walks a fixed distance. This does not mean the bucket brigade can not balance workload. The values in Table 2 still show the workload of the pickers other than the last picker is almost equal. The slightly higher workload of the last picker is then explained by Figure 5 and Figure 4. The variation in completion points is small, most of these points are confined to the interval 4750-5000. Nevertheless, the variation is apparently large enough to cause the last picker to walk different distances after each reset. This variation is what Bartholdi and Eisenstein (1996b) referred to as "randomness of the orders".

The workload values of Table 2 suggest that this variation may be restricted to the last picker only, increasing only his workload. To verify this statement, Figure 6 depicts the hand-over points of the third (middle) picker. The hand-over points of this picker are split into back hand-over and front hand-over points. The points at which the middle picker exchanges orders with the picker in front of him are the front hand-over points. The back hand-over points are then the points at which the picker exchanges orders with the picker behind. Figure 6 also shows the completion points of the middle picker.

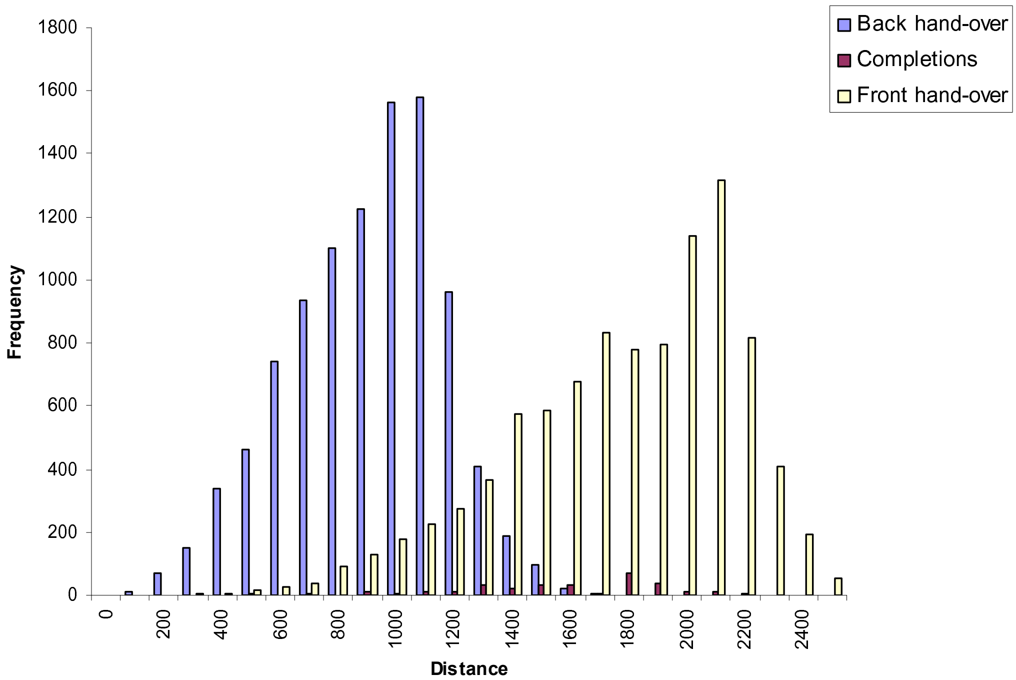


Figure 6: Points at which the middle picker exchanges orders with the picker behind him (back hand-over), the picker in front of him (front hand-over) and points at which the middle picker is finished (completions) with the random storage policy

Compared with Figure 4, the peaks in the frequency of the hand-over points are less distinct for the middle picker. The middle picker exchanges orders with the picker behind him most often within the interval 900-1100. The intervals before the 900-1100 interval however also contain a large amount of hand-over points. On the other hand, the points at which the middle picker exchanges orders with the picker in front of him are within the interval 1900-2100. This interval does show a relatively higher frequency than the intervals surrounding it.

The two identified intervals again do not guarantee the distance walked by the middle picker is fixed or at least confined to a certain interval. Figure 7 therefore depicts the frequencies of the distances walked by the middle picker.

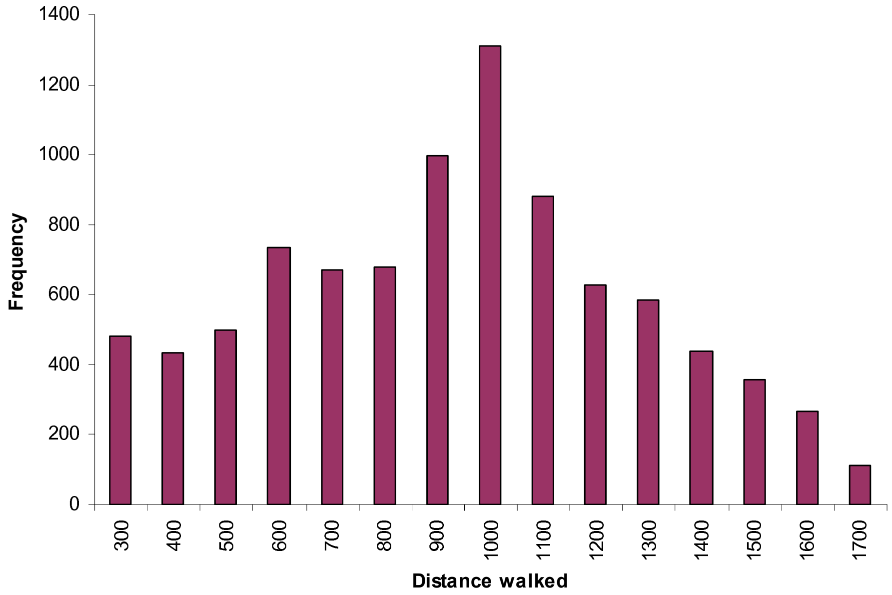


Figure 7: Frequency of the distance walked by the middle picker in a bucket brigade with the random storage policy

The results are similar to Figure 5. The distances walked by the middle picker can be confined to a certain interval but the overall variation in distances is large. This contradicts the hypothesis that the last picker safeguards the other pickers from the variation in the completion points. This variation also appears in Figure 7 and makes it less likely the middle picker walks a fixed distance. However, the distances walked most often are within the interval 900-1100. Despite the variation in these distances, the workload values of Table 2 point to a balanced system. This suggests that the intervals and variations found in Figure 5 and Figure 7 are not large. The bucket brigade implemented with the random storage policy is able to handle the variation caused by the different total walking distances of the orders.

The role of the "randomness of the orders" identified by Bartholdi and Eisenstein (1996b) is therefore limited. These results also suggest that fixed hand-over points and fixed walking distances are not a necessary requirement as long as these points and distances can be largely confined to a certain interval.

The main reason why the bucket brigade with the random storage policy approximates a balanced system is because of the uniform spreading of work content. To further emphasize the compliance of the random storage policy to the assumptions of the *Normative Model*, Table 3 presents the number of completed orders of each picker.

| Picker | Random storage |
|---|---|
| 1 | 55 |
| 2 | 83 |
| 3 | 311 |
| 4 | 452 |
| 5 | 9099 |

Table 3: Number of completed orders for each picker when random storage is applied

Bartholdi and Eisenstein (1996a) assumed the last order picker always completes the order, but I allowed for other pickers to complete the order as well (Section 3.3). For the random storage policy this assumption does not make a difference because the order is almost always completed by the last order picker. This storage policy therefore complies with all the assumptions of the *Normative Model* (Section 2.3), except for one restriction. This restriction, which enforces the blocking of order pickers, is implemented in Section 4.2. Nevertheless, the conclusion is that a bucket brigade combined with a random storage assignment policy is able to achieve a balanced workload.

### Dedicated storage

As concluded from Table 2, the dedicated storage policy does not have the balancing property. An explanation for this result is provided in the next paragraphs by considering the hand-over points and distances walked by the last picker. Figure 8 shows the completion points of the last picker and the points at which the last picker exchanges orders with the picker behind him (back hand-over).

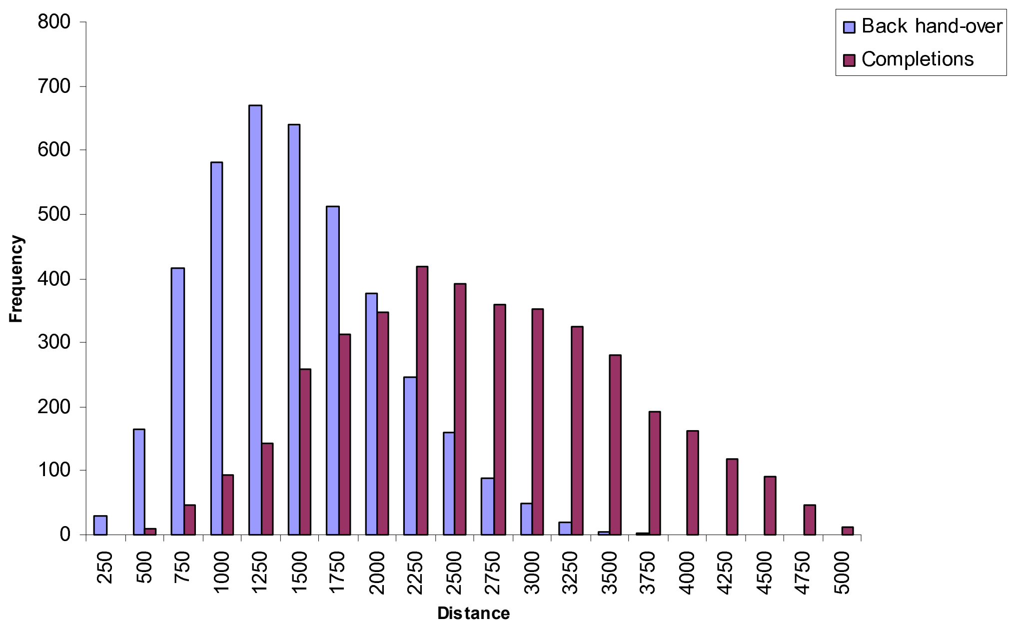


Figure 8: Points at which the last picker exchanges orders with the picker behind him (back hand-over) and points at which the last picker is finished (completions) with the dedicated storage policy

Different from random storage, dedicated storage results in fewer orders which require articles located at a large distance. The last picker therefore finishes faster in comparison to the last picker in a random storage setting. This explains the low order throughput time. The hand-over points with the picker behind the last picker also occur at a shorter distance but do not show a significant peak. Approximately 50% of these points are between 750 and 1500, which is a wider interval compared to the random storage policy.

The hand-over points suggest the last picker does not always return to the same point after each reset. The wide interval in which most of these points lie also suggest there is a large variation in the distances walked by the last picker. Figure 9 presents the frequencies of these distances for the last picker.

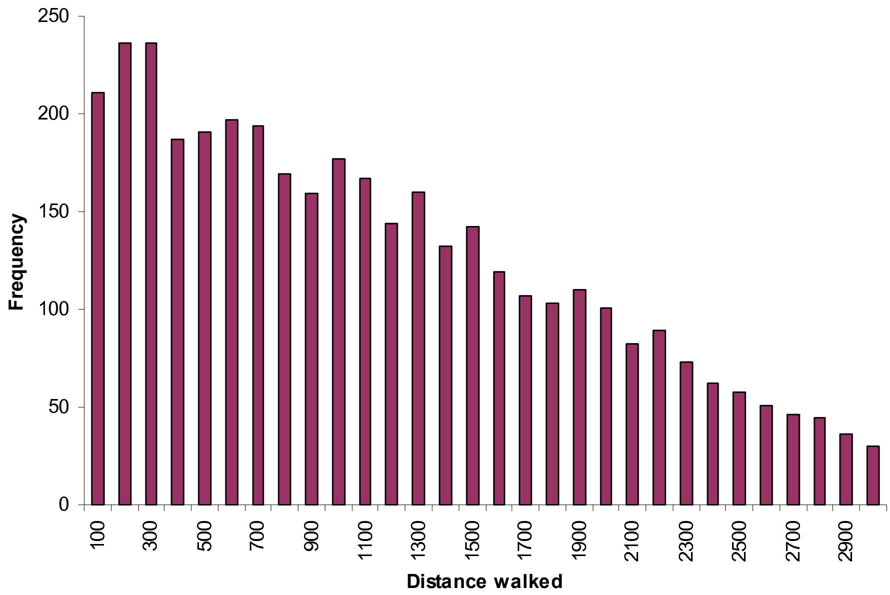


Figure 9: Frequency of the distance walked by the last picker in a bucket brigade with the dedicated storage policy

Figure 9 shows there is a large variation in the distances walked by the last picker. The bucket brigade is able to handle a certain amount of variation but the workload values in Table 2 suggest the variation caused by the dedicated storage policy is too large. Because of this variation the bucket brigade is not able to balance workload. This is not the only reason why a bucket brigade combined with the dedicated storage policy can not achieve a balance of the workload. Table 4 and Figure 10 help to explain the next reason. Table 4 presents the number of completed orders for each picker. Figure 10 presents the hand-over points for the middle picker. These hand-over points are split into the points at which the middle picker exchanges orders with the picker behind him and at which the picker exchanges orders with the picker in front of him. Figure 10 also presents the completion points.

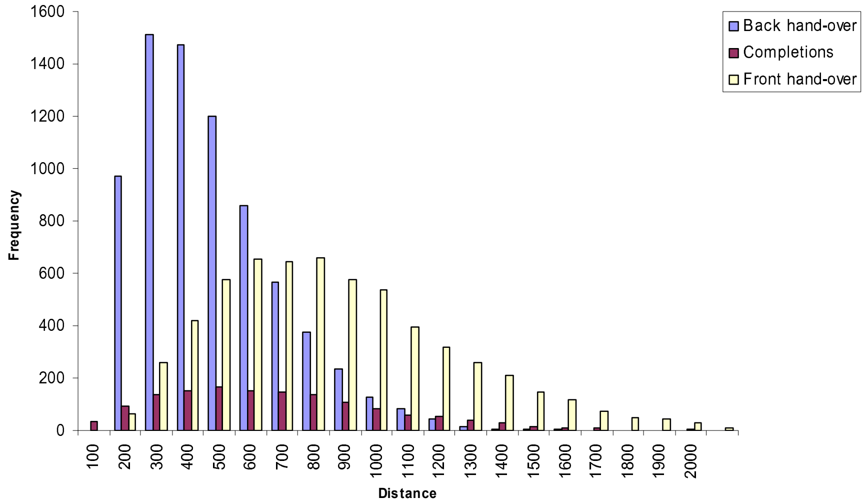


Figure 10: Points at which the middle picker exchanges orders with the picker behind him (back hand-over), the picker in front of him (front hand-over) and points at which the middle picker is finished (completions) with the dedicated storage policy

| Picker | Dedicated storage |
|---|---|
| 1 | 1395 |
| 2 | 1132 |
| 3 | 1419 |
| 4 | 2095 |
| 5 | 3959 |

Table 4: Number of completed orders for each picker when dedicated storage is applied

The variation in completion points does not only originate from the last picker. The other pickers also complete a large number of orders. For example, Figure 10 and Table 4 show the middle picker (#3) completes 1419 orders. Each time the middle picker completes an order, the bucket brigade is reset. However, only the pickers behind this picker are affected. This distorts the balance in workload because the position of pickers 4 and 5 is not reset. They therefore walk a certain amount of distance which is spread over fewer orders compared to the pickers whose position is reset.

The advantage of the dedicated storage policy is the allocation of high demand products near the depot. This generates orders with a shorter total walking distance and allows the first pickers to complete a large number of orders. This is an advantage because it decreases order throughput time. It is however also a disadvantage because it distorts the workload balance as it exposes the bucket brigade to the “randomness of the orders” as identified by Bartholdi and Eisenstein (1996b). A bucket brigade combined with dedicated storage is therefore not capable of achieving a balanced system when pickers are allowed to pass one another.

**Class-based storage**

Finally, the bucket brigade is implemented with the class-based storage policy. This policy shares characteristics with both the dedicated storage and random storage policies. The results of Table 2 show that the workload values for the class-based storage policy are not balanced, similar to the case for the dedicated storage policy. Class-based storage therefore has more in common with dedicated storage than with random storage. In the next paragraphs the hand-over points and distances walked are discussed to reveal the exact behaviour of the bucket brigade combined with class-based storage. Figure 11 shows the completion points of the last picker and the points at which the last picker exchanges orders with the picker behind him (back hand-over).

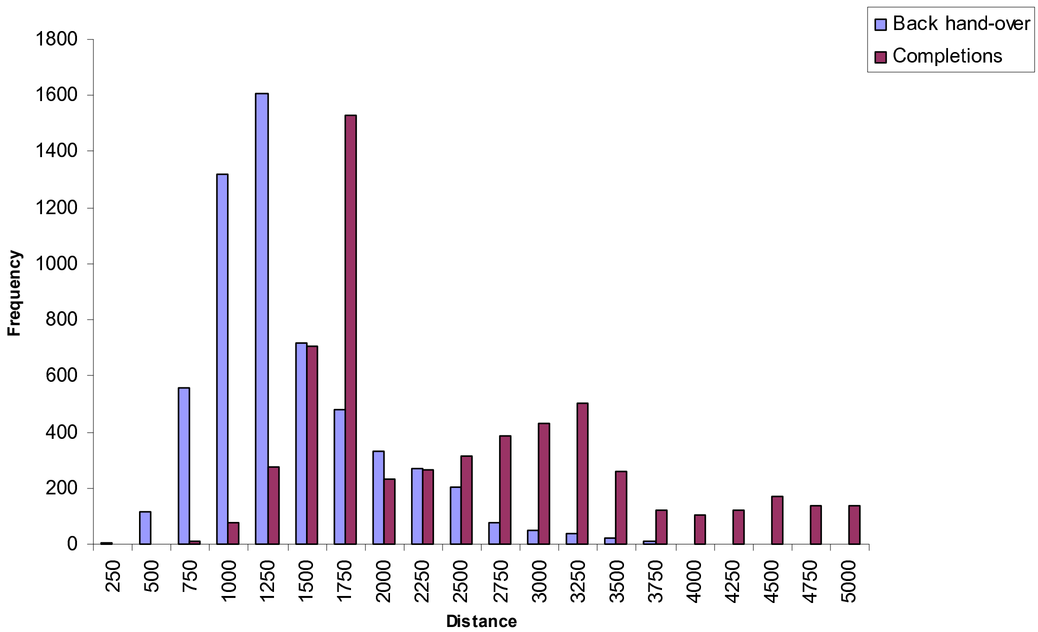


Figure 11: Points at which the last picker exchanges orders with the picker behind him (back hand-over) and points at which the last picker is finished (completions) with the class-based storage policy

As explained in Section 3.1, the class-based storage policy splits the articles into three separate classes based on the demand distribution. This distribution is the Pareto distribution in cooperation with Pareto's principle, which states that 20% of the articles are responsible for 80% of the demand. Class-based storage commits these 20% to the first class as they belong to the high demand articles. For the basis scenario, which considers 5000 articles, the first class comprises of 1667 high demand articles (5000 divided by 3 and rounded upwards). Furthermore, the locations assigned to the articles in the first class are between 0 and 1667. The second class receives the locations between 1668 and 3334 and the third class receives the locations between 3335 and 5000.

This division of the locations and the properties of the demand distribution are reflected in Figure 11. The significant peak in the completion points at the interval 1500-1750 reflects the high demand of the articles in the first class. The smaller frequencies in the subsequent intervals between 2000 and 3250 reflect the demand of the articles in the second class. Finally, the low frequencies in the final intervals signify the low demand of the articles in the third class.

The peak in the completion points also signifies the last picker walks most often to this distance and no further. The consequence for the other pickers is that they also do not walk beyond the 1500-1750 interval. As the other pickers are confined to the articles of the first class, the hand-over points in Figure 11 are concentrated on the left side of the figure. Several completion points however lie beyond 1750, which may cause outliers in the distances walked by the last picker. Figure 12 presents the frequencies of these distances.

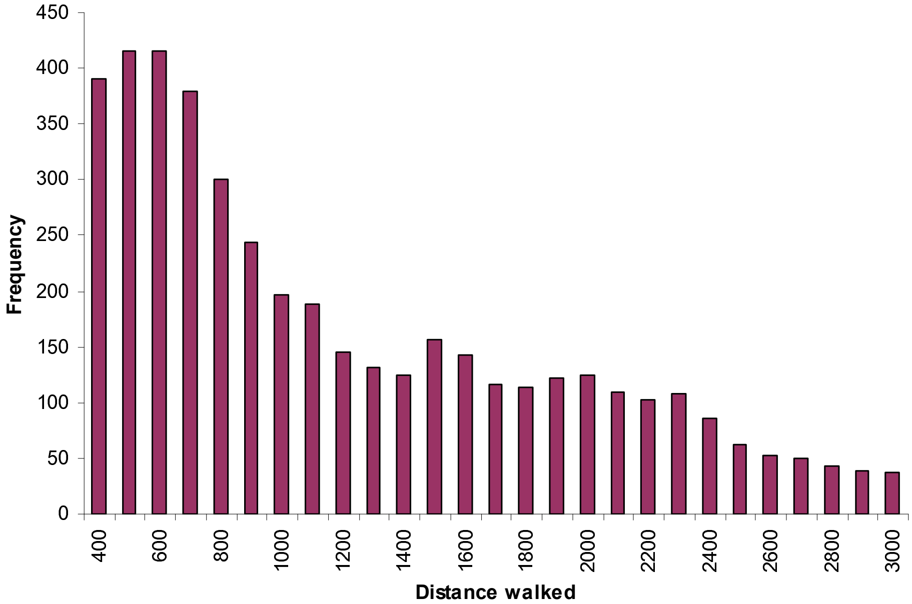


Figure 12: Frequency of the distance walked by the last picker in a bucket brigade with the class-based storage policy

The outliers in the completion points do indeed cause several outliers in the distances walked. These outliers are the cause for the large difference in workload between the last picker and the fourth picker in Table 2. They therefore prevent the bucket brigade to balance the workload. However, the workload values of Table 2 also suggest that the workload of the first couple of pickers is balanced. This is because of the division into classes and the demand distribution. As explained, the articles of the first class comprise more than 80% of the demand. In addition, the class-based storage policy randomly assigns locations to these articles. As the last picker prevents the other pickers from walking beyond the first class their situation is therefore comparable to a random storage setting. Their workload can thus be equal while the last picker accounts for the workload associated with products that are located further away.

To examine the situation for the middle picker, Figure 13 presents the hand-over points of the middle picker. Next to the back and the front hand-over points, Figure 13 also depicts the completion points of the middle picker.

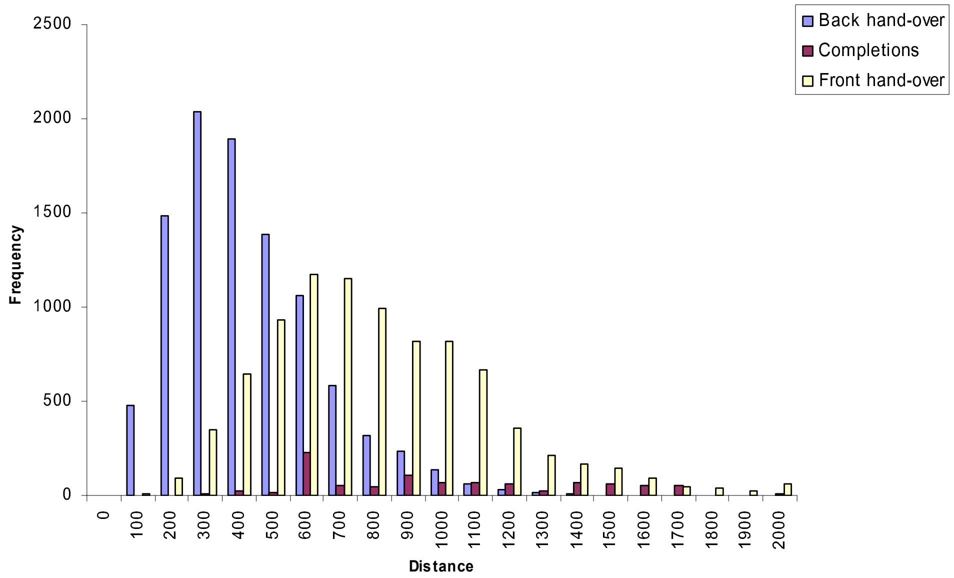


Figure 13: Points at which the middle picker exchanges orders with the picker behind him (back hand-over), the picker in front of him (front hand-over) and points at which the middle picker is finished (completions) with the class-based storage policy

Although the last picker handles most of the orders for articles in the second and third classes, the variation this causes in the completion points and hand-over points is passed down to the other pickers. Figure 13 shows the front hand-over points are confined to a larger interval than the back hand-over points. The results for the middle picker thus resemble the results for the last picker. The distances walked will therefore also resemble the distances walked by the final picker.

Despite the inability of class-based storage to approximate a completely balanced system, like random storage, it is able to achieve a higher order throughput time than random storage. That is the advantage of the division into three classes. The disadvantage is nevertheless that workload is not balanced. This is analogous to the dedicated storage policy, which also has a low order throughput time but no balance of the workload. The random storage policy therefore is the only storage policy which can approximate a balanced system. Its throughput time is however the highest of all policies.

## 4.2 Blocking of order pickers

It may not be a realistic assumption to allow blocking to occur in a warehouse. When a picker can pass another picker he may find it illogical that he must wait. Nevertheless, Bartholdi and Eisenstein (1996a) implemented this assumption to preserve the ordering of the pickers. According to Bartholdi and Eisenstein (1996a) the ordering from slowest to fastest is the ordering which maximizes throughput and achieves a balanced system.

As explained in Section 4.1, the picker ordering converges to a slowest to fastest ordering when blocking is not implemented, regardless of the initial ordering of the pickers. To obtain a similar ordering for the bucket brigade with blocking, the initial ordering must be from slowest to fastest. However, this will not yield different results than in Section 4.1 because no blocking will occur. A picker who is constantly slower than the picker before him will never be blocked.

To still be able to reveal the impact of blocking I therefore order the pickers in reverse order, from fastest to slowest. I believe this is the ordering which produces the largest amount of blocking and therefore will reveal the upper bound on the impact of blocking. This approach is different from Bartholdi and Eisenstein (1996a) and other papers, because the objective is not to achieve a balanced system but to reveal the impact of blocking. If the assumptions of Bartholdi and Eisenstein (1996a) are correct, the implemented bucket brigades should not achieve a balance of the workload.

The organization of this section is similar to Section 4.1. Table 5 shows the order throughput times for the three storage assignment policies.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 2177 |
| Dedicated | 547 |
| Class-based | 743 |

Table 5: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario, with blocking and a reverse ordering of the pickers

The largest change in order throughput time, compared with Table 1, belongs to the random storage policy. Not allowing order pickers to pass one another when they are ordered from fastest to slowest, increases the order throughput time by 50% for random storage. On the other hand, for both dedicated and class-based storage the order throughput time does not increase by more than 20%. Furthermore, the differences between the storage policies do not change. Dedicated storage still has the lowest order throughput time and random storage the highest.

Table 6 and Table 7 help to explain the increase in throughput time when blocking is allowed. Table 6 presents the number of completed orders for each picker and each storage policy. The number of completed orders does not represent the total amount of work for a particular order picker. It only shows how many orders are passed to the next picker. For example, if the last picker were to complete 20 orders, this would mean these 20 orders have been passed through the entire flow line. Each picker has thus worked on these 20 orders. Table 7 shows the percentage time a picker was blocked, also for each policy. Picker 1 is the last and slowest picker because the pickers are ordered from fastest to slowest and are not allowed to pass one another.

| Picker | Random | Dedicated | Class-based |
|---|---|---|---|
| 1 | 4613 | 1306 | 1653 |
| 2 | 2095 | 1339 | 1964 |
| 3 | 1499 | 1607 | 2555 |
| 4 | 1103 | 2052 | 2380 |
| 5 | 690 | 3696 | 1448 |

Table 6: Number of completed orders for each picker when random storage, dedicated storage or class-based storage is applied. Picker 1 is the last and slowest picker and picker 5 is the first and fastest picker

| Picker | Random | Dedicated | Class-based |
|---|---|---|---|
| 1 | 0% | 0% | 0% |
| 2 | 54% | 33% | 31% |
| 3 | 48% | 27% | 26% |
| 4 | 39% | 20% | 21% |
| 5 | 24% | 11% | 13% |

Table 7: Percentage idle time for each picker and storage policy. Picker 1 is the last and slowest picker and picker 5 is the first and fastest picker

For the random storage policy the order often passes to the last picker (picker 1 in Table 6). In addition, orders have a higher chance of requiring an article further away from the depot when random storage is applied. The last picker therefore has to walk further to complete the order and because his speed is lowest, the order throughput time increases. These throughput times are lower for dedicated and class-based storage, compared with random storage, because the pickers who are faster complete a larger portion of the orders. This is possible because articles with a high demand are located close to the depot. Most of the completion points for orders generated under dedicated or class-based storage therefore lie close to the depot. The faster pickers can reach these points without encountering another picker and can therefore complete the order.

The results also confirm the findings of Bartholdi and Eisenstein (1996a), the ordering of the pickers from slowest to fastest produces the best order throughput time as shown in Table 1. In addition, the ordering is more important for random storage than for dedicated and class-based storage. Table 7 suggests more blocking occurs for random storage, while the idle time of the faster pickers is lower for dedicated and class-based storage. This is also because the orders for random storage require a large distance most of the time. On the other hand, the orders for the other storage policies can be finished at a shorter distance by the faster pickers. They are therefore not blocked by the slower pickers.

However this does have consequences for the workload of the individual pickers in the bucket brigade. Table 8 shows the average distance walked per order and the associated workload for each picker and each storage assignment policy.

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 944 | 944 | 839 | 839 | 899 | 899 |
| 2 | 841 | 421 | 621 | 310 | 629 | 314 |
| 3 | 912 | 304 | 591 | 197 | 567 | 189 |
| 4 | 1105 | 276 | 580 | 145 | 586 | 146 |
| 5 | 1487 | 297 | 505 | 101 | 682 | 136 |

Table 8: Average distance walked per order (Average) for each picker and the amount of workload this represents for the random, dedicated and class-based (class) storage policies. Picker 1 is the last and slowest picker and picker 5 is the first and fastest picker

These results are different from Table 2 in a number of aspects. First, the workload values of the pickers in the random storage policy are not balanced. This again confirms the findings of Bartholdi and Eisenstein (1996a) which state a balance is only achieved when the pickers are ordered from slowest to fastest. The workload values for the other storage policies are also not balanced which means that when blocking is enforced the conclusions for these policies derived from Table 2 are not changed.

Second, for the dedicated and the class-based storage policies average distance walked is the highest for picker 1. This is different from Table 2, where picker 1 has the lowest average distance and picker 5 has the highest average distance. This difference is caused by blocking and the reverse ordering of the pickers. The blocking prevents the first pickers, who are the fastest, to walk the distance which is within their capacity. This also causes their workload to be lower compared to the last picker. Furthermore, the first pickers complete a large number of orders in the case of dedicated and class-based storage as shown in Table 6. This means that fewer orders reach the last picker (picker 1) and his distance walked is therefore divided by a smaller number. This number is larger for the first pickers which contributes to their low average distance.

This situation does not occur for random storage because most of the orders require a large distance to be walked. The faster pickers must therefore either walk this distance or exchange the order with the picker in front of them. Either way, the faster pickers walk a larger distance than in the case of dedicated or class-based storage. The average distance of picker 5 is therefore higher than the average distance of picker 1, similar to Table 2

Third, the workload values of Table 8 are in a different sequence compared to Table 2. This third aspect is different from the second because it also applies to the random storage policy. Blocking does not allow pickers to pass one another. Therefore, when the last (slowest) picker exchanges orders with the picker behind him he must walk the total distance required by the order because he is the last picker. Because of his low speed this represents a large amount of workload. For the random storage policy this is the primary cause of the imbalance. For the other storage policies the smaller average distances of the faster pickers also cause the workload values of Table 8 to differ from the values of Table 2.

To further explore the impact of blocking on the bucket brigade, the hand-over points are examined and if necessary the distances walked. Analogous to Section 4.1, the random storage policy is first discussed. These hand-over points are presented for the middle and the last picker. Results for the remaining pickers are depicted in Appendix 7.2.

**Random storage**

The workload values of Table 8 for the random storage suggest the workload is not balanced. This is not surprising as the ordering of fastest to slowest no longer complies with all of the assumptions of the *Normative Model* (Section 2.2). This imbalance should be reflected in the hand-over points of the last picker. Figure 14 presents these points and also the completion points of the last picker.

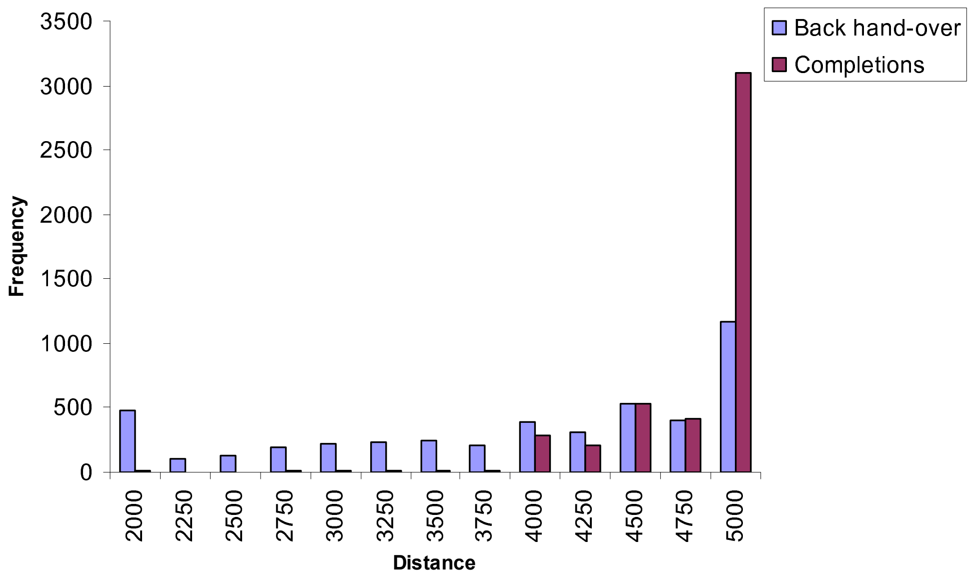


Figure 14: Points at which the last picker exchanges orders with the picker behind him (back hand-over) and points at which the last picker is finished (completions) with the random storage policy and blocking

The high frequency of the completion points in the last interval of Figure 14 suggests that most orders require a large distance to be walked. This supports the reasoning for the increased workload of the last picker, because the large distance must be walked with the slowest speed. Furthermore, because the last picker is the slowest picker, all the pickers behind him can travel a great distance before he is finished. The large number of back hand-over points in the latter intervals (above 4500) suggests the faster pickers are indeed right behind the last picker. However, there are also back hand-over points which are located further away from the end point (5000). This indicates the distances walked by the last picker also contain large values. If these values are frequent enough they may cause a large variation which could also be the reason for the imbalance of the workload.

To verify if the distances walked by the last picker have a large variation, Figure 15 presents the frequencies of these distances.

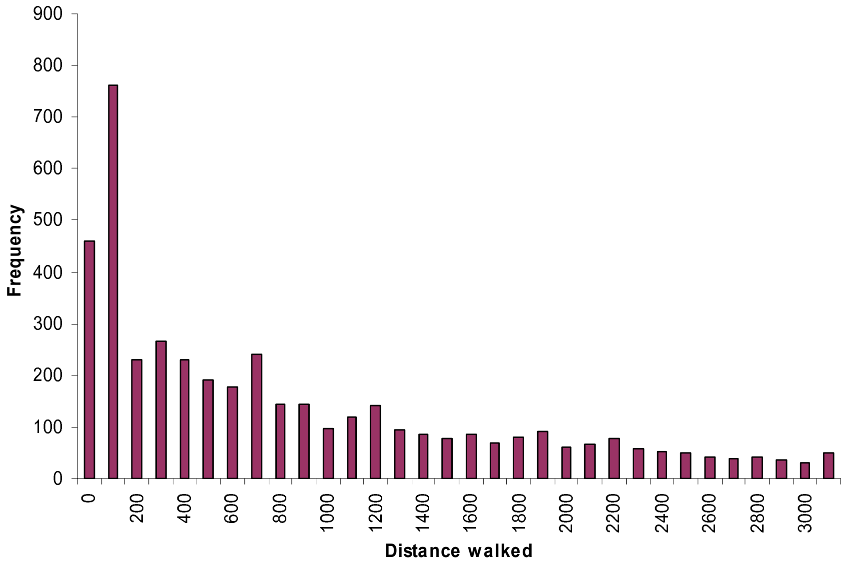


Figure 15: Frequency of the distance walked by the last picker in a bucket brigade with the random storage policy and blocking

The peaks on the left side of Figure 15 show that the distance walked by the last picker is small most of the time. The last picker always walks to the completion point of the order. The small distances of Figure 15 signify the distance between the back hand-over points and the completions points is small. This therefore confirms the hand-over points occur close to these completion points. Figure 15 also confirms that there are several outliers in the distances walked by the last picker. These large distances are caused by the large number of orders which are completed by the pickers behind the last picker. As the last picker is still in the process of finishing his order, the other pickers have already exchanged positions away from the end point (5000). When the last picker then finishes his order, the picker behind him is located far away from the end point. The hand-over point will then also be located far away from the end point. The total distance required by the order is nevertheless still near the end which means the last picker must walk a larger distance to complete the order. This causes a large variation in the distances walked which can not be handled by the bucket brigade, resulting in a large workload for the last picker.

Similar to Section 4.1, this variation may not be confined to the last picker. Figure 16 therefore presents the hand-over points of the middle picker. These include points with the picker behind the middle picker and the picker in front of the middle picker. Figure 16 also depicts the completion points.

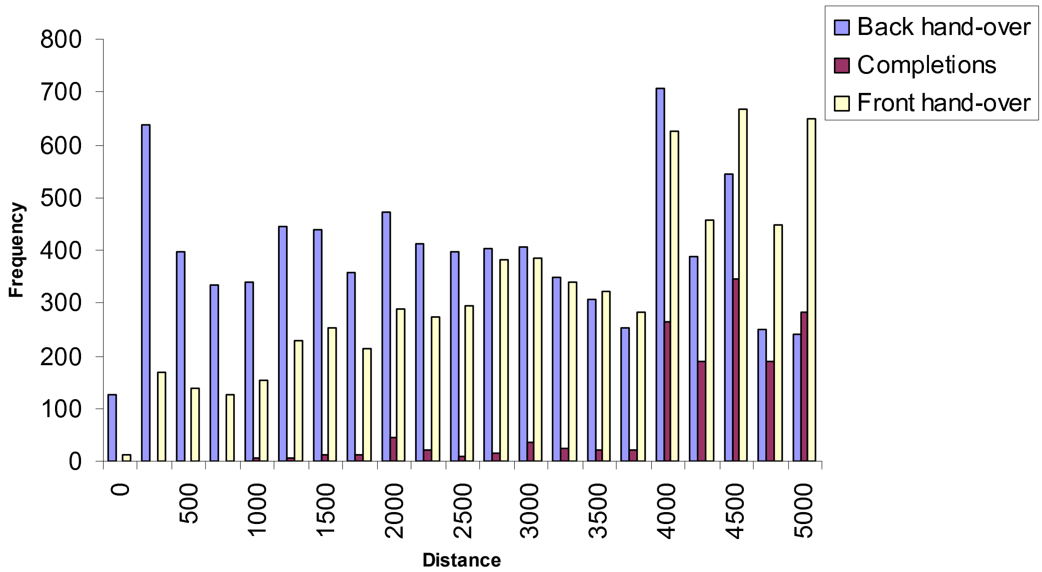


Figure 16: Points at which the middle picker exchanges orders with the picker behind him (back hand-over), in front of him (front hand-over) and points at which the middle picker is finished (completions) with the random storage policy and blocking

The hand-over points (both in front and behind) of the middle picker are distributed over almost all distances. There are two effects influencing these points. First, the middle picker is blocked by the picker in front of him and blocks the picker behind him. Whenever a reset is then initiated by a picker in front of the middle picker, the front and back hand-over points occur at the same location. These points therefore follow the same pattern as the back hand-over points of the last picker. Second, the high number of completions by the middle picker and the pickers in front also cause resets. These two effects prevent the bucket brigade to achieve a balanced system and cause the hand-over points to occur at random distances.

The results presented for the random storage policies are in line with the findings of Bartholdi and Eisenstein (1996a). When the order pickers are ordered from fastest to slowest and blocking is enforced, it is not possible to balance workload. Furthermore, the throughput time is higher compared to the situation in which the pickers are ordered from slowest to fastest, as in Section 4.1.

**Dedicated storage**

The results of Table 8 indicate that for the dedicated storage policy blocking does not change the conclusions derived from Section 4.1. The workload is not balanced, but the blocking does increase the order throughput time. As no balance is found the hand-over points should also be similar to the hand-over points found in Section 4.1. Figure 17 and Figure 18 present the hand-over and completion points for the last and middle picker.

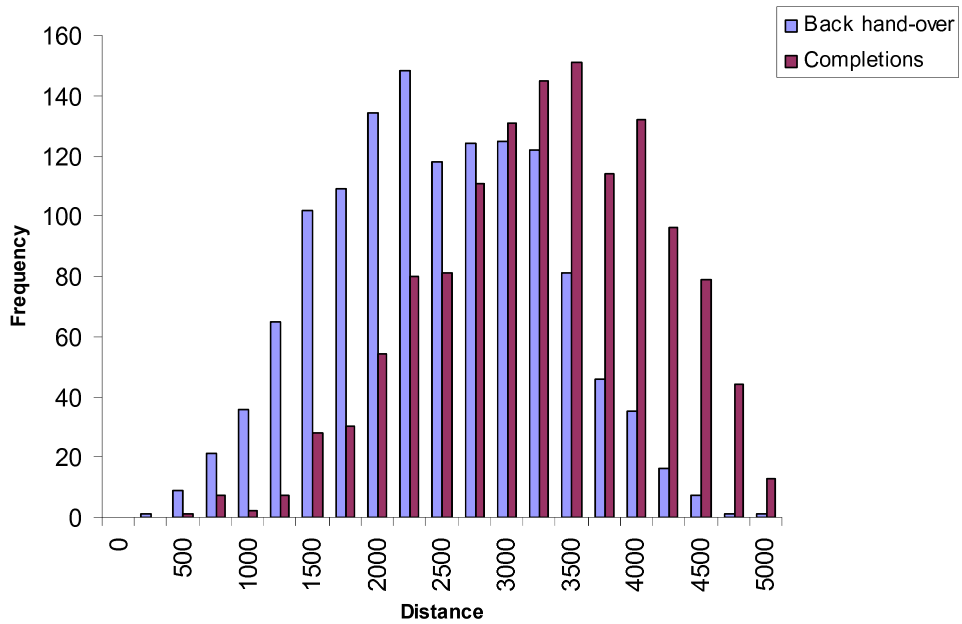


Figure 17: Points at which the last picker exchanges orders with the picker behind him (back hand-over) and points at which the last picker is finished (completions) with the dedicated storage policy and blocking

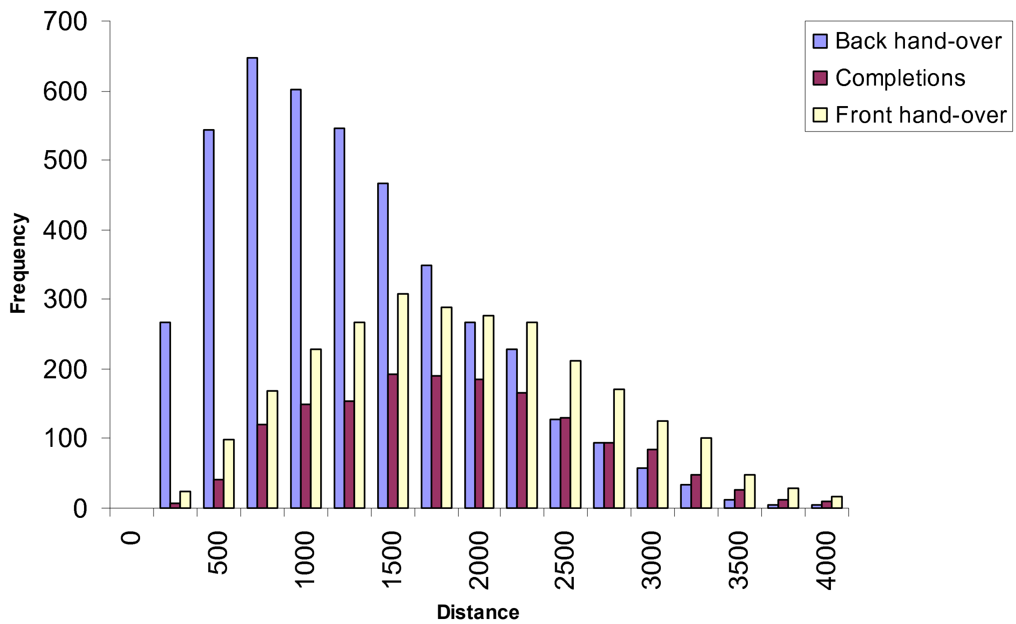


Figure 18: Points at which the middle picker exchanges orders with the picker behind him (back hand-over), in front of him (front hand-over) and points at which the middle picker is finished (completions) with the dedicated storage policy and blocking

The values in Figure 17 and Figure 18 are different from the values in Figure 8 and Figure 10. However, the overall shape of the histograms is comparable. The difference in values is caused by the larger number of early completions by the first couple of pickers. This property also prevents these faster pickers from being blocked and thus limits the impact of blocking. That was apparent from the values for dedicated storage in Table 6 and Table 7 and it is confirmed by Figure 17 and Figure 18.

**Class-based storage**

In Section 4.1 it was revealed that the behaviour of the bucket brigade combined with class-based storage is dominated by its commonalities with the dedicated storage policy. This was however in an environment where pickers were allowed to pass one another. Nevertheless, Table 8 shows that a balance of the workload is also not obtained when pickers are blocking one another. This imbalance should be reflected in the hand-over points of the order pickers. Figure 19 presents these hand-over points for the last picker with the completion points of this picker.

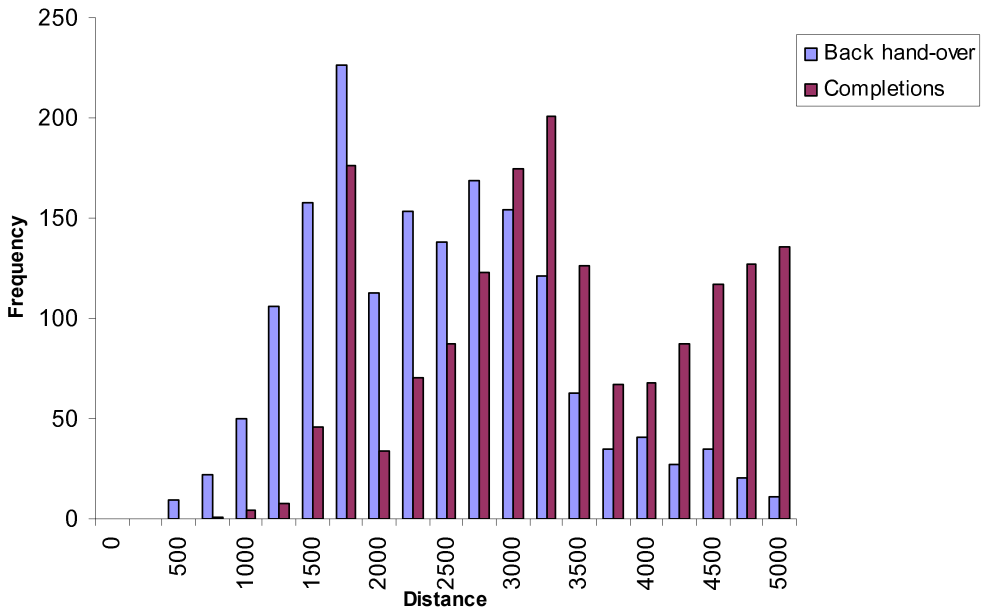


Figure 19: Points at which the last picker exchanges orders with the picker behind him (back hand-over) and points at which the last picker is finished (completions) with the class-based storage policy and blocking

The shape of Figure 19 is comparable to the shape of Figure 11. The three classes in which the articles are divided are more visible from Figure 19 than from Figure 11. In Figure 11, the last picker completes a large number of orders within the 1500-1750 interval. However, in Figure 19 these orders form a smaller portion of the total number of completed orders. Instead, the orders completed by the last picker at a larger distance form a larger portion of the completion points. This explains the peaks at 3250 and 5000. These peaks were also present in Figure 11 but were overshadowed by the large number of completions at a shorter distance.

Another difference between Figure 19 and Figure 11 is the distribution of the back hand-over points. For Figure 11 these are less spread out than for Figure 19. This can be explained by the larger distance the faster pickers walk before the last picker is finished. The picker behind the last picker therefore catches up with the last picker and is therefore close behind him when he finishes his order. The hand-over point thus takes place nearer to the total walking distance of the order. This is not specific to the class-based storage but caused by the ordering from fastest to slowest which allows the faster pickers to catch up with the (slower) pickers in front of them.

One of these faster pickers is the middle picker. The hand-over points for this picker are presented in Figure 20 with the completion points of the middle picker.

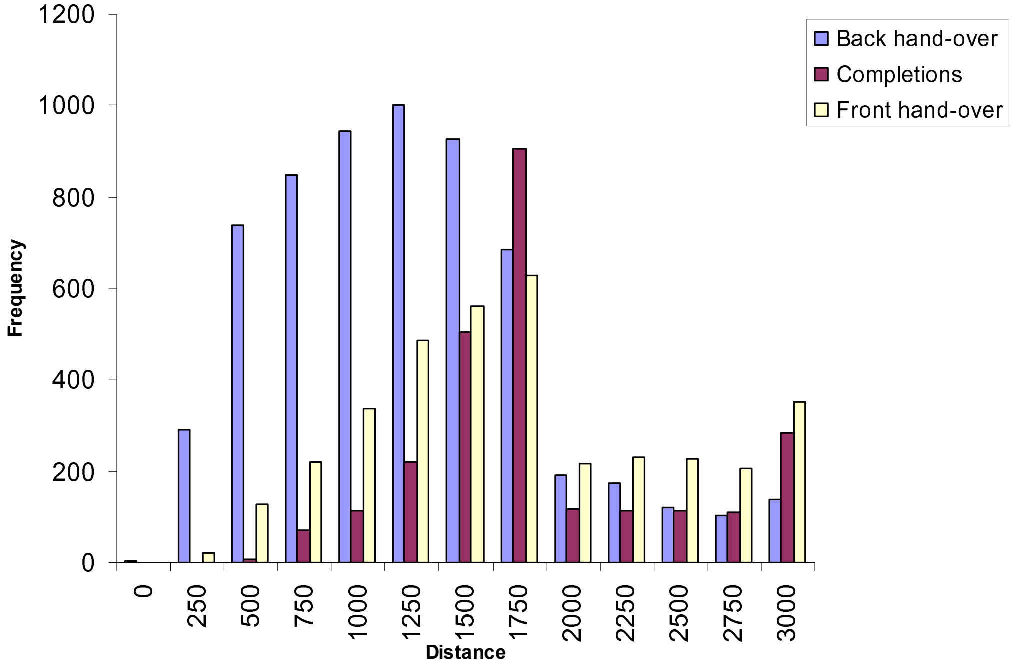


Figure 20: Points at which the middle picker exchanges orders with the picker behind him (back hand-over), in front of him (front hand-over) and points at which the middle picker is finished (completions) with the class-based storage policy and blocking

Figure 20 shows the impact of blocking as the number of completions is substantially higher compared with Figure 13. Figure 13 suggests that when there is no blocking, the middle picker almost never reaches 1750 without meeting the picker in front of him. However, Figure 20 indicates that with blocking this is not the case as the middle picker is faster than the pickers in front of him and can therefore catch up with them until he is blocked. Figure 20 also shows that the number of orders which require articles from the second and third classes is handled by the picker in front of the middle picker. This also explains the low amount of hand-over points at these distances.

Both the results for the middle picker and the last picker are different from the results in Section 4.1. The commonalities the class-based storage policy has with the dedicated storage policy limit the impact from blocking as shown by Table 7. However, the commonalities with the random storage policy expose class-based storage to the impact of blocking as is evident from Figure 19 and Figure 20. This causes the difference with the results of Section 4.1.

The findings of this section are the result of a combination of blocking and a reverse ordering. They are therefore not only caused by blocking. Nevertheless, to reveal the maximum impact of blocking a reverse ordering is needed.

# 5 Sensitivity analysis

The results of Chapter 4 are based on a number of assumptions. These assumptions, defined in Chapter 3, concern the number of pickers, the number of articles, the distribution of the order length, the picker velocities and the distribution of the demand. The last assumption is the only assumption based on a finding from academic literature and it is therefore not changed. In this section I examine the sensitivity of the conclusions of Chapter 4 to the assumptions besides the demand distribution.

This chapter is organized in a section for each assumption. In these sections the bucket brigade is implemented for each storage policy. The sensitivity analysis is performed when passing is allowed but also when blocking is enforced.

## 5.1 Number of pickers

The conclusions of Chapter 4 may be specific to the case of five pickers. It is possible that fewer or more pickers do not achieve a balance of the workload. A different number of pickers may also significantly change the differences between the storage policies. Therefore I implement the bucket brigade with two pickers and ten pickers. The analysis is first performed for an environment where pickers are allowed to pass one another.

**Passing of pickers**

Table 9 presents the throughput times for the case with two pickers. The average distance per order and the workload for each picker are presented in Table 10.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 2843 |
| Dedicated | 930 |
| Class-based | 1273 |

Table 9: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with two pickers

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 1421 | 1421 | 465 | 465 | 637 | 637 |
| 2 | 2932 | 1466 | 1419 | 710 | 1437 | 719 |

Table 10: Average distance walked per order for each picker and the amount of workload this represents for the random, dedicated and class-based storage policies. Picker 1 is the slowest and picker 2 is the fastest

The values for the order throughput time in Table 9 are higher than in Table 1 because there are fewer pickers which have to walk the same distance. Nevertheless, the differences between the storage policies are the same. Dedicated storage has the lowest order throughput time and random storage the highest. The workload values in Table 10 also do not change the conclusions derived in Chapter 4. Random storage is the only storage policy which achieves a balanced system with respect to workload. As the conclusions drawn from these results are not different from Section 4.1, I do not examine the hand-over points. For subsequent analyses the hand-over points are only examined when the conclusions are significantly different from the conclusions of Chapter 4.

The results for a bucket brigade with ten pickers are presented in Table 11 and Table 12. Table 11 presents the throughput times and Table 12 presents the average distance per order and the workload for each picker.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 775 |
| Dedicated | 254 |
| Class-based | 347 |

Table 11: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with ten pickers

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 77 | 77 | 25 | 25 | 35 | 35 |
| 2 | 155 | 78 | 55 | 27 | 70 | 35 |
| 3 | 234 | 78 | 88 | 29 | 105 | 35 |
| 4 | 313 | 78 | 124 | 31 | 140 | 35 |
| 5 | 392 | 78 | 167 | 33 | 177 | 35 |
| 6 | 475 | 79 | 220 | 37 | 217 | 36 |
| 7 | 565 | 81 | 287 | 41 | 267 | 38 |
| 8 | 662 | 83 | 386 | 48 | 342 | 43 |
| 9 | 766 | 85 | 549 | 61 | 498 | 55 |
| 10 | 944 | 94 | 983 | 98 | 1007 | 101 |

Table 12: Average distance walked per order for each picker and the amount of workload this represents for the random, dedicated and class-based storage policies. Picker 1 is the slowest and picker 10 is the fastest

The dedicated storage policy again exhibits the lowest order throughput time and random storage the highest. The workload values of Table 12 also point to random storage as the closest approximation of a balanced system. Furthermore, the similar values for the first pickers, in the case of class-based storage, are not different from the results for five pickers in Table 2. This only confirms that when the number of pickers increases the distances walked of more pickers are restricted to a single class. The articles within this class are randomly assigned a location and therefore a balanced system is possible. Nevertheless, the balance again disappears for the last pickers.

These results show the relationship of the bucket brigade with the storage assignment policies is not sensitive to the number of pickers when these pickers are allowed to pass one another. The next paragraphs present the results for an environment where blocking of pickers is enforced.

**Blocking of pickers**

Table 13 presents the throughput times and Table 14 presents the workload values for a bucket brigade with two pickers. The results of Chapter 4 indicated that none of the storage policies achieve a balanced system when blocking is enforced and the pickers are ordered from fastest to slowest (reverse ordering).

| Storage assignment policy | Throughput time |
|---|---|
| Random | 3529 |
| Dedicated | 1027 |
| Class-based | 1407 |

Table 13: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with two pickers, with blocking and a reverse ordering

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 2356 | 2356 | 1278 | 1278 | 1272 | 1272 |
| 2 | 2501 | 1574 | 882 | 481 | 1205 | 658 |

Table 14: Average distance walked per order for each picker and the amount of workload this represents for the random, dedicated and class-based storage policies. Picker 1 is the last and slowest picker and picker 2 is the first and fastest picker

The conclusions of Section 4.2 are not changed for the case with two pickers. The throughput time is lowest for dedicated storage and highest for random storage. In addition, there is no balanced system to be detected from Table 14 and the sequences of the statistics is also the same. Table 15 and Table 16 present the results for a bucket brigade with ten pickers which are not allowed to pass one another. Table 15 presents the order throughput times and Table 16 presents the average distances and workload values.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 1458 |
| Dedicated | 306 |
| Class-based | 413 |

Table 15: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with ten pickers, with blocking and a reverse ordering

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 582 | 582 | 671 | 671 | 726 | 726 |
| 2 | 467 | 333 | 448 | 284 | 492 | 315 |
| 3 | 436 | 216 | 373 | 156 | 415 | 172 |
| 4 | 426 | 152 | 357 | 105 | 385 | 112 |
| 5 | 443 | 117 | 351 | 79 | 354 | 79 |
| 6 | 477 | 99 | 347 | 63 | 331 | 60 |
| 7 | 537 | 90 | 349 | 53 | 328 | 49 |
| 8 | 634 | 88 | 381 | 49 | 337 | 44 |
| 9 | 777 | 92 | 342 | 39 | 362 | 41 |
| 10 | 1012 | 105 | 297 | 30 | 399 | 40 |

Table 16: Average distance walked per order for each picker and the amount of workload this represents for the random, dedicated and class-based (class) storage policies. Picker 1 is the last and slowest picker and picker 10 is the first and fastest picker

Again, the relationship between the bucket brigade and the storage assignment policies is not affected by an increase or a decrease of the number of pickers. In all instances dedicated storage is the policy with the lowest throughput time and random storage is the policy with the highest throughput time. Regarding balance of workload, varying the number of pickers does not change the conclusions of Chapter 4.

## 5.2 Number of articles

Each article has two properties, its demand and its location. The location determines the distance a picker has to walk to retrieve the article. An increase (decrease) in the number of articles increases (decreases) the distance a picker has to walk because more (less) room in the warehouse is needed. This will affect the order throughput time and it may also affect the balance of the workload. To test the sensitivity of the conclusions of Chapter 4 to this property I implement a bucket brigade with 1000 articles and 10000 articles. For the basis scenario the number of articles was assumed to be 5000.

**Passing of pickers**

Table 17 shows the order throughput times and Table 18 shows the workload values when there are 1000 articles in the warehouse.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 279 |
| Dedicated | 119 |
| Class-based | 144 |

Table 17: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with 1000 articles

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 56 | 56 | 24 | 24 | 29 | 29 |
| 2 | 112 | 56 | 52 | 26 | 58 | 29 |
| 3 | 169 | 56 | 85 | 28 | 90 | 30 |
| 4 | 232 | 58 | 135 | 34 | 133 | 33 |
| 5 | 323 | 65 | 246 | 49 | 252 | 50 |

Table 18: Average distance walked per order for each picker and the amount of workload this represents for the random, dedicated and class-based storage policies. Picker 1 is the slowest and picker 5 is the fastest

When there are only 1000 articles in the warehouse, it is not surprising the order throughput times are lower. The orders require a shorter distance to be walked and with an unchanged speed of the pickers, the time to complete an order decreases. Aside from this decrease in throughput time, the results of Table 17 and Table 18 do not exhibit a significantly different conclusion compared to Table 1 and Table 2. Dedicated storage is the policy with the lowest throughput time and random storage is the policy with the highest throughput time. Furthermore, random storage is still the only policy which approximates a balanced system. When the class-based storage policy is applied, the first pickers again approximate a balanced system but the workload of the last picker pushes the bucket brigade away from a complete balance.

Table 19 presents the order throughput time and Table 20 presents the workload values when there are 10000 articles in the warehouse.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 2730 |
| Dedicated | 884 |
| Class-based | 1267 |

Table 19: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with 10000 articles

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 546 | 546 | 177 | 177 | 253 | 253 |
| 2 | 1099 | 549 | 416 | 208 | 510 | 255 |
| 3 | 1670 | 557 | 726 | 242 | 780 | 260 |
| 4 | 2287 | 572 | 1213 | 303 | 1123 | 281 |
| 5 | 3073 | 615 | 2326 | 465 | 2079 | 416 |

Table 20: Average distance walked per order for each picker and the amount of workload this represents for the random, dedicated and class-based storage policies. Picker 1 is the slowest and picker 5 is the fastest

An increase in the number of articles forces the order pickers to walk a larger distance, which increases the order throughput time and the workload. However, this increase occurs for all storage assignment policies. It therefore does not change the conclusions of Section 4.1.

**Blocking of pickers**

Whether the conclusions of Section 4.2 are also robust against an increase or a decrease in the number of articles is revealed in the next paragraphs. These present the results of a bucket brigade which does not allow pickers to pass one another. Table 21 presents the order throughput times and Table 22 presents the workload values when there are 1000 articles in the warehouse.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 478 |
| Dedicated | 143 |
| Class-based | 171 |

Table 21: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with 1000 articles, blocking and a reverse ordering

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 217 | 217 | 172 | 172 | 191 | 191 |
| 2 | 185 | 132 | 132 | 80 | 137 | 83 |
| 3 | 179 | 90 | 125 | 48 | 128 | 49 |
| 4 | 205 | 68 | 130 | 35 | 135 | 36 |
| 5 | 291 | 65 | 131 | 27 | 156 | 32 |

Table 22: Average distance walked per order for each picker and the amount of workload this represents for the random, dedicated and class-based storage policies. Picker 1 is the last and slowest picker and picker 5 is the first and fastest picker

Similar to Table 17, the order throughput times decrease. However, the dedicated storage policy continues to exhibit the lowest throughput time and the random storage policy also exhibits the highest throughput time. In addition, decreasing the number of articles does not produce a balanced system with respect to workload. The inability of the bucket brigade to attain a balanced system, when blocking is enforced and pickers are reverse ordered, does not disappear by decreasing the number of articles.

Table 23 shows the order throughput times and Table 24 presents the workload values when the number of articles equals 10000 and blocking is enforced with a reverse ordering.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 4018 |
| Dedicated | 1047 |
| Class-based | 1501 |

Table 23: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with 10000 articles, blocking and a reverse ordering

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 1895 | 1895 | 1694 | 1694 | 1790 | 1790 |
| 2 | 1604 | 1107 | 1231 | 744 | 1216 | 706 |
| 3 | 1737 | 760 | 1166 | 445 | 1098 | 415 |
| 4 | 2149 | 626 | 1146 | 308 | 1136 | 310 |
| 5 | 2976 | 631 | 968 | 199 | 1340 | 280 |

Table 24: Average distance walked per order for each picker and the amount of workload this represents for the random, dedicated and class-based storage policies. Picker 1 is the last and slowest picker and picker 5 is the first and fastest picker

Analogous to the number of pickers, the relationship of the bucket brigade with the storage assignment policies is unaffected by the number of articles. Increasing or decreasing this number does not produce different conclusions than in Section 4.1 and Section 4.2.

## 5.3 Picker speed

For the picker speeds the assumptions of Bartholdi and Eisenstein (1996a) are followed in Chapter 4 as these velocities are considered constant and each picker's velocity is distinct. It is however not tested whether the difference in speed between two pickers is of importance to the results. I therefore implement a bucket brigade where the picker velocities are close to one another (within one tenth of a speed unit) and where the picker velocities are far apart (at least 10 speed units).

### Passing of pickers

The next paragraphs present the results for the bucket brigade which allows pickers to pass one another. Table 25 presents the order throughput times and Table 26 presents the workload values for the case where picker speeds are close to one another.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 1333 |
| Dedicated | 436 |
| Class-based | 597 |

Table 25: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with picker speeds close to one another

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 800 | 267 | 262 | 87 | 358 | 119 |
| 2 | 840 | 271 | 351 | 113 | 383 | 124 |
| 3 | 891 | 278 | 451 | 141 | 435 | 136 |
| 4 | 962 | 292 | 610 | 185 | 564 | 171 |
| 5 | 1151 | 338 | 1034 | 304 | 1031 | 303 |

Table 26: Average distance walked per order for each picker and the amount of workload this represents for the random, dedicated and class-based storage policies. Picker 1 is the slowest and picker 5 is the fastest

The order throughput times are comparable to the values of Table 1 because the average picker speed is almost equal to the average picker speed of the basis scenario. The results show that picker velocities which are closer to one another do not lead to different conclusions. The order throughput times are still in the same sequence when ordered from lowest to highest. Further, the workload values of random storage are still the only values which approximate a balanced system although the differences between the pickers are larger compared to Table 2. This is because the impact of early completions is larger when there is little variation in the picker speeds. An early completion distorts the balance of the bucket brigade as shown in Section 4.1. Recovering from this distortion takes more time when the picker speeds are close to one another. This also indicates that when the variation is further reduced to for example zero, the balance of workload may disappear altogether.

In assigning picker velocities which are far apart it is no longer possible to have an average picker speed which is equal to the average picker speed of the basis scenario. For this case Table 27 presents the order throughput times and Table 28 presents the workload values.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 91 |
| Dedicated | 30 |
| Class-based | 41 |

Table 27: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with picker speeds far apart

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 18 | 18 | 6 | 6 | 8 | 8 |
| 2 | 163 | 18 | 56 | 6 | 73 | 8 |
| 3 | 529 | 18 | 194 | 7 | 236 | 8 |
| 4 | 1230 | 18 | 516 | 8 | 554 | 8 |
| 5 | 2474 | 19 | 1346 | 10 | 1275 | 10 |

Table 28: Average distance walked per order for each picker and the amount of workload this represents for the random, dedicated and class-based storage policies. Picker 1 is the slowest and picker 5 is the fastest

To obtain a situation where the velocities are far apart, the last pickers have gained a large velocity. The distance required by an order is thus walked in a short time, as is evident from the small throughput times. Nevertheless, the sequence of the storage policies according to throughput time remains the same. Dedicated storage continues to exhibit the lowest value, followed by class-based storage and then random storage.

The workload values also do not suggest a different conclusion although that is not easily observed from Table 28. The bucket brigade combined with dedicated storage is still unable to achieve a balanced system. For class-based storage however, the high velocity of the last picker forces the pickers behind him to retrieve articles from the first class only. As these articles are randomly assigned locations, the environment in which the first pickers operate resembles a random storage setting and their workload approximates a balanced system. Nonetheless, the workload of the last picker is still higher than the workload of the other pickers. The difference is not large but it is larger compared to random storage. The random storage policy therefore continues to be the policy which best approximates a balanced system.

**Blocking of pickers**

Similar to Section 5.1 and 5.2, the sensitivity analysis is also applied to a bucket brigade which does not allow pickers to pass one another. Table 29 presents the order throughput times and Table 30 presents the workload values for the case where picker speeds are close to one another.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 1337 |
| Dedicated | 437 |
| Class-based | 597 |

Table 29: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with picker speeds close to one another

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 1150 | 383 | 1002 | 334 | 1019 | 340 |
| 2 | 951 | 308 | 616 | 199 | 572 | 185 |
| 3 | 907 | 285 | 474 | 148 | 449 | 141 |
| 4 | 900 | 274 | 385 | 117 | 412 | 125 |
| 5 | 907 | 267 | 297 | 87 | 406 | 119 |

Table 30: Average distance walked per order for each picker and the amount of workload this represents for the random, dedicated and class-based storage policies. Picker 1 is the last and slowest picker and picker 5 is the first and fastest picker

The most surprising property of the order throughput times of Table 29 is that they are equal to the values of Table 25. When picker speeds have a small variation the number of blockings is small because it takes longer for pickers to catch up with one another. Although the number of blockings is small, these are still present as is evident from the workload values of Table 30. These are not entirely equal to the values of Table 26 but they do suggest that when the picker speeds are close to one another it is of less importance whether blocking is enforced or not. This also means the ordering of the pickers is less important because the above results are obtained with a reverse ordering.

Altogether, these conclusions are significantly different from the conclusions of Section 4.2. They therefore warrant further investigation. Figure 21 presents the hand-over points and completion points of the last picker for random storage. I only consider random storage because the workload values for this policy in Table 30 suggest a balanced system may be possible. For the other policies this suggestion is not so strong.

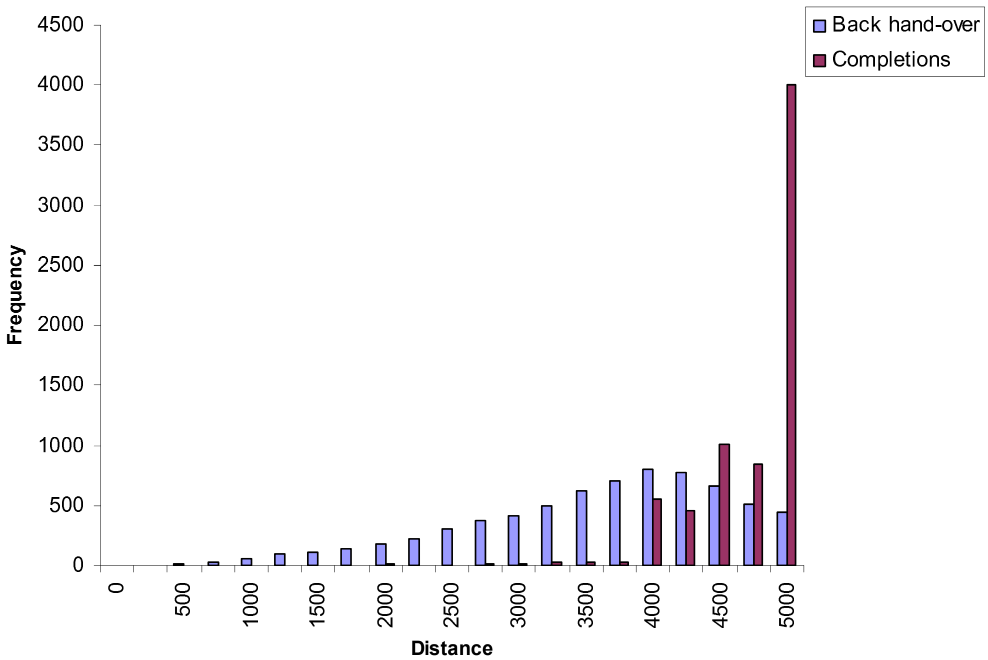


Figure 21: Points at which the last picker exchanges orders with the picker behind him (back hand-over) and points at which the last picker is finished (completions) with the random storage policy, blocking and little variation in the picker speeds

The difference with Figure 14, the case with normal speeds, is the location of the peak in the back hand-over points. Although Figure 14 does not exhibit this peak, a large part of the points are concentrated between 4000-5000. For Figure 21 however, the number of back hand-over points increases until the 4000 mark. Between 4000 and 5000 this number decreases. A large part of the back hand-over points therefore occurs before the 4000 mark. This behaviour resembles the results of Figure 4, the case with normal speeds and no blocking. The extent of this resemblance is nevertheless limited because of blocking. The blocking prevents a large peak in the back hand-over points, which would indicate the last picker walks a distance confined to small interval.

Before reaching a final conclusion the hand-over points of the middle picker are also considered in Figure 22. These points include the points at which orders are exchanged with the picker behind the middle picker and the picker in front of the middle picker. Figure 22 also depicts the completion points of the middle picker.

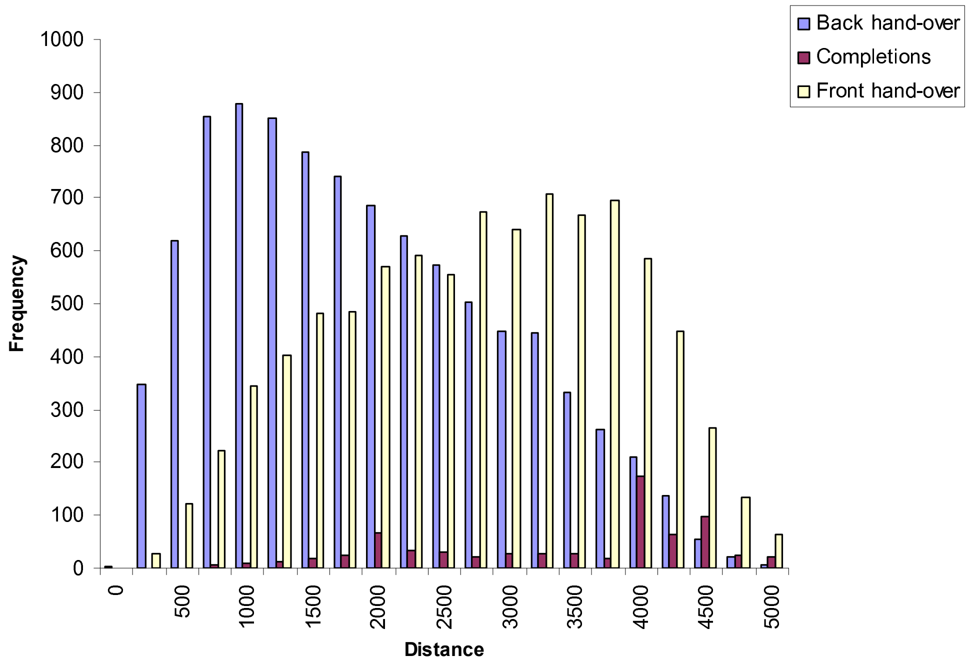


Figure 22: Points at which the middle picker exchanges orders with the picker behind him (back hand-over), in front of him (front hand-over) and points at which the middle picker is finished (completions) with the random storage policy, blocking and little variation in the picker speeds

This figure differs from Figure 16 regarding the location of the mass of the hand-over points. For Figure 16 this mass was spread over almost all distances, while for Figure 22 the back hand-over points are concentrated on the left side of the figure and the front hand-over points on the right side. This again resembles the case with no blocking and normal speeds, presented in Figure 6. Nevertheless, because of blocking the hand-over points do not show a significant peak which would indicate the distance walked by the middle picker is restricted to a small interval.

The results of Table 30 suggest a balanced system with respect to workload is possible for random storage when picker speeds are close to one another. Figure 21 and Figure 22 however show that the influence of blocking has not disappeared and it prevents a balance to occur. Table 31 presents the order throughput times and Table 32 presents the workload values for the case where picker speeds are far apart.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 1108 |
| Dedicated | 88 |
| Class-based | 597 |

Table 31: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with picker speeds far apart

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 843 | 843 | 679 | 679 | 697 | 697 |
| 2 | 667 | 234 | 558 | 99 | 525 | 104 |
| 3 | 651 | 59 | 620 | 28 | 596 | 28 |
| 4 | 984 | 22 | 854 | 14 | 849 | 14 |
| 5 | 2709 | 23 | 973 | 8 | 1378 | 11 |

Table 32: Average distance walked per order for each picker and the amount of workload this represents for the random, dedicated and class-based storage policies. Picker 1 is the slowest and picker 5 is the fastest

Unlike the case when picker speeds are close to one another, considering picker speeds which are far apart exaggerates the difference between the storage policies. The difference in throughput time between dedicated storage and the other policies is larger in Table 31 compared to all other instances of order throughput times. Furthermore, the workload values do not indicate a balance is possible for any storage policy when blocking is allowed and picker speeds are far apart.

In summary, I conclude the results of Chapter 4 are sensitive to changes in the picker speeds to a certain extent. Decreasing the variation in the speeds for a bucket brigade which allows passing does not affect the conclusions. When picker speeds are far apart however, the class-based storage policy approximates a balanced system but the last picker prevents a complete balance.

When blocking is enforced, decreasing the variation in speeds reduces the importance of blocking as the results more closely resemble the bucket brigade without blocking. Nevertheless, the influence of blocking does not disappear as a balanced system with respect to workload is not achieved. Increasing the variation in the velocities also does not lead to different conclusions for the case with blocking.

## 5.4 Order length

The final sensitivity analysis concerns the order length. In Chapter 3 I assumed this property to be exponentially distributed with an average of 10. The analysis presented in this section examines the sensitivity of the results to a change in the distribution of the order length. Table 33 presents the order throughput times and Table 34 presents the workload values for a bucket brigade with an average order length of 20.

| Storage assignment policy | Throughput time |
|---|---|
| Random | 1505 |
| Dedicated | 627 |
| Class-based | 768 |

Table 33: Throughput times for a bucket brigade combined with three storage assignment policies, based on the basis scenario with an average order length of 20

| Picker | Random | | Dedicated | | Class-based | |
|---|---|---|---|---|---|---|
| | Average distance | Workload | Average distance | Workload | Average distance | Workload |
| 1 | 301 | 301 | 125 | 125 | 154 | 154 |
| 2 | 604 | 302 | 275 | 138 | 308 | 154 |
| 3 | 911 | 304 | 454 | 151 | 471 | 157 |
| 4 | 1237 | 309 | 710 | 177 | 694 | 173 |
| 5 | 1586 | 317 | 1255 | 251 | 1293 | 259 |

Table 34: Average distance walked per order (Average) for each picker and the amount of workload this represents for the random, dedicated and class-based (class) storage policies. Picker 1 is the slowest and picker 5 is the fastest

An increase in the order length means more articles can be added to the order. This increases the chance an article with a large distance occurs on the order, thus increasing the walking distance. This explains why the order throughput time and the workload values have increased. A decrease of the order length will have a reverse effect as fewer articles with a large distance are placed on the orders. At this point, I conclude the results exhibit the same sensitivity to the order length as the sensitivity to the number of articles. It is therefore not worth examining another distribution for the order length. An increase (decrease) of the order length will automatically lead to a larger (smaller) throughput time and workload for all storage policies. Therefore changing the order length does not change the conclusions derived in Chapter 4.

# 6 Conclusion

The applications of bucket brigades mentioned in Chapter 1 are examples of the flexibility of this order picking system. In the research presented in this paper this flexibility is further explored. The bucket brigades are applied to a warehouse with three different storage assignment policies. The objective is to study the relationship between bucket brigades and these storage assignment policies. In Section 6.1 the results of this study are summarized and the conclusions based on these results are presented. Section 6.2 presents suggestions for further research.

## 6.1 Summary

The first storage policy examined is random storage, a policy which randomly assigns locations to articles. This policy complies with the assumptions Bartholdi and Eisenstein (1996a) defined for a bucket brigade which achieves a maximum throughput and a balance of the workload. The results of this paper confirm the findings of Bartholdi and Eisenstein (1996a), the bucket brigade with random storage achieves a balanced system with respect to workload. The randomness of the orders, as indicated by Bartholdi and Eisenstein (1996b), complicate the convergence to balance of workload but the bucket brigade is able to handle this randomness.

These results for random storage are achieved without the restriction of blocking. Nevertheless, the ordering of the workers converges to the required ordering where pickers are ordered from slowest to fastest. The importance of this ordering is revealed when blocking is enforced. Pickers who are ordered from slowest to fastest do not produce any blocking. However, a bucket brigade which enforces blocking and where pickers are ordered from fastest to slowest does not achieve a balance of the workload and order throughput time increases with 50%.

The second storage assignment policy examined is dedicated storage, a policy which assigns locations close to the depot to the articles with a high demand. This policy therefore does not comply with the assumptions of Bartholdi and Eisenstein (1996a) because it does not spread work content uniformly over the warehouse. This results in the lowest order throughput time in all instances of the bucket brigade. Nonetheless, a balance of the workload is never achieved. Dedicated storage increases the randomness of the orders, mainly through the early completion points, to a point where the bucket brigade is not able to converge to a balanced system. However, this property of the dedicated storage policy also limits the influence of blocking on the bucket brigade.

Finally, the third storage assignment policy considered is class-based storage. This policy divides the articles into separate classes. Each class receives its own area in the warehouse and articles are assigned locations in this area randomly. The class-based storage policy has commonalities with dedicated storage and random storage. This is also evident from the results. The influence of dedicated storage is that class-based storage also does not achieve a balance of the workload. Its throughput time is also lower than the throughput time of random storage. The influence of random storage within the classes is that the first couple of pickers in the line are able to balance their workload. However, this increases the workload of the pickers at the end of the line. Furthermore, the random characteristic of class-based storage exposes it to the influence of blocking. The impact of blocking on the results of class-based storage is therefore larger compared to dedicated storage.

The conclusions of this paper are robust to changes in almost all of the assumptions. Specifically, they are not sensitive to changes in the number of pickers, number of articles and the distribution of the order length. However, changes in the picker velocities do have an influence on the results. Increasing the variation in these velocities, when allowing pickers to pass one another, brings a bucket brigade with class-based storage closer to a balanced system. Decreasing the variation reduces the ability of a bucket brigade with random storage to converge to a balanced system. For a bucket brigade which enforces blocking, reducing the variation in the picker velocities reduces the importance of blocking. Therefore, when combined with random storage the bucket brigade approximates a balanced system. Nevertheless, the influence of blocking is still visible in the hand-over points.

## 6.2 Further research

In Section 3.1, the assumption is made to implement the bucket brigade in a warehouse setting with only one aisle. It is also noted that this is not representative of a warehouse in reality. Further research can therefore apply the bucket brigade to an environment with multiple aisles. The considerations in this case concern the routing of pickers and the preservation of the ordering from slowest to fastest. Bartholdi and Eisenstein (1996b) note that an implementation in a warehouse with multiple aisles complicates the preservation because it is more difficult for pickers to find one another. A possible solution would be to forgo the restrictions of the *Normative Model* and allow pickers to pass one another. The picker who is finished would then exchange orders with the first picker he finds.

A second avenue for further research concerns the randomness of the orders and their completion points. This randomness greatly affects the results of this paper. Bartholdi and Eisenstein (1996b) suggest that order pickers are given multiple orders. By combining orders which require a short distance to be walked with orders that require a longer distance, this solution prevents early completions to take place. The order is therefore always passed to the last picker who then initiates all resets. This solution is also referred to as the batching of orders, as discussed in Section 2.1. Further research could thus implement the different batching methods.

# 7 Appendix

## 7.1 Hand-over points with passing of pickers

Section 4.1 presented the hand-over points of the last and middle picker. In this appendix the hand-over points of the remaining pickers are presented. The results for are first presented for the random storage policy, then for the dedicated storage policy and finally for the class-based storage policy.

### Random storage

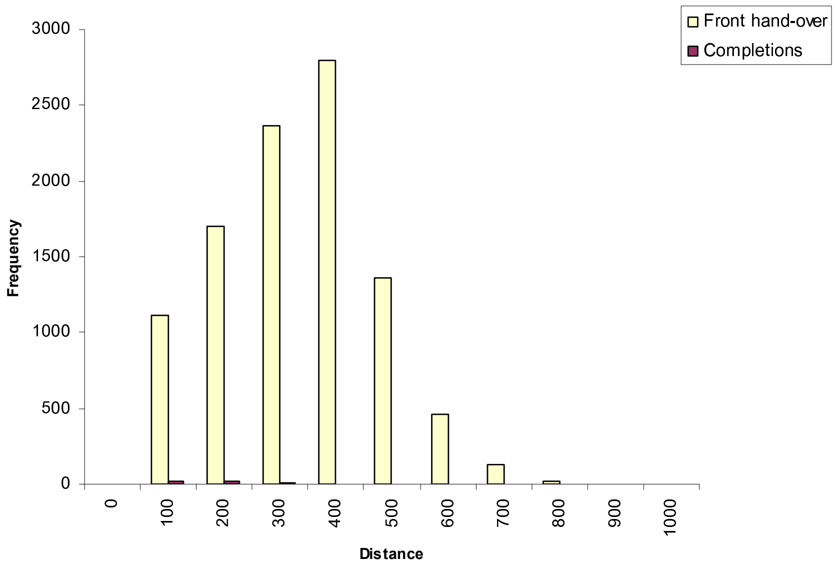


Figure 23: Points at which the first picker exchanges orders with the picker in front of him (front hand-over) and points at which the first picker is finished (completions)

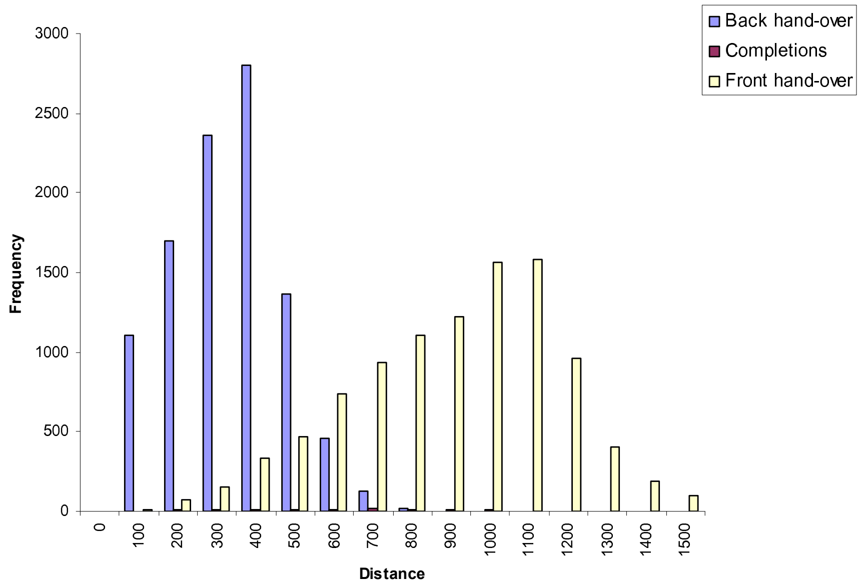


Figure 24: Points at which the second picker exchanges orders with the picker behind him (back hand-over), in front of him (front hand-over) and points at which the second picker is finished (completions)

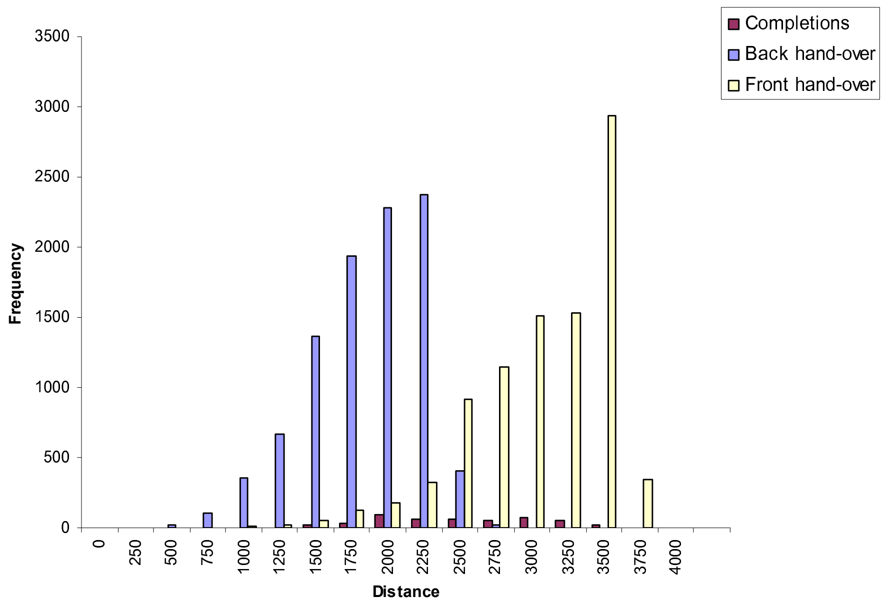


Figure 25: Points at which the fourth picker exchanges orders with the picker behind him (back hand-over), in front of him (front hand-over) and points at which the fourth picker is finished (completions)

## Dedicated storage

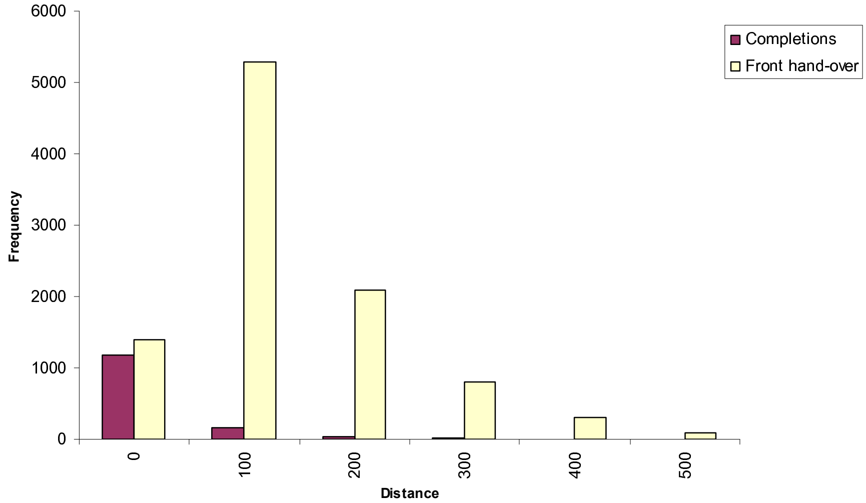


Figure 26: Points at which the first picker exchanges orders with the picker in front of him (front hand-over) and points at which the first picker is finished (completions)

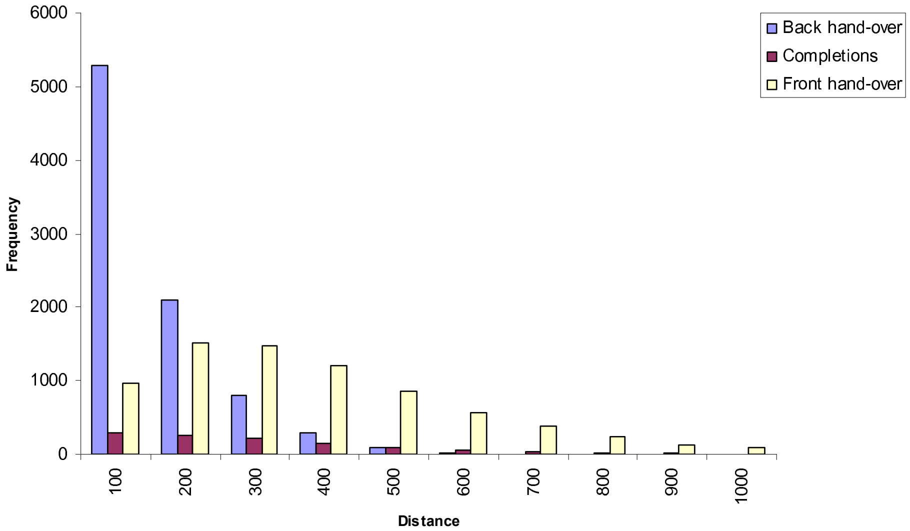


Figure 27: Points at which the second picker exchanges orders with the picker behind him (back hand-over), in front of him (front hand-over) and points at which the second picker is finished (completions)

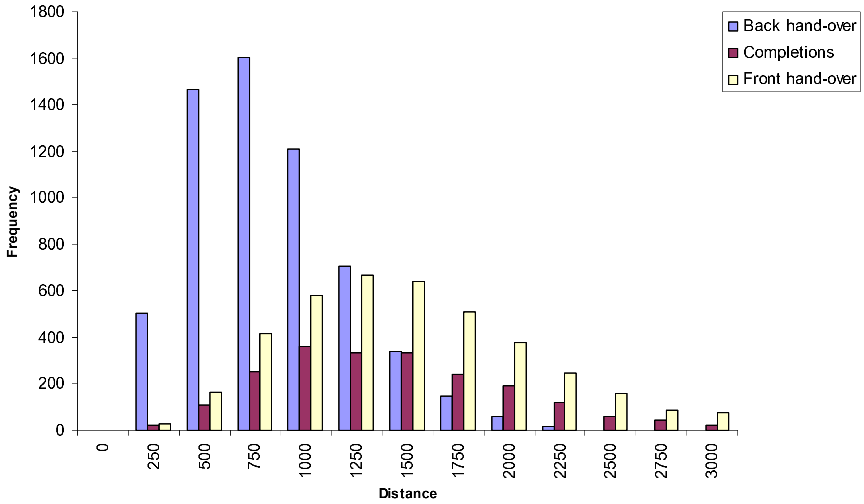


Figure 28: Points at which the fourth picker exchanges orders with the picker behind him (back hand-over), in front of him (front hand-over) and points at which the fourth picker is finished (completions)

## Class-based storage

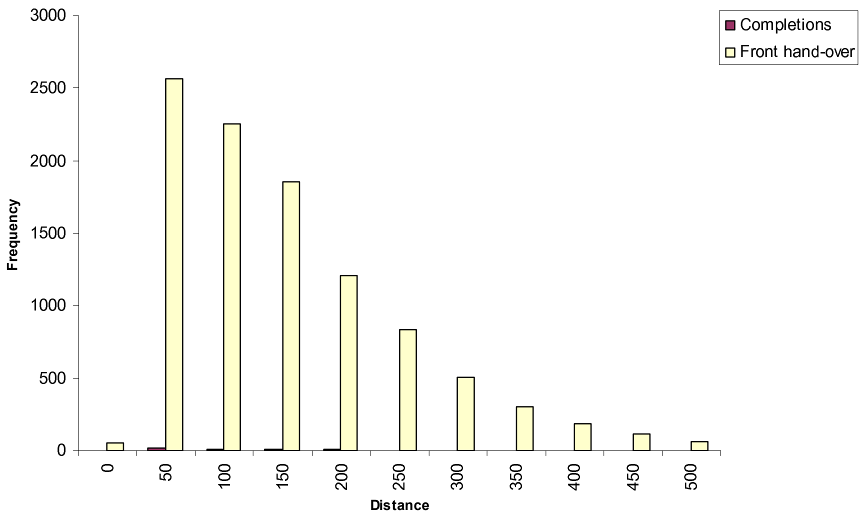


Figure 29: Points at which the first picker exchanges orders with the picker in front of him (front hand-over) and points at which the first picker is finished (completions)

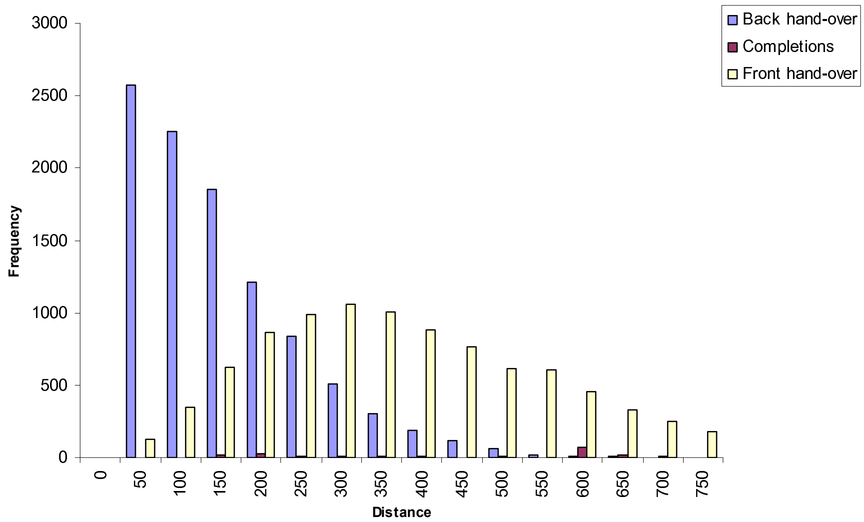


Figure 30: Points at which the second picker exchanges orders with the picker behind him (back hand-over), in front of him (front hand-over) and points at which the second picker is finished (completions)

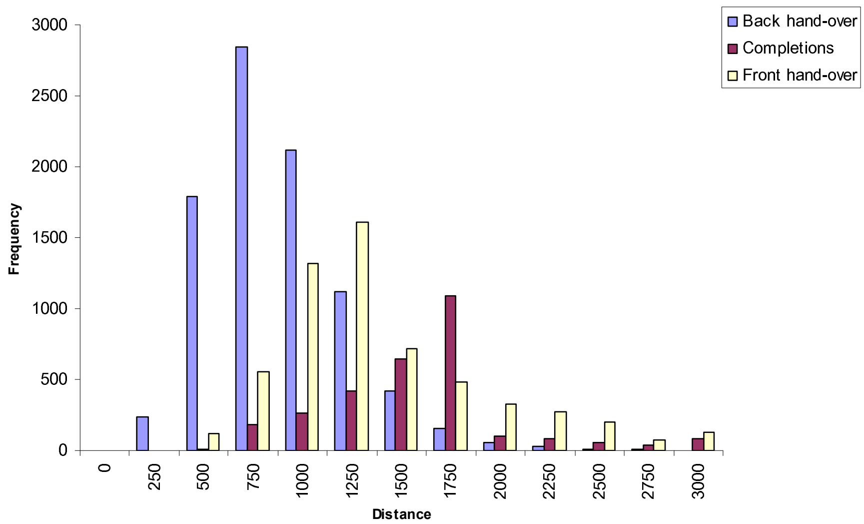


Figure 31: Points at which the fourth picker exchanges orders with the picker behind him (back hand-over), in front of him (front hand-over) and points at which the fourth picker is finished (completions)

## 7.2 Hand-over points with blocking of pickers

Section 4.2 presented the hand-over points of the last and middle picker for a bucket brigade which enforces blocking. In this appendix the hand-over points of the remaining pickers are presented. The results are first presented for the random storage policy, then for the dedicated storage policy and finally for the class-based storage policy.

**Random storage**

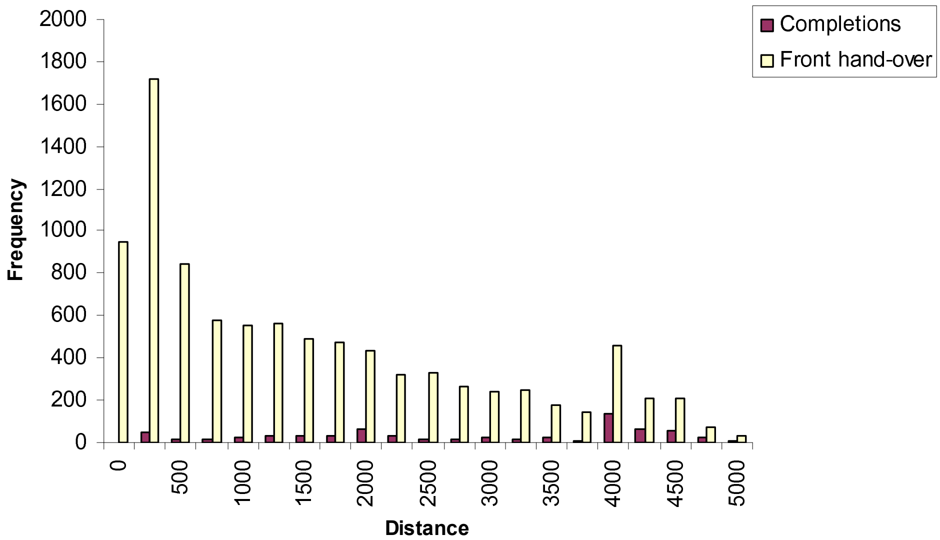


Figure 32: Points at which the first picker exchanges orders with the picker in front of him (front hand-over) and points at which the first picker is finished (completions) when blocking is enforced

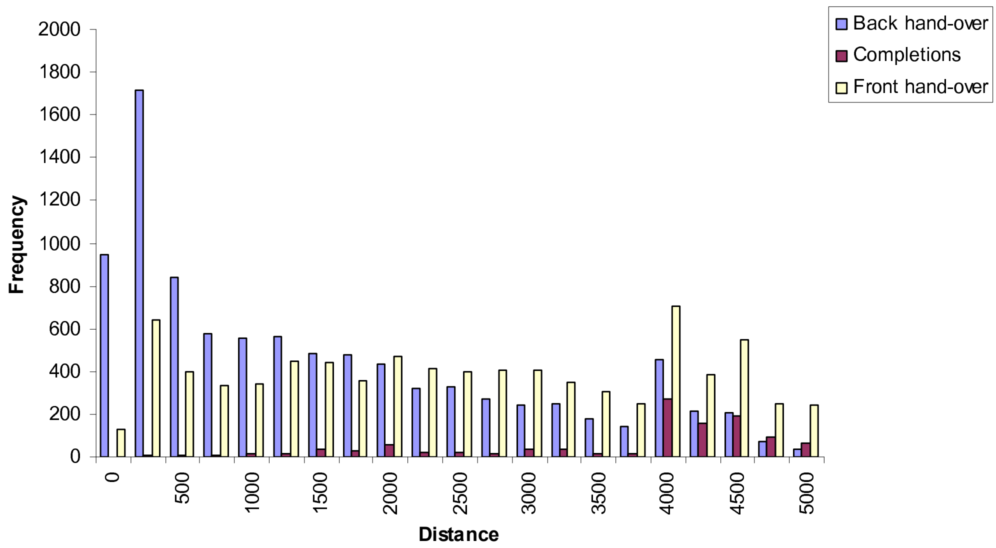


Figure 33: Points at which the second picker exchanges orders with the picker behind him (back hand-over) , in front of him (front hand-over) and points at which the second picker is finished (completions) when blocking is enforced

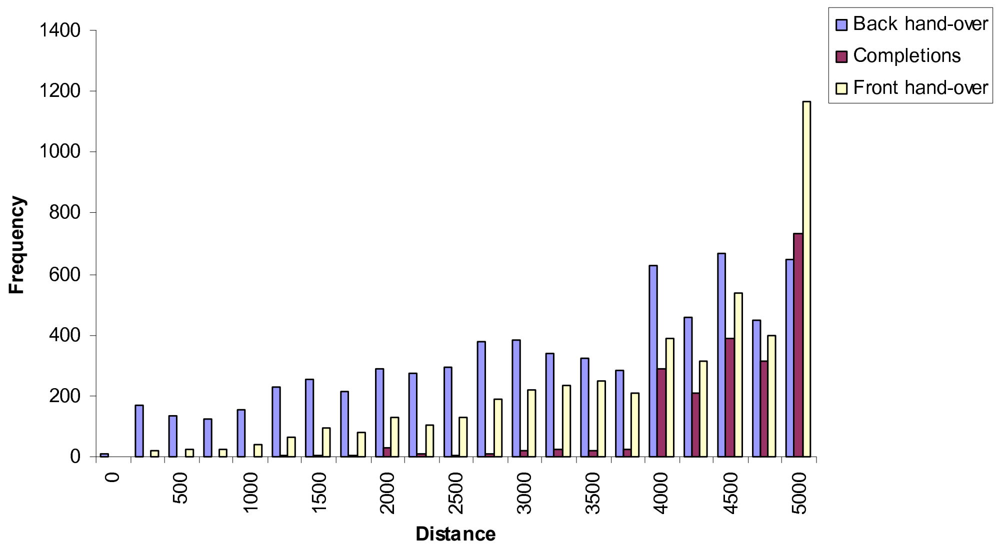


Figure 34: Points at which the fourth picker exchanges orders with the picker behind him (back hand-over) , in front of him (front hand-over) and points at which the fourth picker is finished (completions) when blocking is enforced

## Dedicated storage

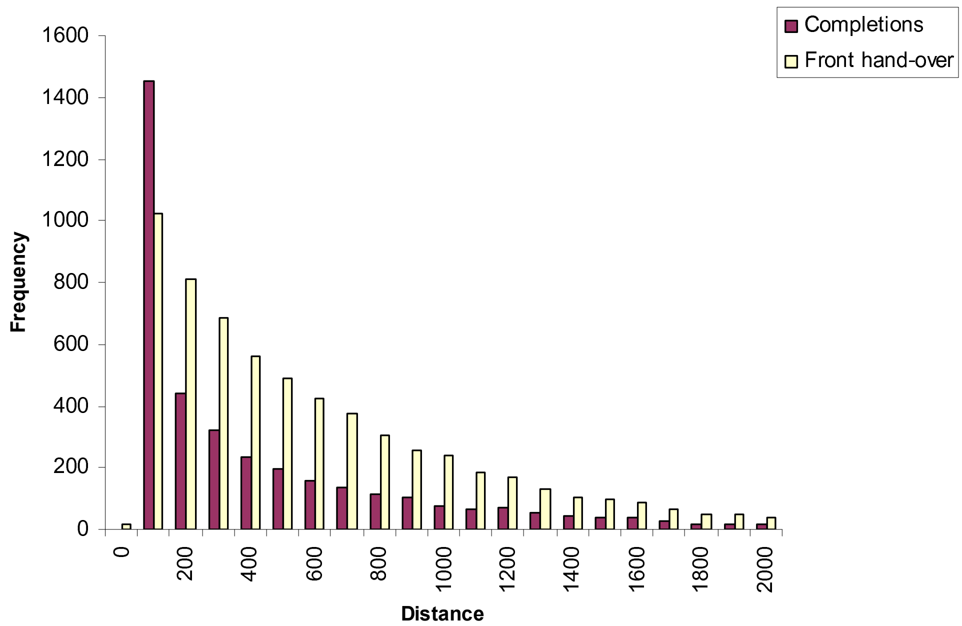


Figure 35: Points at which the first picker exchanges orders with the picker in front of him (front hand-over) and points at which the first picker is finished (completions) when blocking is enforced

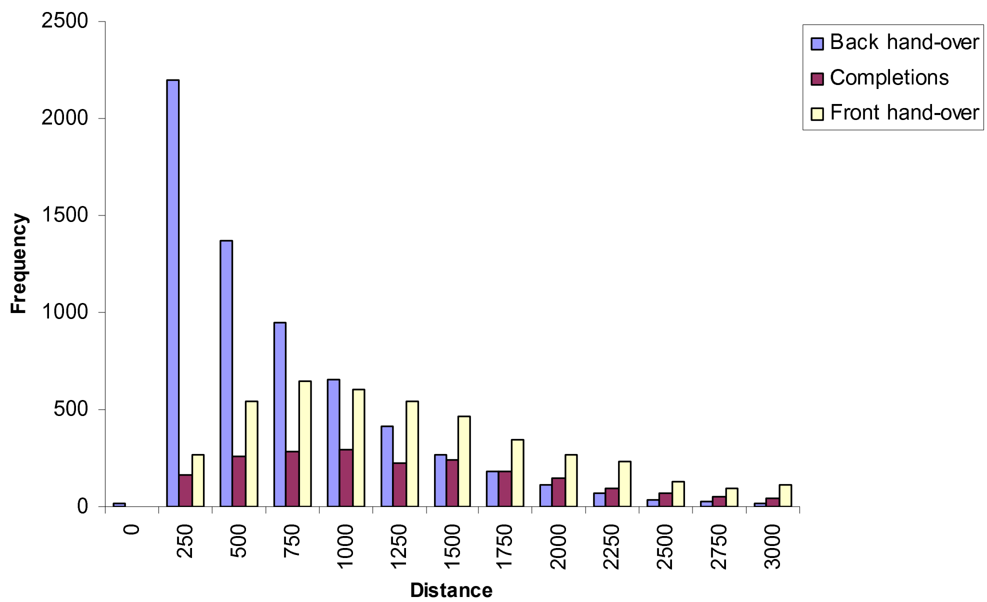


Figure 36: Points at which the second picker exchanges orders with the picker behind him (back hand-over) , in front of him (front hand-over) and points at which the second picker is finished (completions) when blocking is enforced

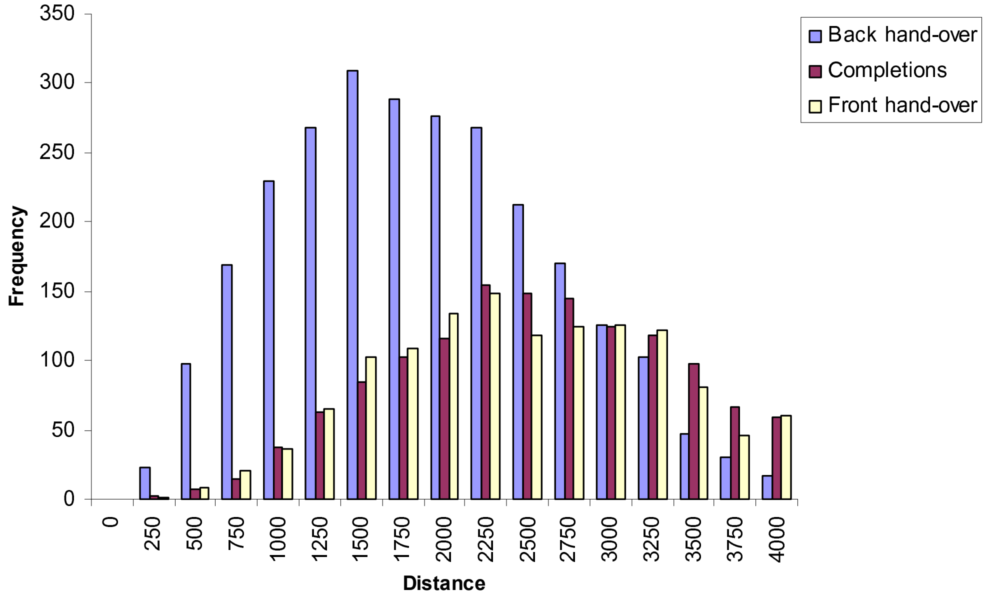


Figure 37: Points at which the fourth picker exchanges orders with the picker behind him (back hand-over) , in front of him (front hand-over) and points at which the fourth picker is finished (completions) when blocking is enforced

## Class-based storage

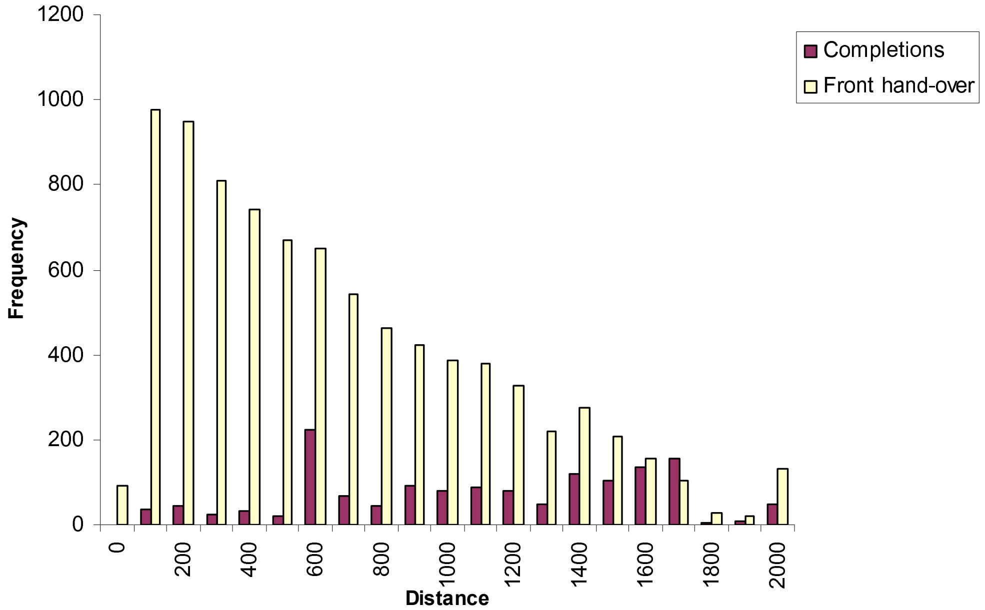


Figure 38: Points at which the first picker exchanges orders with the picker in front of him (front hand-over) and points at which the first picker is finished (completions) when blocking is enforced

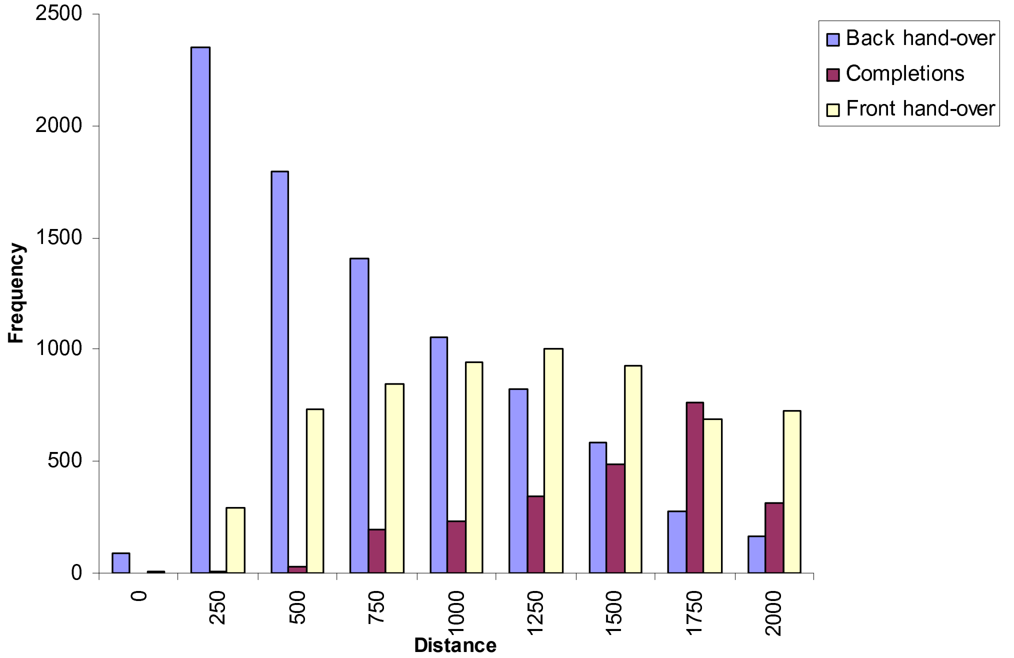


Figure 39: Points at which the second picker exchanges orders with the picker behind him (back hand-over) , in front of him (front hand-over) and points at which the second picker is finished (completions) when blocking is enforced

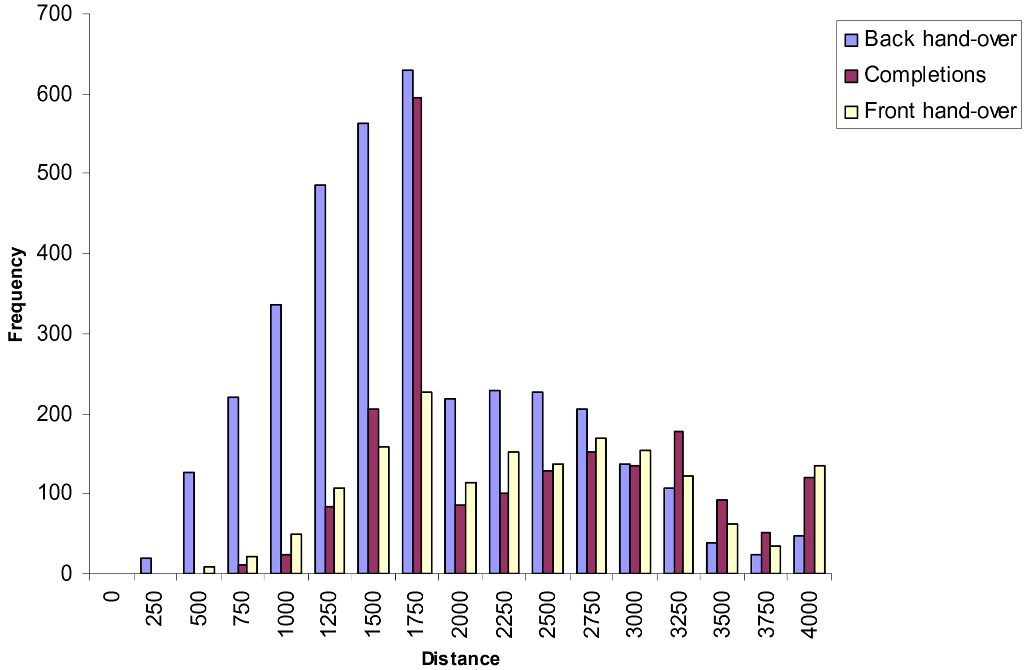


Figure 40: Points at which the fourth picker exchanges orders with the picker behind him (back hand-over) , in front of him (front hand-over) and points at which the fourth picker is finished (completions) when blocking is enforced

# 8 References


Armbruster, D. and Gel, E.S. (2002) Bucket brigades when worker speeds do not dominate each other uniformly, Working paper, Department of Industrial Engineering, Arizona State University

Bartholdi III, J.J., Bunimovich, L.A., and Eisenstein, D.D. (1999) Dynamics of two- and three-worker "bucket brigade" production lines, *Operations Research* **47(3)**, 488-491

Bartholdi III, J.J. and Eisenstein, D.D. (1996a) A production line that balances itself, *Operations Research* **44(1)**, 21-34

Bartholdi III, J.J. and Eisenstein, D.D. (1996b) Bucket brigades: a self-organizing order-picking system for a warehouse, Report, School of Industrial Engineering, Georgia Tech, Atlanta

Bartholdi III, J.J. and Eisenstein, D.D. (2005) Using bucket brigades to migrate from craft manufacturing to assembly lines, *Manufacturing & Service Operations Management* **7(2)**, 121-129

Bartholdi III, J.J., Eisenstein, D.D., and Foley, R. (2001) Performance of bucket brigades when work is stochastic, *Operations Research* **49(5)**, 710-719

Bartholdi III, J.J., Eisenstein, D.D., and Lim, Y.F. (2003) Chaos and convergence in bucket brigades with finite walk-back velocities, Working paper, School of Industrial and Systems Engineering, Georgia Institute of Technology

Bassan, Y., Roll, Y. and Rosenblatt, M.J. (1980) Internal layout design of a warehouse, *AIIE Transactions* **12(4)**, 317-322

Bratcu, A. and Dolgui, A. (2005) A survey of the self-balancing production lines ("bucket brigades"), *Journal of Intelligent Manufacturing* **16(2)**, 139-158

Caron, F., Marchet, G. and Perego, A. (2000) Optimal layout in low-level picker-to-part systems, *International Journal of Production Research* **38(1)**, 101-117

Choe, K. and Sharp, G.P. (1991) Small parts order picking: design and operation, available online at: http://www.isye.gatech.edu/logisticstutorial/order/article.htm (Accessed August 2007)

Clarke, G. and Wright, W. (1964) Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research* **12**, 568-581

Cornuéjols, G., Fonlupt, J., and Naddef, D. (1985) The travelling salesman problem on a graph and some related integer polyhedra, *Mathematical Programming* **33**, 1-27

De Koster, R. (2004) How to assess a warehouse operation in a single tour, Report, RSM Erasmus University, The Netherlands

De Koster, R., Le Duc, T., and Roodbergen, K.J. (2006) Design and Control of Warehouse Order Picking: a literature review, Research Paper ERS, Erasmus Research Institute of Management (ERIM)

De Koster, R. and Van der Poort, E.S. (1998) Routing order pickers in a warehouse: a comparison between optimal and heuristic solutions, *IIE Transactions* **30**, 469-480

De Koster, R., Van der Poort, E.S., and Wolters, M. (1999b) Efficient order batching methods in warehouses, *International Journal of Production Research* **37(7)**, 1479-1504

Drury, J. (1988) Towards more efficient order picking, IMM Monograph No. 1, Report, The institute of Materials Management, Cranfield, United Kingdom

ELA/AT Kearney (2004) Excellence in logistics 2004, Brussels: ELA

Elsayed, E.A. (1981) Algorithms for optimal material handling in automatic warehousing systems, *International Journal of Production Research* **19(5)**, 525-535

Gademann, A.J.R.N., Van Berg, J.P., and Van der Hoff, H.H. (2001) An order batching algorithm for wave picking in parallel-aisle warehouse, *IIE Transactions* **33**, 385-398

Gademann, A.J.R.N. and Van de Velde, S. (2005) Batching to minimize total travel time in a parallel-aisle warehouse, *IIE Transactions* **37(1)**, 63-75

Gibson, D.R. and Sharp, G.P. (1992) Order batching procedures, *European Journal of Operations Research* **58(1)**, 57-67

Goetschalckx, M. and Ashayeri, J. (1989) Classification and design of order picking systems, *Logistics World* **June**, 99-106

Hall, R.W. (1993) Distance approximation for routing manual pickers in a warehouse, *IIE Transactions* **25**, 77-87

Hausman, W.H., Schwarz, K.B., and Graves, S.C. (1976) Optimal storage assignment in automatic warehousing systems, *Management Science* **22(6)**, 629-638

Hwang, H., Baek, W. and Lee, M. (1988) Cluster algorithms for order picking in an automated storage and retrieval system, *International Journal of Production Research* **26**, 189-204

Jane, C.C. (2000) Storage location assignment in a distribution center, *International Journal of Physical and Logistics Management* **30(1)**, 55-71

Jane, C.C., and Laih, Y.W. (2005) A clustering algorithm for item assignment in a synchronized zone order picking system, *European journal of operational research* **166(2)**, 489-496

Kusiak, A. and Heragu, S. S. (1987) The facility layout problem, *European Journal of Operational Research* **29(3)**, 229-251

Lambert, D.M., Stock, J., and Ellram, L.M. (1998) Fundamentals of logistics management, McGraw-Hill, Singapore

Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., and Shmoys, D.B. (1995) The travelling salesman problem, Wiley, Chichester

Le Duc, T. and De Koster, R. (2003) An approximation for determining the optimal picking batch size for order picker in single aisle warehouses, in R. Meller, M.K. Ogle, B.A. Peters, G.D. Taylor, J. Usher (eds.), *Progress in Material Handling Research: 2002*, 267-286

Le Duc, T. and De Koster, R. (2005a) Determining the optimal number of zones in a pick-and-pack order picking system, Report ERS, RSM Erasmus University, the Netherlands

Le Duc, T. and De Koster, R. (2005b) Layout optimization for class-based storage strategy, *Supply Chain Management – European Perspective*, 191-214

Le Duc, T. and De Koster, R. (2007) Travel time estimation and order batching in a 2-block warehouse , *European Journal of Operational Research* **176(1)**, 374-388

Malmborg, C.J. (1995) Optimization of Cubic-per-Order Index layouts with zoning constraints, *International Journal of Production Research* **33(2)**, 465-482

Meller, R.D. and Gau, K.Y. (1996), The facility layout problem: recent and emerging trends and perspectives, *Journal of Manufacturing Systems* **15(5)**, 351-366

Pan, C.H. and Liu, S.Y. (1995) A comparative study of order batching algorithms, *Omega International Journal of Management* **23(6)**, 691-700

Petersen, C.G. (1997) An evaluation of order picking routing policies, *International Journal of Operations & Production Management* **17(11)**, 1098-1111

Peterson, C.G. (2002) Considerations in order picking zone configuration, *International Journal of Operations & Production Management* **27(7)**, 793-805

Petersen, C.G., Aase, G., and Heiser, D.R. (2004) Improving order-picking performance through the implementation of class-based storage, *International Journal of Physical Distribution & Logistics Management* **34(7)**, 534-544

Ratliff, H.D. and Rosenthal, A.S. (1983) Order picking in a rectangular warehouse: a solvable case of the travelling salesman problem, *Operations Research* **31(3)**, 507-521

Rosenblatt, M.J. and Roll, Y. (1984) Warehouse design with storage policy considerations, *International Journal of Production Research* **22(5)**, 809-821

Tang, L.C., and Chew, E.P. (1997) Order picking systems: batching and storage assignment strategies, *Computer & Industrial Engineering* **33(3)**, 817-820

Tompkins, J.A, White, J.A, Bozer, Y.A., Frazelle, E.H., and Tanchoco, J.M.A. (2003) Facilities Planning, John Wiley & Sons, New Jersey