# DYNAMIC ABC STORAGE POLICY IN ERRATIC DEMAND ENVIRONMENTS

**Benjamin Pierre, Bart Vannieuwenhuyse, Denis Dominanta**
Centrum voor Industrieel Beleid Katholieke Universiteit Leuven, Belgium

**Hendrik Van Dessel**
Applied Logistics NV, Mechelen, Belgium

## ABSTRACT

The paper describes a dynamic variant of the traditional ABC storage policy. The variant is to be used in manual order picking warehouses where SKUs experience a rather unstable demand. Its objective is to reduce the order picking time. It mainly consists in shifting items between the storage areas A, B and C according to the way their daily number of order lines changes. A few case studies based on computer simulations have shown that this variant can yield interesting time savings if its parameters are chosen with care. One of those case studies is presented in this paper. Because of the pioneering nature of this variant, some ideas for further research are also sketched.

**Keywords:** warehousing, class-based storage, order picking, ABC analysis, heuristic.

## 1. INTRODUCTION

One important activity in manual warehouses consists in picking up items that have been ordered by customers (Roodbergen, 2001). This activity is labour intensive and usually accounts for more than the half of operating costs. It is therefore an excellent source of cost savings. On the other hand, items (SKUs) to be stored in manual warehouses experience fluctuating demands, as a result of the growing customers' power. This paper explores a new storage policy that aims at improving the order picking activity by taking into account changing demand patterns.

The performance of the order picking activity is commonly measured by the order picking time, namely the time spent on satisfying a single order made up of a certain number of order lines. The order picking time includes four main components: the travel time (walking or driving to the storage locations), the processing time (once arrived at a storage location, picking up the right item), the acceleration/deceleration time (of the cart) and the administrative time (getting/depositing an order). The new storage policy described in this paper primarily intends to reduce the travel time.

Many parameters affect the performance of the order picking activity: the layout of the warehouse (e.g. dimensions, number of cross-aisles, etc.), the storage system (e.g. shelves, drawers, etc.), the handling equipment (e.g. fork-lift trucks, carts, etc.) and the various operating policies. An operating policy is basically a set of decision rules that control processes in the warehouse. For instance, the order picking policy tells the order picker how he has to satisfy orders (routes to travel through the warehouse; orders to satisfy at the same time; zones where to pick up items; etc.). This paper deals with storage policies, which assign items to storage locations.

Storage policies differ according to their accuracy and their easiness. The most basic one is the random storage policy, which randomly assigns items to empty storage locations. On the other hand, the frequency-based storage policy assigns the item with the largest picking frequency (i.e. total number of order lines corresponding to this item) to the storage location closest to the depot, which is the spot where empty/full carts are respectively fetched/left. All other items are assigned one by one to storage locations according to that frequency criterion. In between, the class-based storage policy is a compromise between the accuracy of the frequency-based storage policy and the easiness of the random storage policy. A particular case of the class-based storage policy is the ABC storage policy. Variants of the ABC storage policy that are implemented nowadays suffer from one major drawback: they are quite static and are not appropriate to today's erratic environments. The objective of this paper is to introduce a new variant of the ABC storage policy that is dynamic and takes into account the fluctuating nature of demand patterns.

The paper is structured as follows. The next section explains the basic principles of the ABC storage policy and differentiates between its static and dynamic variants. Then, assumptions on which the new dynamic variant developed by the authors is based are presented. The following section describes in detail the new dynamic variant. Finally, its relevance is shown through a case study coming from the durable consumer goods business. We conclude the paper with a set of recommendations for further research.

## 2. ABC STORAGE POLICY

The ABC storage policy is based on the well-known ABC analysis, also called Pareto analysis (Bulfin, 1998). This analysis consists in classifying items into three classes (A, B and C) according to their contribution to the total number of order lines. The few A-items (10 percent of items), also called fast-movers, have the highest numbers of order lines among all items, whereas the numerous C-items (70 percent of items), also called slow-movers, have the lowest numbers of order lines. Table 1 and Figure 1 illustrate how the classification is performed. Figure 2 shows how storage areas in the warehouse are allocated to classes. Within each storage area, a random storage policy is implemented. It is important to note that there are other definitions of classes (instead of 10-20-70 percent of items) and other allocations of storage areas to classes (instead of the within-aisle allocation). The context (i.e. kind of business, layout of the warehouse, order picking policy, etc.) will dictate the most suiTable choice (Petersen, 1999).

**Table 1. ABC Analysis**

| Item's tag | Number of order lines | Cumulative of items (%) | Cumulative of order lines (%) |
|---|---|---|---|
| 54 | 121 | 1 | 6.6 |
| 27 | 102 | 2 | 12.2 |
| 13 | 96 | 3 | 17.4 |
| 84 | 93 | 4 | 22.5 |
| 100 | 90 | 5 | 27.5 |
| 38 | 82 | 6 | 32 |
| 2 | 78 | 7 | 36.2 |
| … | … | … | … |
| 8 | 0 | 99 | 100 |
| 92 | 0 | 100 | 100 |

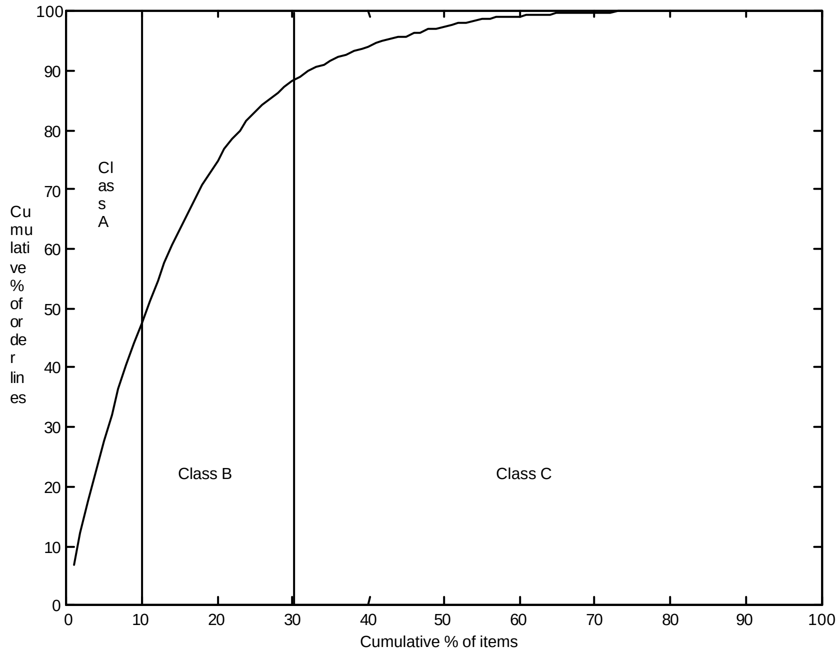


**Figure 1. ABC Curve**

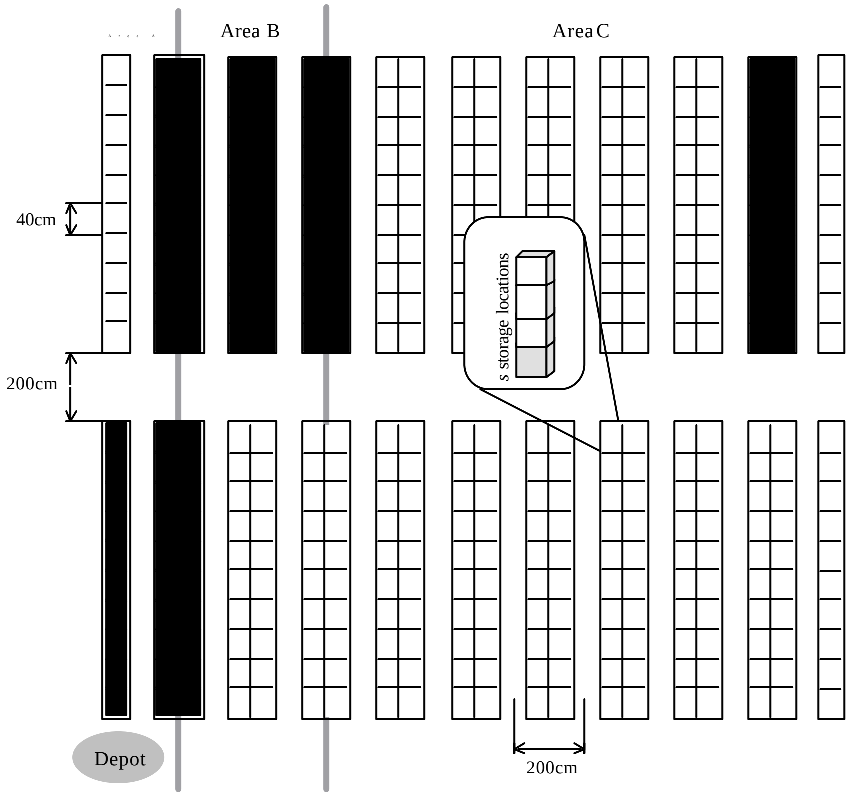


**Figure 2. Layout of the Warehouse**

In static variants of the ABC storage policy, the classification is reviewed on a medium- or long-term basis, for example every 6 months. In the dynamic variant developed by the authors, the classification is reviewed on a short-term basis, for example every day. If the classification changes, some items will have to be shifted through the warehouse from one area to another area (this activity is hereafter called "reshuffle"). A dynamic variant is obviously much more appropriate to fluctuating demand patterns. Nowadays, a fast-moving item can disappear in a few days because a better substitute has been launched on the market, while a slow-moving item can temporarily become fast-moving because of a smart marketing action. Since items are always stored in the most suiTable area, one can hope to have shortened travel times and an increased performance of order picking activity. There is one drawback however: a dynamic ABC strategy requires additional operations, reducing the profit of increased picking performance. We will pay attention to the cost balancing issue.

## 3. ASSUMPTIONS

The dynamic variant of the ABC storage policy has been developed and tested under the following main assumptions.

- The layout of the warehouse is depicted in Figure 2. Each square represents *s* storage location. Each storage location may contain a reasonable amount (i.e. stock-outs are avoided) of any item (i.e. items are not significantly physically different), and there is only one storage location per item.
- An important underlying problem consists in determining the shortest route that the operator has to travel through the warehouse in order to pick up items (order picking) or to shift items (reshuffle). Once the shortest route has been determined, the order picking/reshuffle time mentioned earlier can be directly computed. In the literature (Ratliff, 1983), this routing problem is recognized as a variant of the Travelling Salesman Problem (TSP). Unfortunately, the TSP is NP-hard and cannot be solved optimally in a reasonable amount of time in this case. Therefore, many heuristics are proposed in the literature, for instance: S-shape, largest-gap, etc (Caron, 1998; Petersen, 1997). Although those heuristics can deal with the basic routing problem, they are not designed to take into account additional constraints that arise during the reshuffle (Muralidharan, 1995). For instance, an item that is to be shifted cannot be dropped off at a certain storage location before having been picked up at another storage location. As a consequence, the authors have opted for a more flexible heuristic, the Nearest Neighbour Heuristic (Winston, 1994), which they have slightly modified in order to include those additional constraints.

## 4. DYNAMIC ABC HEURISTIC

The dynamic ABC heuristic basically performs two tasks through an iterative procedure. Firstly, it reviews the current classification and determines which items have to be assigned to another class. Secondly, it determines the new storage locations of those items. The iterative procedure is repeated until a feasible and beneficial solution is obtained. Both tasks are closely interrelated, as depicted in Figure 3. The Figure is explained in detail below. The key characteristic is that the dynamic ABC heuristic is based on the past evolution of the number of order lines per item and per day (previously referred to as "demand patterns" for the sake of simplicity). Although the primary objective of the heuristic is to reduce the average travel time per day, the real objective is to reduce the average working time per day, that is to say, the sum of the average order picking time and reshuffle time per day.

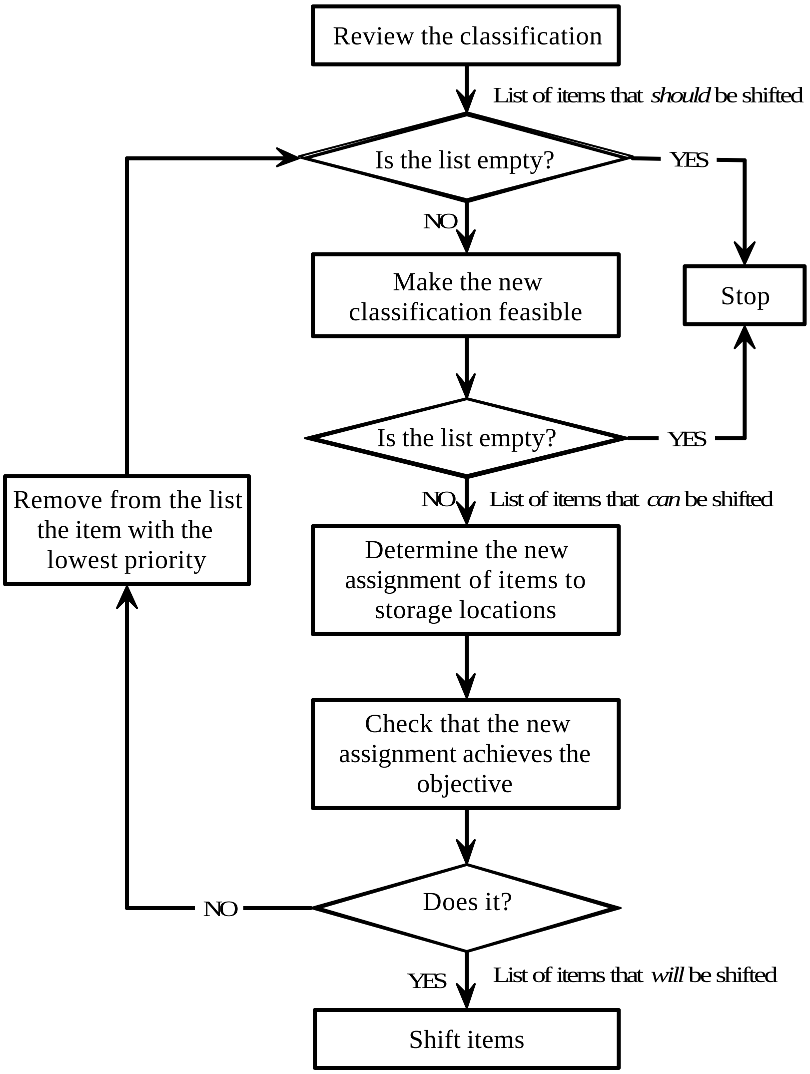


**Figure 3. Dynamic ABC Storage Policy**

### 4.1 Review the Classification

During this step, we determine the list of items that are not stored in the right storage area (i.e. that are misclassified), as well as their new storage area (i.e. their new class). Every item of this list also receives a priority.

The review of the classification is illustrated by the chart of Figure 4, which represents the evolution of the number of order lines per item and per day (only one item is represented). Every day, two limits $l_{AB}$ and $l_{BC}$ are also plotted. Those limits define three zones, corresponding to the three classes A, B and C. They are computed in the following way. First, an ABC analysis is performed on the number of order lines per item of that day. Then, based on this analysis, the last item of class A ($a_2$), the first of class B ($b_1$), the last of class of B ($b_2$) and the first of class C ($a_3$) are identified and their numbers of order lines of that day are recorded, respectively: o($a_2$), o($b_1$), o($b_2$) and o($a_3$). Finally, the limits are calculated according to:

$l_{AB} = (o(a_2) + o(b_1)) / 2$
$l_{BC} = (o(b_2) + o(a_3)) / 2$

We define a threshold value $n$. To explain how the classification is reviewed, we consider a certain item $i$. We suppose that this item belongs to the class $c_i$ (either A, B or C) and is currently stored in the area allocated to the class $c_i$. We look at the last $n$ dots of the chart that correspond to this item. If those $n$ dots are not in the zone of the chart corresponding to the class $c_i$, the item $i$ is misclassified. The new class $c_i$' is determined by the position of the last dot of the chart. In other words, as a result of the evolution of the number of order lines per item and per day, the item $i$ *should be* shifted to the storage area corresponding to the class $c_i$'.

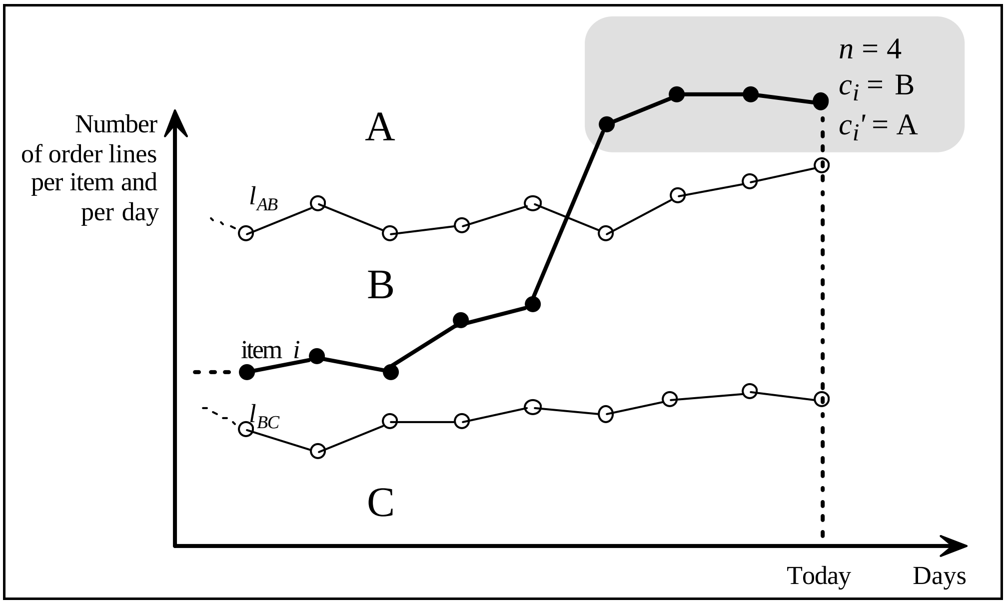


**Figure 4. Review of the ABC Classification**

As explained below, it may happen that some items, which have been identified as misclassified, cannot be shifted to the storage area corresponding to their new class. In that case, they will remain in their former class and they will not be shifted. In order to select the items that *will be* shifted among the items that *should be* shifted, every item $i$ receives a priority $p_i$, which is the product of three components $p_{1,i}$, $p_{2,i}$ and $p_{3,i}$. Each component can get a value between 1 and 3.

Furthermore, $p_{1,i}$ accounts for the urgency of the shift, which is a function of the number $n_i$ of consecutive dots that are not in the zone corresponding to the class $c_i$ ($n_i \geq n$)

The aim of $p_{1,i}$ is to favour the shift of items that should have already been shifted, but have not yet been shifted for any reason. On the other hand, $p_{2,i}$ accounts for the importance of the shift, which is a function of the average distance on the chart between the $n_i$ dots and the limit $l_{AB}$ or $l_{AC}$. $p_{2i}$ aims at favouring the shift of items whose number of order lines per day has recently changed significantly. Finally, $p_{3,i}$ accounts for the variability of item *i*'s number of order lines per day, which is a function of the standard deviation $\sigma_i$ of item *i*'s number of order lines per day. The aim of $p_{3,i}$ is to unfavour the shift of items whose number of order lines per day is historically known to be greatly fluctuating, since such shifts may become rapidly outdated whereas they may require a lot of working time.

### 4.2 Make the New Classification Feasible

During this step, we ensure that the items of the list that has been previously determined can actually been shifted to their new storage area.

To check the feasibility of the suggested shifts, several factors have to be calculated: the number $s_{1,i}$ of storage locations that are already empty in the storage area allocated to the class *i* before any shift; the number $s_{2,i}$ of storage locations that become empty in the storage area allocated to the class *i* because some items of the list are shifted from those locations to their new storage location; the number $s_{3,i}$ of storage locations that become occupied in the storage area allocated to the class *i* because some items of the list are shifted from their former storage location to those locations. Then, the following value is calculated for every class *i*, $s_i = s_{1,i} + s_{2,i} - s_{3,i}$.

If all $s_i$ are positive ($\geq 0$), the suggested shifts are feasible and we proceed to the next step. If at least one $s_i$ is negative ($< 0$), the suggested shifts are not feasible because we are short of one or more free storage locations in one or more storage areas. In that case, we select the class *j* with the most negative value $s_j$ and we remove from the list of items that should be shifted to the storage area allocated to the class *j* the $|s_j|$ ones with the lowest priority. Finally, we update all $s_i$. The procedure is repeated until all $s_i$ are positive. The new classification is then feasible.

### 4.3 Determine the New assignment of Items to Storage Locations

During this step, we assign the items of the list that has been previously processed to their new storage location (inside their new storage area).

We can assign items to occupied storage locations that become empty, or to storage locations that are already empty. To-be-emptied storage locations are chosen first. If there is no to-be-emptied storage location, empty storage locations are chosen instead. The objective is to minimize the total processing time and the total acceleration/deceleration time, since swaps (an item takes the place of another: a pick-up and a drop-off operation are combined) are generally less time-consuming than the corresponding single operations (either a pick-up or a drop-off operation). Furthermore, items are assigned to storage locations within storage areas according to their number of order lines, as it is done in a volume-based storage policy. The aim is to minimize the total travel time.

### 4.4 Check that the New Assignment Achieves the Objective

During this step, we check that the new assignment of items to storage locations is beneficial.

Several criteria can be used. Those criteria may be based on the time required to carry out the new assignment, that is to say, to shift items ($T_{reshuffle}$) and on the average order picking time per day ($T_{picking}$, which is actually a moving average). The way $T_{picking}$ and $T_{reshuffle}$ are computed has been briefly addressed previously in the text.

The criterion used in the case study described in the next section is $T_{reshuffle} < T$, where T is a threshold ensuring that the reshuffle is limited in time.

A variant of the previous criterion is $T_{reshuffle} < a\ .\ T_{picking}$, where α (e.g. 5%) is a fraction that relates the reshuffle activity to the order picking activity.

A more sophisticated criterion is $T_{reshuffle} < a\ .\ (w\ .\ h - T_{picking})$ where *w* is the number of workers, *h* the number of working hours per day per worker and α (e.g. 90%) a fraction that relates the reshuffle time to the workers' idle time (the time during which they are not busy with the orders).

While the first criterion is very simple to apply, it does not take into account the relationship between the reshuffle activity and the order picking activity. On the other hand, the third criterion offers a simple model of this relationship, but is much more difficult to apply, since it requires the knowledge of the workforce plan. Other criteria can easily be designed in order to include other aspects such as the possibility of working overtime at a different hourly wage.

If the criterion is not satisfied, it means that the reshuffle takes too much time. In order to shrink the reshuffle time, an item that should be shifted will be removed from the list, as shown in Figure 3. This item will be the one with the lowest priority.

## 5. CASE STUDY

The dynamic variant of the ABC storage policy that has been described in the previous section is a heuristic and not the result of a mathematical optimisation. Its effectiveness is therefore not guaranteed. The dynamic ABC heuristic will be considered as effective if $T_{picking}(dynamic) + T_{reshuffle} < T_{picking}(static)$ for a certain period of time (e.g. half a year). In this criterion, we compare the traditional static variant with the new dynamic variant.

In order to check the effectiveness of the dynamic ABC heuristic in various situations, the authors have developed a simulator. This program has been written in Matlab and simulates the reshuffle and the order picking activities during a certain period of time (e.g. half a year). The orders and associated order lines used by this simulator are provided by real-life cases. One such case is described below as illustration.

### 5.1 Presentation of the Case Study

The data of this case study come from the durable consumer goods business. The main characteristics are given in Table II. Other characteristics (i.e. dimensions of the warehouse, speed of the order picker, administrative, acceleration/deceleration and processing times) are also reproduced in Table 2. The chart of Figure 5 shows the fluctuating nature of the number of order lines per day of some items, which tends to advocate the use of dynamic variants of the ABC storage policy. However, it is important to note that not all items of the data set have such an erratic demand and that a majority of them experience a rather sTable demand.

**Table 2. Data of the Case Study**

| | |
|---|---|
| Data | |
| Number of days | 127 |
| Period | 1/8/00 – 28/12/00 |
| Number of items | 9896 |
| Number of orders per day: mean | 1931 |
| Number of orders per day: std deviation | 934 |
| Number of order lines per order: mean | 8.8 |
| Number of order lines per order: std deviation | 10.9 |
| Order picker | |
| Speed (m/s) | 0.8 |
| Administrative time (s) | 60 (order picking) – 30 (reshuffle) |
| Acceleration/Deceleration time (s) | 5 (per stop) |
| Processing time (s) | 2 (order picking) – 4 (reshuffle) (per item) |
| Warehouse | |
| Number of aisles | 20 |
| Number of blocks | 3 |
| Number of spots per sub-aisle | 25 |
| Number of storage locations per spot | 4 |

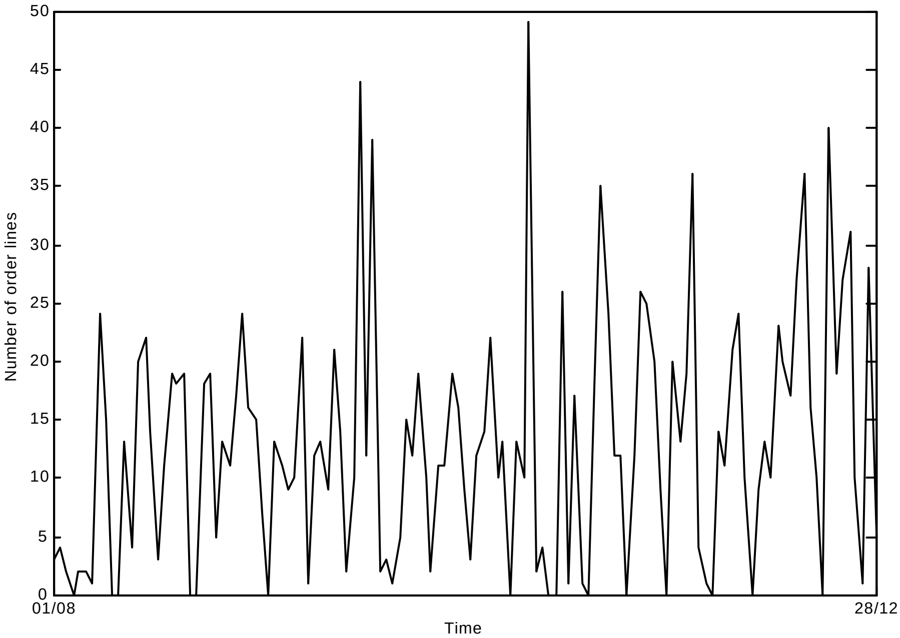


**Figure 5. Typical Erratic Demand Pattern**

### 5.2 Presentation of the Simulations

Eleven simulations are presented here. Their common characteristics (i.e. layout of the warehouse, parameters of the ABC storage policy, order picking and reshuffle policies, etc) have been described through the previous sections. Their specific characteristics are given in Table 3.

**Table 3. Details of the Simulations**

| Simulations | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Frequency of the review (days) | / | / | 1 | 2 | 5 | 30 | 2 | 2 | 2 | 2 | 2 |
| $N$ | / | / | 2 | 2 | 2 | 2 | 1 | 3 | 4 | 5 | 2 |
| $T$ | / | / | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ | 7200 |
| Init. storage policy | Rnd | ABC | ABC | ABC | ABC | ABC | ABC | ABC | ABC | ABC | ABC |

The first simulation uses a random storage policy and can be used to assess the performance of the ABC storage policy. The second one uses the static variant of the ABC storage policy. It can be compared to the other simulations, which are based on the dynamic ABC heuristic. Simulations 3, 4, 5 and 6 study the influence of the frequency at which the dynamic ABC heuristic is executed, that is the frequency of the review of the ABC classification. Simulations 7, 4, 8, 9 and 10 examine the influence of the threshold value $n$, which is used to identify misclassified items. Finally, simulations 4 and 11 evaluate the influence of the parameter T, which limits the time devoted to the reshuffle activity.

### 5.3 Results of the Simulations

The results are presented in Figure 6 according to the equation written at the beginning of this section.

First of all, we notice that the ABC storage policy (simulation 2) is better than the random storage policy (simulation 1) in terms of average order picking time per day. Likewise, the dynamic variant of the ABC storage policy (simulations 3 to 11) developed by the authors performs better than its static variant (simulation 2), even if we take into account the additional reshuffle time.

Secondly, we observe that if the reshuffle time increases, the order picking time decreases, and vice versa. As a consequence, the average working time per day, which is the sum of those two components, is not monotonous. The parameters that will yield its minimum value will definitely vary from one case study to another. In this case, a period of 5 days (simulation 5) and a parameter $n$ equal to 4 (simulation 9) turn out to be a reasonable choice.

Finally, we check that the average order picking time per day decreases if we review the classification more often (simulations 3 to 6), or if $n$ has a low value (simulations 7 to 10 and 4), which corresponds to a smaller "inertia" of the system. On the other hand, we confirm that limiting the time available for the reshuffle (simulation 11) has a bad impact on the average order picking time.

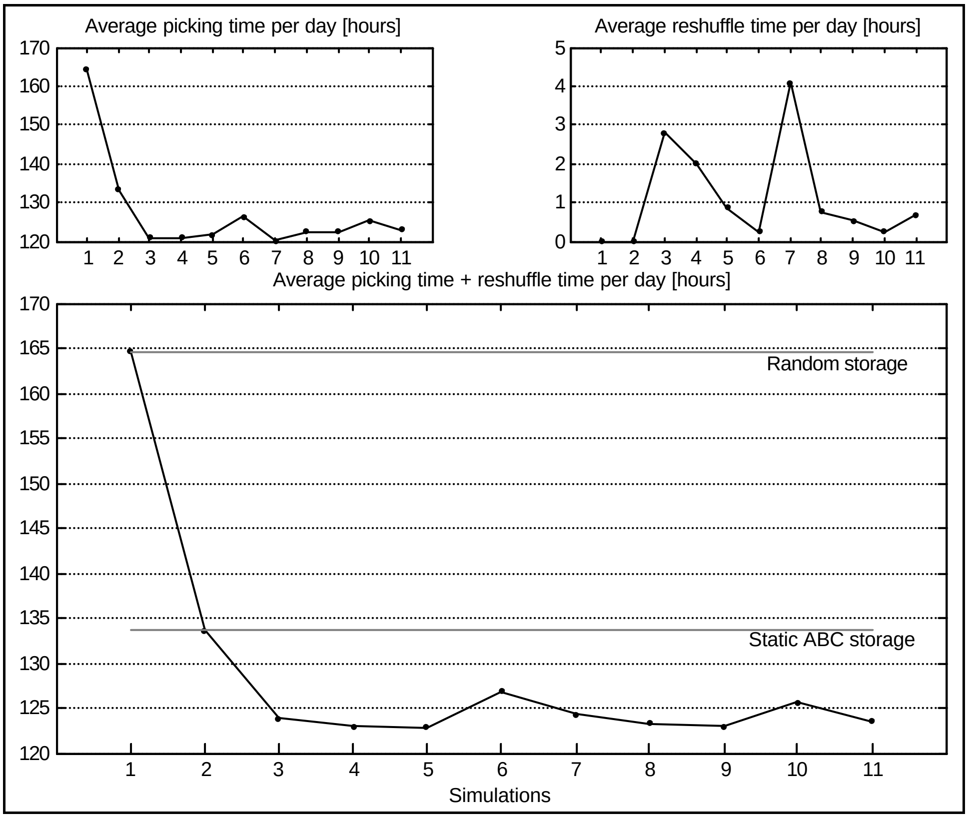


**Figure 6. Results of the Simulations**

## 6. CONCLUSIONS AND FURTHER RESEARCH

It is well known that the order picking activity is labour-intensive and costly. Many attempts have been made to improve its performance. This paper describes a new approach that is especially suiTable for today's erratic demand environments. This approach consists of a dynamic variant of the ABC storage policy.

This dynamic variant takes the form of a heuristic. The performance of a heuristic is never guaranteed. However, the authors have checked through various case studies that the dynamic ABC heuristic is a promising technique. Besides, they have noticed that its parameters need to be adapted with care to each particular situation in order to reveal the best of its essence. In short, the authors are convinced that dynamic variants of the ABC storage policy deserve further attention in the future.

First, straightforward variations and improvements of the dynamic ABC heuristic developed by the authors should be studied. Then, related aspects should be considered, for example the reshuffle policy (which organizes the reshuffle of the warehouse based on a list of items to be shifted), the capacity of the warehouse (in terms of space and workforce), the human intervention on decisions made by computers, etc. Finally, when a fully realistic dynamic ABC storage policy will have been developed, it should be validated in real-life situations, on the field.

## REFERENCES

Bulfin, R.L., and D. Sipper, 1998. *Production Planning, Control, and Integration*, McGraw-Hill.

Caron, F., G. Marchet, and A. Perego, 1998. "Routing policies and COI-based storage policies in picker-to-part systems", *International Journal of Production Research*, Vol. 36, No. 3, pp. 713-732.

Muralidharan, B., J.L. Richard, and R. Pandit, 1995. "Shuffling Heuristics for the Storage Location Assignment in an AS/RS", *International Journal of Production Research*, Vol. 33, No. 6, pp. 1661-1672.

Petersen, C.G., "Implementation issues of class-based storage strategies", *Working Paper*, Northern Illinois University.

Petersen, C.G., 1997. "An evaluation of order picking routeing policies", *International Journal of Operations and Production Management*, Vol. 17, No. 11, pp. 1098-1111.

Petersen, C.G. and K.J. Roodbergen, 1999. "How to improve order picking efficiency with routing and storage policy", *Perspectives on Material Handling Practice*.

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

Roodbergen, K.J., 2001. "Layout and Routing Methods for Warehouses", *PhD thesis*, Erasmus University Rotterdam.

Winston, W.L., 1994. *Operations Research: Applications and Algorithms*, Thomson.