Optimizing the Storage Assignment in a Warehouse Served by Milkrun Logistics

András Kovács
Computer and Automation Research Institute, Budapest, Hungary
E-mail address: akovacs@sztaki.hu

June 23, 2009

Abstract

This paper addresses the problem of storage assignment in a warehouse characterized by multi-command picking and served by milkrun logistics. In such a logistic system, vehicles circulate between the warehouse and the production facilities of the plant according to a pre-defined schedule, often with multiple cycles (routes) serving different departments. We assume that a request probability can be assigned to each item and each cycle, which leads to a special case of the correlated storage assignment problem. A MIP model is proposed for finding a class-based storage policy that minimizes the order cycle time, the average picking effort, or a linear combination of these two criteria. Computational experiments show that our approach can achieve an up to 36-38% improvement in either criterion compared to the classical COI-based strategy.

Keywords: Storage assignment, correlated, warehousing, milkrun, mixed-integer programming.

1 Introduction

The storage assignment problem involves the placement of a set of items in a warehouse in such a way that some performance measure is optimal. We investigate this problem in the context of a warehouse from where items are delivered to the production departments of the plant by milkrun logistics, i.e. vehicles circulating according to pre-defined schedules. It is assumed that each milkrun cycle serves the material requirements of different departments, and departments assemble diverse sets of end products, which implies that each milkrun cycle has a different probability of requiring an item. We consider the milkrun routes to be part of the input, and hence, disregard the vehicle routing aspect of the problem. Item requests are processed simultaneously by multiple human order pickers.

The most important performance measures in a warehouse are generally related to the time or effort required for order picking, i.e. the retrieval of items from the shelves and delivering them to the point where they will be picked up by the appropriate vehicle. Informally, these performance measures can be optimized by placing high-runner items near to the entrance of the warehouse, and by storing items that are often ordered together close to each other. Technically, this problem is a special case of correlated storage assignment, where the organizational structure behind the correlation among request probabilities of different items is known and can be exploited by a mathematical model. We present a mixed integer programming (MIP) formulation of the problem that can be solved by commercial software in practically relevant problem sizes.

We show that minimizing order cycle time (maximum of individual pickers' times) and minimizing expected picking effort (sum of pickers' times) are conflicting criteria. Our MIP allows controlling the trade-off between these criteria by minimizing a linear combination of the two, or minimizing one of them subject to an upper bound on the other. The novel strategy has been compared to classical cube-per-order index-based (COI) techniques in computational experiments, where it achieved an up to 36-38% improvement compared to COI, according either criterion.

1.1 A Motivating Industrial Application

Our research was motivated by an industrial application in a warehouse of an electronics manufacturer. The warehouse serves directly the requests arriving from a couple of manufacturing and assembly departments located on the same site. The examined warehouse stores a huge variety of small electronic components, some of them consumed in large quantities by the plant. Large and heavy goods, packaging material, or items that require special handling are stored in separate warehouses. At the time of writing this paper, the warehouse is about to be extended significantly. A decision support tool is looked for to compute the best storage policy for the initially almost empty warehouse. The main objective is minimizing the picking effort needed for retrieving the requested items.

The warehouse is located in a spacious hall. After the extension, it will be divided to four floors, with each floor consisting of a matrix structure of ca. 10 aisles and 4 cross-aisles. All items are stored in uniform boxes that can be easily handled by a human picker. However, the company is planning to apply a couple of different box sizes in the future to better match the characteristics of different items. The total number of storage places approaches 10^5, while the number of different items stored is expected to increase to 5000 in the future. The stock held of a given item occupies 1-100 storage places, but only a single oldest box is opened for picking. Therefore, stock splitting is not applicable.

The requested items are retrieved and delivered to the single entrance of the warehouse simultaneously by a group of human order pickers. For the sake of clear responsibilities, disjoint areas of the warehouse are assigned to individual pickers. At the entrance of the warehouse, the picked items are loaded onto little trains, which circulate according to a pre-defined periodic schedule in a milkrun system. The schedule contains 4 train departures – so-called milkrun cycles – in each hourly period. Each cycle serves the material requirements of different departments of the factory. About the half of the items are consumed in several milkrun cycles. It is possible to estimate the probability p_i^k that a pick list in cycle k contains item i based on historic data, with occasional adjustments based on forecasts when the product mix changes. However, more precise data, e.g. the covariance of the random variables cannot be reliably extracted from this data. Stock level forecasts are, however, available.

1.2 Related Literature

The storage assignment problem involves deciding where and how to store a set of items in order to ensure optimal operation of the logistic system (de Koster et al., 2007). Since order picking is the most critical and laborious operation in the warehouse, it is common to optimize a measure of order picking performance, e.g. to minimize expected picking (or travel) times, or order cycle times. The two main families of storage strategies are the *dedicated* and the *shared* strategies. Dedicated strategies store always the same item in the same slot. For this kind of strategy and single-command picking, Heskett (1963) proved that the *cube-per-order index* (COI) policy minimizes the average picking (or travel) time. This policy sorts the items by increasing COI, i.e. the ratio of the stock volume to the demand rate, and then places them sequentially to the closest free slots to the entrance. The reciprocal of COI is called the turnover rate.

Shared storage strategies, in contrast, do not reserve slots for specific items, which makes them more convenient when stock levels change over time. The most important representatives of shared storage strategies are *class-based* storage strategies (Hausman et al., 1976; Petersen and Aase, 2004). These form classes of items and partition the warehouse floor to zones, and finally assign each class of items to a specific zone. Storage within a zone is random. In the case of single-command (single-item) picking, it is typical to apply 2–6 zones, and define classes based on the turnover rates of the items. In order to compensate for inventory level changes, some empty reserve slots are kept in each zone. The positioning of zones is addressed, e.g. in (Le-Duc and De Koster, 2005).

An alternative to class-based strategies is the *duration of stay* (DoS) policy (Goetschalckx and Ratliff, 1990), which places different units of the same item to different zones, depending on their expected duration of stay. Hence, when a large shipment of an item arrives, a smaller quantity that will be used shortly is stored near to the pick-up point, while the rest is put further away. The DoS policy is optimal for single-command systems with perfectly balanced input and output, i.e. when the quantity of arriving and departing items is the same for each set of items that share the same turnover rate. However, (Kulturel et al., 1999) has shown in simulation experiments that under more realistic assumptions, class-based strategies outperform the DoS policy.

The case of multi-command picking (or multi-item orders) opens new grounds for optimization: items that are frequently ordered together can be stored near to each other in order to save travel time. This question is addressed in different versions of the *correlated storage assignment problem* (CSAP) (Garfinkel, 2005). Since most variants of CSAP are too difficult to be solved by exact methods for practically relevant problem sizes, various heuristics have been studied. A general approach is the cluster-first / zone-second heuristic (Frazelle and Sharp, 1989): items are first clustered according to some measure of correlation, and then clusters are assigned to zones based on their turnover rates. However, this decomposition may cause difficulties when there are multiple storage types, or zones with different capacities and access times. Typically, a characterization of correlations is extracted from historical data, but in some applications product structure can also be used (Brynzér and Johansson, 1996).

Multi-command picking is often combined with multiple order pickers, each assigned to disjoint zones, working simultaneously on the same order. This extension results in a situation where the classical optimization criteria are in conflict. Roughly speaking, placing the items with positive correlation into the same zone decreases expected total travel times, but may increase order cycle times, while distributing them evenly does vice versa. The storage assignment problem with multi-command picking has been investigated with the criteria of minimizing the travel time between two products in the same order (Frazelle, 1990), minimizing total travel time (Brynzér and Johansson, 1996), minimizing the number of multi-zone orders (Garfinkel, 2005), and balancing the workload among pickers, i.e. assigning the same number of items from each order to each zone (Jane and Laih, 2005).

For correlated assignment with a dedicated storage policy, Mantel et al. (2007) introduced the *order oriented slotting* strategy and some corresponding heuristics. The problem is examined by Sadiq et al. (1996) in a dynamic setting, with a periodic revision of the assignments due to the variation of item turnovers over time. Hence, decisions are made on re-warehousing (re-locating) the items.

The concept of milkrun logistics originates from the dairy industry. The notion covers a transportation network where all input and output (I/O) material requirements of several stations are covered by one vehicle that visits all these stations, and circulates according to a pre-defined schedule (Baudin, 2005). This transportation concept is economical when the I/O volume of each single station is essentially smaller than a truckload. The milkrun concept is frequently applied in internal plant logistics to transport raw materials, finished goods, and waste between manufacturing or assembly stations and the warehouses of the plant. Often, the plant is too large to cover it by a single milkrun cycle (single vehicle route), and therefore multiple cycles are applied. This leads to a special type of correlation among the request probabilities of items in the warehouse. To the best of our knowledge, this special case of the correlated storage assignment problem has not been studied yet in the literature.

2 Problem Definition

Below we give a formal definition of the storage assignment problem for the above warehouse configuration and milkrun logistic system, which we will denote as SAP-MR. A class-based storage strategy is looked for, with zones defined and order pickers assigned to zones a priori.

Let us assume that the warehouse floor is divided into $P \times B$ disjoint zones, where P is the number of order pickers, and B is the (maximum) number of zones that belong to a picker. Each zone $\langle p, b \rangle$ (with $p = 1, ..., P$ and $b = 1, ..., B$) has a capacity of $C_{p,b}$ completely identical slots (with $C_{p,b} = 0$ allowed, which means that some of the pickers have less than B zones assigned). The warehouse has a single entrance. There is a set of N items to be placed in this warehouse. Each item i must be assigned to a single zone, where it will occupy s_i slots.

Orders are generated to this warehouse in accordance with the fixed milkrun schedule: each time before a vehicle departs, an order is created that contains all the items that are requested at the destinations of the vehicle. Each vehicle departure belongs to one of the K milkrun cycles. The probability that item i requested in milkrun cycle k is denoted by p_i^k, and it is assumed that p_i^k is independent of the probabilities that characterize other items or other cycles. It is assumed that the ordered quantity of an item does not influence essentially the picking time of the item.

An order is handled by all the pickers simultaneously: each picker retrieves the items stored in the zones assigned to him, and carries the items to the warehouse entrance using a cart with unlimited capacity. The order picking time of each individual picker is composed of the access times $h_{p,b}$ to the visited zones $\langle p, b\rangle$ and a constant retrieval time e for each item picked. It is assumed that the zones of each picker are located sequentially behind each other. Hence, if there is an item to be picked in zone $\langle p, b\rangle$, then all the zones $\langle p, b'\rangle$ with $b' \leq b$ must be crossed (accessed) to pick it. Accordingly, $h_{p,1}$ stands for the double travel time between the entrance and zone $\langle p, 1\rangle$, while $h_{p,b}$ with $b \geq 2$ denotes the double travel time between zones $\langle p, b-1\rangle$ and $\langle p, b\rangle$.

The SAP-MR problem looks for finding an optimal class-based storage strategy, i.e, an assignment of items to zones. We consider SAP-MR with two different performance measures: *minimizing order cycle time*, T, which equals the maximum of the expected picking times of individual pickers; and minimizing *average picking effort*, W, which is the sum of the individual picker's picking times. These two optimization criteria are in conflict, as it will be illustrated in an example in Section 2.1. Furthermore, they can have different relative importance in different applications. Consequently, we might wish to minimize one of these criteria, a weighted sum of the two, or potentially minimize one criterion subject to an upper bound on the other. In this paper, we focus on minimizing the weighted sum, but it is straightforward to adapt the model to any of the previous scenarios.

The example in Figure 1 presents the schematic drawing of a warehouse oper-

Figure 1: A warehouse operated by 3 pickers, with 3 zones for each picker. Physically, each zone is composed of a matrix structure of aisles and cross-aisles.

ated by 3 order pickers. The notation used in this paper is summarized in Table 1. Possible extensions to this basic model will be discussed later in Section 5.

2.1 Comparison to Classical Models

The solution of SAP-MR is an assignment of items to pre-defined zones, i.e., a classical class-based storage policy that can be implemented easily in existing ERP systems. Nevertheless, our approach computes this policy as a solution of an optimization problem incorporating a rather sophisticated warehouse model. This leads to significantly better performance compared to classical policies that classify the items solely by their turnover rates.

We model explicitly the times required for each individual picker to retrieve the required items from the warehouse. This allows us to measure the warehouse performance from several aspects, and to find the most suitable tradeoff between the different objectives. Namely, in this study we consider the order cycle time and the average picking effort, which are conflicting criteria. To illustrate this, let us consider a warehouse with two pickers and two zones for each picker, and a single milkrun cycle. Assume there are two items, each of them occupying a

Dimensions	
N	Number of items
P	Number of pickers
B	Number of zones per picker
K	Number of milkrun cycles
Parameters	
p_i^k	Request probability for item i in cycle k
s_i	Capacity requirement of item i
$C_{p,b}$	Capacity of zone $\langle p, b\rangle$
$h_{p,b}$	Access time to zone $\langle p, b\rangle$
e	Unit item retrieval time
α_T	Weight factor of the picking time criterion
α_W	Weight factor of the picking effort criterion
Variables	
$x_{p,b}^i$	Binary variable indicating whether item i is assigned to zone $\langle p, b\rangle$
$q_{p,b}^k$	Probability of accessing zone $\langle p, b\rangle$ in cycle k
$a_{p,b}^k$	Auxiliary binary variable, indicates whether the saturated sum approximation of $q_{p,b}^k$ equals 1
$r_{p,b}^k$	Expected number of items picked from zone $\langle p, b\rangle$ in cycle k
t_p^k	Picking time for order picker p in cycle k
T	Order cycle time
W	Average picking effort

Table 1: Notation used in the paper.

complete zone. Let $e = 1$, $h_{1,1} = h_{2,1} = 5$ and $h_{1,2} = h_{2,2} = 1$. Now, there are two reasonable solutions to this problem. On the one hand, assigning both items to the same picker (e.g., picker 1) results in $t_1 = 8$ and $t_2 = 0$, hence, $T = 8$ and $W = 8$. On the other hand, assigning the items to different pickers leads to $t_1 = t_2 = 6$, hence, $T = 6$ and $W = 12$, as illustrated in Fig. 2. Therefore, SAP-MR is a multi-criteria optimization problem. Our model allows to control the tradeoff between the two criteria by minimizing their linear combination, or by minimizing one of them subject to an upper bound on the other.

On the other hand, the consideration of milkrun cycles leads to a special case

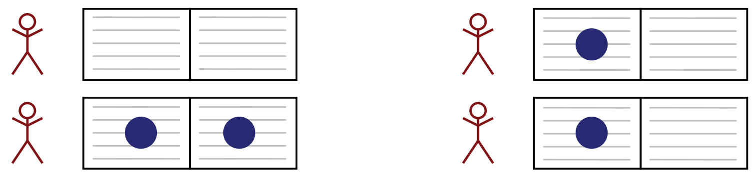

Figure 2: Two solutions of a sample problem. In solution (a), $T = 8$ and $W = 8$, whereas in solution (b), $T = 6$ and $W = 12$. Hence, T and W are conflicting criteria.

of the correlated storage assignment problem (CSAP). Exploiting the correlation among request probabilities of different items by a mathematical model helps achieve better warehouse performance than with classical models that disregard any relation among the items.

Since milkrun logistics are commonly applied in internal plant logistics, SAP-MR is a practically relevant special case. Moreover, data for characterizing the correlation among items can be extracted reliably from existing databases, and as it will be shown, the resulting problem can be solved by commercial MIP solvers in reasonable time.

3 A Mathematical Programming Approach

3.1 Estimating Picking Times

With a given storage assignment, the probability that item i will be picked from zone $\langle p, b\rangle$ in cycle k is $p_i^k x_{p,b}^i$. This zone must be accessed by the picker if and only if there is an item to be picked in this zone, or another zone behind it. Hence, the probability that zone $\langle p, b\rangle$ must be accessed in cycle k equals

$$q^k_{p,b} = 1 - \prod_{b' \geq b} \prod_i (1 - p^k_i x^i_{p,b'}). \tag{1}$$

Furthermore, the expected number of items that must be picked from zone $\langle p, b \rangle$ equals

$$r^k_{p,b} = \sum_i p^k_i x^i_{p,b}. \tag{2}$$

Hence, the expected picking time of picker p in cycle k is

$$t^k_p = \sum_b (h_{p,b} q^k_{p,b} \; + \; e r^k_{p,b}). \tag{3}$$

Then, the average picking effort is the sum of the expected picking times of individual pickers,

$$W = \frac{1}{K} \sum_k \sum_p t^k_p. \tag{4}$$

The order cycle time is the expected maximum of the picking times of individual pickers. To obtain T, we estimate the expected value of the maximum by the maximum of the expected values:

$$T \approx \max_k \max_p t^k_p. \tag{5}$$

Most of the above equalities can be handled well in a mixed-integer programming (MIP) model, except for (1). To overcome this difficulty, we apply the first-order saturating sum approximation of $q^k_{p,b}$ as follows:

$$q^k_{p,b} \approx \min(\sum_{b' \geq b} \sum_i p^k_i x^i_{p,b'}, \; 1). \tag{6}$$

Note that the above approximation always overestimates the probability of accessing the zones. Nevertheless, when this probability is either close to 0 or 1, this approximation is sufficiently precise. On the other hand, our model may underestimate T.

3.2 A MIP Model

The MIP model of SAP-MR is presented below.

Minimize

$$\alpha_W W + \alpha_T T \tag{7}$$

subject to

$$W = \frac{1}{K} \sum_k \sum_p t_p^k \tag{8}$$

$$T \geq t_p^k \qquad \forall p, k \tag{9}$$

$$\sum_p \sum_b x_{p,b}^i = 1 \qquad \forall i \tag{10}$$

$$\sum_i s_i x_{p,b}^i \leq C_{p,b} \qquad \forall p, b \tag{11}$$

$$t_p^k = \sum_b (h_{p,b} q_{p,b}^k \;+\; e r_{p,b}^k) \qquad \forall p, k \tag{12}$$

$$r_{p,b}^k = \sum_i p_i^k x_{p,b}^i \qquad \forall p, b, k \tag{13}$$

$$q_{p,b}^k \geq a_{p,b}^k \qquad \forall p, b, k \tag{14}$$

$$a_{p,b}^k \geq \frac{1}{M} (\sum_{b' \geq b} \sum_i p_i^k x_{p,b'}^i - 1) \qquad \forall p, b, k \tag{15}$$

$$q_{p,b}^k \geq \sum_{b' \geq b} \sum_i p_i^k x_{p,b'}^i \;-\; M a_{p,b}^k \qquad \forall p, b, k \tag{16}$$

$$x_{p,b}^i \geq 0 \qquad \forall p, b, i \tag{17}$$

$$a_{p,b}^k \in \{0, 1\} \qquad \forall p, b, k \tag{18}$$

The model addresses the minimization of a linear combination of the average picking effort, W, and the order cycle time, T (7). The former is computed as the sum, while the latter as the maximum of the individual pickers times (8 and 9). Each item must be assigned to exactly one zone (10), in such a way that the capacity constraints are respected (11). Equality (12) states that the picking time of individual pickers is composed of zone access times and unit item retrieval times. Constraint (13) describes how the expected number of items to be picked is computed for individual pickers. Finally, inequalities (14–16) encode the saturated sum approximation of the probability of having to access a given zone in a given milkrun cycle. Parameter M is a sufficiently large number, e.g.

$$M = \max_{p,b} C_{p,b} \, \max_{i,k} \frac{p_i^k}{s_i}. \tag{19}$$

For efficiency reasons, we allow variables $x_{p,b}^i$ to be fractional. This corresponds to splitting the stock (and also the request probabilities) of item i among several zones. Nevertheless, typical MIP solution methods, applying simplex-based algorithms for solving the LP sub-problems in a branch-and bound search, ensure that relatively few items will be split, which can be handled by rounding without major difficulties.

3.3 Discussion and Extensions

The SAP-MR problem, as it has been stated above, is NP-hard. This can be shown by reduction from the two-partition problem. Note however, that the complexity status depends on various details of the model. If the saturating sum approximation (6) is replaced by a maximum approximation ($q_{p,b}^k \approx \max_i p_i^k x_{p,b'}^i$), then the MIP reduces to an LP, and the problem becomes solvable in polynomial time.

We do not suggest, however, the use of the maximum approximation, because it does not capture an essential characteristic of the request probabilities, namely that the joint probability of ordering at *least one* of the many low-turnover items from a zone can be high. At the same time, if the assignment variables, $x^i_{p,b}$, were constrained to be integral, then the problem would be NP-complete independently of the approximation applied, because it would contain bin packing as a sub-problem.

The above presented MIP generalizes naturally to more realistic warehouse models and picking processes, including

- various slot types; this requires stating the capacity constraints (11) separately for each slot type in the MIP.[1]
- different milkrun cycles occurring with different frequencies; this implies that different milkrun cycles have to be considered with different weights when computing the average picking effort (see Eqn. 8);
- different pickers having different number of zones assigned;

4 Computational Experiments

The performance of the proposed approach to SAP-MR has been investigated in computational experiments from two aspects: the improvement it achieved compared to a COI class-based strategy, and the computational effort required.

The experiments were performed on randomly generated problem instances. The parameters of our problem generator were the number of items ($N \in \{1000,$

[1]Note that the problem cannot be decomposed by slot types, since the sub-problems would be still interconnected through the access probabilities of the zones.

2000, 3000}), the number of pickers ($P \in \{2, 4, 6\}$), the number of zones per picker ($B \in \{1, 2, 3\}$), and the warehouse completeness ($\gamma \in \{0.6, 0.9\}$) that indicates the ratio of the total capacity requirement of the items to the warehouse capacity. The number of milkrun cycles was fixed to $K = 4$. We also fixed the unit item retrieval time to $e = 1$, and used four different schemes for determining the access times to zones (see Table 2). Generating 5 instances with all the 216 possible combinations of the above parameters resulted in 1080 problem instances altogether. For each problem instance, we computed the optimal storage assignments corresponding to the scenarios shown in Table 3 using the proposed MIP approach, and a reference solution using COI.[2]

The proposed MIP has been implemented in ILOG CPLEX version 9.1. The experiments were run on a 1.86 GHz Intel Xeon computer with 2 GB of RAM under a MS Windows Server 2003 operating system, with a time limit of 1200 CPU seconds for each instance and each strategy.

N	Access times	Corresponding warehouse structure
δ_1	$h_{p,b} = 10 + 5b$	All pickers on the same floor, quick access
δ_2	$h_{p,b} = 20 + 10b$	All pickers on the same floor, slow access
δ_3	$h_{p,b} = 10p + 5b$	Pickers on different floors, quick access
δ_4	$h_{p,b} = 20p + 10b$	Pickers on different floors, slow access

Table 2: Access time schemes.

4.1 Comparison to the COI class-based strategy

For the comparison of the different strategies, the values of T and W originating from the MIP solutions were used. Note that when implementing a storage

[2]Technically, the COI approach has been implemented as a pre-assignment of the variables $x^i_{p,b}$ in the MIP according to the turnover rates of the items. Hence, T^{COI} and W^{COI} have been computed in a similar fashion as all other performance measures.

Scenario	Objective values		Weights	
			α_T	α_W
Minimize order cycle time	T^{min}		1	0
Minimize average picking effort		W^{min}	0	1
Tradeoff between the two criteria	T^*	W^*	1	1

Table 3: Optimization scenarios investigated.

strategy in reality, those values depend on various aspects neglected in this study, such as the physical layout of the aisles and cross-aisles and the picking strategy. Hence, the figures presented below are indicative values, and better estimates can be computed only by a more detailed simulation of the picking process.

Observe that the COI class-based strategy is also a feasible solution of the proposed MIP. This implies that our approach leads to at least as good performance as COI, which means $T^{min} \leq T^{COI}$, $W^{min} \leq W^{COI}$, and $T^* + W^* \leq T^{COI} + W^{COI}$ for the three optimization scenarios. The relevant question is the extent of cost saving for the different warehouse structures. Table 4 presents the results grouped by the number of pickers, P, number of zones per picker, B, and access schemes, δ, which were the parameters that affected the performance the most. Each row displays average results over 30 instances, for each of the three optimization scenarios. The results show that the improvement compared to COI is the largest for high values of P and B, and access schemes δ_1 and δ_2 (especially δ_1 in case of the order cycle time criterion). The latter observance is explained by the fact that when different pickers have very different access times (access schemes δ_3 and δ_4), then COI leads to close-to-optimal strategies.

For the order cycle time criterion, the largest improvement achieved was 36.7% for the $P = 6$, $B = 1$, δ_1 instances. The smallest difference was 1.8% for $P = 2$, $B = 1$, δ_3, while the average improvement was 21.3%. For the average picking effort criterion, our approach reached a 38.7% improvement for the $P = 6$, $B = 3$, δ_2 instances, and an average decrease of 19.4%. At the same time, this criterion could not be improved for the $P = 2$, $B = 1$ instances. This was due to the characteristic of those problem instances that one of the two zones could be filled with items whose joint request probability was lower than 1 according to the saturating sum approximation, which guarantees that the simple COI strategy is optimal.

Another important observation is that for most of the instances, the two conflicting criteria can be improved simultaneously compared to COI. This is illustrated by the results for the tradeoff optimization scenario. The choice of weights $\alpha_T = \alpha_W = 1$ favors rather the order cycle time criterion when there are few pickers ($P = 2$), whereas it leads to a more significant decrease of the average picking effort otherwise. Nevertheless, this behavior alters with different choices of weights.

4.2 Run times

For each instance and each strategy, we measured the run time necessary for the MIP solver to find a proven optimal solution. We experienced that the run times depend mostly on the number of binary variables (number of pickers, P, number of zones per picker, B, and number of milkrun cycles, K), and much less significantly on the number of continuous variables (including, notably, the number of items, N) or other parameters. Hence, Table 5 presents the results grouped by P and B in rows, and contains the different optimization scenarios in columns. Columns *Opt* display the number of instances out of 120 that could be solved to optimality, while columns *Time* show the average run times measured in seconds, or 1200 when the time limit was hit. The results illustrate that minimizing the

P	B	δ	Order cycle time T^{min}/T^{COI}	Avg. picking effort W^{min}/W^{COI}	Tradeoff T^*/T^{COI}	W^*/W^{COI}
2	1	δ_1	0.756	1.000	0.847	1.083
		δ_2	0.839	1.000	0.932	1.054
		δ_3	0.899	1.000	0.961	1.027
		δ_4	0.982	1.000	1.000	1.000
2	2	δ_1	0.770	0.821	0.813	0.919
		δ_2	0.843	0.793	0.892	0.886
		δ_3	0.800	0.967	0.847	1.040
		δ_4	0.847	0.961	0.909	0.987
2	3	δ_1	0.747	0.767	0.776	0.878
		δ_2	0.806	0.736	0.842	0.854
		δ_3	0.780	0.837	0.864	0.908
		δ_4	0.837	0.814	0.936	0.860
4	1	δ_1	0.653	0.906	0.802	0.930
		δ_2	0.773	0.892	0.892	0.895
		δ_3	0.899	0.879	0.937	0.914
		δ_4	0.944	0.865	1.040	0.901
4	2	δ_1	0.693	0.722	0.837	0.781
		δ_2	0.798	0.693	0.920	0.747
		δ_3	0.771	0.851	0.860	0.897
		δ_4	0.823	0.835	0.964	0.887
4	3	δ_1	0.696	0.670	0.833	0.759
		δ_2	0.780	0.641	0.947	0.708
		δ_3	0.723	0.774	0.878	0.826
		δ_4	0.778	0.754	0.963	0.804
6	1	δ_1	0.633	0.838	0.795	0.858
		δ_2	0.763	0.818	0.886	0.821
		δ_3	0.889	0.797	0.986	0.873
		δ_4	0.836	0.779	1.034	0.846
6	2	δ_1	0.687	0.683	0.854	0.741
		δ_2	0.799	0.657	0.957	0.702
		δ_3	0.773	0.781	0.955	0.849
		δ_4	0.774	0.767	1.064	0.824
6	3	δ_1	0.700	0.639	0.859	0.719
		δ_2	0.793	0.613	0.988	0.678
		δ_3	0.724	0.747	0.989	0.804
		δ_4	0.738	0.732	1.021	0.803

Table 4: Comparison to the COI approach, for three different optimization scenarios. The figures show the quotient of the objective values achieved by the proposed method and the COI approach. Values below 1 indicate improved performance.

order cycle time is in general less complicated than minimizing the average picking effort or finding a tradeoff. A commercial solver is able to solve instances to proven optimality within reasonable response time if $P \times B \times K \leq 40$.

P	B	T^{min}		W^{min}		T^* and W^*	
		Opt	Time (sec)	Opt	Time (sec)	Opt	Time (sec)
2	1	120	0.0	120	0.0	120	0.0
	2	120	3.0	120	6.7	120	9.2
	3	120	16.0	120	26.0	120	44.0
4	1	120	2.0	120	26.3	120	27.9
	2	120	26.3	100	358.9	106	353.7
	3	120	207.8	85	567.3	67	675.6
6	1	120	6.7	108	356.0	103	356.5
	2	120	98.5	61	765.6	60	759.0
	3	87	432.9	35	966.0	30	989.3

Table 5: Computation times for different problem sizes and performance measures. *Opt* shows the number of instances solved to optimality out of 120, *Time* displays the average solution times in seconds.

5 Conclusions and Discussion

In this paper we investigated the problem of storage assignment optimization in warehouses served by milkrun logistics. We showed that the milkrun system leads to a correlation among the request probabilities of items that can be exploited in a mathematical programming model. The proposed approach offers a number of advantages over previous clustering-based solution methods for the correlated storage assignment problem. It allows the explicit modeling of picking performance, and finding the optimal tradeoff between conflicting criteria such as the minimal order cycle time and the average picking effort. It can handle richer warehouse models, such as zones with different capacities, or it can be extended easily to various slot types. A MIP model of the problem has been proposed, and it has been shown in computational experiments that the proposed approach results in an up to 36-38% decrease of order cycle time or average picking effort compared to the classical COI approach. We see as a promising direction for future research the extension of this approach to dynamic warehousing environments, i.e., the adjustment of an existing storage assignment to changes in item properties, with the consideration of re-warehousing costs.

Acknowledgements

The author thanks Balázs Csanád Csáji, Botond Kádár, András Pfeiffer, and Gergely Popovics for the helpful discussions. The support of the János Bolyai scholarship No. BO/00138/07 is gratefully acknowledged.

References

M. Baudin. *Lean Logistics: The Nuts And Bolts Of Delivering Materials And Goods*. Productivity Press, 2005.

H. Brynzér and M. I. Johansson. Storage location assignment: Using the product structure to reduce order picking times. *International Journal of Production Economics*, 46–47:595–603, 1996.

R. de Koster, T. Le-Duc, and K. J. Roodbergen. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182:481–501, 2007.

E. H. Frazelle. *Stock location assignment and order picking productivity*. PhD thesis, Georgia Institute of Technology, 1990.

E. H. Frazelle and G. P. Sharp. Correlated assignment strategy can improve order-picking operation. *Industrial Engineering*, 4:33–37, 1989.

M. Garfinkel. *Minimizing Multi-zone Orders in the Correlated Storage Assignment Problem*. PhD thesis, Georgia Institute of Technology, 2005.

M. Goetschalckx and H. D. Ratliff. Shared storage policies based on the duration of stay of unit loads. *Management Science*, 36:1120–1132, 1990.

W. Hausman, L. Schwarz, and S. Graves. Optimal storage assignment in automatic warehousing systems. *Management Science*, 22(6):629–638, 1976.

J. L. Heskett. Cube-per-order index: a key to warehouse stock location. *Transportation and Distribution Management*, 3:27–31, 1963.

C.-C. Jane and Y.-W. Laih. A clustering algorithm for item assignment in a synchronized zone order picking system. *European Journal of Operational Research*, 166:489–496, 2005.

S. Kulturel, N. E. Ozdemirel, C. Sepil, and Z. Bozkurt. Experimental investigation of shared storage assignment policies in automated storage/retrieval systems. *IIE Transactions*, 31:739–749, 1999.

T. Le-Duc and R. De Koster. Travel distance estimation and storage zone optimization in a 2-block class-based storage strategy warehouse. *International Journal of Production Research*, 43(17):3561–3581, 2005.

R. J. Mantel, P. C. Schuur, and S. S. Heragu. Order oriented slotting: a new assignment strategy for warehouses. *European Journal of Industrial Engineering*, 1(3):301–316, 2007.

C. Petersen and G. Aase. A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics*, 92: 11–19, 2004.

M. Sadiq, T. L. Landers, and G. D. Taylor. An assignment algorithm for dynamic picking systems. *IIE Transactions*, 28(8):607–616, 1996.