2018

Optimization of a Fast-Pick Area in a Cosmetics Distribution Center

Mario C. Velez-Gallego
Universidad EAFIT

Alice E. Smith
Auburn University

Optimization of a fast-pick area in a cosmetics distribution center

Mario C. Vélez-Gallego
Departamento de Ingeniería de Producción
Universidad EAFIT

Alice E. Smith
Department of Industrial and Systems Engineering
Auburn University

***Abstract*—Fast-pick areas are commonly used in order picking warehouses to improve labor efficiency by concentrating picking activities within a compact area, thus minimizing the distance traveled by the pickers. One problem that must be solved when a fast-pick area is to be implemented is the so-called assignment-allocation problem. This deals with deciding which products should be assigned to the fast-pick area, and how much space should be allocated to these products. This research was motivated by the picking operation of a cosmetics distribution center where several fast-pick areas are in place. A mixed integer linear programming formulation is proposed for solving the variant of the assignment-allocation problem found in this company. Our computational experiments show that the proposed model is efficient for solving small yet realistic instances of the problem.**

I. Introduction

As the picking operation is probably the most labor intensive activity in many order-picking warehouses, the idea of concentrating the picking activities in a relatively compact area in order to reduce the distance traveled by the pickers has drawn the attention of researchers and practitioners in recent years. Simply put, the approach consists of dividing the warehouse into two areas: the fast-pick or forward area, where the picking activities take place, and the reserve or storage area, from which the fast-pick area is replenished. The research problem that arises when designing such system is twofold. Firstly, the size of the fast-pick area needs to be defined; and secondly, the allotted space needs to be distributed among the SKUs in the warehouse. Solving these two problems is challenging as two conflicting objectives arise: the smaller the fast-pick area, the less distance traveled by the pickers, but more replenishment trips are needed between the reserve and the fast-pick area.

This research was motivated by the picking operation that takes place at the warehouse of a company that produces cosmetics and personal care products. The warehouse is divided into a reserve area and several fast-pick areas, each one dedicated to one product family. Figure 1 presents one of the fast-pick areas in the warehouse. The picking process is performed manually by a group of operators, each of whom processes one customer order at a time as depicted in Figure 2. To complete one customer order, the picker starts from point I located at one side of the aisle, travels down the aisle picking the items that are stored on that side until reaching the SKU that is stored furthest. At this point the picker crosses the aisle and returns to point O, picking the SKUs stored on the other side of the aisle on the way back. As a consequence of the configuration of the fast-pick area, the picker always performs a U-shaped trip, with the depth of the trip being determined by the SKU stored furthest in the fast-pick area (see Figure 2). As these fast-pick areas are already designed and operational, the problem is not to choose on the size or the layout of the fast-pick area, but how many and which storage positions (i.e., bins) should be assigned to each SKU. Because the demand of these products changes over time depending primarily on fashion trends (makeup, for instance), the company is interested in a formal approach that allows decision-makers to solve both the assignment and space allocation problems several times during the year. If this is not addressed properly, the efficiency of the picking operation decreases, adversely affecting both the cost and the service level.

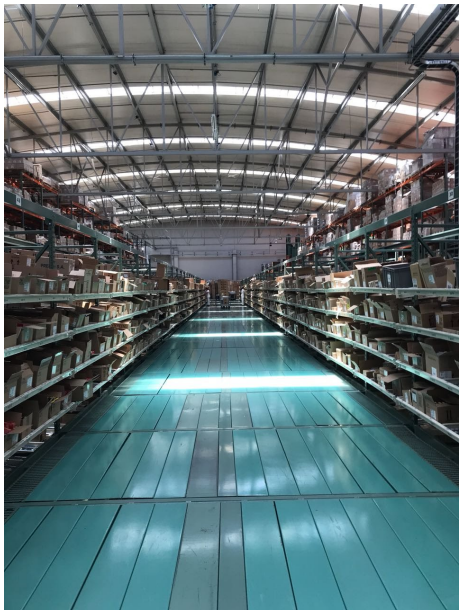

Fig. 1. A fast-pick area in a cosmetics warehouse

The aim of this work is to develop a mixed integer linear formulation for solving the product assignment and space allocation problem within the fast-pick area currently in operation. The remainder of this paper is organized as follows. A review of the literature is presented in section II. A formal description of the problem addressed in this work is presented in section III, along with a mixed integer linear formulation for solving

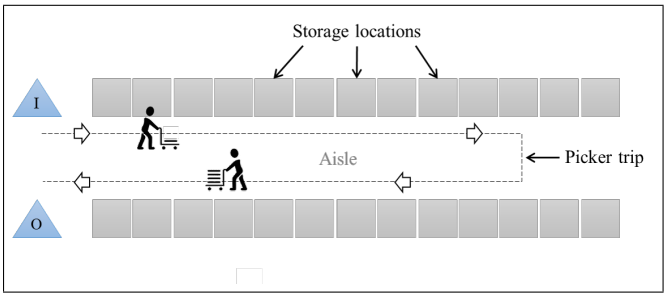

Fig. 2. The picking process

it. The results of a computational experiment carried out on a set of randomly generated instances is summarized in section IV, whereas some conclusions and future work opportunities are presented in section V.

II. Literature review

The problem of designing a fast-pick area has been addressed by several authors. The fluid models of Hackman, et al. [1] and Bartholdi and Hackman [2] were the first works that addressed the problem of deciding the amount of space that should be allocated to each SKU with the objective of minimizing the cost of replenishment. Recently, Subramanian [3] built upon these works and introduced a powers-of-two heuristic scheme for a single fast-pick area, and a near-optimal ranking heuristic for the case of multiple fast-pick areas. As all of these approaches assume that the fast-pick area can be continuously partitioned among the SKUs, some drawbacks appear when implemented in a realistic setting where the SKUs need to be assigned to discrete bins or storage positions. To address this issue Walter, et al. [4] considered and solved the so-called discrete forward–reserve problem. The works of Walter, et al. [4], Frazelle, et. al. [5], and Gu [6] extended the problem to include determining the optimal size of the fast-pick area. Comprehensive reviews of the literature can be found in [3] and [4].

The problem in the cosmetics company is similar to one of the problems addressed in [4], except that the authors of [4] assume that the fast-pick area is very small and, thus, that the travel time spent by the picker within the fast-pick area is negligible. In particular, the authors state that *"due to the compact size of the forward area the locations of SKUs are assumed to not affect picking efficiency"*. Following this common assumption, the aim of this and similar solution approaches found in the literature is to minimize the number of replenishments needed to maintain the fast-pick area; this is the number of times that product needs to be transported from the reserve to the fast-pick area in a given time window. However, in the cosmetics company that motivated this work, the fast-pick area is approximately 60 meters deep (i.e., 196 feet), and therefore the travel time the pickers spend preparing customer orders is definitely not negligible and depends on the actual bin positions assigned to the SKUs (see Figure 1).

III. Problem description

Formally, we are given the set O of the customer orders that need to be prepared (i.e., picked) manually by a group of pickers assigned to the fast-pick area, and the set S of the SKUs (i.e., products) that are held by the warehouse. We let $Q_k \subset S$ be the subset of SKUs in customer order $k \in O$. In order to complete customer order $k \in O$, the picker must visit all storage locations where the SKUs in Q_k are stored. Finally, we define the fast-pick area as a set of storage locations L. Each storage location $i \in L$ is in turn comprised of n_i storage bins, all identical in size, each of which can hold one SKU. As an example, a single storage location comprised of 20 storage bins is depicted in Figure 3. The objective is to assign the products in S to the storage locations in L to minimize the labor cost represented by the total distance traveled by the workers while performing both the picking and replenishment activities. The following is a list of the most important assumptions used to model this problem:

- A single SKU is restocked during a replenishment trip.
- At each replenishment trip all of the bins assigned to the SKU being restocked are filled to their maximum capacity.
- The picker processes only one customer order at a time.
- The actual picking time is not considered as it is assumed to be independent of the slotting decisions.
- All SKUs must be assigned to at least one storage bin in the fast-pick area as it is not practical to pick a product directly from the reserve.
- In the current formulation of the problem, the objective is to minimize the sum of the distance traveled by the pickers, and the distance traveled to replenish the fast-pick area. The underlying assumption behind this objective is that the cost on a per-meter basis is the same for both activities.
- The distance between the reserve and the fast-pick areas is fixed (i.e., r) and independent of the slotting decisions in both areas.

Fig. 3. A single storage location

To solve the above described problem a mixed integer linear programming formulation follows.

Sets

O	Customer orders
S	SKUs
Q_k	SKUs in customer order $k \in O$ (i.e., $Q_k \subset S$)
L	Storage locations

Parameters

h_i	Distance to storage location $i \in L$
r	Distance between the fast-pick and reserve areas
q_s	Units of SKU $s \in S$ that can be stored in one bin
d_s	Demand of SKU $s \in S$ in customer orders in O
n_i	Number of bins at location $i \in L$
f_{ist}	Distance traveled to replenish SKU $s \in S$ if assigned to t bins at location $i \in L$

Decision variables

y_{ist}	Binary variable. $y_{ist} = 1$ if SKU $s \in S$ is assigned to t bins at location $i \in L$, and $y_{ist} = 0$ otherwise
z_k	Distance traveled to prepare customer order $k \in O$

Objective function

$$\textit{Minimize} \qquad \sum_{k \in O} z_k + \sum_{i \in L} \sum_{s \in S} \sum_{t=1}^{n_i} f_{ist} \cdot y_{ist} \tag{1}$$

Constraints

$$\sum_{i \in L} \sum_{t=1}^{n_i} y_{ist} = 1 \qquad \forall s \in S \tag{2}$$

$$\sum_{s \in S} \sum_{t=1}^{n_i} t \cdot y_{ist} \leq n_i \qquad \forall i \in L \tag{3}$$

$$z_k \geq \sum_{t=1}^{n_i} 2 \cdot h_i \cdot y_{ist} \qquad \forall i \in L, k \in O, s \in Q_k \tag{4}$$

$$y_{ist} \in \{0, 1\} \qquad \forall s \in S, i \in L, 1 \leq t \leq n_i \tag{5}$$

In the formulation above, expression 1 represents the objective of minimizing the total distance traveled by the workers due to picking and replenishment activities. The first term in the expression is the total distance traveled by the pickers, whereas the second term accounts for the total distance traveled to replenish the fast-pick area. Note that the latter distance is a non-linear function of the number of bins assigned to a given SKU: the more bins are assigned to it, the fewer times the SKU needs to be replenished. In order to be able to model the objective as a linear function on the decision variables, we firstly enumerated all possible values of the distance traveled to replenish each SKU as a function of the number of bins assigned to it, from 1 to the maximum number of bins per location (i.e., see the definition of parameter f_{ist}), as in expression 6. By doing so, and by defining the binary variable y_{ist} in a similar fashion, we were able to model the problem as a mixed integer linear program.

$$f_{ist} = 2 \cdot r \cdot \left\lceil \frac{d_s}{t \cdot q_s} \right\rceil \quad \forall i \in L,\ s \in S,\ 1 \leq t \leq n_i \tag{6}$$

Each SKU can only occupy part or all of a single location. An SKU cannot be spread among more than one location. Constraints in expression 2 assure this. Expression 3 ensures that the number of bins assigned to an SKU do not exceed the number of bins available at that storage location. Expression 4 defines the distance traveled by the picker to prepare customer order k, defined as twice the distance to the location of the SKU in customer order k that is stored furthest. The aisle width was not included in expression 5 as it would just add a constant to the objective value. Finally, expression 5 defines the domain of the binary decision variables.

IV. Computational experience

To evaluate the performance of the formulation proposed when implemented on a commercial solver, a set of test instances were generated randomly, resembling the real setting of the fast-pick areas in the warehouse that motivated this research. To mimic the real setting of a given fast-pick area, three parameters were considered to generate the test instances: (1) the number of SKUs, (2) the size of the fast-pick area, and (3) the number of customer orders to include in the instance. As the number of SKUs that the real fast-pick areas hold vary from 100 in the smallest to 1500 in the largest, in this experiment we considered three values for this parameter, namely 50, 100 and 200. As the problem that motivated this work requires that all the SKUs are assigned to at least one bin in the fast-pick area, we defined the parameter λ as the ratio between the total number of available bins and the number of SKUs in the instance. We used three values of λ, namely 2.0, 3.2, and 4.0. The number of customer orders was fixed to 3600 to reflect the operation of an average month. The composition of each customer order was generated as follows. Each SKU appears in a customer order with probability $p = (s + 1)^{-1}$, where s is the index of the SKU, to reflect a realistic setting where a few SKUs account for a high percentage of the total demand. For those SKUs in a given customer order, the amount requested (i.e., d_s) was sampled from discrete uniform distribution between 1 and 10. Finally, the number of units that can be stored in a single bin (i.e., q_s) was sampled from a discrete uniform distribution between 10 and 100, the distance between consecutive storage locations was set to 2 meters, and the distance between the fast-pick and reserve areas to 100 meters for all cases. Five independent instances were generated for each combination of the number of SKUs and the value of λ, for a total of 45 instances. Table I summarizes the values of the parameters used to generate the set of test instances.

TABLE I
TEST INSTANCES

SKUs	λ	Storage locations	Bins per storage location
50	2.0	10	10
	3.2	16	10
	4.0	20	10
100	2.0	10	20
	3.2	16	20
	4.0	20	20
200	2.0	20	20
	3.2	32	20
	4.0	40	20

The formulation was coded in Xpress Mosel Version 4.8.2, and each instance was solved using Gurobi 8.0 as the commercial solver. The solver was allowed to run for a maximum of one hour on a machine with 64 GB of memory and 12 Intel Xeon processors running at 3.5 GHz under Windows 7 Enterprise at 64 bits. If the solver could not find the optimal solution whithin this time, the best integer solution and the best lower bound were kept for further analysis. For each instance e, the optimality gap (i.e., G_e) was computed as a percentage, as in equation 7, where BIS_e and LB_e are respectively, the best integer solution and the best lower bound found by the solver for problem instance e after one hour of computational time.

$$G_e = \frac{BIS_e - LB_i}{LB_e} \times 100\% \tag{7}$$

After completing the computational experiments the solver was able to find at least one feasible integer solution in all cases, but none of the instances was solved to optimality within the allocated time. The optimality gap for all the test instances is presented in Table II.

TABLE II
OPTIMALITY GAP AFTER ONE HOUR

SKUs	λ	Instance					Average
		1	2	3	4	5	
50	2.0	22.6	22.1	21.0	21.7	20.0	21.48
	3.2	39.6	41.0	40.4	41.8	35.6	39.68
	4.0	48.3	52.8	41.3	49.6	56.9	49.90
100	2.0	21.6	23.9	22.3	22.8	22.5	22.62
	3.2	39.1	39.0	40.3	40.1	39.6	39.62
	4.0	50.9	48.8	48.9	100	100	69.72
200	2.0	69.2	100.0	61.6	62.3	100	78.62
	3.2	100	100	100	100	100	100.00
	4.0	100	100	100	100	100	100.00

As seen in Table II, the optimality gap ranged from 20% to 24% for the instances with 50 and 100 SKUs, and $\lambda = 2.0$. In the case of the instances with 200 SKUs, the average optimality gap was roughly between 60% and 100%, suggesting that only small-sized instances can be solved using the formulation proposed above. A chart with the average optimality gap by the number of SKUs and the λ ratio is presented in Figure 4.

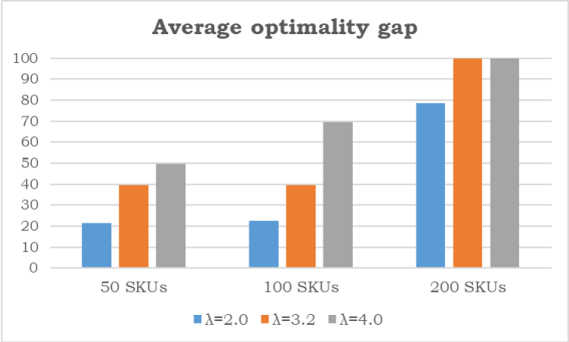

Fig. 4. Average optimality gap

In a second set of experiments we allowed the solver to run for a maximum of three hours on the set of instances with 50 SKUs and $\lambda = 2.0$. In this case, even though none of the instances was solved to optimality, the gap was below 5% in all cases. A summary of the results for this set of experiments is presented in Table III. Finally, we allowed the solver to run for 12 hours on the first instance in the latter set. The solver did not converge to the optimal solution after that time either, showing a very slow convergence rate. The optimality gap at the end of this final experiment was 3.01%.

TABLE III
OPTIMALITY GAP AFTER THREE HOURS

Instance	1	2	3	4	5
Optimality gap	3.91	4.26	3.40	2.51	2.73

V. CONCLUSIONS

Fast-pick areas are commonly used in labor intensive order picking warehouses. In this work we considered a relatively large fast-pick area where the distance traveled by the pickers should not be neglected. We addressed the problem of deciding where in the fast-pick area the SKUs should be placed, and the number of storage bins that should be assigned to each SKU. A mixed integer linear programming formulation was proposed and implemented on a commercial solver with the objective of minimizing the summation of the pick costs (that is, time or distance) and the cost (that is, labor cost) of replenishment. As such, this is a substantially different version of previous approaches for designing a fast-pick area. It is also a practical variant of the forward pick area design problem that is applicable to distribution centers with relatively large quick pick areas. Our computational experience showed that although we were able to find several feasible solutions in all the cases, to find the optimal solution for realistic sized instances is challenging. As per future research opportunities, the following are some natural extensions to the problem:

– Develop an approximation solution approach (i.e., a heuristic) to solve larger instances of the problem, as

the computational cost of solving these instances using an exact approach like the one proposed in this work is apparently too high.

- Relax the constraint in the current model that requires that all SKUs need to be assigned to the fast-pick area. In a more general setting, it would be possible to pick an SKU directly from the reserve area, if such SKU is not assigned to the fast-pick area.
- The current model restricts the number of storage bins assigned to a given SKU to include only those in a single storage location. This assumption makes it simpler to calculate the distance travelled by the picker to prepare a single customer order, but it may also lead to a sub-optimal solution.
- The model addresses the problem that arises in the fast-pick area of the cosmetics company described before; that is, one with a single aisle, and where all the trips performed by the picker have the same U-shaped pattern. The model could be extended so that a more general layout can be investigated.
- The set O of customer orders used to develop the proposed solution approach provides a useful way to model the workload of the picking process, but it also conveys an important drawback. In a realistic setting the set O is not known in advance, but the assignment–allocation problem needs to be solved prior to the arrival of the orders in a given period of time. Thus, the assumption behind this modeling framework is that the set of future orders will present identical behavior to those in the past. This is a difficult issue that could be addressed in the future.
- Another assumption that could be relaxed to make the problem more general is that of the slotting of the reserve area being already defined and fixed.
- To consider order batching so that several customer orders are picked within a single trip.

REFERENCES

[1] Steven T. Hackman, Meir J. Rosenblatt, and John M.Olin. Allocating items to an automated storage and retrieval system. *IIE Transactions*, 22(1):7–14, 1990.

[2] John J. Bartholdi III and Steven T. Hackman. Allocating space in a forward pick area of a distribution center for small parts. *IIE Transactions*, 40(11):1046–1053, 2008.

[3] Sriram Subramanian. *Managing space in forward pick areas of warehouses for small parts*. PhD thesis, Georgia Institute of Technology, 2013.

[4] Rico Walter, Nils Boysen, and Armin Scholl. The discrete forward–reserve problem – allocating space, selecting products, and area sizing in forward order picking. *European Journal of Operational Research*, 229(3):585 – 594, 2013.

[5] E.H. Frazelle, S.T. Hackman, U. Passy, and L.K. Platzman. The forward-reserve problem. In T. Ciriani and R.C. Leachman, editors, *Optimization in Industry, Vol II*, pages 43–61. John Wiley & Sons, 1994.

[6] Jinxiang Gu. *The forward reserve warehouse sizing and dimensioning problem*. PhD thesis, Georgia Institute of Technology, 2005.