Tactical and Operational Models for the Management of a Warehouse

Neil Jami and Michael Schröder

Abstract This paper deals with the modeling, routing, and managing aspects of a warehouse with parallel aisles and cross-aisles, in which we assume a picker-to-part process. Pickers either retrieve in the aisles-stored products to fulfill a customer order, or do some nonurgent activity. The main contribution of this paper is the consideration of constraints, which are often disregarded by other papers. We study in particular the consequences of some 'working conditions' for the pickers on the overall solution quality. We analyze a warehouse layout designated to vehicle routing. We study the organization of products into locations, given some statistical forecasts on the future orders. Then we describe a management strategy to regulate the number of pickers doing the picking activity. Finally, an algorithm is proposed as a solution and tested by simulation experiments.

Keywords Warehousing · Simulation · Optimization · Dynamic process

Introduction

Warehousing is an important part in a logistic process (Roodbergen 2001). A careful designing and planning of a warehouse provides a better service for lower cost. We study here the order picking process, as it is known as the most time-consuming task in a warehouse (Tompkins et al. 1996). Order picking has been widely studied for decades on the designing of the warehouse, the storing strategies, and the control of the pickers (De Koster et al. 2007).

N. Jami (✉) · M. Schröder
Fraunhofer Institute for Industrial Mathematics (ITWM), Fraunhofer Platz 1, 67663 Kaiserslautern, Germany
e-mail: neil.jami@itwm.fraunhofer.de

M. Schröder
e-mail: michael.schroeder@itwm.fraunhofer.de

H. Kotzab et al. (eds.), *Dynamics in Logistics*,
Lecture Notes in Logistics, DOI 10.1007/978-3-319-23512-7_65

The designing of the warehouse is a complex task with few structured approaches (Backer and Canessa 2009). We consider here the most common design of the warehouse composed of parallel aisles and cross-aisles. Concerning the organization of the storage area, we refer to Roodbergen and De Koster (2001), De Koster et al. (2007). The control of the pickers raises two questions, namely the routing of the pickers and the management of the number of needed pickers. Several papers propose efficient heuristics to solve the routing problem (Roodbergen and De Koster 2001; Hall 1993; Ratliff and Rosenhal 1983; Theys et al. 2010). However, there have only been a few papers about a dynamic management of the pickers. Our work is close to the one of Mazalov and Gurtov (2012), who considered a queuing model with a dynamic number of servers depending on the queue length. Both works assume that we can call additional pickers or send them back to do some nonurgent activities in the warehouse, in order to minimize the number of pickers while optimizing the service quality. However, instead of running simulations to find out the maximal number of pickers we need, we fix the number of pickers present in the warehouse, and evaluate the service quality under different working constraints.

The contributions of this paper are the following. First, current control strategies for the pickers disregard their situation. Pickers are asked for an intensive work and a high flexibility. Our main contribution is to provide a compromise between the service quality, the number of pickers at disposal, and the picker's flexibility. The second contribution is a storage strategy for a given order forecast and warehouse design. The third contribution is the modeling of car-traffic like routing regulation in the warehouse in order to facilitate dense picker movements.

In this paper, we first complete the warehouse model to define exactly how the pickers move. Second, we study a strategy to store products efficiently in the warehouse. Then, we present a multicriteria problem, which is to determine how many pickers are needed to fulfill efficiently the orders. The number of pickers is then dynamically adapted. Finally, we describe an algorithm for the management of pickers and simulate it under different conditions.

Description of the Warehouse

We consider a rectangular layout composed of several *parallel pick aisles* (Roodbergen and De Koster 2001), which is one of the most common structures for a warehouse. The warehouse is subdivided into several blocks separated by *cross-aisles*. A cross-aisle does not contain any product location, but can be used to travel from a pick aisle to another. Two other cross-aisles are also present at the front and at the back of the warehouse. The products are placed into locations on both sides of each pick aisle. These locations can be, for example, pallet racks or stacking blocks (Roodbergen 2001).

The proposed warehouse layout is intended for a *picker-to-part* system with carts or other vehicles, i.e., the pickers are supposed to move from location to location to retrieve the products corresponding to a pick list, and finally bring the products to a packing station called *depot*. We can argue that placing the depot in the middle of the frontal cross-aisle provides better travel times (Merkuryev et al. 2009). To facilitate the traffic in the warehouse and speed up the retrieval of products, the aisles will be wide enough to let carts cross to each other. Several pickers can then pick up products in the same picking area.

The aisles and cross-aisles are divided into two unidirectional corridors. For example, we only let the pickers move in the right corridor of the aisle in traffic direction, as shown in Fig. 1. A picker can freely change corridors to move in the opposite direction, or move out of the corridor to the side of the aisle to pick up a product without disturbing the other pickers movements.

An interesting point of this layout is that there always exists a shortest path from a location L_1 to a location L_2 using at most one cross-aisle. Therefore, the shortest path from a location to another can be computed in constant time $O(1)$.

A last problem to deal with is to determine in which sequence the products of an order should be retrieved. This task is an instance of the Travelling Salesman Problem (TSP) (Lawler et al. 1985), which is NP-hard. The shortest path to take begins and ends at the depot and must go through every location corresponding to an ordered product. A lot of work has been done in this domain, especially when

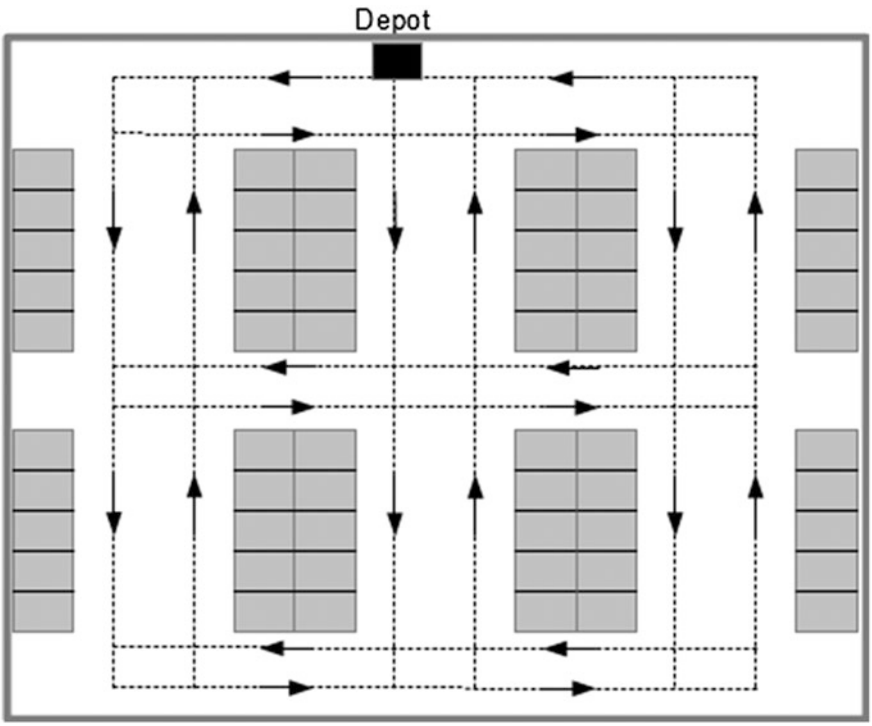

Fig. 1 Layout of the warehouse. The *dashed lines* represent the corridors, and the *arrows* give the movement directions

there are only two or three cross-aisles (Roodbergen and De Koster 2001). In our simulation, the TSP has been solved by enumeration, which is fast enough to compute the sequence of less than eight products. For higher numbers of products, we refer to more efficient algorithms (Ratliff and Rosenhal 1983; Theys et al. 2010).

Positioning the Products

In this section, we present a strategy to position the products in the warehouse locations. For this purpose, we must predict which products will be the most frequently ordered and which products will be often ordered together. Indeed, frequently ordered products should be placed as close to the depot as possible, while products which are often ordered together should be placed close to each other. This step is independent of the design of the layout, as we just have to know the shortest path between each pair of locations.

We assume here that some forecasting method could generate a large set of orders representing the future orders. Then we optimize the positioning of products according to this sample. From a *sequence function* S giving the retrieving sequence of the products for each order, we can compute a *transition probability matrix* $P(S)$ so that $P_{i,j}$ is the probability that the next product to pick up after product P_i is P_j.[1]

Our objective is to determine the position of each product. This product organization is characterized by the *position matrix* X, where $X_{i,j} = 1$ if the product P_i should be located in the location L_j, $X_{i,j} = 0$ otherwise. We denote by D the (already known) *distance matrix* providing the distance $D_{i,j}$ between each pair of locations (L_i, L_j). The *average path length* $Z(P, X)$ to fulfill an order is

$$Z(S,X) = \sum_{i,j} (X^T \cdot P(S) \cdot X)_{i,j} \cdot D_{i,j} \tag{1}$$

The objective is to find an optimal value of X, i.e., a value minimizing the average path length. We have to solve the following optimization problem:

$$\min_{S,X} Z(S,X) \tag{2}$$

under;
S gives the products retrieving sequence for each order
$\forall (i,j)\colon X_{i,j} \in \{0,1\}$
$\forall i\colon \sum_j X_{i,j} = 1, \quad \forall j\colon \sum_i X_{i,j} \leq 1,$

[1] We set for example that the product noted P_0 represents the depot, whose location is fixed.

We already saw that for fixed product position X, the optimal path for each order is the solution of a TSP. For given X, computing the solution S' of a TSP for each order of the set provides a new transition probability matrix $P(S')$. Since the new product sequence for each order is optimal, we have

$$\forall S, Z(S', X) \leq Z(S, X) \tag{3}$$

On the other side, when the sequence function S is known, (2) becomes a quadratic assignment problem (QAP) (Finke et al. 1987). Given S, solving QAP provides a new positioning of products X' so that

$$\forall X, Z(S, X') \leq Z(S, X) \tag{4}$$

Given a sample of orders to fulfill, the following heuristic organizes efficiently the products in the warehouse:

1. Initialization:
   (a) Set an initial sequence function S.
   (b) Compute X solution of the QAP minimizing $Z(S, Y)$.
2. Optimization loop:
   (a) Compute S' solution of the TSP minimizing $Z(Y, X)$.
   (b) **If** $Z(S, X) = Z(S', X)$: **stop** the loop. **Else** $S := S'$
   (c) Compute X' solution of the QAP minimizing $Z(S, Y)$.
   (d) **If** $Z(S, X) = Z(S, X')$: **stop** the loop. **Else** $X := X'$.

The computed solution is locally optimal. The algorithm terminates because of the following observation: at each step, we compute the best sequence function S for the given positioning matrix X, and then update the value of X. Since the value $Z(S, X)$ of the solution improves at each step, we cannot compute twice the same matrix X until the last step. Therefore, the number of steps is limited to $|X| + 1$, where $|X|$ is the number of possible product organizations.

This heuristic can be accelerated by fixing a maximal number of iterations and a precision error ϵ so that we stop the algorithm when $Z(P, X) \leq Z(P, X') + \epsilon$. Finally, the initialization step can be replaced by more meaningful values for S and X.

Dynamic Picker Management

In the two previous sections, we studied how to minimize the service time, i.e., the time needed to fulfill an order, assuming that this time is proportional to the length of the travelled path. In this section, we consider a different problem. We suppose that the products are already well positioned in the warehouse, and that we can quickly compute the path to travel to pick up products and fulfill the corresponding order. We denote by $\mu = 1/Z(P, X)$ *the service rate* resulting from the position of the products and the computation of the shortest path to fulfill an order.

We are interested in the number of pickers that are required to carry out this picking activity. By *manager,* we refer to a decision maker who decides when to call pickers to the depot to do the picking activity, and when to make a picker leave the picking activity to pursue other tasks. The manager can either a human being or computer algorithm.

Among the different activities assigned to the pickers, the picking activity is one of the most critical and requires a lot of efforts (Roodbergen and De Koster 2001). Therefore, it is usually the most costly one. For this reason, it is important to use as few pickers as possible while providing a good service. We define here the objectives of three different stakeholders: the clients want to get the best service possible, the pickers in the warehouse ask for good working conditions, while the manager wants to minimize the costs of the picking activity while obeying the objective of the other groups.

Having the best service usually means minimizing not only the average service time, but also the worst case waiting time, as large service times may create strong dissatisfaction of clients.

The pickers' working conditions are considered as constraints, because they are supposed to be guaranteed to them, and do not need to be further optimized. The number of pickers is limited, and the pickers are guaranteed minimal activity durations as well as an arrival delay. So, a picker must not be affected on the picking activity for duration below γ, and a picker leaving the picking activity should not be called back for duration β. Furthermore, called pickers dispose of a delay δ to finish their current activity and go to the depot.

Before introducing the key performance indicators (KPI) to measure the efficiency of picker management, we present some theoretical results.

We consider a queuing model for the waiting orders. The waiting orders are then recorded in a queue and treated in their arrival sequence. We denote by λ the average arrival rate of the orders. The optimal number of pickers that should be on the picking activity is $\alpha = \lambda/\mu$. If we always use less than α pickers, then the orders will arrive faster than they are fulfilled. The queue will thus grow infinitely, so as the waiting time of the orders, following the theorem of Little (Little 1961):

$$\mathrm{E[Lq]} = \lambda \cdot E[\mathrm{Wt}], \tag{5}$$

where E[Lq] denotes the average queue length and E[Wt] denotes the average waiting time of an order.

Thus, the manager must minimize the number of active pickers while keeping it on average above α. An active picker is either *busy*, i.e., fulfilling an order, or *idle* if he waits for an order to arrive, which happens when the queue is empty. We say here that a strategy to manage pickers has an *optimal cost* when the queue length does not grow indefinitely and the picker costs are minimal. The following theorem provides sufficient conditions for a strategy to have optimal cost:

Theorem *Consider a strategy providing in average n_a active pickers and n_i idle pickers:*

- $n_a \leq \alpha$ *if and only if* $n_i = 0$.
- $n_a \geq \alpha$ *if and only if the queue does not grow up indefinitely.*

Proof We consider a probabilistic model with the following notations:

- Δ: considered time period where the pickers work while orders arrive.
- $\tau = n_a \Delta$: sum of the picking activity durations of all pickers.
- $\Omega_\Delta = \lambda\Delta$: number of orders arrived during period Δ.
- $R = \mu(n_a - n_i) \leq \lambda$: the order fulfillment rate.
- $\Omega_\tau = \Delta R \leq \Omega_\Delta$: number of orders fulfilled during period Δ.

Suppose first that $n_i = 0$:

$$\alpha = \lambda \cdot \mu^{-1} = \left(\Omega_\Delta \cdot \Delta^{-1}\right) \cdot \left(n_a \cdot \Delta \cdot \Omega_\tau^{-1}\right) = \Omega_\Delta \cdot \Omega_\tau^{-1} \cdot n_a \geq n_a$$

Suppose instead that $n_a < \alpha$. The order fulfillment rate $n_a\mu$ is smaller than the order arrival rate $\lambda = c\mu$, and thus the queue length will in average keep growing up. There will always be waiting orders and thus no idle picker.

Suppose now that α is an integer and $n_a = \alpha$. If we had $n_i \neq 0$, the order fulfillment rate $\mu\ (n_a - n_i)$ would be smaller than the order arrival rate $\alpha\mu$. There would be no idle picker, which is a contradiction. Therefore, $n_i = 0$ if and only if $n_a \leq \alpha$.

Consider the second rule of the theorem. The queue does not grow up indefinitely if and only if the order arrival rate $\lambda = \alpha\mu$ is not strictly greater than the order fulfillment rate $\mu\ (n_a - n_i)$. When the queue is growing up, there are no idle pickers. Therefore, the queue does not grow up indefinitely if and only if $\alpha \leq n_a$.

Minimizing the number of active pickers is actually not a good criterion, since it is not clear how to have in average α active pickers, while minimizing the queue length. This theorem points out that minimizing the costs actually means having in average zero idle pickers. Therefore, it makes more sense to minimize the *wasted manpower*, i.e., the number of active pickers that are idle. The criterion of wasted manpower is simple to minimize, as we just make sure that no active picker is idle. We can then freely optimize the other criteria.

In practice, the minimal duration γ of the picking activity may make it impossible to make a picker leave while the queue is empty; hence he will be idle for some time. Likewise, the minimal duration β before he can return to the picking activity can increase the maximal waiting time for orders.

The calling delay of the pickers increases also the difficulty, as a picker also loses some time every time he is called to come to the depot. Thus, we should also minimize the *call frequency*, i.e., the frequency of calling pickers to the depot to do the picking activity.

Another interesting objective is to maximize the average number of the pickers that have been inactive for duration greater than β, and therefore can be called to the depot. This improves neither the cost nor the service quality. Nonetheless, it indicates a certain comfort in the management, as it shows how many pickers are not necessary for the picking activity with the given strategy, and how well unexpected increase of the order arrival rate can be dealt with.

We define the five KPI of a strategy to manage pickers as the following:

- The wasted manpower (WMp): the average number of idle active pickers.
- The average waiting time of an order (AWt).
- The maximal waiting time of an order (MWt).
- The call frequency (CF): the rate of picker calls.
- The extra manpower (EMp): the average number of inactive pickers that can be called to the picking activity.

We finally present an algorithm to efficiently manage the number of pickers. This algorithm is nevertheless generic and must be calibrated to meet the real objectives of the manager. The algorithm obeys two rules of the theorem to provide an optimal cost. The algorithm decomposes the set of possible queue lengths into several intervals (I_n) so that

- The interval I_n is the set of queue lengths where n pickers can be active. The value of n only changes if the queue length gets a value outside I_n.
- There are no active pickers if and only if the queue is empty.
- Interval I_n must not be larger than interval I_{n+1}.
- The final decision to make a picker leave occurs when he finishes fulfilling his current order.

If the queue length exceeds the interval bound of the current number of active pickers, the algorithm calls an additional picker. If the queue length falls below the interval bound, the algorithm will make leave the next picker who finishes an order. If the intervals overlap, then the current number of active pickers also depends on its previous value and is chosen to minimize the call frequency.

Simulation

We implemented a simulation model to test our algorithm on randomly generated data. The order arrival is generated with a Poisson process, with in average between 75 and 90 incoming orders per hour. The order generation and the warehouse dimensions are set to obtain an average order fulfilling time of $\mu^{-1} = 4$ min. The optimal average number of active pickers is then [5, 6]. At the beginning of each test, the queue is supposed to be empty and none of the eight pickers at disposal is active. Each simulation lasts 5 hours, which is long enough so that the initial state of the process has a minimal importance in the results. The time scale between each decision is 15 s. We present here simulations for two scenarios providing different working quality, as presented in Table 1. A more extensive simulation of the algorithm will be presented in an upcoming paper.

Table 1 Simulation Scenarios

Scenario 1: good working conditions	γ = 10 min, β = 20 min, δ = 2 min
Scenario 2: bad working conditions	$\gamma = \beta = \delta = 0$ min

For the algorithm, we divide the intervals into two groups $(I_n)_{n<\alpha}$ and $(I_n)_{n>\alpha}$:

- $n < \alpha$: $I_0 = \{0\}$; $I_1 = \{1, 2\}$; $I_2 = \{1, 2, 3\}$; $I_3 = \{2, 3, 4\}$; $I_4 = \{3, 4, 5\}$; $I_5 = \{4, 5, 6\}$
- $n > \alpha$:
  - $I_6 = \{5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$
  - $\forall n$, $|I_n| = 11$, $|I_n \cap I_{n+1}| = 6$

The results of the simulations are presented in Table 2. We first note that the algorithm managed to avoid any waste of manpower. Thus, the manager can afford to ensure a minimal picking activity duration of γ = 20 min to the pickers.

The simulation with good working conditions provides a good compromise of the average waiting time, namely between 100 and 200 % of the average picking time, in order to deal with order arrival rate variations. Furthermore, the algorithm usually keeps one or two extra pickers. This is enough in this situation to face a sudden increase of the order arrival rate.

In comparison to the simulation with bad working conditions, the average waiting time is a little larger while the call frequency is slightly smaller. This means that sometimes, the algorithm would like to add a picker but cannot, resulting in a small increase of the waiting time. However, the maximal waiting time has a similar value in both scenarios. Moreover, giving better working conditions to pickers does not decrease too much the extra manpower to make it critical.

Further simulations that we do not present here also show that a high value of γ generates a significant waste of manpower, and a long duration δ emphasize this waste. A high value of β leads to higher waiting times, and this effect is increased by a low value of δ, that is for a higher call frequency.

We mention again that satisfying results require the calibration of the algorithm, which should be set depending on the requirements of the warehouse.

Finally, it is meaningful to take some picker constraints into account, namely by reducing the picker call frequency, in order to get a more accurate estimation of the service time and the picker costs.

Table 2 Simulation results

KPI	AWt (min)	MWt (min)	CF (h − 1)	WMp	EMp[a]
Scenario 1	5.25	10.0	4.9	0	1.6
Scenario 2	4.56	9.4	5.7	0	2.9

The duration of each simulation is 5 h

[a]The EMp is the only of the four KPI that we want to maximize, in order to be sure that we can always call pickers, and to see if we can reduce the number of pickers present in the warehouse

Conclusion

This paper dealt with the management of a warehouse with parallel wide aisles. We studied the three main problems of an order picking process, which are the layout of the warehouse, the storing strategies for the products and the control of the number of pickers.

For the layout, we used a car-traffic like circulation to facilitate the movement of vehicles that the pickers would use. We then presented an algorithm organizing the products into locations according to a sample of predicted orders. Finally, we considered the problem of dynamic management of picker activities. The goal was to minimize the number of pickers on the order picking activity while ensuring low picker flexibility and a good service quality.

KPI have been presented to model the objective functions of this multicriteria problem. We described a generic algorithm deciding on the number of pickers, and used it in a simulation experiment. With a good calibration of the algorithm, we can minimize the waste of manpower, while providing efficient service times and keeping available pickers for sudden increases of the order arrival rate.

An interesting topic for future research would be to study the restock of the products at the same time as their picking, in order to create a strategy allocating the pickers between these two activities. Further research can also be done on the algorithm positioning the products, in order to determine how many iterations of the loop are needed to place the products, and how much computation time it takes in different situations. Finally, it would be interesting to develop the picker's constraints to provide a better dispatching or to send tome pickers home when the order arrival rare is too low, hence reducing the picker's costs.

References

Backer P, Canessa M (2009) Warehouse design: a structured approach. EJOR 193:425–436

De Koster R, Le-Duc T, Roodbergen KJ (2007) Design and control of warehouse order picking: a literature review. EJOR 182:481–501

Finke G, Burkard RE, Rendl F (1987) Quadratic assignment problems. Ann of Discr Math 31: 61–82

Hall RWH (1993) Distance approximation for routing manual pickers in a warehouse. IIE Trans 25 (4):76–87

Lawler EL, Lenstra JK, Rinnooy Kan AHG, Shmoys DB (1985) The traveling salesman problem. A guided tour of combinatorial optimization. Wiley, Chichester

Little JDC (1961) A proof for the queuing formula: $L = \lambda W$. Oper Res 9:383–387

Mazalov VV, Gurtov A (2012) Queuing system with on-demand number of servers. Mathematica Applicanda 40(2):1–12

Merkuryev Y, Merkuryeva G, Piera MA, Guasch A (2009) Simulation-based case studies in logistics. Springer, Berlin

Ratliff HD, Rosenhal AS (1983) Order picking in a rectangular warehouse. A solvable case of the travelling salesman problem. Oper Res 31:507–521

Roodbergen KJ (2001) Layout and routing methods for warehouses. Ph.D., thesis, Erasmus University Rotterdam, The Netherlands

Roodbergen KJ, De Koster R (2001) Routing methods for warehouses with multiple cross aisles. Int J Prod Res 39(9):1865–1883

Theys C, Bräysy O, Dullaert W, Raa B (2010) Using a TSP heuristic for routing order pickers in warehouses. EJOR 200:755–763

Tompkins JA, White JA., Bozer YA., Frazelle EH, Tanchoco JMA, Trevino J (1996) Facilities Planning, 2nd edn. Wiley, Newyork, p 36–47