Routing order pickers in a warehouse with a middle aisle

Kees Jan Roodbergen and René de Koster
Rotterdam School of Management, Erasmus University Rotterdam,
P.O. box 1738, 3000 DR Rotterdam, The Netherlands

Please refer to this article as:

Abstract
This paper considers a parallel aisle warehouse, where order pickers can change aisles at the ends of every aisle and also at a cross aisle halfway along the aisles. An algorithm is presented that can find shortest order picking tours in this type of warehouses. The algorithm is applicable in warehouse situations with up to three aisle changing possibilities.

Average tour length is compared for warehouses with and without a middle aisle. It appears that in many cases the average order picking time can be decreased significantly by adding a middle aisle to the layout.

Keywords
dynamic programming, routing, simulation, warehousing, orderpicking

1 Introduction

In warehouses and distribution centers, products have to be picked from specified storage locations on the basis of customer orders. In general, the order picking process is the most laborious of all warehouse processes. Furthermore, it typically amounts to 55% of all warehouse operating expenses (see e.g. Tompkins et al., 1996). One way to achieve savings in equipment and the number of order pickers is by optimizing order picking tours. Given that the order picker has to collect a number of products in specified quantities at known locations, the question is in what sequence the order picker should visit these locations in order to minimize the distance traveled?

The basic warehouse layout is one with parallel aisles, a central depot and possibilities for changing aisles at the front and rear of the warehouse. Such a warehouse is said to have two cross aisles, that is two possibilities to go from one aisle to another aisle. The problem of finding shortest order picking tours for a warehouse with basic layout can be solved in running time linear in the number of aisles and the number of pick locations, see Ratliff and Rosenthal (1983).

In practice, the problem of finding order picking tours in a warehouse is mainly solved by the so-called S-shape heuristic in which order pickers move in a S-shape curve along the pick locations skipping the aisles where nothing has to be picked. More advanced heuristics are considered in Hall (1993). Performance comparisons between heuristics and the optimal algorithm are given in Petersen (1997) and De Koster and Van der Poort (1998).

In this paper, we construct a routing algorithm for a warehouse where aisle changing is possible at the front, the rear and in the middle of the warehouse (a warehouse with three cross aisles). This type of layout is fairly common in practice, but finding a shortest order picking tour
is not possible with the existing algorithm for the basic layout. Other methods to determine order picking tours in a warehouse with three cross aisles include branch-and-bound (see e.g. Little et al., 1963) or heuristic methods (see Vaughan and Petersen, 1999). This paper describes an efficient algorithm to determine the shortest order picking tours. Furthermore, we compare average travel time in warehouses of basic layout to average travel time in warehouses with a middle aisle. It is shown that average travel time often is lower in warehouses with a middle aisle.

In Section 2 we model the warehouse and order picking locations using graph theory. A dynamic programming formulation is given in Section 3 to solve the problem of finding a shortest order picking tour. In Section 4 we compare average tour length in warehouses with a middle aisle to the average tour length in similar warehouses without a middle aisle. Section 5 contains concluding remarks.

2 The warehouse

A warehouse consists of a number of parallel aisles. The items are stored on both sides of the aisles. Order pickers are assumed to be able to traverse the aisles in both directions and to change direction within the aisles. Each order consists of a number of items that are usually spread out over a number of aisles. We assume that the items of an order can be picked in a single tour. Aisle changes are possible at the front end, the rear end and in the middle of the aisles. Picked orders have to be deposited at the depot, where the picker also receives the instructions for the next tour.

In order to determine an order picking tour of minimum length, the travel time between each pair of adjacent (item) locations in the warehouse needs to be specified. In the specification of the travel time we can take into account the time for entering an aisle and the time for accelerating and decelerating while driving from one location to another. We will focus only on minimizing the travel time. Other order picking activities, like positioning the truck or crane at the pick location, picking items from the pick location and putting them onto a product carrier, have to be performed anyway. Therefore, they do not impact the choice of an order picking tour.

Consider a warehouse with \( n \) aisles where we can change aisles at the front, the rear and the middle of each aisle. Stated differently, we have a parallel aisle warehouse consisting of 2 blocks of each \( n \) aisles. See Figure 1(a) for an example layout. In this warehouse we have to pick an order of \( m \) items. Of these \( m \) items there are \( m_x \) items located in block \( X \) and \( m_y \) items in block \( Y \) \((m_x + m_y = m)\).

The warehouse with order picking locations can be modeled as a graph \( G \) with vertices:

\[
\begin{align*}
v_{x,i} & \ (i = 1, \ldots, m_x) \text{ corresponding to the pick locations in block } X, \\
v_{y,i} & \ (i = 1, \ldots, m_y) \text{ corresponding to the pick locations in block } Y, \\
a_i & \ (i = 1, \ldots, N) \text{ corresponding to the rear end of aisle } i, \\
b_i & \ (i = 1, \ldots, N) \text{ corresponding to the middle of aisle } i, \\
c_i & \ (i = 1, \ldots, N) \text{ corresponding to the front end of aisle } i, \\
v_{y,0} & \text{ corresponding to the depot.}
\end{align*}
\]
Any two vertices that correspond to adjacent locations in the warehouse are connected by two parallel edges. No more than two parallel edges are needed, since a shortest tour contains no more than two edges between any pair of vertices (see Corollary A1.1). The length of the edges indicates the travel times in the warehouse. See Figure 1(b) for a graph representation of Figure 1(a).

Any order picking tour will be considered as being a special kind of subgraph of the warehouse graph, and is called a tour subgraph. That is, any subgraph $T$ of the warehouse graph $G$ is called a tour subgraph if its edges form a cycle that includes the depot and each of the pick locations at least once (see Theorem A1 for a more exact description). The length of a subgraph is defined as the sum of the length of the edges in this subgraph. In Ratliff and Rosenthal (1983) an algorithm is given that constructs an order picking tour from a given tour subgraph. The problem of finding a shortest order picking tour can therefore be solved by finding a tour subgraph of minimum length.

3 Finding a shortest tour subgraph

Let $L_j^-$ be the subgraph of the warehouse graph, consisting of vertices $a_j$, $b_j$ and $c_j$ together with all edges and vertices to the left of $a_j$, $b_j$ and $c_j$. Let $Y_j$ be the subgraph of the warehouse graph consisting of vertices $b_j$ and $c_j$ together with all edges and vertices between $b_j$ and $c_j$ and define $L_j^{+y} = L_j^- \cup Y_j$. Similarly, let $X_j$ be the subgraph of the warehouse graph consisting of vertices $a_j$ and $b_j$ together with all edges and vertices between $a_j$ and $b_j$ and define $L_j^{+x} = L_j^{+y} \cup X_j$.

We use $L_j$ to indicate that a result holds if we let $L_j = L_j^-$, $L_j = L_j^{+y}$ or $L_j = L_j^{+x}$.

For any subgraph $L_j \subseteq G$, a subgraph $T_j \subseteq L_j$ is called a $L_j$ partial tour subgraph if there exists another subgraph of $G$ (called completion) consisting of edges and vertices not contained in $L_j$, such that the union of these two subgraphs forms a tour subgraph (see also Theorem A2). Two $L_j$ partial tour subgraphs are equivalent if any completion of one partial tour subgraph is a completion for the other (see also Theorem A3).

The algorithm uses the concept of dynamic programming to construct a minimum tour subgraph. We start with all $L_j^{+y}$ partial tour subgraphs consisting only of vertices and edges between $b_1$ and $c_1$. In the next step, we extend the $L_j^{+y}$ partial tour subgraphs by adding vertices and edges between $a_1$ and $b_1$ to obtain $L_j^{+x}$ partial tour subgraphs. Next, $L_j^{+x}$ partial tour subgraphs are determined by extending $L_j^{+x}$ partial tour subgraphs with edges between aisle 1 and aisle 2. From $L_j^{+x}$ partial tour subgraphs we can obtain $L_j^{+y}$ partial tour subgraphs. Continuing this way, we finally get the $L_j^{+x}$ partial tour subgraphs, which are precisely the tour subgraphs.

To describe the algorithm within the concept of dynamic programming, we define the potential states, the possible transitions between states, and the costs (tour lengths) involved in such a transition.

3.1 States

The classes of equivalent $L_j$ partial tour subgraphs can be characterized by the five features (see Theorem A2): degree parity of $a_j$, degree parity of $b_j$, degree parity of $c_j$, connectivity, and distribution of $a_j$, $b_j$ and $c_j$ over the various components.

Degree parity describes whether the number of edges incident with the vertex is odd, even or zero. The term connectivity gives the number of connected components of the partial tour subgraph. The distribution of $a_j$, $b_j$ and $c_j$ over the various components indicates which of $a_j$, $b_j$ and $c_j$ are contained in the same component.
We denote the five features in a quintuplet. The degree parity is given by \( u \) (odd, \( u \) for 'uneven'), \( e \) (even) or 0 (zero). The connectivity, giving the number of components, is an integer between 0 and 3. For ease of notation we suppress the distribution of \( a_j \), \( b_j \) and \( c_j \) over the various components if there is only one possibility, given the other four features. In fact, we only need to give the distribution of \( a_j \), \( b_j \) and \( c_j \) over the various components for the case of 2 components each having even degree parity (see Theorem A3). Therefore, the fifth feature only needs the following possibilities: \( a \)-\( bc \), \( b \)-\( ac \), \( c \)-\( ab \). For example, with \( a \)-\( bc \) we denote that \( a \) is in one component and \( b_j \) and \( c_j \) are in the other.

As an example consider the equivalence class \((u, e, u, 2)\). This class has odd degree parity in \( a_j \) and \( c_j \), even degree parity in \( b_j \) and consists of two components (the fifth feature is not given, since the only valid possibility is \( a_j \) and \( c_j \) in one component and \( b_j \) in the other. Any other possibility would violate condition (b) of Theorem A2, because if \( a_j \) has odd degree parity, then at least one other vertex in the same component must have odd degree parity as well. Theorem A2(b) implies that this other vertex must be either \( b_j \) or \( c_j \). Only \( c_j \) has odd degree parity and must therefore be in the same component as \( a_j \).

Using theorems and corollaries from the Appendix, it can be proven that the only 25 equivalence classes are:

\[
\begin{align*}
(0, 0, 0, 0), & \quad (0, 0, 0, 1), \quad (e, e, e, 1), \quad (e, e, e, 3), \\
(e, 0, 0, 1), & \quad (0, e, 0, 1), \quad (0, 0, e, 1), \quad (e, 0, e, 1), \quad (0, e, e, 1), \\
(u, u, 0, 1), & \quad (u, 0, u, 1), \quad (0, u, u, 1), \quad (e, u, u, 1), \quad (u, u, e, 1), \\
(e, e, 0, 2), & \quad (e, 0, e, 2), \quad (0, e, e, 2), \quad (e, u, u, 2), \quad (u, e, u, 2), \quad (u, u, e, 2), \\
(e, e, e, 2, a-bc), & \quad (e, e, e, 2, b-ac), \quad (e, e, e, 2, c-ab).
\end{align*}
\]

Note that \((0, 0, 0, 0)\) is only possible if none of the aisles in \( L_j \) contains an item to be picked and \((0, 0, 0, 1)\) is only possible if none of the aisles in \( G - L_j \) contain an item to be picked (\( G - L_j \) denotes the subgraph that remains after all edges and vertices in \( L_j \) have been deleted from \( G \)).

Using Theorem A1, it can be derived that after calculating the \( L_{j}^{+x} \) partial tour subgraphs, the minimum length tour subgraph is the shortest of the following partial tour subgraphs:

\[
\begin{align*}
(0, 0, 0, 1), & \quad (e, 0, 0, 1), \quad (0, 0, e, 1), \\
(e, e, 0, 1), & \quad (e, 0, e, 1), \quad (0, e, e, 1), \quad (e, e, e, 1).
\end{align*}
\]

### 3.2 Transitions

The transitions between states consist of adding vertices and edges. We distinguish three different transitions. In the first type of transition, from \( L_{j}^{-} \) to \( L_{j}^{+y} \), vertices and edges between \( b_j \) and \( c_j \) are added. In the second type, from \( L_{j}^{+y} \) to \( L_{j}^{+z} \), vertices and edges between \( a_j \) and \( b_j \) are added. In the last type, from \( L_{j}^{+x} \) to \( L_{j+1}^{-} \) the connection between aisle \( j \) and aisle \( j + 1 \) is made.

**Transition from \( L_{j}^{-} \) to \( L_{j}^{+y} \)**

Consider any aisle \( j \). We know that we never need more than two edges between any pair of vertices (see Corollary A1.1). Therefore, the edges between \( b_j \) and \( c_j \) can be configured as one of the possibilities in Figure 2. The equivalence classes that we obtain by this transition are given in table B1.

X

Insert Figure 2

X
Transition from $L_j^{+y}$ to $L_j^{+x}$

This transition is very similar to the transition from $L_j^{-}$ to $L_j^{+y}$. Again we can distinguish six ways to traverse the edges in aisle $j$ between $a_j$ and $b_j$. The equivalence classes that we obtain by this transition are given in table B2.

Transition from $L_j^{+x}$ to $L_{j+1}^{-}$

This transition makes the connection between aisle $j$ and aisle $j+1$ by adding configurations given in Figure 3. The equivalence classes that we obtain by this transition are given in table B3.

3.3 Cost

The cost of each transition is equal to the sum of the lengths of the edges added in the transition.

3.4 Applicability

The algorithm is applicable to a wide variety of warehouses. Clearly, the middle aisle does not have to be exactly in the middle, but can be placed anywhere between the front and the rear of the warehouse. Furthermore, the algorithm can be used in warehouses with only one or two possibilities for aisle changing, by setting the appropriate distances between the aisles to infinity. In this way, the algorithm can also be used to find order picking tours in warehouses with basic layout. In spite of its apparent complexity, the algorithm solves any practical sized problem within fractions of a second.

The algorithm considers all aisles and items, and for each aisle and item a constant number of operations has to be done. Hence, the time-complexity function of the algorithm is linear in the number of aisles and the number of items.

4 Performance comparison

In this section a performance comparison is made between warehouses with a middle aisle and warehouses without a middle aisle. In order to compare the two types of warehouse layouts, we use simulation to determine the average travel time needed to pick an order. However, average travel time is not only influenced by the presence or absence of a middle aisle, but also by factors like warehouse type, warehouse size, number of aisles, location of the depot, order picking equipment, picklist size, storage assignment rules, and the location of the middle aisle (if present).

For our comparisons, we consider a very common type of warehouse, namely a shelf area. Order pickers walk through this warehouse to pick items, using a small pick cart. Before starting a tour, order pickers collect a picklist at the depot, which we assume to be located at the head of the left most aisle. Other positions of the depot are possible, but the exact location has only a minor influence on average travel time (see Petersen, 1997). The average walking speed in this type of warehouses is usually around 0.6 meters per second. The distance between two neighboring aisles is 2.5 meters. Products are assumed to be assigned randomly to storage
locations according to a uniform distribution. Average travel time is determined by taking the average of the travel times of 10,000 simulated orders.

If we consider this warehouse type, then three important factors influencing travel time remain, namely (1) warehouse size, (2) warehouse layout and (3) picklist size. Warehouse layout covers (a) the presence or absence of a middle aisle, (b) the number of aisles and (c) the location of the middle aisle. Aisle length is determined as the ratio of warehouse size and the number of aisles. For this analysis we assume that a middle aisle will always be located in the exact middle of the picking aisles. One may consider a middle aisle that is closer to the rear of the warehouse than to the front. This might actually be better than a location in the exact middle for example in a situation with two aisles and many items to pick. In this case a 'good' route visits all pick locations in the first aisle, goes to the second aisle through the nearest cross aisle and then visits all pick locations in the second aisle. A good location for the middle aisle would then be around the expected location of the pick that is closest to the rear of the warehouse. However, in most situations with a moderate number of items and aisles, the exact middle will be close to optimal due to the uniform distribution of the items. Besides, for practical reasons of flexibility it may be undesirable to locate the middle aisle anywhere else than in the exact middle. If at some point the depot location is changed from front to back then the initial advantage of an eccentric middle aisle may turn into a disadvantage.

In this section, we will evaluate two warehouse sizes, namely a warehouse with a total aisle length of 70 meters and one with a total aisle length of 450 meters. The size of a shelf area served by one order picker is in practice generally between these two extremes. Now, we can determine the optimal layout (i.e. the number of aisles) for the warehouse with a middle aisle and for the warehouse without a middle aisle for each picklist size. Then we compare average travel time in a warehouse of optimal layout with a middle aisle to average travel time in a warehouse of optimal layout without a middle aisle.

Firstly, we determine average travel time for picklists of 30 items. The number of aisles is varied from 1 to 50. Total aisle length is kept constant at 450 meters. Thus, we consider warehouses with the following layouts: 1 aisle of 450 meters, 2 aisles of 225 meters, ..., 50 aisles of 9 meters. In each of the 10,000 replications per warehouse layout, a random order is generated for the warehouse without middle aisle and the travel time is calculated. Thereafter, a middle aisle is inserted in the layout and the travel time is again calculated for the same order.

Figure 4 depicts the average travel time as a function of the number of aisles. One curve gives the results for a warehouse without a middle aisle, the other curve for a warehouse with a middle aisle. The curve for the warehouse without middle aisle is above the other curve for all configurations, except for warehouses with one or two aisles. It follows that average travel time is lower for the warehouse with a middle aisle for any number of aisles, with the exception of warehouses with 1 or 2 aisles. Also, the best possible layout without a middle aisle (27 aisles of 16.7 meters with an average travel time of 630 seconds) results in a considerably higher travel time than the best possible layout with a middle (22 aisles of 20.5 meters with an average travel time of 527 seconds). The small peak for 3 aisles in the curve for the warehouse without cross aisles, is due to the fact that at least one of the aisles has to be entered and left from the same side to ensure that the order picker ends his tour at the front of the warehouse.

We have analyzed average travel time for warehouses with and without a middle aisle for a fixed picklist size of 30 items. Also, we determined the optimal number of aisles in each of the
two types of warehouses by locating the minimum of the curves in Figure 4. In a similar fashion, we determine the optimal number of aisles for all picklist sizes ranging from 1 to 50 items. Total aisle length is kept constant at 450 meters. Thus, for each picklist size we determine the optimal number of aisles separately. The corresponding average travel time is depicted in Figure 5 as a function of the picklist size. There are two curves in Figure 5, one curve for a warehouse layout with a middle aisle and one for a layout without a middle aisle. It appears from this figure that for all picklist sizes (except size 1), the layout with a middle aisle gives lower average travel times. Savings on travel time of more than 15% are possible.

Repeating the previous experiment for warehouses with a total aisle length of 70 meters gives the results as depicted in Figure 6. As before we determine for each picklist size separately the optimal number of aisles for a warehouse without a middle aisle and for a warehouse with a middle aisle. Average travel time for the optimal number of aisles is depicted as a function of the picklist size. In this warehouse, a middle aisle is beneficial only for picklist sizes from 3 to 22 items. For picklists containing 1, 2 or more than 22 items, a layout without a middle aisle would give lower average travel times. In fact, if we would continue to increase the picklist size beyond 50 items, then at some point the two curves for the warehouse with 450 meters of aisles (Figure 5) will also intersect. This can be explained as follows. For small picklists, the introduction of a middle aisle offers more possibilities for creating tours. This will result in shorter tours. If the number of picks increases, then at some point the optimal tour will traverse nearly every aisle entirely. In this situation the middle aisle is not in use anymore to skip some aisles, but order pickers still have to cross the middle aisle. This results in higher travel times compared to a situation without middle aisle. However, for most practically sized orders a middle aisle will give savings on travel time in such a large warehouse.

5 Concluding remarks

Average travel time in warehouses depends on many factors such as warehouse type, warehouse size, number of aisles, location of the depot, order picking equipment, picklist size and storage assignment rules. Each of these factors may have a significant influence on travel time. In this paper we have evaluated the impact on average travel time of warehouse layout. Specifically we evaluated whether or not a middle aisle could improve the efficiency. To this end, we have constructed a dynamic programming algorithm for calculating order picking tours of minimal length in warehouses with up to three cross aisles. The algorithm is more complex than the algorithm for the basic layout with two cross aisles (Ratliff and Rosenthal, 1983). For the basic layout only 7 equivalence classes are needed, whereas this algorithm needs 25 equivalence classes. Calculations for any practically sized problem only take fractions of a second. Further extensions to more cross aisles are clearly possible, but may not be interesting for practice.

In a simulation experiment, we used the algorithm to determine average travel time for two warehouse sizes and varying picklist sizes. We determined the number of aisles that minimize
average travel time for each combination of warehouse size and picklist size. The number of aisles was determined separately for the situation that there is a middle aisle and for the situation that there is no middle aisle. In the majority of the situations evaluated it appeared that the layout with a middle aisle resulted in lower average travel time than the basic layout. Possible efficiency gains by introducing a middle aisle are especially present for large warehouses.

6 References


7 Appendix

A subgraph \( T \subset G \) is a tour subgraph if and only if
(a) all vertices \( v_i \) for \( i = 0, 1, 2, ..., m \) have positive degree in \( T \),
(b) excluding vertices with zero degree, \( T \) is connected,
(c) every vertex in \( T \) has even or zero degree.

Corollary A1.1 Ratliff and Rosenthal (1983)
A minimum length tour subgraph contains no more than two edges between any pair of vertices.

Corollary A1.2 Ratliff and Rosenthal (1983)
If \( (P_1, P_2) \) is any node partition of a tour subgraph, there is an even number of edges with one end in \( P_1 \) and the other end in \( P_2 \).

Theorem A2
Necessary and sufficient conditions for \( T_j \subset L_j \) to be an \( L_j \) partial tour subgraph are
(a) for all \( v_i \in L_j \), the degree of \( v_i \) is positive in \( T_j \),
(b) every vertex, except possibly \( a_j, b_j \) and \( c_j \), has even degree or zero degree,
(c) excluding vertices with zero degree, \( T_j \) has either
   - no connected component,
   - a single connected component containing at least one of \( a_j, b_j \) and \( c_j \),
   - two connected components with in each component at least one of \( a_j, b_j \) and \( c_j \), and each of \( a_j, b_j \) and \( c_j \) contained in at most one component,
   - three connected components with \( a_j, b_j \) and \( c_j \) each in a different component.

Proof
The proof is similar to that of Theorem 2 of Ratliff and Rosenthal (1983).

**Theorem A3**

Two $L_j$ partial tour subgraphs are equivalent if

(a) $a_j$, $b_j$ and $c_j$ each have the same degree parity (i.e. even, odd or zero) in both partial tour subgraphs.

(b) excluding vertices with zero degree, both partial tour subgraphs have either

- no connected component,
- a single connected component containing at least one of $a_j$, $b_j$ and $c_j$,
- two connected components with at least one of $a_j$, $b_j$ and $c_j$, and each of $a_j$, $b_j$ and $c_j$ contained in at most one component,
- three connected components with $a_j$, $b_j$ and $c_j$ each in a different component.

(c) the distribution of $a_j$, $b_j$ and $c_j$ over the various components is the same for both partial tour subgraphs.

**Proof**

The theorem and therefore the proof is largely similar to Theorem 2 of Ratliff and Rosenthal (1983), only condition (c) is added.

Note: For the layout considered in Ratliff and Rosenthal (1983), condition (c) is always satisfied. However, with a middle aisle this is not the case for 2 components with each of $a_j$, $b_j$ and $c_j$ having even degree parity. In all other cases, it can be proven that (c) is satisfied if (a) and (b) are satisfied.

8 Tables

Table B1 gives the possible transitions from $L_j^-$ equivalence classes to $L_j^{+\#}$ equivalence classes. Table B2 for $L_j^{+\#}$ to $L_j^{+x}$ and table B3 for $L_j^{+x}$ to $L_{j+1}^-$. Denote an entry in row $i$, column $j$ by $(i,j)$. An entry $(i,j)$ gives the equivalence class resulting from adding the edge configuration of column $j$ (as depicted in either Figure 2 or 3) to the equivalence class of row $i$. Edge configurations from Figure 2 are used for Tables B1 and B2; edge configurations from Figure 3 are used for Table B3.

Some of the transitions are not possible or will never lead to the optimal solution. We can distinguish 2 reasons why a transition between two equivalence classes is not possible. (1) The transition would give a configuration violating condition (b) of Theorem A2. (2) The transition would give a configuration violating condition (c) of Theorem A2.

For example, consider Table B3. Adding edge configuration (2) from Figure 3 to equivalence class $(u,u,0,1)$ may seem to result in equivalence class $(u,0,u,1)$. However, this would violate condition (b) of Theorem A2. The theorem implies that after the transition $a_j$, $b_j$ and $c_j$ should have even degree or zero degree and that $a_{j+1}$, $b_{j+1}$ and $c_{j+1}$ may have odd degree. Adding configuration (2) to class $(u,u,0,1)$ would leave $b_j$ and $c_j$ with odd degree and is therefore not permitted. Now in Table B3, consider adding arc configuration (1) from Figure 3 to class $(u,u,e,2)$. Theorem A2(c) requires that after the transition each of the two components contains at least one of $a_{j+1}$, $b_{j+1}$ and $c_{j+1}$. Since $a_j$ and $b_j$ are in the same component, $a_{j+1}$ and $b_{j+1}$ will be in the same component as well. So the connection with the second component (containing $c_j$) will be lost with this transition. Therefore it is not allowed.
### 8.1 Table B1

<table>
<thead>
<tr>
<th>$L_j^-$ to $L_j^{+y}$</th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
<th>(6) $^a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0,0,0,0) $^b_j$</td>
<td>(0,u,u,1)</td>
<td>(0,e,e,1)</td>
<td>(0,e,0,1)</td>
<td>(0,e,e,2)</td>
<td>(0,0,0,0)</td>
<td></td>
</tr>
<tr>
<td>(0,0,0,1) $^b_j$</td>
<td>e)</td>
<td>e)</td>
<td>e)</td>
<td>e)</td>
<td>e)</td>
<td>(0,0,0,1)</td>
</tr>
<tr>
<td>(e,0,0,1)</td>
<td>(e,u,u,2)</td>
<td>(e,e,e,2,a-bc)</td>
<td>(e,e,0,2)</td>
<td>(e,0,e,2)</td>
<td>(e,e,e,3)</td>
<td>(e,0,0,1)</td>
</tr>
<tr>
<td>(0,e,0,1)</td>
<td>(0,u,u,1)</td>
<td>(0,e,e,1)</td>
<td>(0,e,0,1)</td>
<td>(0,e,e,2)</td>
<td>(0,0,e,2)</td>
<td>(0,e,0,1)</td>
</tr>
<tr>
<td>(e,e,0,1)</td>
<td>(e,u,u,1)</td>
<td>(e,e,e,1)</td>
<td>(e,e,0,1)</td>
<td>(e,e,e,2,a-bc)</td>
<td>(e,e,e,2,c-ab)</td>
<td>(e,e,0,1)</td>
</tr>
<tr>
<td>(e,0,e,1)</td>
<td>(e,u,u,1)</td>
<td>(e,e,e,1)</td>
<td>(e,e,2,b-ac)</td>
<td>(e,0,e,1)</td>
<td>(e,e,2,b-ac)</td>
<td>(e,0,e,1)</td>
</tr>
<tr>
<td>(0,e,1)</td>
<td>(0,u,u,1)</td>
<td>d)</td>
<td>(0,e,e,1)</td>
<td>(0,e,e,1)</td>
<td>(0,e,1)</td>
<td>(0,e,1)</td>
</tr>
<tr>
<td>(e,e,1)</td>
<td>(e,u,u,1)</td>
<td>d)</td>
<td>(e,e,1)</td>
<td>(e,e,1)</td>
<td>(e,e,1)</td>
<td>(e,e,1)</td>
</tr>
<tr>
<td>(u,u,0,1)</td>
<td>(u,e,u,1)</td>
<td>(u,u,e,1)</td>
<td>(u,u,0,1)</td>
<td>(u,u,e,2)</td>
<td>(u,u,0,2)</td>
<td>(u,u,0,1)</td>
</tr>
<tr>
<td>(u,0,0,1)</td>
<td>(u,e,u,1)</td>
<td>(u,e,u,1)</td>
<td>(u,u,2)</td>
<td>(u,0,u,1)</td>
<td>(u,u,2)</td>
<td>(u,0,0,1)</td>
</tr>
<tr>
<td>(0,0,1)</td>
<td>(0,e,1)</td>
<td>d)</td>
<td>(0,u,u,1)</td>
<td>(0,u,1)</td>
<td>(0,u,1)</td>
<td>(0,0,1)</td>
</tr>
<tr>
<td>(e,0,1)</td>
<td>(e,u,1)</td>
<td>d)</td>
<td>(e,u,1)</td>
<td>(e,u,1)</td>
<td>(e,u,1)</td>
<td>(e,u,1)</td>
</tr>
<tr>
<td>(u,e,1)</td>
<td>(u,u,1)</td>
<td>d)</td>
<td>(u,u,1)</td>
<td>(u,u,1)</td>
<td>(u,u,1)</td>
<td>(u,e,1)</td>
</tr>
<tr>
<td>(e,e,0,2)</td>
<td>(e,u,2)</td>
<td>(e,e,e,2,a-bc)</td>
<td>(e,e,0,2)</td>
<td>(e,e,3)</td>
<td>(e,e,3)</td>
<td>(e,e,0,2)</td>
</tr>
<tr>
<td>(e,e,2)</td>
<td>(e,u,2)</td>
<td>(e,e,e,2,a-bc)</td>
<td>(e,e,3)</td>
<td>(e,e,3)</td>
<td>(e,e,3)</td>
<td>(e,e,2)</td>
</tr>
<tr>
<td>(0,e,2)</td>
<td>(0,u,1)</td>
<td>(0,e,1)</td>
<td>(0,e,2)</td>
<td>(0,e,2)</td>
<td>(0,e,2)</td>
<td>(0,e,2)</td>
</tr>
<tr>
<td>(e,e,2,a-bc)</td>
<td>(e,u,2)</td>
<td>d)</td>
<td>(e,e,e,2,a-bc)</td>
<td>(e,e,2,a-bc)</td>
<td>(e,e,2,a-bc)</td>
<td>(e,e,2,a-bc)</td>
</tr>
<tr>
<td>(e,e,2,b-ac)</td>
<td>(e,u,1)</td>
<td>(e,e,e,1)</td>
<td>(e,e,2,b-ac)</td>
<td>(e,e,2,b-ac)</td>
<td>(e,e,2,b-ac)</td>
<td>(e,e,2,b-ac)</td>
</tr>
<tr>
<td>(e,e,2,c-ab)</td>
<td>(e,u,1)</td>
<td>(e,e,e,1)</td>
<td>(e,e,2,c-ab)</td>
<td>(e,e,2,c-ab)</td>
<td>(e,e,2,c-ab)</td>
<td>(e,e,2,c-ab)</td>
</tr>
<tr>
<td>(e,u,2)</td>
<td>(e,e,2,a-bc)</td>
<td>d)</td>
<td>(e,u,2)</td>
<td>(e,u,2)</td>
<td>(e,u,2)</td>
<td>(e,u,2)</td>
</tr>
<tr>
<td>(u,e,2)</td>
<td>(u,e,1)</td>
<td>(u,e,1)</td>
<td>(u,e,2)</td>
<td>(u,e,2)</td>
<td>(u,e,2)</td>
<td>(u,e,2)</td>
</tr>
<tr>
<td>(u,e,3)</td>
<td>(u,e,2)</td>
<td>(e,e,e,2,a-bc)</td>
<td>(e,e,3)</td>
<td>(e,e,3)</td>
<td>(e,e,3)</td>
<td>(e,e,3)</td>
</tr>
</tbody>
</table>

---

a) This transition is only allowed if there are no items in this part of the aisle  
b) This class can occur only if there are no items to be picked in $L_j^-$  
c) This class can only occur if there are no items to be picked in $G - L_j^-$  
d) This transition will never lead to the optimal solution  
e) This transition would violate condition A2(c)
8.2 Table B2

<table>
<thead>
<tr>
<th>$L_j^y$ to $L_j^x$</th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
<th>(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0,0,0,0)</td>
<td>(u,u,0,1)</td>
<td>(e,e,0,1)</td>
<td>(e,0,0,1)</td>
<td>(e,0,0,1)</td>
<td>(e,e,0,2)</td>
<td>(0,0,0,0)</td>
</tr>
<tr>
<td>(0,0,0,1)</td>
<td>(0,0,0,1)</td>
<td>(0,0,0,1)</td>
<td>(0,0,0,1)</td>
<td>(0,0,0,1)</td>
<td>(0,0,0,1)</td>
<td>(0,0,0,1)</td>
</tr>
<tr>
<td>(0,0,1)</td>
<td>(0,0,1)</td>
<td>(0,0,1)</td>
<td>(0,0,1)</td>
<td>(0,0,1)</td>
<td>(0,0,1)</td>
<td>(0,0,1)</td>
</tr>
<tr>
<td>(0,1)</td>
<td>(0,1)</td>
<td>(0,1)</td>
<td>(0,1)</td>
<td>(0,1)</td>
<td>(0,1)</td>
<td>(0,1)</td>
</tr>
<tr>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
</tr>
</tbody>
</table>

a) This transition is only allowed if there are no items in this part of the aisle
b) This class can occur only if there are no items to be picked in $L_j^y$
c) This class can only occur if there are no items to be picked in $G - L_j^y$
d) This transition will never lead to the optimal solution
e) This transition would violate condition A2(c)
### 8.3 Table B3

#### $L_{j}^{x}$ to $L_{j+1}^{-}$

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(7)</th>
<th>(8)</th>
<th>(9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(u,u,0,1)$</td>
<td>$(u,u,0,1)$</td>
<td>a)</td>
<td>a)</td>
<td>a)</td>
<td>a)</td>
<td>c)</td>
</tr>
<tr>
<td>$(u,0,u,1)$</td>
<td>(u,0,u,1)</td>
<td>a)</td>
<td>a)</td>
<td>(0,u,u,1)</td>
<td>c)</td>
<td>a)</td>
</tr>
<tr>
<td>$(u,u,u,1)$</td>
<td>(u,u,u,1)</td>
<td>a)</td>
<td>a)</td>
<td>(0,u,u,1)</td>
<td>c)</td>
<td>a)</td>
</tr>
<tr>
<td>$(u,e,u,1)$</td>
<td>(u,e,u,1)</td>
<td>a)</td>
<td>(u,0,u,1)</td>
<td>a)</td>
<td>(u,u,u,1)</td>
<td>a)</td>
</tr>
<tr>
<td>$(u,u,e,1)$</td>
<td>(u,u,0,1)</td>
<td>a)</td>
<td>a)</td>
<td>a)</td>
<td>a)</td>
<td>(u,u,e,1)</td>
</tr>
<tr>
<td>$(e,u,2)$</td>
<td>a)</td>
<td>a)</td>
<td>b)</td>
<td>(e,u,2)</td>
<td>a)</td>
<td>a)</td>
</tr>
<tr>
<td>$(u,e,2)$</td>
<td>a)</td>
<td>b)</td>
<td>a)</td>
<td>a)</td>
<td>(u,e,2)</td>
<td>a)</td>
</tr>
<tr>
<td>$(u,u,2)$</td>
<td>b)</td>
<td>a)</td>
<td>a)</td>
<td>a)</td>
<td>a)</td>
<td>(u,u,2)</td>
</tr>
</tbody>
</table>

#### $L_{j}^{x}$ to $L_{j+1}^{-}$

<table>
<thead>
<tr>
<th></th>
<th>(4)</th>
<th>(5)</th>
<th>(6)</th>
<th>(10)</th>
<th>(11)</th>
<th>(12)</th>
<th>(13)</th>
<th>(14)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(0,0,0,0)$</td>
<td>c)</td>
<td>c)</td>
<td>c)</td>
<td>c)</td>
<td>c)</td>
<td>c)</td>
<td>c)</td>
<td>$(0,0,0,0)$</td>
</tr>
<tr>
<td>$(0,0,0,1)$</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>$(0,0,0,1)$</td>
</tr>
<tr>
<td>$(e,0,0,1)$</td>
<td>(e,0,0,1)</td>
<td>b)</td>
<td>b)</td>
<td>c)</td>
<td>b)</td>
<td>c)</td>
<td>b)</td>
<td>$(0,0,0,1)$</td>
</tr>
<tr>
<td>$(0,0,1)$</td>
<td>b)</td>
<td>(0,e,0,1)</td>
<td>b)</td>
<td>(0,e,0,1)</td>
<td>c)</td>
<td>b)</td>
<td>c)</td>
<td>$(0,0,0,1)$</td>
</tr>
<tr>
<td>$(e,0,0,1)$</td>
<td>(e,0,0,1)</td>
<td>(e,0,0,1)</td>
<td>(0,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>(e,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>$(0,0,0,1)$</td>
</tr>
<tr>
<td>$(e,0,1)$</td>
<td>(e,0,1)</td>
<td>b)</td>
<td>(0,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>c)</td>
<td>(0,e,0,1)</td>
<td>c)</td>
<td>(0,e,0,1)</td>
</tr>
<tr>
<td>$(0,e,1)$</td>
<td>(0,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>(e,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>$(0,0,0,1)$</td>
</tr>
<tr>
<td>$(e,e,1)$</td>
<td>(e,e,1)</td>
<td>(e,e,1)</td>
<td>(e,e,0,1)</td>
<td>(e,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>(e,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>$(0,0,0,1)$</td>
</tr>
<tr>
<td>$(e,e,0,1)$</td>
<td>(e,e,0,1)</td>
<td>(e,e,0,1)</td>
<td>(e,e,0,1)</td>
<td>(e,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>(e,e,0,1)</td>
<td>(0,e,0,1)</td>
<td>$(0,0,0,1)$</td>
</tr>
<tr>
<td>$(e,e,2)$</td>
<td>(e,e,2)</td>
<td>b)</td>
<td>(e,e,2)</td>
<td>b)</td>
<td>b)</td>
<td>(e,e,2)</td>
<td>b)</td>
<td>c)</td>
</tr>
<tr>
<td>$(e,e,2)$</td>
<td>(e,e,2)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>(e,e,2)</td>
<td>b)</td>
<td>c)</td>
</tr>
<tr>
<td>$(e,e,2)$</td>
<td>(e,e,2)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>(e,e,2)</td>
<td>b)</td>
<td>c)</td>
</tr>
<tr>
<td>$(e,e,2,a-bc)$</td>
<td>(e,e,2,a-bc)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>(e,e,2,a-bc)</td>
<td>b)</td>
<td></td>
</tr>
<tr>
<td>$(e,e,2,b-ac)$</td>
<td>(e,e,2,b-ac)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>(e,e,2,b-ac)</td>
<td>b)</td>
<td></td>
</tr>
<tr>
<td>$(e,e,2,c-ab)$</td>
<td>(e,e,2,c-ab)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>(e,e,2,c-ab)</td>
<td>b)</td>
<td></td>
</tr>
<tr>
<td>$(e,e,3)$</td>
<td>(e,e,3)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>b)</td>
<td>(e,e,3)</td>
<td>b)</td>
<td></td>
</tr>
</tbody>
</table>

- **a)** This transition would violate condition A2(b)
- **b)** This transition would violate condition A2(c)
- **c)** This transition will never lead to the optimal solution
- **d)** This class can occur only if there are no items to be picked in $L_{j}^{x}$
- **e)** This class can only occur if there are no items to be picked in $G - L_{j}^{x}$
- **f)** This transition is only allowed if there are no items to be picked in $G - L_{j}^{x}$

All combinations not in this table would violate condition A2(b).
Figure 1: Part (a) gives an example layout of a warehouse consisting of two blocks. Each solid square corresponds to a location, where items have to be picked. Part (b) gives a graph representation of this warehouse with the pick locations.

Figure 2: Six ways to traverse the edges in aisle $j$ between $b_i$ and $c_j$. In transition (5) only the longest double edge is not traversed. Transitions (3) an (4) are only possible if there is at least one item in this part of the aisle. Transition (5) is only possible if there are two or more items in this part of the aisle and transition (6) is only allowed if this part of the aisle is empty.
Figure 3: 14 possible arc configurations to change from aisle $j$ to aisle $j+1$.

Figure 4: Average travel time for an order of 30 items as a function of the number of aisles for a warehouse with 450 meters of aisles, with a middle aisle (dashed line) and without a middle aisle (solid line).
Figure 5: Average travel time as a function of the picklist size for a warehouse with 450 meters of aisles, with a middle aisle (dashed line) and without a middle aisle (solid line).

Figure 6: Average travel time as a function of the picklist size for a warehouse with 70 meters of aisles, with a middle aisle (dashed line) and without a middle aisle (solid line).