

УДК 658.5

## Процесс комплектования заказов на складе. Задача маршрутизации сборщиков заказов

Коробков Е. В.<sup>1,\*</sup>

[\\*eug.korobkov@gmail.com](mailto:eug.korobkov@gmail.com)

<sup>1</sup>МГТУ им. Н.Э. Баумана, Москва, Россия

---

Задача маршрутизации сборщиков заказов на складе является наиболее трудозатратной подзадачей процесса комплектования заказов. Перемещение сборщиков по складу во время ее выполнения достигает до 50% от всех временных затрат. В статье приводятся аналогии задачи маршрутизации сборщиков заказов на складе с частной формулировкой задачи коммивояжера, описывается оптимальный алгоритм Рэтлиффа-Розенталя для решения задачи маршрутизации сборщиков заказов на одноблочном складе, а также его расширение на большее число блоков, приводятся причины, по которым на практике использование точных оптимальных алгоритмов заменяется использованием эвристик, а также представлены формулировки восьми эвристических методов маршрутизации сборщиков заказов на складе. Статья является второй в цикле статей «Процесс комплектования заказов на складе».

**Ключевые слова:** склад, комплектование, заказ, маршрутизация, сборщик, коммивояжер, эвристики

---

### Введение

Среди складских функций можно выделить комплектование заказов (*order picking*) как наиболее важную. Комплектование заказов — это процесс поиска и извлечения товаров из ячеек хранения с целью удовлетворения потребительского заказа. Подзадачами процесса комплектования заказов являются:

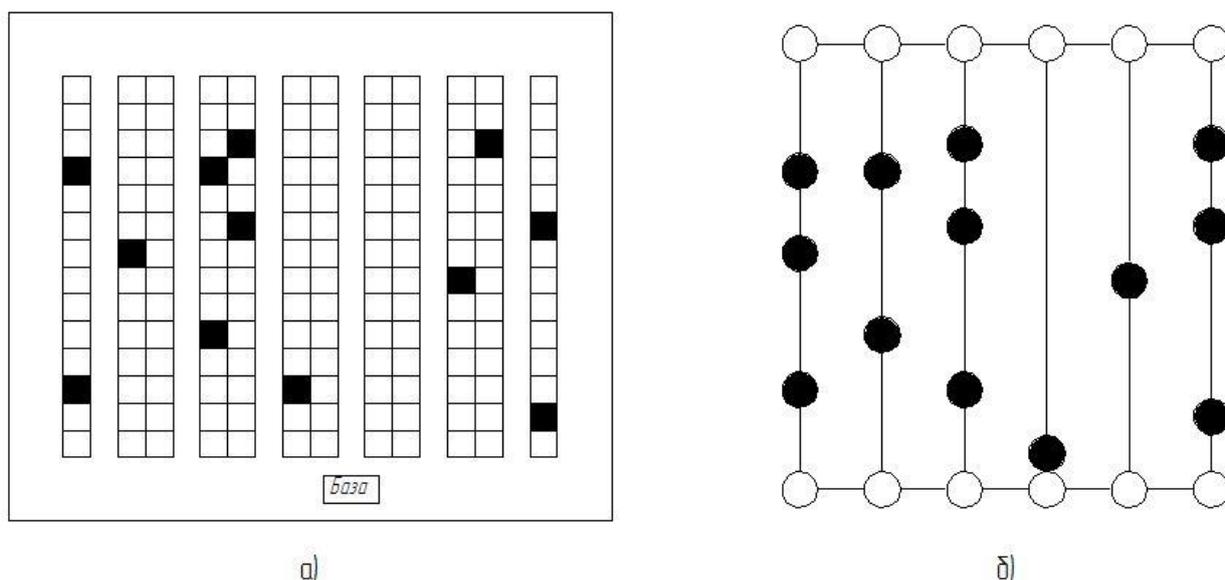
- проектирование планировки (*layout design*) и размеров (*dimensioning*) складской системы;
- распределение товаров по ячейкам хранения (*storage assignment*);
- распределение по партиям (*batching*) и зонирование (*zoning*);
- накопление заказов/сортировка (*accumulation/sortation*);
- маршрутизация (*routing*) сборщиков заказов.

Первые четыре подзадачи рассмотрены в первой статье цикла «Процесс комплектования заказов на складе» [1]. Данная статья посвящена подзадаче маршрутизации сборщиков заказов. Структура статьи следующая. В п. 1 задача маршрутизации сборщиков заказов описывается как частный случай задачи коммивояжера. В п. 2 представлено получение оптимального алгоритма решения задачи маршрутизации сборщиков заказов для одноблочных складов. В п. 3 полученный

алгоритм расширяется на многоблочные склады, а также приводятся недостатки использования точных методов. В п. 4 представлены эвристические методы решения задачи маршрутизации сборщиков заказов.

## 1. Штейнеровская задача коммивояжера

Задача маршрутизации сборщиков заказов (*order-pickers routing problem*) является частным случаем задачи коммивояжера (*traveling salesman problem*) [2]. Задача коммивояжера получила свое название благодаря следующей ситуации. Коммивояжеру требуется посетить определенное число городов только один раз и вернуться обратно в родной город на место старта. Он знает расстояния между каждой парой городов и хочет определить такой порядок их посещения, чтобы общее пройденное расстояние было минимальным. Очевидно, что ситуация с коммивояжером имеет несколько точек соприкосновения с задачей сборщиков заказов на складе. Сборщик стартует с базы, т.е. места получения заказа, (*depot*, родного города), где получает сборочный лист (*picking list*), содержащий строки заказа, затем ему необходимо «посетить» определенное число сборочных ячеек, т.е. ячеек с необходимым товаром (*picking cells*, городов) и, наконец, вернуться обратно на базу. Пример планировки склада с типовым заданием на сборку заказа, а также соответствующее представление в виде графа проиллюстрированы на рис. 1.



**Рис. 1.** а) Планировка склада с типовым заданием на сборку (затрихованные прямоугольники обозначают сборочные ячейки) и б) представление планировки склада с типовым заданием на сборку в виде графа (заштрихованные вершины — сборочные ячейки и база; незаштрихованные — пересечения проходов)

Между стандартной формулировкой задачи коммивояжера и ситуацией со сборщиком заказов на складе существует несколько различий. Во-первых, если взглянуть на рис. 1, можно заметить, что некоторые вершины-«города» могут остаться не посещенными (незаштрихованные вершины). Эти вершины являются пересечениями продольных (*crossing*) и поперечных (или, сборочных, *picking*) проходов (*aisles*). Сборщик заказов может, но не обязан посещать их. Заштрихованные вершины представляют собой

сборочные ячейки и базу, которые сборщику заказов посетить необходимо. Разрешается посещать сборочные ячейки и/или базу более одного раза. Задача маршрутизации сборщиков заказов классифицируется как штейнеровская задача коммивояжера (*Steiner traveling salesman problem*) по двум причинам — некоторые вершины могут остаться посещенными, а некоторые — посещены более одного раза. Штейнеровская задача коммивояжера является *NP*-трудной. Однако, для планировок складов, аналогичных представленной на рис. 1 (в виде последовательно-параллельных графов, *parallel-series graphs*), существует алгоритм, позволяющий найти решение задачи за линейное время в зависимости от числа проходов и сборочных ячеек [3].

## 2. Оптимальный алгоритм Рэтлиффа-Розенталя решения задачи маршрутизации сборщиков заказов для одноблочных складов

Рассмотрим заказ, содержащий  $m$  элементов (товаров), которые необходимо собрать на одноблочном (с двумя поперечными проходами, *single-block*) складе, конфигурация которого представлена на рис. 1.

### 2.1. Представление планировки склада и типового заказа в виде графа

Определим граф  $G$  как совокупность вершин  $v_i$ , представляющих собой местоположение каждого из  $i = 1, 2, \dots, m$  элементов заказа (т.е. расположение сборочных ячеек), и вершин  $a_j$  и  $b_j$ , являющих собой начало и конец каждого из  $j = 1, 2, \dots, n$  проходов. Все смежные вершины графа соединяем дугами, длины которых соответствуют реальному расстоянию между точками склада, представленными смежными вершинами графа. Граф, описывающий конфигурацию склада, заданную на рис. 1, представлен на рис. 2. Для наглядности, между совпадающими вершинами  $v_0$  и  $b_4$  добавлена дуга нулевой длины.

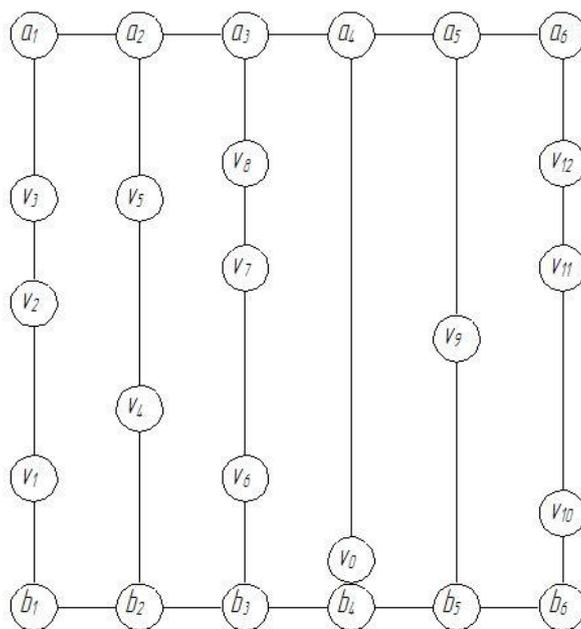


Рис. 2. Граф, описывающий конфигурацию склада, представленную на рис. 1

## 2.2. Эквивалентные классы частичных маршрутных подграфов (ЧМП)

Маршрут сборщика заказов — это цикл в  $G$ , включающий в себя все вершины  $v_i, i = 0, 1, \dots, m$ , хотя бы один раз. Задачей маршрутизации сборщика заказов является нахождение минимального такого цикла.

Подграф  $T \subset G$ , который содержит все вершины  $v_i, i = 0, 1, \dots, m$ , назовем маршрутным подграфом (*tour subgraph*), если существует сборочный маршрут, использующий каждую дугу из  $T$  только один раз.

Таким образом, подграф  $T \subset G$  является маршрутным подграфом тогда, и только тогда, когда а) все вершины  $v_i, i = 0, 1, 2, \dots, m$  имеют положительную степень в  $T$ ; б) при исключении всех вершин нулевой степени граф  $T$  является связным; в) каждая вершина в  $T$  имеет четную, либо нулевую степень.

Для нахождения минимального маршрута обхода сборщиком заказов сборочных ячеек необходимо найти минимальный маршрутный подграф, то есть маршрутный подграф минимальной длины, при этом доказывается, что такой подграф не имеет вершин, соединенных друг с другом более чем двумя дугами [3].

Для любого подграфа  $L \subset G$  подграф  $T_j \subset L$  является частичным маршрутным подграфом (ЧМП, *partial tour subgraph, PTS*) для  $L$ , если существует подграф  $C_j \subset G - L$  ( $G - L$  определяет граф, получающийся удалением всех вершин и дуг графа  $L$  из графа  $G$ ) такой, что  $T_j \cup C_j$  является маршрутным подграфом  $G$ . Подграф  $C_j$  назовем дополнением (*completion*) к  $T_j$ .

Положим, что  $L_j^-$  — подграф  $G$ , включающий вершины  $a_j$  и  $b_j$ , а также все вершины и дуги из  $G$  по левую сторону от  $a_j$  и  $b_j$ . Положим, что  $A_j$  — подграф  $G$ , состоящий из вершин  $a_j$  и  $b_j$ , а также всех вершин и дуг из  $G$  между  $a_j$  и  $b_j$ . Наконец, положим, что  $L_j^+ = L_j^- \cup A_j$ . Рис. 3 иллюстрирует данные определения. Обозначение  $L_j$  используется для случаев, когда результат справедлив как для  $L_j = L_j^-$ , так и для  $L_j = L_j^+$ .

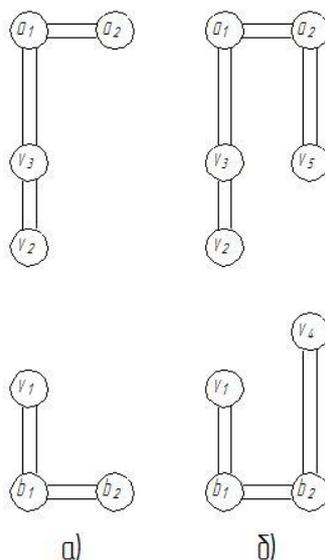


Рис. 3. а)  $L_2^-$  ЧМП; б)  $L_2^+$  ЧМП для графа, представленного на рис. 2

Существуют следующие необходимые и достаточные условия, при которых подграф  $T_j \subset L_j$  будет ЧМП  $L_j$ :

- степени всех вершин  $v_j \in L_j$  — четны в  $T_j$ ;
- каждая вершина в  $T_j$ , за возможным исключением  $a_j$  и  $b_j$ , имеет четную, либо нулевую степень;
- по исключению вершин с нулевой степенью,  $T_j$  либо не имеет связных компонент, либо имеет одну связную компоненту с вершиной  $a_j$  или  $b_j$  в ней, либо — две связные компоненты с вершиной  $a_j$  в одной, и  $b_j$  в другой.

Два ЧМП  $L_j$   $T_j^1$  и  $T_j^2$  называются эквивалентными, если для любого подграфа  $C_j \subset G - L_j$  такого, что  $T_j^1 \cup C_j$  является маршрутным подграфом,  $T_j^2 \cup C_j$  также является маршрутным подграфом (т.е., любое дополнение одного ЧМП является дополнением другого ЧМП).

Таким образом, два ЧМП  $L_j$   $T_j^1$  и  $T_j^2$  эквивалентны, если:

- $a_j$  имеет тождественную степень (т.е., четную, нечетную, либо нулевую) в обоих подграфах и  $b_j$  имеет тождественную степень в обоих подграфах;
- по исключению вершин с нулевой степенью, и  $T_j^1$ , и  $T_j^2$  — либо оба не имеют связных компонент, либо оба имеют одну связную компоненту с вершиной  $a_j$  или  $b_j$  в ней, либо — две связные компоненты с вершиной  $a_j$  в одной и  $b_j$  в другой.

Определим эквивалентные классы каждого ЧМП  $L_j$  как совокупность трех параметров — четности степени  $a_j$ , четности степени  $b_j$  и числа связных компонент. Например, ЧМП  $L_2$  (рис. 4) являются эквивалентные классы (0, Ч, 1), (Ч, Ч, 2), (Н, Н, 1), где 0 обозначает нулевую степень, Ч — четную, Н — нечетную, а третья цифра — количество компонент связности.

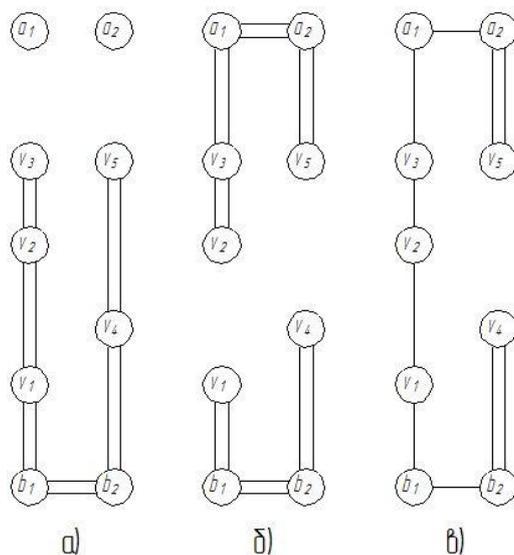


Рис. 4. Примеры  $L_2$  ЧМП для графа, представленного на рис. 2

Существует семь эквивалентных классов ЧМП  $L_j$ : (Н, Н, 1), (0, Ч, 2), (Ч, 0, 2), (Ч, Ч, 1), (Ч, Ч, 2), (0, 0, 0), (0, 0, 1). Класс (0, 0, 0) возможен только в случае, если ни один проход в  $L_j$  не содержит сборочных ячеек, а класс (0, 0, 1) — если ни один проход в  $G - L_j$  не содержит сборочных ячеек (т.е., одна связная компонента находится по левую сторону от  $a_j$  или  $b_j$ , и не содержит ни  $a_j$ , ни  $b_j$ ). В случае, когда все проходы (поперечные и продольные), имеют одинаковую длину, проходы не содержащие сборочных ячеек, а

следовательно и классы  $(0, 0, 0)$  и  $(0, 0, 1)$ , могут быть удалены и не использоваться в расчетах.

Если  $T_j^i$  — ЧМП  $L_j$  минимальной длины  $i$ -го класса, а  $C_j^i$  — дополнение минимальной длины для  $T_j^i$ , то маршрутным подграфом  $G$  минимальной длины является частичный подграф  $T_j^i \cup C_j^i$  минимальной длины для  $i \in \{1, 2, \dots, 7\}$ .

Заметим, что для  $L_n^+$  дополнением минимальной длины для ЧМП классов  $(Ч, 0, 1)$ ,  $(0, Ч, 1)$ ,  $(Ч, Ч, 1)$  и  $(0, 0, 1)$  является нулевой граф (т.е., ЧМП уже является маршрутным подграфом минимальной длины). Также, т.к.  $G - L_j^+ = 0$ , то для оставшихся классов дополнения не существует. Таким образом, при нахождении ЧМП минимальной длины классов  $(Ч, 0, 1)$ ,  $(0, Ч, 1)$ ,  $(Ч, Ч, 1)$  и  $(0, 0, 1)$  в  $L_n^+$ , минимальный из подграфов этих классов будет минимальным маршрутом сборщика заказов.

### 2.3. Построение ЧМП минимальной длины

Поиск маршрутного подграфа минимальной длины ведется для каждого прохода  $j = 1, 2, \dots, n$  по очереди. С использованием ЧМП минимальной длины  $L_j^-$  каждого эквивалентного класса, строятся ЧМП минимальной длины  $L_j^+$ . С использованием ЧМП минимальной длины  $L_j^+$  каждого эквивалентного класса, строятся ЧМП минимальной длины  $L_{j+1}^-$ . Наконец, после определения ЧМП минимальной длины  $L_n^+$  для каждого из эквивалентных классов, циклический подграф минимальной длины является ЧМП минимальной длины среди  $(Ч, 0, 1)$ ,  $(0, Ч, 1)$ ,  $(Ч, Ч, 1)$  и  $(0, 0, 1)$  из  $L_n^+$ .

Рассмотрим некий  $j$ -ый проход. Поскольку ЧМП минимальной длины не может содержать более двух дуг между двумя любыми парами вершин, то для данного прохода возможны лишь шесть случаев соединения вершин дугами, которые представлены на рис. 5. В случае двух заходов в один проход с противоположных сторон (рис. 5г), необходимо рассматривать наилучший возможный случай (с наименьшей длиной). Простейшим способом отыскания такого случая, является отыскание наиболее длинного участка между смежными вершинами прохода, и добавление промежутка на место связывающей их дуги.

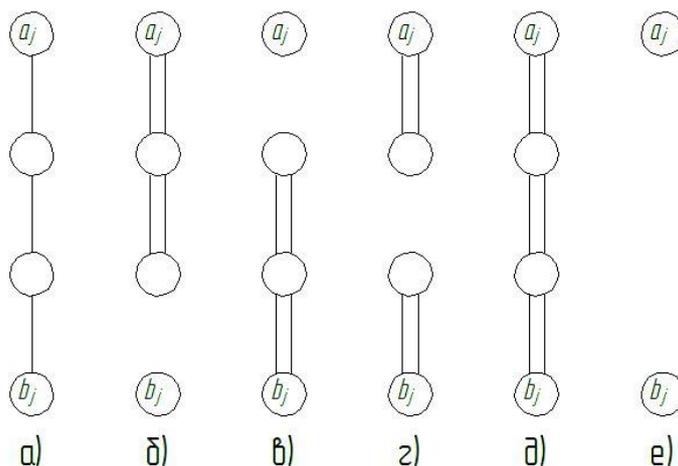


Рис. 5. Возможные конфигурации дуг для  $j$ -го прохода в оптимальном ЧМП

В табл. 1 приведены эквивалентные классы  $L_j^+$ , получаемые добавлением конфигураций минимальной длины среди представленных на рис. 5 к ЧМП  $L_j^-$  каждого из эквивалентных классов.

При наличии ЧМП  $L_j^-$  минимальной длины для каждого эквивалентного класса, возможно получить ЧМП  $L_j^+$  минимальной длины для каждого эквивалентного класса путем выбора из табл. 1 конфигурации, обеспечивающей минимальную длину ЧМП  $L_j^+$  для каждого из эквивалентных классов.

Для определения ЧМП  $L_{j+1}^-$  минимальной длины для каждого эквивалентного класса, необходимо для заданных ЧМП  $L_j^+$  для каждого эквивалентного класса добавить одну из конфигураций дуг, представленных на рис. 6 для обеспечения перехода между  $j$ -ым и  $(j + 1)$ -ым проходами.

**Табл. 1.** Эквивалентные классы  $L_j^+$ , добавлением конфигурации минимальной длины среди представленных на рис. 5 к ЧМП  $L_j^-$  каждого из эквивалентных классов

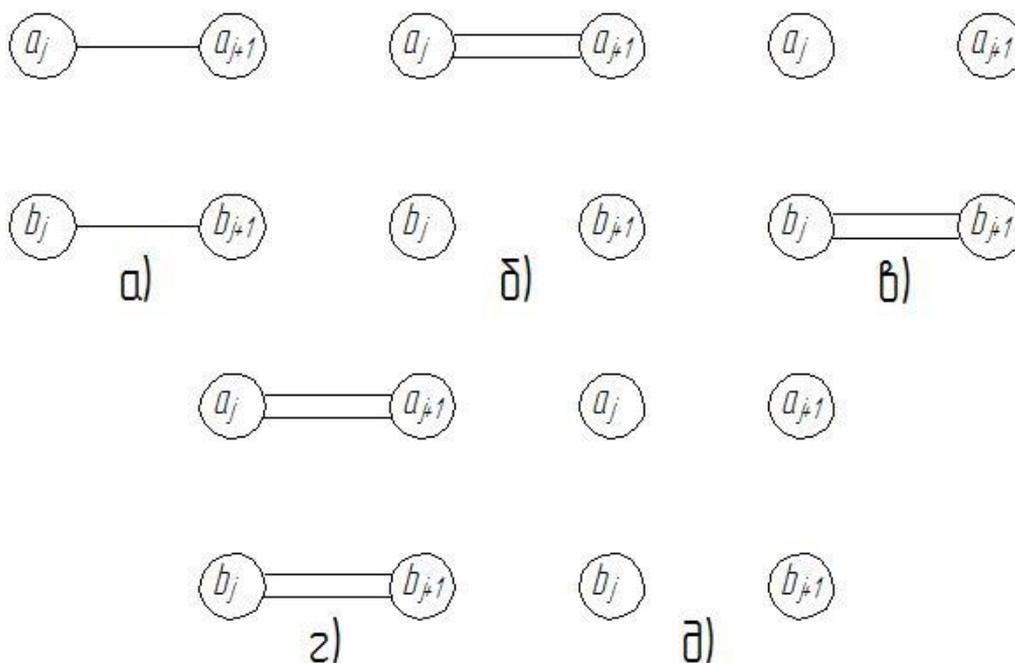
$L_j^-$	Конфигурация дуг с рис. 5					
	а	б	в	г	д	е <sup>1</sup>
(Н, Н, 1)	(Ч, Ч, 1)	(Н, Н, 1)	(Н, Н, 1)	(Н, Н, 1)	(Н, Н, 1)	(Н, Н, 1)
(Ч, 0, 1)	(Н, Н, 1)	(Ч, 0, 1)	(Ч, Ч, 2)	(Ч, Ч, 2)	(Ч, Ч, 1)	(Ч, 0, 1)
(0, Ч, 1)	(Н, Н, 1)	(Ч, Ч, 2)	(Ч, 0, 1)	(Ч, Ч, 2)	(Ч, Ч, 1)	(0, Ч, 1)
(Ч, Ч, 1)	(Н, Н, 1)	(Ч, Ч, 1)	(Ч, Ч, 1)	(Ч, Ч, 1)	(Ч, Ч, 1)	(Ч, Ч, 1)
(Ч, Ч, 2)	(Н, Н, 1)	(Ч, Ч, 2)	(Ч, Ч, 2)	(Ч, Ч, 2)	(Ч, Ч, 1)	(Ч, Ч, 2)
(0, 0, 0) <sup>2</sup>	(Н, Н, 1)	(Ч, 0, 1)	(Ч, 0, 1)	(Ч, Ч, 2)	(Ч, Ч, 1)	(0, 0, 0)
(0, 0, 1) <sup>3</sup>	— <sup>4</sup>	—	—	—	—	(0, 0, 1)

<sup>1</sup>нецелесообразная конфигурация при наличии в проходе сборочных ячеек

<sup>2</sup>только если слева от  $j$ -го прохода нет сборочных ячеек

<sup>3</sup>только если справа от  $j$ -го прохода нет сборочных ячеек

<sup>4</sup>не может быть оптимальным



**Рис. 6.** Возможные конфигурации дуг при переходе от  $j$ -го к  $(j+1)$ -му проходу в оптимальном ЧМП

Табл. 2 описывает эквивалентные классы  $L_{j+1}^-$ , получаемые добавлением конфигураций минимальной длины на рис. 6 к ЧМП  $L_j^+$  минимальной длины для каждого эквивалентного класса. Прочерки соответствуют конфигурациям однозначно не ведущим к оптимальному маршруту.

Таким образом, при наличии ЧМП  $L_j^+$  минимальной длины для каждого эквивалентного класса, возможно построить ЧМП  $L_{j+1}^-$  минимальной длины путем выбора из табл. 2 конфигурации, обеспечивающей минимальную длину ЧМП  $L_{j+1}^-$  для каждого эквивалентного класса.

Таким образом, путем последовательного вычисления ЧМП для каждого из проходов, возможно получить оптимальный маршрут для сборки всего заказа.

**Табл. 2.** Эквивалентные классы, получаемые путем добавления каждой из представленных на рис. 6 конфигураций дуг к ЧМП  $L_j^+$  каждого из эквивалентных классов

$L_j^-$	Конфигурация дуг с рис. 6				
	а	б	в	г	д
(Н, Н, 1)	(Н, Н, 1)	— <sup>1</sup>	— <sup>1</sup>	— <sup>1</sup>	— <sup>1</sup>
(Ч, 0, 1)	— <sup>1</sup>	(Ч, 0, 1)	— <sup>2</sup>	(Ч, Ч, 2)	(0, 0, 1)
(0, Ч, 1)	— <sup>1</sup>	— <sup>2</sup>	(0, Ч, 1)	(Ч, Ч, 2)	(0, 0, 1)
(Ч, Ч, 1)	— <sup>1</sup>	(Ч, 0, 1)	(0, Ч, 1)	(Ч, Ч, 1)	(0, 0, 1)
(Ч, Ч, 2)	— <sup>1</sup>	— <sup>2</sup>	— <sup>2</sup>	(Ч, Ч, 2)	— <sup>2</sup>
(0, 0, 0)	— <sup>1</sup>	— <sup>3</sup>	— <sup>3</sup>	— <sup>3</sup>	(0, 0, 0)
(0, 0, 1)	— <sup>2</sup>	— <sup>2</sup>	— <sup>2</sup>	— <sup>2</sup>	(0, 0, 1)

<sup>1</sup>степени  $a_j$  и  $b_j$  — нечетные

<sup>2</sup>ни одно дополнение не может сделать граф связным

<sup>3</sup>никогда не будет оптимальным

## 2.4. Числовой пример

В качестве примера воспользуемся типовым заказом (рис. 1) на складе, представленном на рис. 2. Расстояния между смежными вершинами представлены на рис. 7. Табл. 3 содержит информацию по определению ЧМП минимальной длины для каждого эквивалентного класса. Три числа в каждой ячейке таблицы обозначают длину ЧМП минимальной длины для эквивалентного класса, номер предшествующего эквивалентного класса, на основе которого был построен текущий частичный подграф минимальной длины, и конфигурацию дуг (из рис. 5 для  $L_j^+$  и рис. 6 для  $L_j^-$ ), которые были добавлены для получения текущего частичного подграфа.

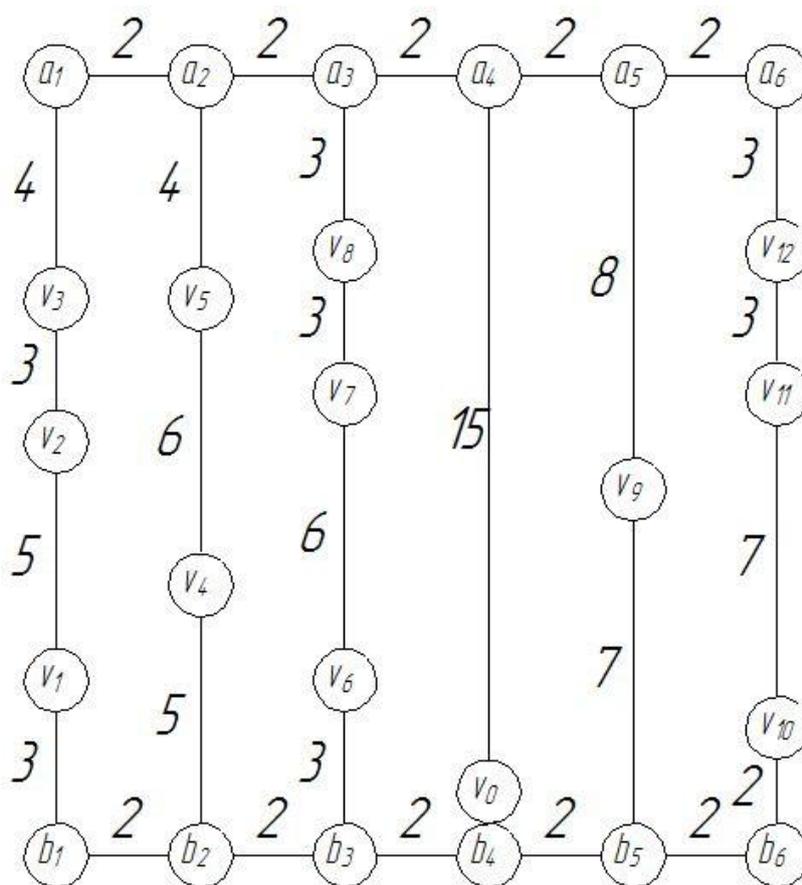


Рис. 7. Пример графа-представления типового заказа

Табл. 3. Решение примера из рис. 8

№	Эквивалентный класс	Проход 1		Проход 2		Проход 3	
		$L_1^-$	$L_1^+$	$L_2^-$	$L_2^+$	$L_3^-$	$L_3^+$
1	(Н, Н, 1)	—	15, -, а	19, 1, а	37, 1, г	41, 1, а	53, 2, а
2	(Ч, 0, 1)	—	24, -, б	28, 2, б	48, 2, б	38, 4, б	62, 2, б
3	(0, Ч, 1)	—	22, -, в	26, 3, в	48, 3, в	38, 4, в	62, 3, в
4	(Ч, Ч, 1)	—	30, -, д	38, 4, г	34, 1, а	42, 4, г	56, 1, а
5	(Ч, Ч, 2)	—	20, -, г	28, 5, г	44, 3, г	52, 5, г	56, 2, г
6	(0, 0, 0)	—	—	—	—	—	—
7	(0, 0, 1)	—	—	22, 3, д	—	34, 4, д	—
	Эквивалентный класс	Проход 4		Проход 5		Проход 6	
		$L_4^-$	$L_4^+$	$L_5^-$	$L_5^+$	$L_6^-$	$L_6^+$
1	(Н, Н, 1)	57, 1, а	57, 1, в	61, 1, а	75, 1, в	79, 1, а	95, 1, г
2	(Ч, 0, 1)	60, 4, б	90, 2, б	68, 4, б	84, 2, б	80, 4, б	106, 2, б
3	(0, Ч, 1)	60, 4, в	60, 3, в	64, 3, в	78, 3, в	80, 4, в	104, 3, в
4	(Ч, Ч, 1)	64, 4, г	64, 4, в	72, 4, г	76, 1, а	84, 4, г	94, 1, а
5	(Ч, Ч, 2)	64, 5, г	60, 2, в	68, 3, г	80, 3, б	86, 3, г	96, 2, г
6	(0, 0, 0)	—	—	—	—	—	—
7	(0, 0, 1)	56, 4, д	—	60, 3, д	—	76, 4, д	—

Не существует ЧМП  $L_1^-$ . ЧМП  $L_1^+$  минимальной длины получены добавлением конфигураций дуг из рис. 5 к нулевому графу. Прочерки у ЧМП  $L_1^+$  означают, что у ЧМП не было предшественника.

Для получения столбца  $L_2^-$  используется табл. 2. Например, из табл. 2 видно, что существует два способа получения ЧМП  $L_2^-$  для эквивалентного класса (Ч, 0, 1). Возможно добавление конфигурации б) из рис. 6 к ЧМП  $L_1^+$  эквивалентных классов (Ч, 0, 1) и (Ч, Ч, 1). Из табл. 3 видно, что ЧМП класса (Ч, 0, 1) имеет длину 24, а ЧМП класса (Ч, Ч, 1) — 30. Таким образом, ЧМП  $L_2^-$  минимальной длины для эквивалентного класса (Ч, 0, 1) получается путем добавления конфигурации дуг б) из рис. 6 к ЧМП  $L_1^+$  минимальной длины класса (Ч, 0, 1). Длина результирующего ЧМП равна 28. Таким образом, происходит последовательная обработка проходов с поиском ЧМП минимальных длин.

Оптимальный маршрут сборщика заказов находится как наикратчайший из ЧМП  $L_6^+$  классов (Ч, 0, 1), (0, Ч, 1), (Ч, Ч, 1), (0, 0, 1). Из столбца  $L_6^+$  табл. 3 видно, что оптимальный маршрут сборщика заказов — класса (Ч, Ч, 1) и имеет длину 94. Последовательность конфигураций дуг может быть прослежена путем из подчеркнутых элементов табл. 3. Построение маршрута возможно путем добавления конфигураций дуг из рис. 5–6 к подчеркнутым элементам табл. 3 для каждого прохода. На рис. 8 представлен итоговый маршрут минимальной длины.

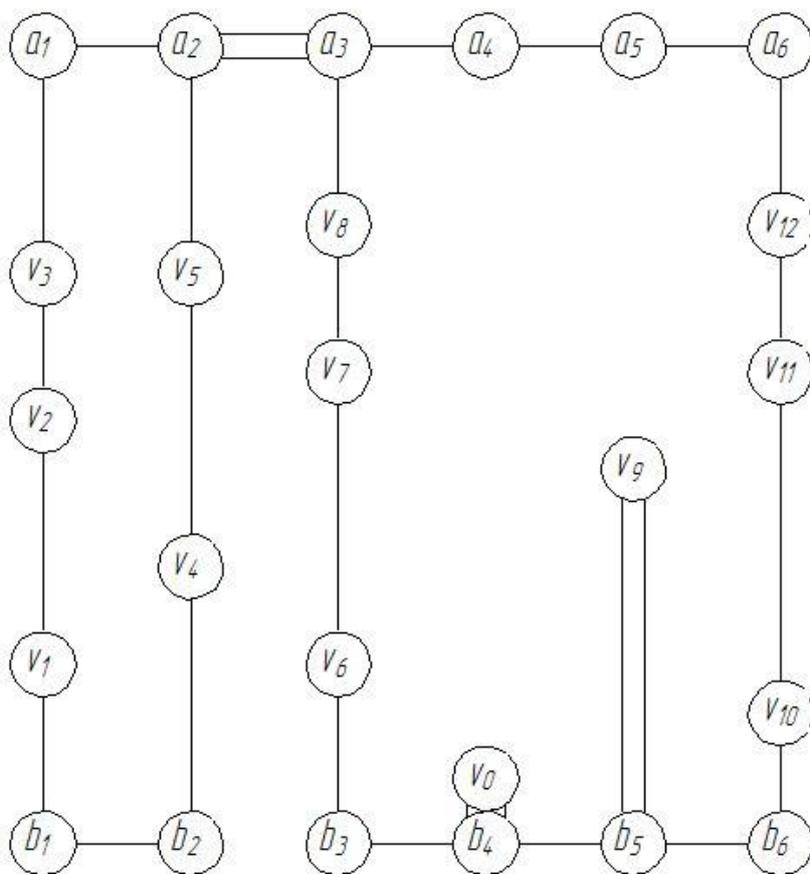


Рис. 8. Маршрут минимальной длины для примера, представленного на рис. 7

## 2.5. Процедура получения последовательности обхода по оптимальному маршруту минимальной длины

После описанной процедуры получения маршрута минимальной длины  $T$ , последовательность обхода ячеек сборщиком заказов может быть получена следующим образом.

- 1) Начинаем обход из вершины  $v_0$ .
- 2) Допустим,  $v^*$  — только что посещенная вершина.
- 3) Если существует пара неиспользованных параллельных дуг в  $T$ , инцидентных с  $v^*$ , используем одну из них, чтобы добраться до следующей вершины. Возвращаемся на шаг 2.
- 4) Если существуют неиспользованные одиночные дуги в  $T$ , используем одну из них, чтобы добраться до следующей вершины. Возвращаемся на шаг 2.
- 5) Если существует пара параллельных дуг в  $T$ , и одна из дуг пары все еще не была использована, используем ее, чтобы добраться до следующей вершины. Возвращаемся на шаг 2.
- 6) Завершение. Последовательность обхода сборщиком заказов сборочных ячеек составлена.

## 3. Расширение алгоритма Рэтлиффа-Розенталя для решения задачи маршрутизации сборщиков заказов во многоблочных складах

Рассмотренный выше алгоритм Рэтлиффа-Розенталя для получения оптимального маршрута сборщика заказов может быть расширен на многоблочные (*multi-block warehouses*) склады. Ввиду экспоненциального роста числа эквивалентных классов ЧМП, в рамках данной статьи будет рассмотрено только расширение на двухблочные склады, т.е. склады с тремя поперечными проходами.

### 3.1. Представление планировки склада и типового заказа в виде графа

Склад состоит из определенного числа параллельных проходов. Артикулы хранятся по обе стороны от прохода. Сборщики заказов могут двигаться вдоль проходов в оба направления и изменять направление движения внутри прохода. Каждый заказ состоит из артикулов, хранящихся в ячейках, распределенных по складу. Подразумевается, что артикулы одного заказа могут быть собраны за один обход. Смена прохода возможна в начале, конце и середине прохода. Собранные товары должны быть доставлены на базу, где сборщик заказов затем получает следующий сборочный лист на заказ.

В качестве примера рассмотрим склад, состоящий из  $n$  сборочных и трех поперечных проходов. Таким образом, можно заключить, что склад состоит из двух блоков по  $n$  проходов в каждом. Конфигурация склада представлена на рис. 9. Типовой заказ состоит из  $m$  строк (артикулов). Среди них,  $m_x$  расположены во блоке X, а  $m_y$  — во блоке Y ( $m_x + m_y = m$ ).

Склад со сборочными ячейками заказа может быть описан в виде графа  $G$ , состоящего из следующих вершин:

$v_{x,i}$  ( $i = 1, \dots, m_x$ ) — вершины, соответствующие сборочным ячейкам блока  $X$ ;

$v_{y,i}$  ( $i = 1, \dots, m_y$ ) — вершины, соответствующие сборочным ячейкам блока  $Y$ ;

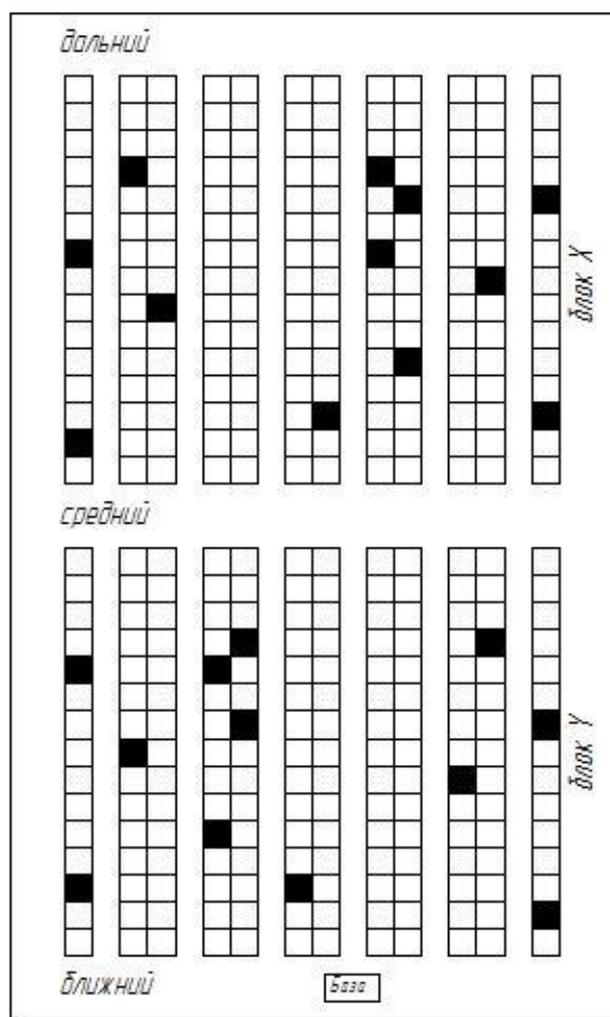
$a_i$  ( $i = 1, \dots, N$ ) — вершины, соответствующие дальнему (от базы) концу  $i$ -го прохода;

$b_i$  ( $i = 1, \dots, N$ ) — вершины, соответствующие середине  $i$ -го прохода;

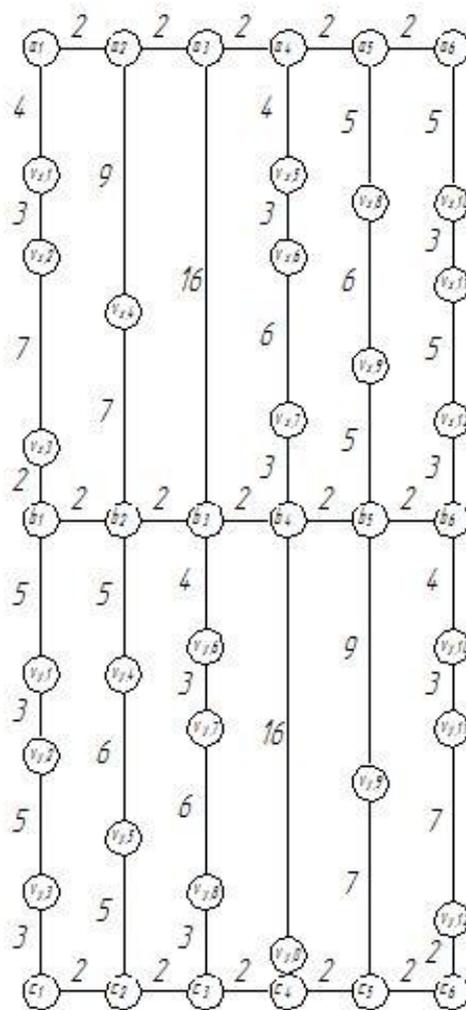
$c_i$  ( $i = 1, \dots, N$ ) — вершины, соответствующие ближнему (к базе) концу  $i$ -го прохода;

$y_{y,0}$  — вершина, соответствующая базе.

Любые две вершины, соответствующие смежным ячейкам на складе соединены дугами, длина которых определяется временем передвижения между соответствующими ячейками на складе.



а)



б)

**Рис. 9.** а) Планировка склада с типовым заказом;  
б) представление планировки склада с типовым заказом в виде графа

### 3.2. Построение ЧМП минимальной длины

Положим, что  $L_j^-$  является подграфом графа склада, включающим в себя вершины  $a_j, b_j, c_j$ , а также все ребра и вершины по левую сторону от  $a_j, b_j, c_j$ . Положим, что  $Y_j$  является подграфом графа склада, включающим в себя вершины  $b_j, c_j$ , а также все ребра и вершины между  $b_j$  и  $c_j$ , и определим  $L_j^{+y} = L_j^- \cup Y_j$ . Аналогично, положим, что  $X_j$  является подграфом графа склада, состоящим из вершин  $a_j, b_j$ , а также всех ребер и вершин между ними, и определим  $L_j^{+x} = L_j^- \cup X_j$ . Обозначение  $L_j$  используется, при сохранении результата для  $L_j = L_j^-, L_j = L_j^{+y}, L_j = L_j^{+x}$ .

Алгоритм построения ЧМП минимальной длины основан на динамическом программировании. Поиск начинается для всех ЧМП  $L_1^{+y}$ , состоящих только из ребер и вершин между  $b_1$  и  $c_1$ . На следующем шаге, ЧМП  $L_1^{+y}$  расширяется путем добавления вершин и ребер между  $a_1$  и  $b_1$  для получения ЧМП  $L_1^{+x}$ . Затем, определяются ЧМП  $L_2^-$  путем расширения ЧМП  $L_1^{+x}$  ребрами между проходами 1 и 2. Из ЧМП  $L_2^-$  могут быть найдены ЧМП  $L_2^{+y}$ . Таким образом, в завершении определяются ЧМП  $L_n^{+x}$ , которые являются искомыми маршрутными подграфами, среди которых и выбирается оптимальный. Для описания алгоритма введем несколько понятий.

**Параметры эквивалентных классов ЧМП.** Эквивалентные классы ЧМП  $L_j$  могут быть охарактеризованы пятью параметрами: четностью степени  $a_j$ , четностью степени  $b_j$ , четностью степени  $c_j$ , числом компонент связности, а также распределением  $a_j, b_j, c_j$  по различным компонентам.

Четность степени описывает число инцидентных данной вершине ребер. Оно может быть четным (Ч), нечетным (Н), либо нулевым (0). Число компонент связности может варьироваться от нуля до трех. Распределение  $a_j, b_j, c_j$  по различным компонентам показывает, какие из вершин  $a_j, b_j, c_j$  содержатся в одном и том же связном компоненте. Для простоты нотации, опустим параметр распределения  $a_j, b_j, c_j$  по компонентам связности, за исключением случая, когда две компоненты связности имеют четную степень. Таким образом, пятый параметр может принимать значения:  $a-bc, b-ac, c-ab$ . Например, нотация  $a-bc$  подразумевает, что вершина  $a_j$  находится в одной компоненте связности, а вершины  $b_j, c_j$  — в другой.

Доказывается, что существует 25 следующих эквивалентных классов [4]:

(0, 0, 0, 0), (0, 0, 0, 1), (Ч, Ч, Ч, 1), (Ч, Ч, Ч, 3),  
(Ч, 0, 0, 1), (0, Ч, 0, 1), (0, 0, Ч, 1), (Ч, Ч, 0, 1), (Ч, 0, Ч, 1), (0, Ч, Ч, 1),  
(Н, Н, 0, 1), (Н, 0, Н, 1), (0, Н, Н, 1), (Ч, Н, Н, 1), (Н, Ч, Н, 1), (Н, Н, Ч, 1),  
(Ч, Ч, 0, 2), (Ч, 0, Ч, 2), (0, Ч, Ч, 2), (Ч, Н, Н, 2), (Н, Ч, Н, 2), (Н, Н, Ч, 2),

(Ч, Ч, Ч, 2,  $a-bc$ ), (Ч, Ч, Ч, 2,  $b-ac$ ), (Ч, Ч, Ч, 2,  $c-ab$ ).

Заметим, что класс  $(0, 0, 0, 0)$  возможен, только если ни один из проходов в  $L_j$  не содержит сборочных ячеек, а класс  $(0, 0, 0, 1)$  — только если ни один из проходов в  $(G - L_j)$  не содержит сборочных ячеек.

Согласно [3–4], после вычисления ЧМП  $L_n^{+x}$ , маршрутным подграфом минимальной длины маршрута сборщика заказов будет являться кратчайший среди следующих ЧМП:

$(0, 0, 0, 1)$ ,  $(Ч, 0, 0, 1)$ ,  $(0, Ч, 0, 1)$ ,  $(0, 0, Ч, 1)$ ,  
 $(Ч, Ч, 0, 1)$ ,  $(Ч, 0, Ч, 1)$ ,  $(0, Ч, Ч, 1)$ ,  $(Ч, Ч, Ч, 1)$ .

**Перемещения.** Перемещения (*transitions*) между вершинами при построении маршрутного подграфа подразумевают добавление вершин и ребер. Различают три следующих вида перемещений. В первом случае, при перемещении от  $L_j^-$  к  $L_j^{+y}$ , добавляются вершины и ребра между вершинами  $b_j$ ,  $c_j$ . Во втором случае, при перемещении от  $L_j^{+y}$  к  $L_j^{+x}$ , добавляются вершины и ребра между вершинами  $a_j$ ,  $b_j$ . Наконец, в третьем случае, при перемещении от  $L_j^{+x}$  к  $L_{j+1}^-$ , делается связь между  $j$ -ым и  $(j + 1)$ -ым проходами.

Перемещение от  $L_j^-$  к  $L_j^{+y}$ . Рассмотрим некоторый  $j$ -ый проход. Поскольку, между парой любых вершин не может быть более двух ребер (п. 2.2), ребра между  $b_j$  и  $c_j$  могут быть одного из возможных видов, изображенных на рис. 10. Эквивалентные классы, получаемые при данных перемещениях, представлены в табл. 4.

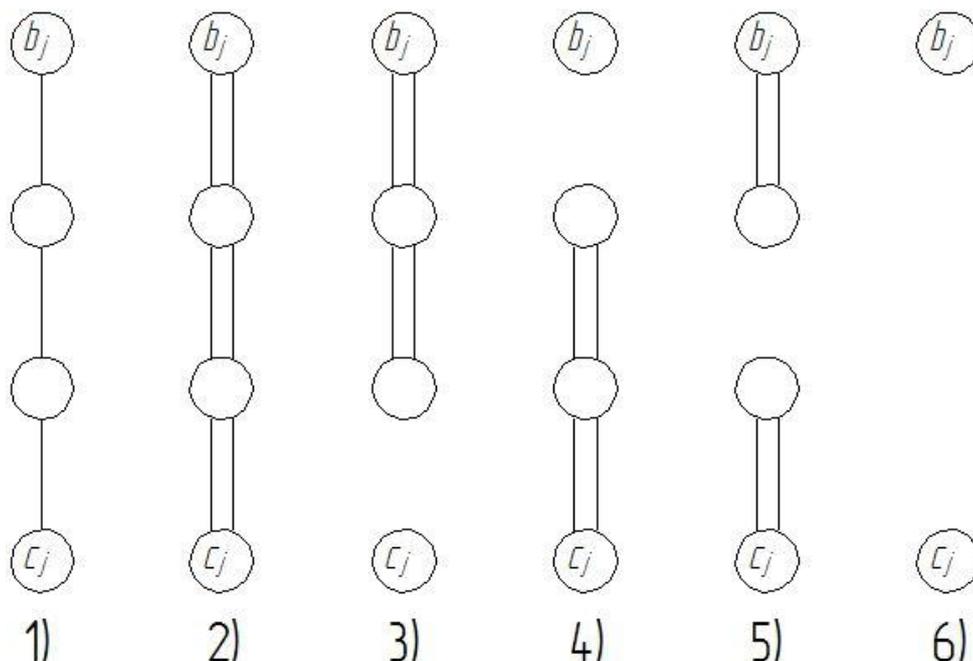


Рис. 10. Возможные способы перемещения внутри  $j$ -го прохода между  $b_j$  и  $c_j$

**Табл. 4.** Эквивалентные классы, получаемые при добавлении перемещений из рис. 10  
ко всем возможным конфигурациям  $L_j^-$

$L_j^- \rightarrow L_j^{+y}$	1	2	3	4	5	6 <sup>1</sup>
$(0, 0, 0, 0)^2$	(0, Н, Н, 1)	(0, Ч, Ч, 1)	(0, Ч, 0, 1)	(0, 0, Ч, 1)	(0, Ч, Ч, 2)	(0, 0, 0, 0)
$(0, 0, 0, 1)^3$	— <sup>4</sup>	(0, 0, 0, 1)				
(Ч, 0, 0, 1)	(Ч, Н, Н, 2)	(Ч, Ч, Ч, 2, <i>a-bc</i> )	(Ч, Ч, 0, 2)	(Ч, 0, Ч, 2)	(Ч, Ч, Ч, 3)	(Ч, 0, 0, 1)
(0, Ч, 0, 1)	(0, Н, Н, 1)	(0, Ч, Ч, 1)	(0, Ч, 0, 1)	(0, Ч, Ч, 2)	(0, Ч, Ч, 2)	(0, Ч, 0, 1)
(0, 0, Ч, 1)	(0, Н, Н, 1)	(0, Ч, Ч, 1)	(0, Ч, Ч, 2)	(0, 0, Ч, 1)	(0, Ч, Ч, 2)	(0, 0, Ч, 1)
(Ч, Ч, 0, 1)	(Ч, Н, Н, 1)	(Ч, Ч, Ч, 1)	(Ч, Ч, 0, 1)	(Ч, Ч, Ч, 2, <i>c-ab</i> )	(Ч, Ч, Ч, 2, <i>c-ab</i> )	(Ч, Ч, 0, 1)
(Ч, 0, Ч, 1)	(Ч, Н, Н, 1)	(Ч, Ч, Ч, 1)	(Ч, Ч, Ч, 2, <i>b-ac</i> )	(Ч, 0, Ч, 1)	(Ч, Ч, Ч, 2, <i>b-ac</i> )	(Ч, 0, Ч, 1)
(0, Ч, Ч, 1)	(0, Н, Н, 1)	— <sup>5</sup>	(0, Ч, Ч, 1)			
(Ч, Ч, Ч, 1)	(Ч, Н, Н, 1)	— <sup>5</sup>	(Ч, Ч, Ч, 1)			
(Н, Н, 0, 1)	(Н, Ч, Н, 1)	(Н, Н, Ч, 1)	(Н, Н, 0, 1)	(Н, Н, Ч, 2)	(Н, Н, Ч, 2)	(Н, Н, 0, 1)
(Н, 0, Н, 1)	(Н, Н, Ч, 1)	(Н, Ч, Н, 1)	(Н, Ч, Н, 2)	(Н, 0, Н, 1)	(Н, Ч, Н, 2)	(Н, 0, Н, 1)
(0, Н, Н, 1)	(0, Ч, Ч, 1)	— <sup>5</sup>	(0, Н, Н, 1)			
(Ч, Н, Н, 1)	(Ч, Ч, Ч, 1)	— <sup>5</sup>	(Ч, Н, Н, 1)			
(Н, Ч, Н, 1)	(Н, Н, Ч, 1)	— <sup>5</sup>	(Н, Ч, Н, 1)			
(Н, Н, Ч, 1)	(Н, Ч, Н, 1)	— <sup>5</sup>	(Н, Н, Ч, 1)			
(Ч, Ч, 0, 2)	(Ч, Н, Н, 2)	(Ч, Ч, Ч, 2, <i>a-bc</i> )	(Ч, Ч, 0, 2)	(Ч, Ч, Ч, 3)	(Ч, Ч, Ч, 3)	(Ч, Ч, 0, 2)
(Ч, 0, Ч, 2)	(Ч, Н, Н, 2)	(Ч, Ч, Ч, 2, <i>a-bc</i> )	(Ч, Ч, Ч, 3)	(Ч, 0, Ч, 2)	(Ч, Ч, Ч, 3)	(Ч, 0, Ч, 2)
(0, Ч, Ч, 2)	(0, Н, Н, 1)	(0, Ч, Ч, 1)	(0, Ч, Ч, 2)			
(Ч, Ч, Ч, 2, <i>a-bc</i> )	(Ч, Н, Н, 2)	— <sup>5</sup>	(Ч, Ч, Ч, 2, <i>a-bc</i> )			
(Ч, Ч, Ч, 2, <i>b-ac</i> )	(Ч, Н, Н, 1)	(Ч, Ч, Ч, 1)	(Ч, Ч, Ч, 2, <i>b-ac</i> )			
(Ч, Ч, Ч, 2, <i>c-ab</i> )	(Ч, Н, Н, 1)	(Ч, Ч, Ч, 1)	(Ч, Ч, Ч, 2, <i>c-ab</i> )			
(Ч, Н, Н, 2)	(Ч, Ч, Ч, 2, <i>a-bc</i> )	— <sup>5</sup>	(Ч, Н, Н, 2)			
(Н, Ч, Н, 2)	(Н, Н, Ч, 1)	(Н, Ч, Н, 1)	(Н, Ч, Н, 2)			
(Н, Н, Ч, 2)	(Н, Ч, Н, 1)	(Н, Н, Ч, 1)	(Н, Н, Ч, 2)			
(Ч, Ч, Ч, 3)	(Ч, Н, Н, 2)	(Ч, Ч, Ч, 2, <i>a-bc</i> )	(Ч, Ч, Ч, 3)			

<sup>1</sup> возможно только при отсутствии в проходе сборочных ячеек

<sup>2</sup> может быть использован только, если в  $L_j^-$  нет сборочных ячеек

<sup>3</sup> может быть использован только, если в  $G - L_j^-$  нет сборочных ячеек

<sup>4</sup> нарушает введенные ограничения

<sup>5</sup> никогда не приведет к оптимальному решению

Перемещение от  $L_j^{+y}$  к  $L_j^{+x}$ . Данное перемещение похоже на перемещение от  $L_j^-$  к  $L_j^{+y}$ .  
Различают также шесть способов преодоления ребер в  $j$ -ом проходе между вершинами  $a_j$  и  $b_j$ . Эквивалентные классы, получаемые при данном перемещении, приведены в табл. 5.

**Табл. 5.** Эквивалентные классы, получаемые при добавлении перемещений из рис. 10  
ко всем возможным конфигурациям  $L_j^{+y}$

$L_j^{+y} \rightarrow L_j^{+x}$	1	2	3	4	5	6 <sup>1</sup>
$(0, 0, 0, 0)^2$	(Н, Н, 0, 1)	(Ч, Ч, 0, 1)	(Ч, 0, 0, 1)	(0, 0, Ч, 1)	(0, Ч, Ч, 2)	(0, 0, 0, 0)
$(0, 0, 0, 1)^3$	— <sup>4</sup>	(0, 0, 0, 1)				
(Ч, 0, 0, 1)	(Н, Н, 0, 1)	(Ч, Ч, 0, 1)	(Ч, 0, 0, 1)	(Ч, Ч, 0, 2)	(Ч, Ч, 0, 2)	(Ч, 0, 0, 1)
(0, Ч, 0, 1)	(Н, Н, 0, 1)	(Ч, Ч, 0, 1)	(Ч, Ч, 0, 2)	(0, Ч, 0, 1)	(Ч, Ч, 0, 2)	(0, Ч, 0, 1)
(0, 0, Ч, 1)	(Н, Н, Ч, 2)	(Ч, Ч, Ч, 2, <i>c-ab</i> )	(Ч, 0, Ч, 2)	(0, Ч, Ч, 2)	(Ч, Ч, Ч, 3)	(0, 0, Ч, 1)
(Ч, Ч, 0, 1)	(Н, Н, 0, 1)	— <sup>5</sup>	(Ч, Ч, 0, 1)			
(Ч, 0, Ч, 1)	(Н, Н, Ч, 1)	(Ч, Ч, Ч, 1)	(Ч, 0, Ч, 1)	(Ч, Ч, Ч, 2, <i>b-ac</i> )	(Ч, Ч, Ч, 2, <i>b-ac</i> )	(Ч, 0, Ч, 1)
(0, Ч, Ч, 1)	(Н, Н, Ч, 1)	(Ч, Ч, Ч, 1)	(Ч, Ч, Ч, 2, <i>a-bc</i> )	(0, Ч, Ч, 1)	(Ч, Ч, Ч, 2, <i>a-bc</i> )	(0, Ч, Ч, 1)
(Ч, Ч, Ч, 1)	(Н, Н, Ч, 1)	— <sup>5</sup>	(Ч, Ч, Ч, 1)			
(Н, Н, 0, 1)	(Ч, Ч, 0, 1)	— <sup>5</sup>	(Н, Н, 0, 1)			
(Н, 0, Н, 1)	(Ч, Н, Н, 1)	(Н, Ч, Н, 1)	(Н, 0, Н, 1)	(Н, Ч, Н, 2)	(Н, Ч, Н, 2)	(Н, 0, Н, 1)
(0, Н, Н, 1)	(Н, Ч, Н, 1)	(Ч, Н, Н, 1)	(Ч, Н, Н, 2)	(0, Н, Н, 1)	(Ч, Н, Н, 2)	(0, Н, Н, 1)
(Ч, Н, Н, 1)	(Н, Ч, Н, 1)	— <sup>5</sup>	(Ч, Н, Н, 1)			
(Н, Ч, Н, 1)	(Ч, Н, Н, 1)	— <sup>5</sup>	(Н, Ч, Н, 1)			
(Н, Н, Ч, 1)	(Ч, Ч, Ч, 1)	— <sup>5</sup>	(Н, Н, Ч, 1)			
(Ч, Ч, 0, 2)	(Н, Н, 0, 1)	(Ч, Ч, 0, 1)	(Ч, Ч, 0, 2)			
(Ч, 0, Ч, 2)	(Н, Н, Ч, 2)	(Ч, Ч, Ч, 2, <i>c-ab</i> )	(Ч, 0, Ч, 2)	(Ч, Ч, Ч, 3)	(Ч, Ч, Ч, 3)	(Ч, 0, Ч, 2)
(0, Ч, Ч, 2)	(Н, Н, Ч, 2)	(Ч, Ч, Ч, 2, <i>c-ab</i> )	(Ч, Ч, Ч, 3)	(0, Ч, Ч, 2)	(Ч, Ч, Ч, 3)	(0, Ч, Ч, 2)
(Ч, Ч, Ч, 2, <i>a-bc</i> )	(Н, Н, Ч, 1)	(Ч, Ч, Ч, 1)	(Ч, Ч, Ч, 2, <i>a-bc</i> )			
(Ч, Ч, Ч, 2, <i>b-ac</i> )	(Н, Н, Ч, 1)	(Ч, Ч, Ч, 1)	(Ч, Ч, Ч, 2, <i>b-ac</i> )			
(Ч, Ч, Ч, 2, <i>c-ab</i> )	(Н, Н, Ч, 2)	— <sup>5</sup>	(Ч, Ч, Ч, 2, <i>c-ab</i> )			
(Ч, Н, Н, 2)	(Н, Ч, Н, 1)	(Ч, Н, Н, 1)	(Ч, Н, Н, 2)			
(Н, Ч, Н, 2)	(Ч, Н, Н, 1)	(Н, Ч, Н, 1)	(Н, Ч, Н, 2)			
(Н, Н, Ч, 2)	(Ч, Ч, Ч, 2, <i>c-ab</i> )	— <sup>5</sup>	(Н, Н, Ч, 2)			
(Ч, Ч, Ч, 3)	(Н, Н, Ч, 2)	(Ч, Ч, Ч, 2, <i>c-ab</i> )	(Ч, Ч, Ч, 3)			

<sup>1</sup>возможно только при отсутствии в проходе сборочных ячеек

<sup>2</sup>может быть использован только, если в  $L_j^{+y}$  нет сборочных ячеек

<sup>3</sup>может быть использован только, если в  $G - L_j^{+y}$  нет сборочных ячеек

<sup>4</sup>нарушает введенные ограничения

<sup>5</sup>никогда не приведет к оптимальному решению

Перемещение от  $L_j^{+x}$  к  $L_{j+1}^-$ . Данное перемещение устанавливает связь между  $j$ -ым и  $(j + 1)$ -ым проходами путем добавления одной из конфигураций, представленных на рис. 11. Эквивалентные классы, получаемые при данном перемещении, приведены в табл. 6–7. Все комбинации, не попавшие в данные таблицы, нарушают накладываемые ограничения.

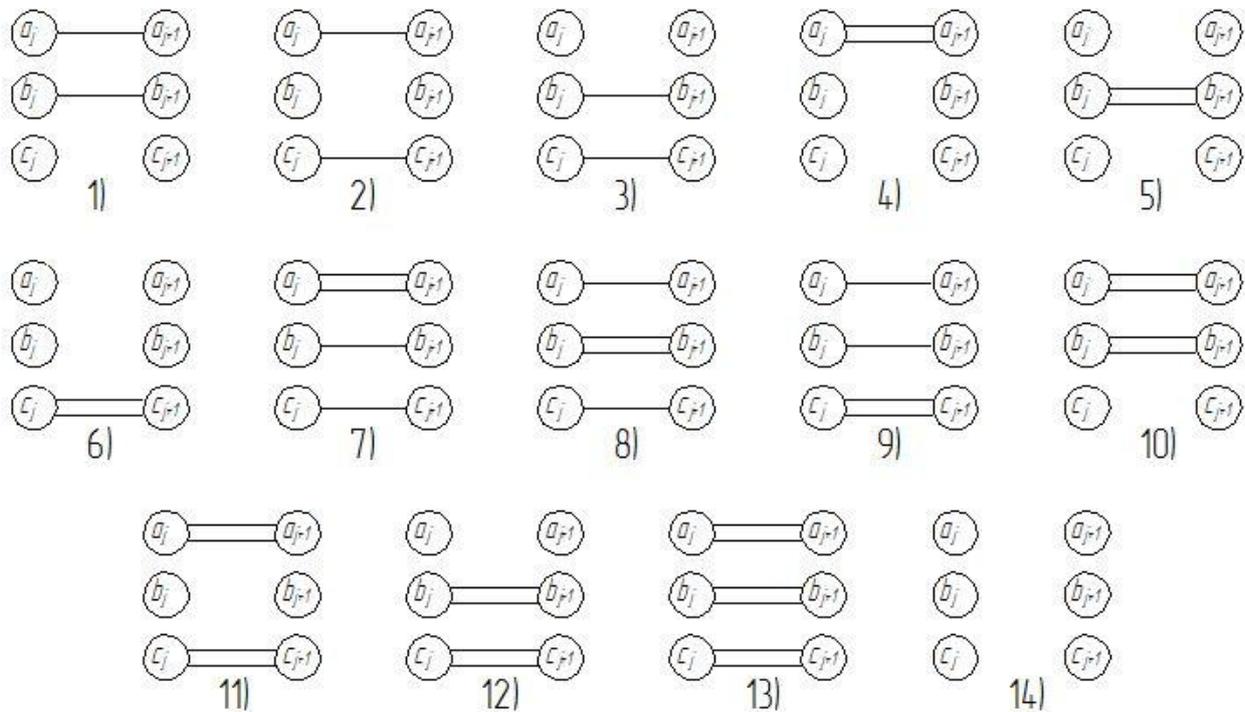


Рис. 11. Возможные конфигурации дуг при смене  $j$ -го прохода на  $(j + 1)$ -ый проход

Табл. 6. Эквивалентные классы, получаемые при добавлении перемещений из рис. 11 ко всем возможным конфигурациям  $L_j^{+x}$

$L_j^{+x} \rightarrow L_{j+1}^-$	1	2	3	7	8	9
(Н, Н, 0, 1)	(Н, Н, 0, 1)	— <sup>1</sup>	— <sup>1</sup>	— <sup>1</sup>	— <sup>1</sup>	— <sup>2</sup>
(Н, 0, Н, 1)	— <sup>1</sup>	(Н, 0, Н, 1)	— <sup>1</sup>	— <sup>1</sup>	— <sup>2</sup>	— <sup>1</sup>
(0, Н, Н, 1)	— <sup>1</sup>	— <sup>1</sup>	(0, Н, Н, 1)	— <sup>2</sup>	— <sup>1</sup>	— <sup>1</sup>
(Ч, Н, Н, 1)	— <sup>1</sup>	— <sup>1</sup>	(0, Н, Н, 1)	(Ч, Н, Н, 1)	— <sup>1</sup>	— <sup>1</sup>
(Н, Ч, Н, 1)	— <sup>1</sup>	(Н, 0, Н, 1)	— <sup>1</sup>	— <sup>1</sup>	(Н, Ч, Н, 1)	— <sup>1</sup>
(Н, Н, Ч, 1)	(Н, Н, 0, 1)	— <sup>1</sup>	— <sup>1</sup>	— <sup>1</sup>	— <sup>1</sup>	(Н, Н, Ч, 1)
(Ч, Н, Н, 2)	— <sup>1</sup>	— <sup>1</sup>	— <sup>1</sup>	(Ч, Н, Н, 2)	— <sup>1</sup>	— <sup>1</sup>
(Н, Ч, Н, 2)	— <sup>1</sup>	— <sup>1</sup>	— <sup>1</sup>	— <sup>1</sup>	(Н, Ч, Н, 2)	— <sup>1</sup>
(Н, Н, Ч, 2)	— <sup>1</sup>	(Н, Н, Ч, 2)				

<sup>1</sup>нарушает введенные ограничения

<sup>2</sup>никогда не приведет к оптимальному решению

**Табл. 7.** Эквивалентные классы, получаемые при добавлении перемещений из рис. 11  
ко всем возможным конфигурациям  $L_j^{+y}$

$L_j^{+x} \rightarrow L_{j+1}^-$	4	5	6	10	11	12	13	14
$(0, 0, 0, 0)^1$	— <sup>2</sup>	$(0, 0, 0, 0)$						
$(0, 0, 0, 1)^3$	— <sup>4</sup>	$(0, 0, 0, 1)^5$						
$(Ч, 0, 0, 1)$	$(Ч, 0, 0, 1)$	— <sup>4</sup>	— <sup>4</sup>	— <sup>2</sup>	— <sup>2</sup>	— <sup>4</sup>	— <sup>2</sup>	$(0, 0, 0, 1)^5$
$(0, Ч, 0, 1)$	— <sup>4</sup>	$(0, Ч, 0, 1)$	— <sup>4</sup>	— <sup>2</sup>	— <sup>4</sup>	— <sup>2</sup>	— <sup>2</sup>	$(0, 0, 0, 1)^5$
$(0, 0, Ч, 1)$	— <sup>4</sup>	— <sup>4</sup>	$(0, 0, Ч, 1)$	— <sup>4</sup>	— <sup>2</sup>	— <sup>2</sup>	— <sup>2</sup>	$(0, 0, 0, 1)^5$
$(Ч, Ч, 0, 1)$	$(Ч, 0, 0, 1)$	$(0, Ч, 0, 1)$	— <sup>4</sup>	$(Ч, Ч, 0, 1)$	— <sup>2</sup>	— <sup>2</sup>	— <sup>2</sup>	$(0, 0, 0, 1)^5$
$(Ч, 0, Ч, 1)$	$(Ч, 0, 0, 1)$	— <sup>4</sup>	$(0, 0, Ч, 1)$	— <sup>2</sup>	$(Ч, 0, Ч, 1)$	— <sup>2</sup>	— <sup>2</sup>	$(0, 0, 0, 1)^5$
$(0, Ч, Ч, 1)$	— <sup>4</sup>	$(0, Ч, 0, 1)$	$(0, 0, Ч, 1)$	— <sup>2</sup>	— <sup>2</sup>	$(0, Ч, Ч, 1)$	— <sup>2</sup>	$(0, 0, 0, 1)^5$
$(Ч, Ч, Ч, 1)$	$(Ч, 0, 0, 1)$	$(0, Ч, 0, 1)$	$(0, 0, Ч, 1)$	$(Ч, Ч, 0, 1)$	$(Ч, 0, Ч, 1)$	$(0, Ч, Ч, 1)$	$(Ч, Ч, Ч, 1)$	$(0, 0, 0, 1)^5$
$(Ч, Ч, 0, 2)$	— <sup>4</sup>	— <sup>4</sup>	— <sup>4</sup>	$(Ч, Ч, 0, 2)$	— <sup>4</sup>	— <sup>4</sup>	— <sup>2</sup>	— <sup>4</sup>
$(Ч, 0, Ч, 2)$	— <sup>4</sup>	— <sup>4</sup>	— <sup>4</sup>	— <sup>4</sup>	$(Ч, 0, Ч, 2)$	— <sup>4</sup>	— <sup>2</sup>	— <sup>4</sup>
$(0, Ч, Ч, 2)$	— <sup>4</sup>	$(0, Ч, Ч, 2)$	— <sup>2</sup>	— <sup>4</sup>				
$(Ч, Ч, Ч, 2, a-bc)$	— <sup>4</sup>	— <sup>4</sup>	— <sup>4</sup>	$(Ч, Ч, 0, 2)$	$(Ч, 0, Ч, 2)$	— <sup>4</sup>	$(Ч, Ч, Ч, 2, a-bc)$	— <sup>4</sup>
$(Ч, Ч, Ч, 2, b-ac)$	— <sup>4</sup>	— <sup>4</sup>	— <sup>4</sup>	$(Ч, Ч, 0, 2)$	— <sup>4</sup>	$(0, Ч, Ч, 2)$	$(Ч, Ч, Ч, 2, b-ac)$	— <sup>4</sup>
$(Ч, Ч, Ч, 2, c-ab)$	— <sup>4</sup>	— <sup>4</sup>	— <sup>4</sup>	— <sup>4</sup>	$(Ч, 0, Ч, 2)$	$(0, Ч, Ч, 2)$	$(Ч, Ч, Ч, 2, c-ab)$	— <sup>4</sup>
$(Ч, Ч, Ч, 3)$	— <sup>4</sup>	$(Ч, Ч, Ч, 3)$	— <sup>4</sup>					

<sup>1</sup> может быть использован только, если в  $L_j^{+x}$  нет сборочных ячеек

<sup>2</sup> нарушает введенные ограничения

<sup>3</sup> может быть использован только, если в  $G - L_j^{+x}$  нет сборочных ячеек

<sup>4</sup> никогда не приведет к оптимальному решению

<sup>5</sup> возможно только, если в  $G - L_j^{+x}$  нет сборочных ячеек

**Стоимость.** Стоимость (*cost*) каждого перемещения равна сумме длин ребер, включаемых в перемещение.

### 3.3. Применение расширенного алгоритма Рэтлиффа-Розенталя

Расширенный алгоритм Рэтлиффа-Розенталя может быть применен на складах с различной планировкой. Очевидно, что средний поперечный проход не обязательно должен находиться точно в середине, но может быть расположен между ближним и дальним поперечными проходами. Более того, данный алгоритм может быть применен на складах с одной или двумя возможностями для смены проходов, путем установления соответствующих расстояний между проходами в бесконечность. Алгоритм обладает линейной сложностью в зависимости от числа проходов и числа строк заказа (числа артикулов, ячейки хранения которых необходимо посетить).

### 3.4. Недостатки точных оптимальных методов

Существуют следующие недостатки точных оптимальных методов маршрутизации.

Во-первых, маршрут, получаемый благодаря использованию данных методов, может представляться сборщику заказов нелогичным, или неоптимальным, что в свою очередь может привести к отклонениям от заданного маршрута [5]. Авторами работ [6–7] данный феномен исследуется на реально существующих складах нидерландских компаний *De Bijenkorf* и *Ankor*.

Во-вторых, точные методы зависят от таких параметров, как расположение базы, ее фиксированности, числа блоков и формы склада (прямоугольная или нет). Точные методы могут быть использованы только для стандартных планировок (т.е., на прямоугольных одно- или многоблочных складах). Использование точных методов для многоблочных складов нецелесообразно ввиду огромного числа возможных эквивалентных классов и их комбинаций.

В-третьих, при использовании точных методов необходимо рассчитывать маршруты сборщиков для каждого заказа, что может сильно осложнить и замедлить работу всей системы управления складом.

В-четвертых, точные методы не принимают во внимание загруженность проходов, в то время как использование эвристических методов маршрутизации позволяет в некоторых случаях избежать чрезмерной загруженности.

Наконец, точные методы не учитывают тот факт, что на практике смена прохода или направления движения внутри прохода также влияют на затраты по времени. Использование эвристических методов маршрутизации позволяет снизить число смен проходов.

По этим причинам, обычно используются простые или комбинированные эвристические методы маршрутизации.

#### 4. Эвристические методы маршрутизации

Конфигурация склада, на которой рассматриваются описываемые далее эвристические методы представлена на рис. 12.

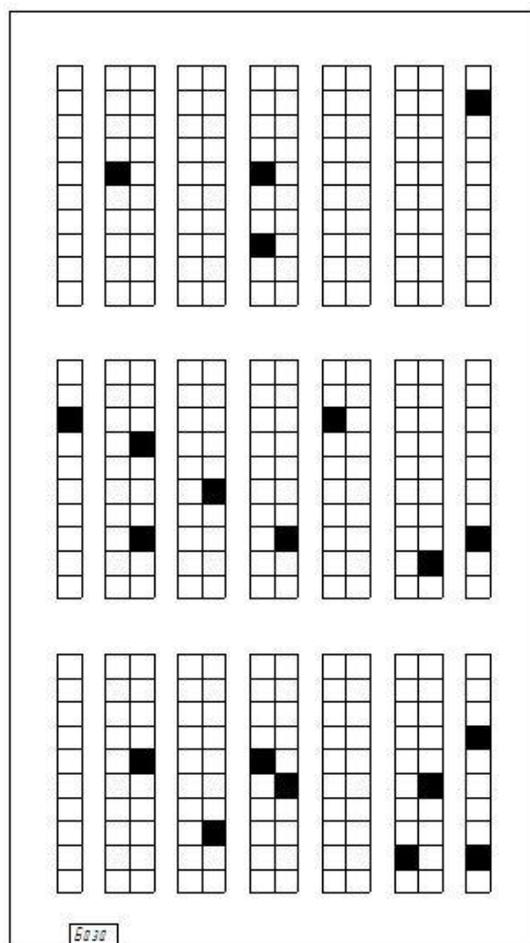


Рис. 12. Конфигурация склада с ячейками типового заказа (заштрихованные прямоугольники)

Прямоугольный склад состоит из шести сборочных проходов. Склад разделен на три блока, каждый состоит из шести под-проходов. Под-проход (*sub-aisle*) — это часть сборочного прохода внутри одного блока. Дальним блоком считается блок, наиболее удаленный от базы, которая располагается в ближней стороне склада, рядом с первым (слева) проходом. Заметим, что расположение базы также может оказывать влияние на среднее время сборки заказа. В работе [8] рассматривается влияние расположения базы, а также размера заказа на среднее время сборочного процесса. Показывается, что разница между расположением базы по центру ближнего прохода и расположением вблизи первого прохода меньше 1%. Также, на складе присутствует три поперечных прохода, один — в ближней стороне, один — в дальней, один — в середине. Поперечные проходы не содержат сборочных ячеек и используются только для перемещения от одного

сборочного прохода к другому. Для каждого блока существуют ближний и дальний (по отношению к базе) поперечные проходы, при этом дальний проход  $n$ -го блока является также ближним для  $(n + 1)$ -го блока. Число поперечных проходов определяется числом блоков плюс один и равно четырем.

Сборщики заказов могут пересекать проходы, двигаясь в любом направлении, а также могут изменять направление движения внутри прохода. Допускается, что проходы достаточно узки, чтобы сборщик заказов мог собирать товары с обеих сторон прохода без дополнительного изменения своего положения. Каждый заказ состоит из определенного числа товаров, случайным образом распределенных по складским ячейкам. Подразумевается, что заказ может быть собран полностью за один проход. Смена прохода возможна с использованием любого из поперечных проходов. Сборочные листы получают на базе, также как и собранный заказ должен быть доставлен на базу.

Для моделирования конфигурации склада, распределения типовых заказов и маршрутов, получаемых с использованием различных методов маршрутизации, используется программа «*Interactive Warehouse*» [9].

#### 4.1. S-образный метод

Наиболее простой эвристикой является S-образный (*S-shaped*) метод маршрутизации (еще одно встречающееся в литературе названием данного метода — метод пересечений, *traversal*). Он основан на том, что любой проход, содержащий хотя бы одну сборочную ячейку должен быть пересечен полностью от ближнего (дальнего) поперечного прохода до дальнего (ближнего) поперечного прохода. Пример применения S-образной эвристики на типовом заказе представлен на рис. 13. Данной эвристике посвящены работы [3–4, 6–8, 10–19].

Алгоритм S-образной эвристики состоит из следующих шагов [19].

- 1) Определяем крайний левый (далее, КЛ) проход, содержащий как минимум одну сборочную ячейку, и наиболее удаленный от базы блок (далее, НУБ), содержащий как минимум одну сборочную ячейку.
- 2) Маршрут начинается из базы по направлению к ближнему концу КЛ прохода.
- 3) Пересекаем КЛ проход до ближнего поперечного прохода НУБ, попутно собирая товары из пересекаемых сборочных ячеек.
- 4) Идем направо по ближнему поперечному проходу НУБ до первого под-прохода с непосещенными сборочными ячейками (далее, ПНСЯ). Если это единственный ПНСЯ в данном блоке, то собираем все необходимые товары и возвращаемся на ближний поперечный проход данного блока, не меняя под-проход. Если в данном блоке ПНСЯ более одного, то полностью пересекаем под-проход, не меняя направления движения.



## 4.2. Метод с возвратами

Еще одним простым эвристическим методом маршрутизации является метод с возвратами (*return*), при использовании которого сборщик заказов заходит и выходит из сборочных проходов через один поперечный проход, пересекая сборочный проход целиком только в случае перехода из одного блока в другой, причем это возможно сделать только либо через КЛ ПНСЯ, либо через крайний правый (далее, КП) ПНСЯ. Данный эвристический метод маршрутизации рассматривается в работах [8, 11, 13–14, 20–21]. Пример применения эвристики с возвратами на типовом заказе представлен на рис. 14.

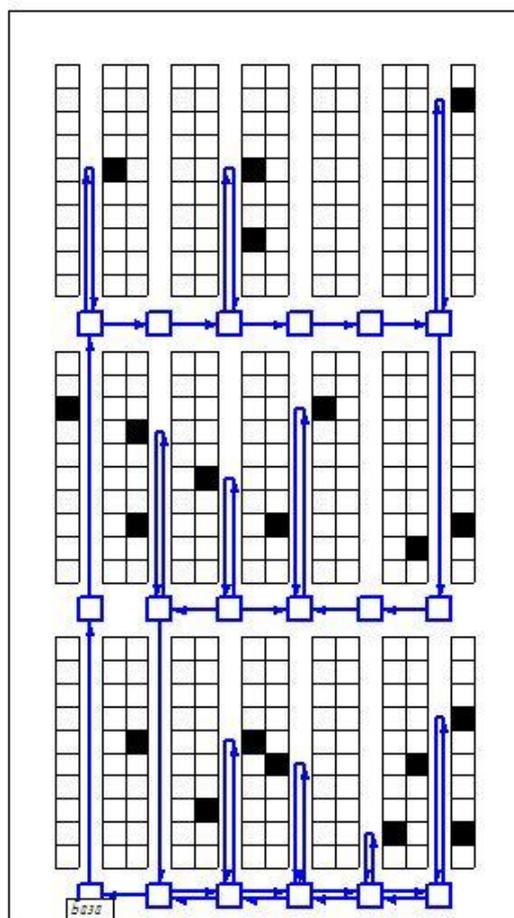


Рис. 14. Пример маршрута, полученного с использованием эвристики маршрутизации с возвратами

Алгоритм эвристики с возвратами состоит из следующих шагов.

- 1) Определяем КЛ ПНСЯ и НУБ, содержащий как минимум одну сборочную ячейку.
- 2) Маршрут начинается из базы по направлению к ближнему концу КЛ ПНСЯ.
- 3) Пересекаем КЛ ПНСЯ до ближнего поперечного прохода НУБ, попутно собирая товары из пересекаемых сборочных ячеек.
- 4) Идем направо по ближнему поперечному проходу НУБ до первого ПНСЯ. Собираем все необходимые товары в ПНСЯ и возвращаемся на ближний поперечный проход данного блока, не меняя под-проход.
- 5) На данном шаге, сборщик заказов находится на ближнем поперечном проходе блока. Возможны следующие два варианта.

- а) Если в текущем блоке еще остаются ПНСЯ. Определяем наиболее удаленный от текущей позиции ПНСЯ текущего блока. Идем на шаг 6.
  - б) Если в текущем блоке больше нет ПНСЯ. В данном случае, определяем наиболее удаленный от текущей позиции ПНСЯ следующего блока, идем по направлению к ближнему из них, пересекаем его полностью, попутно собрав в нем необходимые артикулы, идем на шаг 7.
- б) Следуем наиболее коротким путем по ближнему поперечному проходу, начиная с текущей позиции, заходя во все еще ПНСЯ, собираем все необходимые товары и возвращаемся на ближний поперечный проход блока. Повторяем данный шаг для всех ПНСЯ текущего блока.
  - 7) Если ближний к базе блок еще не был пройден, то возвращаемся на шаг 5.
  - 8) Возвращаемся на базу.

### 4.3. Серединный метод

Серединный (*midpoint*) метод маршрутизации фактически разбивает каждый блок склада на две секции. Сборочные ячейки, относящиеся к ближней (к базе) секции блока, посещаются через ближний поперечный проход блока; сборочные ячейки дальней секции — через дальний поперечный проход. Сборщик заказов пересекает проход целиком только для перехода из одного блока в другой, при этом это возможно сделать только через КЛ ПНСЯ, либо через КП ПНСЯ. Данный эвристический метод маршрутизации представлен в работах [8, 11, 20–21]. Пример применения серединной эвристики на типовом заказе представлен на рис. 15.

Алгоритм серединной эвристики состоит из следующих шагов.

- 1) Определяем КЛ ПНСЯ и НУБ, содержащий как минимум одну сборочную ячейку.
- 2) Маршрут начинается из базы по направлению к ближнему концу КЛ ПНСЯ.
- 3) Пересекаем КЛ ПНСЯ до ближнего поперечного прохода НУБ, попутно собирая товары из пересекаемых сборочных ячеек.
- 4) Идем направо по ближнему поперечному проходу НУБ до первого ПНСЯ. Если это единственный ПНСЯ в данном блоке, то собираем все необходимые товары и возвращаемся на ближний поперечный проход данного блока, не меняя под-проход. Если в данном блоке более одного ПНСЯ, то полностью пересекаем под-проход.
- 5) На данном шаге, сборщик заказов находится на дальнем поперечном проходе блока. Возможны следующие два варианта.
  - а) Если в текущем блоке еще остаются ПНСЯ. Определяем наиболее удаленный от текущей позиции ПНСЯ текущего блока. Идем на шаг 6.
  - б) Если в текущем блоке больше нет ПНСЯ. В данном случае, продолжаем движение в текущем сборочном проходе до следующего поперечного прохода и идем на шаг 9.
- б) Следуем наиболее коротким путем по дальнему поперечному проходу, начиная с текущей позиции, заходя во все ПНСЯ на расстояние, не превышающее половины длины под-прохода, собираем все необходимые товары, возвращаемся на дальний поперечный проход текущего блока.

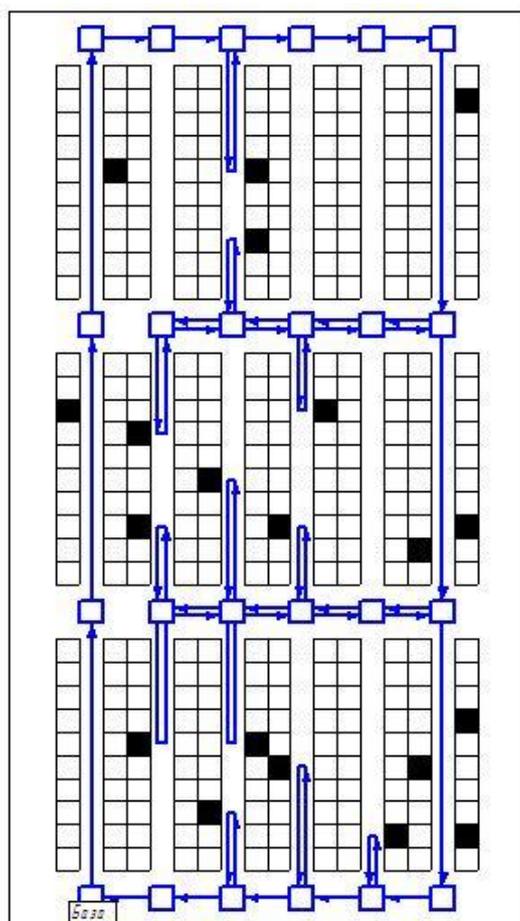
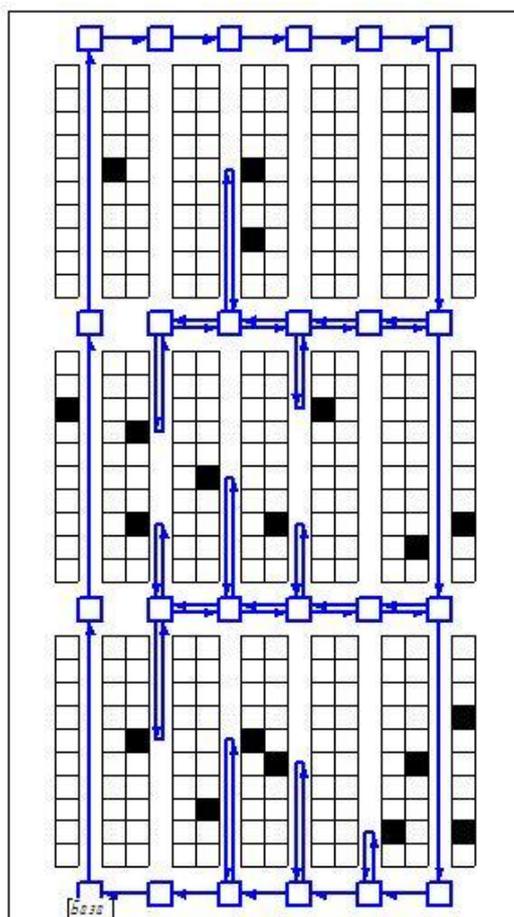


Рис. 15. Пример маршрута, полученного с использованием серединной эвристики маршрутизации

- 7) Полностью пересекаем последний под-проход текущего блока до ближнего поперечного прохода данного блока.
- 8) На данном шаге, сборщик заказов находится на ближнем поперечном проходе блока. Возможны следующие два варианта.
  - а) Если в текущем блоке еще остаются ПНСЯ. Определяем наиболее удаленный от текущей позиции ПНСЯ текущего блока. Идем на шаг 9.
  - б) Если в текущем блоке больше нет ПНСЯ. В данном случае, определяем наиболее удаленный от текущей позиции ПНСЯ следующего блока, идем по направлению к ближнему из них, переходим на шаг 10.
- 9) Следуем наиболее коротким путем по ближнему поперечному проходу, начиная с текущей позиции, заходя во все ПНСЯ на расстояние, не превышающее половины длины под-прохода, собираем все необходимые товары, возвращаемся на ближний поперечный проход текущего блока.
- 10) Если ближний к базе блок еще не был пройден, то возвращаемся на шаг 5, принимая во внимание, что ближний поперечный проход  $n$ -го блока является дальним поперечным проходом  $(n - 1)$ -го блока.
- 11) Возвращаемся на базу.

#### 4.4. Метод с захождением на наибольший интервал

Метод с захождением на наибольший интервал (*largest gap*) схож со срединным методом за исключением того, что сборщик заказов заходит в под-проход со сборочными ячейками не на половину длины под-прохода, а на длину наибольшего интервала между: а) смежными сборочными ячейками данного под-прохода; б) наиболее удаленной сборочной ячейкой и дальним поперечным проходом; в) наименее удаленной сборочной ячейкой и ближним поперечным проходом. Если наибольший интервал приходится на расстояние между двумя смежными сборочными ячейками, то сборщик заказов заходит в под-проход дважды — один раз через ближний поперечный проход, один раз через дальний поперечный проход. Таким образом, наибольший интервал — это та часть под-прохода, которая остается непосещенной сборщиком заказов. Дальний поперечный проход блока может быть достигнут только через КЛ или КП ПНСЯ. Данный эвристический метод маршрутизации представлен в работах [8, 14, 19–20, 22]. Пример применения эвристики с захождением на наибольший интервал на типовом заказе представлен на рис. 16.



**Рис. 16.** Пример маршрута, полученного с использованием эвристики маршрутизации с захождением на наибольший интервал

Алгоритм срединной эвристики состоит из следующих шагов.

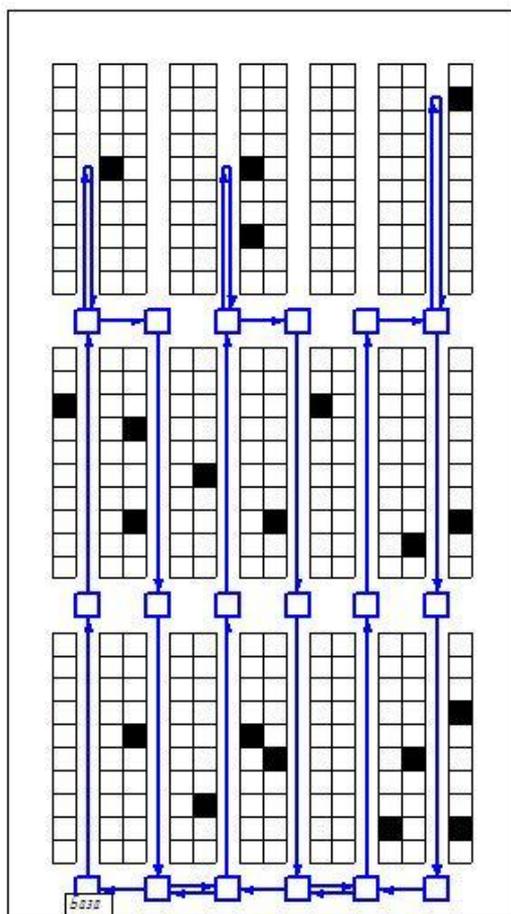
- 1) Определяем КЛ ПНСЯ и НУБ, содержащий как минимум одну сборочную ячейку.

- 2) Маршрут начинается из базы по направлению к ближнему концу КЛ ПНСЯ.
- 3) Пересекаем КЛ ПНСЯ до ближнего поперечного прохода НУБ, попутно собирая товары из пересекаемых сборочных ячеек.
- 4) Идем направо по ближнему поперечному проходу НУБ до первого ПНСЯ. Если это единственный ПНСЯ в данном блоке, то собираем все необходимые товары и возвращаемся на ближний поперечный проход данного блока, не меняя под-проход. Если в данном блоке более одного ПНСЯ, то полностью пересекаем под-проход.
- 5) На данном шаге, сборщик заказов находится на дальнем поперечном проходе блока. Возможны следующие два варианта.
  - а) Если в текущем блоке еще остаются ПНСЯ. Определяем наиболее удаленный от текущей позиции ПНСЯ текущего блока. Идем на шаг 6.
  - б) Если в текущем блоке больше нет ПНСЯ. В данном случае, продолжаем движение в текущем сборочном проходе до следующего поперечного прохода и идем на шаг 9.
- 6) Следуем наиболее коротким путем по дальнему поперечному проходу, начиная с текущей позиции, заходя во все ПНСЯ на расстояние, не превышающее наибольший интервал для под-прохода, собираем все необходимые товары, возвращаемся на дальний поперечный проход текущего блока.
- 7) Полностью пересекаем последний под-проход текущего блока до ближнего поперечного прохода данного блока.
- 8) На данном шаге, сборщик заказов находится на ближнем поперечном проходе блока. Возможны следующие два варианта.
  - а) Если в текущем блоке еще остаются ПНСЯ. Определяем наиболее удаленный от текущей позиции ПНСЯ текущего блока. Идем на шаг 9.
  - б) Если в текущем блоке больше нет ПНСЯ. В данном случае, определяем наиболее удаленный от текущей позиции ПНСЯ следующего блока, идем по направлению к ближнему из них, переходим на шаг 10.
- 9) Следуем наиболее коротким путем по ближнему поперечному проходу, начиная с текущей позиции, заходя во все ПНСЯ на расстояние, не превышающее наибольший интервал для под-прохода, собираем все необходимые товары, возвращаемся на ближний поперечный проход текущего блока.
- 10) Если ближний к базе блок еще не был пройден, то возвращаемся на шаг 5, принимая во внимание, что ближний поперечный проход  $n$ -го блока является дальним поперечным проходом  $(n - 1)$ -го блока.
- 11) Возвращаемся на базу.

#### 4.5. Последовательный метод

Последовательный (или, метод «проход за проходом», aisle-by-aisle) эвристический метод маршрутизации во многоблочных складах представлен в работе [23]. Сборочный маршрут, получающийся в результате использования данной эвристики, посещает каждый

сборочный проход только один раз. Таким образом, вначале собираются все необходимые артикулы из первого прохода, затем — из второго и т.д. Для определения поперечного прохода для перехода между сборочными проходами используется динамическое программирование. Пример применения последовательной эвристики на типовом заказе представлен на рис. 17.



**Рис. 17.** Пример маршрута, полученного с использованием последовательной эвристики маршрутизации

Данный последовательный метод основывается на работах [24–25]. Он позволяет, в частности, сузить ширину поперечных проходов, т.к. движение по ним будет осуществляться только в одну сторону.

Алгоритм последовательной эвристики состоит из следующих шагов.

- 1) Определяем КЛ ПНСЯ и НУБ, содержащий как минимум одну сборочную ячейку.
- 2) Маршрут начинается из базы по направлению к ближнему концу КЛ ПНСЯ.
- 3) Пересекаем КЛ ПНСЯ до ближнего поперечного прохода НУБ, попутно собирая товары из пересекаемых сборочных ячеек.
- 4) Если существуют ПНСЯ, то выбираем поперечный проход для перехода на следующий сборочный проход, иначе идем на шаг 5. Для текущего сборочного прохода  $m$  рассчитываем расстояние  $C_m(i, j)$ , которое требуется пройти, стартовав из  $i$ -го поперечного прохода (на первой итерации, из ближайшего к базе ближнего

поперечного прохода), собрав все необходимые артикулы в первом проходе и перейдя на следующий сборочный проход через  $j$ -ый поперечный проход, следующим образом:

$$C_m(i, j) = B_{1m}(i, j) + |i - j|(L + A) + B_{2m}(i, j).$$

Здесь  $B_{1m}(i, j)$  — длина плеча для обхода всех сборочных ячеек прохода  $m$ , находящихся дальше поперечного прохода  $\min(i, j)$ ;  $B_{2m}(i, j)$  — длина плеча для обхода всех сборочных ячеек прохода  $m$ , находящихся ближе поперечного прохода  $\max(i, j)$ ;  $L$  — длина блока склада (без учета поперечных проходов);  $A$  — ширина поперечных проходов.

Величина  $B_{1m}(i, j)$  определяется по формуле

$$B_{1m}(i, j) = \begin{cases} 0, & \text{если } K_m = 0 \text{ или } X_m^- \geq \min(iL, jL); \\ 2 \left[ \min(iL, jL) - X_m^- + A \left( 0.5 + \left\lfloor \frac{\min(iL, jL) - X_m^-}{L} \right\rfloor \right) \right] & \text{в противном случае.} \end{cases}$$

Здесь  $K_m$  — число сборочных ячеек прохода  $m$ ;  $X_m^- = \min_t \{X_m(t)\}$  — продольная координата наиболее удаленной (от базы) сборочной ячейки прохода  $m$ ;  $X_m(t)$  — продольная координата сборочной ячейки  $t$  прохода  $m$  ( $0 \leq X_m(T) < T$ ),  $m = 1, 2, \dots, M$ ,  $t = 1, 2, \dots, K_m$ .

Величина  $B_{2m}(i, j)$  определяется по формуле

$$B_{2m}(i, j) = \begin{cases} 0, & \text{если } K_m = 0 \text{ или } X_m^+ < \max(iL, jL); \\ 2 \left[ \max(iL, jL) - X_m^+ + A \left( 0.5 + \left\lfloor \frac{X_m^+ - \max(iL, jL)}{L} \right\rfloor \right) \right] & \text{в противном случае.} \end{cases}$$

Здесь  $X_m^+ = \max_t \{X_m(t)\}$  — продольная координата ближайшей (к базе) сборочной ячейки прохода  $m$ . Переходим на следующий проход, используя такой поперечный проход, для которого это расстояние минимально. Выполняем данный шаг, пока не останется ПНСЯ.

5) Возвращаемся на базу.

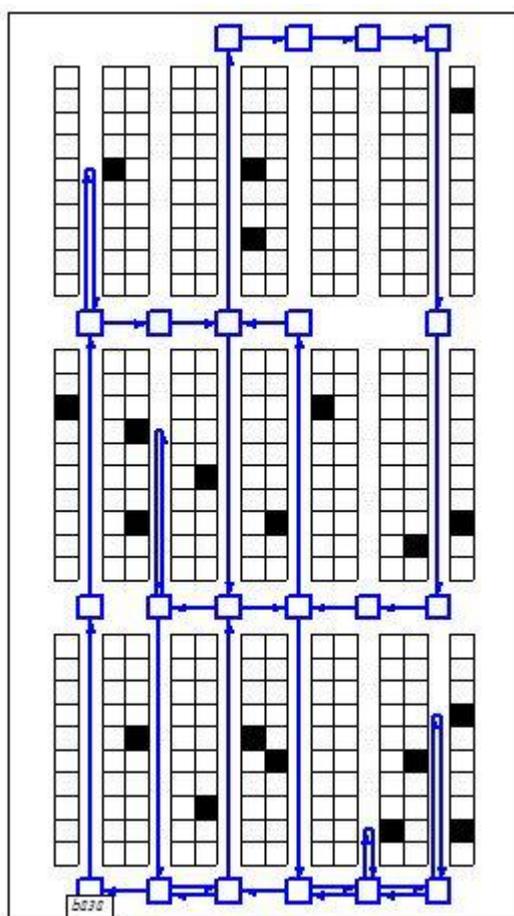
#### 4.6. Составной метод

Составной (*composite*) метод предложен в работе [26]. Он сочетает в себе преимущества  $S$ -образной эвристики и эвристики с возвратами. Метод минимизирует преодолеваемое расстояние между наиболее удаленными сборочными ячейками двух смежных ПНСЯ и определяет наилучший способ преодоления прохода — пересечение или заход с возвратом. Пример применения составной эвристики на типовом заказе представлен на рис. 18.

Алгоритм составной эвристики состоит из следующих шагов.

- 1) Определяем КЛ ПНСЯ и НУБ, содержащий как минимум одну сборочную ячейку.
- 2) Маршрут начинается из базы по направлению к ближайшему концу КЛ ПНСЯ.

- 3) Пересекаем КЛ ПНСЯ до ближнего поперечного прохода НУБ, попутно собирая товары из пересекаемых сборочных ячеек.
- 4) Идем направо по ближнему поперечному проходу НУБ до первого ПНСЯ. Если это единственный ПНСЯ в данном блоке, то собираем все необходимые товары и возвращаемся на ближний поперечный проход данного блока, не меняя под-проход, переходим на шаг 6.
- 5) Если в данном блоке более одного ПНСЯ, то вычисляем расстояние между наиболее удаленными ячейками текущего ПНСЯ и ближайшего смежного ПНСЯ текущего блока при пересечении текущего прохода целиком и при заходе в него с возвратом. Выбираем способ, ведущий к наименьшему преодолеваемому расстоянию. Повторяем данный шаг до тех пор, пока не попадем в последний проход со сборочными ячейками.



**Рис. 18.** Пример маршрута, полученного с использованием составной эвристики маршрутизации

- 6) Определяем ближайший ПНСЯ следующего блока. Если это единственный ПНСЯ в данном блоке, то собираем все необходимые артикулы, пересекаем его целиком и вновь выполняем данный шаг.
- 7) Если ближний к базе блок еще не был пройден, то возвращаемся на шаг 5.
- 8) Возвращаемся на базу.

#### 4.7. Комбинированный метод

Комбинированный (*combined*) метод схож с составным методом, однако решение, пересекать проход целиком или сделать заход с возвратом без изменения поперечного прохода принимается на основе динамического программирования [4, 19]. Каждый ПНСЯ посещается только один раз. Пример применения комбинированной эвристики на типовом заказе представлен на рис. 19.

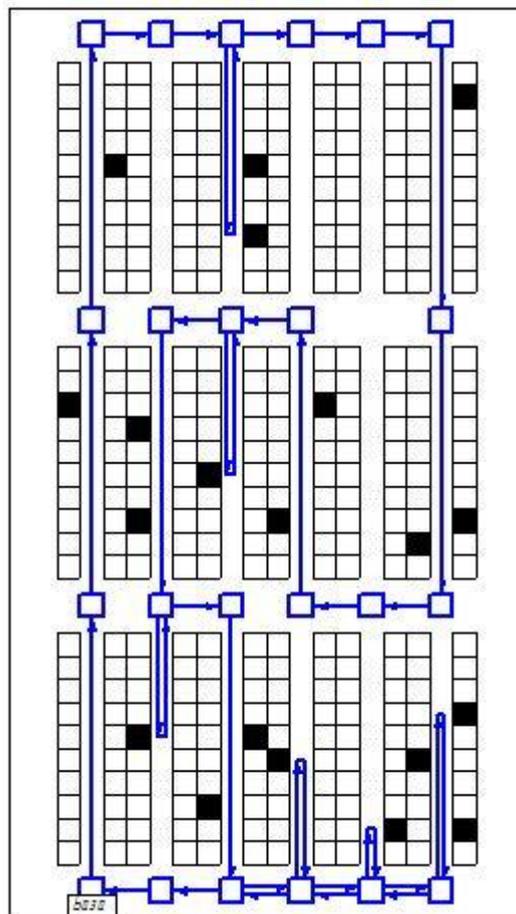


Рис. 19. Пример маршрута, полученного с использованием комбинированной эвристики маршрутизации

Введем следующие обозначения:

$k$  — число блоков;

$n$  — число сборочных проходов;

$a_{ij}$  — координата дальнего (от базы) конца  $j$ -го под-прохода  $i$ -го блока ( $i = 1, \dots, k$ ;  
 $j = 1, \dots, n$ );

$b_{ij}$  — координата ближнего (к базе) конца  $j$ -го под-прохода  $i$ -го блока ( $i = 1, \dots, k$ ;  
 $j = 1, \dots, n$ );

$d$  — база.

Заметим, что для  $i = 1, 2, \dots, k - 1$  справедливо равенство  $b_{ij} = a_{i+1,j}$ , т.к. мы допускаем, что сборщик заказов перемещается по середине поперечных проходов.

Расстояние между концами под-проходов и серединой поперечных проходов включается в расстояние под-проходов. Для каждого блока используется метод динамического программирования.

**Метод динамического программирования** используется для маршрутизации сборщиков заказов внутри  $i$ -го блока ( $i = 1, \dots, k$ ). Маршрут начинается с КЛ ПНСЯ  $l$  и заканчивается в КП ПНСЯ  $r$ . Пусть  $L_j$  определяет ЧМП, с помощью которого можно обойти все сборочные ячейки начиная с под-прохода  $l$  до под-прохода  $j$ . Будем различать два вида ЧМП:

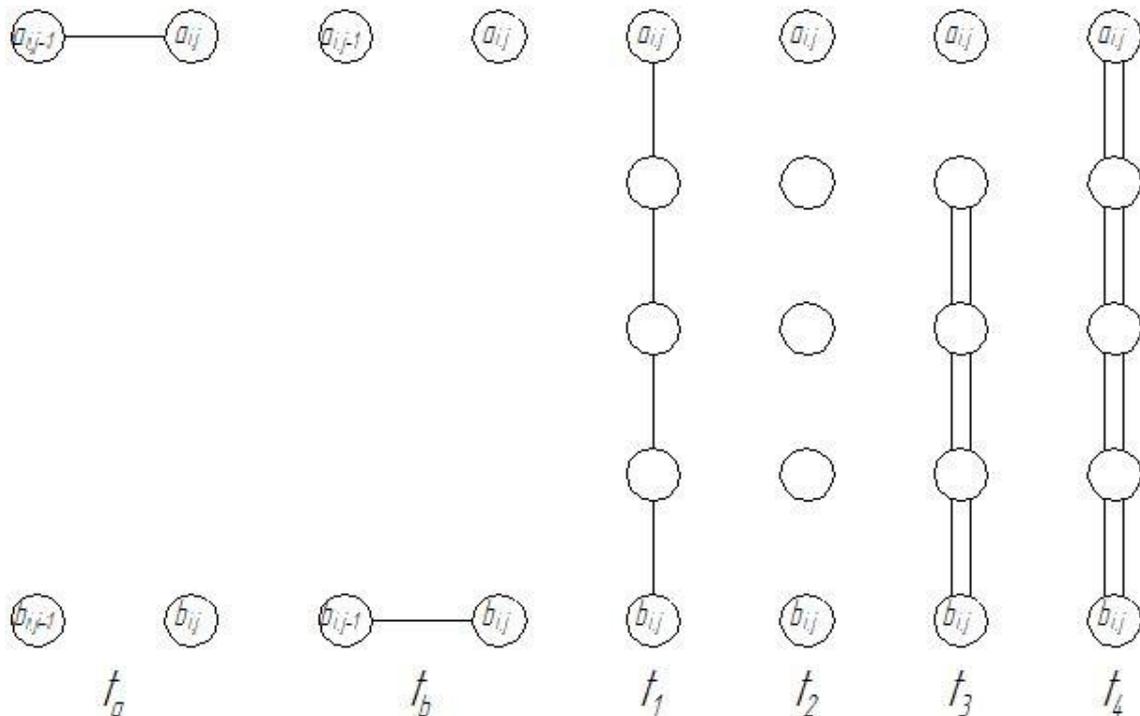
$L_j^a$  — ЧМП, заканчивающийся в дальнем поперечном проходе  $j$ ;

$L_j^b$  — ЧМП, заканчивающийся в ближнем поперечном проходе  $j$ .

Будем различать два способа перехода из  $(j-1)$ -го под-прохода в  $j$ -ый под-проход (рис. 20):

$t_a$  — переход через дальний поперечный проход блока;

$t_b$  — переход через ближний поперечный проход блока.



**Рис. 20.** Перемещения, используемые в методе динамического программирования

Будем различать способы сборки всех артикулов из  $j$ -го под-прохода. Для этого определим четыре вида перемещений (рис. 20):

$t_1$  — полное перемещение через под-проход;

$t_2$  — без захождения в под-проход;

$t_3$  — входение в под-проход и выход из него осуществляется через ближний поперечный проход блока;

$t_4$  — вхождение в под-проход и выход из него осуществляется через дальний поперечный проход блока.

Очевидно, что перемещение  $t_2$  применяется только в случае отсутствия в данном под-проходе сборочных ячеек.

С помощью  $L_j + t_w$  определим, что ЧМП  $L_j$  дополнен перемещением  $t_w$  ( $w = 1, 2, 3, 4, a, b$ ). Функция  $c(\cdot)$  возвращает время передвижения, соответствующее аргументу функции, например,  $c(L_j^b + t_1)$  возвращает время, необходимое для прохода по частичному маршруту  $L_j^b$ , плюс время, необходимое для совершения перемещения  $t_1$ . Заметим, что перемещения содержат информацию только о том, каким образом происходит посещения под-проходов или выход из них. Точный путь внутри под-прохода, и, таким образом, соответствующее время передвижения, зависит от расположения сборочных ячеек внутри рассматриваемого под-прохода.

С использованием потенциальных состояний, возможных перемещений между состояниями и стоимостей (времени передвижения) данных перемещений, можно описать метод динамического программирования. Данный метод будет определять ЧМП передвижения через один блок.

Рассмотрим  $i$ -ый блок. Если  $i$ -ый блок — НУБ, содержащий хотя бы один ПНСЯ, то начинаем с двух ЧМП:

–  $L_l^a$ , который начинается в узле  $b_{il}$ , заканчивается в узле  $a_{il}$  и состоит из перемещения  $t_1$ ;

–  $L_l^b$ , который начинается и заканчивается в узле  $b_{il}$ , и состоит из перемещения  $t_3$ .

В противном случае, начинаем с двух ЧМП:

–  $L_l^a$ , который начинается и заканчивается в узле  $a_{il}$  и состоит из перемещения  $t_4$ ;

–  $L_l^b$ , который начинается в узле  $a_{il}$ , заканчивается в узле  $b_{il}$ , и состоит из перемещения  $t_1$ .

Для каждого последующего  $j$ -го под-прохода ( $l < j < r$ ) определим  $L_j^a$  и  $L_j^b$  следующим образом. Если  $j$ -ый под-проход — ПНСЯ, то

$$L_j^a = \begin{cases} L_{j-1}^a + t_a + t_4, & \text{если } c(L_{j-1}^a + t_a + t_4) < c(L_{j-1}^b + t_b + t_1), \\ L_{j-1}^b + t_b + t_1, & \text{в противном случае,} \end{cases}$$

$$L_j^b = \begin{cases} L_{j-1}^b + t_b + t_3, & \text{если } c(L_{j-1}^b + t_b + t_3) < c(L_{j-1}^a + t_a + t_1), \\ L_{j-1}^a + t_a + t_1, & \text{в противном случае.} \end{cases}$$

Если  $j$ -ый под-проход — не ПНСЯ, то

$$L_j^a = L_{j-1}^a + t_a,$$

$$L_j^b = L_{j-1}^b + t_b.$$

Для последнего под-прохода блока (под-прохода  $r$ ) необходимо вычислить только величину  $L_r^b$  по следующей формуле:

$$L_r^b = \begin{cases} L_{r-1}^b + t_b + t_3, & \text{если } c(L_{r-1}^b + t_b + t_3) < c(L_{r-1}^a + t_a + t_1), \\ L_{r-1}^a + t_a + t_1, & \text{в противном случае.} \end{cases}$$

Получившийся в результате ЧМП  $L_r^b$  будет использован для формирования полного сборочного маршрута. На данном шаге  $L_r^a$  не требуется, т.к. по завершении сборки всех элементов в блоке, необходимо перейти на ближайший (к базе) поперечный проход блока для возможности продолжения маршрута на следующем блоке.

#### Алгоритм построения маршрута.

- 1) Определяем КЛ ПНСЯ и НУБ, содержащий хотя бы одну сборочную ячейку.
- 2) Маршрут начинается из базы по направлению к началу КЛ ПНСЯ.
- 3) Пересекаем КЛ ПНСЯ до ближнего поперечного прохода НУБ (блок  $i_{\min}$ ), попутно собирая товары из посещаемых сборочных ячеек.
- 4) Устанавливаем  $i = i_{\min}$ .
- 5) Определяем, содержит ли  $i$ -ый блок ПНСЯ, которые не были посещены на шаге 3.
  - а) Если в  $i$ -ом блоке нет ни одного ПНСЯ, то пересекаем ближайший под-проход  $i$ -го блока до следующего блока. Идем на шаг 7.
  - б) Если в  $i$ -ом блоке есть ПНСЯ, то определяем КЛ ПНСЯ и КП ПНСЯ (под-проход  $l$  и под-проход  $r$ , соответственно), исключая все под-проходы, которые уже были посещены на шаге 3. Из текущей позиции переходим в ближайший из данных двух под-проходов ( $j_{\min}$ ).
- 6) Применяем для  $i$ -го блока описанный выше метод динамического программирования. Если на шаге 5 величина  $j_{\min} = l$ , то добавляем получившийся ЧМП к маршруту сборщика заказов. Если  $j_{\min} = r$ , то «переворачиваем» получившийся ЧМП (таким образом, чтобы маршрут начинался в под-проходе  $r$  и завершался в под-проходе  $l$ ). Добавляем «перевернутый» ЧМП к маршруту сборщика заказов. «Переворачивание» маршрута означает, что сборщик заказов будет посещать под-проходы справа налево, в то время как вычисления были произведены слева направо.
- 7) Когда был вычислен  $k$ -ый (ближайший к базе) блок, сборщик заказов возвращается на базу. В противном случае, увеличиваем  $i$  на единицу и возвращаемся к шагу 5.

#### 4.8. Модифицированный комбинированный метод

Комбинированный метод, рассмотренный в предыдущем разделе, возможно модифицировать следующим образом [19] и получить модифицированный комбинированный метод (*combined+*).



на пути к НУБ, и для правых ( $n - x$ ) под-проходов каждого блока на обратном пути. В итоге, мы получаем маршрут, гарантированно короче, или такой же длины, что и маршрут, полученный комбинированной эвристикой.

### Заключение

Задача маршрутизации сборщиков заказов на складе является частным случаем задачи коммивояжера. Существующие точные алгоритмы получения оптимальных маршрутов сложны в применении, вследствие чего широко используются эвристические методы маршрутизации. Алгоритмы восьми эвристических методов маршрутизации сборщиков заказов приведены в данной статье. В продолжении цикла статей «Процесс комплектования заказов на складе» будет проведено сравнение эффективности приведенных эвристических методов, а также их отклонение от оптимальных результатов.

### Список литературы

1. Коробков Е.В. Процесс комплектования заказов на складе. Обзор // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2015. № 3. С. 153-183. DOI: [10.7463/0315.0763161](https://doi.org/10.7463/0315.0763161)
2. Lawler E.L., Lenstra J.K., Kan A.R., Shmoys D.B. The Travelling Salesman Problem: A Guided Tour of Combinational Optimization. Vol. 3. New York, NY: Wiley, 1985. 476 p.
3. Ratliff H.D., Rosenthal A.S. Order-Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem // Operations Research. 1983. Vol. 31, no. 3. P. 507-521. DOI: [10.1287/opre.31.3.507](https://doi.org/10.1287/opre.31.3.507)
4. Roodbergen K.J., de Koster R. Routing Order Pickers in a Warehouse with a Middle Aisle // European Journal of Operational Research. 2001. Vol. 133, no. 1. P. 32-43. DOI: [10.1016/S0377-2217\(00\)00177-6](https://doi.org/10.1016/S0377-2217(00)00177-6)
5. Gademann N., Velde S. Order Batching to Minimize Total Travel Time in a Parallel-Aisle Warehouse // IIE Transactions. 2005. Vol. 37, no. 1. P. 63–75. DOI: [10.1080/07408170590516917](https://doi.org/10.1080/07408170590516917)
6. De Koster R., Roodbergen K.J., van Voorden R. Reduction of Walking Time in the Distribution Center of De Bijenkorf // In: New Trends in Distribution Logistics / ed. by M.G. Speranza. Berlin: Springer-Verlag, 2000. P. 215-234. DOI: [10.1007/978-3-642-58568-5\\_11](https://doi.org/10.1007/978-3-642-58568-5_11)
7. Dekker R., de Koster R., Roodbergen K.J., van Kalleveen H. Improving Order-Picking Response Time at Ankor's Warehouse // Interface. 2004. Vol. 34, no. 4. P. 303–313. DOI: [10.1287/inte.1040.0083](https://doi.org/10.1287/inte.1040.0083)
8. Petersen II C.G., Schmenner R.W. An Evaluation of Routing and Volume-Based Storage Policies in an Order Picking Operation // Decision Sciences. 1999. Vol. 30, no. 2. P. 481–501. DOI: [10.1111/j.1540-5915.1999.tb01619.x](https://doi.org/10.1111/j.1540-5915.1999.tb01619.x)

9. Interactive Warehouse: web-site. Available at:  
<http://www.roodbergen.com/warehouse/index.php> , accessed January 2015.
10. Petersen II C.G., Aase G. A Comparison of Picking, Storage, and Routing Policies in Manual Order Picking // International Journal of Production Economics. 2004. Vol. 92, no. 1. P. 11-19. DOI: [10.1016/j.ijpe.2003.09.006](https://doi.org/10.1016/j.ijpe.2003.09.006)
11. De Koster R., Le-Duc T., Roodbergen K.J. Design and Control of Warehouse Order Picking: A Literature Review // European Journal of Operational Research. 2007. Vol. 182, no. 2. P. 481–501. DOI: [10.1016/j.ejor.2006.07.009](https://doi.org/10.1016/j.ejor.2006.07.009)
12. Gu J., Goetschalckx M., McGinnis L.F. Research on Warehouse Design and Performance Evaluation: A Comprehensive Review // European Journal of Operational Research. 2010. Vol. 203, no. 3. P. 539-549. DOI: [10.1016/j.ejor.2009.07.031](https://doi.org/10.1016/j.ejor.2009.07.031)
13. Le-Duc T., De Koster R. Travel Distance Estimation and Storage Zone Optimization in a 2-Block Class-Based Strategy Warehouse // International Journal of Production Research. 2005. Vol. 43, no. 17. P. 3561–3581. DOI: [10.1080/00207540500142894](https://doi.org/10.1080/00207540500142894)
14. Petersen II C.G. The Impact of Routing and Storage Policies on Warehouse Efficiency // International Journal of Operations and Production Management. 1999. Vol. 19, no. 10. P. 1053–1064. DOI: [10.1108/01443579910287073](https://doi.org/10.1108/01443579910287073)
15. Petersen II C.G., Aase G.R., Heiser D.R. Improving Order-Picking Performance through the Implementation of Class-Based Storage // International Journal of Physical Distribution and Logistics Management. 2004. Vol. 34, no. 7. P. 532-544. DOI: [10.1108/09600030410552230](https://doi.org/10.1108/09600030410552230)
16. De Koster R., van der Poort E.S., Wolters M. Efficient Orderbatching Methods in Warehouse // International Journal of Production Research. 1999. Vol. 37, no. 7. P. 1479-1504. DOI: [10.1080/002075499191094](https://doi.org/10.1080/002075499191094)
17. Le-Duc T., De Koster R. Travel Time Estimation and Order Batching in a 2-Block Warehouse // European Journal of Operational Research. 2007. Vol. 176, no. 1. P. 374–388. DOI: [10.1016/j.ejor.2005.03.052](https://doi.org/10.1016/j.ejor.2005.03.052)
18. De Koster R., van der Poort E. Routing Orderpickers in a Warehouse: A Comparison between Optimal and Heuristic Solutions // IIE Transactions. 1998. Vol. 30, no. 5. P. 469-480. DOI: [10.1080/07408179808966487](https://doi.org/10.1080/07408179808966487)
19. Roodbergen K.J., de Koster R. Routing Methods for Warehouses with Multiple Cross Aisles // International Journal of Production Research. 2001. Vol. 39, no. 9. P. 1865-1883. DOI: [10.1080/00207540110028128](https://doi.org/10.1080/00207540110028128)
20. Hall R.W. Distance Approximations for Routing Manual Pickers in a Warehouse // IIE Transactions. 1993. Vol. 25, no. 4. P. 76–87. DOI: [10.1080/07408179308964306](https://doi.org/10.1080/07408179308964306)
21. Hwang H., Oh Y.H., Lee Y.K. An Evaluation of Routing Policies for Order-Picking Operations in Low-Level Picker-to-Part System // International Journal of Production Research. 2004. Vol. 42, no. 18. P. 3873–3889. DOI: [10.1080/00207540410001696339](https://doi.org/10.1080/00207540410001696339)

22. Roodbergen K.J., Sharp G.P., Vis I.F. Designing the Layout Structure of Manual Order Picking Areas in Warehouses // IIE Transactions. 2008. Vol. 40, no. 11. P. 1032–1045. DOI: [10.1080/07408170802167639](https://doi.org/10.1080/07408170802167639)
23. Vaughan T.S. The Effect of Warehouse Cross Aisles on Order Picking Efficiency // International Journal of Production Research. 1999. Vol. 37, no. 4. P. 881–897. DOI: [10.1080/002075499191580](https://doi.org/10.1080/002075499191580)
24. Jarvis J.M., McDowell E.D. Optimal Product Layout in an Order Picking Warehouse // IIE Transactions. 1991. Vol. 23, no. 1. P. 93-102. DOI: [10.1080/07408179108963844](https://doi.org/10.1080/07408179108963844)
25. Gray A.E., Karmarkar U.S., Seidmann A. Design and Operation of an Order-Consolidation Warehouse: Models and Application // European Journal of Operational Research. 1992. Vol. 58, no. 1. P. 14–36. DOI: [10.1016/0377-2217\(92\)90232-X](https://doi.org/10.1016/0377-2217(92)90232-X)
26. Petersen II C.G. An Evaluation of Order Picking Routeing Policies // International Journal of Operations and Production Management. 1997. Vol. 17, no. 11. P. 1098–1111. DOI: [10.1108/01443579710177860](https://doi.org/10.1108/01443579710177860)

## **Warehouse order-picking process. Order-picker routing problem**

E.V. Korobkov<sup>1,\*</sup>

[\\*eug.korobkov@gmail.com](mailto:eug.korobkov@gmail.com)

<sup>1</sup>Bauman Moscow State Technical University, Moscow, Russia

---

**Keywords:** warehouse, order-picking, order, routing, order-picker, traveling salesman, heuristics

---

This article continues “Warehouse order-picking process” cycle and describes order-picker routing sub-problem of a warehouse order-picking process. It draws analogies between the order-pickers’ routing problem and traveling salesman’s problem, shows differences between the standard problem statement of a traveling salesman and routing problem of warehouse order-pickers, and gives the particular Steiner’s problem statement of a traveling salesman.

Warehouse layout with a typical order is represented by a graph, with some its vertices corresponding to mandatory order-picker’s visits and some other ones being noncompulsory. The paper describes an optimal Ratliff-Rosenthal algorithm to solve order-picker’s routing problem for the single-block warehouses, i.e. warehouses with only two crossing aisles, defines seven equivalent classes of partial routing sub-graphs and five transitions used to have an optimal routing sub-graph of a order-picker. An extension of optimal Ratliff-Rosenthal order-picker routing algorithm for multi-block warehouses is presented and also reasons for using the routing heuristics instead of exact optimal algorithms are given. The paper offers algorithmic description of the following seven routing heuristics: S-shaped, return, midpoint, largest gap, aisle-by-aisle, composite, and combined as well as modification of combined heuristics. The comparison of order-picker routing heuristics for one- and two-block warehouses is to be described in the next article of the “Warehouse order-picking process” cycle.

### **References**

1. Korobkov E.V. Warehouse Order-Picking Process. Review. *Nauka i obrazovanie MGTU im. N.E. Baumana = Science and Education of the Bauman MSTU*, 2015, no. 3, pp. 153-183. DOI: [10.7463/0315.0763161](https://doi.org/10.7463/0315.0763161) (in Russian).
2. Lawler E.L., Lenstra J.K., Kan A.R., Shmoys D.B. *The Travelling Salesman Problem: A Guided Tour of Combinational Optimization. Vol. 3*. New York, NY, Wiley, 1985. 476 p.
3. Ratliff H.D., Rosenthal A.S. Order-Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem. *Operations Research*, 1983, vol. 31, no. 3, pp. 507-521. DOI: [10.1287/opre.31.3.507](https://doi.org/10.1287/opre.31.3.507)

4. Roodbergen K.J., de Koster R. Routing Order Pickers in a Warehouse with a Middle Aisle. *European Journal of Operational Research*, 2001, vol. 133, no. 1, pp. 32-43. DOI: [10.1016/S0377-2217\(00\)00177-6](https://doi.org/10.1016/S0377-2217(00)00177-6)
5. Gademann N., Velde S. Order Batching to Minimize Total Travel Time in a Parallel-Aisle Warehouse. *IIE Transactions*, 2005, vol. 37, no. 1, pp. 63–75. DOI: [10.1080/07408170590516917](https://doi.org/10.1080/07408170590516917)
6. De Koster R., Roodbergen K.J., van Voorden R. Reduction of Walking Time in the Distribution Center of De Bijenkorf. In: Speranza M.G., ed. *New Trends in Distribution Logistics*. Berlin, Springer-Verlag, 2000, pp. 215-234. DOI: [10.1007/978-3-642-58568-5\\_11](https://doi.org/10.1007/978-3-642-58568-5_11)
7. Dekker R., de Koster R., Roodbergen K.J., van Kalleveen H. Improving Order-Picking Response Time at Ankor's Warehouse. *Interface*, 2004, vol. 34, no. 4, pp. 303–313. DOI: [10.1287/inte.1040.0083](https://doi.org/10.1287/inte.1040.0083)
8. Petersen II C.G., Schmenner R.W. An Evaluation of Routing and Volume-Based Storage Policies in an Order Picking Operation. *Decision Sciences*, 1999, vol. 30, no. 2, pp. 481–501. DOI: [10.1111/j.1540-5915.1999.tb01619.x](https://doi.org/10.1111/j.1540-5915.1999.tb01619.x)
9. Interactive Warehouse: web-site. Available at: <http://www.roodbergen.com/warehouse/index.php> , accessed January 2015.
10. Petersen II C.G., Aase G. A Comparison of Picking, Storage, and Routing Policies in Manual Order Picking. *International Journal of Production Economics*, 2004, vol. 92, no. 1, pp. 11-19. DOI: [10.1016/j.ijpe.2003.09.006](https://doi.org/10.1016/j.ijpe.2003.09.006)
11. De Koster R., Le-Duc T., Roodbergen K.J. Design and Control of Warehouse Order Picking: A Literature Review. *European Journal of Operational Research*, 2007, vol. 182, no. 2, pp. 481–501. DOI: [10.1016/j.ejor.2006.07.009](https://doi.org/10.1016/j.ejor.2006.07.009)
12. Gu J., Goetschalckx M., McGinnis L.F. Research on Warehouse Design and Performance Evaluation: A Comprehensive Review. *European Journal of Operational Research*, 2010, vol. 203, no. 3, pp. 539-549. DOI: [10.1016/j.ejor.2009.07.031](https://doi.org/10.1016/j.ejor.2009.07.031)
13. Le-Duc T., De Koster R. Travel Distance Estimation and Storage Zone Optimization in a 2-Block Class-Based Strategy Warehouse. *International Journal of Production Research*, 2005, vol. 43, no. 17, pp. 3561–3581. DOI: [10.1080/00207540500142894](https://doi.org/10.1080/00207540500142894)
14. Petersen II C.G. The Impact of Routing and Storage Policies on Warehouse Efficiency. *International Journal of Operations and Production Management*, 1999, vol. 19, no. 10, pp. 1053–1064. DOI: [10.1108/01443579910287073](https://doi.org/10.1108/01443579910287073)
15. Petersen II C.G., Aase G.R., Heiser D.R. Improving Order-Picking Performance through the Implementation of Class-Based Storage. *International Journal of Physical Distribution and Logistics Management*, 2004, vol. 34, no. 7, pp. 532-544. DOI: [10.1108/09600030410552230](https://doi.org/10.1108/09600030410552230)
16. De Koster R., van der Poort E.S., Wolters M. Efficient Orderbatching Methods in Warehouse. *International Journal of Production Research*, 1999, vol. 37, no. 7, pp. 1479–1504. DOI: [10.1080/002075499191094](https://doi.org/10.1080/002075499191094)

17. Le-Duc T., De Koster R. Travel Time Estimation and Order Batching in a 2-Block Warehouse. *European Journal of Operational Research*, 2007, vol. 176, no. 1, pp. 374–388. DOI: [10.1016/j.ejor.2005.03.052](https://doi.org/10.1016/j.ejor.2005.03.052)
18. De Koster R., van der Poort E. Routing Orderpickers in a Warehouse: A Comparison between Optimal and Heuristic Solutions. *IIE Transactions*, 1998, vol. 30, no. 5, pp. 469-480. DOI: [10.1080/07408179808966487](https://doi.org/10.1080/07408179808966487)
19. Roodbergen K.J., de Koster R. Routing Methods for Warehouses with Multiple Cross Aisles. *International Journal of Production Research*, 2001, vol. 39, no. 9, pp. 1865-1883. DOI: [10.1080/00207540110028128](https://doi.org/10.1080/00207540110028128)
20. Hall R.W. Distance Approximations for Routing Manual Pickers in a Warehouse. *IIE Transactions*, 1993, vol. 25, no. 4, pp. 76–87. DOI: [10.1080/07408179308964306](https://doi.org/10.1080/07408179308964306)
21. Hwang H., Oh Y.H., Lee Y.K. An Evaluation of Routing Policies for Order-Picking Operations in Low-Level Picker-to-Part System. *International Journal of Production Research*, 2004, vol. 42, no. 18, pp. 3873-3889. DOI: [10.1080/00207540410001696339](https://doi.org/10.1080/00207540410001696339)
22. Roodbergen K.J., Sharp G.P., Vis I.F. Designing the Layout Structure of Manual Order Picking Areas in Warehouses. *IIE Transactions*, 2008, vol. 40, no. 11, pp. 1032-1045. DOI: [10.1080/07408170802167639](https://doi.org/10.1080/07408170802167639)
23. Vaughan T.S. The Effect of Warehouse Cross Aisles on Order Picking Efficiency. *International Journal of Production Research*, 1999, vol. 37, no. 4, pp. 881-897. DOI: [10.1080/002075499191580](https://doi.org/10.1080/002075499191580)
24. Jarvis J.M., McDowell E.D. Optimal Product Layout in an Order Picking Warehouse. *IIE Transactions*, 1991, vol. 23, no. 1, pp. 93-102. DOI: [10.1080/07408179108963844](https://doi.org/10.1080/07408179108963844)
25. Gray A.E., Karmarkar U.S., Seidmann A. Design and Operation of an Order-Consolidation Warehouse: Models and Application. *European Journal of Operational Research*, 1992, vol. 58, no. 1, pp. 14-36. DOI: [10.1016/0377-2217\(92\)90232-X](https://doi.org/10.1016/0377-2217(92)90232-X)
26. Petersen II C.G. An Evaluation of Order Picking Routing Policies. *International Journal of Operations and Production Management*, 1997, vol. 17, no. 11, pp. 1098–1111. DOI: [10.1108/01443579710177860](https://doi.org/10.1108/01443579710177860)