



Ссылка на статью:

// Ученые записки УлГУ. Сер. Математика и информационные технологии. УлГУ. Электрон. журн. 2020, № 1, с. 124-132.

Поступила: 28.02.2020

Окончательный вариант: 17.05.2020

© УлГУ

УДК 519.178

Комбинаторные алгоритмы для построения многовариантных маршрутов доставки

Стремнев А.Ю.

nml2351@yandex.ru

БГТУ им. В.Г. Шухова, Белгород, Россия

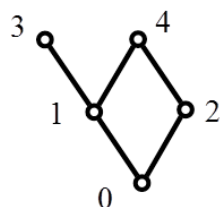
Аналитическое решение обобщенной задачи маршрутизации включает рассмотрение целого комплекса условий – от проверки наличия товаров (грузов) в точках отправления до минимизации затрат на перевозку. Немаловажным вопросом является также анализ структуры самой транспортной сети. В статье делается попытка построения алгоритма для нахождения всех допустимых маршрутов доставки в точку назначения. При этом будут рассматриваться комбинированные варианты и соответствующие ограничения, учитывающие отправку из одного или нескольких пунктов.

Ключевые слова: карта маршрутов, условия доставки, комбинации, рекурсия, алгоритмы.

Введение

Каждая конкретная задача маршрутизации имеет свои оптимизационные критерии и ограничения, учет которых может быть реализован посредством формализованных алгоритмов.

Допустим, что имеется заданное количество магазинов-складов с определенным запасом товаров на них (поз. 1-4, рис. 1). Есть также пункт-адресат для доставки товаров (поз. 0, рис. 1). Магазины-склады и пункт доставки связаны дорожной сетью. Причем сеть охватывает все магазины. Пункт доставки имеет определенную потребность в товаре, которую должны удовлетворить магазины-склады. Задача – определить маршрут доставки необходимого количества товара в пункт назначения. Естественно возникает условие минимизации затрат на доставку с учетом стоимости провоза товара по дорогам, связывающим пункты. Товар может быть доставлен из одного или сразу нескольких магазинов (в зависимости от наличия необходимого количества товара в них).



0 – пункт назначения

Рис. 1. Карта-схема транспортной сети.

Решение данной задачи удобно разделить на несколько этапов. Мы остановимся на рассмотрении одного из них, а именно поиске всех теоретически возможных маршрутов доставки.

1. Обзор топологии

Умозрительный перебор всех вариантов маршрутов даже весьма ограниченной по размерам схемы (см. рис. 1) указывает на необходимость автоматизации этого процесса. Во-первых, маршрутов очень много, а во-вторых, необходимо учитывать наличие товара на складах и стоимость провоза. Допустим, что нам удалось найти все возможные простые маршруты, связывающие каждый из магазинов-складов с конечной точкой доставки (рис. 2).

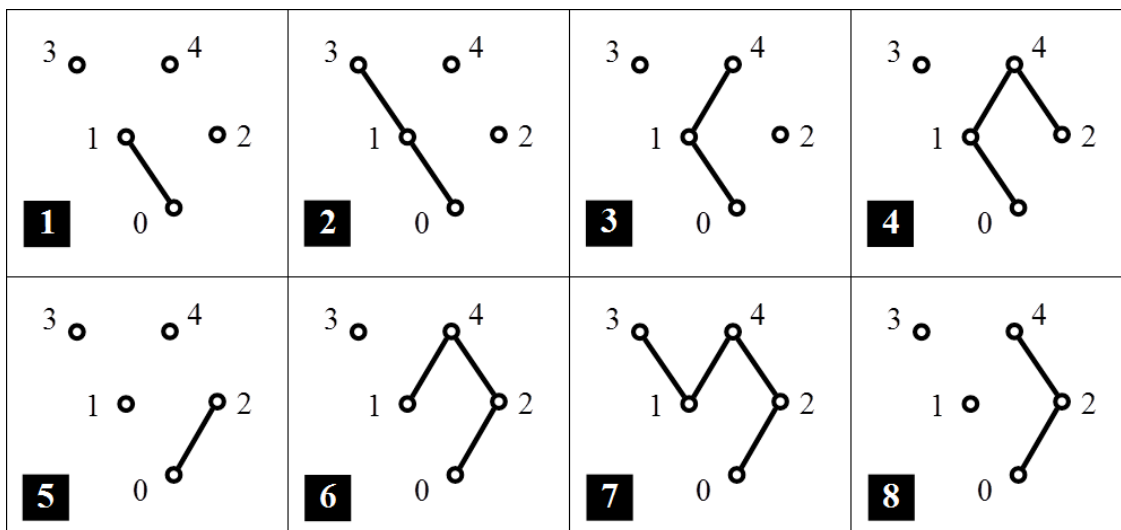


Рис. 2. Простые маршруты доставки в пункт назначения (точку 0).

Простой маршрут предполагает, что весь товар доставляется из одного склада-магазина, расположенного в начальной точке пути. Если предположить, что доставка производится независимо из нескольких магазинов, то простые маршруты (см. рис. 2) могут комбинироваться между собой, например, так как показано на рис. 3.

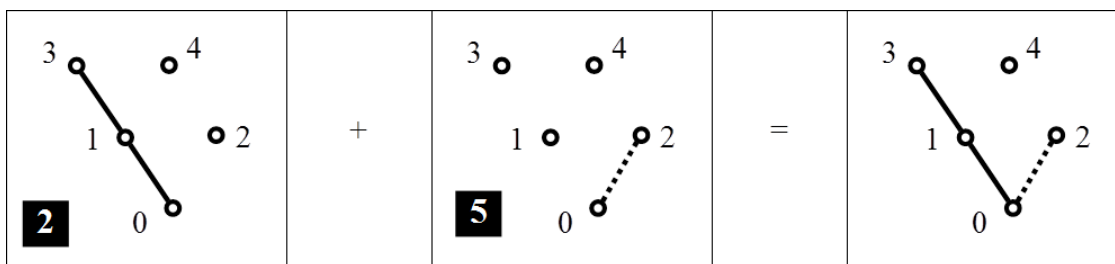


Рис. 3. Пример комбинирования простых маршрутов доставки.

Каждый маршрут (будь он простой или комбинированный) должен быть рациональным, т.е. не иметь "петель" и "разветвлений". Первое означает, что при движении не должно происходить возврата к уже пройденным точкам, а второе предполагает отправку товара из отдельного магазина в пункт назначения разными дорогами (по частям). "Петли" сами по себе означают лишние расходы на перевозку, а "разветвления" (рис. 4) противоречат условию оптимальности – действительно, самый дешевый путь от конкретного магазина к пункту назначения должен быть только один).

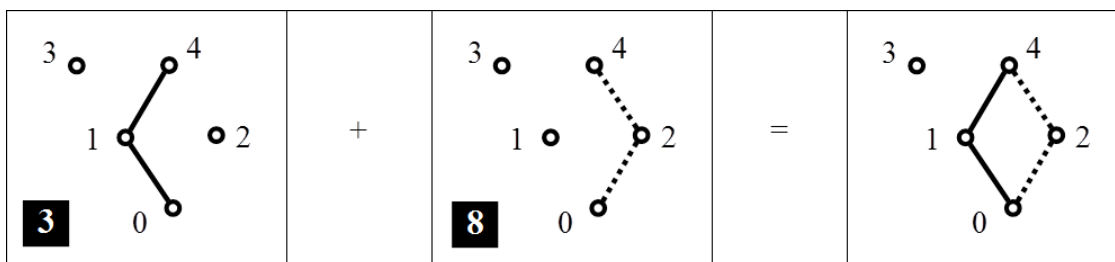


Рис. 4. Недопускаемое комбинирование маршрутов, имеющих общую точку отправления.

"Разветвление" на маршруте допускается только в том случае, если в "узловую" точку сходятся пути от нескольких разных магазинов-складов (рис. 5). В этом случае из "узловой" точки товар движется единым потоком.

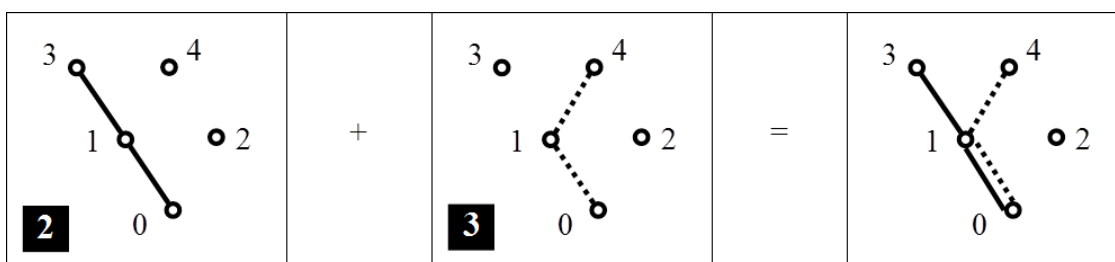


Рис. 5. Допускаемое комбинирование простых маршрутов с образованием единого потока к точке назначения.

Естественно, что для "попутной" доставки грузов комбинировать маршруты нет необходимости (рис. 6). Если склад, ближайший к точке доставки, лежит на траектории от более удаленного склада, то единственным рабочим будет считаться наиболее протяженный маршрут.

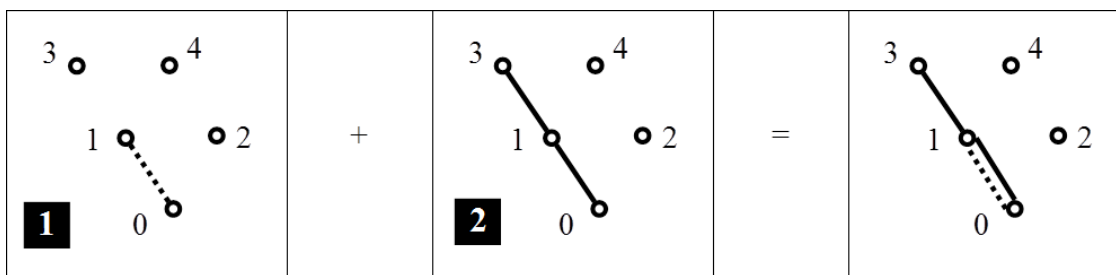


Рис. 6. Недопускаемое комбинирование при наложении маршрутов.

Таким образом, подзадача поиска всех возможных путей доставки в указанный пункт назначения сводится к нахождению всех "деревьев" связанного графа. В дополнение к простым маршрутам (см. рис. 2) требуется найти все комбинированные деревья (рис. 7).

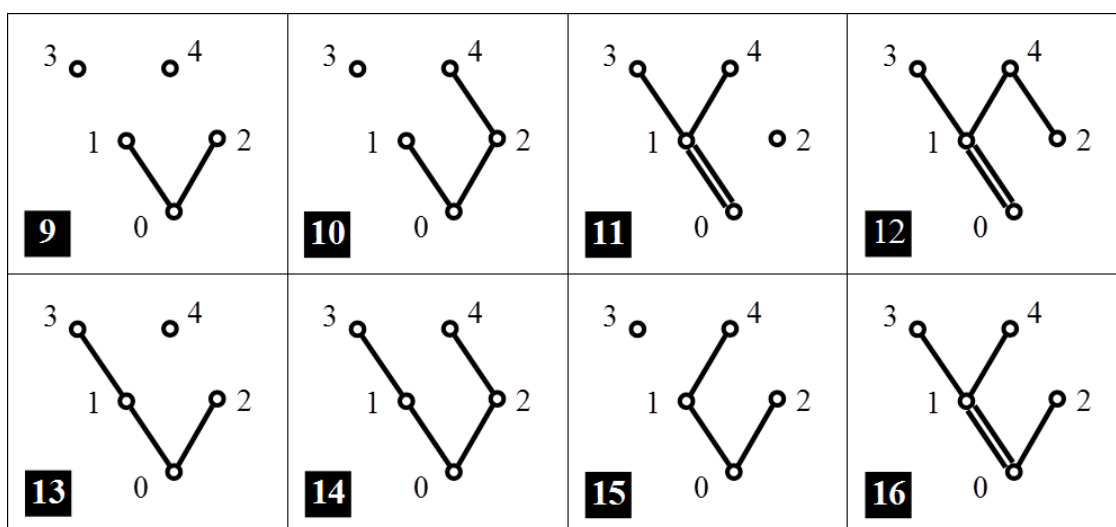


Рис. 7. Допускаемые комбинации простых маршрутов.

2. Построение алгоритма

На рис. 8 представлен обобщенный алгоритм получения всех допускаемых маршрутов. В блок-схеме используются следующие обозначения:

u, v – номера-идентификаторы точек схемы (0 – пункт-адресат);

$Edge$ – объект-ребро, дорога, соединяющая два соседних пункта u и v схемы;

$Edges[]$ – массив ребер-дорог, "карта" транспортной системы;

$LW[]$ – массив допускаемых простых и комбинированных маршрутов;

nsc – количество простых маршрутов;

nb – максимальное количество путей в комбинированном маршруте;

k – возможное количество простых маршрутов в комбинированном;

$c1[]$ – массив-комбинация порядковых номеров простых маршрутов;

$AddTF$ – логический признак добавления комбинации простых маршрутов в список допускаемых;

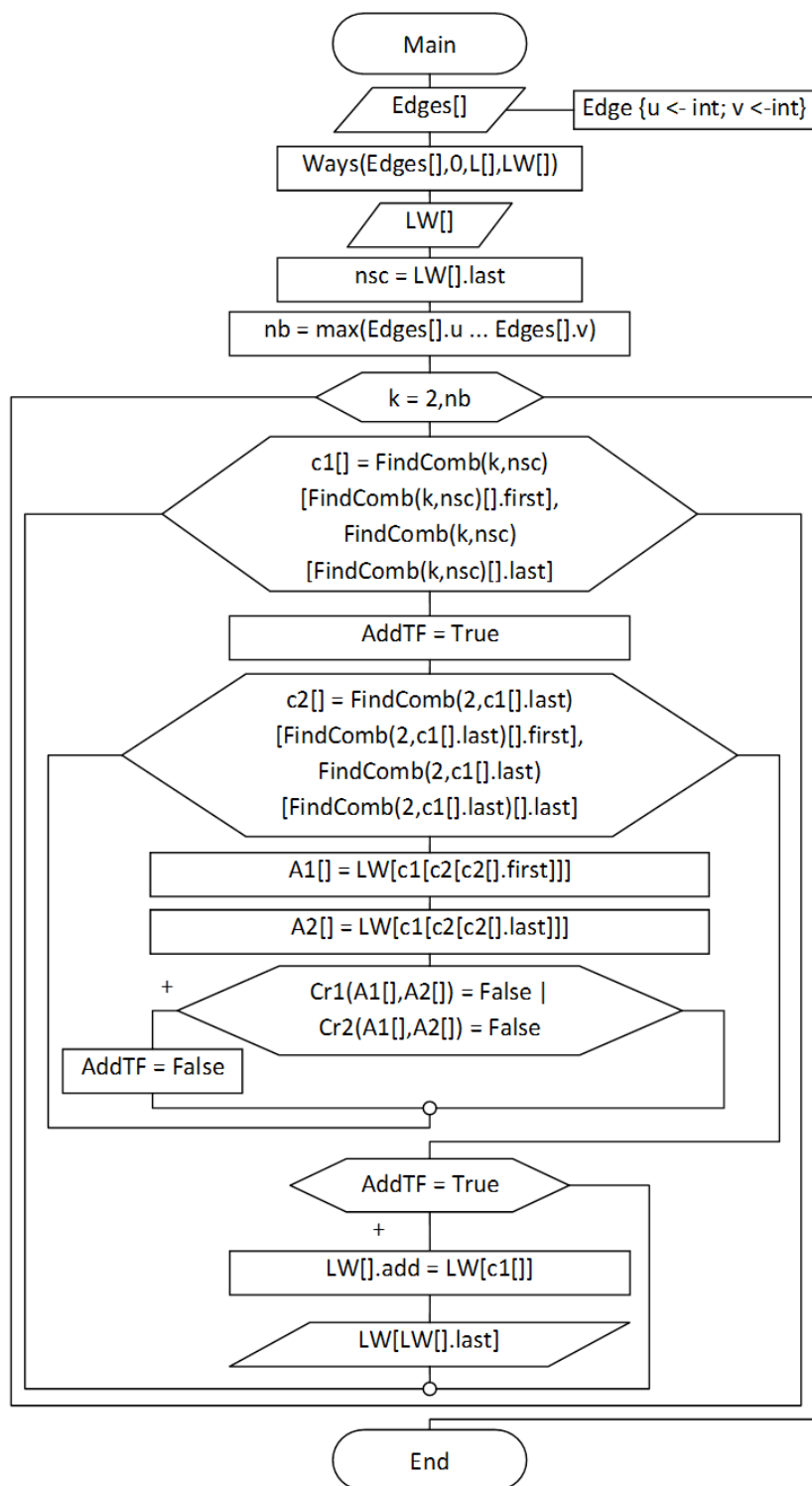


Рис. 8. Базовый алгоритм нахождения всех допустимых маршрутов доставки в пункт назначения.

$c2$ – массив из номеров двух простых маршрутов, входящих в комбинацию;

$A1[]$ – массив номеров-идентификаторов точек схемы, образующих первый простой маршрут в комбинации;

$A2[]$ – массив номеров-идентификаторов точек схемы, образующих второй простой маршрут в комбинации;

$Ways(Edges[], 0, \dots)$ – рекурсивный алгоритм нахождения всех простых маршрутов к точке 0 в схеме, заданной массивом ребер $Edges[]$;

$FindComb(p1, p2)$ – функция, возвращающая массив комбинаций из $p2$ по $p1$;

$Cr1(A1[], A2[])$ – функция проверки полного совпадения одного простого маршрута с участком другого;

$Cr2(A1[], A2[])$ – функция проверки пересечения простых маршрутов в комбинированном.

Алгоритм нахождения всех простых маршрутов к точке 0 ($Ways$) может быть построен, например, перебором всех связей-ребер [1]. Функция для генерирования всех комбинаций определенной длины из заданного количества уникальных значений ($FindComb$) может генерироваться, например, в лексикографическом порядке с использованием рекурсии или обычным итеративным алгоритмом [2].

Алгоритм для реализации функции ($Cr1$) проверки "полной включенности" одного простого маршрута в другой (см. рис. 6) приведен на рис. 9. Логика алгоритма заключается в том, что если хотя бы одна точка (начиная с точки 0) в проверяемых маршрутах не совпадает, то функция сигнализирует (значением $True$) о несовпадении маршрутов в комбинации.

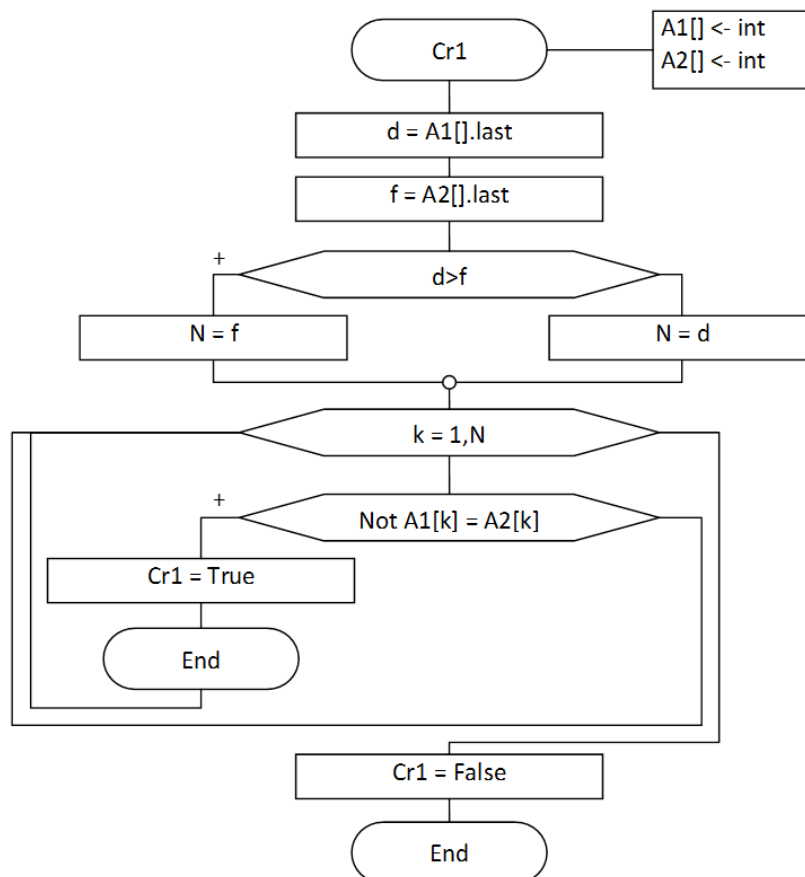


Рис. 9. Алгоритм функции проверки совпадения (наложения) двух заданных маршрутов.

Алгоритм функции ($Cr2$) проверки пересечения простых маршрутов (см. рис. 4) приведен на рис. 10.

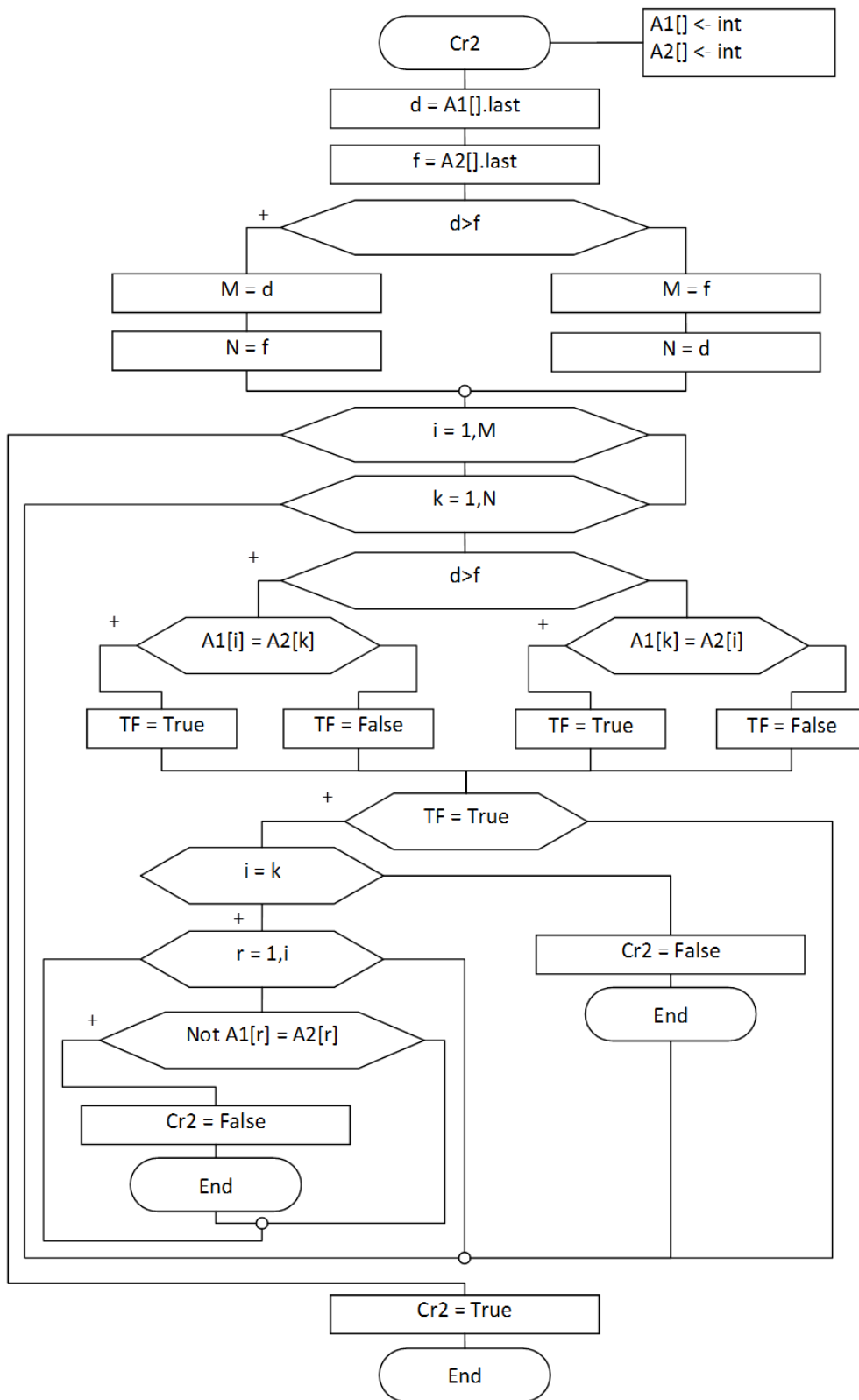


Рис. 10. Алгоритм функции проверки пересечения двух заданных маршрутов.

Алгоритм основан на том, что маршруты считаются пересекающимися (значение функции равно *False*) если у них есть общая точка (кроме точки назначения 0) и от этой точки до точки назначения маршруты не совпадают. Допустимость совмещения нескольких маршрутов в комбинации устанавливается по базовому алгоритму (см. рис. 8), если ни один из функциональных критериев (*Cr1* или *Cr2*) не равен *False*. Допускаемые совмещаемые маршруты (см. рис. 7) дополняют перечень простых (см. рис. 2). Например, в варианте № 16 (см. рис. 7) комбинируются три простых маршрута, причем ни один из них не включает в себя полностью другой и ни в одной из пар маршрутов не наблюдается пересечений (маршруты от складов 3 и 4 после точки 1 образуют единый поток доставки).

3. Пример реализации

На рис. 11 показан результат использования алгоритма поиска всех допускаемых маршрутов доставки грузов из магазинов-складов в пункт назначения (точку 0) в соответствии с заданной картой-схемой (см. рис. 1).

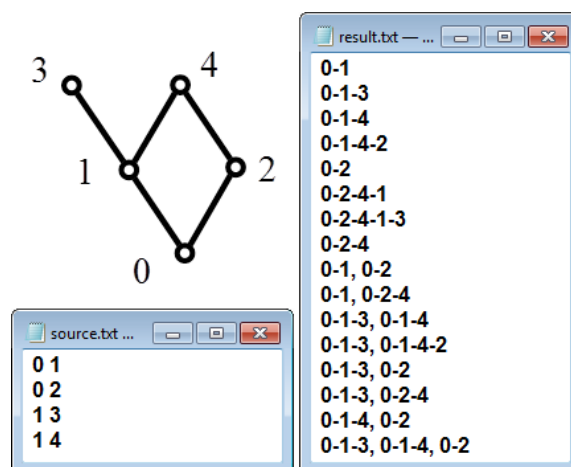


Рис. 11. Результат использования алгоритма нахождения всех допускаемых маршрутов доставки в пункт назначения (точку 0).

С исходным кодом проекта и готовым приложением, реализующим алгоритм поиска всех допускаемых вариантов доставки грузов в точку карты-схемы, можно ознакомиться по указанной ссылке [3], соответствующее веб-приложение доступно по ссылке [4].

Заключение

Подготовленный с помощью обобщенного алгоритма полный перечень простых и комбинированных путей может служить основой для дальнейшей аналитической обработки с учетом всех условий конкретной задачи маршрутизации.

Список литературы

1. Ковалев В.А. В. И. Мунерман. Использование многомерных матриц для решения задач нахождения путей в графе // *Системы компьютерной математики и их приложения (Смоленский государственный университет)*. 2016, № 17, с. 52-53.
2. Левин В.И. Непрерывно-логические алгоритмы решения комбинаторных задач / В.И. Левин // *Вестник Тамбовского государственного технического университета*. 2007, № 3, с. 687-698.
3. Стремнев А.Ю. Реализация алгоритма поиска всех допускаемых маршрутов доставки [Электронный ресурс]. Режим доступа: https://yadi.sk/d/OuI_OH538pzV9A (дата обращения 28.02.2020).
4. Стремнев А.Ю. Веб-приложение, реализующее алгоритм поиска всех допускаемых маршрутов доставки [Электронный ресурс]. Режим доступа: <http://t1.somee.com/> (дата обращения 17.05.2020).