

УДК 656.073.7

ОБ АЛГОРИТМЕ РЕШЕНИЯ ЗАДАЧИ РАЗВОЗКИ И ЕГО РЕАЛИЗАЦИИ
ABOUT THE ALGORITHM OF THE SOLUTION OF THE PROBLEM CARRIAGE
AND ITS REALIZATION

©Иванова А. А.

Южно-Российский государственный
политехнический университет им. М. И. Платова
г. Новочеркасск, Россия, anna17.5@yandex.ru

©Ivanova A.

Platov South Russian state polytechnic university
Novocherkassk, Russia, anna17.5@yandex.ru

©Черноморова Т. С.

Южно-Российский государственный
политехнический университет им. М. И. Платова
г. Новочеркасск, Россия, tatyana.chernomorova@gmail.com

©Chernomorova T.

Platov South Russian state polytechnic university
Novocherkassk, Russia, tatyana.chernomorova@gmail.com

Аннотация. Предложен вариант оригинального алгоритма решения задачи оптимизации планирования развозки груза, который позволит разрабатывать кольцевые маршруты движения транспорта. В результате маршрутизации перевозок достигается сокращение непроизводительных порожних пробегов, повышение качества обслуживания и сокращение транспортных издержек.

Abstract. A variant of the original algorithm for solving the problem of optimization of cargo delivery planning is proposed, which will allow to develop circular routes of transport. As a result of the routing of traffic, a reduction in unproductive empty runs is achieved, improving the quality of service and reducing transport costs.

Ключевые слова: алгоритм развозки, задача коммивояжера, метод ветвей и границ, кольцевые маршруты движения.

Keywords: carriage algorithm, traveling salesman problem, branch and border method, circular traffic routes.

При оперативном планировании грузоперевозок возникает задача развозки грузов, находящихся в пункте D_0 , в n различных пунктов D_1, D_2, \dots, D_n . При этом существует проблема, которая заключается в том, что транспорт отправляют в рейс не по оптимальному маршруту, используя не весь потенциал машины, а именно: транспорт движется с малой загруженностью, либо вообще без груза [1].

В настоящее время работы по тематике оптимизации развозки грузов предлагают решение таких задач, как: оптимизация расписания маршрута перевозки; определение перечня грузов, перевозимых каждым транспортным средством; контроль загруженности транспортного средства; синхронизация движения в зависимости от наличия «пробок» на отдельных участках трассы [2]. Проводятся исследования по управлению транспортными потоками и распределением грузов в системе [3].

При доставке груза потребителям компания разрабатывает маршруты движения. В результате маршрутизации перевозок достигается сокращение непроизводительных порожних пробегов, повышение качества обслуживания и сокращение транспортных издержек.

Маршруты движения могут быть маятниковые и кольцевые. Маятниковые маршруты — путь следования между двумя пунктами неоднократно повторяется. Кольцевые маршруты могут быть [1]:

- с обратным холостым пробегом;
- с обратным пробегом, не полностью загруженным;
- с обратным загруженным пробегом;
- кольцевой маршрут.

При разработке маршрутов за критерий оптимальности могут быть приняты транспортные затраты, себестоимость перевозок, коэффициент использования пробега автомобиля, общий пробег автомобиля и время доставки груза, своевременность доставки груза. В качестве общего критерия оптимальности принимается минимальный пробег транспортного средства [1].

В настоящей работе требуется оптимизировать планирование кольцевых маршрутов развозки груза от одного поставщика нескольким потребителям, переезжая из города в город с возможностью дозагрузки в них, чтобы обеспечить перевозку всех грузов за минимальное время. Схемы организации такого процесса представлены на Рисунке 1.

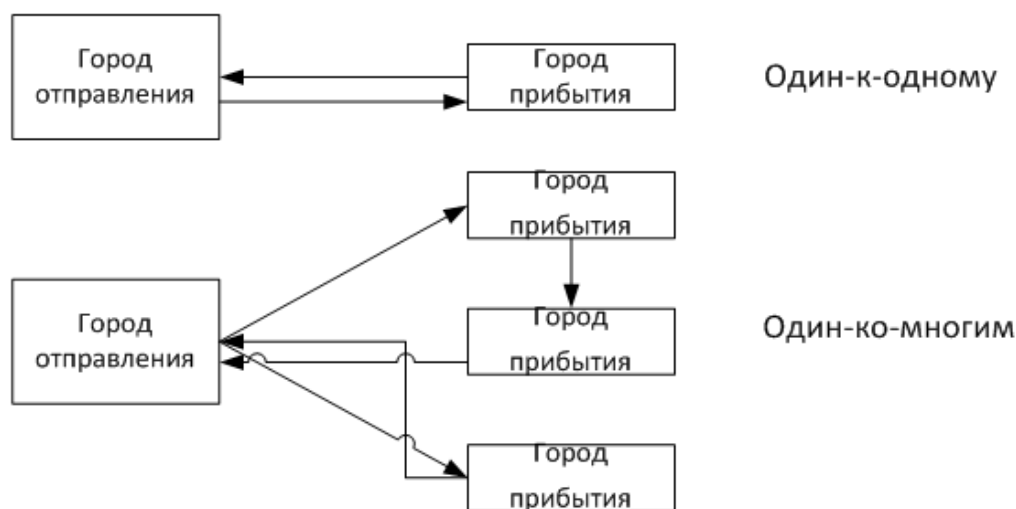


Рисунок 1. Схемы организации процесса развозки груза.

Постановка задачи развозки

Основной целью частной транспортной компании является получение максимальной прибыли от рейса. Заключив договор на доставку небольшого груза, диспетчер закрепляет его за транспортным средством (ТС) с самой меньшей грузоподъемностью. Если груз не заполнит весь объем грузового отсека, то можно будет осуществить дозагрузку в другом городе. Таким образом, увеличивается прибыль компании от этого рейса [4].

Задано множество перевозок A . Каждая перевозка p характеризуется городом назначения x_{n+p} ($p=1, \dots, n$), городом отправления x_p ($p=1, \dots, n$) и весом груза v_p . Известна матрица расстояний между городами x_i и x_j - $\{c_{ij}\}$. Если эти города в действительности

являются одним городом, то мы будем их считать разными, а $c_{ij} = 0$. В базовом городе x_0 находятся s ТС грузоподъемностью v_g ($g=1, \dots, s$). Под допустимым круговым маршрутом движения ТС s будем понимать перестановку $a(x_0, x_{i_1}^s, x_{i_2}^s, \dots, x_0)$, для которой выполняются следующие условия:

- 1) если ТС пришло в x_i , то оно берет груз q_i , если ТС пришло в x_{n+i} , то выгружает груз q_i ;
- 2) суммарный вес груза, перевозимого по любой дуге $(x_{i_s}, x_{i_{s+1}})$, не превышает грузоподъемности ТС v_g .

Требуется так построить круговые маршруты движения транспорта, чтобы минимизировать суммарную длину маршрутов. Следовательно, задача развозки груза сведена к задаче s коммивояжеров, в которой на посещение городов наложено условие частичной упорядоченности.

Алгоритм решения задачи развозки

При $s=1$ известна целочисленная постановка задачи коммивояжера [5]. Как известно, целочисленные методы являются неэффективными для решения задачи коммивояжера большой размерности. Самым эффективным методом решения этой задачи является метод ветвей и границ.

В работе предлагается оригинальный алгоритм заикливания метода определения оптимального маршрута для решения задачи развозки груза, который представлен на Рисунке 2.

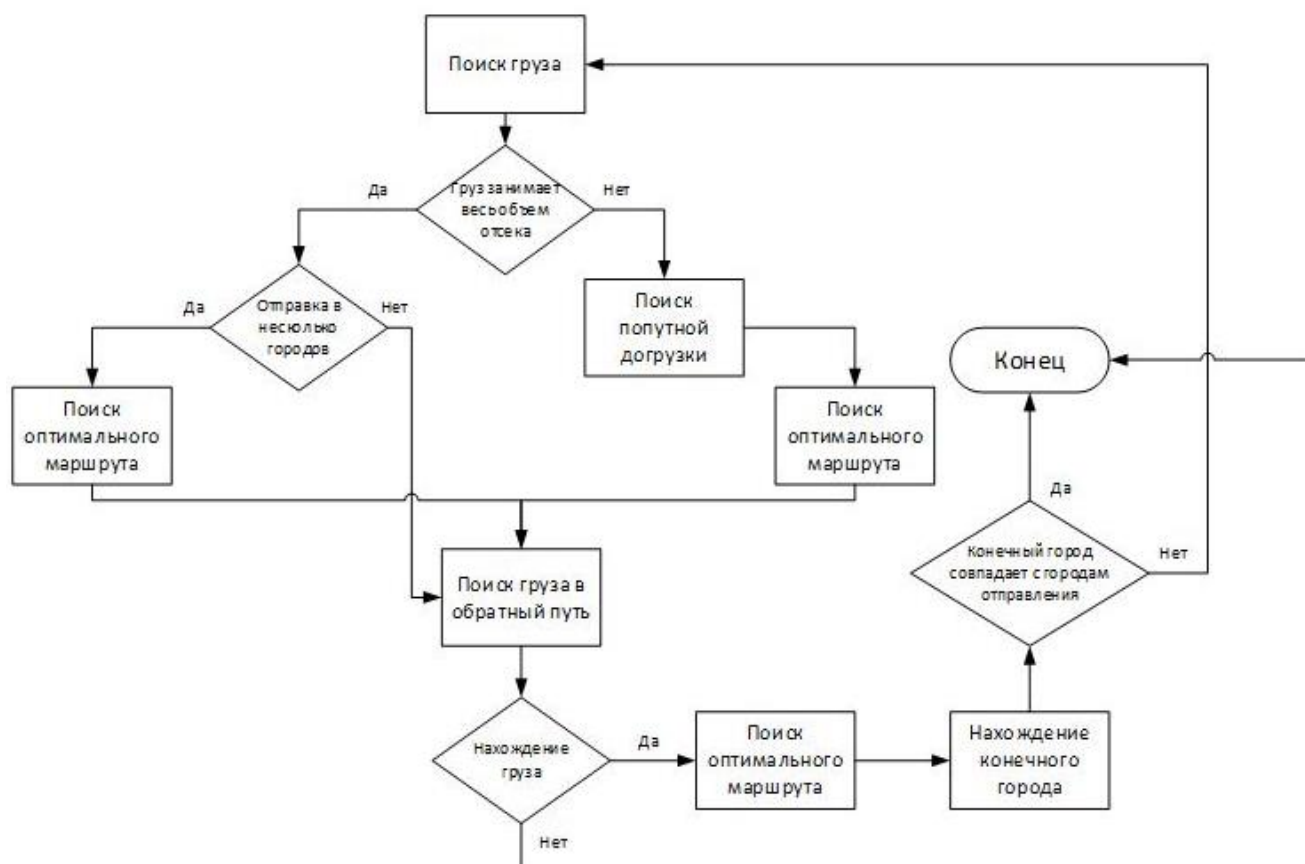


Рисунок 2. Алгоритм решения задачи развозки груза.

На начальном этапе необходимо найти оптимальный маршрут, учитывая то, что компания получает заказ на перевозку из одного города в другой с возможностью попутной дозагрузки машины, но самое главное — это определиться с городом, в котором имеется груз для доставки в обратном направлении или для продолжения движения по маршруту. Задачу следует решать в несколько этапов.

Транспорт отправляется из исходного города, где получает груз. Груз не обязательно займет весь отсек, поэтому транспорт можно догрузить в одном из попутных городов. В любом случае транспорт будет двигаться через определенное количество городов. Для начала, решая задачу коммивояжера, необходимо найти оптимальный маршрут из исходного города в города попутчики с потенциальной догрузкой. Таким образом, будет найден конечный город прибытия, в котором необходимо найти груз в обратную дорогу и вновь решить задачу коммивояжера.

Формализованное описание алгоритма решения задачи коммивояжера представлено в пособии [5]. Имеется n городов. Заданы расстояния между каждым городом $a_1, a_2, \dots, a_n (c_{ij}: a_i \rightarrow a_j)$, где $c_{ij} \neq c_{ji}$. Из города a_i надо объехать все города, не побывав ни в одном дважды и вернуться в исходный город. При этом суммарное расстояние должно быть минимальным.

Для решения задачи используется матрица, в которой строки и столбцы — это города, а на их пересечении (c_{ij}) — это расстояния между городами.

Например, груз из города $a_1 =$ Шахты надо развести в города: $a_2 =$ Ростов, $a_3 =$ Таганрог, $a_4 =$ Краснодар. Таким образом, размерность матрицы расстояний между городами равна 4×4 :

$$c_{ij} = \begin{pmatrix} \infty & 31 & 152 & 360 \\ 31 & \infty & 120 & 284 \\ 151 & 76 & \infty & 363 \\ 351 & 277 & 352 & \infty \end{pmatrix};$$

Далее формируется матрица c^0 , для этого из исходной матрицы вычитается матрица h_i , которая содержит минимальный элемент из каждой строки:

$$c^0 = \begin{pmatrix} \infty & 0 & 121 & 329 \\ 0 & \infty & 89 & 253 \\ 75 & 0 & \infty & 287 \\ 74 & 0 & 75 & \infty \end{pmatrix};$$

Аналогичным образом формируется матрица c^1 , для этого из полученной матрицы c^0 вычитается матрица h_j , которая содержит минимальный элемент из каждого столбца матрицы c^0 :

$$c^1 = \begin{pmatrix} \infty & 0 & 46 & 76 \\ 0 & \infty & 14 & 0 \\ 75 & 0 & \infty & 34 \\ 74 & 0 & 0 & \infty \end{pmatrix};$$

Находится нижняя оценка по формуле $\Xi(G_0) = \sum h_i + \sum h_j$.

Алгоритм реализован на языке программирования C# в вычислительной среде *Microsoft Visual Studio*. На Рисунке 3 демонстрируется скрипт функции *vichitaem()* — это процедура приведения, которая реализует формирование матриц c^0 и c^1 для определения нижней оценки.

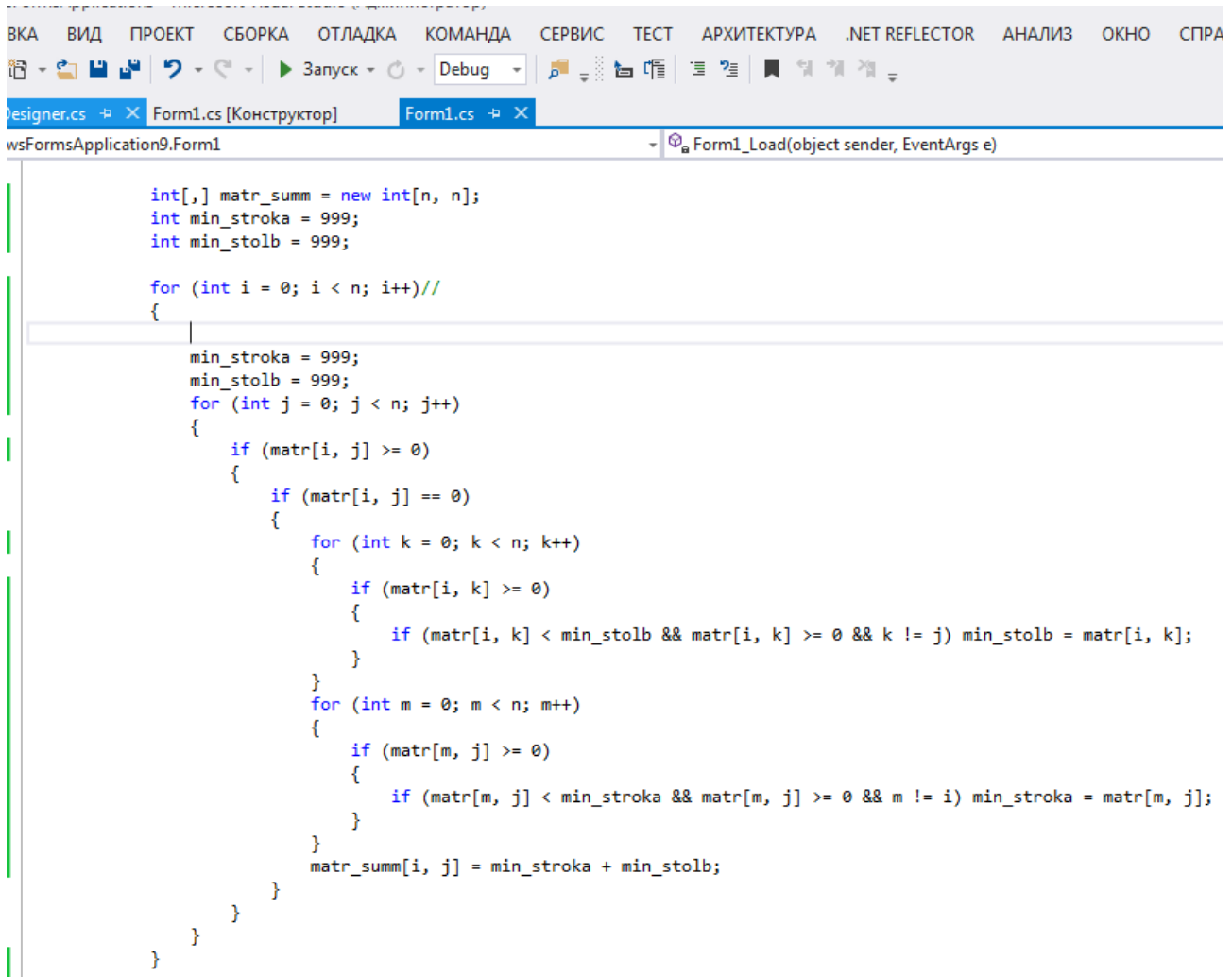
Затем определяется максимальная оценка перспективной пары (Рисунок 4). Оценка перспективной пары вычисляется по формуле:

$$\vartheta(p, q) = \min_{i \neq p} c_{ij}^0 + \min_{j \neq q} c_{ij}^0;$$

Полученные столбец и строка вычеркиваются, а в клетке (1,2) ставится ∞ , тем самым получаем матрицу $c_1^{(1)}$.

```
L.Designer.cs Form1.cs [Конструктор] Form1.cs [X]
JowsFormsApplication9.Form1 vichitaem(int[,] matr, int n1, out int G)
3
4 ссылка 3
5 public static int[,] vichitaem(int[,] matr, int n1, out int G)
6 {
7     int n = n1;
8     G = 0;
9     int[] matr_st = new int[n];
10    int[] matr_sto = new int[n];
11    int[,] matr2 = new int[n, n];
12
13    for (int i = 0; i < n; i++)//
14    {
15        matr_st[i] = 999;
16        matr_sto[i] = 999;
17    }
18
19    for (int i = 0; i < n; i++)//
20    {
21        for (int j = 0; j < n; j++)
22        {
23            if (matr[i, j] < matr_st[i] && matr[i, j] >= 0) matr_st[i] = matr[i, j];
24        }
25    }
26    for (int i = 0; i < n; i++)//
27    {
28        for (int j = 0; j < n; j++)
29        {
30            matr[i, j] = matr[i, j] - matr_st[i];
31        }
32    }
33
34    for (int j = 0; j < n; j++)
35    {
36        for (int i = 0; i < n; i++)
37        {
38            if (matr[i, j] < matr_sto[j] && matr[i, j] >= 0) matr_sto[j] = matr[i, j];
39        }
40    }
41
42    for (int j = 0; j < n; j++)
43    {
44        for (int i = 0; i < n; i++)
45        {
46            matr[i, j] = matr[i, j] - matr_sto[j];
47        }
48    }
49 }
```

Рисунок 3. Функция *vichitaem()*.



```
int[,] matr_summ = new int[n, n];
int min_stroka = 999;
int min_stolb = 999;

for (int i = 0; i < n; i++)//
{
    min_stroka = 999;
    min_stolb = 999;
    for (int j = 0; j < n; j++)
    {
        if (matr[i, j] >= 0)
        {
            if (matr[i, j] == 0)
            {
                for (int k = 0; k < n; k++)
                {
                    if (matr[i, k] >= 0)
                    {
                        if (matr[i, k] < min_stolb && matr[i, k] >= 0 && k != j) min_stolb = matr[i, k];
                    }
                }
                for (int m = 0; m < n; m++)
                {
                    if (matr[m, j] >= 0)
                    {
                        if (matr[m, j] < min_stroka && matr[m, j] >= 0 && m != i) min_stroka = matr[m, j];
                    }
                }
                matr_summ[i, j] = min_stroka + min_stolb;
            }
        }
    }
}
```

Рисунок 4. Вычисление перспективной пары.

Матрица $c_2^{(1)}$ определяется аналогичным образом из исходной матрицы c^1 исключением полученного маршрута (2,1) и в соответствующей клетке ставится ∞ .

После получения матриц $c_1^{(1)}$ и $c_2^{(1)}$ определяем их оценки по формуле:

$$\Xi(G_{1/2}^{(1)}) = \Xi(G_0) + \sum h_i + \sum h_j.$$

Нас интересует та матрица, для которой оценка минимальная. Итерационный процесс продолжается до тех пор, пока не будет найден оптимальный маршрут (Рисунок 5).

Оптимальный маршрут $1 \Rightarrow 2 \Rightarrow 4 \Rightarrow 3$ демонстрирует последовательность посещения городов в течение рейса, в процессе которого будет пройдено наименьшее расстояние, следовательно, затрачено минимум топлива и минимум времени. Экономия на издержках позволяет увеличить прибыль предприятия от данного рейса.

Найдя оптимальный маршрут следования из города в город, можно перейти к следующему этапу решения задачи развозки, учитывая вид дорог (одностороннее или двустороннее движение), пробки, остановки транспорта.

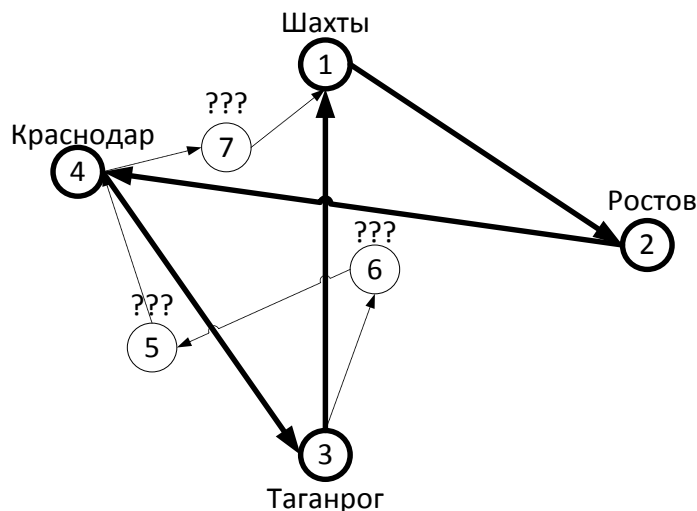


Рисунок 5. Оптимальный маршрут.

Разбив полученный маршрут на более мелкие маршруты, можно точнее решить задачу развозки. Такие подзадачи также относятся к задаче коммивояжера, но для решения можно использовать алгоритм Дейкстры. Суть алгоритма заключается в следующем: в ориентированной, неориентированной или смешанной сети найти кратчайший путь между двумя заданными городами.

Таким образом, решая новые подзадачи, можно учитывать маршрут следования транспорта по дорогам с односторонним движением и найти новый, более оптимальный маршрут.

В рамках оптимального маршрута (Рисунок 5) развозки груза из пункта 1 (город Шахты) в пункт 2 (город Ростов) можно проложить ряд новых маршрутов через город Новочеркасск или через станицу Грушевская с односторонним движением транспорта. Учитывая тот факт, что на трассе Шахты–Ростов движению транспорта часто препятствуют «пробки», можно разработать новые маршруты через эти пункты, получив в итоге более прибыльный рейс (Рисунок 6). Но решение этой задачи — тема для следующей статьи.

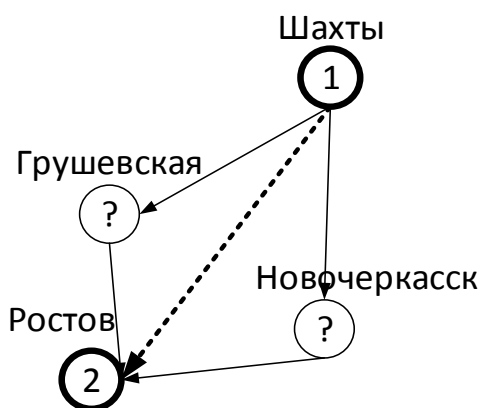


Рисунок 6. Нахождение более выгодного маршрута.

Заключение

Эффективность и простота используемого алгоритма (Рисунок 2) делает его отличным инструментом для решения задачи развозки. В результате выполнения программы на C# найден оптимальный маршрут $1 \Rightarrow 2 \Rightarrow 4 \Rightarrow 3$. Далее предстоит дорога домой — в город Шахты.

Определив конечный город, осуществив поиск обратного груза, заново решается задача коммивояжера. Только в данном случае задача будет нециклической, так как начальная и конечная точки маршрута фиксированы ($3 \Rightarrow 1$). При построении обратного маршрута могут появиться новые населенные пункты, но конечным останется тот, из которого первоначально состоялось отправление. Возможный новый маршрут изображен на Рисунке 5, а именно: $3 \Rightarrow 6 \Rightarrow 5 \Rightarrow 4 \Rightarrow 7 \Rightarrow 1$.

Предложенный оригинальный алгоритм позволит оптимизировать планирование развозки грузов путем разработки новых маршрутов движения транспортных средств.

В статье представлены отдельные скрипты оригинального алгоритма на языке высокого уровня C#, которые предстоит дополнить и улучшить на этапе выполнения выпускной квалификационной работы.

Список литературы:

1. Бауэрсокс Д. Д., Клосс Д. Д. Логистика. М.: ОЛИМП–БИЗНЕС, 2008. 636 с.
2. Блинов И. В. Моделирование процесса развозки однородного груза от однородного отправителя нескольким получателям: дис. ... канд. физ.-мат. наук. Воронеж, 2010. 182 с.
3. Лебедева О. А., Антонов Д. В. Международный опыт оптимизации грузовых перевозок в городской среде // Международный союз ученых «Наука. Технологии. Производство». 2015. №3 (7). Ч. 1. С. 21–23.
4. Бородин Д. В. Оптимизация и управление в системе транспортной логистики // Экономика и предпринимательство. 2014. №12–3 (53–3). С. 712–716.
5. Черноморов Г. А. Теория принятия решений: учебное пособие. 3-е изд. перераб. и доп. Новочеркасск: Изв. вузов. Электромеханика, 2005. 276 с.

References:

1. Bauersoks, D. Dzh., & Kloss, D. Dzh. (2008). Logistika. Moscow, OLIMP–BIZNES, 636.
2. Blinov, I. V. (2010). Modelirovanie protsessa razvozki odnorodnogo gruzha ot odnorodnogo otpravatelya neskolkim poluchatelyam. Dissert. kand. fiz-mat. nauk. Voronezhsk. gos. tekhn. akademiya, Voronezh, 182.
3. Lebedeva, O. A., & Antonov, D. V. (2015). Mezhdunarodnyi opyt optimizatsii gruzovykh perevozk v gorodskoi srede. *Mezhdunarodnyi soyuz uchennykh "Nauka. Tekhnologii. Proizvodstvo"*, (3–1), 21–23.
4. Borodin, D. V. (2014). Optimizatsiya i upravlenie v sisteme transportnoi logistiki. *Ekonomika i predprinimatelstvo*, (12–3), 712–716.
5. Chernomorov, G. A. (2005). Teoriya prinyatiya reshenii: uchebnoe posobie. 3-e izd. pererab. i dop. Novocherkassk, Izv. vuzov. Elektromekhanika, 276.

*Работа поступила
в редакцию 24.03.2017 г.*

*Принята к публикации
28.03.2017 г.*

Ссылка для цитирования:

Иванова А. А., Черноморова Т. С. Об алгоритме решения задачи развозки и его реализации // Бюллетень науки и практики. Электрон. журн. 2017. №4 (17). С. 107–114. Режим доступа: <http://www.bulletennauki.com/ivanova-chernomorova> (дата обращения 15.04.2017).

Cite as (APA):

Ivanova, A., & Chernomorova, T. (2017). About the algorithm of the solution of the problem carriage and its realization. *Bulletin of Science and Practice*, (4), 107–114.