



29. Биркгоф Г., Барти Т. Современная прикладная алгебра. М., 1976.  
30. Давыдова Е. М., Леденева Т. М., Мецераков Р. В., Подвальный С. Л. Дискретная математика : учебник. Томск, 2007.  
31. Журавлёв Ю. И., Флёров Ю. А., Вялый М. Н. Дискретный анализ. Формальные системы и алгоритмы : учеб. пособие. М., 2010.  
32. Мендельсон Э. Введение в математическую логику. М., 1976.  
33. Кэрролл Л. История с узелками. М., 2000.  
34. Carroll L. Symbolic Logic. URL: <http://www.twirpx.com/file/117987/> (дата обращения: 20.11.2016).

#### Об авторе

Сергей Валентинович Мациевский — канд. физ.-мат. наук, доц., Балтийский федеральный университет им. И. Канта, Калининград.  
E-mail: matsievsky@newmail.ru

#### About the author

Dr Sergey Matsievsky, ass. prof., I. Kant Baltic Federal University, Kaliningrad.  
E-mail: matsievsky@newmail.ru

УДК 51-77: 338.47

### В. В. Музычин

#### РЕШЕНИЕ ТРАНСПОРТНОЙ ЗАДАЧИ ПРИМЕНИТЕЛЬНО К ГРУЗОВЫМ ЖЕЛЕЗНОДОРОЖНЫМ ПЕРЕВОЗКАМ

*Сформулирована задача поиска оптимального маршрута, максимизирующего прибыль грузоперевозчика. Проведено сравнение нескольких методов решения этой задачи.*

*A task of searching of the optimal route maximizing the carrier profits is formulated. A comparison of several methods of this task solution is carried out.*

**Ключевые слова:** грузовая железнодорожная перевозка, комбинаторная оптимизация, база данных, граф железных дорог, математическое моделирование.

**Key words:** freight rail transportation, combinatorial optimization, database, railways graph, mathematical modeling.

#### Введение

При планировании маршрутов перевозки перед собственником вагонов стоит задача выбора потенциальных заказов на перевозку. В статье формулируется задача максимизации долгосрочной прибыли собственника вагона посредством прогнозирования маршрута на несколько этапов вперед и рассматриваются ее решения.



## 1. Постановка задачи поиска оптимального маршрута

Имеется связный граф железных дорог, заданный через список смежности. На каждой станции могут быть одна или несколько заявок на перевозку, характеризующиеся ценой, а также желаемой датой отправления груза. На одной из станций находится вагон. Необходимо так спланировать маршрут движения, чтобы максимизировать доход собственника вагона в долгосрочном периоде. Основное отличие от классической задачи коммивояжера в том, что нет необходимости перевезти все грузы и, более того, это может быть невозможно, так как у нескольких заявок может быть одна дата отправления.

В такой постановке задача называется по-английски prize-collecting traveling salesman problem with time windows (PC TSP WTW). Для решения PC TSP в работе [1] был предложен эвристический метод, в [2] было найдено точное решение методом ветвей и границ, а в [3; 4] предложено решение с использованием метода введения нормы на многограннике.

Решение TSP WTW было дано в [5].

Авторы статьи [6] свели PC TSP WTW к поиску наибольшей монотонной подпоследовательности.

## 2. Описание математической модели

Опишем исходные параметры модели.

Исходя из экспертных оценок, сделаем вывод, что максимальный радиус подбора не будет превышать 600 км. Это связано с тем, что согласно прејскуранту № 10-01 «Тарифы на перевозки грузов и услуги инфраструктуры, выполняемые российскими железными дорогами» тариф за порожний пробег на такие расстояния составляет 15 руб./км. При средней цене аренды полувагона в 2015 г. 500 руб./сутки подбор груза на большем расстоянии является экономически нецелесообразным. Из Правил исчисления сроков доставки грузов порожних грузовых вагонов железнодорожным транспортом от 7 августа 2015 г. № 245 следует, что порожний пробег на расстояние до 199 км происходит со скоростью 110 км/сутки, а на расстояние до 599 км — 160 км/сутки. Кроме того, согласно пункту 5.1 вышеуказанных правил требуется еще двое суток на операции, связанные с отправлением и прибытием груза порожних вагонов. Все эти ограничения нашли отражение в математической модели грузоперевозок.

Были сгенерированы несколько графов, моделирующих структуру железных дорог европейской (с высокой связностью вершин) и азиатской (с низкой связностью вершин) частей России. Также были созданы заявки на перевозку, содержащие следующую информацию:

- 1) сумма денежного вознаграждения;
- 2) срок перевозки груза в днях;
- 3) пункт отправления;
- 4) пункт прибытия.



Срок перевозки рассчитывается согласно правилам РЖД, а сумма вознаграждения — случайным образом в диапазоне сумм от 400 до 600 руб./сутки.

После этого помещаем вагон грузоперевозчика в случайно выбранную вершину графа и попытаемся максимизировать его прибыль, используя различные алгоритмы.

### 3. «Жадные» алгоритмы

40

Сначала рассмотрим алгоритм, дающий точное оптимальное решение, — полный перебор. Положим, у нас есть  $N$  заказов. На первом шаге мы можем выбрать перевозку по любой заявке, на следующем — по любой из  $N - 1$  оставшихся. Таким образом получаем экспоненциальную сложность, неприемлемую для реальных вычислений.

Поэтому имеет смысл искать приближенное решение. Отметим, что классические алгоритмы решения задачи коммивояжера (имитации отжига, Прайма — Эйлера, Литтла) нельзя адаптировать под нашу задачу по следующим причинам:

1) они дают ответ за приемлемое время только на малом количестве вершин;

2) их логика предусматривает обход всех вершин.

В нашем случае, если мы возьмем один заказ, то можем не успеть к другому. Иными словами, все вершины обойти может не получиться.

Первым из «жадных» алгоритмов рассмотрим метод ближайшего соседа (алгоритм 1). Суть его сводится к тому, что выполняется ближайшая к текущему положению вагона заявка. Он имеет линейную сложность.

Алгоритм 2 представляет собой модификации метода ближайшего соседа. Среди всех заявок, находящихся в заданном радиусе (радиусе подбора) от вагона, ищем ту, которая максимизирует прибыль собственника вагона и выражается формулой

$$\pi = TR - TC, \quad (1)$$

где  $\pi$  — прибыль собственника вагона;  $TR$  — выручка, полученная за исполнение заявки;  $TC$  — стоимость порожнего пробега.

Если в радиусе подбора заявок нет, применим алгоритм поиска ближайшего соседа.

Алгоритм 3 максимизирует среднедневную прибыль собственника вагона, рассчитываемую по формуле (1):

$$\bar{\pi} = \frac{TR - TC}{t_a - t_d + t_e}, \quad (2)$$

где  $\bar{\pi}$  — среднедневная прибыль при исполнении заявки;  $t_d$  — дата прибытия вагона на погрузку по заявке;  $t_a$  — дата освобождения вагона после разгрузки по заявке;  $t_e$  — дни, затраченные на порожний пробег.



Среди всех заявок, находящихся в радиусе подбора от вагона, ищем заявку с максимальной среднедневной прибылью. Если в радиусе подбора вагона заявок нет, применяем алгоритм 1.

#### 4. Алгоритмы, рассчитывающие на шаг вперед

Алгоритм 4 рассчитывает издержки на шаг вперед. Для всех заявок, находящихся в радиусе подбора вагона, выбираем их конечные пункты. Для каждого выбранного конечного пункта измеряем расстояние до ближайшей следующей заявки. Выбираем такой маршрут, чтобы суммарная длина двух порожних пробегов была минимальной.

Алгоритм 5 рассчитывает прибыль на шаг вперед. Для всех заявок, находящихся в радиусе подбора вагона, выбираем их конечные пункты. Для каждого выбранного конечного пункта рассчитываем потенциальную прибыль по формуле (1). Выбираем такой маршрут, чтобы суммарная прибыль двух поездов была максимальной. Если на одном из шагов в радиусе подбора заявок нет, выбираем ближайшую.

Алгоритм 6 рассчитывает среднедневную прибыль собственника вагонов на шаг вперед. Для всех заявок, находящихся в радиусе подбора вагона, выбираем их конечные пункты. Для каждого выбранного конечного пункта рассчитываем потенциальную прибыль по формуле (1). Выбираем такой маршрут, чтобы среднедневная прибыль двух поездов была максимальной. Если на одном из шагов в радиусе подбора заявок нет, выбираем ближайшую заявку.

#### 5. Алгоритмы с интегральной оценкой заявки

Алгоритмы 7–10 вводят интегральную оценку каждой заявке. В них в первую очередь выполняются заявки с максимальной оценкой среди заявок из радиуса подбора. Среди заявок с равной оценкой выбирается ближайшая. Если на одном из шагов в радиусе подбора заявок нет, выбираем ближайшую заявку.

Фактически мы по определенным критериям выделяем «хорошие» места на карте и стараемся двигать вагоны в них.

Характерный пример — производство пластмасс, при котором сырье завозится железнодорожным транспортом и таким же образом отгружается товар покупателям.

А с другой стороны — экспортный порт, в который нужно везти товар, но «выбраться» из него сложно.

Алгоритм 7 оценивает заявку по формуле

$$R = TR_s(k+1) + \sum TR(k \frac{r-d}{r} + 1). \quad (3)$$

Суммирование производится по всем заявкам, находящимся в радиусе подбора от станции назначения исходной заявки.



Здесь  $R$  — оценка исходной заявки;  $TR_s$  — выручка от исполнения исходной заявки;  $r$  — радиус подбора;  $d$  — расстояние от станции назначения исходной заявки до станции отправления суммируемой заявки;  $k$  — весовой коэффициент.

Алгоритм 8 усредняет оценку, полученную по формуле (3), рассчитывая ее по формуле

$$R = \begin{cases} TR_s(k+1) + \sum \frac{TR(k \frac{r-d}{r} + 1)}{N}, & N > 0, \\ 0, & N = 0, \end{cases}$$

42

где  $N$  — общее количество просуммированных заявок.

Суммирование производится по всем заявкам, находящимся в радиусе подбора от станции назначения исходной заявки.

Алгоритм 9 дополнительно учитывает стоимость порожнего пробега, рассчитывая ее по формуле

$$R = \begin{cases} \pi_s(k+1) + \sum \frac{\pi(k \frac{r-d}{r} + 1)}{N}, & N > 0, \\ 0, & N = 0, \end{cases}$$

где прибыль собственника вагона рассчитывается по формуле (1).

Суммирование производится по всем заявкам, находящимся в радиусе подбора от станции назначения исходной заявки.

Алгоритм 10 формирует оценку по среднедневной прибыли от заявок, находящихся в радиусе подбора от станции назначения исходной заявки, рассчитывая ее по формуле

$$R = \begin{cases} \bar{\pi}_s(k+1) + \sum \frac{\bar{\pi}(k \frac{r-d}{r} + 1)}{N}, & N > 0, \\ 0, & N = 0, \end{cases}$$

где  $\bar{\pi}$  — среднедневная прибыль, рассчитывается по формуле (2).

Суммирование производится по всем заявкам, находящимся в радиусе подбора от станции назначения исходной заявки.

## 6. Результаты численных экспериментов

### 6.1. Виды графов

Расчеты проводились на разных типах входных данных.

1. *Плотный граф* представляет собой сетку из 16 вершин с 34 ребрами. Здесь и далее расстояния между вершинами указаны в сотнях километров. Средняя степень вершин — 4,25. Его внешний вид, в сравнении с Московской железной дорогой (рис. 2), изображен на рисунке 1.

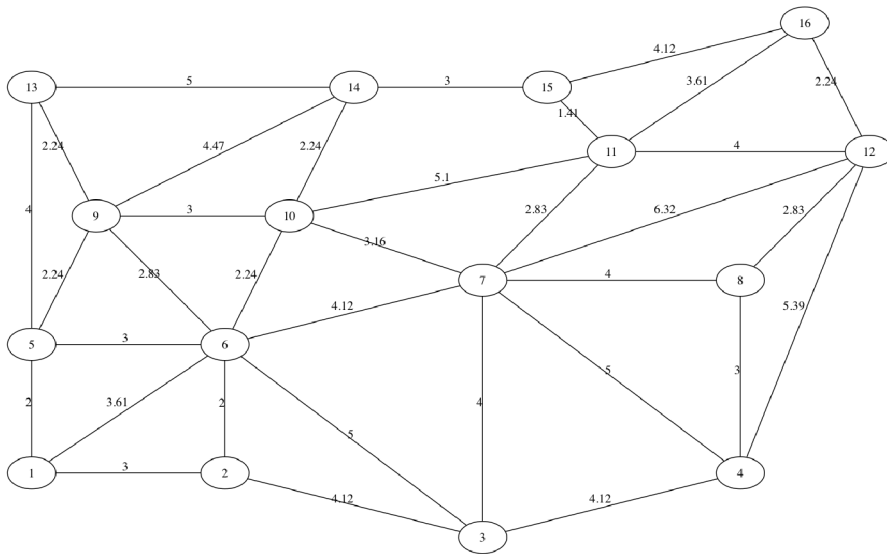


Рис. 1. Пример плотного графа



Рис. 2. Московская железная дорога

2. Граф средней плотности, изображенный на рисунке 3, является сеткой из 16 вершин, но с 17 ребрами. Средняя степень вершин – 2,125. Похожую структуру имеет Западно-Сибирская железная дорога.

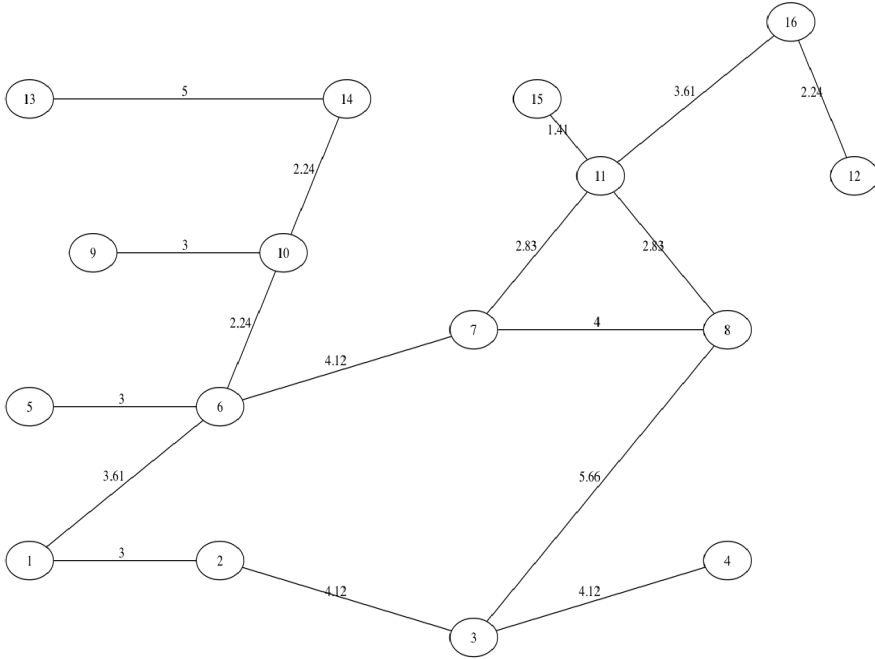


Рис. 3. Пример графа средней плотности

3. *Разреженный граф* представляет из себя сетку из 16 вершин, но уже с 15 ребрами (рис. 4). Средняя степень вершин – 1,875. Он моделирует Дальневосточную железную дорогу.

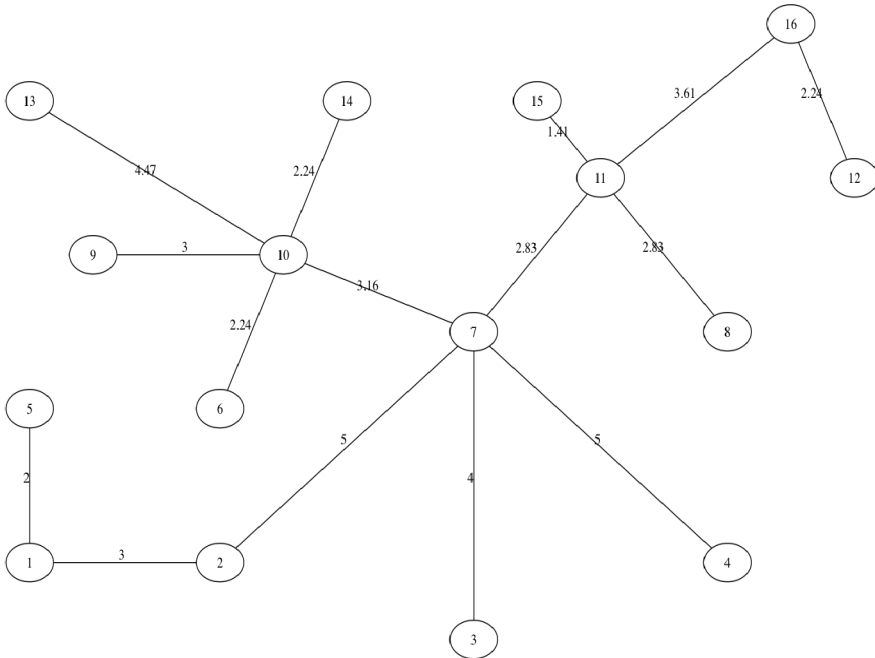


Рис. 4. Пример разреженного графа



## 6.2. Виды распределения заявок

В свою очередь, на каждом из типов графов были проведены тестирования с различным распределением заявок.

1. *Сильное распределение заявок* предполагает наличие 10 заявок в каждой из вершин (всего 160 заявок).

2. *Среднее распределение заявок* предполагает наличие 0–5 заявок в каждой из вершин (всего в среднем 30–50 заявок).

3. *Слабое распределение заявок* предполагает наличие 0–2 заявок в каждой из вершин (всего в среднем 10–20 заявок).

4. *Локальное распределение заявок* предполагает наличие в графе со слабым распределением подграфов с сильным распределением заявок, а также подграфов с полным отсутствием заявок (всего в среднем 35–55 заявок).

45

## 6.3. Сравнение алгоритмов

В качестве подробных числовых расчетов приведем сравнение эффективности вышеописанных 10 алгоритмов на графе средней плотности и радиусе подбора 600 км. Эти числовые расчеты сведены в таблицу 1.

Таблица 1

**Сравнение эффективности алгоритмов  
на графе средней плотности и радиусе подбора 600 км,  
максимальная эффективность – 100 % (знак % опущен)**

Номер алгоритма и его характеристика	Дни					
	10	30	50	70	80	90
1	100	87,52	89,56	88,33	89,29	89,26
2	75,3	80,13	75,04	82,25	78,81	81,08
3	100	78,23	91,9	93,19	89,07	89,75
4	63,25	81,16	100	99,99	100	100
5	74,96	79,82	91,08	88,12	90,25	90,09
6	96,98	100	93,52	95,18	96,06	91,63
7. $K = 0,1$	51,33	57,59	75,32	75,58	78,58	80,5
7. $K = 1$	74,96	79,82	75,25	82,39	78,85	81,24
7. $K = 10$	74,96	77,29	81,96	85,75	82,33	84,62
8. $K = 0,1$	50,97	56,88	68,01	76,41	73,09	75,94
8. $K = 1$	50,97	57,59	60,21	70,14	67,77	70,94
8. $K = 10$	75,3	79,09	74,85	81,67	78,28	80,45
9. $K = 0,1$	74,96	78,78	83,09	93,42	88,9	90,72
9. $K = 1$	74,96	78,78	82,97	93,2	88,63	90,16
9. $K = 10$	74,96	78,78	83	88,54	84,41	86,31
10. $K = 0,1$	100	90,79	91,26	100	99,07	98,62
10. $K = 1$	100	89,5	98,3	98,47	98,8	98,82
10. $K = 10$	45,59	86,02	87,5	94,81	94,84	94,21





#### 6.4. Описание результатов

В плотном графе наиболее быстро и с приемлемым качеством оценки работают алгоритмы 1–3. Это связано с тем, что эти алгоритмы не зависят от выбора радиуса, а при большом объеме заказов и их равномерном распределении естественным образом минимизируется количество порожних перегонов и, следовательно, максимизируется прибыль.

На всех типах распределения заказов были выявлены следующие закономерности.

При  $R = 6$  наилучшие результаты показывают алгоритмы 4–5, а также интегральная оценка алгоритма 8.

При увеличении радиуса до 12 качество оценки вышеупомянутых алгоритмов сохраняется, в то же время качество оценки алгоритмов 9 и 10 падает.

При  $R \geq 18$  алгоритмы 4–5 и 8 сохраняют стабильность, в то время как качество оценки алгоритмов 9 и 10 по-прежнему снижается.

Экспериментальным образом выявлено, что алгоритмы 7–10 предоставляют наилучшую оценку при  $K = 1$ .

Есть основания полагать, что зависимость качества интегральной оценки от радиуса обусловлена топологией графа. В самом деле, при сильной удаленности узлов друг от друга выбор радиуса малой величины не имеет смысла, в то время как выбор слишком большого радиуса грозит полным перебором заявок и может сильно ухудшить время работы алгоритмов 4–10.

В графе средней плотности при сильном распределении заказов помимо ожидаемого успешного результата алгоритмов 1–3 один из лучших результатов демонстрирует алгоритм 10.

В графе средней плотности при среднем распределении заказов и  $R = 6$  алгоритмы 4–5 показывают наилучший результат. Увеличение радиуса до 12 не улучшает качества алгоритма 10, как это было в случае плотного графа. Это обусловлено уменьшением количества вершин и увеличением расстояний между ними. Данное утверждение также подтверждается тем фактом, что при  $R \geq 18$  результаты работы алгоритма 10 улучшаются.

В графе средней плотности при слабом распределении заказов, а также при локальном распределении и малых радиусах ( $R < 18$ ) не удается однозначно определить алгоритм, демонстрирующий лучшие результаты. При  $R \geq 18$  алгоритмы 4–6 и алгоритмы 7–10 незначительно преобладают в качестве предоставляемых результатов.

Тестирование алгоритмов на разреженном графе при различных распределениях заказов и радиусах приводит к аналогичным результатам.

#### Выводы

Для графов любой плотности при сильном распределении заказов наиболее оптимально использовать группу алгоритмов 4–6. Они дают приемлемый результат и имеют высокую скорость работы.



Для графов высокой плотности при среднем, слабом и локальном распределении заказов наилучшие результаты показывают алгоритмы 4–6, а также интегральные алгоритмы 8–10.

Алгоритмы 4–6 более стабильны и слабо зависят от изменения радиуса подбора  $R$ .

Алгоритмы 7–10 менее стабильны и более зависимы от топологии графа и распределения заявок, а также от радиуса подбора  $R$ , но в некоторых случаях дают лучший результат.

Слишком малые радиусы неэффективны для алгоритмов 4–10 на графах средней плотности и разреженных графах.

Слишком большие радиусы замедляют работу алгоритмов 4–10.

В группе «жадных» алгоритмов 1–3 наилучшие результаты показывает алгоритм 3, учитывающий при расчетах среднедневную прибыль.

В группе алгоритмов 4–6 наилучшие результаты показывают алгоритм 4, основанный на минимизации издержек, связанных с порожним пробегом, а также алгоритм 5, учитывающий максимально возможную прибыль.

#### Список литературы

1. Balas E., Martin G. ROLL-A-ROUND: Software Package for Scheduling the Rounds of a Rolling Mill. Pittsburgh, PA, 1986.
2. Fischetti M., Toth P. An additive approach for the optimal solution of the prize-collecting travelling salesman problem // Vehicle Routing: Methods and Studies. North-Holland, 1988. P. 319–343.
3. Balas E. The prize collecting traveling salesman problem // Networks. 1989. Vol. 19. P. 621–636.
4. Balas E. The prize collecting traveling salesman problem: II. Polyhedral results // Ibid. 1995. Vol. 25. P. 199–216.
5. Focacci F., Lodi A., Milano M. A hybrid exact algorithm for the TSPTW // INFORMS Journal on Computing. 2002. № 14 (4). P. 403–417.
6. Bar-Yehuda R., Evenb G., Shahar S.(M.) On Approximating a Geometric Prize-Collecting Traveling Salesman Problem with Time Windows // Journal of Algorithms. 2005. April. P. 76–93.

#### Об авторе

Владимир Витальевич Музычин — асп., Балтийский федеральный университет им. И. Канта, Калининград.

E-mail: mv@1c39.ru

#### About the author

Vladimir Muzychin, PhD student, I. Kant Baltic Federal University, Kaliningrad.

E-mail: mv@1c39.ru