

ГРАФОВАЯ МОДЕЛЬ РАСПРЕДЕЛЕНИЯ ЛОКОМОТИВОВ ДЛЯ ГРУЗОВЫХ ПЕРЕВОЗОК НА ЛИНЕЙНОМ УЧАСТКЕ ЖЕЛЕЗНОЙ ДОРОГИ. ЗАДАЧА О МАКСИМАЛЬНОМ ПО ВКЛЮЧЕНИЮ ПОКРЫТИИ ГРАФИКА¹

Л.Ю. Жиликова, Н.А. Кузнецов, В.Г. Матюхин, А.Б. Шабунин, А.К. Такмазьян

Работа посвящена формальной постановке и решению задачи, возникающей при распределении локомотивов между составами для осуществления грузовых железнодорожных перевозок с заранее заданным графиком. Цель состоит в том, чтобы определить, достаточно ли число локомотивов при заданном их начальном расположении для выполнения всех перевозок. Решение представлено в виде алгоритма, строящего покрытие графика: полное, если оно существует, или, в противном случае, частичное — максимальное по включению. Доказана теорема о взаимно однозначном соответствии существования полного покрытия и достаточности числа локомотивов.

Ключевые слова: графовая модель, потоки в сетях, назначение локомотивов, грузовые железнодорожные перевозки.

ВВЕДЕНИЕ

Задача составления расписания для грузовых железнодорожных перевозок имеет много различных аспектов и, несмотря на ряд существующих успешных моделей и методов поиска решений, в силу многих специфических особенностей в разных предметных областях остается весьма актуальной. Общий подход к созданию интеллектуальной мультиагентной системы управления перевозочным процессом на железнодорожном транспорте изложен в работах [1, 2]. Описано составление оптимального расписания движения поездов между двумя станциями, соединенными однопутной железной дорогой с разъездом [3, 4]. Рассмотрена задача формирования грузовых составов из имеющихся на станциях заказов и определения маршрутов и расписания их следования по железнодорожной сети так, чтобы минимизировать суммар-

ное взвешенное время выполнения заказов [5]. Задачи назначения локомотивов грузовым составам, оптимизирующие один или несколько критериев, с множеством разнообразных ограничений на функционирование, физические параметры и бюджет, составляют отдельный большой класс задач составления расписаний. Обзор и классификация моделей назначения локомотивов приведены в работе [6]; различные модели и алгоритмы назначения локомотивов даны в работах [7–9].

При описании и формализации предметной области очень важен выбор языка и формального аппарата для моделирования. Для решения задач на двудольных графах применяется многоагентный подход [10, 11]. Предложен метод аукционов, где, помимо многоагентного подхода, применяются стохастические и генетические алгоритмы [11]. В этой же работе впервые сформулирована идея эвристического разбиения временного горизонта планирования на атомарные части, в пределах которых можно предположить соответствие один к одному (локомотив — поезд) и искать глобально-оптимальные (в сумме по атомарному интервалу)

¹ Работа выполнена при частичной финансовой поддержке РФФИ (проект 17-20-01180).

паросочетания методом задачи о назначениях. В результате формируется «временной фронт назначений» распространяющийся по всему горизонту планирования. В работе [12] проблема назначения локомотивов грузовым составам решается поиском локально-оптимальных паросочетаний между двумя списками (локомотивов и поездов) на каждой станции. При этом минимизируется число локомотивов; для поиска решения используются предложенные авторами эвристики. В работе [13] представлен алгоритм решения задачи о назначении локомотивов, основанный на разработанных алгоритмах формирования множества максимальных (по включению) путей ориентированного графа и покрытия вершин ориентированного графа построенным множеством максимальных путей.

В работах [14, 15] решается задача оптимального подбора тяговых ресурсов для грузовых железнодорожных перевозок. Авторами построена модель, цель которой заключается в том, чтобы для заданного графика, состоящего из времен отправления составов с начальных станций и их прибытия на конечные станции (определяемых скоростями движения), а также начального множества локомотивов, задать оптимальный план перевозок — назначить каждому составу локомотив таким образом, чтобы были осуществлены все перевозки и непроизводительное время локомотивов было минимальным. В процессе построения модели на графической (пространственно-временной) плоскости строится альтернативный граф [16], в котором, кроме маршрутов составов, присутствуют также виртуальные ребра нескольких типов: ребра подцепки, перецепки и терминальные ребра. Графы, содержащие ребра разных типов, не все из которых доступны, рассматриваются в моделях графов с нестандартной достижимостью [17, 18]. Потоки, процессы рассеяния, а также сами графовые характеристики (такие, как кратчайшие пути) в этих графах отличаются от обычных. Соответственно, подвергаются модификации и стандартные алгоритмы для их поиска [19, 20]. С этой особенностью сталкивается и описанная далее модель.

Задача, которая будет решаться в настоящей статье, состоит в том, чтобы до поиска оптимального плана назначений, описанного в работе [14, 15], определить, является ли график перевозок выполнимым при заданном числе и начальном расположении локомотивов, и если он невыполним, найти выполнимый подграфик максимальной мощности. Используется идея построения и распространения временного фронта вдоль горизонта планирования, сходная с описанной в работе [11]. Однако помимо отличий в подходах и методах описания сдвига фронта, применяемых в предлагаемой работе, главная ее отличительная черта состоит в смещении цели моделирования. Вместо задачи по-

иска оптимального назначения локомотивов решается более частная проблема: определение максимального подграфика перевозок, для которого заданное множество локомотивов при заданном их начальном расположении является достаточным.

1. ОПИСАНИЕ МОДЕЛИ И ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Рассмотрим участок железной дороги с k станциями.

Множество станций $S = \{S_0, \dots, S_k\}$ линейно упорядоченное.

В модели вводится дискретное время $T = \{1, 2, \dots, T_{fin}\}$. Горизонт планирования T_{fin} составляет, как правило, 24–48 ч.

Множество составов и их траекторий движения $Tr = \{Tr_1, \dots, Tr_n\}$.

Множество локомотивов $L = \{L_1, \dots, L_m\}$.

Начальное положение составов и локомотивов задается отображениями:

$$f_{Tr_start}: Tr \rightarrow (T, S); \quad f_{L_start}: L \rightarrow (T, S).$$

Целевое состояние для составов задается отображением:

$$f_{Tr_goal}: Tr \rightarrow (T, S).$$

Задача перевозки формулируется в терминах пространственно-временного графика.

На графической плоскости (пространственно-временном графике) задан план перевозок грузовых составов на отрезке $[0, T_{fin}]$. На этой же плоскости задано множество локомотивов (рис. 1). Задача состоит в том, чтобы каждому составу назначить, как минимум, один локомотив для его перевозки. Локомотивы могут двигаться порожняком от станции прибытия предыдущего состава к станции отправления следующего. Оптимизация построенных назначений в настоящую задачу не включается.

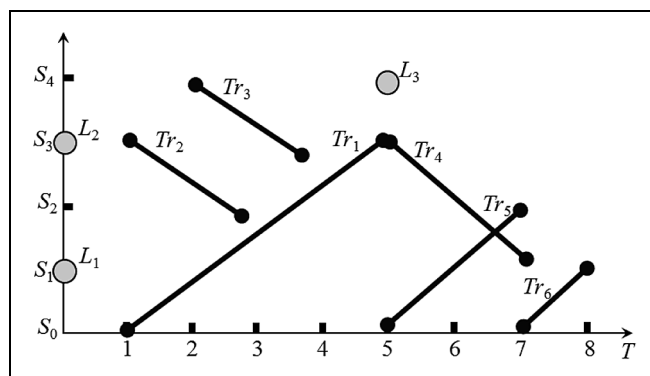


Рис. 1. Графическая плоскость с заданными перевозками и локомотивами

1.1. Допущения модели

При построении модели примем ряд упрощений.

- Локомотивы характеризуются двумя атрибутами: *допустимой областью тяги* (плечо, домен локомотива) и *грузоподъемностью*. В первой версии модели будем полагать, что все локомотивы и составы находятся в общей области тяги и что грузоподъемности любого локомотива достаточно для любого состава.
- На протяжении всего пути состав не меняет локомотив, т. е. сохраняется соответствие (Tr_i, L_j) .
- Скорости всех локомотивов при осуществлении перевозки составов одинаковы, не зависят от характеристик составов, постоянны и равны v .
- Скорости локомотивов и составов v задаются таким образом, что в них учитывается время подцепки и перецепки локомотива к составу. Таким образом, перецепка локомотива может быть осуществлена в одной точке графической плоскости (t, S_j) . На рис. 1 такая перецепка локомотива возможна между локо-слотами Tr_1 и Tr_4 в момент $t = 5$.
- Поезда не конкурируют при отправлении со станции. В каждый момент времени с одной станции в одном направлении отправляется не более одного поезда. Время между отправлениями поездов всегда не менее заданного интервала безопасности.
- Поезда не конкурируют за пути: все перевозки, представленные на графической плоскости, выполнимы.
- Станции расположены линейно в порядке возрастания номеров. Таким образом:
 - граф железнодорожной сети является простым путем;
 - маршрут между любыми двумя станциями определен однозначно;
 - для расстояний между станциями выполняется равенство

$$|S_{i-1}, S_{i+1}| = |S_{i-1}, S_i| + |S_i, S_{i+1}|.$$

1.2. Основные определения

Определение 1. Отрезки, задающие траектории движения составов на графической плоскости, будем называть *локо-слотами*. ♦

Это название означает, что каждый отрезок Tr_i представляет собой потенциальный слот для подцепки локомотива.

Все локо-слоты имеют одинаковые углы наклона, тангенсы которых равны v и $-v$, где $v = \text{const}$ — скорости локомотивов.

Определение 2. *Графиком G* называется объединение всех локо-слотов на графической плоскости. ♦

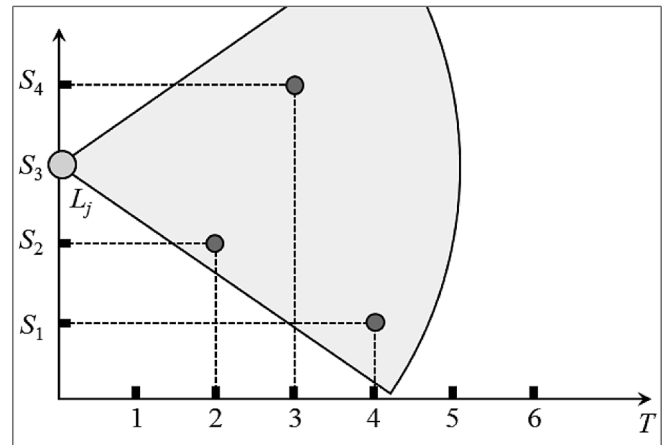


Рис. 2. Конус скорости локомотива L_j

Если локомотива, способного быть подцепленным к локо-слоту Tr_i , не существует, перевозка не может быть выполнена. В этом случае не может быть выполнен и весь график. В данной версии модели будем считать, что график фиксирован и локо-слоты не сдвигаются.

Произвольному локо-слоту Tr_i можно поставить в соответствие все локомотивы, для которых его станция отправления S_{p_i} достижима в момент времени t_i , где $f_{Tr_start}(Tr_i) = (t_i, S_{p_i})$. Определим, какие локо-слоты будут доступны для произвольного локомотива L_j , такого, что $f_{L_start}(L_j) = (t_j, S_{p_j})$.

Определение 3. *Конусом скорости локомотива $Cone(L_j)$* называется пространственно-временной конус на графической плоскости с вершиной в точке $f_{L_start}(L_j) = (t_j, S_{p_j})$, образованный двумя прямыми, тангенс угла наклона которых равен v и $-v$, соответственно. ♦

Для каждого локомотива все доступные локо-слоты находятся в конусе скорости (рис. 2). Заметим, что углы наклонов локо-слотов и направляющих конуса скорости локомотива одинаковы, поэтому, если начало некоторого локо-слота лежит в данном конусе скорости, то в этом конусе лежит весь локо-слот целиком.

Из рис. 2 видно, что конус скорости единственным образом задает времена, в которые данный локомотив может гарантированно оказаться на указанной станции. Так, он может заполнить слоты, начинающиеся в станции S_2 не ранее $t = 2$, S_4 не ранее $t = 3$ и S_1 не ранее $t = 4$.

Определение 4. Все возможные виртуальные ребра, ведущие из локомотивов в начальном состоянии к начальным вершинам локо-слотов, лежащим в их конусах скорости, будем называть *ребрами подцепки*. ♦

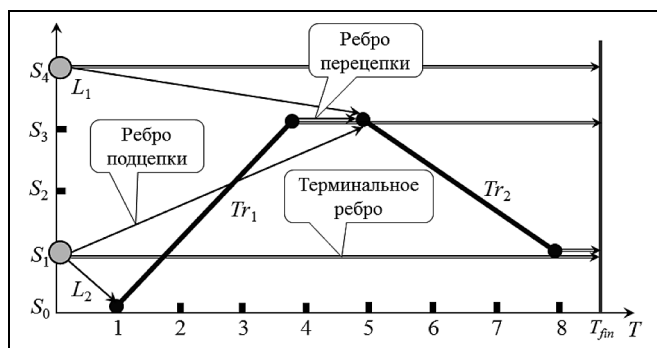


Рис. 3. Ребра подцепки и перецепки и терминальные ребра

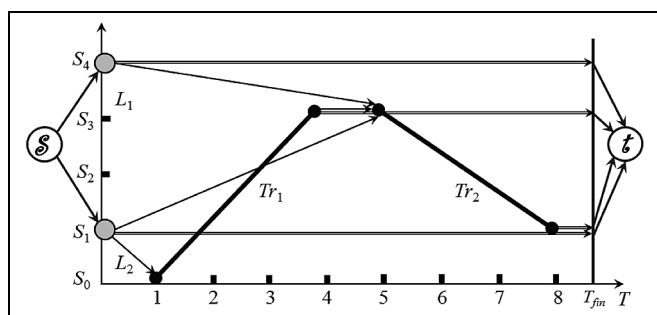


Рис. 4. Источник и сток

За время, составляющее горизонт планирования, как правило, каждый локомотив перевозит более одного состава, поэтому, кроме ребер подцепки локомотивов, существуют также *ребра перецепки*.

Определение 5. Все возможные виртуальные ребра, ведущие из концов одних локо-слотов в начала других, лежащие в их конусах скорости, будем называть *ребрами перецепки*. ♦

Каждый локомотив, закончив свою работу на промежутке времени $[0, T_{fin}]$, движется во времени к горизонту планирования, обозначенному на графике вертикальной линией. Все ребра, ведущие к этой линии от локомотивов и перпендикулярные ей, будем называть *терминальными*. Локомотивы могут переходить в состояние T_{fin} из своего начального состояния (если они не участвовали в перевозках) и из всех концов локо-слотов (рис. 3).

На графике (см. рис. 3) присутствуют четыре терминальных ребра: по одному из начальных позиций локомотивов и из концов двух локо-слотов.

Замечание 1. Тангенсы углов наклона ребер подцепки и перецепки могут принимать любые значения из диапазона $[-v, v]$. На рисунках эти ребра представлены отрезками, в которых средняя скорость локомотива может быть по абсолютной величине меньше v . Такие отрезки — упрощение модели. Они заменяют ломаные линии, состоя-

щие из горизонтальных участков (соответствующих простому локомотивам) и из участков с наклоном, равным $\pm v$, соответствующих движению. ♦

Введем две фиктивные вершины: источник S , из которого выходят связи ко всем локомотивам, и сток t , к которому ведут связи от всех окончаний терминальных ребер (рис. 4).

Таким образом, в графе существуют четыре типа ребер, три из которых виртуальные. Ребра подцепки и перецепки и терминальные ребра будем называть *альтернативными ребрами графа*.

Определение 6. *Простым путем* $Path_i$ называется путь из источника до стока, содержащий альтернативные и терминальные ребра только для локомотива L_i (число локо-слотов в простом пути произвольно). Простой путь, состоящий только из терминального ребра, будем называть *тривиальным*.

Определение 7. *Покрытием* C графика G на альтернативном графе называется такое объединение простых путей $C = \bigcup_{i=1}^m Path_i$, что:

- они попарно не пересекаются $Path_i \cap Path_j = \emptyset \forall i, j \in \overline{1, m}$;
- их объединение содержит весь график G .

Определение 8. График называется *выполнимым*, если для данного множества и начального положения локомотивов существует его покрытие.

Определение 9. Если для данного графика покрытия не существует, график называется *невыполнимым*, а число локомотивов при данном их начальном размещении называется *недостаточным*. ♦

Если число локомотивов недостаточно, будем рассматривать *частичное покрытие* C_{part} графика G , удовлетворяющее первому условию определения покрытия.

Определение 10. Частичное покрытие графика C_{part} будем называть *максимальным по включению* (*максимальным*), если для данного начального расположения локомотивов *не существует* покрытия C_{part}^* , состоящего из таких простых путей, что

$$Path_j \subseteq Path_j^*, j \in \overline{1, m}, \text{ причем, хотя бы один из них содержит одноименный путь из покрытия } C_{part} \text{ в качестве собственного подмножества: } \exists Path_i^* : Path_i \subset Path_i^* .$$

2. СООТНОШЕНИЕ ЧИСЛА ЛОКОМОТИВОВ И ЛОКО-СЛОТОВ

Сформулируем критерий существования полного покрытия графика в терминах числа локомотивов и их начального расположения. Если локомотивов недостаточно для перевозки всех со-



ставов (заполнения всех локо-слотов), то должно быть найдено максимальное по включению покрытие. Если число локомотивов избыточно, возникает проблема отсеечения решений, при которых задействуются лишние локомотивы. Эта проблема сводится к задаче редукции множества локомотивов до минимально достаточного. Редуктировать множество локомотивов можно уточнением весов альтернативных ребер — этому будут посвящены наши дальнейшие исследования. В настоящей работе сконцентрируемся на задаче определения достаточности локомотивов и поиске максимального частичного покрытия, если локомотивов недостаточно. Здесь и далее под «максимальным» частичным покрытием будем понимать покрытие, максимальное по включению (см. определение 10).

Понятие недостаточности (как и понятие избыточности) неформально вполне очевидно. Предпримем попытку его формализации.

Пример. Для графика на рис. 5 заведомо не существует покрытия.

В ситуации, представленной на рис. 5, существуют три частичных покрытия, каждое из которых соответствует отдельному терминальному ребру. Два из них — нетривиальны. Ясно, что оптимальным из этих двух частичных покрытий является покрытие, включающее локо-слот Tr_1 : поскольку станция S_3 , начальная для локомотива, ближе к станции S_4 , чем к станции S_1 (откуда начинаются локо-слоты Tr_1 и Tr_2 соответственно), а прочие условия для двух подцепок одинаковы. ♦

Для того чтобы узнать, достаточно ли число локомотивов для данного графика, воспользовавшись определением, необходимо среди всех возможных комбинаций простых путей найти ту, которая является покрытием графика. Это трудоемкая задача. Если локомотивов недостаточно, требуется полный перебор, чтобы в этом убедиться. Желательно сформулировать более легко проверяемый критерий достаточности локомотивов. Для этого сначала введем дополнительные определения.

Определение 11. Проекцию локо-слота Tr_i на ось времени будем называть *интервалом движе-*

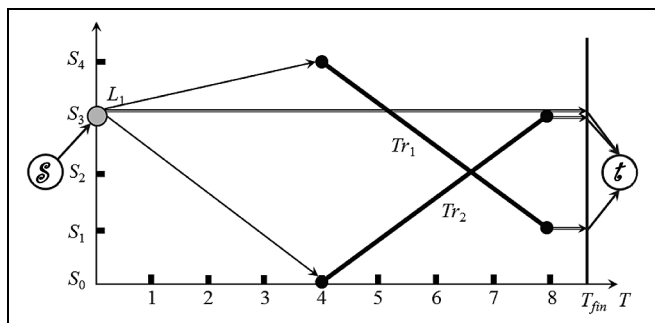


Рис. 5. Два локо-слота с пересекающимися проекциями на шкалу времени и один локомотив

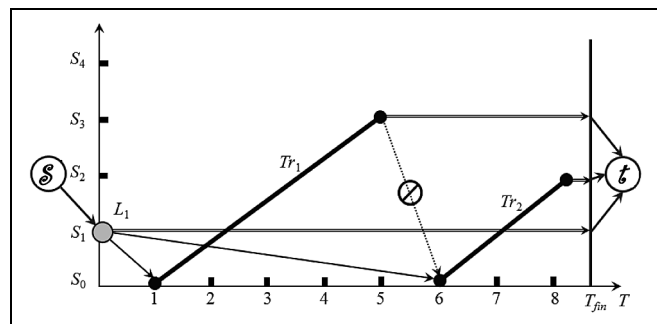


Рис. 6. Недостаточность локомотивов при непересекающихся локо-слотах

ния состава Tr_i и обозначать как I_{Tr_i} : $I_{Tr_i} = [t(f_{Tr_start}(Tr_i)), t(f_{Tr_goal}(Tr_i))]$.

Определение 12. Пересечением локо-слотов будем называть пересечение их интервалов движения: $I_{Tr_i} \cap I_{Tr_j}$.

Определение 13. Мощностью пересечения локо-слотов p_I внутри некоторого интервала $I = [t_1, t_2]$ будем называть максимальное число локо-слотов, пересечение которых непусто и принадлежит интервалу I . ♦

Так, на всем горизонте планирования для графика, представленного на рис. 5, мощность пересечения $p_I = 0$, для графика, представленного на рис. 6, $p_I = 2$.

Утверждение 1 (необходимое условие достаточности локомотивов). Для того чтобы график, представленный объединением локо-слотов, был выполнен, необходимо, чтобы мощность пересечения локо-слотов на интервале горизонта планирования $I = [0, T_{fin}]$ не превосходила числа локомотивов.

Доказательство. Для существования покрытия графика на интервале максимального пересечения локо-слотов необходимо, чтобы каждому из пересекающихся локо-слотов соответствовал свой локомотив. Если локомотивов меньше, это означает, что покрытия не существует. Тогда локомотивов недостаточно по определению. ♦

Условие утверждения 1 является необходимым для выполнимости графика, однако далеко не всегда достаточным. В нем описывается ситуация, когда несколько поездов должны двигаться одновременно. Тогда, если число локомотивов меньше числа движущихся поездов, их заведомо не хватает.

Вторая ситуация нехватки заключается в том, что интервал между прибытием поезда Tr_i на станцию S_j и отравлением поезда Tr_k со станции S_j меньше, чем время перегона локомотива от станции S_j до станции S_i (плюс время необходимое на отцепку-прицепку локомотива с поездом) (см. рис. 6).

В графике на рис. 6 для локомотива L_1 ребро перецепки, обозначенное пунктиром, не является доступным, поскольку начало локо-слота Tr_2 лежит вне конуса скорости с вершиной в окончании локо-слота Tr_1 .

3. АЛГОРИТМ СДВИГА ФРОНТА

3.1. Описание алгоритма

Для любого фиксированного графика зададим рекурсивный алгоритм построения покрытия, который назовем *алгоритмом сдвига фронта*. Алгоритм, который определяет, достаточно ли число локомотивов, должен строить покрытие всегда, когда оно существует. Если покрытие может быть построено, локомотивов достаточно.

В графиках примем такую нумерацию локо-слотов, что $i < j$ для Tr_i и Tr_j , если и только если $t(f_{Tr_{start}}(Tr_i)) \leq t(f_{Tr_{start}}(Tr_j))$. Таким образом, локо-слоты упорядочены по времени отправки поездов.

Определение 14. *Фронт* — это виртуальная ломаная линия, соединяющая начала l локо-слотов с самыми маленькими номерами, достижимых из точек старта l локомотивов, $l \leq m$, где $m = |L|$ — число локомотивов. ♦

Иными словами, фронт — это те выбранные локо-слоты, каждому из которых к моменту его начала можно поставить во взаимно однозначное соответствие доступный локомотив.

Фронт распространяется слева направо таким образом, чтобы по возможности ни один локо-слот не был пропущен. Точки старта локомотивов могут быть распределены не только в пространстве, но и во времени. (Например, локомотив L_3 — см. на рис. 1 — становится доступным только в момент времени $t = 5$.) Локомотив включается в алгоритм, когда он впервые оказывается позади (левее) идущего фронта. Если какой-то локо-слот после прохождения фронта через его начало остался незаполненным, это означает, что покрытия

данного графика не существует (пока это часть определения — для конкретного алгоритма построения фронта это утверждение будет нуждаться в доказательстве). При заполнении (и прохождении) локо-слотов, формирующих фронт с номером k , локомотивы попадают в новые точки графической плоскости, и их положение определяет фронт $k + 1$.

Рассмотрим алгоритм сдвига фронта на примере. Пусть график представляет собой локо-слоты, заданные на рис. 7. Пусть, кроме этого, имеются три локомотива, стартующих в нулевой момент времени.

Шаги алгоритма. Все локо-слоты, начала которых лежат в конусе скорости локомотива, соединим с данным локомотивом ребрами подцепки. Для графика (рис. 7) соответствия можно задать таблицей, в ячейках которой перечислены номера доступных локо-слотов:

Локомотив	Фронт	
	0	...
L_1	1, 4, 5, 6	
L_2	3, 4, 5, 6	
L_3	2, 3, 4, 5, 6	

Поскольку локомотивов три, может быть подцеплено не более трех составов. Для формирования фронта нужно оставить три локо-слота с наименьшими номерами (в данном случае, это номера 1, 2, 3), получим все возможные комбинации подцепки для фронта 0:

Локомотив	Фронт	
	0	...
L_1	1	
L_2	3	
L_3	2, 3	

Отсюда сразу видно, что существует единственная подцепка, при которой будут покрыты все три локо-слота (см. рис. 8).

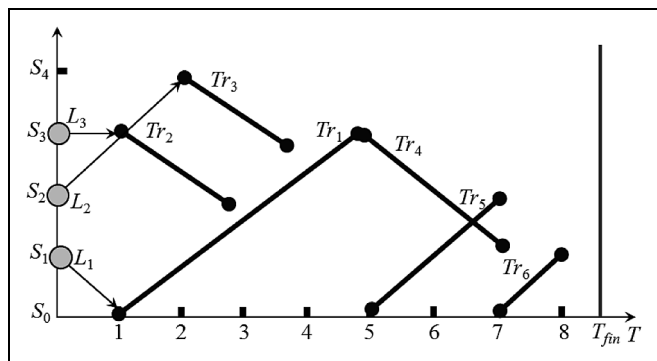


Рис. 7. График и локомотивы

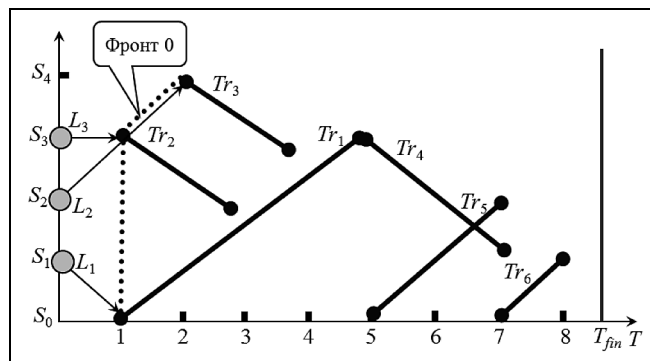


Рис. 8. Фронт 0

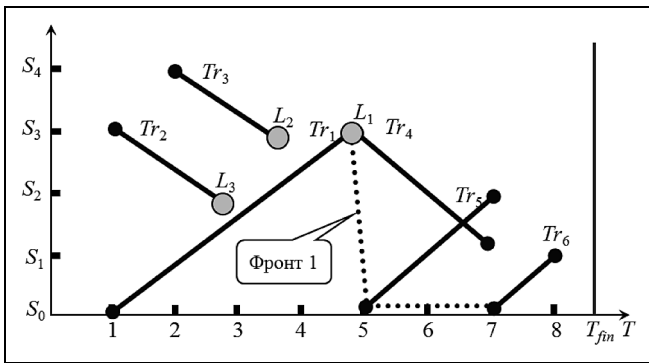


Рис. 9. Новый фронт (фронт 1)

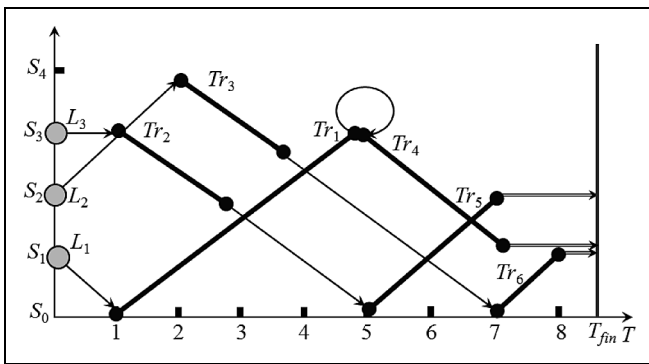


Рис. 10. Покрытие графика

В общем виде, если описывающие фронт ячейки содержат по несколько значений, задачу о поиске соответствий между локомотивами и номерами локо-слотов можно представить в виде двудольного графа. В таком виде решение сводится к поиску наибольшего паросочетания [21].

Замечание 2. Кажется бы, можно сразу построить ребра подцепки только к первым m локо-слотам, где m — число локомотивов, чтобы затем не отбрасывать лишнее. Однако это можно делать только в том случае, когда все m локо-слотов доступны и существует, по крайней мере, одно взаимно однозначное соответствие между локомотивами и локо-слотами. Если наибольшее паросочетание не покрывает m первых локо-слотов, график при таких начальных данных не является выполнимым. Однако алгоритм может продолжить работу для построения частичного покрытия. ♦

Положения локомотивов в концах локо-слотов фронта 0 определяют фронт 1. Новый фронт представляет собой начала локо-слотов с номерами 4, 5 и 6 (рис. 9). Напомним, что время перецепки в случае локо-слотов учитывается в скоростях, и перецепка $Tr_1 \rightarrow Tr_4$ является возможной.

Считая положение локомотивов (см. рис. 9) новым начальным состоянием, повторяем первый

шаг алгоритма. Нетрудно проверить, что ребра подцепки для фронта 1 будут такими:

Локомотив	Фронт		
	0	1	...
L_1	1	4	
L_2	3	6	
L_3	2	5, 6	

Как и прежде, видно, что распределение локомотивов по локо-слотам, покрывающее все три слота, единственно. Имеем два фронта, с помощью которых задается покрытие всего графика (рис. 10):

Локомотив	Фронт	
	0	1
L_1	1	4
L_2	3	6
L_3	2	5

Замечание 3. Как видно из описания алгоритма, он устойчив к появлению или выведению локомотивов на протяжении всего горизонта планирования. Добавленные локомотивы будут формировать ближайший справа к моменту добавления фронт. Выведенные локомотивы не будут принимать участие в формировании нового фронта. Никаких дополнительных изменений вносить не потребуется.

3.2. Условия применимости алгоритма сдвига фронта

Достижение фронтом горизонта планирования с покрытием всех локо-слотов является *достаточным условием достаточности числа локомотивов*.

Однако возможны ситуации, в которых описанный алгоритм не строит *покрытие*, хотя оно существует. Рассмотрим пример, представленный на рис. 11.

Применяя предложенный алгоритм, на фронте 0 получим:

Локомотив	Фронт	
	0	1
L_1	1, 2, 3	...
L_2	2, 3	...

Видно, что на фронте 0 достижимы оба первых локо-слота. Тогда:

Локомотив	Фронт	
	0	...
L_1	1	...
L_2	2	...

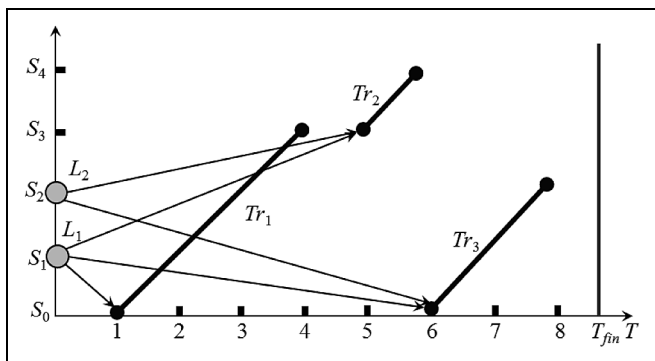


Рис. 11. Пример графика, для которого алгоритм сдвига фронта не строит покрытие

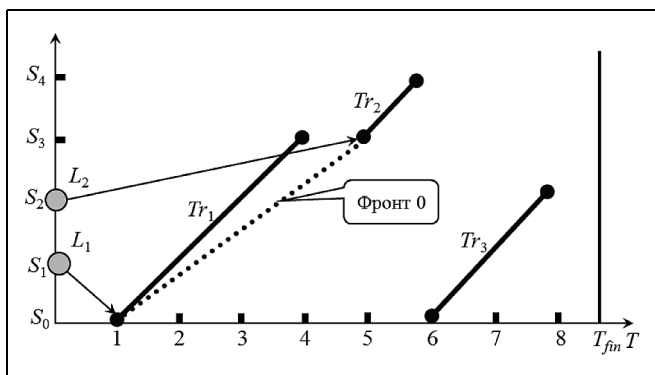


Рис. 12. Фронт 0

Из рис. 11 и 12 видно, что из концов первых двух локо-слотов нельзя попасть в начало локо-слота 3. Таким образом, локо-слот 3 покрыть не удастся. Фронт 1 и покрытие целиком не существуют:

Локомотив	Фронт	
	0	1
L_1	1	⊗
L_2	2	⊗

Однако легко заметить, что если локо-слоты 1 и 2 оставить для первого локомотива, то для этого графика существует покрытие:

Локомотив	Фронт	
	0	1
L_1	1	2
L_2	3	⊗

Ясно, что алгоритм сдвига фронта в описанном выше виде не может построить такое покрытие, потому что локо-слоты в нем распределены по фрон-

там не в порядке возрастания номеров. В этом покрытии утрачена монотонность (фронту с большим номером должны соответствовать локо-слоты с большими номерами).

Для того чтобы алгоритм работал корректно и всегда находил покрытие, если оно существует, нужно произвести *нормализацию фронта*.

3.3. Процедура нормализации фронта

Рассмотрим на примере графика (см. рис. 11 и 12), в чем заключается проблема описанного алгоритма. Из рис. 12 видно, что конец локо-слота Tr_1 оказался позади фронта 0. Это означает, что на начало формирования фронта 1 локомотив L_1 будет по-прежнему находиться в зоне действия фронта 0. Именно из-за этого произошла коллизия с распределением локо-слотов для локомотивов. Чтобы этого избежать, нужно сделать так, чтобы каждый локомотив при переходе в конец локо-слота, занятого на фронте k , проходил через линию фронта k . Таким образом, у каждого локо-слота, принадлежащего фронту k , конец должен лежать правее этого фронта.

При каком условии локо-слот Tr_j не переходит через фронт k ? Это случается, когда в этом же фронте существует другой локо-слот Tr_p , который начинается после окончания локо-слота Tr_j . Более формально, нормализация имеет смысл, если локо-слот Tr_i оказался в конусе скорости, вершина которого находится в конце локо-слота Tr_j . Иначе Tr_i не будет достижим из Tr_j , и поэтому его принадлежность текущему и следующему фронту даст один и тот же результат. Если на рис. 11 и 12 начало локо-слота Tr_2 сместить вниз на станцию S_0 , график не будет выполнен ни при каком распределении слотов по фронтам.

Нормализация фронта будет заключаться в том, что все локо-слоты Tr_p , начало которых лежит правее локо-слота Tr_j в конусе скорости, построенном из его конечной точки, удаляются из фронта k и переходят сразу на фронт $k + 1$. Тогда на фронте k будут задействованы не все локомотивы, а только те, которые соответствуют локо-слотам, недостижимым из конца Tr_j . Это позволит сохранить монотонность номеров локо-слотов — их возрастание в каждом последующем фронте.

Для ситуации, представленной на рис. 11 и 12, процедура нормализации будет выглядеть следующим образом. Из рис. 12 видно, что на фронте 0 локо-слот Tr_2 попадает в конус скорости из окончания локо-слота Tr_1 . Поэтому локо-слот Tr_2 исключается из фронта 0. Тогда локомотив L_2 тоже элиминируется из участия во фронте 0, так как на этом фронте для него нет подходящего слота.



Фронт 0 состоит только из одной точки — начала локо-слота Tr_1 (рис. 13):

Локомотив	Фронт	
	0	1
L_1	1	...
L_2	⊗	...

Фронт 1 в этом случае будет выглядеть, как на рис. 14:

Локомотив	Фронт	
	0	1
L_1	1	2
L_2	⊗	3

Как видно из примера, при нормализации фронта монотонность восстанавливается.

Модифицированный алгоритм назовем *алгоритмом сдвига фронта с нормализацией*.

Замечание 4. Легко увидеть, что по столбцам таблиц алгоритма записаны фронты, а по строкам — простые пути $Path_i$. Причем, заведомо (по построению) выполняется первое свойство покрытия графика G : пути попарно не пересекаются $Path_i \cap Path_j = \emptyset \forall i, j \in \overline{1, m}$. Если выполняется второе свойство: «их объединение содержит весь график G », то строки таблицы после выполнения алгоритма содержат покрытие. Если второе свойство не выполняется, то с помощью алгоритма было найдено лишь частичное покрытие. ♦

Докажем, что если алгоритм не смог найти полное покрытие графика, то такого покрытия не существует.

Для этого сначала докажем вспомогательные утверждения.

Лемма 1. Если при выполнении алгоритма сдвига фронта с нормализацией на графике G на фронте с номером k выбрано $l \leq t$ локо-слотов и существует несколько способов распределить их между l локомотивами, все назначения локомотивов эквивалентны относительно существования покрытия.

Доказательство. Рассмотрим положения локомотивов после прохождения фронта $k - 1$ или начальные положения, если $k = 0$. Если начало локо-слота Tr_p , принадлежащего фронту k , лежит в пересечении конусов скорости нескольких локомотивов, любой назначенный локомотив окажется после прохождения фронта в конце локо-слота Tr_p , откуда будет строиться новый конус скорости. При разных назначениях получатся разные простые пути для каждого локомотива в отдельности. Однако их объединение будет содержать одни и те же локо-слоты в силу построения. А, следовательно, существование покрытия инвариантно относительно альтернативных назначений локомотивов.

Лемма 2. Для данного графика G при данном множестве и начальном расположении локомотивов

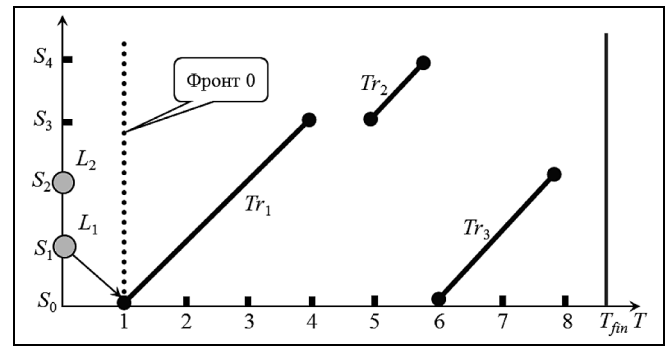


Рис. 13. Фронт 0 с участием одного из двух локомотивов

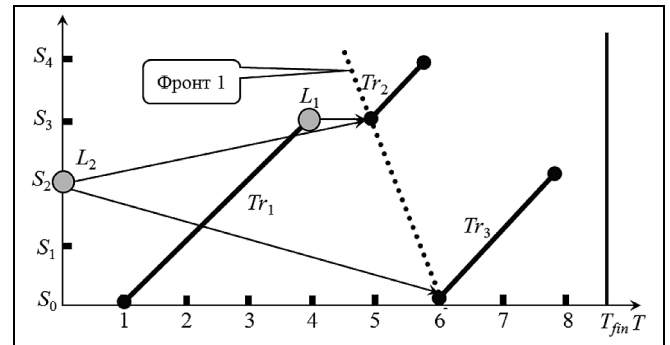


Рис. 14. Фронт 1, формирующий покрытие

нормализованные фронты задаются единственным образом.

Доказательство. Это утверждение верно по построению алгоритма. Не единственность возможна только в одном случае: если на фронте k существует более одного локо-слота с началами, лежащими в конусе скорости локомотива L_j (и только L_j); т. е. выбор состоит в том, какие из локо-слотов останутся не заполненными. В этом случае покрытия не существует. Для построения частичного покрытия добавим в алгоритм еще одно условие: будем считать, что в случае альтернативного выбора заполнения одного из нескольких локо-слотов выбирается локо-слот с наименьшим номером.

Теорема 1. Покрытие графика G при заданном начальном числе и положении локомотивов существует тогда и только тогда, когда оно может быть построено с помощью применения алгоритма сдвига фронта с нормализацией.

Доказательство. Достаточность в утверждении теоремы очевидна: если покрытие построено, то оно существует.

Необходимость докажем от противного. Пусть существует покрытие графика G . И пусть результатом алгоритма сдвига фронта с нормализацией является частичное покрытие графика G . По лемме 1, если алгоритм разветвляется, все его ветви эквивалентны относительно существования покрытия. Это означает, что если построено частичное покрытие, никакое другое назначение локомотивов с помощью алгоритма сдвига с нормализацией не даст полного покрытия. Тогда имеется

по крайней мере один локо-слот Tr_j , не попавший в таблицу алгоритма. По лемме 2 каждый фронт строится единственным образом.

Рассмотрим две ситуации.

1. В таблице алгоритма существует столбец k , содержащий номера слотов, меньшие j , и, по крайней мере, один из номеров, больших j . Это означает, что локо-слот Tr_j должен был оказаться во фронте k , поскольку алгоритм учитывает нормализацию фронта, номера локо-слотов распределяются по фронтам монотонно. Фронт k содержит не более чем m локо-слотов. Из фронта $k - 1$ и, возможно, более ранних при наличии нормализации к нему ведет столько же ребер подцепки. Если $k = 0$, ребра подцепки идут непосредственно из начального положения локомотивов. Слот Tr_j элиминируется из фронта k , если и только если все ребра подцепки, доступные для Tr_j , заняты слотами с меньшими номерами. А ребра, ведущие к следующим по номерам локо-слотам $Tr_l, l > j$, из фронта k , не могут быть переброшены на локо-слот Tr_j . Ни при какой другой расстановке локомотивов не существует возможности поместить на фронте k локомотивы с номерами без пропусков. Тогда, поскольку к фронту k ведут ребра подцепки из ближайших m локомотивов, то нельзя построить покрытие, содержащее все слоты, принадлежащие фронту k , и слот Tr_j . Отсюда следует, что покрытия графика не существует.

2. В таблице алгоритма существует столбец $k - 1$, содержащий номера слотов, меньшие j , и столбец k , все номера слотов в котором больше j . Это означает, что локо-слот Tr_j уже не вошел в фронт $k - 1$, но еще не вошел в фронт k . Рассмотрим фронт $k - 1$. Если в столбце $k - 1$ заняты все m строк, то номер j должен принадлежать столбцу k . Если в нем занято меньше m строк, то была произведена нормализация фронта, и номер j снова должен принадлежать столбцу k . В обоих случаях мы попадаем в ситуацию из п. 1, и покрытия графика не существует, что противоречит условию теоремы.

Что и требовалось доказать.

Следствие. Алгоритм сдвига фронта с нормализацией строит максимальное по включению частичное покрытие графика G при заданном начальном числе и положении локомотивов.

3.4. Оценка сложности алгоритма сдвига фронта с нормализацией

Оценим алгоритмическую сложность нахождения покрытия.

- Для определения фронта k требуется построение не более чем $C_1 m$ ребер подцепки.
- Нормализация фронта k требуется только в том случае, если конец одного из не более чем m локо-слотов лежит левее начала другого локо-слота из этого же множества. Поскольку локо-слоты упорядочены, проверка должна начинаться с правого конца. Если такой локо-слот найден, нужно проверить, попадает ли, по крайней мере, один локо-слот из фронта k , лежащий правее, в его конус скорости. Сложность такой процедуры не более чем $C_2 m^2$.

- Для назначения локомотивов локо-слотам на текущем фронте требуется найти наибольшее паросочетание в двудольном графе, в одной доле которого локомотивы, а в другой — локо-слоты. Задача поиска наибольшего паросочетания в двудольном графе является одной из алгоритмически простых задач, она решается многими методами, в том числе и потоковыми [22], при которых строятся фиктивный источник и сток. Алгоритм Хопкрофта — Карпа для двудольного графа имеет сложность, как максимум, $O(EV^{1/2})$. В нашем случае $E < m^2$, $V \leq 2m$. Если $E = m^2$, сочетания создаются произвольно за один шаг.

- Число фронтов $\sim n/m$.

Оценивая суммарное значение, получим: $(C_1 m)(C_2 m^2)(C_3 m^{3/2})(n/m) = C \cdot n \cdot m^{7/2}$. Общая сложность алгоритма, таким образом, составит в худшем случае $O(nm^{7/2})$. Однако, в общем случае, эта оценка сильно завышена.

ЗАКЛЮЧЕНИЕ

Предложен алгоритм, позволяющий однозначно определить, достаточно ли число локомотивов для выполнения данного графика. Результат его работы так называемое покрытие графика или максимальное частичное покрытие, если полного не существует. Покрытие — способ осуществить все перевозки средствами данных локомотивов; оно представляет собой объединение простых путей. Простой путь — путь от некоторого локомотива до стоковой вершины. В результате работы алгоритма получается столько путей, сколько локомотивов в системе. Каждый путь проходит через несколько фронтов, зависящих от числа и начального расположения локомотивов. В этом смысле алгоритм сдвига фронта сходен с алгоритмами перколяции [23—26]. Теория перколяции возникла в физике как теория, исследующая задачи просачивания жидкостей в пористой среде, а также некоторые задачи, связанные с проводимостью. Формально такого рода задачи можно рассматривать на графе, вершинам которого соответствуют пустоты пористой среды, а ребрам — каналы, соединяющие пустоты. Процесс просачивания выглядит при этом как постепенное и случайное заполнение вершин графа, при котором в графе возникают пути. Особое внимание в теории перколяции уделяется регулярным графам. На конечных решетках исследуют появление такого состояния, при котором существует хотя бы один непрерывный путь одного края решетки до противоположного. В случае алгоритма сдвига фронта случайные процессы заменяются детерминированными, при этом перколяция происходит через фронты и заканчивается образованием m путей, прошедших через весь граф.



Таким образом, видно, что при существовании нескольких полных покрытий графика, помимо стандартных методов поиска оптимального покрытия, можно применять неклассические методы, приводящие к решению задачи оптимизации на построенном графе. Выбор этих методов и нахождение оптимального решения для произвольного графика, так же, как описание свойств полученных графов — предмет дальнейших исследований.

ЛИТЕРАТУРА

1. Шабунин А.Б., Чехов А.В., Сазуров С.В. и др. Разработка сетцентрического подхода к созданию интеграционной платформы и интеллектуальной системы управления ресурсами грузовых сортировочных станций в реальном времени // Тр. третьей российской конф. с междунар. участием «Технические и программные средства систем управления, контроля и измерения» (УКИ'12), Москва, 16–19 апреля 2012, ИПУ РАН. — М., 2012. — С. 1560–1572.
2. Шабунин А.Б., Марков С.Н., Дмитриев Д.В. и др. Интеграционная платформа для реализации сетцентрического подхода к созданию распределенных интеллектуальных систем управления ресурсами ОАО «РЖД» // Программная инженерия. — 2012. — № 9. — С. 23–28.
3. Лазарев А.А., Мусатова Е.Г., Тарасов И.А. Решение задачи планирования двухстороннего движения на однопутном участке железной дороги с разездом // Автоматика и телемеханика. — 2016. — № 11. — С. 158–174.
4. Gafarov E., Dolgui A., Lazarev A. Two-Station Single-Track Railway Scheduling Problem With Trains of Equal Speed // Comput. and Industrial Engin. — 2015. — Vol. 85. — P. 260–267.
5. Лазарев А.А., Мусатова Е.Г. Целочисленные постановки задачи формирования железнодорожных составов и расписания их движения // Управление большими системами. — 2012. — Вып. 38. — С. 161–169.
6. Piu F. and Speranza M.G. The locomotive assignment problem: a survey on optimization models // Intl. Trans. in Op. Res. — Vol. 21. — P. 327–352. DOI: 10.1111/itor.12062.
7. Ahuja R.K., Liu J., Orlin J.B., et al. Solving real-life locomotive scheduling problems // Transportation Science. — 2005. — Vol. 39, N 4. — P. 503–517.
8. Jaumard B., Tian H. Multi-Column Generation Model for the Locomotive Assignment Problem // Proc. of 16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'16). — 2016. — P. 6:1–6:13.
9. Kasalica S., Mandić D., Vukadinović V. Locomotive Assignment Optimization Including Train Delays // Promet – Traffic & Transportation. — 2013. — Vol. 25, N 5. — P. 421–429.
10. Такмазьян А.К., Шелудяков А.В. Мультиагентное решение методом аукционов многопродуктовой транспортной задачи с объединенными потребностями // Тр. четвертой науч.-техн. конф. с междунар. участием «Интеллектуальные системы управления на железнодорожном транспорте. Компьютерное и математическое моделирование» ИСУЖТ-2015. — М., 2015. — Сер. 1. — С. 110–112.
11. Кузнецов Н.А., Пащенко Ф.Ф., Рябых Н.Г. и др. Алгоритмы оптимизации в задачах планирования на рельсовом транспорте // Информационные процессы. — 2014. — Т. 14, № 4. — С. 307–318.
12. Azanov V.M., Vyukanov M.V., Gaynanov D.N., Ivanov S.V. Algorithm and software development to allocate locomotives for transportation of freight trains // Вестник ЮУрГУ. Сер. Матем. моделирование и программирование. — 2016. — Т. 9, № 4. — С. 73–85.
13. Гайнанов Д.Н., Кибзун А.И., Рассказова В.А. Теоретико-графовый алгоритм решения задачи о назначении и перемещении локомотивов // Вестник компьютерных и информационных технологий. — 2017. — № 5. — С. 51–56.
14. Matyukhin V.G., Shabunin A.B., Kuznetsov N.A., Takmazjan A.K. Rail Transport Control by Combinatorial Optimization Approach. 11th IEEE International Conference on Application of Information and Communication Technologies AICT2017. Conf. Proc., IEEE Communications Society. — 2017. — Vol. 1. — P. 419–422.
15. Матюхин В.Г., Шабунин А.Б., Кузнецов Н.А. и др. Графовая динамическая модель задачи подбора тяговых ресурсов для грузовых железнодорожных перевозок // Тр. шестой науч.-техн. конф. «Интеллектуальные системы управления на железнодорожном транспорте. Компьютерное и математическое моделирование» (ИСУЖТ-2017), АО «НИИАС». — М., 2017. — С. 14–18.
16. Mascis A., Pacciarelli D. Job-shop scheduling with blocking and no-wait constraints. European Journal of Operations Research. — 2005. — Vol. 143, N 3. — P. 498–517.
17. Ерусалимский Я.М. Потоки в сетях с нестандартной достижимостью // Известия вузов. Северо-Кавказский регион. Ест. науки. — 2012. — № 1. — С. 5–7.
18. Ерусалимский Я.М., Скороходов В.А., Кузьмина М.В., Петросян А.Г. Графы с нестандартной достижимостью: задачи, приложения: монография. — Ростов-на-Дону: ЮФУ, 2009. — 195 с.
19. Ловас Л., Пламмер М. Прикладные задачи теории графов. Теория паросочетаний в математике, физике, химии: пер. с англ. — М.: Мир, 1998. — 653 с.
20. Седжвик Р. Фундаментальные алгоритмы на C++. Алгоритмы на графах. — СПб.: ДиаСофтЮП, 2002. — 496 с.
21. Lovász L., Plummer M.D. Matching Theory. — North-Holland, 1986. — 554 p.
22. Ahuja R.K., Magnanti T.L., Orlin J.B. Network Flows: Theory, Algorithms, and Applications. — Prentice Hall, United States, 1993.
23. Bollobas B., Riordan O. Percolation: 2nd ed. — Cambridge: Cambridge Univ. Press, 2009.
24. Тарасенко Ю.Ю. Перколяция: Теория, приложения, алгоритмы: 2-е изд. — М.: Либроком, 2012.
25. Кузнецов О.П. Сложные сети и распространение активности // Автоматика и телемеханика. — 2015. — № 12. — С. 3–26.
26. Radicchi F. Percolation in real interdependent networks // Nature Physics. — 2015. — Vol. 11. — P. 597–602. DOI:10.1038/nphys3374.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Жилякова Людмила Юрьевна — д-р физ.-мат. наук, вед. науч. сотрудник, Институт проблем управления им. В.А. Трапезникова РАН, г. Москва, ✉ zhilyakova.ludmila@gmail.com,

Кузнецов Николай Александрович — академик РАН, советник РАН; зав. лабораторией, Институт радиотехники и электроники им. В.А. Котельникова РАН, г. Москва, ✉ kuznetsov@cplire.ru,

Матюхин Владимир Георгиевич — д-р техн. наук, первый зам. ген. директора, Научно-исследовательский и проектно-конструкторский институт информатизации, автоматизации и связи на железнодорожном транспорте, г. Москва,

Шабунин Александр Борисович — руководитель науч.-техн. комплекса, Научно-исследовательский и проектно-конструкторский институт информатизации, автоматизации и связи на железнодорожном транспорте, г. Москва, ✉ a.shabunin@vniias.ru,

Такмазьян Андрей Куркенович — канд. физ.-мат. наук, гл. специалист, ООО «Програмпарк», г. Москва, ✉ takmazjan@gmail.com.