УДК 519.178

ФОРМИРОВАНИЕ МАРШРУТОВ ДВИЖЕНИЯ ПОЕЗДОВ НА СТАНЦИИ[1]

С.А. Браништов, А.М. Ширванян, Д.А. Тумченок

С целью автоматизации управления движением поездов на станции исследована задача поиска маршрутов. Рассмотрена проблема выбора маршрутов приема/отправления поездов и маневровых перемещений. Отмечено, что для станций с разветвленной инфраструктурой путей число всех возможных маршрутов высоко, и объем вычислений при поиске велик. Предложен метод разделения станции на районы для сокращения объемов вычислений. Поставлена задача нахождения мест разрезов графа сети с учетом особенностей железнодорожной инфраструктуры, предложен алгоритм ее решения.

Ключевые слова: организация движения поездов, железнодорожная сеть, маршруты движения поездов, разрез графа.

ВВЕДЕНИЕ

При организации безопасного и своевременного перемещения поездов по станции для каждого поезда в расписании необходимо найти свободный и безопасный маршрут и приготовить его для движения к нужному моменту времени. Среди всех возможных маршрутов следует выбирать тот, который позволит без затруднений составить маршруты и для всех последующих поездов, выбор должен быть сделан обдуманно и с учетом графика движения. Сейчас задача поиска подходящего маршрута выполняется в реальном времени, и для сокращения времени целесообразно не искать все возможные маршруты между заданными точками, а выбирать из подготовленного списка. Поэтому требуется заранее на известной сети найти все маршруты между всеми допустимыми точками начала и конца маршрута.

Для железнодорожных станций характерна разветвленная схема путей и, как правило, существует несколько возможных маршрутов движения между двумя точками станции. А на крупных железнодорожных станциях может существовать несколько тысяч возможных маршрутов, в таком случае поиск всех возможных маршрутов займет значительное время. Уменьшить число возможных маршрутов на схеме станции можно, если разделить ее на районы и маршруты находить внутри каждого района отдельно, а затем их комбинировать. Для этой цели предлагается метод разбиения железнодорожной сети станции на районы, учитывая топологические свойства сети и правила движения поездов.

Попробуем оценить, сколько маршрутов может быть на станции ограниченной размерности и какова вычислительная сложность задачи поиска всех маршрутов.

Железнодорожная сеть станции состоит из нескольких типов элементов, это: стояночные пути, стрелки, светофоры, выходные пути и др. Маршруты возможны между стояночными путями и выходными путями. Рассмотрим два типа железнодорожных станций — «сквозные» и «тупиковые». Для инфраструктуры «тупиковой» станции характерно такое расположение путей, при котором с одной стороны находятся стояночные пути, а на противоположной стороне горловины станции находится группа выходных путей, а между этими группами разветвленная сеть путей станции. Для «сквозной» станции характерно наличие групп выходных путей с разных сторон, а стояночные пути расположены преимущественно между ними [1].

Для представления разветвленности сети путей можно привести пример количественных данных одной из московских станций. Станция является «тупиковой» и содержит 108 стояночных путей, 14 выходных путей, 222 стрелки, из которых 21 двойная. Разветвленность сети обуславливается стрелками, и число всех возможных маршрутов зависит от их количества. Вывести аналитическую зависимость числа возможных маршрутов только

[1] Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 13-08-13207.

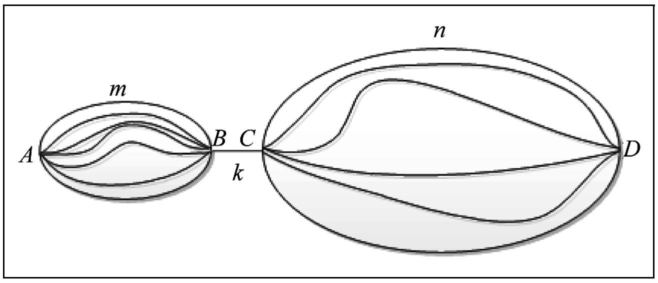

Рис. 1. Абстрактный пример разветвленного участка станции

лишь от количества стояночных путей, выходных путей и стрелок не представляется возможным, поскольку это число зависит от конфигурации связей между ними. Только между некоторыми двумя стояночными путями, находящимися в разных частях станции, число вариантов маршрутов достигло нескольких тысяч. Всего на рассматриваемой станции было найдено 148 840 маршрутов. Для построения такого количества маршрутов необходимы значительные временные затраты, кроме того всевозможные маршруты сохраняются в базе данных, а так как каждый маршрут состоит в среднем из 80 элементов, то для этого понадобится выделить достаточно большой объем памяти.

Рассмотрим пример, наглядно иллюстрирующий случай, когда необходимо разбиение сети на районы.

На рис. 1 показан граф, в котором есть две разветвленные части и «узкое» место. Под «узким» местом будем понимать участок графа, который соединяет между собой две разветвленные части графа, а сам содержит значительно меньшее число связей.

Пусть граф моделирует сеть путей станции, и m — число возможных путей на графе (далее — маршрутов на станции) от точки A до точки B, n — число возможных маршрутов от точки C до точки D, а k — число связей между точками B и C, тогда между точками A и D получается mkn маршрутов, тогда как при разбиении на районы AB и CD получается $mk + kn$ маршрутов. Очевидно, что чем более разветвлена станция, тем большее количество маршрутов можно провести. Нетрудно заметить, что для $m > 2$ и $n > 2$ справедливо неравенство $mk + kn < mnk$. А для больших m и n: $mk + kn \ll mnk$. Поэтому целесообразно разбивать сети путей железнодорожных станций на несколько различных районов.

Цель работы заключается в разработке метода разбиения железнодорожной сети станции на районы, учитывая топологические свойства сети и принцип движения поездов.

1. ЗАДАЧА РАЗРЕЗАНИЯ ГРАФА

Проблема разрезания графа на подграфы является одной из актуальных комбинаторных проблем теории графов. Полученные решения не только важны для самой теории графов, но и для практического применения в различных областях. Во многих прикладных задачах, связанных с техническим проектированием, обработкой и классификацией данных, широко применяются различные алгоритмы разрезания графа на подграфы [2]. При разрезании выполняется анализ структурных свойств графа с тем, чтобы найти места разрезов графа. Результатов разрезания может быть получено множество, размеры этого множества полиномиально зависят от размеров графа. Подобная задача относится к классу NP-полных задач, т. е. к таким задачам, время решения которых зависит нелинейно от размерности входных данных. Для выбора варианта разрезаний и оценки их качества часто прибегают к алгоритмам перебора, которые гарантируют решение, но требуют значительных временных ресурсов. Такие способы хороши и удобны для задач малой входной размерности, так как для размерности m необходимо сравнить $m!$ вариантов. Чтобы упростить поиск решения, применяются различные эвристические способы ограничения перебора, которые базируются на математических закономерностях, сокращающих число возможных решений и количество шагов алгоритма [3]. Алгоритмы разрезания графов применяются для решения различных задач, например, для проектирования интегральных схем, трассировки печатных плат, проектирования топологии локальной сети и др.

Сформулируем задачу *разрезания графа* $G = (X, U)$ на подграфы $G_i = (X_i, U_i)$, где X — множество вершин графа и U — множество ребер графа, $i \in I = \{1, 2, ..., l\}$, где l — число подграфов, на которые разрезается граф. Разрезание графа определим по аналогии с разбиением множеств.

Совокупность подграфов $B(G_i)$ называется *разрезанием* графа $G = (X, U)$, если

$$(\forall G_i \in B(G_i))[G_i \neq \varnothing], \quad i \in I,$$

$$(\forall G_i, G_j \in B(G_i)[G_i \neq G_j \; \& \; X_i \cap X_j = \varnothing \; \& \; U_{ij}],$$
$$i, j \in I, \bigcup_{i \in I} G_i = G.$$

Обозначим $U_{ij} = U_i \cap U_j$ — множество ребер, попадающих в разрез между подграфами G_i и G_j; $|U_{ij}| = k_{ij}$ — *число реберного соединения* подграфов; $k = \dfrac{1}{2}\sum\limits_{i=1}^{l}\sum\limits_{j=1}^{l} k_{ij}$, $i \neq j$. Если $k_{ij} = 1$, то подграф G_i называется *полуостовом*.

Задача разрезания графа $G = (X, U)$ заключается в нахождении такой совокупности подграфов, чтобы суммарное число реберного соединения удовлетворяло заданному критерию оптимальности. Под оптимальным разрезанием графа $G = (X, U)$ будем понимать такое разрезание $B(G_i)$, при котором выполняется условие $(\forall G_i \in B(G_i))\left[\left|\bigcup_{i \in I} U_{ij}\right| = k_{\min}\right]$, т. е. критерием оптимальности служит минимальное число ребер между подграфами графа G.

Пусть граф G разрезан на подграфы $G_1, G_2, ..., G_l$. В соответствии с этим множество ребер графа G представим в виде $U = U_1 \cup U_2 \cup ... \cup U_l = \bigcup_{i \in 1}^{l} U_i$.

Каждое U_i запишем в виде:

$$\begin{cases} U_1 = U_{11} \cup U_{12} \ldots \cup U_{1i} \ldots \cup U_{1l}, \\ U_2 = U_{21} \cup U_{22} \ldots \cup U_{2i} \ldots \cup U_{2l}, \\ \qquad \ldots, \\ U_i = U_{i1} \cup U_{i2} \ldots \cup U_{ii} \ldots \cup U_{il}, \\ \qquad \ldots, \\ U_l = U_{l1} \cup U_{l2} \ldots \cup U_{li} \ldots \cup U_{ll}, \end{cases}$$

где U_i — множество всех ребер, инцидентных вершинам X_i подграфа G_i, а U_{ij} — множество ребер, соединяющих только вершины X_i подграфа G_i между собой; $U_{ij}(U_{ji})$ — множество ребер, соединяющих подграфы G_i и G_j.

Поставленная задача является задачей комбинаторно-логического типа, и решение ее связано с перебором различных вариантов разрезания графа на подграфы. В силу экспоненциальной сложности алгоритма перебора необходимо разрабатывать такие алгоритмы, которые не связанны с полным перебором, хотя и приводят к нахождению не общего, а локального минимума, достаточного для практических целей.

Введем некоторые классы разрезаний. Если при разрезании графа $G = (X, U)$ в каждый подграф попадает по одной вершине, то такое разрезание называется *поэлементным*. Разрезание называется *целым*, если $B(G_l) = \{G_i\}$. Очевидно, что поэлементное и целое разрезания тривиальны и не представляют практического интереса. Заметим, что разрезание полного графа также не имеет смысла.

Существует значительное число алгоритмов разрезания графа, которые можно разделить на:

— алгоритмы, основанные на методах целочисленного программирования;

— последовательные алгоритмы;

— итерационные алгоритмы;

— алгоритмы разрезания, основанные на методе ветвей и границ;

— смешанные алгоритмы, содержащие последовательную и итерационные части;

— алгоритмы нахождения максимального потока.

Заметим, что алгоритмы, основанные на методах целочисленного программирования, не нашли широкого применения в связи с большими трудностями реализации их на ЭВМ. На практике применяются итерационные, последовательные и смешанные алгоритмы. При последовательных алгоритмах сначала выбирается по определенному критерию вершина графа, затем к ней присоединяются другие вершины с целью образования первого подграфа, второго подграфа и так далее до получения желаемого разрезания. Очевидно, что наибольшая эффективность таких алгоритмов достигается, когда число вершин графа значительно больше числа вершин в любом подграфе. Эти алгоритмы весьма просты, легко реализуются на ЭВМ, позволяют быстро получить результат разрезания, однако в общем случае приводят к результатам, далеким от оптимальных.

В соответствии с итерационными алгоритмами граф разбивается на определенное число подграфов произвольным образом. Затем производится перестановка вершин из одного подграфа в другой по некоторым критериям с целью минимизации числа ребер между подграфами. Заметим, что при работе таких алгоритмов количество итераций, время решения и оптимальность результата сильно зависят от того, насколько удачно выбрано начальное разрезание. Для устранения этого недостатка необходимо алгоритм применять несколько раз для различных начальных условий [4].

В теории оптимизации и теории графов, задача о максимальном потоке заключается в нахождении такого потока по транспортной сети, что сумма потоков из истока или, иными словами, сумма потоков в сток максимальна. По теореме о максимальном потоке и минимальном разрезе (теореме Форда — Фалкерсона) максимальный поток равен пропускной способности минимального разреза. По сути, это теорема двойственности для задачи линейного программирования. Большинство алгоритмов нахождения максимального потока имеют сложность $O(N^3)$, где N — число ребер, или близкую к ней и не гарантируют, что время их работы полиномиально. Кроме того, некоторые алгоритмы этого класса сильно зависят от начальных условий, например, для нахождения максимального потока может понадобиться как одна итерация, так и тысяча [5].

Таким образом, для решения поставленной задачи необходимо определить принцип поиска мест разрезов и аналитически выявить и предложить «узкие» места для разреза разветвленной схемы железнодорожной станции. Рассмотренные ранее методы не могут быть применены для решения задачи разбиения сети железнодорожной станции в силу разных своих особенностей. Поэтому было предложено разработать новый метод, позволяющий разбивать на части разветвленную сеть.

2. МЕТОД РАЗБИЕНИЯ ЖЕЛЕЗНОДОРОЖНОЙ СТАНЦИИ НА РАЙОНЫ

Суть метода заключается в нахождении минимального разреза графа через поиск максимального потока (теорема Форда — Фалкерсона). В отличие от известных методов, важная черта предложенного метода заключается в свойстве учитывать особенность перемещения поездов на станции. На графе станции могут существовать только такие потоки, которые соответствуют маршрутам движения поездов между определенными целевыми вершинами (сток, исток). Целевая вершина — это вершина графа, соответствующая начальному или конечному элементу маршрута в разных районах железнодорожной сети. Чтобы найти поток ребра графа, нужно проложить только возможные маршруты движения поездов между целевыми вершинами разных районов сети и подсчитать число использований каждого элемента сети (ребра графа) в разных маршрутах. Ребра с максимальным коэффициентом использований покажут максимальную плотность движения и места разреза графа.

Пусть дан граф $G = (X, U)$, где X — множество вершин графа, U — множество ребер графа. Необходимо разбить граф G на k подграфов. Известна принадлежность каждой целевой вершины к одной из k областей. Места разреза должны быть найдены для каждой пары областей A и B.

Пусть $a_i \in A$ — целевые вершины области A и $b_j \in B$ — целевые вершины области B, тогда $\forall(a_i, b_j) \to R_{ij}$, где $R_{ij} = \{r_1, r_2, ..., r_s\}$ — множество возможных путей на графе G (маршрутов движения на сети) между вершинами a_i и b_j, i, j — индексы элементов множеств A и B соответственно.

Пусть $u_l \in U$ — ребро графа G с индексом l и показатель m_l этого ребра, такой что $m_l = p/|R_{ij}|$, где p — число маршрутов, проходящих по ребру $u_l \subset R_{ij}$.

Показатель m_l ребра отражает долю участия этого ребра во всех путях R_{ij} между двумя конкретными вершинами a_i и b_j. Однако этого показателя недостаточно, чтобы по его значению определить место разреза, так как он не учитывает пути от других целевых вершин и соответственно не получает доли веса от остальной части графа. Если найти показатель m_l для других пар целевых вершин и разделить на число этих пар, то полученный вес ребра будет показывать степень использования его во всех этих маршрутах.

Пусть f_l — число множеств R_{ij}, включающих ребро u_l, $\forall u_l \subset R_{ij}$: $f_l = \#\{R_{ij}\}$. Тогда отношение $(\Sigma m_l)/f_l$ — степень использования l-го ребра графа. Возможные места разреза графа имеют максимальные показатели степени использования.

Примечание. Ребра с максимальной степенью использования и такие, для которых число пар целевых вершин f_l равно единице, следует исключить из мест разреза. Такими ребрами являются начальные и конечные ребра, а также те, по которым не проходят маршруты других целевых вершин, значит, эти ребра не являются «узкими» местами. ♦

Алгоритм оценки частоты использования для каждой пары заключается в следующем.

1. Выбирается пара областей A и B, между которыми необходимо найти место разреза.
2. Составляются множества целевых вершин областей A и B: (A, $B \in X$).
3. Строятся всевозможные пути r_p для пары вершин a_i и b_i.
4. Вычисляются значения m_l и f_l.
5. $\forall u_l \in U$ вычисляются значения $(\Sigma m_l)/f_l$.

Рассмотрим алгоритм поиска разреза на графе, изображенном на рис. 2. Пусть вершины *1*, *2*, *3* и *10* — целевые, причем вершины *1*, *2* и *3* прина-

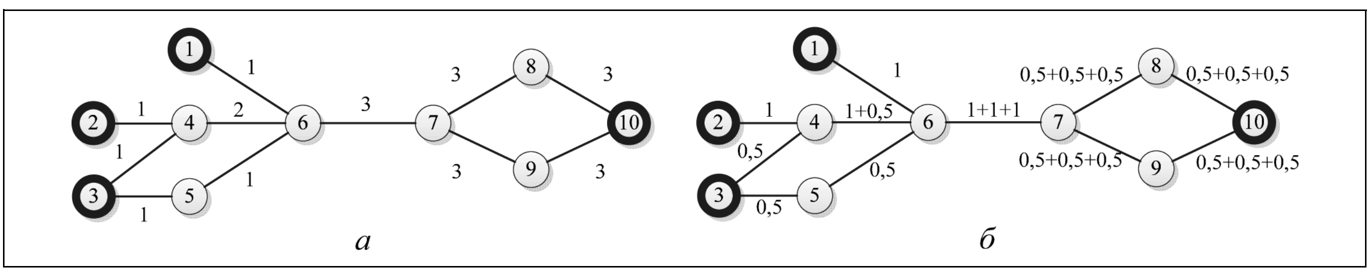

Рис. 2. Примеры графа: *а* — значения f_l; *б* — значения m_l

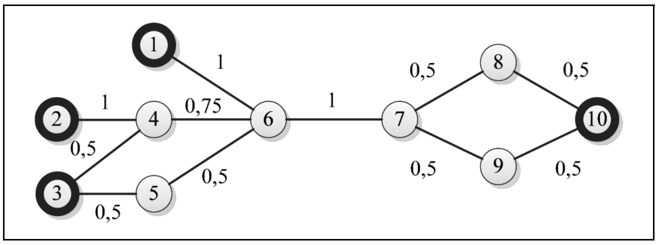

Рис. 3. Степени используемости ребер

длежат области А, а вершина *10* принадлежит области В. Необходимо найти место разреза графа.

Построим всевозможные маршруты из области А в область В. Для данного примера множества $R_{1,10}$ и $R_{2,10}$ содержат по два, а множество $R_{3,10}$ — четыре маршрута. Вычисляем значения m_l и f_l для множеств $R_{1,10}$, $R_{2,10}$ и $R_{3,10}$ и получаем значения f_l (рис. 2, *а*)) и m_l (рис. 2, *б*)). Для каждого ребра графа вычисляем отношение $(\Sigma m_l)/f_l$, получим граф, представленный на рис. 3.

Полученные результаты (см. рис. 3) позволяют сделать вывод, что данную схему следует разрезать между вершинами *6* и *7*. Для проверки разработанного алгоритма создана компьютерная программа. Цель проведенного эксперимента: найти «узкие» участки схемы между районом приема/отправления и районом стояночного парка одной крупной пассажирской станции. Для этого были построены маршруты от некоторых случайно выбранных приемо-отправочных стояночных и выходных путей к путям парка. Результат применения метода показан в таблице.

На рис. 4 изображена схема железнодорожных путей станции. На ней обозначены целевые точки начала и конца маршрутов движения. Вертикальными овалами светло-серого цвета слева показаны приемо-отправочные пути. Вертикальными овалами серого цвета в середине схемы показаны стояночные пути, которые нужно отнести к другому району. С правой стороны схемы вертикальный овал показывает пути входа (выхода) на станцию.

При моделировании маршрутов между этими точками были получены области с максимальной степенью использования во всех маршрутах. Горизонтальными овалами темно-серого цвета показаны области, найденные с помощью описанного метода, в которых рекомендуется разрезать схему для разделения станции на районы. Центральная и правая области разрезов отделяют район с группой

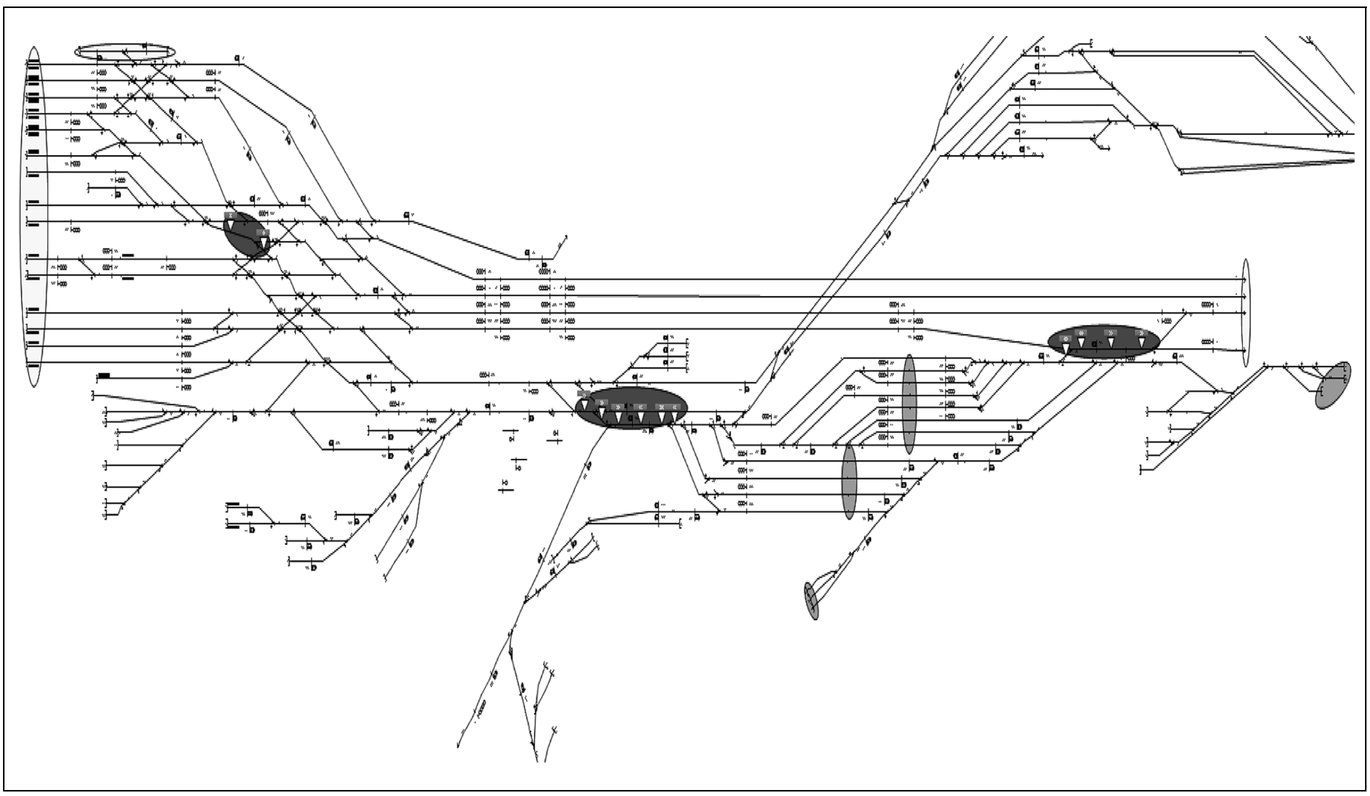

Рис. 4. Схема сети путей железнодорожной станции

Результаты работы алгоритма

<table>
<tr><th>Номер элемента</th><th>Число использований, ед.</th><th>Кол-во множеств</th><th>Число стартовых вершин</th><th>Частота использования, %</th></tr>
<tr><td>70
73</td><td>30</td><td>8</td><td>2</td><td rowspan="4">100</td></tr>
<tr><td>248
254
255
259</td><td>300</td><td>16</td><td>4</td></tr>
<tr><td>564
348
435
436</td><td>10</td><td>10</td><td>2</td></tr>
<tr><td>562</td><td>300</td><td rowspan="2">16</td><td rowspan="2">4</td></tr>
<tr><td>263</td><td>260</td><td>96</td></tr>
</table>

стояночных путей от района приема/отправления и путей входа на станцию. Такое деление подтверждается фактическим разделением станции на действующей эксплуатационной схеме.

Здесь важно учитывать условия эксперимента. Например, темный горизонтальный овал слева также показывает возможную зону разреза схемы. Однако высокая степень использования этих участков пути сформирована только для маршрутов между этими двумя районами. Если включить в оценку маршруты от путей приема/отправления и в другие районы, то этот показатель будет резко снижен и не укажет на разрез, который в действительности здесь не нужен.

3. ВЫБОР МАРШРУТОВ

Имеющаяся система централизации управления (СЦБ) автоматически предлагает дежурному основной маршрут движения. Она поможет проверить маршрут на скрещение с уже занятыми и зависимыми путями и приготовить его, т. е. установить в нужное положение стрелки и сигналы светофора. Если требуется альтернативный маршрут, то дежурный самостоятельно вручную переключает стрелки так, чтобы сформировать иной маршрут. При назначении маршрута дежурный осуществляет выбор самостоятельно на основании текущей ситуации и личного опыта. Для автоматизации этого процесса авторами разработан алгоритм выбора маршрута. Чтобы избежать полного перебора возможных вариантов, решено воспользоваться следующим критерием выбора лучшего маршрута. Прежде всего, маршрут следует выбирать такой, который не является враждебным уже выбранным ранее маршрутам. Затем для каждого поезда находится такой маршрут, который перекрывается меньше всего раз по общим участкам пути с вероятными маршрутами других поездов в течение времени своего движения. Так найденный маршрут будет иметь наименьшую вероятность скрещения с последующими поездами. Проблема выбора маршрута в том, что необходимо учитывать пункты следования и маршруты движения последующих в расписании поездов. Выбираемый маршрут должен занимать горловину станции так, чтобы не блокировать движение других поездов. В периоды высокой интенсивности движения контролировать это непросто, могут возникать ситуации, когда использование путевого развития таково, что нет больше возможности найти свободные пути для движения поезда. В таком случае должны применяться и другие меры по управлению поездами, например: допустимое манипулирование временем движения поездов по станции, планирование маневровых перемещений, смена платформы приема, использование сложных маршрутов со сменой направления движения и др. [6]. Для решения задачи выбора маршрутов нужно прогнозировать занятость путей во времени, соответствие типа поезда и специализации пути приема, зависимости стрелок, обеспечение интервалов движения между поездами, ограничения скорости движения на участках и стрелках, направления движений на перегонах.

В соответствии с алгоритмом выбора вначале находится множество вариантов маршрутов для рассматриваемого поезда и вычисляется время движения поезда по каждому из них в соответствии с профилем скорости вдоль маршрута (Δt_{dr}). Чтобы учесть влияние движения этого поезда на максимально возможное число последующих поездов, берется наибольшее время.

Определим время движения. Пусть t — момент начала движения (маршрута) поезда по станции. Для поездов, отправляющихся со станции, этот момент соответствует указанному в расписании t_{dep}, а для прибывающих поездов это момент времени входа в горловину станции, он должен быть рассчитан так, чтобы поезд прибыл на платформу не позднее указанного в расписании момента времени t_{arr}. Таким образом, максимальное время движения по станции $[t, t + \Delta t_{dr}]$, где $t = t_{dep}$ — для отправляющихся поездов, $t = (t_{arr} - \Delta t_{dr})$ — для прибывающих.

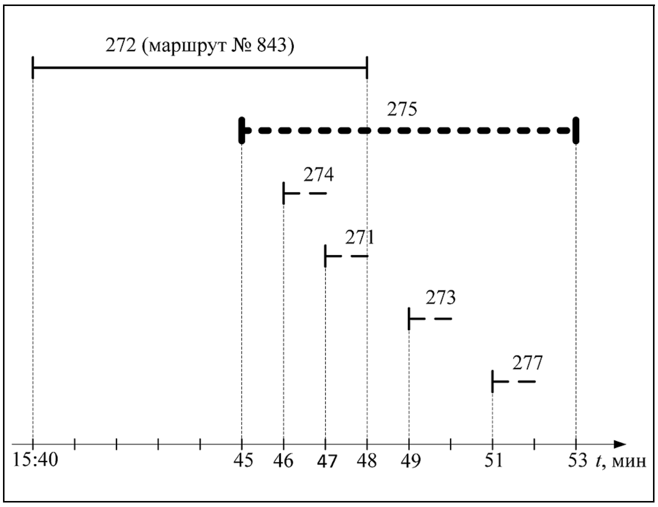

Рис. 5. Диаграмма движения поездов

На втором этапе отбрасываются те варианты маршрутов, которые пересекаются с уже приготовленными маршрутами других поездов по общим элементам сети. Оставшимся вариантам движения рассматриваемого поезда раздаются веса в соответствии с влиянием на последующие поезда. Вес вычисляется как сумма числа использования элементов маршрута всеми поездами в рассматриваемом интервале времени. Выбирается тот маршрут, чей вес наименьший. Для примера на рис. 5 показана шкала времени и моменты движения поездов. Для поезда № 272 на отрезке времени [15:40; 15:48] задан маршрут № 843 и этот поезд находится в движении. Для рассматриваемого поезда № 275 имеется 18 вариантов маршрута, и необходимо выбрать один из них. Максимальное время его движения 8 мин (с 15:45 до 15:53), в этот период начнут свое движение поезда: 274, 271, 273 и 277. Выбираемый маршрут должен быть тем, который менее остальных влияет на движение этих поездов и не пересекается с поездом 272. После выбора маршрута для каждого последующего поезда должен быть хотя бы один незаблокированный маршрут.

Практически на исследуемой пассажирской станции обнаруживались моменты времени, когда в движении находились одновременно восемь поездов. В этой ситуации ресурсы путевого развития сильно сокращены и подобрать свободные маршруты для всех поездов непросто.

При таком подходе, когда последовательно для каждого поезда выбираются маршруты с наименьшим влиянием на другие поезда, на практике достигается высокая вероятность нахождения реализуемых маршрутов.

ЗАКЛЮЧЕНИЕ

Описаны проблемы автоматизации поездной и маневровой работы на железнодорожной станции. Предложен метод поиска и выбора маршрута перемещения поезда по разветвленной инфраструктуре путей станции. Метод опирается на алгоритм разбиения железнодорожной сети станции на районы для ускорения операции поиска маршрутов. Достоинства метода заключаются в простоте применения и реализации на ЭВМ, легкость вычислений, а также возможность аналитически обнаружить «узкие» места схемы сети станции за короткое время. К недостаткам метода можно отнести невозможность применения в других транспортных системах из-за особенностей маршрутов движения поездов по станции. Разработанная программа, реализующая данный алгоритм, протестирована на модели крупной железнодорожной станции и указала на возможные места разделения большой сети на районы. Особенность данного метода в том, что в результате выделяется не единственное место разреза (одно ребро), а локальная область, и место разбиения сети выбирает эксперт.

ЛИТЕРАТУРА

1. *Грау Б.* Проектирование железнодорожных станций. — М.: Транспорт, 1978. — 487 с.
2. *Naveed Sherwani.* Algorithms for VLSI physical design automation. — Boston-Dordrecht-London: Kluwer Academic Publisher, 1995.
3. *МакКоннелл Дж.* Основы современных алгоритмов. — М.: Техносфера, 2004.
4. *Мелихов А.Н.* Применение графов для проектирования дискретных устройств. — М.: Наука, 1974. — 304 с.
5. *Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.* Алгоритмы: построение и анализ / 2-е изд., пер. с англ. — М.: Вильямс, 2005. — 1296 с.
6. *Инструкция* по движению поездов и маневровой работе на железнодорожном транспорте РФ (приказ Минтранса России от 04.06.2012 г. № 162).

Статья представлена к публикации членом редколлегии В.Н. Бурковым.

Браништов Сергей Александрович — канд. техн. наук, и. о. зав. лабораторией, ✉pochta-na@mail.ru,

Ширванян Артем Мартиросович — аспирант, мл. науч. сотрудник, ✉artshirvanyan@mail.ru,

Тумченок Дмитрий Александрович — аспирант, мл. науч. сотрудник, ✉dmitriy_tumchenok@mail.ru,

Институт проблем управления им. В.А. Трапезникова РАН, г. Москва, ☎ (495) 334-93-70.