УДК 519.854.2
ББК 22.1

ОПЕРАТИВНОЕ УПРАВЛЕНИЕ ДВИЖЕНИЕМ СОСТАВОВ ПРИ ПРОВЕДЕНИИ РЕМОНТНЫХ РАБОТ НА ДВУХПУТНОЙ ЖЕЛЕЗНОЙ ДОРОГЕ

Лазарев А.А.[1], **Хуснуллин Н.Ф.**[2]

(Учреждение Российской академии наук Институт проблем управления им. В.А. Трапезникова РАН, Москва)

В данной работе рассматривается задача нахождения расписания на двухколейной железной дороге при условии, что один из участков между семафорами закрыт. Предложен точный алгоритм, основанный на методе динамического программирования.

Ключевые слова: теория расписания, железнодорожный транспорт, динамическое программирование.

Введение

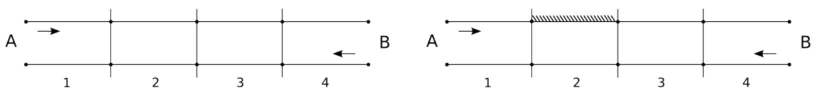
В эксплуатационной работе железных дорог различают нормативный (НГДП) и вариантный (ВГДП) графики движения поездов. Первый, разрабатывается и составляется ежегодно технологами отдела разработки графиков движения поездов службы перевозок железных дорог. ВГДП разрабатываются на участках, где предоставляются окна для ремонтных и строительных работ, влияющие на условия пропуска поездов и размеры движения поездов. Он должен обеспечивать пропуск установленных среднесуточных размеров движения поездов на железнодорожном участке; в противном случае совместно с департаментом перевозок решается вопрос об отклонении части поездов на параллельные участки дороги. [1,2].

[1] *Александр Алексеевич Лазарев, доктор физико-математических наук, (jobmath@mail.ru).*

[2] *Наиль Фаридович Хуснуллин, аспирант, (nhusnullin@gmail.com).*

1. Постановка задачи

Рассматривается задача управления движением составов между пунктами A и B (рис.1а). Сеть железных дорог на рассматриваемом узле разбивается на участки. Границами участков являются путевые семафоры. На каждом из них в одном направлении может находиться не более одного железнодорожного состава. Участки пронумерованы слева направо номерами $1, 2, ..., n$, $\mu = k_1, ..., k_n$ -– множество участков. Верхняя колея предназначена для движения поездов идущих слева направо, а нижняя для встречного движения. Поезда не имеют возможности двигаться назад и могут совершать переезды на соседнюю ветку на семафорах, если участок закрыт на ремонт. Для общности положим, что p – время движения по участку для всех поездов равно.



а) *б)*

Рис. 1. Двухколейная железная дорога, разделенная семафорами

Пусть существует расписание π', предписывающее всем поездам, идущим по направлению к станции назначения проезжать каждый семафор согласно расписанию.Вычисляемое в рамках задачи расписание движения составов учитывает изначально заданные интервалы недоступности отдельных участков, которые задаются в виде набора множеств $E = [t'_{i1}, t''_{i1}], [t'_{i2}, t''_{i2}], ..., i \in \mu$.

Пусть N – множество составов, для которых необходимо составить оперативное расписание движения на рассматриваемом узле. Каждому из составов $j \in N$ назначается приоритет (вес) ω_j, который учитывается в процессе построения расписания движения составов. На первом этапе будем рассматривать задачу, когда $\omega_j = 1, \forall j \in N$.

Рейсом в задаче будем называть тройку (s, t, j), где s – сегмент на котором находится j поезд в момент времени t. Допустимым расписанием задачи будем называть последовательность реисов (s_m, t_m, j), где $m = 1, ..., M$, где M – число рейсов таких, что

$$t_m >= t_{m-1} + p, m = 2, ..., M,$$
$$t_m^i \notin E_i, j \in \mu.$$

Будем говорить, что система находится в состоянии $S(t, s_0, ..., s_j) = S(t, s_0, ..., s_l, s_{l+1}, ..., s_{l+r})$, если в момент времени t поезда идущие слева направо (по верхней колее) находятся на участках $s_0, ..., s_l$, а поезда идущие справа налево (по нижней колее) на $s_{l+1}, ..., s_{l+r}$, где

- l -– количество поездов идущих из пункта A в B;
- r -– количество поездов идущих из B в A;
- $L = \{0, ..., l\}, R = \{0, ..., r\}$;
- $s_j \in \{0, ..., n\}, j \in \{0, ..., l + r\}$.

В дальнейшем $s_0, ..., s_l$ будем называть левой группой, а $s_{l+1}, ..., s_{l+r}$ правой группой. Пусть в начальный момент времени значение переменных входящих в левую группу равно 0, а значение переменных правой группы установим равным максимально возможному $n+1$, где n -– количество участков. Конечным состоянием считается состояние, в котором значение всех переменных левой группы равно $n + 1$, а правой соответственно 0.

Пусть $C(\pi)$ и $C'_j(\pi')$ -– значения окончания прохождения участка AB (или BA) поездом j для нормативного и вариантного графиков движения поездов соответственно. Необходимо сформировать новое расписание движения поездов π. Можно рассмотреть несколько вариантов целевой функции:

- Минимизация суммарного запаздывания:
$$min \sum_{j \in N} \omega_j max\{0, C_j(\pi) - C'_j(\pi')\};$$

- Минимизация среднего времени нахождения состава в пути:
$$min \sum_{j \in N} C_j(\pi);$$

- Минимизация максимального временного смещения:
$$minmax_{j \in N}\{C_j(\pi) - C'_j(\pi')\}.$$

Стоит отметить, что для решения задачи предлагаемым методом можно использовать любую регулярную целевую функцию.

2. Описание алгоритма

Шагом будем называть этап алгоритма, в котором происходит генерация возможных состояний для следующего момента времени основываясь на информации из предыдущего состояния. Для решения поставленной задачи строится граф состояний. Так как разрешение спорной ситуации в узком месте происходит с участием не более двух поездов, то дерево решений в общем виде будет бинарным.

Пример 1. Пусть в каждом направлении идут по три поезда и предположим, что второй участок закрыт для движения слева направо, $s_{break} = 2$. Тогда поезда обоих направлений будут проезжать второй участок по нижней колее (рис.1б). На одном участке не может быть более одного поезда, если на нем закрыта одна из веток для движения. Следовательно, при формировании состояния S значение $s_{break} = 2$ может встретиться лишь раз. Если в левой стороне встречается $s_{break} = 2$, то в правой стороне мы уменьшаем на 1 лишь до первой переменной, которая будет равна s_{break} , т.е. не возможно следующее состояние: $S(3, 2, 1, 2, 3, 4)$. Считается, что поезд проехал узкое место, если:

- $s_1 > s_{break}$ –- для поездов идущих слева направо;

- $s_r < s_{break}$ –- для поездов идущих справа налево.

Алгоритм генерации состояний будет следующим:

- $S(s_1, s_2, s_3, s_4, s_5, s_6) = S(0, 0, 0, 5, 5, 5)$ -- начальное состояние (чтобы не усложнять описание алгоритма, переменную t временно опустим из рассмотрения);

- $S(1, 0, 0, 4, 5, 5)$ -- 1 шаг — на 1 увеличивается s_1, на 1 уменьшается s_4;

- $S(2, 1, 0, 3, 4, 5)$ -- 2 шаг -- на 1 увеличивается s_1, s_2, на 1 уменьшается s_4, s_5;

- в следующий момент времени возникнет спорная ситуация в узком месте, кто-то кого-то должен пропустить. На 3 шаге ($i = 3$) генерируется 2 состояния, применяя следующее правило: если поезд вышедший из пункта B пропускает встречный поезд, на 1 увеличивается $s_1, s_2, ..., s_{m_l}$, где $m_l = min\{i, l\}$ и на 1 уменьшается $s_{l+1}, s_{l+2}, ..., s_{m_r}$, где $m_r = min\{i, r\}$ и $s_{m_r} > s_{break}$. В свою очередь, если поезд вышедший из пункта A, ожидает освобождения участка s_{break}, то на 1 уменьшается $s_{l+1}, s_{l+2}, ..., s_{m_r}$, где $s_{m_r} = min\{i, r\}$ и на 1 увеличивается $s_1, s_2, ..., s_{m_l}$, где $s_{m_l} = min\{i, l\}$ и $s_{m_l} < s_{break}$;

- ...

Достижение состояния $S(5, 5, 5, 0, 0, 0)$ является условием остановки алгоритма. •

3. Алгоритм генерации состояния

1) Полагаем, что аргументами функции являются: состояние $State$, полученное на предыдущем шаге и переменная i – номер текущего шага. Также полагаем, что $S = State$.

2) Элементы массива S с $j = 1$ до $min\{i, l\}$, где l – количество поездов идущих из пункта A в B , увеличиваем на 1; элементы с $j = l + 1$ до $min\{l + i, l + r\}$, где r – количество поездов идущих из B в A , уменьшаем на 1.

3) Если в полученном состоянии S значение S_{break} встречается не более одного раза, то возвращаем состояние S, и выходим из функции. В противном случае переходим к пункту 4.

4) Полагаем,что $S_1 = State$ и $S_2 = State$.

5) Элементы массива S_1 с $j = 1$ до $min\{i, l\}$ увеличиваем на 1; элементы с $j = l + 1$ до $min\{l + i, l + r\}$ уменьшаем на 1, при условии, что значение $S_1[j] + 1 > S_{break}$.

6) Элементы массива S_2 с $j = 1$ до $min\{i, l\}$ увеличиваем на 1, при условии, что значение $S_2[j] + 1 < S_{break}$; элементы с $j = l + 1$ до $min\{l + i, l + r\}$ уменьшаем на 1.

7) Полученные массивы S_1, S_2 и номер текущего шага $i = i+1$ являются результатами работы функции.

4. Заключение

В данной работе удалось получить точный алгоритм решения задачи ремонта участков железной. Предлагаемую идею, также можно использовать для решения задачи планирования ремонта участков на железной дороге выбирая отрезок времени, когда это выгодно с экономической точки зрения. Стоит отметить, что данный подход может быть использован для любых регулярных функций. Для апробации полученных результатов использовался вариантный график движения поездов при производстве работ на Северной и Октябрьской железных дорогах за 2009-2010 г. Был выбран участок дороги между станциями Кошта и Бабаево, состоящий из 9 семафоров. Согласно нему в сутки в каждую сторону отправляется порядка 50 составов. Текущая реализация алгоритма на 8 ядерном персональном компьютере позволяет найти решение для 80 составов в каждую сторону (160 в сумме, соответственно) за 30 минут. Работа выполнена при поддержке Российского фонда фундаментальных исследований (13-01-12108).

Литература

1. ГАФАРОВ Е.Р., КВАРАЦХЕЛИЯ А.Г., ЛАЗАРЕВ А.А., МУСАТОВА Е.Г. *Теория расписаний. Задачи железнодорожного планирования.* - М.: ИПУ РАН, 2012. – С.92.
2. ГАФАРОВ Е.Р., КВАРАЦХЕЛИЯ А.Г., ЛАЗАРЕВ А.А., МУСАТОВА Е.Г. *Теория расписаний. Задачи управления транспортными системами.* - М.: Физический факультет МГУ им. М.В.Ломоносова, 2012. – С.160.

OPERATIONAL MANAGEMENT OF THE REPAIR WORK ON THE DOUBLE-TRACK RAILWAY

Alexander Lazarev, Institute of Control Sciences of RAS, Moscow, Doctor of Science, professor (jobmath@mail.ru).
Nail Khusnullin, Institute of Control Sciences of RAS, Moscow (nhusnullin@gmail.com).

Abstract: In this research it was considered the optimal scheduling of the train operation by a double-track railroad when one of the segments is closed (under repair works).

Keywords: theory of schedulling, railway, dynamic programming.