Архипов Д.И., Лазарев А.А., *ИПУ РАН им. В.А. Трапезникова, г. Москва*

Минимизация максимального взвешенного временного смещения для заказов на доставку грузов между двумя станциями в условиях ограниченного движения составов

Введение

Имеются две станции, соединенные двухпутной железной дорогой. Необходимо выполнить множества заказов $N=\left\{J_1^1,\ldots,J_n^1\right\}$ и $N=\left\{J_1^2,\ldots,J_m^2\right\}$ на поставку грузов между станциями. Заказы множества N^1 необходимо доставить с первой станции на вторую, а заказы множества N^2 со второй на первую. Каждый заказ состоит из одного вагона. Все вагоны одинаковые. Так как железная дорога двухпутная, то расписания для множества N^1 и N^2 мы можем составить отдельно. Рассматриваемое множество, не умаляя общности, обозначим за $N=\{J_1,\ldots,J_n\}$. Пусть r_i – *момент поступления* заказа J_i на станцию. Без потери общности предположим, что заказы пронумерованы в порядке их поступления. Каждый заказ имеет свою ценность $w_j>0$.

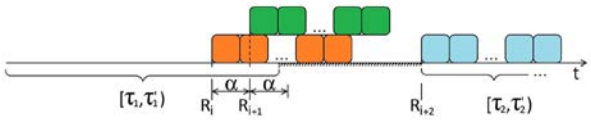

Рис. 1: Допустимые интервалы отправки

Доставка вагонов с одной станции на другую осуществляется поездам $T_1,T_2,\ldots,T_q$, отправляющимися в моменты времени $R_1,R_2,\ldots,R_q$ и состоящими из $k_1,k_2,\ldots,k_q$ вагонов соответственно. Обозначим за p время движения состава между станциями, а за α – время, которое должно разделять моменты отправки двух поездов. Движение пассажирских поездов и плановые работы учитываются в предложенной модели с помощью *допустимых интервалов отправки* $[r_1,r_2'),\ldots,[r_z,r_z')$ (рис. 1). Поезда могут быть отправлены только в момент времени, принадлежащий одному из этих интервалов. Множество всех расписаний, в которых все заказы отправляются после своих моментов поступления и все поезда отправляются не мешая друг другу в допустимые интервалы, обозначим за П. Для каждого заказа определён *директивный срок* $d_j=r_j+\delta$ – момент времени, до которого заказ может быть доставлен на станцию назначения без опоздания, где δ – запас времени на доставку. Пусть C_j – *время доставки* заказа J_j. *Временно́е смещение* заказа обозначим за $L_j=C_j-d_j$. Тогда целевая функция задачи записывается как:

$$\min_{\pi\in\Pi}\max_{j=\overline{1,n}} w_j L_j$$

Задача минимизации общего времени выполнения заказов при ограничении на максимальное взвешенное временное смещение

Решим вспомогательную задачу. Дано множество заказов N и положительное число y. Задача заключается в составлении расписания Θ (N, y), удовлетворяющего условию:

$$\min_{\pi\in\Pi,\ i=\overline{1,q}} R_i(\pi)\ \Big|\ \max_{j=1,n} w_j L_j < y$$

Введем дополнительные обозначения. Для данного значения ограничения y для каждого заказа J_j может быть определен момент времени t_j' до которого данный заказ должен быть отправлен. Из ограничения на максимальное временно́е смещение имеем: $y>w_j\left(C_j-d_j\right)=w_j\left(C_j-r_j-\delta\right)\Rightarrow C_j<\frac{y}{w_j}+r_j+\delta$. Так как после отправки работы требуется ещё некоторое время p, чтобы она доехала до станции назначения, то получаем условие на момент отправки работы j:

$$C_j-p<\frac{y}{w_j}+r_j+\delta\Rightarrow t_j'=\frac{y}{w_j}+r_j+\delta-p \text{ (рис. 2).}$$

Заметим, что для любого значения δ и любого множества значений w_j мы можем свести данную задачу к задаче с весами w_j' и $\delta'=0$ таким образом, что у каждой работы j моменты обязательной отправки t_j' будут теми же. Это может быть получено благодаря преобразованию весов:

$$t_j'=\frac{y}{w_j}+r_j+\delta=\frac{y}{w_j}+r_j\Rightarrow\frac{y}{w_j}+\delta=\frac{y}{w'_j}\Rightarrow w_j'=\frac{y}{\frac{y}{w_j}+\delta}.$$

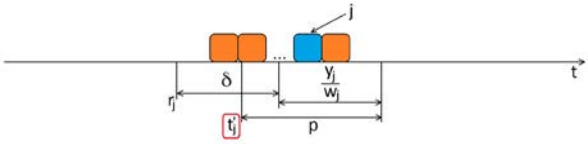

Рис. 2: Момент обязательной отправки

Таким образом, задача может быть сформулирована следующим образом: имеется множество из

$$n=\sum_{i=1}^{q}k_i$$

заказов, для каждого из которых определены момент поступления r_j и момент обязательной отправки t_j'. Цель – перевезти заказы на q поездах, каждый из ко-

торых тратит на дорогу время p так, чтобы каждый заказ был отправлен на поезде, отправляющемся в промежутке $R_m \geq [\max(r(T_m), T_{m-1}+\alpha)]$, и составленное расписание удовлетворяло целевой функции.

Введем функцию [*], определённую на множестве моментов времени t. Для любого момента времени t_j значение фунцкии равно:

$$[t_i] = \min_{t \in [\tau_1, \tau_1'], \ldots, [\tau_z, \tau_z'], t \geq t_i} t,$$

т.е. $[t_j]$ – минимальный момент времени, превышающий t_j, принадлежащий одному из допустимых интервалов отправки.

Введем функцию $r(U)$, определенную на множестве заказов. Для любого множества заказов U значение функции равно: $r(U) = \max_{j \in U} r_j$, т.е. данная функция возвращает момент времени, к которому поступят все заказы из данного множества.

Введем семейство множеств $S_1 \subset S_2 \subset \ldots \subset S_q$ – работы, которые должны уехать первым поездом (S_1), первым или вторым поездом (S_2), первым или вторым или третьим (S_3) и т.д. Для любого $i = \overline{1,n}$ мощность множества (S_i) не может превышать количество заказов, которое можно увезти первыми i поездами, т.е. $|S_i| \leq \sum_{l=1}^{i} k_l = K_l$. Заметим, что до начала работы алгоритма построения Θ (N,y) не пусто только множество S_q, которое состоит из всех n работ. Будем говорить, что расписание π *удовлетворяет* набору множеств $S_1, \ldots, S_q$, если $\forall i = \overline{1,n}$ $i = \overline{1,n}$ выполняется $S_i \in T_1(\pi) \ldots \bigcup T_i(\pi)$.

Рассмотрим дополнения $\overline{S_i} = N \setminus S_i$ ко множествам $S_1, \ldots, S_q$. Из определения множества S_i следует, что любой поезд, с номером больше чем i должен состоять только из заказов, принадлежащих множеству $\overline{S_i}$, т.е. $\forall l, q \geq l > i : T_l \subset \overline{S_i}$.

В расписании Θ(N,y) отправка поезда T_m в момент времени R_m должна удовлетворять следующим свойствам:

После отправки m-ого поезда все заказы из множества S_m должны быть отправлены, т.е. $S_m \subseteq T_1 \bigcup \ldots \bigcup T_m$;

К моменту поступления $r(T_m)$ заказов из T_m должны поступить хотя бы k_m заказов, в том числе все заказы из S_m, т.е. $r(T_m) \geq \max(r(S_m), r_{K_m})$;

Момент времени R_m, определенный для поезда T_m, удовлетворяет следующему неравенству: $R_m \geq [\max(r(T_m), T_{m-1}+\alpha)]$.

Свойство 1 следует из определения S_m.

Свойство 2 следует из того, что для отправки первых mm необходимо, чтобы поступило хотя бы K_m заказов и поступили все заказы из S_m, чтобы было выполнено свойство 1.

Свойство 3. Поезд m не может быть отправлен до того момента, как:

- поступят все работы из T_m;
- m–1-ый поезд освободит пути;
- момент времени будет принадлежать допустимому интервалу отправки.

Алгоритм построения расписания Θ(N,y)

Алгоритм 1. На первом шаге рассматриваем формирование последнего поезда T_q. Учитывая, что $T_q \subseteq \overline{S_{q-1}}$, выбираем из множества S_{q-1} k_q заказов с наибольшими значениями моментов поступления r_j и включаем их в поезд $\overline{T_q} = \{J_{q_1}, J_{q_2}, \ldots, J_{q_{k_q}}\}$, после чего переходим к формированию поезда T_{q-1}.

Так как один заказ не может быть включен в два разных поезда, то в поезд T_{q-1} могут войти только заказы из $T_{q-1} \subseteq (\overline{S_{q-2}} \setminus T_q)$, не включенные в T_q, т.е. $\forall J_j \in T_i : t_j' \geq R_i$ Выбираем k_{q-1} заказов из множества $(\overline{S_{q-2}} \setminus T_q)$ и включаем их в T_{q-1}. На каждом следующем шаге поступаем аналогично. Таким образом, при формировании поезда T_j мы выбираем k_j заказов с наибольшими моментами поступления из множества $\overline{S_{i-1}} \setminus (T_{i+1} \bigcup T_{i+2} \bigcup \ldots \bigcup T_q)$ и включаем их в T_i.

После того, как все поезда сформированы, вычисляем их моменты отправки R_i, пользуясь свойством 3. Последовательно находим значения $R_1 = [\max(r(T_1))], \ldots, R_q = [\max(r(T_q), T_{q-1}+\alpha)]$. Поочередно проверяем все заказы на выполнимость условия отправки: $\forall J_j \in T_i : t_j' \geq R_i$. В случае, если все заказы удовлетворяют данному условию, считаем расписание π, удовлетворяющее целевой функции, найденным, после чего алгоритм прерывает свою работу. В случае, если обнаружены один или несколько заказов $J_1, J_2, \ldots, J_l$, отправляемых в поездах $T_{i_1}, T_{i_2}, \ldots, T_{i_l}$ соответственно, для которых t_j' меньше либо равно моментам отправки их поездов, для каждого из таких заказов выполняется процедура включения во множества $S_{i_1-1}, S_{i_2-1}, \ldots, S_{i_q-1}$ соответственно. Так как $S_1 \subseteq S_2 \subseteq \ldots \subseteq S_q$, после включения каждого заказа J_x во множество S_{i_x} происходит включение заказа J_x во все множества $S_{i_x+1}, \ldots S_q$ (рис. 3). В те множества, в которых J_x уже состоит, повторного включения не происходит.

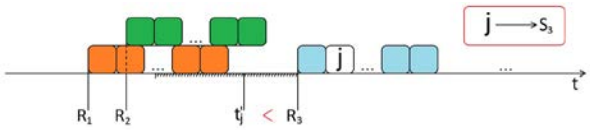

Рис. 3: Включение заказа j в S3

После того, как включение выполнено и новые множества $S_1, S_2, \ldots, S_q$ составлены, проверяем их мощность. В случае, если мощность какого-либо множества превысила допустимое значение, т.е. $|S_i| > K_i$, алгоритм прерывается, делается вывод, что расписания, удовлетворяющего заданной целевой функции, не существует. Если мощности всех множеств корректны, алгоритм повторяет свое действие до тех пор, пока не будет найдено расписание без включений в S_i или пока

одно из множеств S_i не переполнится.

Лемма 1. *Даны множества S_1, S_2,…, S_q. Для данного набора множеств алгоритм 1 сформировал поезда $T_1(\pi), T_2(\pi),\ldots,T_q(\pi)$ и вычислил моменты отправки $R_1(\pi), R_2(\pi),\ldots,R_q(\pi)$. Тогда, для любого расписания π', удовлетворяющего множествам S_1, S_2,…, S_q, для любого номера поезда $i \in 1,\ldots,q$ верно неравенство: $R_i(\pi) \le R_i(\pi')$.*

Доказательство. Будем сравнивать составы поездов расписаний *π и π'*, перебирая их в обратном порядке. Пусть на каком-то шаге, в поезде $T_l(\pi)$ нашелся заказ J_x, который отсутствует в $T_l(\pi')$. В таком случае в поезде $T_l(\pi')$ должен присутствовать заказ J_v, который отсутствует в $T_l(\pi)$.

Заметим, что при предыдущих сравнениях различий обнаружено не было, т.е.

$$T_{l+1}(\pi) \equiv T_{l+1}(\pi'),\ldots,T_q(\pi) \equiv T_q(\pi').$$

Из того, что заказ J_x имеется во множестве $T_l(\pi)$, но отсутствует во множествах $T_l(\pi')$, $T_{\{l+1\}}(\pi'),\ldots,T_q(\pi')$ следует, что он состоит во множестве $T_a(\pi')$, где $a<l$. Аналогичные рассуждения проводим для заказа J_v, делая вывод, что J_v содержится во множестве $T_b(\pi)$, где $b<l$. Заметим, что так как при формировании поезда $T_l(\pi)$ алгоритм 1 выбирал заказы с наибольшими значениями r_j, можно сделать вывод, что $r_{J_x} \ge r_{Jv}$, и так как $J_x \in T_a(\pi')$, то $R_a(\pi') \ge r_{J_x} \ge r_{J_v}$.

Кроме того, J_x, $J_v \in S_{l-1}$, так как оба расписания удовлетворяют набору множеств S_1, S_2,…, S_q. Отсюда следует, что в расписании π' можно поменять заказы J_x и J_v местами без сдвига моментов отправки поездов $R_1(\pi'), R_2(\pi'),\ldots,R_q(\pi')$, и расписание будет удовлетворять набору множеств S_1, S_2,…, S_q. Последовательно выполняя аналогичные операции, получаем расписание π'', удовлетворяющее S_1, S_2,…, S_q, с поездами $T_1(\pi), T_2(\pi),\ldots,T_q(\pi)$ и моментами отправки $R_1(\pi'), R_2(\pi'),\ldots,R_q(\pi')$. Из свойства 3 следует, что $\forall i = 1,n : R_i(\pi'') \ge [\max(r(T_i), T_{i-1}+\alpha)] = R_i(\pi)$, т.е. $R_i(\pi') = R_i(\pi'') \ge R_i(\pi)$. Лемма доказана.

Теорема 1. *Алгоритм 1 строит расписание $\Theta(N,y) \in \prod$, удовлетворяющее критерию*

$$\min_{\pi \in \prod,\ i=\overline{1,q}} R_i(\pi) \mid \max_{j=\overline{1,n}} w_j L_j < y$$

Если алгоритм 1 прервал свою работу, то не существует такого расписания π, для которого верно $(wL)_{max} < y$.

Доказательство. Пусть существует расписание π^* удовлетворяющее критерию

$$\min_{\pi \in \prod,\ i=\overline{1,q}} R_i(\pi) \mid \max_{j=\overline{1,n}} w_j L_j < y \cdot$$

Сформируем для него множества

$$S_1(\pi^*), S_2(\pi^*),\ldots,S_q(\pi^*)$$

с помощью операции включения:

$S_i(\pi^*) = \{J_{i_1},\ldots,J_{il}\}$ – все заказы, для которых выполнено неравенство $R_{i+1}(\pi^*) > t'_{j_i}$.

Рассмотрим работу алгоритма 1. Будем строить расписание π, пока $\forall i = 1,\ldots,n$ выполняется $S_i(\pi) \subseteq S_i(\pi^*)$. Если алгоритм закончит свою работу, и данное условие будет выполнено, то, по лемме 1, построенное расписание будет удовлетворять критерию

$$\min_{\pi \in \prod,\ i=\overline{1,q}} R_i(\pi) \mid \max_{j=\overline{1,n}} w_j L_j < y \cdot$$

Предположим, что на каком-то шаге нарушилось условие, и для какого-то $i \in 1,\ldots,n$ стало верно $S_i(\pi) \subsetneq S_i(\pi^*)$. Это значит, что в расписании π, построенном на данном шаге алгоритма, $\exists J \in T_l(\pi)$, где $R_{i+1}(\pi) > t'_j$, причем $J \notin S_i(\pi^*)$. Заметим, что так как расписания π и π^* удовлетворяют $S_1(\pi), S_2(\pi),\ldots,S_q(\pi)$, то по лемме 1 $R_{i+1}(\pi^*) \ge R_{i+1}(\pi) > t'_j$, следовательно, заказ $J \in S_i(\pi^*)$ – противоречие.

Алгоритм построения оптимального расписания

Алгоритм 2. Для построения оптимального расписания π(*N*) будем действовать следующим образом. Построим расписание, в котором заказы отправляются по возрастанию моментов поступления. Тем самым мы получим расписание, удовлетворяющее условию минимума R_i для каждого поезда *i*. Рассмотрим заказ, на котором достигается максимум целевой функции (w_{j1} L_{j1}). Пусть этот заказ в данном расписании отправляется m_1-ым поездом. Так как алгоритм 1 строит расписание с минимальными моментами отправками поездов из всех возможных, удовлетворяющих критерию $(wL)_{\max} < y$, то при уменьшении значения y для существования расписания необходимо, чтобы заказ, на котором была достигнута целевая функция, был отправлен поездом с меньшим порядковым номером. Следовательно, для того, чтобы улучшить целевую функцию, необходимо отправить заказ *j* одним из поездов, отправившимся до поезда с номером m_1, то есть мы должны включить работу *j* во множество S_{m_1-1}. Это значит, что если мы построим расписание Θ(*N*, w_{j1} L_{j1}), то в нем заказ *j* будет отправлен одним из поездов, номер которого меньше m_1, а целевая функция будет меньше, чем w_{j1} L_{j1} и достигается на заказе j_2, который отправлен поездом m_2. Следующим шагом мы строим расписание Θ(*N*, w_{j2} L_{j2}), в котором заказ j_2 должен быть отправлен поездом с номером меньше, чем m_2, а значит заказ j_2 попадет во множество S_{m_2-1}. Будем повторять эту процедуру до тех пор пока не наступит такой шаг *s*, что расписания Θ(*N*, w_{j2} L_{j2}) не существует. После чего говорим, что не существует расписания со значением целевой функции меньшем, чем w_{js} L_{js} и при этом существует расписание, полученное на s-м шаге со значением целевой функции w_{js} L_{js}, следовательно оно и будет оптимальным.

Теорема 2. *Алгоритм 2 строит расписание π^*, оптимальное по критерию $(wL)_{\max}$ и имеющее минималь-*

ные моменты отправки всех поездов R_i среди всех расписаний с таким значением $(wL)_{\max}$.

Доказательство. Расписание $\pi^* = \Theta(N, y)$ имеет значение целевой функции, равное y'. Из несуществования расписания $\Theta(N, y')$ по теореме 1 следует, что расписаний, удовлетворяющих критерию $(wL)_{\max} < y'$, не существует, а значит расписание π^* оптимально по критерию $(wL)_{\max}$. Из леммы 1 следует, что расписание $\pi^* = \Theta(N, y)$ имеет минимальные моменты отправки всех поездов R_i среди всех расписаний со значением целевой функции $(wL)_{\max} < y'$. Теорема доказана.

Оценка сложности алгоритма

На каждом шаге алгоритма 1 происходит добавление заказа в одно или несколько множеств $S_1, S_2, \ldots, S_{q-1}$. Так как алгоритм 1 строит расписание с минимальными моментами отправками из всех, удовлетворяющих критерию $(wL)_{\max} < y$, то при уменьшении значения y для существования расписания необходимо, чтобы заказ, на котором была достигнута целевая функция, был отправлен поездом с меньшим порядковым номером. Следовательно, на каждом шаге алгоритма 2 происходит включение заказа, на котором достигнута целевая функция в одно из множеств $S_1, S_2, \ldots, S_{q-1}$, при этом выход заказа из каждого из этих множеств невозможен. Следовательно, суммарное число шагов алгоритмов 1 и 2 не превышает суммы мощностей множеств $S_1, S_2, \ldots, S_{q-1}$, которая равна $K_1 + K_2 + \ldots + K_q = O(n^2 / k_{min})$. На каждом шаге алгоритма 1 выполняется подбор заказов во множества $T_q, T_{q-1}, \ldots, T_1$, на каждом шаге рассматривается не более чем $K_q, \ldots, K_2, K_1$ заказов-кандидатов соответственно. Следовательно, сложность шага алгоритма 1 равна $K_1 + K_2 + \ldots + K_q = O(n^2 / k_{min})$. Из приведенных рассуждений следует, что общая трудоёмкость алгоритма 2 составляет порядка $O(\frac{n^4}{k_{min}^{\ 2}})$ операций.

Литература

1. B.B.Simons. (1978). A fast algorithm for single processor scheduling. Proceedings of the 19 IEEE Annual Symposium on Foundations of Computer Science (стр. 246-252). New York: Ann. Arbor. Mich.
2. J.A.Hoogeven. (1996). Minimizing maximum promptness and maximum. Math. Oper. Res., 100-114.
3. А.А.Лазарев. (2008). Теория расписаний. Оценка абсолютной погрешности и схема приближённого решения задач теории расписаний. Москва: МФТИ.
4. Лазарев, А. А., Архипов, Д. И., & Карпов, И. В. (2010). Polynomially Solvable Case of the NP-Hard Problem 1|r_j|L_{max}. International Conference on Project Management and Scheduling, (стр. 289-293). Tours.
5. Лазарев, А. А., Архипов, Д. И., & Мусатова, Е. Г. (2012). Задача минимизации максимального взвешенного временного смещения выполнения заказа для двух станций. Труды и пленарные доклады участников конференции УКИ'12 (стр. 1962-1967). Москва: ИПУ РАН.