ЭКОНОМИКА
И МАТЕМАТИЧЕСКИЕ МЕТОДЫ

1969
том V, вып. 6

О ПОСТРОЕНИИ ОПТИМАЛЬНЫХ ГРАФИКОВ ОБОРОТА САМОЛЕТОВ

Б. И. КОГАН, Г. Ф. ЯНБЫХ

(Рига)

ВВЕДЕНИЕ

В статье рассматривается транспортная сеть из n аэропортов и p связывающих их рейсов.

Задача заключается в том, чтобы обеспечить выполнение заданного множества рейсов P_1 и некоторого подмножества $P_2' \subseteq P_2$, где $P_1 \bigcup P_2 = 1, \ldots, p$, $P_1 \bigcap P_2 = \varnothing$, указав, какие рейсы должен выполнить каждый самолет $i \in M = 1, \ldots, m$ в течение заданного периода T. Множество P_2' должно быть выбрано таким образом, чтобы все рейсы $k \in P_2'$ приносили максимальный доход.

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ И ПОСТАНОВКА ЗАДАЧИ

Пронумеруем аэропорты числами $j \in N = 1, \ldots, n$, а рейсы — числами $k \in P = 1, \ldots, p$.

Количество рейсов, соединяющих аэропорты j_1 и j_2 в прямом направлении (j_1, j_2), равно количеству рейсов, связывающих эти аэропорты в обратном направлении (j_2, j_1).

В дальнейшем рассматриваются только ежедневные рейсы. Можно показать, что α неежедневных рейсов приводятся к β условным ежедневным рейсам $(\beta < \alpha)$.

Будем полагать, что все рейсы $k \in P$ выполняются самолетами одного типа. Имеющееся множество самолетов пронумеруем числами $i \in M = 1, \ldots, m$.

Базовыми назовем аэропорты $j \in N^* \subseteq N$, в которых базируются самолеты $i \in M_j = 1(j), \ldots, m(j)$, где

$$\bigcup_{j \in N^*} M_j = M, M_j \bigcap M_{j'} = \varnothing, \quad j \neq j', j, j' \in N^*,$$

выполняющие совокупность рейсов $P(j)$.

$$\bigcup_{j \in N^*} P(j) = P, P(j) \bigcap P(j') = \varnothing, \quad j \neq j', j, j' \in N^*.$$

Каждому рейсу $k \in P$ поставим в соответствие вектор

$$R_k = (\tau_k, t_k, \Delta t_k, j_k, l_k, C_k),$$

где τ_k — продолжительность рейса k, включающая межрейсовую стоянку; t_k — желаемое время вылета рейса k; $\pm\Delta t_k$ — допустимый интервал сдвига t_k; j_k, l_k — соответственно начальный и конечный аэропорты рейса k; C_k — весовой коэффициент, характеризующий эффективность рейса k.

Описанную транспортную сеть можно интерпретировать как n-вершинный граф (N, P). (Здесь и далее используется известная терминология теории графов [1]). Вершины графа соответствуют аэропортам N, а дуги — рейсам P.

Множество рейсов

$$S = \{k_1, k_2, \ldots, k_r\}, \quad k_\alpha \in P, \ \alpha \in 1, \ldots, r,$$

удовлетворяющее условиям

$$j_{k_\alpha} \equiv l_{k_{\alpha-1}}, \quad \alpha \in 2, \ldots, r; \tag{1}$$

$$t'_{k_\alpha} + \tau_{k_\alpha} \leqslant t_{k_{\alpha+1}} + \Delta t_{k_{\alpha+1}}, \quad \alpha \in 1, \ldots, (r-1), \tag{2}$$

где

$$t_{k_\alpha}' = \begin{cases} \max\left[(t_{k_\alpha} - \Delta t_{k_\alpha}), (t_{k_{\alpha-1}}' + \tau_{k_{\alpha-1}})\right], \Leftarrow \alpha \in 2, \ldots, r, \\ t_{k_\alpha} - \Delta t_{k_\alpha}, \Leftarrow \alpha = 1; \end{cases}$$

$$k_\alpha \neq k_\beta, \quad \alpha, \beta \in 1, \ldots, r, \ \alpha \neq \beta; \tag{3}$$

$$\bigcup_{\alpha \in 1, \ldots, r} \{j_{k_\alpha}, l_{k_\alpha}\} \cap N^* \neq \varnothing; \tag{4}$$

$$(t_{k_r}' + \tau_{k_r}) - (t_{k_1}'' - \tau_{k_1}) \leqslant T; \tag{5}$$

при

$$t_{k_\alpha}'' = \begin{cases} \min\left[(t_{k_\alpha} + \Delta t_{k_\alpha} + \tau_{k_\alpha}), \ (t''_{k_{\alpha+1}} - \tau_{k_{\alpha+1}})\right], \Leftarrow \alpha \in 1, \ldots, (r-1), \\ t'_{k_\alpha} + \tau_{k_\alpha}, \Leftarrow \alpha = r; \end{cases}$$

$$j_{k_1} \equiv l_{k_r}, \tag{6}$$

будем называть допустимым контуром на графе (N, P) или графиком оборота.

График оборота S может выполняться самолетом, базирующимся в одном из аэропортов $j \in N_S^*$, где

$$N_S^* = N^* \bigcap_{k \in S} \{j_k, l_k\}.$$

Разбиением B_m графа (N, P) назовем семейство $S_1, S_2, \ldots, S_m$, отвечающее требованиям

$$S_\alpha \neq \varnothing, \quad \alpha \in 1, \ldots, m; \tag{7}$$

$$S_\alpha \cap S_\beta = \varnothing, \ \alpha, \beta \in 1, \ldots, m, \ \alpha \neq \beta; \tag{8}$$

$$\bigcup_{\alpha \in 1, \ldots, m} S_\alpha = P_1 \cup P_2'. \tag{9}$$

Замечание 1. Рейсы $k \in P_2$ назовем рейсами-претендентами. Это могут быть, например, рейсы, которые не выполнялись в предшествующей навигации и рекомендованы к выполнению в наступающей.

Представим разбиение B_m и множество самолетов M в виде простого [1] графа (B_m, M, Γ), где отображение Γ определено $M_j \in \Gamma S, \Leftarrow j \in N_S^*$.

Тогда необходимое и достаточное условие того, что граф (B_m, M, Γ) имеет паросочетание и что графики оборота S можно выполнить различными самолетами без их перебазирования, будет [1]

$$|\Gamma A| \geqslant |A|, \quad A \subset B_m. \tag{10}$$

Разбиение B_m^* назовем оптимальным, если оно удовлетворяет (10) и

$$\sum_{k \in P_2'} C_k = \max. \tag{11}$$

ЭКОНОМИКО-МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Обозначим через P_j^+ множество рейсов, заканчивающихся в аэропорту $j \in N$, и через P_j^- — множество рейсов, начинающихся в аэропорту $j \in N$.

Введем переменные

$$x^i_{k,\alpha} = \begin{cases} 1, \text{ если } i\text{-й самолет выполняет рейс } k \text{ с порядковым номером } \alpha \\ \text{в графике оборота,} \\ 0 \text{ — в противном случае} \end{cases}$$

Задача построения искомых графиков оборота сводится к задаче линейного целочисленного программирования при ограничениях

$$\sum_{k \in P} \left[(t_k + \Delta t_k) x^i_{k,\alpha+1} - (t_k - \Delta t_k) x^i_{k,\beta} - \sum_{\varkappa \in \beta, \ldots, \alpha}^{\alpha} \tau_k x^i_{k,\varkappa} \right] \geqslant 0, \qquad (12)$$

$$i \in M, \quad \alpha \in \beta, \ldots, \alpha_{\max} - 1, \quad \beta = 1, \ldots, \alpha_{\max} - 1,$$

где $\alpha_{\max}$ наибольший возможный порядковый номер рейса в графике оборота.

$$\sum_{\alpha \in 1, \ldots, \alpha_{\max}} \left(\sum_{k \in P_j+} x_{k,\alpha} - \sum_{k \in P_j-} x^i_{k,\alpha} \right) = 0, \quad i \in M, j \in N; \qquad (13)$$

$$\sum_{k \in P_j+} (x^i_{k,\alpha+1} + x^i_{k,1}) = \sum_{k \in P_j-} x^i_{k,\alpha}, \quad \begin{matrix} i \in M, j \in N, \\ \alpha \in 1, \ldots, (\alpha_{\max} - 1); \end{matrix} \qquad (13a)$$

$$\sum_{\alpha \in 1, \ldots, \alpha_{\max}} \sum_{i \in M} x^i_{k,\alpha} = 1, \quad k \in P_1; \qquad (14)$$

$$\sum_{\alpha \in 1, \ldots, \alpha_{\max}} \sum_{i \in M} x^i_{k,\alpha} \leqslant 1, \quad k \in P_2; \qquad (15)$$

$$\sum_{\alpha \in 1, \ldots, \alpha_{\max}} \sum_{\substack{k \in P_j- \\ j \in N^*}} x_{k,\alpha} \geqslant 1, \quad i \in M. \qquad (16)$$

Максимизируется функционал

$$F = \sum_{\alpha \in 1, \ldots, \alpha_{\max}} \sum_{i \in M} \sum_{k \in P_2} x_{k,\alpha} C_k. \qquad (17)$$

Остановимся кратко на возможностях существующих методов решения целочисленных задач линейного программирования с переменными, принимающими значения 0 или 1.

В [2] описан аддитивный алгоритм решения задач с булевыми переменными. Алгоритм реализован на ЭВМ «Урал-4». Время решения задачи размером 49 × 16 составляет 4 часа.

Специальный вид матрицы ограничений (12) — (16) позволяет решать задачи большего размера (до 150 ÷ 200 переменных) в течение приемлемого времени. В [3] предлагается модификация аддитивного алгоритма, в некоторых случаях приводящая к лучшим результатам.

Потребности практики выдвигают задачи с числом переменных 1000 ÷ ÷ 10 000 и более. В связи с этим возникает необходимость применения более быстродействующих ЭВМ с бо́льшим объемом оперативной памяти. Не менее важно также интенсивно разрабатывать блочные методы решения задач указанного класса [4], создавать методы отыскания решений, близких к оптимальному, а также искать пути непосредственного поиска решения с более ограниченным перебором, чем перебор в известных методах.

Предлагается один из таких методов, позволяющий значительно сократить перебор.

МЕТОД ПРЯМОГО ПОИСКА ОПТИМАЛЬНЫХ РАЗБИЕНИЙ ГРАФА (N, P)

Проблема построения оптимальных графиков оборота сводится к решению следующих задач: 1) определение всех возможных графиков оборота; 2) определение разбиения B_m^*. Решение первой задачи не представляет труда. Можно воспользоваться известными методами, например [1, 4, 5] и др., рассматривая только такие контуры, которые удовлетворяют условиям (2) — (4).

Остановимся более подробно на отыскании разбиения B_m^*.

Пусть построены все возможные графики оборота. Пронумеруем их числами $\varepsilon \in E = 1, \ldots, \sigma$.

Определим множества I_k

$$I_k = \{\varepsilon / k \in S_\varepsilon,\ \varepsilon \in E\},\ k \in P. \tag{18}$$

Введем граф (E, Γ), отображение которого Γ определяется следующим образом.

$$\Gamma_\varepsilon = \bigcap_{k \in S_\varepsilon} (E \setminus I_k), \quad \varepsilon \in E.$$

Оптимальное разбиение соответствует некоторому полному m-вершинному подграфу $G_m^*, = (V, \Gamma_V)$ удовлетворяющему (9), (11).

Очевидно,

$$V \cap I_k \neq \varnothing, \quad k \in P_1. \tag{19}$$

На основании (19) можно исключить из рассмотрения такие S_{ε^-}, для которых выполняется соотношение

$$(\varepsilon^- \cup \Gamma_{\varepsilon^-}) \cap I_k = \varnothing \tag{20}$$

хотя бы для одного $k \in P_1$.

После удаления из графа (E, Γ) некоторого множества вершин $\{\varepsilon^-\}$ вновь определяем I_k в соответствии с (18), заменяя множество E на множество $E \setminus \{\varepsilon^-\}$, и опять осуществляем проверку согласно (20), вычеркивая такие S_{ε^-}, для которых (20) выполняется.

Эту процедуру следует повторить до тех пор, пока не найдется ни одного ε^-, которое можно исключить из рассмотрения.

Замечание 2. Если в результате вычеркивания $\{\varepsilon^-\}$ множество E окажется пустым, то не существует ни одного разбиения B_m.

В этом случае необходимо, изменив исходные данные (например, Δt_k, t_k и др.), попытаться повторно построить множество E.

Далее будем использовать обозначение (a) — множество (число), находящееся в ячейке памяти ЭВМ с номером a.

Приступим к описанию алгоритма.

0. Полагаем $u = 0$, $q = 0$, $z = 0$, $B^0 = \varnothing$, $P^0 = \varnothing$, $L^0 = E$, $(c) = \infty$.
1. Выбираем некоторое $k_z \in P \setminus P^q$. Полагая $Q_u{}^q = \varnothing$, переходим к 2.
2. Получаем множество $D_u = I_{k_z} \cap L^q \setminus Q_u{}^q$.

2а. Если $D_u = \varnothing$, переходим к 3.
2б. Если $D_u \neq \varnothing$, переходим к 4.
3. Уменьшаем z на единицу и переходим к 9.
4. Увеличив q на единицу, выбираем $\varepsilon^q \in D_u$, получим множество $B^q = B^{q-1} \cup \varepsilon^q$.
5. Проверяем выполнение соотношения $m \geqslant q$.
5а. Если $m > q$, переходим к 6.
5б. Если $m = q$, переходим к 11.
6. Получаем $L^q = L^{q-1} \cap \Gamma\varepsilon^q$.
7. Проверяем выполнение соотношений $L^q \cap I_k \neq \varnothing$, $k \in P_1 \setminus P^q$, где

$$P^q = P^{q-1} \cup S_{\varepsilon^q}. \tag{21}$$

7а. Если (21) удовлетворяется для всех $k \in P_1 \setminus P^q$, переходим к 8.
7б. Если (21) не удовлетворяется хотя бы для одного $k \in P_1 \setminus P^q$, переходим к 9.
8. Проверяем выполнение соотношения

$$\sum_{k \in A_q} C_k < (c), \tag{22}$$

где $A_q = \{k \,/\, I_k \cap L^q = \varnothing, \quad k \in P_2 \setminus P^q\}'$
8а. Если (22) удовлетворяется, переходим к 1, увеличивая z на единицу и полагая $u = 0$.
8б. Если (22) не удовлетворяется, переходим к 9.
9. Уменьшаем q на единицу.
9а. Если $q \geqslant 0$, получаем

$$B^q = B^{q+1} \setminus \varepsilon^{q+1}, \quad L^q = \bigcap_{\varepsilon \in B^q} \Gamma_\varepsilon, \quad P^q = P^{q+1} \setminus S_\varepsilon^{q+1} \tag{23}$$

и переходим к 10.
9б. Если $q < 0$, переходим к 15.
10. Увеличиваем u на единицу, получаем $Q_u{}^q = Q_{u-1}^q \cup \varepsilon^{q+1}$ и переходим к 2.
11. Получаем $P^q = P^{q-1} \cup S_{\varepsilon^q}$.
12. Проверяем соотношение

$$P_1 \setminus P^q = \varnothing. \tag{24}$$

12а. Если (24) удовлетворяется, переходим к 13.
12б. Если (24) не удовлетворяется, переходим к 9.
13. Проверяем условия (10).
13а. Если (10) удовлетворяется, переходим к 14.
13б. Если (10) не удовлетворяется, переходим к 9.
14. Проверяем соотношение

$$C^- = \sum_{k \in P_2 \setminus P^q} C_k < (c). \tag{25}$$

14а. Если (25) удовлетворяется, записываем $B_m = B^m$ в ячейку b, а C^- — в ячейку c и переходим к 9.
14б. Если (25) не удовлетворяется, переходим к 9.
15. Построение разбиений заканчивается. Оптимальным является разбиение (b).

Замечание 3. Заменив в соотношениях (22), (25) знак $<$ на $\leqslant$ и запоминая в п. 14а все разбиения B_m в ячейках b^β, $\beta = 1, 2, \ldots$, получим все оптимальные решения.

Остановимся на некоторых возможностях ускорения сходимости алгоритма.

Если нет необходимости в получении точного решения, то, заменив в п. 8 соотношение (22) на соотношение

$$\gamma(c) < \sum_{k \in A_q} C_k, \quad 0 < \gamma < 1,$$

можно значительно сократить перебор.

Алгоритм также небезразличен к выбору k_z в п. 1. Эксперименты на ЭВМ показали, что алгоритм в большинстве случаев сходится быстрее, если выбирать k_z^* такой, что

$$k_z^* = k_z / |I_{k_z^*}| = \min |I_{k_z}|, \quad k_z \in P \diagdown P^q.$$

Для больших m проверку (10) удобнее производить следующим образом [1].

Построим граф (B_m, M, Γ) описанным ранее способом.

Пронумеруем Γ_ε, $\varepsilon \in B_m$ и Γ_i^{-1}, $i \in M$ так, чтобы

$$|\Gamma_{\varepsilon_1}| \leqslant |\Gamma_{\varepsilon_2}| \leqslant \ldots \leqslant |\Gamma_{\varepsilon_m}|, \text{ а}$$

$$|\Gamma_{i_1}| \geqslant |\Gamma_{i_2}| \geqslant \ldots \geqslant |\Gamma_{i_{m-1}}|,$$

тогда достаточное условие существования паросочетания запишется так

$$|\Gamma_{\varepsilon_1}| + |\Gamma_{\varepsilon_2}| + \ldots + |\Gamma_{\varepsilon_\alpha}| > |\Gamma_{i_1}^{-1}| + |\Gamma_{i_2}^{-1}| + \ldots + |\Gamma_{i_{\alpha-1}}^{-1}|,$$

$\alpha \in 1, \ldots, m$.

Описанный алгоритм был реализован на ЭВМ «Минск-22». Время решения задачи с параметрами $p = 40$, $n = 7$, $m = 9$, $\sigma = 179$ составило 22 мин.

ЛИТЕРАТУРА

1. К. Берж. Теория графов и ее применения. М., Изд. иностр. лит. 1962.
2. E. Balas. An additive algorithm for solving linear programs with zero-one variables. Operat. Res., 1965, v. 13, № 4.
3. F. Glover. A. Multiphase-Dual Algorithm for the Zero-One Integer Programming Problems. Operat. Res., 1965, v. 13, № 6.
4. А. Е. Бахтин, А. Б. Горстко. О решении нелинейных экстремальных задач с линейными ограничениями специального вида. В сб. Математическое программирование. М., «Наука», 1966.
5. Ю. И. Ватрусов. Об анализе конечного графа. В сб. [4].

Поступила в редакцию
21 VIII 1967