

Оптимизация, системный анализ и исследование операций

© 2019 г. В.Н. БУРКОВ, д-р техн. наук (vlab17@br.ru)
(Институт проблем управления им. В.А. Трапезникова РАН, Москва),
Л.В. РОССИХИНА, д-р техн. наук (rossihina_lv@mail.ru)
(Воронежский институт Федеральной службы исполнения наказаний, Воронеж),
А.П. ВЬЮНОВ (nvant@mail.ru)
(Центральная нормативно-техническая лаборатория Федеральной службы
исполнения наказаний, Москва),
Л.В. РОГОВАЯ (rogovayala@yandex.ru)
(Научно-исследовательский центр АО “Воронежсинтезкаучук”, Воронеж)

ЗАДАЧА ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ КОМАНД СПЕЦИАЛИСТОВ

Рассматривается задача распределения команд специалистов, совместно выполняющих некоторые работы (услуги по консультированию, оценка соответствия или несоответствия изделий, товаров или услуг необходимым требованиям и т.д.). Команда, выполняющая работу, состоит из множества специалистов различных типов (специальностей), число которых ограничено. Задача заключается в разработке плана работы команд, при котором время выполнения всех работ минимально. Задача сведена к задаче линейного программирования. Рассмотрены также эвристические алгоритмы и частные случаи задачи.

Ключевые слова: команда специалистов, календарный план, набор команд, задача линейного программирования.

DOI: 10.1134/S0005231019010082

1. Введение

Задача распределения ресурсов в управлении проектами и задачи календарного планирования относятся к сложным задачам нелинейной оптимизации. Особой сложностью отличаются задачи распределения ресурсов, в которых каждая работа выполняется несколькими видами ресурсов. Для таких задач в публикациях [1, 2] было введено понятие набора ресурсов.

Предполагается, что ресурсы $\bar{U}_i(t)$ выполняют работу i в определенном соотношении, т.е. $U_{ij}(t) = a_{ij}v_i(t)$. При этом $\bar{U}_i(t)$ — набор ресурсов в момент t , a_{ij} — параметры набора, $v_i(t)$ — мощность набора. Если $v_i(t)$ принимает всего два значения 0 или 1, то будем говорить, что работа выполняется с фиксированной интенсивностью.

Определение 1. Командой специалистов (далее просто команда) называется множество специалистов различного вида, необходимых для выполнения работы.

Отметим, что команда работает как единое целое, поскольку при выполнении работы специалисты взаимодействуют друг с другом. Примерами являются команды управления проектом, создаваемые на период реализации проекта, команда специалистов по проверке изделий на соответствие требуемым условиям, команда специалистов, выезжающая для проверки деятельности различных организаций, например ВУЗов, и др.

Очевидно, что команда — это набор с фиксированной интенсивностью.

В статье рассматривается задача формирования календарного плана работы команд, при котором продолжительность выполнения всех работ минимальна при заданных ограничениях на число специалистов каждого вида.

2. Постановка задачи

Рассмотрим программу из n независимых работ. Каждая работа выполняется командой специалистов различного типа за время τ_i .

Задано число специалистов j -го типа N_j , $j = \overline{1, m}$.

Обозначим через a_{kj} количество специалистов j -го типа, включенных в команду k , $y_k(t) = 1$, если команда k в момент t выполняет работу k , $y_k(t) = 0$ в противном случае. В силу ограниченности числа специалистов каждого типа имеют место ограничения

$$(1) \quad \sum_k a_{kj} y_k(t) \leq N_j, \quad t \in [0, T], \quad j = \overline{1, m},$$

где T — момент завершения выполнения всех работ. Условия выполнения всех работ имеют вид

$$(2) \quad \int_0^T y_k(\tau) d\tau = \tau_i, \quad i = \overline{1, n}.$$

Задача 1. Определить $y_k(t)$, $k = \overline{1, n}$, при которых время T выполнения всех работ минимально при ограничениях (1), (2).

Это динамическая задача дискретной оптимизации, которая является NP -трудной в силу ограничений (1). При сравнительно небольших размерностях (числа работ) задачу можно свести к задаче линейного программирования. Для этого введем понятие набора команд.

Определение 2. Набором команд (далее просто набором) называется совокупность команд Q , такая что выполняются условия:

$$(3) \quad \sum_{i \in Q} a_{ij} \leq N_j, \quad j = \overline{1, m}.$$

Заметим, что команды, входящие в набор, могут работать одновременно.

Определение 3. Максимальным набором называется набор, к которому нельзя добавить ни одной команды, не нарушив ограничения (3).

Обозначим через R множество всех максимальных наборов $N = |R|$ и через x_j продолжительность работы j -го набора. Считаем, что допустимы перерывы в выполнении работ.

Любой набор $i \in Q$ будем описывать вектором $b_i = \{b_{ij}\}$, где $b_{ij} = 1$, если в наборе присутствует команда j , $b_{ij} = 0$ в противном случае.

Задача 2. Определить $x = \{x_j\}$, минимизирующие

$$(4) \quad T(x) = \sum_{j=1}^N x_j$$

при ограничениях $x_i \geq 0$ и

$$(5) \quad \sum_{j=1}^N b_{ij} x_j \geq \tau_i, \quad i = \overline{1, n}.$$

Получилась задача 2 линейного программирования.

Заметим, что число максимальных наборов может быть значительным (порядка 2^n). Отсюда решить большеразмерную задачу линейного программирования известными методами затруднительно.

3. Частный случай

Рассмотрим частный случай задачи 2, когда максимальные наборы содержат все команды за исключением одной. Очевидно, что таких наборов будет n .

Примем, что i -й набор не содержит i -й команды. В этом случае задача принимает вид: минимизировать (2) при ограничениях $x_i \geq 0$, $i = \overline{1, n}$, и

$$(6) \quad \sum_{j \neq i} x_j \geq \tau_i, \quad i = \overline{1, n}.$$

Поскольку $\theta = \sum_i x_i$, то перепишем (4) в виде

$$(7) \quad \theta - x_i \geq \tau_i, \quad i = \overline{1, n}.$$

Упорядочим наборы по убыванию τ_i , т.е. $\tau_{i_1} \geq \tau_{i_2} \geq \dots \geq \tau_{i_n}$.

Покажем, что существует оптимальное решение задачи (4), (6).

1. В случае если $\tau_{i_{n-1}} + \tau_{i_n} \geq \tau_{i_1}$, то

$$\begin{aligned} x_{i_1} &= \frac{\tau_{i_{n-1}} + \tau_{i_n} - \tau_{i_1}}{2}, & x_{i_j} &= 0, & j &= \overline{2, i_{n-2}}, \\ x_{i_{n-1}} &= \tau_{i_n} - x_{i_1} = \frac{\tau_{i_n} - \tau_{i_{n-1}} + \tau_{i_1}}{2}, \\ x_{i_{n-2}} &= \tau_{i_{n-1}} - x_{i_1} = \frac{\tau_{i_{n-1}} - \tau_{i_n} + \tau_{i_1}}{2}, \end{aligned}$$

при этом

$$\theta_{\min} = \frac{\tau_{i_1} + \tau_{i_{n-1}} + \tau_{i_n}}{2}.$$

2. Если $\tau_{i_{n-1}} + \tau_{i_n} < \tau_{i_1}$, то существуют несколько оптимальных решений:

$$\begin{aligned} x_{i_j} &= 0, \quad j = \overline{1, n-2}, \\ x_{i_{n-1}} &\geq \tau_{i_n}, \\ x_{i_n} &\geq \tau_{i_{n-1}}, \\ x_{i_{n-1}} + x_{i_n} &= \tau_{i_1}, \end{aligned}$$

при этом $\theta_{\min} = \tau_{i_1}$.

Исключим все наборы кроме i_1 , i_{n-1} и i . Фиксируем время работы x_{i_1} набора i_1 . Имеем

$$\begin{aligned} x_{i_{n-1}} &\geq \tau_{i_n} - x_{i_1}, \\ x_{i_n} &\geq \tau_{i_{n-1}} - x_{i_1}, \\ x_{i_{n-1}} + x_{i_n} &\geq \tau_{i_1}. \end{aligned}$$

Пусть $\tau_{i_{n-1}} + \tau_{i_n} \geq \tau_{i_1}$. Возьмем $x_{i_{n-1}} = \tau_{i_n} - x_{i_1}$, $x_{i_n} = \tau_{i_{n-1}} - x_{i_1}$, $\theta = x_{i_n} + x_{i_{n-1}} + x_1 = \tau_{i_{n-1}} + \tau_{i_n} - x_{i_1}$.

Продолжительность всех работ θ тем меньше, чем больше x_{i_1} . Имеем

$$x_{i_{n-1}} + x_{i_n} = \tau_{i_n} + \tau_{i_{n-1}} - 2x_{i_1} = \tau_{i_1}.$$

Получаем:

$$\begin{aligned} x_{i_1} &= \frac{\tau_{i_n} + \tau_{i_{n-1}} + \tau_{i_1}}{2}, \\ x_{i_{n-1}} &= \tau_{i_n} - x_{i_1} = \frac{\tau_{i_n} + \tau_{i_{n-1}} + \tau_{i_1}}{2}, \\ x_{i_n} &= \tau_{i_{n-1}} - x_{i_1} = \frac{\tau_{i_{n-1}} - \tau_{i_n} + \tau_{i_1}}{2}, \\ \theta_{\min} &= \frac{\tau_{i_{n-1}} + \tau_{i_n} + \tau_{i_1}}{2}. \end{aligned}$$

Пусть $\tau_{i_{n-1}} + \tau_{i_n} < \tau_{i_1}$. В этом случае $x_{i_1} = 0$, а $x_{i_{n-1}} + x_{i_n} = \tau_{i_1}$. Любые значения $x_{i_{n-1}}$, x_{i_n} , удовлетворяющие условиям

$$\begin{aligned} x_{i_{n-1}} &\geq \tau_{i_n}, \\ x_{i_n} &\geq \tau_{i_{n-1}}, \\ x_{i_{n-1}} + x_{i_n} &\geq \tau_{i_1}, \end{aligned}$$

определяют оптимальные решения задачи с величиной $\theta_{\min} = \tau_{i_1}$.

Заметим, что при полученном решении все задачи от i_2 до i_{n-2} также будут выполнены.

Пример 1. Рассмотрим численный пример. Пусть $\tau_1 = 10$, $\tau_2 = 8$, $\tau_3 = 4$. Поскольку $\tau_2 + \tau_3 > 10$, то $x_1 = \frac{12-10}{2} = 1$, $x_2 = 4 - 1 = 3$, $x_3 = 8 - 1 = 7$, $\theta_{\min} = 11$.

Заметим, что при любой очередности наборов одна работа выполняется с перерывом.

4. Задача минимизации числа прерываний работ

Недостатком рассмотренного выше алгоритма являются возможные прерывания в выполнении работ. Естественно, желательно минимизировать число прерываний. Возможные планы выполнения работ определяются перестановками $\pi = (i_1, i_2, \dots, i_n)$ наборов.

Задача 3. Определить перестановку π , при которой число прерываний работ минимально.

Задача 3 похожа на задачу коммивояжера [3], в которой также необходимо найти перестановку посещения городов. Однако имеется существенная разница. Она состоит в том, что для данной задачи 3 не существует одной таблицы числа прерываний между наборами. Чтобы показать это, рассмотрим пример 2.

Пример 2. Имеются пять работ и получено оптимальное решение из пяти наборов команд. В табл. 1 приведены данные при очередности наборов $\pi_0 = (1, 2, 3, 4, 5)$.

Таблица 1

$j \backslash i$	1	2	3	4	5	τ_i	L_j
1	1				1	9	1
2	1		1			5	0
3		1		1		8	1
4		1		1		7	0
5			1		1	6	1
x_i	5	7	5	3	6	$\theta = 26$	3

Единица в клетках табл. 1 означает, что в данном интервале i выполняется работа j , L_j — число прерываний работы j . Определим общее число прерываний работ. Работа 1, очевидно, прерывается. Работа 2 заканчивается в первом интервале, поэтому не прерывается. Работа 3 прерывается. Работа 4 не прерывается, поскольку заканчивается во втором интервале. Наконец, работа 5 прерывается. Всего имеем три прерывания работ. Невозможность иметь таблицу прерываний (расстояний между городами в задаче коммивояжера) затрудняет применение методов типа метода ветвей и границ, требующих получение нижних оценок. Поэтому применим приближенный метод локальной оптимизации решения задачи минимизации числа прерываний работ. Окрестность решений в методе локальной оптимизации получается путем операций транспозиции (перестановки соседних элементов) исходного решения.

Описание метода.

1. Выбираем начальную перестановку π_0 .
2. Строим окрестность $Q(\pi_0)$, переставляя пары соседних наборов.

3. Определяем лучшее решение в окрестности π^* .

4. Возвращаемся к шагу 2, т.е. строим окрестность $Q(\pi^*)$ и т.д. вплоть до получения локально-оптимального решения.

Далее можно выбрать новое начальное решение и повторить процедуру.

Пример 3. Возьмем данные примера 2. Начальная перестановка $\pi_0 = (1, 2, 3, 4, 5)$. Окрестность $Q(\pi_0)$ содержит четыре перестановки:

$$\pi_1 = (2, 1, 3, 4, 5), \quad \pi_2 = (1, 3, 2, 4, 5), \quad \pi_3 = (1, 2, 4, 3, 5), \quad \pi_4 = (1, 2, 3, 5, 4).$$

Определяем для каждого решения число прерываний работ.

1. $\pi_1 = (2, 1, 3, 4, 5)$. Решение приведено в табл. 2, $L(\pi_1) = 3$.

Таблица 2

$j \backslash i$	1	2	3	4	5	τ_j	L_j
1	0	1	0	0	1	9	1
2	0	1	1	0	0	5	0
3	1	0	0	1	0	8	1
4	1	0	0	1	0	7	0
5	0	0	1	0	1	6	1
x_i	7	5	5	3	6	26	3

Действуя аналогично, получаем наилучшую перестановку в окрестности $\pi_3 = (1, 2, 4, 3, 5)$, $L(\pi_3) = 1$.

Строим окрестность $Q(\pi_3)$:

$$\pi_5 = (2, 1, 4, 3, 5), \quad \pi_6 = (1, 4, 2, 3, 5), \quad \pi_7 = (1, 2, 4, 5, 3).$$

Наилучшие решения в окрестности $\pi_6 = (1, 4, 2, 3, 5)$ и $\pi_3 = (1, 2, 4, 3, 5)$,

$$L(\pi_6) = L(\pi_3) = 1.$$

Поэтому π_1 является локально-оптимальным решением. Можно, конечно, продолжить процедуру для второго лучшего в окрестности решения, но вряд ли получим решение с меньшим $L(\pi)$, поскольку наборы 1 и 5 очень далеки друг от друга и прерывание работы 1 неизбежно.

Попробуем взять другое начальное решение, сблизив наборы 1 и 5, сохранив близость наборов 3 и 5.

Возьмем, например, перестановку $\pi_8 = (3, 5, 1, 2, 4)$. Расчеты приведены в табл. 3.

Таблица 3

$j \backslash i$	1	2	3	4	5	τ_j	L_j
1	0	1	1	0	0	9	0
2	1	0	1	0	0	5	0
3	0	0	0	1	1	8	0
4	0	0	0	1	1	7	0
5	1	1	0	0	0	6	0
x_i	5	6	5	7	3	26	0

Получим оптимальное решение без прерывания работ.

5. Задача назначения команд при запрещении прерывания работ

При запрещении прерывания работ задача назначения команд становится сложной (NP-трудной) задачей дискретной оптимизации.

Для ее решения применяются эвристические алгоритмы, основанные на правилах приоритета работ. Достаточно популярным является правило приоритета работ, согласно которому в первую очередь вычисляются так называемые критические работы [4].

В рассматриваемом случае критические работы — это работы с наибольшей продолжительностью τ_i .

5.1. Описание алгоритма

1. В начальный момент времени $t = 0$ назначаем команды в очередности убывания продолжительностей соответствующих работ до получения максимального набора.
2. В момент окончания какой-либо работы назначаем команды также в очередности убывания продолжительностей еще невыполненных работ.

Пример 4. Имеются пять работ и соответственно пять команд, данные о которых приведены в табл. 4.

Таблица 4

$j \backslash i$	1	2	3	4	5	τ_j
1	1	1	0	1	0	12
2	0	1	1	0	1	10
3	1	0	0	1	0	8
4	0	1	1	0	0	4
5	0	0	1	1	1	2

Имеются также пять типов специалистов, причем по два специалиста каждого типа.

В строках табл. 4 приведены команды для каждой работы, а в столбцах — занятость специалистов каждого типа.

Алгоритм.

1 шаг. В момент $t = 0$ начинаем работы 1, 2 и 3, которые образуют максимальный набор.

2 шаг. В момент $t = 8$ заканчивается работа 3. Начинаем работу 5. С работы 4 начать не можем, поскольку набор (1, 2, 4) недопустим.

3 шаг. В момент $t = 10$ заканчивается работа 2. Начинаем работу 4, поскольку набор (1, 4, 5) допустим.

4 шаг. В момент $t = 14$ заканчивается работа 4.

Продолжительность выполнения всех работ равна 14.

Оценку снизу можно получить, получив прерывание работ.

Имеются шесть максимальных наборов команд:

$$x_2 + x_4 + x_6 \geq 4,$$

$$x_3 + x_5 + x_6 \geq 2.$$

Ее оптимальное решение $x_1 = 7, x_2 = 4, x_3 = 1, x_4 = 0, x_5 = 1, x_6 = 0$ с величиной $\theta_{\min} = 13$.

При этом одна работа выполняется с перерывом, так что полученное решение является оптимальным. Решение большого числа примеров показало, что средняя относительная погрешность равна 5%, причем с ростом числа задач (команд) относительная ошибка уменьшается.

5.2. Сети с упорядоченными событиями

Рассмотрим задачу назначения команд для случая, когда существуют зависимости между работами. Эти зависимости будем описывать сетевым графиком, вершины которого соответствуют событиям, а дуги — работам. Примем, что события сетевого графика упорядочены, т.е. $t_{i-1} \leq t_i, i = \overline{1, r}$, где t_i — момент окончания i -го события.

Обозначим через R_s множество работ, которые могут выполняться в s -м интервале (интервал между $s-1$ и s -м событиями, $s = \overline{1, r}$), Q_i — множество интервалов, в которых может выполняться работа i , $y_s = \{y_{is}, i \in R_s\}$ — продолжительности работ, выполняемых в s -м интервале, $\Delta_s(y_s)$ — минимальная длина s -го интервала как функция y_s . Продолжительность проекта равна

$$(8) \quad \theta(y) = \sum_{s=1}^r \Delta_s(y_s).$$

Теорема. $\theta(y)$ — выпуклая функция y .

Доказательство. Пусть y^1 и y^2 — два допустимых решения.

Возьмем $y = ay^1 + (1-a)y^2, 0 \leq a \leq 1$. Это решение допустимо.

Обозначим через x^1 оптимальное решение для y^1, x^2 — оптимальное решение для y^2 . Очевидно, что $x = ax^1 + (1-a)x^2$ является допустимым решением для y . При этом продолжительность выполнения всех работ равна

$$T(x) = a\Delta y^1 + (1-a)\Delta y^2.$$

Имеем

$$\theta(y) \leq T(x) = a\Delta y^1 + (1-a)\Delta y^2.$$

Рассмотрим решение задачи для частного случая раздела 3. В этом случае решение задачи для каждого интервала выписывается в явном виде:

$$(9) \quad \Delta_s(y_s) = \begin{cases} y_{i,s}, & \text{если } y_{i_1,s} > y_{i_{n-1},s} + y_{i_n,s}, \\ \frac{y_{i_1,s} + y_{i_{n-1},s} + y_{i_n,s}}{2}, & \text{если } y_{i_1,s} \leq y_{i_{n-1},s}. \end{cases}$$

Учитывая выпуклость целевой функции (8), задача может быть решена методами градиентной оптимизации, в частности методом покоординатной оптимизации.

Рассмотрим численный пример.

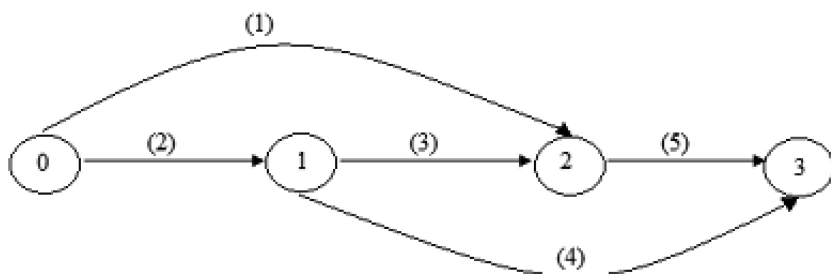


Рисунок.

Пример 5. Рассмотрим сетевой график, представленный на рисунке.

У дуг указаны номера работ. В табл. 5 приведены команды, выполняющие работы проекта, а в табл. 6 — продолжительности работ.

Таблица 5

$j \backslash i$	1	2	3	τ_j
(1, 1)	0	1	1	—
(0, 2)	0	1	1	—
(1, 2)	1	0	1	—
(1, 3)	1	1	0	—
(2, 3)	0	1	1	—

Таблица 6

i	1	2	3	4	5
τ_j	24	10	8	28	8

1 шаг. Возьмем начальное решение

$$y_{11} = 12, \quad y_{12} = 12, \quad y_{42} = 14, \quad y_{43} = 14,$$

продолжительность программы

$$\theta_1 = 12 + \frac{12 + 8 + 14}{2} + 14 = 43.$$

2 шаг. Оптимизируем по первой работе. Оптимальное решение

$$y_{11} = 10, \quad y_{12} = 14, \quad y_{42} = 14, \quad y_{43} = 14,$$

продолжительность программы

$$\theta_2 = 10 + \frac{14 + 8 + 14}{2} + 14 = 42.$$

3 шаг. Оптимизируем по четвертой работе. Оптимальное решение

$$y_{11} = 10, \quad y_{12} = 14, \quad y_{42} = 20, \quad y_{43} = 8,$$

продолжительность программы

$$\theta_3 = 10 + \frac{14 + 20 + 8}{2} + 8 = 39.$$

4 шаг. Оптимизируя по первой работе, получаем то же самое решение. Поэтому решение третьего шага является оптимальным.

В полученном решении имеется одно прерывание третьей работы. Если прерывания запрещены, то применяем эвристический алгоритм по степени критичности работ, т.е. начинаем в первую очередь критические работы. Критической является вторая работа.

1 шаг. В момент $t_1 = 0$ выполняем работы (1) и (2).

2 шаг. В момент $t_2 = 10$ начинаем выполнение критической работы (4).

3 шаг. В момент $t_3 = 24$ начинаем выполнение работы (3).

4 шаг. В момент $t_4 = 32$ начинаем выполнение работы (5).

Продолжительность программы $T = 40$.

Полученное решение является оптимальным, поскольку не существует решения без прерываний работ с продолжительностью 39.

6. Заключение

Полученные в статье результаты позволяют составлять оптимальные планы при допущении прерываний работы и достаточно хорошие планы при запрещении прерываний работ. Представляется интересным развитие исследований в следующих направлениях:

1. Добавление числа специалистов, что приведет к росту затрат, но ускорит выполнение работ;

2. Решение задачи по критерию минимизации максимального превышения сроков выполнения работ или по критерию минимизации упущенной выгоды.

СПИСОК ЛИТЕРАТУРЫ

1. Бурков В.Н. Распределение ресурсов как задача оптимального быстродействия // *АиТ*. 1966. № 7. С. 119–129.
Burkov V.N. Resource Allocation as a Problem of Optimal Rapidity // Autom. Remote Control. 1966. V. 27. No. 7. P. 1251–1260.
2. *Burkov V.N. Problems of Optimum Distribution of Resources // Control Cybern*. 1972. V. 1. No. 1–2. P. 27–41.
3. Чудров В.И. Задача о коммивояжере. М.: Изд-во “Знание”, 1969.
4. Баркалов С.А., Ворopaев В.И., Секлетова Г.И. Математические основы управления проектами / Под ред. В.Н. Буркова. М.: Высш. шк., 2005.

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

Поступила в редакцию 07.11.2017

После доработки 23.06.2018

Принята к публикации 08.11.2018