УДК 519.854.2
ББК 22.18

АЛГОРИТМ ПЛАНИРОВАНИЯ МЕРОПРИЯТИЙ ПО ПОДГОТОВКЕ КОСМОНАВТОВ МКС[1]

Лазарев А.А.[2]
(Учреждение Российской академии наук Институт проблем управления им. В.А. Трапезникова РАН, Москва; Московский Государственный Университет им. М.В. Ломоносова, Москва; Московский физико-технический институт государственный университет, Москва; Высшая школа экономики Национальный исследовательский университет, Москва.)
Сологуб А.А.[3]
(Московский Государственный Университет им. М.В. Ломоносова, Москва; Учреждение Российской академии наук Институт проблем управления им. В.А. Трапезникова РАН, Москва.)

В работе рассмотрена задача по планированию мероприятий подготовки космонавтов. Разработан подход для составления плана занятий. Проведена оценка применимости разработанного подхода.

Ключевые слова: теория расписаний, оптимизация, жадные алгоритмы, планирование, $\mathcal{NP}$-трудность.

Введение

При планировании мероприятий на МКС важным этапом является процесс подготовки космонавтов к полету. На этом этапе необходимо выбрать, какие работы поручить каждому космонавту. Известно, сколько времени необходимо для подготовки каждого из космонавтов по каждой из работ, также известно, сколько

[1] *Международная Космическая Станция.*
[2] *Александр Алексеевич Лазарев, профессор (jobmath@mail.ru).*
[3] *Александр Александрович Сологуб, студент (sologub10@gmail.com).*

займет тренировка, чтобы пилот был готов выполнить работу при допустимом или среднем качестве. Важным параметром такой задачи является время, когда должна закончиться подготовка и все должно быть готово к полету.

1. Постановка задачи

Пусть полетное задание состоит из множества операций $N = \{1, ..., n\}$, которые нужно распределить между космонавтами $M = \{1, ..., m\}$. Известно время, которое необходимо для подготовки космонавта $j \in M$ для работы $i \in N$, и это время равно $p(i, j, q)$, где $q \in \{0, ..., k\}$ — качество, с которым космонавт сможет выполнить данную работу. Очевидно, что вся тренировочная программа должна закончиться до момента старта космического корабля C. Требуется так организовать тренировочное расписание, чтобы разброс между окончаниями тренировок космонавтов был минимален.

$$\text{minimize} \quad \delta$$

$$\text{(1)} \quad \text{subject to} \quad C - \sum_{q=1}^{k} \sum_{i=1}^{n} p(i,j,q)\, x(i,j,q) \leqslant \delta, \quad j = 1, ..., m,$$

$$\text{(2)} \quad \sum_{j=1}^{m} \sum_{q=1}^{k} x(i,j,q) = 1, \qquad i = 1, ..., n,$$

$$\text{(3)} \quad x(i,j,q) = \{0, 1\},$$

где $x(i, j, q)$ — бинарная переменная, равная 1, если i-тая работа назначается космонавту j, а подготовка идет по качеству q, и равная 0 в противном случае.

Данная задача является $\mathcal{NP}$-трудной в сильном смысле, поэтому любой алгоритм будет приводить к экспоненциально растущей трудоемкости [3], при предположении, что $\mathcal{P} \neq \mathcal{NP}$.

2. Алгоритм

Введем вспомогательную функцию δ_i^j, $i \in N$, $j \in M$.

Определение 1. *Функция δ_i^j, $i = 1, ..., n$, $j = 1, ..., m$, описывает отступ от времени окончания общей подготовки j космонавта, при добавлении ему обучения по i работе.*

(4)

$$\delta_i^j = \begin{cases} \delta_{i-1}^j - p(i,j), & \textit{если мы готовим j космонавта по i работе;} \\ \\ \delta_{i-1}^j, & \textit{в противном случае.} \end{cases}$$

В качестве начального условия для всех $j = 1, ..., m$ возьмем $\delta_0^j = C$.

Замечание 1. Функцию δ_i^j можно записать в виде $\delta_i^j = \delta_{i-1}^j - p(i,j)x(i,j)$.

Для каждой работы i будем последвательно искать разность между минимальным и максимальным значением функции δ_i^j и выбирать такую подстановку работы космонавту, при которой значение (4) является минимальным. Таким образом сводим нашу задачу к следующей:

$$\begin{aligned} &\text{minimize} \quad \max_{j \in M} \delta_i^j - \min_{j \in M} \delta_i^j, \quad i = 1, ..., n, \\ &\text{subject to} \quad (1), (2), (3). \end{aligned}$$

Для решения задачи может быть применен жадный алгоритм. Жадный алгоритм является приближеным [1, 2].

Алгоритм 1 (Распределение работ между космонавтами).

1. Задаем $p(i,j)$, $i \in N$, $j \in M$;
2. Задаем C;
3. $\delta_0^j = C$, $j \in M$;
4. $x(i,j) = 0$, $i \in N$, $j \in M$;
5. **for** a $\leftarrow$ 0 **to** n
6. lock = 0
7. **for** b $\leftarrow$ 0 **to** m
8. **for** c $\leftarrow$ 0 **to** m
9. **if** b = lock
10. $\delta 1_b^c = C - p(i,j)$,
11. **else** $\delta 1_b^c = \delta_{a-1}^c$;
12. **endif**

endfor
$dev[b] = \max_j \delta 1_b^j - \min_j \delta 1_b^j$;
lock++;
endfor
$i^* = arg(\min_j dev[j])$;
for i $\leftarrow$ 1 **to** m
$\delta_a^i = \delta 1_{i^*}^i$;
endfor
$x_a^{i^*} = 1$;
endfor

На этапе инициализации задаем начальные условия для δ_0^j в соответствие с Определением 1 и заполняем массив бинарных переменных $x(i,j)$ нулевыми значениями. Внешинй цикл по счетчику a отвечает за продвижение по множеству работ $N = \{1, ..., n\}$. В теле основного цикла происходит последовательная подстановка работы с индексом a каждому из космонавтов, для этого вводится вспомоательная функция $\delta 1_j^j$, которая хранит значения функции δ_a^j для всех возможных подстановок работы a космонавтам. Далее находится переменная dev, в которую записывается разность между максимальным и минимальным значениями $\delta 1$. Таким образом мы находим разброс функции $\delta 1$ при назначении работы a каждому из космонавтов. В итоге работа a назначается тому космонавту, у которого значение dev минимально. Трудоемкость такого алгоритма составляет $O(m^2 n)$. В случае задач, связанных с планированием операций на МКС, число космонавтов гораздо меньше числа работ, которые им предстоит выполнить. Поэтому алгоритм можно считать линейным по числу работ [2].

3. Добавление дополнительных условий при составлении расписания

Рассмотрим случай, когда нужно учитывать еще и дополнительные условия (к примеру это может быть условие на строгое незапаздывание полного времени подготовки каждого из пилотов). Будем действовать по описанному ранее алгоритму, при этом учитывая выполнение дополнительного условия $\delta_i^j > 0$. Если на каком-нибудь шаге это условие перестает выполняться, то записываем номер данного шага в память как $b = i$, т.е. как значение границы выполнения данного условия. Для шага b ставим подготовку по работе более низкого качества, т.е. такую, по которой подготовка идет быстрее. Далее работа алгоритма продолжается в прежнем режиме. Если на каком-либо шаге дополнительное условие не может быть выполнено ни при каких временах подготовки, то возвращаемся на шаг $i = b - 1$ и ставим на нем подготовку с худшим качеством и продолжаем алгоритм. Таким образом, мы получим, либо подходящее решение, либо полную невозможность составления расписания при таких начальных параметрах.

4. Результаты и выводы

На основе Алгоритма 1 была написана программа на языке C++, которая принимает в качестве входных данных массив значений $p(i, j)$ и время окончания всех работ C. Итогом получается матрица значений бинарной функции $x(i, j)$, которая отвечает за назначение работы i космонавту j. Из-за того, что алгоритм эвристический, итоговый результат зависит от вида начальных данных. Для проверки работоспособности и применимости программы были проведены проверки на реальных данных, взятых из Центра управления полетами МКС. Итоговый разброс не превышает 5% от среднего значения суммарной времени поготовки.

Литература

1. ЛАЗАРЕВ А.А., ГАФАРОВ Е.Р. *Теория расписаний. Задачи и алгоритмы* //Москва, МГУ им. М. В. Ломоносова, 2011.
2. CORMEN, THOMAS H. AND STEIN, CLIFFORD AND RIVEST, RONALD L. AND LEISERSON, CHARLES E. *Introduction to Algorithms*. // McGraw-Hill Higher Education, 2001
3. DAVID PISINGER, HANS KELLERER, ULRICH PFERSCHY *Knapsack problems*// Springer. 2004

PLANNING ALGORITHM FOR TRAINING COSMONAUTS IN ISS

Alexander Lazarev, Institute of Control Sciences of RAS, Moscow, professor (jobmath@mail.ru).
Alexander Sologub, Lomonosov Moscow State University, Moscow, student (sologub10@gmail.com).

Abstract: In this paper we consider the problem of the planning of training cosmonauts. An approach to plan lessons. An assessment of the applicability of the developed approach.

Keywords: Scheduling theory, optimization, greedy algorithms, planning, $\mathcal{NP}$-hardness.