

УДК 519.854.2

ББК 22.18

ЗАДАЧА СОСТАВЛЕНИЯ РАСПИСАНИЯ РАБОТ ЭКИПАЖА НА БОРТУ МКС¹

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.)

Гущина В.П.³

(Московский Государственный Университет им. М.В. Ломоносова, Москва; Учреждение Российской академии наук Институт проблем управления им. В.А. Трапезникова РАН, Москва)

Рассматривается проблема составления расписания работы космонавтов на борту МКС и диспетчеров в ЦУП. Предлагается алгоритм решения данной задачи и проводится анализ его работы на различных типах данных.

Ключевые слова: теория расписаний, комбинаторная оптимизация, жадный алгоритм, метод ветвей и границ, временнóй горизонт планирования.

Введение

Управление таким объектом, как МКС, требует огромного количества различных человеческих и машинных ресурсов. Задача планирования является одной из самых приоритетных. Составление расписаний действий экипажа Российского сегмента (РС)

¹ *Международная Космическая Станция*

² *Александр Алексеевич Лазарев, профессор, (jobmath@mail.ru).*

³ *Варвара Павловна Гущина, студентка (vg@kvartasoft.ru).*

МКС выполняется Центром управления полетами (ЦУП). Его задачей является своевременное и корректное распределение работ между диспетчерами, находящимися на земле и космонавтами, находящимися в космосе на заданный горизонт планирования.

1. Постановка задачи

Горизонт планирования разбит на временные отрезки по 5 минут. Рабочий день космонавтов начинается в 10 часов утра по московскому времени (6 часов по Гринвичу) [4]. Имеется множество операций $N = \{1, \dots, n\}$ и экипаж, состоящий из нескольких космонавтов. Необходимо:

- 1) выбрать подмножество операций для исполнения в течение горизонта планирования;
- 2) распределить операции между космонавтами и диспетчерами;
- 3) определить для каждой операции момент начала ее исполнения, который должен совпадать с началом некоторого временного отрезка.

Каждая операция $i \in N$ характеризуется следующими параметрами:

- 1) приоритет (или важность) операции w_i ;
 - 2) время исполнения операции p_i ;
 - 3) временное окно исполнения операции: промежуток времени, когда операция может быть выполнена. Временное окно может состоять из нескольких промежутков времени, например, когда выполнение операции привязано к положению станции на орбите. Для каждой операции временное окно исполнения представим в виде множества моментов времени, в которые можно начать выполнение операции $T_i = \{t_{s1}, t_{s2}, \dots, t_{sk_i}\}$.
- Необходимо так составить расписание выполнения операций, чтобы сумма приоритетов выполненных заданий была максимальна, при этом необходимо выполнить все ресурсные ограничения и достичь равномерной загрузки космонавтов и диспетчеров на протяжении всего полета.

2. Обзор существующих задач и алгоритмов

2.1. ЗАДАЧА РАНЕЦ (KNPSACK PROBLEM [KP])

Дано множество $N = \{1, 2, \dots, n\}$ предметов. Предмет с номером j имеет ценность p_j и вес w_j . Также дана максимальная вместимость ранца C . Задача состоит в том, чтобы выбрать из множества N такой набор предметов, чтобы их суммарная ценность была максимальна, но при этом их общий вес не превышал вместимости ранца C .

$$(1) \quad \max \sum_{j=1}^n p_j x_j,$$

$$(2) \quad \sum_{j=1}^n x_j w_j \leq C,$$

где $x_j = 1$, если j -тый был выбран из множества N для помещения в ранец, и $x_j = 0$ в противном случае [2].

2.2. ЗАДАЧА НЕСКОЛЬКИХ РАНЦЕВ (MULTIPLY KNAPSACK PROBLEM [MKP])

Данная задача отличается от классической КР тем, что ранцев некоторое конечное число m , их вместимости обозначены $C_i, i = 1, \dots, m$.

$$(3) \quad \max \sum_{i=1}^m \sum_{j=1}^n p_j x_{ij},$$

$$(4) \quad \sum_{j=1}^n x_{ij} w_j \leq C_i, \quad i = 1, \dots, m,$$

где $x_{ij} = 1$, если j -тый предмет выбран из множества N для помещения в ранец с номером i , и $x_{ij} = 0$ в случае если j -тый предмет выбран для помещения в другой ранец или же не выбран вообще.

3. Существующие алгоритмы решения

3.1. ЖАДНЫЙ АЛГОРИТМ (GREEDY) ДЛЯ КР И МКР

Данный приближенный алгоритм требует введения для каждого предмета новой характеристики, которую в литературе обычно называют эффективностью. Она равна отношению стоимости предмета к его весу:

$$(5) \quad e_i = \frac{p_i}{w_i}.$$

Следуя концепции алгоритма, необходимо помещать в ранец предметы в порядке, соответствующем убыванию эффективности e_i [3].

3.2. МЕТОД ВЕТВЕЙ И ГРАНИЦ (BRANCH-AND-BOUND))

Метод состоит из двух составляющих:

- 1) этап определения (верхней/нижней) границы;
- 2) этап ветвления.

На этапе ветвления производится разделение задачи на несколько подзадач. Например, в случае задачи Ранец ветвление состоит в рассмотрении двух вариантов: поместить предмет в ранец или не помещать. На этапе определения верхней/нижней границы производится оценка (сверху/снизу) значения целевой функции для рассматриваемой подзадачи. Для ветвления обычно выбирается подзадача с наибольшей/наименьшей (верхней/нижней) границей [1].

4. Математическая постановка задачи

Пусть в полётном задании требуется выполнить множество операций $N = \{1, 2, \dots, n\}$. Пусть у космонавта k рабочих дней за полет, в каждом из рабочих дней h единиц рабочего времени. Всего за полет $P = k * h$ единиц t_j , $j = 1, \dots, P$ рабочего времени. Каждая операция $i \in N$ имеет приоритет/важность w_i , сложность c_i и продолжительность выполнения p_i . При этом, для каждой операции существует временн'ое окно исполнения. В связи с этим, будем считать что $w_i = w_{ij}(t_j)$, где $w_{ij}(t_j)$ не равно

0, если момент времени t_j входит во временное окно исполнения операции и $w_{ij}(t_j) = 0$, если не входит. В качестве входных данных задачи определим таблицу, в которой каждому заданию будет соответствовать набор важностей $w_{ij}(t_j)$ в зависимости от времени начала выполнения.

Таблица 1. Представление входных данных задачи.

Задания		Единицы времени						
		1	2	3	P-1	P
1.	Завтрак	1	0	0	0	0
2.	Измерение температуры	4	4	4	4	4
3.	Эксперимент А	1	1	0.75	1	0
4.
5.	Юстировка	1.6	1.6	0	0	0

Если рабочий день космонавта состоит из h единиц рабочего времени, то столбцы 1, 2, ..., h соответствуют первому рабочему дню, столбцы $h + 1$, ..., $2h$ второму и тд.

Задача состоит в том, чтобы максимизировать целевую функцию

$$(6) \quad \max \sum_{i=1}^n \sum_{j=1}^P x_{ij} w_{ij}(t_j),$$

где $x_{ij} = 1$ если задание i выполняется в j -й момент времени, $x_{ij} = 0$ – в противном случае.

При этом максимальная загруженность каждого дня не должна превышать некоторое число C .

$$(7) \quad \sum_{i=1}^n \sum_{j=1+(a+1)h}^{(a+1)h} x_{ij} c_j \leq C, \quad a = 0, \dots, k-1.$$

5. Алгоритм решения

Предлагаемый алгоритм решения данной задачи является объединением жадного алгоритма и метода ветвей и границ. Алгоритм состоит из следующих этапов.

1) Этап определения верхней границы. В каждой строке находится максимальный элемент, все максимальные элементы по строкам суммируются. Получаем верхнюю границу-значение целевой функции в «наилучшем случае»: в случае если все задания войдут в расписание с максимальным возможным для них приоритетом.

2) Сортировка таблицы входных данных по количеству ненулевых элементов в строке. Те задания, у которых в соответствующей строке лишь один ненулевой элемент, добавляем в расписание однозначным образом.

3) После добавления задания в расписания, у других, еще не добавленных в расписание заданий может измениться горизонт планирования. Поэтому следующим этапом является изменение таблицы (обнуление некоторых ее элементов).

4) Повторение пунктов 1-3 до тех пор пока первым в таблице не окажется задание, у которого более одного ненулевого элемента в строке.

5) Этап ветвления. Задача разделяется на несколько подзадач (по количеству возможных размещений в расписании первого задания). Для каждой из подзадач составляется расписание, таблица и подсчитывается верхняя граница.

6) Для дальнейшего рассмотрения из всех уже существующих выбирается подзадача с наибольшей верхней границей.

7) Повторение пунктов 1-6.

6. Программа

По данному алгоритму была написана программа. Программа получает на вход 3 текстовых файла следующего содержания.

1) В первом файле на первой строке стоит число рабочих дней и число единиц времени каждого рабочего дня через пробел. На второй строке – число всех заданий, которые необходимо выполнить. Начиная с третьей строки в файле располагается таблица всех w_{ij} размером $M * P$, где M – число заданий, а P – суммарное число единиц времени за весь полет.

2) Во втором файле через пробел записаны сложности всех заданий.

3) В третьем файле через пробел записаны продолжительности выполнения всех заданий.

На выходе программа выдает текстовый файл с расписаниями.

7. Выводы

Так как реальные данные по причинам секретности пока не были предоставлены, то следует провести анализ работы алгоритма для различных возможных ситуаций.

1) В случае если заданий мало, алгоритм выдает очень много различных расписаний. Чтобы этого избежать следует ужесточить ограничения на равномерную загруженность. Таким образом можно сделать следующий вывод: изменение жесткости ограничений приводит к изменению количества решений.

2) Алгоритм точный для тех случаев, когда все задания должны войти в полученное расписание.

3) Недостаток алгоритма в том, что при больших входных данных выполнение программы занимает много времени.

4) Достоинство алгоритма в том, что при усложнениях в постановке задачи, нет необходимости создавать новый алгоритм, а возможно усовершенствование предложенного, а именно, — создание дополнительных условий (ограничений), которые применяются на каждой итерации и приводят к уменьшению количества ветвей.

5) Также к достоинствам алгоритма можно отнести удобство распараллеливания при программировании.

8. Перспективы и развитие

Примеры возможного развития алгоритма при соответствующих усложнениях задачи.

1) Граф предшествования. При добавлении в постановку задачи графа предшествования между заданиями, в алгоритм добавится еще одно ограничение: если ветвь не удовлетворяет графу

предшествования – ее дальнейшее развитие проводиться не будет в независимости от ее верхней границы.

2) Способ, позволяющий выделить расписания, в которых меньше всего «окон»: промежутков между заданиями, при «перенесении» которых в конец или в начало дня, расписание становится бы удобнее.

3) Ресурсы, затрачиваемые при выполнении задания. При добавлении в постановку задачи ресурсных ограничений в алгоритме также появится ограничение, аналогично случаю описанному в пункте о графах предшествования.

4) Программа, формирующая таблицу входных данных. При большом количестве заданий, пользователю будет неудобно составлять первый файл, необходимый для работы программы, поэтому при получении реальных данных планируется добавить в программу блок, составляющий входную таблицу для первого файла с максимальным удобством для пользователя.

Литература

1. А. А. ЛАЗАРЕВ, Е.Р. ГАФАРОВ *Теория расписаний. Задачи и алгоритмы* // Москва. 2011
2. DAVID PISINGER, HANS KELLERER, ULRICH PFERSCHY *Knapsack problems* // Springer. 2004
3. SILVANO MARTELLO, PAOLO TOTH *Knapsack problems (Algorithms and computer implementations)* // 1990
4. Материалы «РКК Энергия».

ISS SCHEDULING PROBLEM

Alexander Lazarev, Institute of Control Sciences of RAS, Moscow, professor (jobmath@mail.ru).

Varvara Gushchina, Lomonosov Moscow State University, Moscow, student (vg@kvartasoft.ru).

Abstract: The ISS scheduling problem was described. Algorithm for solving this problem have been suggested and analyzed for different input data types.

Keywords: scheduling theory, optimization, greedy algorithm, branch-and-bound, time planning horizon.