

ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ

Научная статья

УДК 519.857

DOI 10.35266/1999-7604-2023-3-82-98

МНОГОКРИТЕРИАЛЬНОЕ ПЛАНИРОВАНИЕ РАБОТЫ ЦЕНТРА УПРАВЛЕНИЯ ПОЛОТОМ КОСМИЧЕСКИХ АППАРАТОВ НА ДИНАМИЧЕСКОЙ МОДЕЛИ С ТУПИКОВЫМИ УПРАВЛЕНИЯМИ

Виктор Павлович Корнеев

*Институт проблем управления им. В. А. Трапезникова Российской академии наук,
Москва, Россия*

vkorn@ipu.ru, <https://orcid.org/0000-0002-3643-1609>

Аннотация. Статья посвящена решению задачи многокритериального планирования запуска расчетов большого числа одновременно решаемых трудоемких вычислительных задач в условиях реализации режима разделения времени их решения в мультипроцессорных вычислительных системах программно-аппаратных средств центра управления полетом космических аппаратов. Планы формирования секторами управления программно-командной информации выполнены на динамической рекуррентной модели с тупиковыми управлениями. Определены критерии выбора эффективного варианта суточного плана работы секторов управления, представленного в виде временных интервалов допуска к ресурсам вычислительной системы. Применение метода тупиковых управлений в рамках динамической модели обусловлено его вычислительной эффективностью по сравнению с традиционными алгоритмами дискретной оптимизации. Приводится пример расчета алгоритмом тупиковых управлений рабочей программы, которая закладывается на борт космического аппарата во время сеанса связи.

Ключевые слова: многокритериальный выбор, план работы центра управления полетом, динамическая модель, тупиковые управления, режим разделения времени

Для цитирования: Корнеев В. П. Многокритериальное планирование работы центра управления полетом космических аппаратов на динамической модели с тупиковыми управлениями // Вестник кибернетики. 2023. Т. 22, № 3. С. 82–98. DOI 10.35266/1999-7604-2023-3-82-98.

Original article

MULTI-CRITERIAL SCHEDULING OF A SPACECRAFT CONTROL CENTER'S OPERATION ON A DYNAMIC MODEL WITH DEAD-END CONTROLS

Viktor P. Korneenko

V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia

vkorn@ipu.ru, <https://orcid.org/0000-0002-3643-1609>

Abstract. The article is devoted to the solution of a problem of multi-criterial scheduling for execution of calculations of a large number of time-consuming computational tasks being solved simultaneously when implementing the time-sharing mode for their solutions in software and hardware's multi-processor computing systems by the spacecraft control center. A dynamic recurrent model with dead-end controls is used to implement the schedule of organizing software command information with control sectors. The criteria for selecting an efficient option of a daily schedule for the control sectors' operation are determined, with the schedule being presented as time intervals of access to the computing system's resources. Compared to the traditional algorithms of discrete optimization, the method of dead-ends control is more efficient in calculation in the framework of a dynamic model. An example of calculations conducted by the dead-end controls algorithm for the operational program, which is used on board of the spacecraft during a communication session, is given.

Keywords: multi-criteria selection, spacecraft control center's operation plan, dynamic model, dead-end controls, time-sharing mode

For citation: Korneenko V. P. Multi-criterial scheduling of a spacecraft control center's operation on a dynamic model with dead-end controls. *Proceedings in Cybernetics*. 2023;22(3):82–98. DOI 10.35266/1999-7604-2023-3-82-98.

ВВЕДЕНИЕ

Сектора управления центра управления полетами (ЦУП) космических аппаратов (КА) решают полный комплекс задач по сбору и обработке телеметрической и баллистической информации, моделированию различных полетных ситуаций, а также расчету программно-командной информации (рабочих программ, разовых команд) для управления бортовой аппаратурой на многопроцессорном вычислительном комплексе (ВК) [1–4].

Существующие алгоритмы планирования вычислительных задач (ВЗ) исходят из предположения, что время выполнения известно заранее, а также базируются на статистических моделях [5–7]. Однако поскольку расчет ВЗ на программно-аппаратных средствах ЦУП может выполняться в любое время, то из-за большого числа одновременно решаемых трудоемких задач время пребывания отдельной ВЗ возрастает. Это связано с тем, что в мультипроцессорных вычислительных системах реализован режим разделения времени для пакета задач, которые обслужи-

ваются одновременно [8, 9]. При этом время пребывания отдельной ВЗ возрастает при увеличении числа одновременно запущенных секторами управления задач.

В связи с этим при управлении большой группировкой КА возникает проблема равномерной загрузки вычислительных ресурсов ЦУП задачами операторов секторов управления на суточном интервале полета КА [10].

Под ресурсами в вычислительных системах обычно подразумеваются следующие характеристики: количество процессоров; объем оперативной и внешней памяти; сетевое оборудование для соединения с удаленными абонентами и др.

Поэтому обычно план работы ЦУП кроме месячного плана профилактических работ по техническому обслуживанию и ремонту программно-аппаратных средств (ПАС) должен включать разработку программы выполнения вычислительных задач (ПВВЗ) на вычислительных ресурсах (серверах) ЦУП. Схема технологии функционирования ЦУП КА представлена на рис. 1.



Рис. 1. Схема технологии функционирования центра управления полетами космических аппаратов

Примечание: составлено автором.

Во время сеансов связи с КА от наземных комплексов управления в ЦУП поступают траекторные измерения (ТИЗ), телеметрическая информация (ТМИ), контрольно-командная информация (ККИ), а также информация, предназначенная для потребителей, которая после обработки помещается в хранилище данных о состоянии КА, радиотехнических станциях (РТС) отдельных командно-измерительных комплексов (ОКИК).

На этапе планирования в соответствии с заданными суточными программами полета (СПП) и технологическими циклами управления (ТЦУ) на основе целевых данных и указаний, от которых зависит объем вычислительных задач на ресурсах ПАС ЦУП КА, секторами управления формируется программно-командная информация (ПКИ), а также временная диаграмма полета (ВДП).

На основе таблиц полетов КА осуществляется планирование работы наземного комплекса управления. Результат планирования оформляется в виде массива и записывается в архив базы данных оперативных планов наземного комплекса (ОПНК) управления, в котором перечислены сеансы управления КА с конкретными наземными радиотехническими средствами на суточном интервале времени.

К показателям, служащим в качестве критериев выбора эффективного варианта плана работы ЦУП, можно отнести среднюю загрузку процессоров, равномерность загрузки программно-аппаратных средств на интервалах технологических циклов управления КА, среднее время прохождения вычислительных заданий в ВК ЦУП, равномерность загрузки операторов активной работой на суточном интервале управления КА и др. [10, 11]. Обеспечение оперативности и устойчивости управления КА в режиме реального времени предъявляют высокие требования к эффективному использованию вычислительных ресурсов [6, 7].

Задача планирования комплекса вычислительных работ (задач) в рамках технологического цикла управления, последовательность выполнения которых представляется обычно в виде ориентированного графа (сети), относится к классу задач большой размерности теории расписания.

В научной литературе общая теория расписаний подробно освещается в работах [12, 13]. В работах [14–16] задача планирования взаимосвязанных работ, представленных в виде ориентированных графов на ограниченных ресурсах, сводится к задаче оптимального управления динамической системой.

В работе [16] логические ограничения в виде функции штрафа введены в правую часть дифференциальных уравнений, описывающих состояние работ, а в качестве минимизируемого функционала используется выпуклая при некоторых допущениях функция суммарных отклонений между заданными и фактическими значениями фазовых переменных состояний работ.

Для моделей с дискретным временем и дискретным пространством состояний наибольшее применение получили методы динамического программирования и только в частном случае для марковских динамических задач [17] – методы дискретной оптимизации [18].

В предлагаемой статье в рамках динамической модели для уменьшения размерности пространства состояний при расчете рекуррентных уравнений применена идея тупиковых управлений, под которыми понимается допустимое управление, в котором увеличение интенсивности выполнения хотя бы одной из работ делает его недопустимым по ресурсным ограничениям [19]. Идея тупиковых управлений взята из задачи минимизации и получения сокращенной (тупиковой) дизъюнктивной нормальной формы, представляющей из себя произвольную дизъюнкцию элементарных конъюнкций логических функций булевой алгебры, которую нельзя упростить [20].

В конце статьи суть алгоритма тупиковых управлений, лежащего в основе формирования ПВВЗ, проиллюстрирована примером расчета рабочей программы, состоящей из подпрограмм, для закладки на борт КА во время сеанса связи.

МАТЕРИАЛЫ И МЕТОДЫ

1. Модель планирования работ в вычислительной системе ЦУП КА

Пусть ЦУП обеспечивает централизованное управление орбитальной группировкой КА нескольких типов на заданном (суточном)

интервале времени $T = [t_0, t_f]$, которую представим в виде семейства множеств:

$$A = \{A_l : l \in L\}, L = \overline{1, n_A},$$

где n_A – число типов КА;

$A_l = \{A_{la} : a = 1, 2, \dots, n_{A_l}\}$ – множество КА различного назначения (типа);

n_{A_l} – число КА l -го типа.

Обозначим через $N_{A_l}(T)$ – число выполняемых технологических циклов управления КА l -го типа в интервале времени $T = [t_0, t_f]$. Тогда пропускная способность ЦУП на заданном интервале времени будет характеризоваться вектором:

$$N(T) = (N_{A_1}(T), N_{A_2}(T), \dots, N_{A_{n_A}}(T)). \quad (1)$$

Обозначим через:

$\Theta = \{\theta_l : l \in L\}$ – множество операторов секторов управления, управляющих КА l -го типа;

$M = \{1, 2, \dots, m, \dots, n_M\}$ – множество программно-аппаратных средств вычислительной системы (ВС), включая сервера локальной сети, аппаратуру системы связи и передачи данных и т. п.;

$Z_l = \{z_s^l : s = \overline{1, n_{Z_l}}\}$ – комплекс частично упорядоченных вычислительных задач (работ, заданий, технологических операций) для формирования программно-командной информации для управления КА l -го типа на суточном интервале времени $T = [t_0, t_f]$.

Логическую структуру, т. е. взаимосвязи между вычислительными работами (технологическими операциями), определяющие порядок их выполнения для КА l -го типа, представим в виде пары:

$$\Gamma_{las} = \langle \Gamma_{las}^-, \Gamma_{las}^+ \rangle,$$

где $\Gamma_{las}^- = \{z_q^{la} \in Z_l \mid z_q^{la} \prec z_s^{la}\}$ – множество z_q^{la} задач, непосредственно предшествующих выполнению z_s^{la} задаче;

$\Gamma_{las}^+ = \{z_q^{la} \in Z_l \mid z_s^{la} \prec z_q^{la}\}$ – множество z_q^{la} задач, непосредственно следующих за выполнением z_s^{la} задачи.

Комплекс вычислительных задач для управления a -м КА l -го типа на суточном интервале времени $T = [t_0, t_f]$ будем представлять в виде:

$$Z_{la} = \{z_s^{la} : s = \overline{1, n_{Z_{la}}}\}.$$

Множество индексов комплексов вычислительных задач для управления всей орбитальной группировки ЦУП КА представим в виде:

$$S = \bigcup_{l \in L} \bigcup_{a \in A_l} \{1, 2, \dots, n_{Z_{la}}\},$$

где $n_S = \sum_{l \in L} \sum_{a \in A_l} n_{Z_{la}}$ – число вычислительных задач комплексов.

Тогда исходные данные для формирования программы выполнения комплексов вычислительных задач на вычислительных ресурсах ЦУП представим в виде объединения комплексов вычислительных работ:

$$Z = \bigcup_{l=1}^{n_A} \bigcup_{a=1}^{n_{A_l}} \bigcup_{s=1}^{n_{Z_{la}}} \{z_s^{la}\}.$$

Для каждой вычислительной задачи $z_s^{la} \in Z$ считаем задан объем Q_s^{la} вычислений. Зная производительность Π_m вычислительного комплекса $m \in M$ и объем Q_s^{la} вычислительной задачи z_s^{la} , найдем длительность решения в монопольном режиме: $\tau_{ms}^{la} = \frac{Q_s^{la}}{\Pi_m} \forall z_s^{la} \in Z$.

Следует заметить, что длительность выполнения вычислительной задачи на конкретном вычислительном средстве (процессоре) легко определить экспериментально.

Объемы ресурсов (оперативной и внешней памяти), необходимых для выполнения вычислительной задачи, зададим в виде:

$$B_m = \{b_{ms}^{la} : m \in M\}.$$

Учитывая, что производительность зависит от вероятностно-надежностных характеристик m -го средства, то априорно можно утверждать, что длительность решения задачи в монопольном режиме на m -м средстве, рассматривая ее как случайную величину с произвольным законом распределения с вероятностью P_{rob} близкой к нулю, уклоняется от среднего значения $E(\tau_{ms}^{la})$ не менее

чем на $\pm 3\sigma_{\tau_{ms}^{la}}$ ($\sigma_{\tau_{ms}^{la}}$ – стандартное или среднеквадратическое отклонение случайной величины τ_{ms}^{la}) [21]:

$$P_{rob} \left[\left| \tau_{ms}^{la} - E(\tau_{ms}^{la}) \right| \geq 3\sigma_{\tau_{ms}^{la}} \right] < 0,003.$$

Поэтому с вероятностью, близкой к единице, можно предположить, что:

$$\tau_{ms}^{la} \in \left[\underline{\tau_{ms}^{la}}, \overline{\tau_{ms}^{la}} \right],$$

где $\underline{\tau_{ms}^{la}} = E(\tau_{ms}^{la}) - 3\sigma_{\tau_{ms}^{la}}$, $\overline{\tau_{ms}^{la}} = E(\tau_{ms}^{la}) + 3\sigma_{\tau_{ms}^{la}}$.

На практике можно ограничиться средним значением длительности выполнения технологической операции на m -м средстве. Исходя из этого формализованное задание на расчет программы выполнения вычислительных задач представим в виде кортежа:

$$\Phi = \left\langle Z; \Theta; \{\tau_{ms}^{la}\}_{m \in M}; \bigcup_{m \in M} \{b_{ms}^{la}\}_{s \in S}; T_{ms}^{lv} \right\rangle, \quad (2)$$

где $s \in S$, $a \in A_l$, $l \in L$.

2. Динамическая модель программного управления с выполнением вычислительных задач. Уравнения, описывающие выполнение задач

Пусть на планируемом временном (суточном) интервале $T = [t_0, t_f]$ для обеспечения целевого функционирования КА необходимо выполнить множество вычислительных задач (операций), формализованное задание которых с учетом состава и структуры программно-аппаратных средств представлено в виде кортежей (2). Каждой задаче $z_s^{la} \in Z_{la}$ с заданным для нее допустимым временным интервалом выполнения в момент времени $t \in T_s^{la}$ поставим в соответствие неотрицательную переменную величину $x_s^{la}(t) \in [0, 1']$, характеризующую долю (процент) выполненного объема этой задачи, и которую будем называть состоянием данной задачи. Задача $z_s^{la}(t)$ считается законченной, если:

$$x_s^{la}(t) = 1 \quad \forall s \in S, \quad a \in A_l, \quad l \in L. \quad (3)$$

При $0 < x_s^{la}(t) < 1$ операция выполнена частично. Корень уравнения (3) обозначим – t_{fs}^{la} . Он будет моментом завершения задачи. Момент начала выполнения операции обозначим – t_{0s}^{la} . Управление состоянием задачи z_s^{la} на интервале планирования будем осуществлять с помощью булевой переменной $u_{ms}^{la} \in \{0, 1\}$, которой дадим следующую интерпретацию:

$$u_{ms}^{la}(t) = \begin{cases} 1, & \text{если } z_s^{la} \text{ задача в момент времени } t \\ & \text{обслуживается } m\text{-м средством,} \\ 0 & \text{– в противном случае.} \end{cases}$$

Средства, работающие в режиме разделения времени (мультипрограммирования, мультипроцессирования), в течение небольшого кванта времени δt циклически выделяются каждому требованию на выполнение задачи [8, 22].

Тогда количество в моменты поступления на обслуживание задач и в моменты ухода с обслуживания задач можно представить в виде функции времени:

$$N_m(t) = \sum_{s \in S} \sum_{a \in A_l} \sum_{l \in L} u_{ms}^{la}(t), \quad (4)$$

$$u_{ms}^{la}(t) = \{0, 1\},$$

где $t \in T = [t_1, t_2]$.

В течение интервала времени $[t_1, t_2]$ среднее число задач в m -й вычислительной системе составит: $\bar{N}_m(t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} N_m(t) dt$.

Введем понятие скорости (интенсивности) выполнения (технологической) операции на программно-аппаратных средствах ЦУП КА.

Под скоростью (интенсивностью) выполнения z_s^{la} задачи на m -м (типе) средстве будем понимать скорость изменения доли выполняемого объема этой задачи в единицу времени.

При данном количестве (4) обслуживаемых задач m -м средством за время δt_m , равное промежутку времени между очередными циклическими моментами предоставления m -го средства в монопольное обслуживание

очередной задачи, доля выполненного объема составит $\delta t_m / \tau_{ms}^{la}$. Если предположить, что на отрезке времени длительностью δt_m число обслуживаемых задач не меняется, т. е. $N_m(\delta t) = \text{const}$, то отсюда весь Q_s^{la} объем z_s^{la} задачи (случай $x_s^{la} = 1$) будет выполнен в режиме разделения времени за время пребывания на обслуживании, например, равное $\tau_{ms}^{la} \times N_m(\delta t)$.

Однако учитывая, что число задач (4) в моменты поступления новых требований и (или) ухода обслуженных требований на выполнение задач флуктуирует во времени, то скорость выполнения каждой задачи на m -м средстве, работающем в режиме разделения времени, представим неоднородным дифференциальным уравнением с флуктуирующей во времени правой частью вида:

$$x_s^{la} = \sum_{m=1}^{n_M} \frac{1}{\tau_{ms}^{la}} \times \frac{1}{N_m(t)} \times u_{ms}^{la}(t) \forall s \in S, \quad (5)$$

$$a \in A_l, l \in L.$$

Величину, равную $1/N_m(t)$ в теории очередей применительно к исследованию вычислительных систем, называют скоростью обслуживания требований и обычно связывают с дисциплиной справедливого разделения (центрального) процессора (ресурса), которая была введена Клейнроком как предельный аналог циклической дисциплины обслуживания требований при $\delta t_m \rightarrow 0$ [8]. От системы уравнений (5) перейдем к разностным уравнениям первого порядка с переменным временным шагом квантования, длина которого определяется (выбирается) в процессе формирования программы выполнения задач на ВС ЦУП КА:

$$x_s^{la}(t_{k+1}) = x_s^{la}(t_k) + \frac{\Delta t_k}{\tau_{ms}^{la}} \times \frac{1}{N_m(t_n)} \times u_{ms}^{la}(t_k); \quad (6)$$

$$\forall (s, a, l),$$

где $x_s^{la}(t_n)$ состояние z_s^{la} задачи в момент времени $t_k \in T(k=1, 2, \dots)$, когда происходит

скачок в скорости обслуживания на m -м средстве.

Нахождение управляющих переменных (воздействий) $u_{sm}^{la}(t)$ в виде кусочно-постоянных функций времени $u_{ms}^{la} = u_{ms}^{la}(t)$ на $t \in T = [t_0, t_f]$ равносильно построению графика выполнения вычислительных задач в интервальном виде для всех $z_s^{la} \in Z$:

$$T_s^{la}[u(t)] = \{[t_{0s}^{la}, t_{fs}^{la}] | u_{ms}^{la}(t) = 1, 0 \leq x_s^{la}(t) \leq 1\},$$

где t_{0s}^{la} – момент начала вычисления z_s^{la} задачи на m -м средстве;

t_{fs}^{la} – момент окончания вычисления z_s^{la} задачи.

Заметим, что длительность пребывания z_s^{la} задачи на обслуживании m -м средством, равная величине $\Delta_{ms}^{la} = t_{fs}^{la} - t_{0s}^{la}$, совпадает с длительностью τ_{sm}^{la} при решении в монопольном режиме вычислительного средства (модуля), превышает ее значение $\Delta_{ms}^{la} \geq \tau_{ms}^{la}$, если задача решается на средстве в режиме разделения времени с другими задачами.

Наряду с переменными, характеризующими состояние z_s^{la} задачи, введем переменные величины y_m , характеризующие задействие ПАС. Изменения состояний этих переменных запишем в дифференциальной и интегральной формах:

$$\dot{y}_m(t) = \sum_{l=1}^{n_A} \sum_{a=1}^{n_{A_l}} \sum_{s=1}^{n_{Z_{la}}} u_{ms}^{la}(t) \forall m \in M,$$

где $y_m(t) = \sum_{l=1}^{n_A} \sum_{a=1}^{n_{A_l}} \sum_{s=1}^{n_{Z_{la}}} \int_{t_0}^t u_{ms}^{la}(\tau) d\tau$ – переменная, которая характеризует общую продолжительность задействия средства с номером $m \in M$ задачами z_s^{la} на интервале управления $[t_0, t)$.

Переменные $\theta \in \Theta$, которые характеризуют время задействия операторов по управлению КА l -го типа, представим в дифференциальной и интегральной формах:

$$v_{l\theta} = \sum_{a=1}^{n_{z_{la}}} u_{\theta}^l(t) \forall \theta \in \Theta; v_{l\theta}(t) = \sum_{a=1}^{n_{z_{la}}} u_{\theta}^{la}(\tau) d\tau,$$

где $u_{\theta}^{la} = 1$, если оператор задействован по управлению КА l -го типа и $u_{\theta}^l(t) = 0$ – в противном случае.

Ограничения, накладываемые на выполнение задач

Ограничения, накладываемые на управляющие воздействия в нелинейной динамической модели с учетом особенностей выполнения задач, разобьем на следующие три группы.

1. Ресурсные ограничения, связанные с воздействием на время выполнения различных видов памяти, программных модулей и т. п. при решении z_s^{la} задачи на m -м средстве, представим в виде неравенств на объем b -го вида ресурса m -го средства (процессора), работающим в режиме разделения времени:

$$\sum_{l=1}^{n_A} \sum_{a=1}^{n_{A_l}} \sum_{s=1}^{n_{z_{la}}} b_{ms}^{la}(t) u_{ms}^{la}(t) \leq B_m(t) \forall m \in M,$$

где $b_{ms}^{la}(t)$ – требуемое количество b -го вида ресурса при выполнении z_s^{la} задачи при монопольном решении на m -м вычислительном средстве;

$B_m(t)$ – общее текущее количество ресурсов, выделяемое для выполнения задач на m -м вычислительном средстве.

2. Ограничения логического характера, определяющие очередность выполнения частично-упорядоченных задач, когда любая задача z_{ms}^{la} не может выполняться ($u_{ms}^{la}(t) = 0$) до завершения предшествующих $z_q^{la} \in \Gamma_{las}^-$, можно записать в виде равенства:

$$\sum_{m \in M} \sum_{z_{mq}^{la} \in \Gamma_{las}^-} (1 - x_{mq}^{la}(t)) \times u_{ms}^{la}(t) = 0 \forall s. \quad (7)$$

При этом переменная $x_s^{la}(t)$ состояния z_{ms}^{la} задачи на m -м средстве достигнет значения $x_s^{la}(t_{fs}^{la}) = 1$, где $t = t_{fs}^{la}$ – момент завершения:

$$x_s^{la}(t) = \sum_{m \in M_j^{va}} \frac{1}{\tau_{ms}^{la}} \times \int_{t_{0s}^{la}}^t \frac{1}{N_m(\tau)} d\tau,$$

по форме являются интегральными уравнениями состояния задачи для $t \geq t_{0s}^{la}$ и эквива-

лентны системе неоднородных дифференциальных уравнений (6).

3. Временные ограничения зададим в виде:

$$Td_s^{la} = [t_{0s}^{la}, t_{fs}^{la}], Td_{0s}^{la} = [t_{0s}^{la}, t_{fs}^{la}), \quad (8)$$

$$Td_{fs}^{la} = (t_{0s}^{la}, t_{fs}^{la}],$$

где t_{0s}^{la} – начальный допустимый (директивный) момент времени выполнения;

t_{fs}^{la} – конечный допустимый (директивный) момент времени выполнения;

Td_s^{la} – фиксирован интервал времени выполнения;

Td_{0s}^{la} – фиксирован начальный (директивный) момент времени выполнения;

Td_{fs}^{la} – фиксирован конечный (директивный) момент времени выполнения.

Временные ограничения запишем в интегральной форме:

$$\sum_{m=1}^{n_M} \sum_{a=1}^{n_{A_l}} \sum_{s=1}^{n_{z_{la}}} \int_{t_{0s}^{la}}^{t_{fs}^{la}} u_{ms}^{la}(\tau) d\tau + \int_{t_{fs}^{la}}^t u_{ms}^{la}(\tau) d\tau = 0 \forall t \in T.$$

Критерии оценки качества выполнения плана работы ЦУП КА

Качество плана работы ЦУП КА, включая программу выполнения вычислительных задач, будем оценивать набором критериев. В постановке задачи многокритериального выбора эффективного варианта плана работы часть критериев подлежат минимизации, а часть – максимизации.

Критерий времени окончания выполнения комплекса работ:

$$I_1 = \sum_{l=1}^{n_A} \sum_{a=1}^{n_{A_l}} \sum_{s=1}^{n_{z_{la}}} t_{fs}^{la}(u_{ms}^{la}(t)) \rightarrow \min_{u_{ml}(t)}$$

Критерий, отражающий пропускную способность ЦУП КА выполняемых технологических циклов управления (1) на интервале $T = [t_0, t_f]$, представим в виде:

$$I_2 = \sum_{l=1}^{n_A} \sum_{a=1}^{n_{A_l}} N_{A_a}(T) \rightarrow \max.$$

Критерий, отражающий равномерность загрузки вычислительными задачами средств (процессоров) ЦУП КА, представим в виде:

$$I_3 = \sum_{m=1}^{n_M} \left| y_m(t) - \frac{1}{n_M} \sum_{m=1}^{n_M} y_m(t) \right| \rightarrow \min,$$

где $t \in T = [t_0, t_f]$.

Критерий, отражающий равномерность загрузки операторов автоматизированных рабочих мест секторов управления КА, представим в виде:

$$I_4 = \left| \int_{t_0}^{t_f} v_\theta(t) dt - \frac{1}{n_A} \sum_{l=1}^{n_A} \int_{t_0}^{t_f} v_\theta(t) dt \right| \rightarrow \min.$$

Для задач, объединенных по технологии в комплексы, в качестве оптимизируемого критерия целесообразно минимизировать самое позднее окончание какой-либо работы комплекса для КА l -го типа:

$$I_5 = \max \left\{ \sum_{a=1}^{n_{A_l}} \sum_{s=1}^{n_{Z_{la}}} t_{fs}^{la} \right\} \rightarrow \min.$$

Пиковые нагрузки в рассматриваемой задаче расчета оптимальной программы вычислительных задач отразим при помощи функционала, минимизирующего на временном интервале планирования максимальные отклонения загрузки m -го средства:

$$I_6 = \max_{t', t'' \in T} |y_m(t') - y_m(t'')| \rightarrow \min.$$

В отдельных случаях приведенный список показателей эффективности программы выполнения задач может быть дополнен другими, учитывающими иные аспекты решаемой задачи.

3. Алгоритм тупиковых управлений

Состояние программы выполнения вычислительных задач на текущий момент времени $t_k \in T$ на m -м вычислительном средстве будет характеризоваться кортежем:

$$X_m(t_k) = \langle x_m(t_k), u_m(t_k) \rangle, m = \overline{1, n_M},$$

где $x_m(t_k) = (x_{m1}(t_k), \dots, x_{ms}(t_k), \dots, x_{mns}(t_k))$ – вектор состояния задач;

$u_m(t_k) = (u_{m1}(t_k), \dots, u_{ms}(t_k), \dots, u_{mns}(t_k))$ – вектор управляющих переменных.

Начальному состоянию программы на $m \in M$ комплексе соответствует нулевой вектор $x_m(t_0) = (0, 0, 0, \dots, 0)$, а конечному состоянию – вектор $x_m(t_f) = (1, \dots, 1)$.

Для компактной записи введем обозначения $x_s \equiv x_{ms}$, $u_s \equiv u_{ms}$, $\tau_s \equiv \tau_{ms}$, $b_s \equiv b_{ms}$. Исходные данные для каждой вычислительной задачи будем характеризовать кортежем $\langle \tau_s, b_s \rangle$, где τ_s – длительность решения задачи в монопольном режиме; b_s – требуемый объем оперативной памяти.

Краевые условия в рассматриваемой задаче программного управления выполнением вычислительных работ для КА на левом и правом конце временного интервала представим в виде:

$$t = t_0 : x_s(t_0) = 0; t = t_f : x_s(t_f) = 1^\circ \forall s \in S.$$

Таким образом, задача расчета оптимальной программы выполнения заданий по формированию ПКИ на программно-аппаратных средствах ЦУП сводится к поиску управлений $u_s(t)$, принадлежащих множеству $U(t)$ допустимых управляющих воздействий, удовлетворяющих заданным ресурсным, логическим и временным ограничениям.

Для компактной записи изложенного введем в рассмотрение вектор переменных состояний и управляющих переменных:

$$x(t) = (x_1(t), \dots, x_s(t), \dots, x_{ns}(t));$$

$$u(t) = (u_1(t), \dots, u_s(t), \dots, u_{ns}(t)).$$

Шаг 1. Формирование фронта вычислительных заданий. На $m \in M$ вычислительном средстве фронт заданий, удовлетворяющих логическим (7) и временным (8) ограничениям на t_k момент времени, представим в виде:

$$Z_m(t_k) = \{z_s : s = \overline{1, n_m}\}. \quad (9)$$

Шаг 2. Построение тупиковых управлений. Построение тупиковых управлений начинается с упорядочения номеров вычислительных

заданий в порядке убывания требуемых объемов оперативной памяти вычислительного ресурса.

Пусть $b_1 \geq \dots b_s \geq \dots \geq b_{n_m}$. В первое тупиковое управление в момент времени $t_k \in T$ включаем первую вычислительных задачу, что соответствует управлению $u_1(t_k) = 1$, если он не нарушает ограничения на объем оперативной памяти:

$$\sum_{s=1}^{n_m} b_s u_s \leq B_m, u_s \in \{0, 1\}, s = 1, 2, \dots, n_m, \quad (10)$$

т. е. выполняется неравенство $b_1 u_1(t_k) \leq B_m$, в противном случае полагаем $u_1(t_k) = 0$.

Точно так же поступаем со второй, третьей и s -й вычислительной задачей в соответствии с формулой:

$$u_s = \begin{cases} 1, & \text{если } \sum_{j=1}^s b_j u_j \leq B_m; \\ 0, & \text{если } \sum_{j=1}^s b_j u_j > B_m \end{cases} \quad (11)$$

последовательно для $i = 1, 2, \dots, n_m$.

В результате получим первое тупиковое управление, состоящее из нулей и единиц, которое будем обозначать в виде (геометрического) вектора:

$$\vec{u}_1 = (u_1^{(1)}, u_2^{(1)}, \dots, u_{n_m}^{(1)}). \quad (12)$$

Построенному вектору управления (12) соответствует некоторое двоичное число $\xi_1 = (11010\dots 1\dots 01001)$, где единицы стоят в тех разрядах, номера которых совпадают с номерами задач, включенных в управление. Замена любого нуля единицей делает это управление недопустимым по ограничению (10).

С помощью первого тупикового управления построим второе. Для этого найдем самый младший разряд числа ξ_1 , в котором записан ноль. Во всех разрядах справа от него вместо единиц записываем нули. В полученном двоичном числе первую справа единицу перенесем на один разряд вправо. Если полученное управление недопустимо по ограничению (10), то эту единицу сдвигаем еще на один разряд вправо до тех пор, пока управление не окажется допустимым. Далее в раз-

ряды справа от этой единицы помещаем единицы по тому же правилу (11), что и при построении числа ξ_1 . В результате получаем двоичное число $\xi_2 < \xi_1$. Этому числу соответствует тупиковое управление:

$$\vec{u}_2 = (u_1^{(2)}, u_2^{(2)}, \dots, u_{n_m}^{(2)}). \quad (13)$$

Точно таким же образом из тупикового управления (14) строится тупиковое управление $\vec{u}_3$, которому соответствует двоичное число $\xi_3 < \xi_2 < \xi_1$ и т. д. В результате получаем множество тупиковых управлений:

$$U = \{\vec{u}_i = (u_1^{(i)}, \dots, u_j^{(i)}, \dots, u_{n_U}^{(i)}) \mid i = 1, 2, \dots, n_U\}, \quad (14)$$

которому соответствует упорядочение $\xi_{n_U} < \xi_{n_U-1} < \dots < \xi_3 < \xi_2 < \xi_1$ двоичных чисел. Описанная процедура дает возможность получить все тупиковые управления, удовлетворяющие ограничению (11).

Шаг 3. Вычисление оптимального тупикового управления. За пошаговый критерий процесса планирования заданий на текущий момент времени $t_k \in T$ на m -м вычислительном средстве примем минимизацию времени выполнения фронта вычислительных заданий $Z_m(t_k)$ (9):

$$F(u_1^*, \dots, u_{n_U}^*) = \min_{u_1, \dots, u_{n_U}} \sum_{s=1}^{n_U} \tau_s(t_k) u_s, \quad (15)$$

$$\sum_{s=1}^{n_U} b_s u_s \leq B_m,$$

где $\tau_s(t_k) = \tau_s \times [1 - x_s(t_k)]$ – оставшееся время решения задачи, поступившей на обслуживание ($0 < x_s(t_k) < 1; u_s = 1$).

Для каждого тупикового управления $\vec{u}_i \in U$ находим значение шагового критерия (15). За оптимальное управление принимаем тупиковое $\vec{u}_* \in U$, обеспечивающее минимальное значение шагового критерия (15):

$$\vec{u}_* = (u_1^*, \dots, u_s^*, \dots, u_{n_m}^*) = \arg \max_{\vec{u}_i \in U} F(u_1^{(i)}, \dots, u_s^{(i)}, \dots, u_{n_U}^{(i)}).$$

Шаг 4. Вычисление переменного шага квантования для тупикового управления. Из рекуррентного уравнения (6) временный шаг квантования находим из условия:

$$\Delta t_k = \min_s \{ \tau_s N_m(t_k) [1 - x_s(t_k)] u_s^*(t_k) \}, \quad (16)$$

откуда очередной дискретный момент времени: $t_{k+1} = t_k + \Delta t_k$.

В выражении (16) величина $\tau_s N_m(t_k)$ означает время пребывания на обслуживании z_s задачи при условии, что в режиме разделения времени процессора число обслуживаемых задач равно $N_m(t_k)$, а $[1 - x_s(t_k)]$ – оставшаяся доля, характеризующая состояние задачи.

Шаг 5. Вычисление переменных состояния программы по рекуррентной формуле (6):

$$x_s(t_{k+1}) = x_s(t_k) + \frac{\Delta t_k}{\tau_s} \times \frac{1}{n_m(t_k)} \times u_s^*(t_k).$$

Шаг 6. Проверка условия окончания объема выполнения вычислительных задач. Если в дискретный момент времени $t_k = t_f$ выполняется условие: $x_s(t_f) = 1 \forall s \in S = \bigcup_{l \in L} \bigcup_{a \in A_l} \{1, 2, \dots, n_{z_{la}}\}$, то процесс формирования программы окончен и каждой вычислительной задаче задается интервал ее выполнения на m -м вычислительном средстве:

$$T[u_{ms}^{la}(t)] = [t_{0s}^{la}, t_{fs}^{la}].$$

Интервал времени выполнения сектором управления комплекса задач Z_{la} (2) для a -го КА задается в виде $T_{la} = [t_0^{la}, t_f^{la}]$, где $t_0^{la} = \min_s t_{0s}^{la}$ – срок начала выполнения комплекса задач Z_{la} (2), поступивших на обслуживание; $t_f^{la} = \max_s t_{fs}^{la}$ – конечный срок выполнения комплекса задач.

Задача (15) носит название «задача о рюкзаке» и относится к классу NP-трудных задач комбинаторной оптимизации [23–25]. Математическое обоснование оптимальности изложенного алгоритма тупиковых управлений базируются на следующей теореме.

Теорема (о существовании оптимального решения). Среди всех тупиковых управлений из множества (14) найдется по крайней мере одно тупиковое управление, обеспечивающее минимум целевой функции задачи (15).

Доказательство. Доказательство теоремы о существовании оптимального решения на минимум критерия (15) проведем по схеме доказательства «задачи о рюкзаке» на максимум целевой функции [19]. Пусть $\vec{u}_*$ – оптимальное тупиковое управление, обеспечивающее минимум критерию (15).

Предположим обратное, что $\vec{u}_*$ не является оптимальным и тупиковым управлением, т. е. $\vec{u}_* = (u_1^{(*)}, \dots, u_s^{(*)}, \dots, u_{n_m}^{(*)}) \notin U$. Дополним его до тупикового $\vec{u}_0 = (u_1^{(0)}, \dots, u_s^{(0)}, \dots, u_{n_m}^{(0)}) \in U$. Это значит, что найдется вычислительная задача $z_{s_0} \in Z_m(t_k)$, которая будет включена на обслуживание, и значение критерия (15) при этом увеличится на величину времени τ_{s_0} выполнения этой вычислительной задачи. Тогда будет выполняться неравенство:

$$F(u_1^{(*)}, \dots, u_s^{(*)}, \dots, u_{n_u}^{(*)}) < F(u_1^{(0)}, \dots, u_s^{(0)}, \dots, u_{n_u}^{(0)}).$$

Следовательно, предположение о неоптимальности $\vec{u}_*$ неверно, что и требовалось доказать. Легко видеть, что вычислительные задачи, назначаемые на обслуживание в соответствии с критерием (15), частично упорядочиваются по возрастанию заданного времени τ_s выполнения задачи z_s [26, с. 40–42], что, в свою очередь, обеспечивает минимум критерия I_1 – минимизации времени окончания выполнения комплекса работ [26].

4. Выбор плана работы ЦУП КА с учетом варьирования данных и факторов неопределенностей

На начальном этапе планирования работы ЦУП КА необходимо сгенерировать различные варианты использования вычислительных ресурсов с учетом статистических данных о нештатных ситуациях, которые будем представлять в виде множества альтернативных планов:

$$V = \{v_\gamma(x, u, y, \theta) : \gamma = 1, 2, \dots, n_V\},$$

$$F(v_1, \dots, v_{n_V}) = \sum_{j=1}^{n_I} w(I_j) d_j^{(\gamma)},$$

где x – вектор переменных состояния вычислительных задач;

u – вектор управляющих переменных;

y – вектор загрузки вычислительных средств;

θ – вектор загрузки операторов секторов управления.

Тогда математическая постановка задачи многокритериального оценивания вариантов плана ЦУП КА сводится к задаче векторной оптимизации [18]:

$$I(v_\gamma) = (I_1(v_\gamma), I_2(v_\gamma), \dots, I_6(v_\gamma)) \rightarrow \text{extr}_{v_\gamma \in V},$$

где $d_j^{(\gamma)} = I_j(v_\gamma)$ – оценка плана по I_j -критерию;
 $\gamma = 1, 2, \dots, n_I$ ($n_I = 6$).

Для задач многокритериального оценивания обычно применяется аддитивный механизм агрегирования данных [18]:

где $w(I_j)$ – количественный вес важности I_j критерия. Чтобы применение аддитивного механизма агрегирования данных было корректным, необходимо от исходных оценок по каждому критерию перейти к результирующей однородной шкале [27]. В этом случае выбор эффективного варианта осуществляется по обобщенным $r_\Sigma^{(\gamma)} = \sum_{j=1}^{n_I} w(I_j) r_j^{(\gamma)}$ оценкам, где $r_j^{(\gamma)}$ – оценка v_γ варианта плана по I_j -критерию в результирующей шкале. Веса важности критериев определяются с применением экспертных методов по матрице парных сравнений [28].

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Рассмотрим пример решения задачи интервального планирования комплекса задач методом тупиковых управлений, граф выполнения которых представлен на рис. 2.

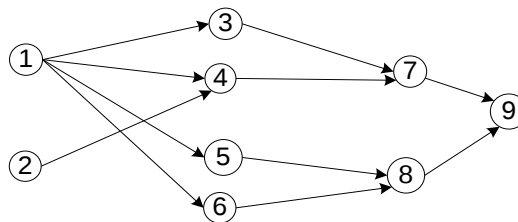


Рис. 2. Граф выполнения комплекса работ

Примечание: составлено автором.

Исходные модельные (условные) данные представлены в табл. 1 для расчета интервалов для 9 заданий, выполняющихся на вычислительном средстве. Максимальный объем

выделяемого ресурса положим равным $B = 180$ для любого $t \in T = [0, t_f)$. Краевые условия для этой задачи запишем в виде: $\vec{x}(0) = (0, 0, 0, 0, 0, 0, 0, 0, 0)$; $\vec{x}(t_f) = (1, 1, 1, 1, 1, 1, 1, 1, 1)$.

Таблица 1

Исходные данные для расчета интервалов

№	Перечень задач, формирующих ПКИ для БА КА	τ_s – время выполнения, в усл. ед. времени	b_s – объем ресурса (оперативной памяти), в усл. ед. памяти
1	Формирование массива данных	1	48
2	Расчет подмассива НИ	3	98
3	Расчет подмассива СС	2	68
4	Расчет подмассива ПСБ	4	60
5	Расчет баллистических параметров	4	62
6	Расчет режимов работы БА КА	3	59
7	Расчет расхода бортовых ресурсов	2	84
8	Формирование ВДП	6	60
9	Компоновка РП	3	32

Примечание: составлено автором. ПКИ – программно-командная информация; БА КА – бортовая аппаратура космического аппарата; НИ – навигационная информация; СС – сеансы связи; ПСБ – панель солнечной батареи; ВДП – временная диаграмма полета; РП – рабочая программа.

Ресурсные ограничения, накладываемые на управляющие переменные, запишем в виде:

$$\sum_{l=1}^{n_U} b_l u_l(t_k) \leq 180 \forall t_k \in T = [0, t_f).$$

Расчет алгоритмом тупиковых управлений

Итерация 1. На шаге 1 для $t_0 = 0$ имеем фронт заданий $Z(t_0) = \{z_1, z_2\}$, удовлетворяющий логическим ограничениям, с которыми отождествим переменные состояния задач:

$$\bar{x}(t_0) = (x_1(0), x_2(0)) = (0, 0).$$

На шаге 2 и 3 имеем единственное тупиковое управление: $\bar{u}(t_0) = (u_1(0), u_2(0)) = (1, 1)$, удовлетворяющее ресурсному ограничению: $48u_1(0) + 98u_2(0) = 146 < 180$.

На шаге 4 вычисляем переменный шаг квантования для $N(t_0) = 2$ задач:

$$\begin{aligned} \Delta t_0 &= \min_{l \in \{1, 2\}} \{\tau_s N(t_0) [1 - x_l(0)]\} = \\ &= \min_{l \in \{1, 2\}} \left\{ \begin{matrix} 1 \times 2 \times [1 - 0] \\ 3 \times 2 \times [1 - 0] \end{matrix} \right\} = \min_{l \in \{1, 2\}} \left\{ \begin{matrix} 2 \\ 6 \end{matrix} \right\} = 2, \end{aligned}$$

тогда дискретный момент времени: $t_1 = t_0 + \Delta t_0 = 2$. Далее на шаге 5 вычисляем значения переменных состояния программы по формуле:

$$x_l(t_{k+1}) = x_l(t_k) + \frac{\Delta t_k}{\tau_s \times N(t_k)} \times u_l(t_k). \quad (17)$$

Имеем $\bar{x}(t_1) = (x_1(2), x_2(2)) = \left(1, \frac{1}{3}\right)$, где

$$x_1(2) = 0 + \frac{2}{1 \times 2} \times 1 = 1; \quad x_2(2) = 0 + \frac{2}{3 \times 2} \times 1 = \frac{1}{3}.$$

Так как критерий останова не выполняется, то перейдем на шаг 1.

Итерация 2. На шаге 1 для $t_1 = 2$ фронт заданий $Z(t_1) = \{z_2, z_3, z_5, z_6\}$, удовлетворяющих логическим ограничениям, с которыми отождествим переменные состояния задач:

$$\bar{x}(t_1) = (x_2(2), x_3(2), x_5(2), x_6(2)) = \left(\frac{1}{3}, 0, 0, 0\right).$$

На шаге 2 построение тупиковых управлений начинается с упорядочения номеров задач в порядке убывания весов: $b_2 \geq b_3 \geq b_5 \geq b_6 \Leftrightarrow 98 > 68 > 62 > 59$, в результате чего получаем три тупиковых управлений для задач z_2, z_3, z_5, z_6 :

$$\begin{aligned} \bar{u}_1(2) &= (1, 1, 0, 0); \quad \bar{u}_2(2) = (1, 0, 1, 0); \\ \bar{u}_3(2) &= (1, 0, 0, 1). \end{aligned}$$

Необходимо заметить, что тупиковыми являются и управления:

$$\bar{u}_4(2) = (0, 1, 1, 0), \quad \bar{u}_5(2) = (0, 0, 1, 1),$$

которые исключаем из рассмотрения, так как в них не входит еще незаконченная z_2 задача.

На шаге 3 за оптимальное принимаем тупиковое $\bar{u}_1(2) = (1, 1, 0, 0)$, обеспечивающее минимальное значение пошаговому критерию (15):

$$F(\bar{u}_1(t_k)) = \sum_{s=1}^n \tau_s(t_k) u_s^{(i)}(t_k),$$

где $\tau_2(2) = 3 \times \left[1 - \frac{1}{3}\right] = 2$ (табл. 2).

Таблица 2

Тупиковые управления итерации 2

№	Тупиковые управления	$\sum_{s=1}^{n_U} b_s u_s^{(i)}(2) \leq 180$	$F(\bar{u}_i(2))$
1	$\bar{u}_1(2) = (1, 1, 0, 0)$	$98 + 68 = 166$	$2 + 2 = 4$
2	$\bar{u}_2(2) = (1, 0, 1, 0)$	$98 + 62 = 160$	$2 + 4 = 6$
3	$\bar{u}_3(2) = (1, 0, 0, 1)$	$98 + 58 = 156$	$2 + 3 = 5$

Примечание: составлено автором.

На шаге 4 вычисляем переменный шаг квантования для $N(t_1) = 2$ задач z_2, z_3 :

$$\Delta t_1 = \min_{l \in \{2, 3\}} \{\tau_s N(t_1) [1 - x_l(2)]\} = \min_{l \in \{2, 3\}} \left\{ \begin{matrix} 3 \times 2 \times \left[1 - \frac{1}{3}\right] \\ 2 \times 2 \times [1 - 0] \end{matrix} \right\} = 4,$$

тогда дискретный момент времени: $t_2 = t_1 + \Delta t_1 = 6$.

Далее на шаге 5 вычисляем значения переменных состояния программы по формуле (17).

Имеем $\bar{x}(t_2) = (x_2(6), x_3(6)) = (1, 1)$, где $x_2(6) = \frac{1}{3} + \frac{4}{3 \times 2} \times 1 = 1$; $x_3(6) = 0 + \frac{4}{2 \times 2} \times 1 = 1$.

Так как критерий останова не выполняется, то перейдем на шаг 1.

Итерация 3. На шаге 1 для $t_2 = 6$ имеем фронт заданий $Z(t_2) = \{z_4, z_5, z_6\}$, удовлетворяющих логическим ограничениям, с которыми отождествим переменные состояния задач:

$$\bar{x}(t_2) = (x_4(6), x_5(6), x_6(6)) = (0, 0, 0).$$

На шаге 2 построение тупиковых управлений начинается с упорядочения номеров за-

дач в порядке убывания весов: $b_4 \geq b_5 \geq b_6 \Leftrightarrow 60 > 62 > 59$, в результате чего получаем три тупиковых управления для задач z_4, z_5, z_6 : $\bar{u}_1(6) = (1, 1, 0)$; $\bar{u}_2(6) = (1, 0, 1)$; $\bar{u}_3(6) = (0, 1, 1)$.

На шаге 3 за оптимальное принимаем тупиковое $\bar{u}_2(6) = (1, 0, 1)$, обеспечивающее минимальное значение пошаговому критерию (табл. 3).

Таблица 3

Тупиковые управления итерации 3

№	Тупиковые управления	$\sum_{s=1}^{m_U} b_s u_s^{(i)}(6) \leq 180$	$F(\bar{u}_i(6))$
1	$\bar{u}_1(6) = (1, 1, 0)$	$60 + 62 = 122$	$4 + 4 = 8$
2	$\bar{u}_2(6) = (1, 0, 1)$	$60 + 59 = 119$	$4 + 3 = 7$
3	$\bar{u}_3(6) = (0, 1, 1)$	$62 + 59 = 121$	$4 + 3 = 7$

Примечание: составлено автором.

На шаге 4 вычисляем переменный шаг квантования для $N(t_2) = 2$ задач z_4, z_6 :

$$\Delta t_2 = \min_{l \in \{4, 6\}} \{\tau_s N(t_2) [1 - x_l(6)]\} =$$

$$= \min_{l \in \{4, 6\}} \left\{ \begin{array}{l} 4 \times 2 \times [1 - 0] \\ 3 \times 2 \times [1 - 0] \end{array} \right\} = 6,$$

тогда дискретный момент времени: $t_3 = t_2 + \Delta t_2 = 12$. Далее на шаге 5 вычисляем значения переменных состояния программы по формуле (17). Имеем:

$$\bar{x}(t_3) = (x_4(12), x_6(12)) = \left(\frac{3}{4}, 1 \right),$$

где $x_4(12) = 0 + \frac{6}{4 \times 2} \times 1 = \frac{3}{4}$; $x_6(12) = 0 + \frac{6}{6 \times 2} \times 1 = 1$.

Так как критерий останова не выполняется, то перейдем на шаг 1.

Итерация 4. На первом шаге для $t_3 = 12$ имеем фронт заданий $Z(t_3) = \{z_4, z_5\}$, удовлетворяющих логическим ограничениям, с которыми отождествим переменные состояния задач:

$$\bar{x}(t_3) = (x_4(12), x_5(12)) = \left(\frac{3}{4}, 0 \right).$$

На шаге 2 и 3 имеем управление: $\bar{u}(t_3) = (u_4(12), u_5(12)) = (1, 1)$, удовлетворяющее ре-

сурсному ограничению: $60u_4(12) + 62u_5(12) = 122 < 180$.

На шаге 4 вычисляем переменный шаг квантования для $N(t_3) = 2$ задач z_4, z_5 :

$$\Delta t_3 = \left\{ \min_{l \in \{4, 5\}} \{\tau_s N(t_3) [1 - x_l(12)]\} \right\} =$$

$$= \min_{l \in \{4, 5\}} \left\{ \begin{array}{l} 4 \times 2 \times \left[1 - \frac{3}{4} \right] \\ 4 \times 2 \times [1 - 0] \end{array} \right\} = 2,$$

тогда дискретный момент времени: $t_4 = t_3 + \Delta t_3 = 14$. Далее на шаге 5 вычисляем значения переменных состояния программы по формуле (17). Имеем:

$$\bar{x}(t_4) = (x_4(14), x_5(14)) = \left(1, \frac{1}{4} \right),$$

где $x_4(14) = \frac{3}{4} + \frac{2}{4 \times 2} \times 1 = 1$; $x_5(14) = 0 + \frac{2}{4 \times 2} \times 1 = \frac{1}{4}$.

Так как критерий останова не выполняется, то перейдем на шаг 1.

Итерация 5. На шаге 1 для $t_4 = 14$ имеем фронт заданий $Z(t_4) = \{z_5, z_7\}$, удовлетворяющих логическим ограничениям, с которыми отождествим переменные состояния задач:

$$\bar{x}(t_4) = (x_5(14), x_7(14)) = \left(\frac{1}{4}, 0 \right).$$

На шаге 2 и 3 имеем управление: $\bar{u}(t_4) = (u_5(14), u_7(14)) = (1, 1)$, удовлетворяющее ограничению: $62u_5(14) + 84u_7(14) = 146 < 180$.

На шаге 4 вычисляем переменный шаг квантования для $N(t_4) = 2$ задач:

$$\Delta t_4 = \{\min\{\tau_s N(t_4)[1 - x_l(14)]\} = \\ = \min_{l \in \{5, 7\}} \left\{ 4 \times 2 \times \left[1 - \frac{1}{4} \right] \right\} = 4,$$

тогда дискретный момент времени: $t_5 = t_4 + \Delta t_4 = 18$. Далее на шаге 5 вычисляем значения переменных состояния программы по формуле (17). Имеем:

$$\bar{x}(t_5) = (x_5(18), x_7(18)) = \left(\frac{3}{4}, 1 \right),$$

где $x_5(18) = \frac{1}{4} + \frac{4}{4 \times 2} \times 1 = \frac{3}{4}$; $x_7(18) = 0 + \frac{4}{2 \times 2} \times 1 = 1$.

Так как критерий останова не выполняется, то перейдем на шаг 1.

Итерация 6. На первом шаге для $t_5 = 18$ имеем фронт заданий $Z(t_5) = \{z_5\}$, удовлетворяющих логическим ограничениям, с которыми отождествим переменные состояния задач:

$$\bar{x}(t_5) = (x_5(18)) = \left(\frac{3}{4} \right).$$

На шаге 2 и 3 имеем управление: $\bar{u}(t_5) = (u_5(18)) = (1)$, удовлетворяющее ограничению: $60u_5(18) = 60 < 180$.

На шаге 4 вычисляем переменный шаг квантования для $N(t_5) = 1$ задач:

$$\Delta t_5 = 4 \times 1 \times \left[1 - \frac{3}{4} \right] = 1,$$

тогда дискретный момент времени: $t_6 = t_5 + \Delta t_5 = 19$. Далее на шаге 5 вычисляем значения переменных состояния программы по формуле (17). Имеем:

$$\bar{x}(t_6) = (x_5(t_6)) = (1),$$

где $x_5(19) = \frac{3}{4} + \frac{1}{4 \times 1} \times 1 = 1$.

Итерация 7. На первом шаге для $t_6 = 19$ имеем фронт заданий $Z(t_6) = \{z_8\}$, удовлетворяющих логическим ограничениям, с которыми отождествим переменные состояния задач:

$$\bar{x}(t_5) = (x_8(18)) = (0).$$

На шаге 2 и 3 имеем управление: $\bar{u}(t_5) = (u_8(18)) = (1)$, удовлетворяющее ограничению: $60u_8(18) = 60 < 180$.

На шаге 4 вычисляем переменный шаг квантования для $N(t_5) = 1$ задач:

$$\Delta t_6 = 6 \times 1 \times [1 - 0] = 6,$$

тогда дискретный момент времени: $t_7 = t_6 + \Delta t_6 = 25$. Далее на шаге 5 вычисляем значения переменных состояния программы по формуле (17). Имеем:

$$\bar{x}(t_6) = (x_8(t_6)) = (1),$$

где $x_8(25) = 0 + \frac{6}{6 \times 1} \times 1 = 1$.

Так как критерий останова не выполняется, то перейдем на шаг 1.

Итерация 8. На шаге 1 для $t_7 = 25$ имеем одну задачу $Z(t_6) = \{z_9\}$, удовлетворяющую логическим ограничениям, с которыми отождествим переменные состояния – $x_9(t_6) = 0$.

На шаге 2 и 3 имеем управление: $u_9(t_7) = 1$, удовлетворяющее ограничению:

$$32u_9(t_6) = 32 < 180.$$

На шаге 4 вычисляем переменный шаг квантования для задачи $z_9(N(t_6) = 1)$:

$$\Delta t_7 = 3 \times 1 \times [1 - 0] = 3,$$

тогда дискретный момент времени: $t_8 = t_7 + \Delta t_7 = 28$. Далее $x_9(28) = 0 + \frac{3}{3 \times 1} \times 1 = 1$. Поскольку выполняется краевое условие $\bar{x}(t_f) = (1, 1, 1, 1, 1, 1, 1, 1)$, для $t_f = 28$, то останов. Результаты вычислений алгоритма тупиковых управлений представлены в табл. 4–5.

Таблица 4

Результаты расчетов

№ итерации	t_k	Оптимальное тупиковое управление $u(t_k)$	Вектор состояния операций $x(t_k)$
1	0	$\bar{u}(0) = (1, 1, 0, 0, 0, 0, 0, 0, 0)$	$\bar{x}(0) = (0, 0, 0, 0, 0, 0, 0, 0, 0)$
2	2	$\bar{u}(2) = (0, 1, 1, 0, 0, 0, 0, 0, 0)$	$\bar{x}(2) = (1, 1/3, 0, 0, 0, 0, 0, 0, 0)$
3	6	$\bar{u}(6) = (0, 0, 0, 1, 0, 1, 0, 0, 0)$	$\bar{x}(6) = (1, 1, 1, 0, 0, 0, 0, 0, 0)$
4	12	$\bar{u}(12) = (0, 0, 0, 1, 1, 0, 0, 0, 0)$	$\bar{x}(12) = (1, 1, 1, 3/4, 0, 1, 0, 0, 0)$
5	14	$\bar{u}(14) = (0, 0, 0, 0, 1, 0, 1, 0, 0)$	$\bar{x}(14) = (1, 1, 1, 1, 1/4, 1, 0, 0, 0)$
6	18	$\bar{u}(18) = (0, 0, 0, 0, 1, 0, 0, 0, 0)$	$\bar{x}(18) = (1, 1, 1, 1, 3/4, 1, 1, 0, 0)$
7	19	$\bar{u}(19) = (0, 0, 0, 0, 0, 0, 0, 0, 1)$	$\bar{x}(19) = (1, 1, 1, 1, 1, 1, 1, 0, 0)$
8	25	$\bar{u}(25) = (0, 0, 0, 0, 0, 0, 0, 0, 1)$	$\bar{x}(25) = (1, 1, 1, 1, 1, 1, 1, 1, 0)$
9	28	z	$\bar{x}(28) = (1, 1, 1, 1, 1, 1, 1, 1, 1)$

Примечание: составлено автором.

Таблица 5

Интервалы выполнения работ

№	Перечень задач, формирующих ПКИ для БА КА	Начало поступления задачи в ВС, усл. ед. времени	Окончание обслуживания задачи в ВС, усл. ед. времени	Длительность интервала обслуживания, усл. ед. времени
1	Формирование массива данных	0	2	2
2	Расчет подмассива НИ	0	6	6
3	Расчет подмассива СС	2	6	4
4	Расчет подмассива ПСБ	6	14	8
5	Расчет баллистических параметров	12	19	7
6	Расчет режимов работы БА КА	6	12	6
7	Расчет расхода бортовых ресурсов	14	18	4
8	Формирование ВДП	19	25	6
9	Компоновка РП	25	28	3

Примечание: составлено автором. Расшифровку аббревиатур см. в примечаниях к табл. 1.

Из табл. 5 следует, что для заданий $z_1 \div z_7$ длительность интервала нахождения в ВС превышает время решения задач при монопольном обслуживании.

ЗАКЛЮЧЕНИЕ

В данной статье показано, что в основе планирования работы ЦУП КА по многим критериям, в частности разработки программы выполнения вычислительных задач, лежит динамическая модель с тупиковыми управлениями, ориентированная прежде всего на снижение вычислительной сложности решения задач, относящихся к классу задач теории расписаний большой размерности. Как показано в данной статье, ветвящийся процесс управления набором заданий сводится на каждом шаге к решению комбинаторной «задачи о рюкзаке», отно-

сящейся к классу NP-полных, временную сложность вычислений которой можно оценить как $O(2^n)$ за n шагов алгоритма.

Для моделей с дискретным временем и дискретным пространством состояний наибольшее применение получили методы динамического программирования, которые требуют большого объема оперативной памяти для запоминания на каждом шаге переменных состояний, что затрудняет решение динамических задач при высокой размерности множества состояний.

Достоинствами метода тупиковых управлений являются его вычислительная простота и более высокое быстродействие по сравнению с известными алгоритмами, что позволяет решать с его помощью характерные для практики задачи большой размерности.

Список источников

1. Нагибин С. Я., Почукаев В. Н., Ухлинов Л. М. Структура и принципы построения математического обеспечения Центра управления полетами космических аппаратов // Вестник Российского общества информатики и вычислительной техники. 1995. Вып. 5. С. 66–72.
2. Мануйлов Ю. С., Калинин В. Н. Управление космическими аппаратами и средствами наземного комплекса управления. СПб. : ВКА им. А. Ф. Можайского, 2010. 609 с.
3. Калинин В. Н. Математическая модель информационного взаимодействия космического аппарата с поверхностью земли // Труды СПИИРАН. 2014. № 3. С. 33–56.
4. Мануйлов Ю. С., Москвин Б. В., Павлов Д. А. Модель планирования распределенной обработки информации в различных звеньях автоматизированных систем управления космическими аппаратами // Труды ВКА им. А. Ф. Можайского. 2014. № 642. С. 32–35.
5. Командровский В. Г. О планировании работ в вычислительной системе // Автоматика и телемеханика. 2005. Вып. 12. С. 65–74.
6. Голубев И. А. Планирование задач в распределённых вычислительных системах на основе метаданных : дис. ... канд. техн. наук. СПб. : Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), 2014. 135 с.
7. Broberg J. A. Effective task assignment strategies for distributed systems under highly variable workloads. Dr. (Philosophy) Thesis. Melbourne, Victoria, Australia: RMIT University; 2006. 208 p.
8. Клейнрок Л. Вычислительные системы с очередями / пер. с англ. под ред. Б. С. Цыбакова. М. : Мир, 1979. 600 с.
9. Баканов В. М. Многомашинные комплексы и многопроцессорные системы. М. : МГУПИ, 2014. 126 с.
10. Колпин М. А., Москвин Б. В. Подход к оптимизации технологии оперативного планирования применения средств наземного автоматизированного комплекса управления космическими аппаратами // i-methods. 2015. Т. 7, № 3. С. 10–14.
11. Минаков Е. П., Привалов А. Е., Бугайченко П. Ю. Модель оценивания эффективности управления многоспутниковыми орбитальными системами // Труды МАИ. 2022. № 125. С. 1–26.
12. Аничкин А. С., Семенов В. А. Современные модели и методы теории расписаний // Труды Института программирования РАН. 2014. Т. 26, № 3. С. 5–50.
13. Лазарев А. А. Теория расписаний. Методы и алгоритмы : моногр. М. : ИПУ РАН, 2019. 408 с.
14. Зимин И. Н., Иванюков Ю. П. Решение задач сетевого планирования сведением их к задачам оптимального управления // Журнал вычислительной математики и математической физики. 1971. Т. 11, № 3. С. 632–641.
15. Калинин В. Н., Соколов Б. В. Динамическая модель и алгоритм оптимального планирования ком-

References

1. Nagibin S. Ya., Pochukaev V. N., Ukhlinov L. M. Struktura i printsipy postroeniia matematicheskogo obespecheniia Tsentra upravleniia poletami kosmicheskikh apparatov. *Vestnik Rossiiskogo obshchestva informatiki i vychislitelnoi tekhniki*. 1995(5):66–72. (In Russian).
2. Manuilov Yu. S., Kalinin V. N. Upravlenie kosmicheskimi apparatami i sredstvami nazemnogo kompleksa upravleniia. St. Petersburg: Mozhaisky Military Aerospace Academy; 2010. 609 p. (In Russian).
3. Kalinin V. N. Mathematical model of informational interaction of the spacecraft with a surface of the Earth. *SPIIRAS Proceedings*. 2014;(3):33–56. (In Russian).
4. Manuilov Yu. S., Moskvina B. V., Pavlov D. A. Model planirovaniia raspredelennoi obrabotki informatsii v razlichnykh zveniyakh avtomatizirovannykh sistem upravleniia kosmicheskimi apparatami. *Proceedings of the Mozhaisky Military Space Academy*. 2014;(642):32–35. (In Russian).
5. Komandrovsky V. G. Job scheduling for a computing system. *Avtomatika i telemekhanika*. 2005;(12):65–74. (In Russian).
6. Golubev I. A. Planirovaniye zadach v raspredelennykh vychislitelnykh sistemakh na osnove metadannykh. Cand. Sci. (Engineering) Thesis. St. Petersburg: Saint Petersburg Electrotechnical University; 2014. 135 p. (In Russian).
7. Broberg J. A. Effective task assignment strategies for distributed systems under highly variable workloads. Dr. (Philosophy) Thesis. Melbourne, Victoria, Australia: RMIT University; 2006. 208 p.
8. Kleinrock L. Queuing systems. Vol. 2. Computer applications. Tsybakov B. S., editor. Moscow: Mir; 1979. 600 p. (In Russian).
9. Bakanov V. M. Mnogomashinnye komplekсы i mnogoprotsessornye sistemy. Moscow: Moscow State University of Instrument Engineering and Computer Science; 2014. 126 p. (In Russian).
10. Kolpin M. A., Moskvina B. V. The approach to technology optimization of operational planning of the use of landing automated complex spaceship control. *i-methods*. 2015;7(3):10–14. (In Russian).
11. Minakov E. P., Privalov A. E., Bugaichenko P. Yu. A model for evaluating the control efficiency of multi-satellite orbital systems. *Trudy MAI*. 2022;(125):1–26. (In Russian).
12. Anichkin A. S., Semenov V. A. A survey of emerging models and methods of scheduling. *Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)*. 2014;26(3):5–50. (In Russian).
13. Lazarev A. A. Teoriia raspisaniy. Metody i algoritmy. Monograph. Moscow: V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences; 2019. 408 p. (In Russian).
14. Zimin I. N., Ivanilov Yu. P. Solution of network planning problems by reducing them to optimal control problems. *Zhurnal vychislitelnoi matematiki i matematicheskoi fiziki*. 1971;11(3):632–641. (In Russian).

- плекса работ с запретами на прерывание // Автоматика и телемеханика. 1987. Вып. 1. С. 106–114.
16. Ivanov D., Dolgui A., Sokolov B. et al. A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory industry 4.0. *International Journal of Production Research*. 2016;54(2):386–402.
17. Richter K. Dynamic tasks of discrete optimization. Berlin, Boston: De Gruyter; 1982. 114 p.
18. Корнеев В. П. Методы оптимизации. М. : Высш. шк., 2007. 663 с.
19. Korneenko V. P. An efficient algorithm of dead-end controls for solving combinatorial optimization problems. *Automation and Remote Control*. 2021;82:1692–1705.
20. Яблонский С. В. Введение в дискретную математику. М. : Наука, 1986. 384 с.
21. Бендат Дж., Пирсол А. Прикладной анализ случайных данных / пер. с англ. В. Е. Привальского, А. И. Кочубинского ; под ред. И. Н. Коваленко. М. : Мир, 1989. 540 с.
22. Климов Г. П., Ляху А. К., Матвеев В. Ф. Математические модели систем с разделением времени. Кишинев : Штиинца, 1983. 110 с.
23. Беллман Р. Динамическое программирование / пер. с англ. И. М. Андреевой, А. А. Корбута, И. В. Романовского, И. Н. Соколовой ; под ред. Н. Н. Воробьева. М. : Изд-во иностр. лит., 1960. 400 с.
24. Сигал И. Х., Иванова А. П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы. М. : Физматлит, 2002. 240 с.
25. Pisinger D. A minimal algorithm for the 0-1 Knapsack Problem. *Operations Research*. 1997;46(5):758–767.
26. Конвей Р. Б., Максвелл В. Л., Миллер Л. В. Теория расписаний / пер. с англ. В. А. Кокотушкина, Д. Г. Михалева ; под ред. Г. П. Башарина. М. : Наука, 1975. 359 с.
27. Корнеев В. П. Метод локального агрегирования данных объектов с многоуровневой структурой в порядковых шкалах // Управление развитием крупномасштабных систем (MLSD-2021) : труды 14-й Междунар. конф., 27–29 сентября 2021 г., г. Москва. М. : ИПУ РАН, 2021. С. 485–493.
28. Корнеев В. П. Метод аппроксимационной матрицы формирования весов объектов в многокритериальных задачах выбора // Вестник кибернетики. 2021. № 1. С. 51–62.
15. Kalinin V. N., Sokolov B. V. A dynamic model and an optimal scheduling algorithm for a set of activities with bans on interrupts. *Avtomatika i telemekhanika*. 1987;(1):106–114. (In Russian).
16. Ivanov D., Dolgui A., Sokolov B. et al. A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory industry 4.0. *International Journal of Production Research*. 2016;54(2):386–402.
17. Richter K. Dynamic tasks of discrete optimization. Berlin, Boston: De Gruyter; 1982. 114 p. (In German).
18. Korneenko V. P. Metody optimizatsii. Moscow: Vyssh. shk.; 2007. 663 p. (In Russian).
19. Korneenko V. P. An efficient algorithm of dead-end controls for solving combinatorial optimization problems. *Automation and Remote Control*. 2021;82:1692–1705.
20. Yablonsky S. V. Vvedenie v diskretnuiu matematiku. Moscow: Nauka; 1986. 384 p. (In Russian).
21. Bendat J. S., Piersol A. G. Random data: Analysis and measurement procedures. Privalsky V. E., Kochubinsky A. I., translators; I. N. Kovalenko, editor. Moscow: Mir; 1989. 540 p. (In Russian).
22. Klimov G. P., Lyakhu A. K., Matveev V. F. Matematicheskie modeli sistem s razdeleniem vremeni. Kishinev: Shtiintsa; 1983. 110 p. (In Russian).
23. Bellman R. Dynamic programming. Andreeva I. M., Korbut A. A., Romanovsky I. V., Sokolova I. N., translators; N. N. Vorobyev, editor. Moscow: Izd-vo inostr. lit.; 1960. 400 p. (In Russian).
24. Sigal I. Kh., Ivanova A. P. Vvedenie v prikladnoe diskretnoe programmirovaniye: modeli i vychislitelnye algoritmy. Moscow: Fizmatlit; 2002. 240 p. (In Russian).
25. Pisinger D. A minimal algorithm for the 0-1 Knapsack Problem. *Operations Research*. 1997;46(5):758–767.
26. Conway R. W., Maxwell W. L., Miller L. W. Theory of scheduling. Kokotushkin V. A., Mikhalev D. G., translators; Basharin G. P., editor. Moscow: Nauka; 1975. 359 p. (In Russian).
27. Korneenko V. P. Metod lokalnogo agregirovaniia dannykh obektov s mnogourovnevnoi strukturoi v poriadkovykh shkalakh. In: *Proceedings of the Fourteenth International Conference "Management of Large-Scale System Development (MLSD-2021)"*, September 27–29, 2021, Moscow. Moscow: V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences; 2021. p. 485–493. (In Russian).
28. Korneenko V. P. Approximation matrix method for weights formation of objects in multicriteria problems. *Proceedings in Cybernetics*. 2021;(1):51–62. (In Russian).

Информация об авторе

В. П. Корнеев – кандидат технических наук, доцент.

Information about the author

V. P. Korneenko – Candidate of Sciences (Engineering), Docent.