ИЗВЕСТИЯ РАН. ТЕОРИЯ И СИСТЕМЫ УПРАВЛЕНИЯ, 2025, № 1, С. 90–98

СИСТЕМНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

УДК 519.86

ПРОЕКТИРОВАНИЕ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ С ЗАДАННЫМИ ХАРАКТЕРИСТИКАМИ

© 2025 г. М. Г. Фуругян[a, *]

[a]ФИЦ ИУ РАН, Москва, Россия
**e-mail: rtsccas@yandex.ru*

Поступила в редакцию 24.06.2024 г.
После доработки 26.10.2024 г.
Принята к публикации 13.01.2025 г.

Рассматривается задача определения параметров вычислительной системы реального времени (производительности процессоров, объемы и эффективность использования ресурсов), позволяющих выполнить заданный комплекс работ в заранее установленные сроки. В случае невозможности подбора таких параметров решается задача минимальной коррекции характеристик заданий (директивные интервалы и объемы работ). Для решения указанных задач применяется сетевое моделирование и алгоритмы нахождения потоков с заданными свойствами в сетях с выигрышами.

Ключевые слова: многопроцессорная система реального времени, распределение ресурсов, допустимое расписание, сетевая модель, максимальный поток, невозобновляемые ресурсы

DOI: 10.31857/S0002338825010079 **EDN:** AHCXEG

DESIGNING A REAL-TIME COMPUTING SYSTEM WITH SPECIFIED CHARACTERISTICS

M.G. Furugyan[a, *]

[a]Federal Research Center "Computer Science and Control" of the RAS, Moscow, Russia
**e-mail: rtsccas@yandex.ru*

The problem of determining the parameters of a real-time computing system (processor performance, volume and efficiency of resource use), allowing to perform a given set of jobs in a predetermined time frame, is considered. If it is impossible to select such parameters, the problem of minimal correction of job characteristics (directive intervals and job volumes) is solved. To solve these problems, network modeling and algorithms for finding flows with specified properties in networks with winnings are used.

Keywords: multiprocessor real-time system, resource allocation, admissible schedule, network model, maximum flow, non-renewable resource

Введение. С конца 70-х гг. прошлого века вычислительные системы реального времени стали широко внедряться в различные области деятельности человека. Главным образом они используются там, где требуемые вычисления необходимо выполнить в заранее установленные временные интервалы. В системах жесткого реального времени такие интервалы могут составлять доли секунды. Например, подобная ситуация имеет место при испытаниях и эксплуатации летательных аппаратов, атомных реакторов, при наблюдении за космическими объектами, проведении военных операций. Менее жесткие временные ограничения возникают при обработке информации экономического и экологического характера.

Во всех указанных выше случаях требуется решение следующих задач. Во-первых, необходимо построить расписание, согласно которому все вычисления будут проведены в заданном темпе. Во-вторых, надо определить производительность системы, позволяющую это сделать. И в-третьих, в случае необходимости, минимальным образом скорректировать параметры системы и характеристики заданий.

По вопросам построения расписаний имеется большое количество публикаций. Так, в работе [1] подробно исследуются задачи, связанные с разработкой алгоритмов построения расписаний для систем обслуживания с одним и несколькими приборами. Рассматриваются задачи как с независимыми заданиями, так и с работами, связанными отношениями предшествования. Большое внимание уделяется вопросам вычислительной сложности задач и построенных алгоритмов. Исследован ряд полиномиально разрешимых и NP-трудных задач. В работе [2] приводится подробная классификация задач теории расписаний. Обсуждаются некоторые задачи комбинаторной оптимизации, задачи составления расписаний для систем с одной и несколькими параллельными машинами, задачи на минимизацию времени выполнения заданий и соблюдения директивных сроков, метод ветвей и границ. Рассмотрены вопросы вычислительной сложности алгоритмов. В отличие от [1, 2], в работе [3] изучаются задачи построения расписаний в многостадийных системах. Исследуются различные задачи с одинаковыми последовательностями прохождения приборов, а также различными и нефиксированными маршрутами. Развивается теоретико-игровой подход к анализу таких задач.

В работах [4, 5] развивается метод “ветвей и границ” для решения задач теории расписаний. Предполагается, что некоторые параметры, такие как длительности выполнения заданий и объемы имеющихся ресурсов, не являются фиксированными. Рассмотрены случаи, когда эти параметры могут принимать значения из заданных интервалов либо являются случайными величинами. При этом множества значений этих параметров разбиваются на области, внутри каждой из которых расписание имеет неизменную структуру. В работах [6, 7] исследован ряд NP-трудных задач составления однопроцессорных и многопроцессорных расписаний. Предложены алгоритмы для критериев минимизации времени выполнения всего комплекса работ, а также минимизации максимального запаздывания. Вводится понятие расстояния между заданиями, на основе которого предложена методика нахождения приближенных решений.

В работах [8–10] предложен метод решения задачи построения расписаний с директивными интервалами для многоядерной вычислительной системы реального времени. Метод основан на построении временной диаграммы, описывающей работу системы. Это позволяет выполнять непосредственную проверку ограничений реального времени, заключающихся в том, что каждая работа успевает завершиться в своем директивном интервале. Указанные исследования проведены с помощью имитационной модели, основанной на использовании обобщенных конечных автоматов.

Упомянутые выше работы посвящены задачам распределения нескладируемых ресурсов, т.е. таких ресурсов, которые могут применяться многократно (приборы, процессоры, машины, станки и т.д.). В отличие от них, складируемые ресурсы повторно использоваться не могут. Примерами являются финансы, горюче-смазочные материалы, электроэнергия, устройства вычислительной системы, предназначенные для конкретного программного модуля. В работах [11, 12] рассмотрены задачи минимизации времени выполнения комплекса работ, потребляющих складируемые ресурсы. При этом предполагается, что длительности выполнения заданий линейно зависят от объема выделенных им ресурсов. В работе [11] также рассмотрена задача минимизации потребления ресурсов при заданном директивном сроке выполнения всех работ.

В работе [13] изучена задача составления допустимого расписания со смешанным набором ресурсов – складируемых и нескладируемых. Решение этой задачи сведено к поиску максимального потока в сети специального вида. Задача нахождения производительностей процессоров, при которых существует допустимое расписание в системе с однородным набором ресурсов, исследована в работе [14] и сведена к системе линейных ограничений.

В настоящей статье рассматривается задача планирования комплекса работ в вычислительной системе реального времени со смешанным набором ресурсов – складируемых и нескладируемых. Исследуется вопрос о существовании и построении решения (допустимого распределения ресурсов и допустимого расписания). В случае, если решения не существует, ставится вопрос о корректировке параметров системы (производительности процессоров, объемы и эффективности ресурсов) и характеристик работ (объемы и директивные интервалы). Предлагаемая методика основана на использовании потоковых сетей с выигрышами. Решение ука-

занных задач имеет большое значение при проектировании и функционировании сложных технических объектов, в частности бортовых систем управления.

1. Обозначения и определения. Введем следующие обозначения: $W=\{w_1, w_2, ..., w_n\}$ – комплекс работ (заданий), которые должны быть выполнены с использованием вычислительной системы, состоящей из m процессоров (нескладируемые ресурсы) и набора складируемых ресурсов. Каждая работа w_i имеет две характеристики – директивный интервал $[b_i;f_i]$ (задание w_i может выполняться только в этом временном интервале) и объем V_i. Далее $P_1, P_2, ..., P_m$ – процессоры, производительности которых равны $s_1, s_2, ..., s_m$ соответственно. Каждая работа может выполняться на любом процессоре. Допускаются прерывания и переключения с одного процессора на другой, которые, по предположению, не требуют временных затрат. Кроме того, возможно параллельное выполнение одной работы несколькими процессорами (в частности, всеми процессорами вместе). Не допускается одновременное выполнение нескольких работ одним процессором. Если процессор с производительностью s выполняет некоторое задание в течение интервала времени δ, то объем сделанной им работы составляет $s\delta$.

Помимо процессоров для выполнения заданий могут использоваться L типов складируемых ресурсов. Для краткости в дальнейшем будем называть их просто ресурсами. Их объемы составляют R_1, R_2, ..., R_L соответственно. В отличие от процессоров, ресурсы повторно использоваться не могут. Если для выполнения задания w_i выделено r_{li} единиц ресурса l-го типа, то это обеспечивает объем работы, равный $r_{li}z_{li}$, где z_{li} – эффективность применения ресурса l-го типа для задания w_i. В отличие от работы [13], эффективность зависит как от типа ресурса, так и от задания. Работе w_i должно быть выделено не менее r_{li}^0 и не более r_{li}^1 единиц ресурса l-го типа, $l=\overline{1,L}$. Величина V_i – это суммарный объем работы, обеспечиваемый процессорами и ресурсами, необходимый для выполнения задания w_i.

Допустимым распределением ресурсов будем называть распределение r_{li}, которое удовлетворяет ограничениям:

$$r_{li}^0 \le r_{li} \le r_{li}^1, l=\overline{1,L}, i=\overline{1,n}, \tag{1.1}$$

$$\sum_{i=1}^{n} r_{li} \le R_l, l=\overline{1,L}. \tag{1.2}$$

Расписание для комплекса W показывает для каждой работы $w_i \in W$, в какие временные интервалы времени какими процессорами она выполняется. Допустимое расписание – это такое расписание для W, при котором каждое задание $w_i \in W$ полностью исполняется в своем директивном интервале $[b_i;f_i]$. Решением задачи будем называть пару "допустимое распределение ресурсов – допустимое расписание".

2. Постановка задачи. Входными данными задачи обозначим совокупность характеристик заданий (директивные интервалы и объемы) и параметров системы (производительности процессоров, объемы и эффективность использования ресурсов). Требуется решить следующие задачи.

З а д а ч а 1. При фиксированных входных данных определить, существует ли решение, и найти его при положительном ответе.

З а д а ч а 2. В случае отрицательного ответа в задаче 1 скорректировать параметры системы и характеристики заданий так, чтобы решение существовало.

Решение сформулированных задач основано на построении потоковой сети и нахождении в ней потоков с определенными свойствами. В отличие от [13], будем применять сети с выигрышами [11].

3. Решение задачи 1. Используем аппарат потоковых сетей с выигрышами. Сеть с выигрышами отличается от обычной сети тем, что в некоторых узлах величина выходящего потока отличается от величины входящего потока некоторой мультипликативной константой [11]. А именно величина выходящего потока из узла a равна величине входящего потока в узел a, умноженной на постоянное для данного узла число $c(a)$, называемое коэффициентом выигрыша узла a.

Пусть $y_0 < y_1 < ... < y_p$ – все различные величины b_i и f_i, $i=\overline{1,n}$. Построим потоковую ориентированную сеть с выигрышами $G=(N,A)$ (см. рисунок). Определим множество узлов:

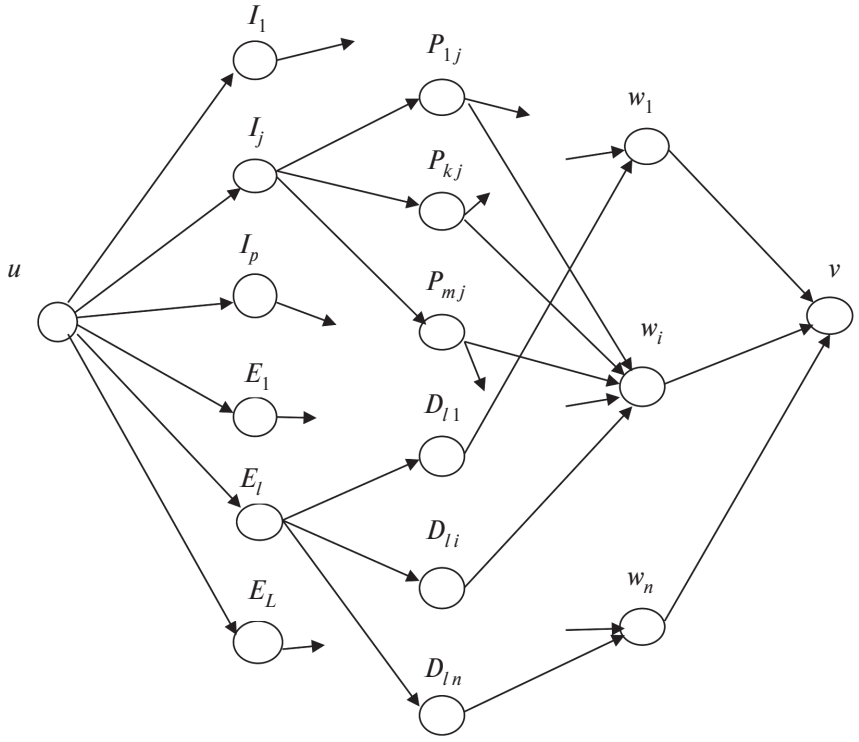

Рисунок. Потоковая сеть G с выигрышами.

$$N = \left\{u, I_j, E_l, D_{li}, P_{kj}, w_i, v\right\}, j = \overline{1,p}\ ,\ l = \overline{1,L}, k = \overline{1,m}\ ,\ i = \overline{1,n}\ .$$

Здесь u – источник, v – сток, I_j соответствует интервалу $[y_{j-1}; y_j]$, E_l – ресурсу l-го типа, D_{li} – ресурсу l-го типа и заданию w_i, P_{kj} – работе процессора P_k в интервале I_j, w_i – заданию $w_i \in W$. Определим множество дуг:

$$A = \left\{(u, I_j),(u, E_l),(E_l, D_{li}),(I_j, P_{kj}),(P_{kj}, w_i),(D_{li}, w_i),(w_i, v)\right\},$$

$j = \overline{1,p}$, $l = \overline{1,L}$, $k = \overline{1,m}$, $i = \overline{1,n}$.

Дуга (P_{kj}, w_i) вводится в сеть G в том случае, если $I_j \subseteq [b_i; f_i]$, т.е. если работа w_i может выполняться в интервале I_j. Сеть G содержит $O(pm + Ln)$ узлов и $O((pm + Ln)n)$ ориентированных дуг.

Узел P_{kj} имеет коэффициент выигрыша, равный s_k. Это означает, что если в узел P_{kj} по дуге (I_j, P_{kj}) входит поток g, то величина суммарного потока, выходящего из узла P_{kj} по дугам (P_{kj}, w_i), равна $s_k g$, т.е.

$$\sum_{i=1}^{n} g\left(P_{kj}, w_i\right) = s_k g\left(I_j, P_{kj}\right). \tag{3.1}$$

Коэффициент выигрыша узла D_{li} равен z_{li}. Это означает, что если в узел D_{li} по дуге (E_l, D_{li}) входит поток g, то величина потока, выходящего из узла D_{li} по дуге (D_{li}, w_i), равна $z_{li} g$, т.е.

$$g\left(D_{li}, w_i\right) = z_{li} g\left(E_l, D_{li}\right). \tag{3.2}$$

Коэффициенты выигрыша остальных узлов равны 1, т.е. для них выполняется условие сохранение потока. В узлах P_{kj} и D_{li} это условие трансформируется в равенства (3.1), (3.2) соответственно. Каждая дуга $(x, y) \in A$ сети G имеет два параметра: $L(x, y)$ – нижняя граница потока по дуге (x, y), $U(x, y)$ – верхняя граница потока по дуге (x, y). Значения параметров L и U приведены в табл. 1.

В табл. 2 поясняется смысловое значение потока g по дугам сети G.

Предположим, что в сети G существует некоторый поток g. С помощью этого потока и пояснений, содержащихся в табл. 2, можно построить решение задачи 1. Используя дополнительно соотношения (3.1), (3.2), сделаем следующие выводы.

Таблица 1. Значения параметров дуг сети G

Дуга	L	U
(u, I_j)	0	$m\delta_j$
(u, E_l)	0	R_l
(E_l, D_{li})	r_{li}^0	r_{li}^1
(I_j, P_{kj})	0	δ_j
(P_{kj}, w_i)	0	$s_k\delta_j$
(D_{li}, w_i)	$z_{li}r_{li}^0$	$z_{li}r_{li}^1$
(w_i, v)	V_i	V_i

Таблица 2. Физическая интерпретация потока g по дугам сети G

Поток по дуге	Физический смысл потока по дуге
$g(u, I_j)$	Суммарное распределяемое процессорное время в интервале I_j
$g(u, E_l)$	Суммарное распределяемое количество ресурса l-го типа
$g(E_l, D_{li})$	Объем ресурса l-го типа, выделяемый заданию w_i
$g(I_j, P_{kj})$	Суммарное время работы процессора P_k в интервале I_j
$g(P_{kj}, w_i)$	Объем работы процессора P_k по выполнению задания w_i в интервале I_j
$g(D_{li}, w_i)$	Объем работы по выполнению задания w_i с помощью ресурса l-го типа
$g(w_i, v)$	Суммарный объем работы по выполнению задания w_i с помощью процессоров и ресурсов

1. Структура сети G такова, что каждая работа w_i выполняется только в своем директивном интервале $[b_i; f_i]$.
2. Суммарное время работы процессора P_k в интервале I_j составляет

$$\sum_{i=1}^{n} g\left(P_{kj}, w_i\right) \Big/ s_k = g\left(I_j, P_{kj}\right) \le \delta_j.$$

Такое расписание выполнения работ W процессорами $P_1, P_2, \ldots, P_m$ может быть реализовано, поскольку по условию задачи допускается параллельное выполнение одного задания произвольным числом процессоров.

3. Суммарное время работы всех m процессоров в интервале I_j составляет

$$\sum_{k=1}^{m} g\left(I_j, P_{kj}\right) = g\left(u, I_j\right) \le m\delta_j,$$

что допустимо.

4. Объем ресурса l-го типа, выделяемый работе w_i, составляет $g(E_l, D_{li})$, что удовлетворяет условию задачи, поскольку $r_{li}^0 \le g\left(E_l, D_{li}\right) = r_{li} \le r_{li}^1$.

5. Суммарный объем ресурса l-го типа, используемый для выполнения работ, составляет

$$\sum_{i=1}^{n} g\left(E_l, D_{li}\right) = g\left(u, E_l\right) \le R_l,$$

что удовлетворяет условию задачи.

6. Каждая работа выполнена полностью, поскольку $g(w_i, v) = V_i$ при всех $i = \overline{1, n}$.

Таким образом, если в сети G существует поток, то в задаче 1 существует решение.

Покажем теперь, что из существования решения в задаче 1 следует существование потока g в сети G. Пусть работе w_i выделено r_{li} единиц ресурса l-го типа. Определим поток по дуге (E_l, D_{li}) равным этой величине, т.е. $g(E_l, D_{li}) = r_{li}$. Пусть далее $g(D_{li}, w_i) = z_{li}r_{li}$,

$$g\left(u, E_l\right) = \sum_{i=1}^{n} g\left(E_l, D_{li}\right) = \sum_{i=1}^{n} r_{li}.$$

Тогда из (1.1), (1.2) следует, что

$$z_{li}r_{li}^0 \le g(D_{li}, w_i) = g(E_l, D_{li})z_{li} \le z_{li}r_{li}^1,\ 0 \le g(u, E_l) \le R_l.$$

Допустим, что t_{kji} – продолжительность выполнения работы w_i процессором P_k в интервале I_j ($t_{kji} \le \delta_j$). Определим $g(P_{kj}, w_i) = s_k t_{kji}$,

$$g(I_j, P_{kj}) = \sum_{i=1}^{n} t_{kji}.$$

Тогда

$$0 \le g(P_{kj}, w_i) \le s_k\delta_j,\ 0 \le g(I_j, P_{kj}) \le \delta_j.$$

Положим, что

$$g(u, I_j) = \sum_{k=1}^{m} g(I_j, P_{kj}).$$

Тогда $0 \le g(u, I_j) \le m\delta_j$. Наконец, определим

$$g(w_i, v) = \sum_{l=1}^{L} g(E_l, w_i) + \sum_{j=1}^{p}\sum_{k=1}^{m} g(P_{kj}, w_i),$$

т.е. $g(w_i, v)$ – это суммарный объем работы по выполнению задания w_i, предоставляемый процессорами и ресурсами. Поскольку работа w_i сделана полностью, то $g(w_i, v) = V_i$. Следовательно, не нарушены ограничения на поток по всем дугам сети G, заданные в табл. 1. Кроме того, верны условия сохранения потока в узлах I_j, $j=\overline{1,p}$, E_l, $l=\overline{1,L}$, и w_i, $i=\overline{1,n}$, а также условия (3.1), (3.2) для узлов D_{li}, $l=\overline{1,L}$, $i=\overline{1,n}$ и P_{kj}, $k=\overline{1,m}$, $j=\overline{1,p}$. Тогда g является потоком в сети G.

Таким образом, доказано, следующее утверждение.

Л е м м а. В задаче 1 решение существует в том и только том случае, когда в сети G существует поток.

Из леммы 1 и проведенных выше рассуждений вытекает следующий алгоритм решения задачи 1.

Ш а г 1. Построить сеть G.

Ш а г 2. Найти в сети G поток g. (Для этого может быть использован, например, алгоритм дефекта [15]. При этом стоимость единицы потока по каждой дуге полагается равной нулю.) Если поток существует, то перейти на шаг 3. В противном случае – на шаг 4.

Ш а г 3. Работе w_i следует выделить ресурс l-го типа в количестве $g(D_{li}, w_i)/z_{li}$, $l=\overline{1,L}$, и выполнять ее процессором P_k в интервале I_j в течение времени

$$g(P_{kj}, w_i)/s_k,\ k=\overline{1,m},\ j=\overline{1,p},\ i=\overline{1,n}.$$

Завершение алгоритма.

Ш а г 4. Решения не существует.

Оценим вычислительную сложность предложенного алгоритма. Учитывая неравенство $p \le 2n-1$, получаем, что число узлов и число дуг в сети G составляет соответственно $O(mn+Ln)$ и $O(mn^2+Ln)$. Тогда сложность отдельных этапов алгоритма следующая: шаг 1 – $O(mn^2+Ln)$, шаг 2 –

$$O\left((mn^2+Ln)^2 \max_{(x,y)\in A} U(x,y)\right),$$

шаг 3 – $O(mn+Ln)$. Таким образом, вычислительная сложность алгоритма составляет

$$O\left(\left(mn^2 + Ln\right)^2 \max_{(x,y)\in A} U(x,y)\right).$$

З а м е ч а н и е. Одним из действий алгоритма дефекта [15], применяемого на шаге 2, является поиск увеличивающего пути. Следует учесть, что при прохождении через узел D_{li} по дугам (E_l, D_{li}) и (D_{li}, w_i) поток по дуге (D_{li}, w_i) необходимо умножить на z_{li}. Аналогично при прохождении через узел P_{kj} по дугам (I_j, P_{kj}) и (P_{kj}, w_i) поток по дуге (P_{kj}, w_i) нужно умножить на s_k. При прохождении через узлы D_{li} и P_{kj} по дугам (w_i, D_{li}), (D_{li}, E_i) и (w_i, P_{kj}), (P_{kj}, I_j) соответственно (т.е. в случае, когда эти дуги используются как обратные) потоки по дугам (D_{li}, E_l) и (P_{kj}, I_j) следует разделить на z_{li} и s_k.

4. Решение задачи 2. Используя поток g в сети G и данные из табл. 1 и 2, опишем задачу 1 в виде следующей системы линейных ограничений, в которой требуется найти значения переменных $g(u, I_j)$, $g(u, E_l)$, $g(E_l, D_{li})$, (P_{kj}, w_i), (D_{li}, w_i), (w_i, v), $j=\overline{1,p}$, $l=\overline{1,L}$, $k=\overline{1,m}$, $i=\overline{1,n}$:

$$g\left(u, I_j\right) = \sum_{k=1}^{m} g\left(I_j, P_{kj}\right), j = \overline{1,p}, \tag{4.1}$$

$$g\left(u, E_l\right) = \sum_{i=1}^{n} g\left(E_l, D_{li}\right), l = \overline{1,L}, \tag{4.2}$$

$$z_{li} g\left(E_l, D_{li}\right) = g\left(D_{li}, w_i\right), l = \overline{1,L}, i = \overline{1,n}, \tag{4.3}$$

$$s_k g\left(I_j, P_{kj}\right) = \sum_{i=1}^{n} g\left(P_{kj}, w_i\right), j = \overline{1,p}, k = \overline{1,m}, \tag{4.4}$$

$$\sum_{j=1}^{p}\sum_{k=1}^{m} g\left(P_{kj}, w_i\right) + \sum_{l=1}^{L} g\left(E_l, w_i\right) = g\left(w_i, v\right), i = \overline{1,n}, \tag{4.5}$$

$$0 \le g\left(u, I_j\right) \le m\delta_j, j = \overline{1,p}, \tag{4.6}$$

$$0 \le g\left(u, E_l\right) \le R_l, l = \overline{1,L}, \tag{4.7}$$

$$0 \le g\left(I_j, P_{kj}\right) \le \delta_j, j = \overline{1,p}, k = \overline{1,m}, \tag{4.8}$$

$$0 \le g\left(P_{kj}, w_i\right) \le s_k\delta_j, j = \overline{1,p}, k = \overline{1,m}, \tag{4.9}$$

$$r_{li}^0 \le g\left(E_l, D_{li}\right) \le r_{li}^1, i = \overline{1,n}, l = \overline{1,L}, \tag{4.10}$$

$$z_{li} r_{li}^0 \le g\left(D_{li}, w_i\right) \le z_{li} r_{li}^1, i = \overline{1,n}, l = \overline{1,L}, \tag{4.11}$$

$$g\left(w_i, v\right) = V_i, i = \overline{1,n}. \tag{4.12}$$

Данная система содержит $O(pmn + Ln)$ переменных и $O(pm + Ln)$ линейных ограничений.

Перейдем к решению задачи 2. Предположим, что при исходных данных решения в задаче 1 не существует. Определим, каким образом следует изменить параметры системы и характеристики заданий, чтобы решение существовало. Сначала исследуем, как надо увеличить производительности процессоров, чтобы решение в задаче 1 существовало. При этом будем минимизировать максимальное приращение производительностей. Пусть производительность s_k процессора P_k увеличена на s_k^0. Найдем такие величины s_k^0, при которых

$$\max_{k=\overline{1,m}} s_k^0$$

принимает минимальное значение. Это означает, что производительности всех процессоров следует увеличить на одну и ту же величину. Иными словами, надо минимизировать величину s^0 при условии, что с производительностями процессоров $s_k + s^0$, $k = \overline{1,m}$, решение в задаче 1 существует. Таким образом, получаем следующую задачу линейного программирования: найти $\min s^0$ при условиях (4.1)–(4.3), (4.5)–(4.8), (4.10)–(4.12) и ограничениях

$$(s_k + s^0)g\left(I_j, P_{kj}\right) = \sum_{i=1}^{n} g\left(P_{kj}, w_i\right), j = \overline{1,p}, k = \overline{1,m},$$

$$0 \le g\left(P_{kj}, w_i\right) \le (s_k + s^0)\delta_j, j = \overline{1,p}, k = \overline{1,m}.$$

Аналогично для определения минимального увеличения объема R^0 ресурсов каждого типа получаем задачу линейного программирования: найти $\min R^0$ при условиях (4.1)–(4.6), (4.8)–(4.12) и ограничениях

$$0 \le g\left(u, E_l\right) \le R_l + R^0, l = \overline{1,L}.$$

Для определения минимального увеличения эффективности z^0 ресурсов каждого типа получаем задачу линейного программирования: найти $\min z^0$ при условиях (4.1), (4.2), (4.4)–(4.10), (4.12) и ограничениях

$$(z_{li} + z^0)g\left(E_l, D_{li}\right) = g\left(D_{li}, w_i\right), i = \overline{1,n}, l = \overline{1,L},$$

$$(z_{li} + z^0)r_{li}^0 \le g\left(D_{li}, w_i\right) \le (z_{li} + z^0)r_{li}^1, i = \overline{1,n}, l = \overline{1,L}.$$

Для определения минимального уменьшения объема работ V^0 получаем задачу линейного программирования: найти $\min V^0$ при условиях (4.1)–(4.11) и ограничениях

$$g\left(w_i, v\right) = V_i - V^0, i = \overline{1,n}.$$

Задача минимизации суммарной стоимости процессоров при условии существования решения в задаче 1 выглядит следующим образом: найти

$$\min_{s_1,\ldots,s_m} \sum_{k=1}^{m} s_k c_k^1$$

при условиях (4.1)–(4.12) и ограничениях $s_k \le s_k^1$, $k = \overline{1,m}$, где c_k^1 – стоимость одной единицы производительности процессора P_k, s_k^1 – верхняя допустимая граница производительности процессора P_k.

Задача минимизации суммарной стоимости используемых ресурсов при условии существования решения в задаче 1 выглядит следующим образом: найти

$$\min_{R_1,\ldots,R_l} \sum_{l=1}^{L} R_l c_l^2$$

при условиях (4.1)–(4.12) и ограничениях $R_l \le R_l^1$, $l = \overline{1,L}$, где c_l^2 – стоимость одной единицы ресурса l-го типа, R_l^1 – верхняя допустимая граница объема ресурса l-го типа.

Эту задачу можно также решить и путем поиска потока минимальной стоимости в сети G. Для этого надо определить стоимость единицы потока по дуге (u, E_l), равной c_l^2, $l = \overline{1,L}$, а в качестве верхней границы объема ресурса l-го типа взять R_l^1, $l = \overline{1,L}$. Далее, используя алгоритм дефекта, найдем искомые величины R_l, $l = \overline{1,L}$, если поток в сети G существует. Если же потока не существует, то и решения в этой задаче нет.

Задачу минимизации увеличения директивных сроков заданий, при котором имеется решение задачи 1, будем решать в предположении, что все начальные директивные сроки b_i совпадают, т.е. $b_i = b_0$, $i = \overline{1,n}$. Без ограничения общности можно считать, что $f_1 < f_2 < \ldots < f_n$. Попрежнему предполагаем, что требуется минимизировать максимальное увеличение директивного срока. Поэтому можно считать, что все величины f_i увеличиваются на одно и то же значение, например на ε. В этом случае длина интервала I_1 увеличится на ε, а длины остальных интервалов I_j, $j = \overline{2,p}$, останутся без изменения. Таким образом, получаем следующую задачу линейного программирования: найти $\min \varepsilon$ при условиях (4.1)–(4.5), (4.7), (4.10) (4.12), ограничениях

$$0 \le g(u, I_1) \le m(\delta_1 + \varepsilon),$$

$$0 \le g(I_1, P_{k1}) \le \delta_1 + \varepsilon,$$

$$0 \le g(P_{k1}, w_i) \le s_k(\delta_1 + \varepsilon)$$

и при условиях (4.6), (4.8), (4.9) для $j = \overline{2,p}$.

Заключение. Исследована задача распределения ресурсов и построения допустимого расписания для комплекса работ в многопроцессорной системе реального времени. Разработан алгоритм нахождения параметров системы (производительности процессоров, объемы и эффективность использования ресурсов), при которых решение в поставленной задаче существует. Решена задача оптимальной коррекции характеристик заданий (директивные интервалы и объемы работ). Для решения указанных задач применяется сетевое моделирование и алгоритмы нахождения потоков с заданными свойствами.

СПИСОК ЛИТЕРАТУРЫ

1. *Танаев В.С., Гордон В.С., Шафранский Я.М.* Теория расписаний. Одностадийные системы. М.: Наука, 1984.
2. *Brucker P.* Scheduling Algorithms. Heidelberg: Springer, 2007.
3. *Танаев В.С., Сотсков Ю.Н., Струсевич В.А.* Теория расписаний. Многостадийные системы. М.: Наука, 1989.
4. *Горский М.А., Мищенко А.В., Нестерович Л.Г., Халиков М.А.* Некоторые модификации целочисленных оптимизационных задач с учетом неопределенности и риска // Изв. РАН. ТиСУ. 2022. № 5. С. 106–117.
5. *Мищенко А.В., Кошелев П.С.* Оптимизация управления работами логистического проекта в условиях неопределенности // Изв. РАН. ТиСУ. 2021. № 4. С. 123–134.
6. *Лазарев А.А.* Теория расписаний. Оценка абсолютной погрешности и схема приближенного решения задач теории расписаний. М.: МФТИ, 2008.
7. *Лазарев А.А.* Теория расписаний. Методы и алгоритмы. М.: ИПУ РАН, 2019.
8. *Глонина А.Б., Балашов В.В.* О корректности моделирования модульных вычислительных систем реального времени с помощью сетей временных автоматов // Моделирование и анализ информационных систем. 2018. Т. 25. № 2. С. 174–192.
9. *Глонина А.Б.* Обобщенная модель функционирования модульных вычислительных систем реального времени для проверки допустимости конфигураций таких систем // Вестн. ЮУрГУ. Сер. Вычисл. математика и информатика. 2017. Т. 6. № 4. С. 43–59.
10. *Глонина А.Б.* Инструментальная система проверки выполнения ограничений реального времени для конфигураций модульных вычислительных систем // Вестн. МГУ. Сер. 15. Вычисл. математика и кибернетика. 2020. № 3. С. 16–29.
11. *Филлипс Д., Гарсиа-Диас А.* Методы анализа сетей. М.: Мир, 1984.
12. *Давыдов Э.Г.* Исследование операций. М.: Высш. шк., 1990.
13. *Фуругян М.Г.* Распределение неоднородного набора ресурсов при составлении многопроцессорного расписания // Изв. РАН. ТиСУ. 2021. № 5. С. 120–127.
14 *Фуругян М.Г.* Синтез многопроцессорной системы при построении расписаний с прерываниями и директивными интервалами // Изв. РАН. ТиСУ. 2019. № 2. С. 41–46.
15. *Майника Э.* Алгоритмы оптимизации на сетях и графах. М.: Мир, 1981.