
СИСТЕМНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

УДК 519.86

ОПТИМИЗАЦИЯ СТРУКТУРЫ МНОГОПРОЦЕССОРНОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

© 2022 г. М. Г. Фуругян

Федеральный исследовательский центр “Информатика и управление” РАН, Москва, Россия

e-mail: rtsccas@yandex.ru

Поступила в редакцию 29.11.2021 г.

После доработки 13.12.2021 г.

Принята к публикации 31.01.2022 г.

Рассматривается задача оптимизации структуры многопроцессорной вычислительной системы реального времени, где под структурой понимается набор связей между процессорами, который может изменяться во времени при определенных условиях. Эти условия, во-первых, связаны с физической возможностью или невозможностью установления связи между каждой парой процессоров в определенные моменты времени и, во-вторых, должны гарантировать существование допустимого расписания для заданного множества заданий с известными характеристиками (директивные интервалы, объемы работ, привязка к определенным процессорам и требуемые объемы памяти). Установление подобных связей дает возможность переключать выполнение работ с одного процессора на другой и требует некоторых затрат. Ставится задача определения структуры с минимальными затратами. Она сведена к многопродуктовой потоковой задаче в сети специального вида, для которой получены необходимые и достаточные условия существования решения в виде системы линейных булевых соотношений для исходных параметров.

DOI: 10.31857/S0002338822030064

Введение. В настоящее время вычислительные системы реального времени востребованы во многих отраслях производства. Они применяются в гражданском и военном строительстве, при разработке газовых и нефтяных месторождений, при испытаниях самолетов, ракет и других технических объектов, в конвейерных производствах, транспортных системах, при разработке систем противовоздушной и противоракетной обороны, обработке больших массивов информации технического, экономического и экологического характера в реальном масштабе времени. Так, при проведении летных испытаний данные, поступающие с борта самолета с высокой частотой, должны успевать обрабатываться в темпе поступления с тем, чтобы передавать на борт корректирующие команды. С использованием вычислительных сетей, многопроцессорной и многоядерной техники возникли новые математические и прикладные задачи, как, например, оптимизация структуры вычислительной системы реального времени. В [1–3] такая задача для многоядерной вычислительной системы реального времени решается с использованием обобщенных конечных автоматов и построенной на их основе имитационной модели. С помощью этой модели строится временная диаграмма, описывающая функционирование системы и позволяющая осуществить непосредственную проверку того, что каждая работа выполняется в заданном директивном интервале. Публикации [4–6] посвящены исследованию изменений работоспособности многопользовательской информационной сети в результате повреждений. Данный анализ проводится с использованием теории потоков в сетях. Для оценки ущерба строится двухкритериальная модель, с помощью которой определяется влияние различных повреждений на функционирование информационных направлений. Проводится анализ устойчивости характеристик многопродуктовой потоковой сети к различным разрушениям. В [7] рассматривается аппаратно-программный комплекс вычислительной системы реального времени и решается задача минимизации затрат при его проектировании с заданными требованиями на производительность и надежность функционирования. Разработана методика сведения указанной задачи к задаче однокритериальной условной оптимизации, для решения которой используются эволюционные алгоритмы. В ряде случаев задачи сетевой оптимизации и составления графиков выполнения работ сводятся к поиску минимакса [8–11]. В [12] определяются производительности

процессоров, при которых многопроцессорная система реального времени способна выполнять требуемые вычисления в заданном темпе. Исследованы постановки с фиксированными и нефиксированными параметрами, с возобновляемыми и невозобновляемыми ресурсами. Показано, что исходная задача сводится к нахождению допустимых решений в системе линейных неравенств. Разработан алгоритм нахождения парето-оптимальных решений. В [13] рассмотрена задача оптимизации выполнения комплекса работ с нефиксированными длительностями, для решения которой применяется метод ветвей и границ.

В статье приводится задача оптимизации структуры многопроцессорной вычислительной системы реального времени. Устанавливаются допустимые и минимальные по стоимости связи между процессорами, которые при определенных условиях и ограничениях в разные моменты времени можно переопределять. Эти ограничения указывают физическую возможность (или невозможность) установления связи между каждой парой процессоров в фиксированный момент времени и, кроме того, требуют существования допустимого расписания для множества заданий с известными характеристиками (директивные интервалы, объемы требуемой работы, привязка к определенным процессорам и необходимые объемы памяти). Установление подобных связей дает возможность в определенные моменты времени переключать выполнение заданий с одного процессора на другой, но требует определенных затрат. Необходимо найти структуру связей между процессорами с минимальными затратами. Эта задача сводится к сетевой многопродуктовой потоковой задаче и для ее решения получены необходимые и достаточные условия существования решения в виде системы линейных булевых соотношений. В отличие от [14] в настоящей работе помимо поиска допустимого расписания решается задача построения допустимой структуры вычислительной системы и минимизации ее стоимости, а также рассматриваются дополнительные ограничения на выполнение заданий (ограничения по памяти, привязке к процессорам и др.).

1. Постановка задачи. Имеется m процессоров, занумерованных от 1 до m , из которых требуется спроектировать вычислительную систему реального времени, и комплекс работ (заданий) $N = \{1, \overline{n}\}$, занумерованных от 1 до n , который необходимо выполнить с помощью этой системы. Горизонтом планирования является отрезок $[0, T]$, который разделен на T временных отрезков единичной длины $\Delta_k = [t_{k-1}, t_k]$ $k = \overline{1, T}$ $t_0 = 0, t_T = T$. Каждая работа не может одновременно выполняться несколькими процессорами, а один процессор не может одновременно исполнять несколько работ. Производительность j -го процессора (или произведенный в единицу времени объем работы) составляет p_j , а объем его оперативной памяти, доступной в интервале Δ_k , равен B_{jk} . Каждое задание $i \in N$ характеризуется директивным временным интервалом $[b_i, f_i]$ (внутри которого допускается его реализация; $b_i, f_i \in \{t_0, t_1, \dots, t_T\}$), объемом требуемой работы процессоров w_i и необходимым объемом оперативной памяти c_i . Внутри каждого интервала $\Delta_k \subseteq [b_i, f_i]$, $k = \overline{1, T}$, не допускаются прерывания и переключения задания $i \in N$ с одного процессора на другой. Если в этом случае исполнителем является процессор j , то объем сделанной им работы составляет p_j .

Имеются следующие ограничения, связанные с исполнением работ.

Ограничения на использование процессоров: в интервале Δ_k задание i может выполняться только одним из процессоров из заданного множества $M(i, k) \subseteq \{1, \overline{m}\}$.

Ограничения на переключения между процессорами: задан четырехмерный массив I , каждый элемент $I(j_1, k_1, j_2, k_2)$ которого равен 1, если можно установить связь между процессорами j_1 и j_2 , позволяющую после выполнения задания в интервале Δ_{k_1} процессором j_1 переключить его на процессор j_2 в интервале Δ_{k_2} , и равен 0, если указанная связь невозможна, $j_1 \in M(i, k_1), j_2 \in M(i, k_2), i \in N, k_1, k_2 = \overline{1, T}, k_1 < k_2$. Суммарный объем работы процессоров j_1 и j_2 , необходимый для такого переключения, составляет $q(j_1, k_1, j_2, k_2)$.

Ограничения по памяти: объем памяти, требуемой для выполнения работы, не должен превосходить доступного в соответствующем интервале объема оперативной памяти процессора, исполняющего ее.

Массив I определяет возможность или невозможность установления связи между процессорами в различные моменты времени. Так, если $I(j_1, k_1, j_2, k_2) = 1$, то при проектировании вычислительной системы возможно установление связи, обеспечивающей переключение задания с процессора j_1 в момент времени t_{k_1} на процессор j_2 в момент времени t_{k_2-1} . При этом стоимость

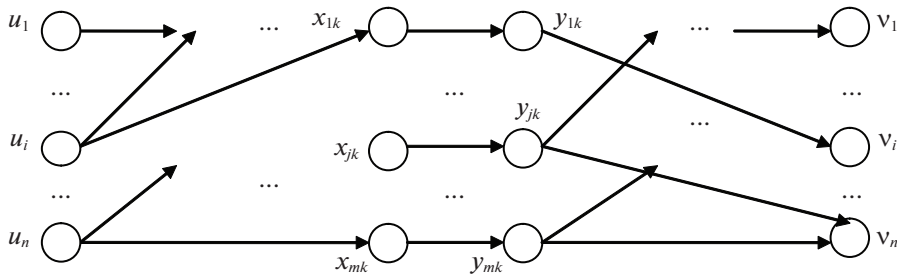


Рис. 1. Сеть $G_1, k = \overline{1, K}$

установления такой связи составляет $s(j_1, k_1, j_2, k_2)$. Допустимая структура Σ вычислительной системы задается установлением указанных связей между процессорами, не противоречащих масшиву I . Иными словами, $\Sigma = \{(j_1, k_1, j_2, k_2) : I(j_1, k_1, j_2, k_2) = 1\}$. Таким образом, связь между процессорами может быть непостоянной во времени, что имеет место, например, в транспьютерных системах. Стоимость $S(\Sigma)$ допустимой структуры Σ – это суммарная стоимость по каждой установленной связи, т.е.

$$S(\Sigma) = \sum_{(j_1, k_1, j_2, k_2) \in \Sigma} s(j_1, k_1, j_2, k_2).$$

Допустимым расписанием $R(\Sigma)$ для комплекса N при выбранной допустимой структуре Σ является такое расписание, при котором каждое задание $i \in N$ начинает выполняться не ранее момента времени b_i и завершается не позднее момента времени f_i , суммарный объем работы процессоров при его исполнении равен w_i . Все переключения с одного процессора на другой осуществляются в соответствии со структурой Σ . Кроме того, выполнены ограничения, накладываемые на объем памяти процессоров и их привязке к заданиям. Допустимым решением задачи будем называть пару $(\Sigma, R(\Sigma))$, где Σ – допустимая структура вычислительной системы, $R(\Sigma)$ – допустимое расписание при структуре Σ , а стоимостью этого решения будет стоимость структуры Σ .

Требуется определить, существует ли в поставленной задаче допустимое решение $(\Sigma, R(\Sigma))$, и в случае положительного ответа найти допустимое решение минимальной стоимости.

Данная задача является NP-трудной. Действительно, используя схему доказательства, аналогичную той, которая применена в [14], можно показать, что известная NP-трудная задача о разбиении [15] полиномиально сводится к рассматриваемой задаче.

2. Решение задачи с помощью сетевого моделирования. Будем рассматривать n -продуктовую потоковую сеть $G_1 = (V_1, A_1)$, где $V_1 = \{u_i, v_i, x_{jk}, y_{jk}, i = \overline{1, n}, j = \overline{1, m}, k = \overline{1, T}\}$ – множество узлов (u_i и v_i – источник и сток i -го продукта соответственно), $A_1 = \{(u_i, x_{jk}), i = \overline{1, n}, k \in [b_i, f_i], j \in M(i, k); (y_{jk}, v_i), i = \overline{1, n}, k \in [b_i, f_i], j \in M(i, k); (x_{jk}, y_{jk}), i = \overline{1, n}, j = \overline{1, m}, k = \overline{1, T}; (y_{j_1 k_1}, y_{j_2 k_2}), j_1, j_2 = \overline{1, m}, k_1, k_2 = \overline{1, T}, k_1 < k_2, I(j_1, k_1, j_2, k_2) = 1\}$ – множество дуг. Узлы x_{jk} и y_{jk} обозначают процессор j и соответственно начало временного отрезка Δ_k и его окончание. Фрагмент сети G_1 изображен на рис. 1.

В сети G_1 будем искать n -продуктовый поток $g_i(a, b), i = \overline{1, n}, (a, b) \in A_1$, принимающий значения 0 или 1, такой, что ровно одна единица потока i -го продукта доставляется из источника u_i в сток v_i . Смысловое значение единицы потока g_i по дугам сети G_1 указано в таблице.

Если по дуге (x_{jk}, y_{jk}) проходит единица потока g_i , то это означает, что в интервале Δ_k задание i выполняется процессором j , объем работы которого составляет p_j , и эту величину следует прибавить к суммарному объему работы процессоров по исполнению задания i . Если единица потока g_i проходит по дуге $(y_{j_1 k_1}, x_{j_2 k_2})$, то это означает, что, во-первых, данная дуга должна быть включена в структуру вычислительной системы и, во-вторых, из суммарного объема работы процессоров по выполнению задания i следует вычесть величину $q(j_1, k_1, j_2, k_2)$. Таким образом, каждому n -продуктовому потоку $g_i(a, b)$ в сети G_1 соответствует некоторая допустимая структура Σ вычислительной системы и некоторое расписание $R(\Sigma)$ выполнения множества заданий N . Будем искать такой n -продуктовый поток, для которого стоимость структуры Σ минимальна, а соответствующее расписание $R(\Sigma)$ является допустимым.

Таблица 1. Смысловое значение единицы потока g_i по дугам сети G_1

Дуга $(a, b) \in A_1$	Смысловое значение единицы потока $g_i(a, b)$
(u_i, x_{jk})	Начало выполнения работы i процессором j в момент времени t_{k-1}
(x_{jk}, y_{jk})	Выполнение работы i процессором j в интервале Δ_k
$(y_{j_1k_1}, x_{j_2k_2})$	Переключение работы i с процессора j_1 в момент времени t_{k_1} на процессор j_2 в момент времени t_{k_2-1}
(y_{jk}, v_i)	Завершение выполнения работы i процессором j в момент времени t_k

3. Необходимые и достаточные условия существования допустимого решения. Задачу построения допустимой структуры минимальной стоимости и поиска допустимого расписания сведем к задаче булева линейного программирования. Запишем условия, накладываемые на поток $g_i(a, b)$ и сформулированные в разд. 2, в виде минимизации линейного функционала и линейных ограничений с булевыми переменными $g_i(a, b)$, $i = \overline{1, n}$, $(a, b) \in A_1$:

$$\min_{g_i} \sum_{i=1}^n \sum_{\substack{j_1 \in M(i, k_1), j_2 \in M(i, k_2), \\ I(j_1, k_1, j_2, k_2)=1}} g_i(y_{j_1k_1}, x_{j_2k_2}) s(y_{j_1k_1}, x_{j_2k_2}), \quad (3.1)$$

$$\sum_{k=b_i}^{f_i} \sum_{j \in M(i, k)} g_i(u_i, x_{jk}) = 1, \quad i = \overline{1, n}, \quad (3.2)$$

$$\sum_{k=b_i}^{f_i} \sum_{j \in M(i, k)} g_i(y_{jk}, v_i) = 1, \quad i = \overline{1, n}, \quad (3.3)$$

$$\sum_{i=1}^n g_i(x_{jk}, y_{jk}) \leq 1, \quad j = \overline{1, m}, \quad k = \overline{1, T}, \quad (3.4)$$

$$g_i(u_i, x_{j_1k_1}) + \sum_{\substack{j_2, k_2: I(j_2, k_2, j_1, k_1)=1, \\ j_2 \in M(i, k_2), k_2 \in [b_i, f_i]}} g_i(y_{j_2k_2}, x_{j_1k_1}) = g_i(x_{j_1k_1}, y_{j_1k_1}), \quad (3.5)$$

$$i = \overline{1, n}, \quad k_1 \in [b_i, f_i], \quad j_1 \in M(i, k_1),$$

$$g_i(x_{j_1k_1}, y_{j_1k_1}) = g_i(y_{j_1k_1}, v_i) + \sum_{\substack{j_2, k_2: I(j_1, k_1, j_2, k_2)=1, \\ j_2 \in M(i, k_2), k_2 \in [b_i, f_i]}} g_i(y_{j_2k_2}, x_{j_1k_1}), \quad (3.6)$$

$$i = \overline{1, n}, \quad k_1 \in [b_i, f_i], \quad j_1 \in M(i, k_1),$$

$$\sum_{k=b_i}^{f_i} \sum_{j \in M(i, k)} g_i(x_{jk}, y_{jk}) p_j - \sum_{\substack{k_1, k_2 \in [b_i, f_i], j_1 \in M(i, k_1), \\ j_2 \in M(i, k_2), I(j_1, k_1, j_2, k_2)=1}} g_i(y_{j_1k_1}, x_{j_2k_2}) q(j_1, k_1, j_2, k_2) \geq w_i, \quad i = \overline{1, n}, \quad (3.7)$$

$$\sum_{i=1}^n g_i(x_{jk}, y_{jk}) c_i \leq B_{jk}, \quad j = \overline{1, m}, \quad k = \overline{1, T}, \quad (3.8)$$

$$g_i(a, b) \in \{0, 1\}, \quad i = \overline{1, n}, \quad (a, b) \in A. \quad (3.9)$$

Данная система содержит $O(n(mT)^2)$ булевых переменных и $O(nmT)$ линейных ограничений.

Поясним физический смысл выражений (3.1)–(3.9). Равенства (3.2), (3.3) означают, что суммарная величина i -го продукта, исходящего из источника u_i и соответственно входящего в сток v_i , $i = \overline{1, n}$, равна единице. В силу (3.4) суммарная величина потока по каждой дуге (x_{jk}, y_{jk}) , $j = \overline{1, m}$,

$k = \overline{1, T}$, не превосходит единицы. Равенства (3.5), (3.6) гарантируют сохранение потока i -го продукта во внутренних узлах x_{jk} и y_{jk} соответственно, $i = \overline{1, n}$, $j = \overline{1, m}$, $k = \overline{1, T}$. Согласно неравенства (3.7), суммарный объем работы процессоров при выполнении задания i не меньше w_i , $i = \overline{1, n}$. Величина, стоящая в левой части соотношения (3.8), равна объему памяти, требуемой для работы, выполняемой j -м процессором в интервале Δ_k , $j = \overline{1, m}$, $k = \overline{1, T}$. Это значение не должно превосходить доступного объема оперативной памяти j -го процессора в интервале Δ_k , т.е. величины B_{jk} . Согласно (3.9), в сети G_1 будут найдены такие n непересекающихся путей, что величина потока i -го продукта по i -му пути равна единице. Минимизация выражения (3.1) обеспечивает построение структуры вычислительной системы минимальной стоимости.

Л е м м а 1. Для существования допустимого решения $(\Sigma, R(\Sigma))$ минимальной стоимости в исходной задаче необходимо и достаточно наличия решения в задаче булева линейного программирования (3.1)–(3.9).

Д о к а з а т е л ь с т в о 1. Необходимость. Если в исходной задаче существует допустимое решение $(\Sigma, R(\Sigma))$, то при структуре Σ для каждого $i = \overline{1, n}$ расписание исполнения i -й работы описывает путь из u_i в v_i в сети G_1 , по которому можно пропустить единицу потока i -го продукта, причем все эти n путей не пересекаются. Все работы полностью завершаются в своих директивных интервалах и, кроме того, не нарушаются ограничения по требуемой памяти. Это обеспечивает выполнение условий (3.2)–(3.9). Поскольку структура Σ имеет минимальную стоимость, то верно условие (3.1).

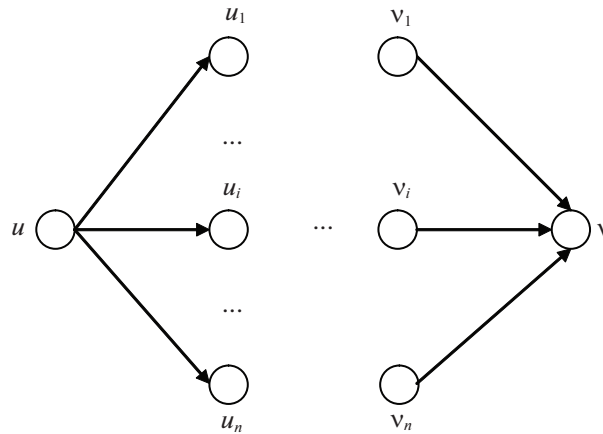
2. Достаточность. В силу соотношений (3.2)–(3.6), (3.9) в сети G_1 существуют n непересекающихся путей из u_i в v_i , $i = \overline{1, n}$, таких, что вдоль i -го из них протекает единица i -го продукта, благодаря чему существует некоторое расписание для комплекса заданий N , согласно которому каждая работа выполняется в своем директивном интервале. Согласно (3.7), каждое задание исполняется полностью, а из (3.8) следует, что не нарушаются ограничения по памяти. Таким образом, это расписание является допустимым. Дуги (y_{jk}, x_{jk}) , по которым поток равен единице, образуют искомую структуру Σ . Поскольку верно условие (3.1), то стоимость этой структуры минимальна. Лемма доказана.

При выполнении условий (3.1)–(3.9) решение $(\Sigma, R(\Sigma))$ исходной задачи строится следующим образом. Если $g_i(y_{j_1 k_1}, x_{j_2 k_2}) = 1$ при некотором $i = \overline{1, n}$, то в проектируемую структуру Σ системы необходимо включить связь, обеспечивающую возможность переключения задания i с процессора j_1 в момент времени t_{k_1} на процессор j_2 в момент времени t_{k_2-1} . Равенство $g_i(x_{j_2 k_2}, y_{j_1 k_1}) = 1$ означает, что, согласно допустимому расписанию $R(\Sigma)$, в интервале Δ_k работа i выполняется процессором j .

4. Упрощенный вид необходимых условий существования допустимого решения. В этом разделе предполагается, что память, требуемая для выполнения каждого задания, не превосходит объема памяти любого процессора. В таком случае неравенства (3.8) верны и их можно опустить. Для данного частного случая сформулируем упрощенный вид необходимых условий существования допустимого решения. Для этого построим однопродуктовую потоковую сеть $G_2 = (V_2, A_2)$ следующим образом. Добавим к сети G_1 два узла (источник u и сток v) и дуги (u, u_i) и (v_i, v) , $i = \overline{1, n}$. Таким образом, $V_2 = V_1 \cup \{u, v\}$, $A_2 = A_1 \cup \{(u, u_i), (v_i, v), i = \overline{1, n}\}$. Фрагмент сети G_2 изображен на рис. 2. Дуги $(a, b) \in A_2$ сети G_2 имеют три параметра: нижнюю и верхнюю границы потока (соответственно $L(a, b)$ и $U(a, b)$), а также стоимость $C(a, b)$ единицы потока. Значения этих параметров определим следующим образом: $L(a, b) = 0$, $U(a, b) = 1$ при всех $(a, b) \in A_2$; $C(x_{jk}, y_{jk}) = p_j$, $C(y_{j_1 k_1}, x_{j_2 k_2}) = -q(j_1, k_1, j_2, k_2)$, $C(a, b) = 0$ для всех остальных дуг $(a, b) \in A_2$. Пусть $h(a, b)$, $(a, b) \in A_2$ – некоторый однопродуктовый поток в сети G_2 , а величина

$$C(h) = \sum_{(a,b) \in A_2} C(a, b)$$

– его стоимость.

Рис. 2. Сеть G_2

Л е м м а 2. Для существования допустимого решения $(\Sigma, R(\Sigma))$ в рассматриваемом частном случае исходной задачи необходимо наличие однопродуктового потока h в сети G_2 , для которого справедливо неравенство

$$C(h) \leq \sum_{i=1}^n w_i. \quad (4.1)$$

Д о к а з а т е л ь с т в о. Если существует допустимое решение $(\Sigma, R(\Sigma))$, то в силу леммы 1 для некоторого n -продуктового потока g в сети G_1 верны соотношения (3.2)–(3.7), (3.9). Определим однопродуктовый поток h в сети G_2 следующим образом:

$$h(u, u_i) = \sum_{k \in [b_i, f_i]} \sum_{j \in M(i, k)} g_i(u_i, x_{jk}), \quad i = \overline{1, n},$$

$$h(v_i, v) = \sum_{k \in [b_i, f_i]} \sum_{j \in M(i, k)} g_i(y_{jk}, v_i), \quad i = \overline{1, n},$$

$$h(x_{jk}, y_{jk}) = \sum_{i=1}^n g_i(x_{jk}, y_{jk}), \quad j = \overline{1, m}, \quad k = \overline{1, T},$$

$$h(y_{j_1 k_1}, x_{j_2 k_2}) = \sum_{i=1}^n g_i(y_{j_1 k_1}, x_{j_2 k_2}), \quad j_1, j_2, k_1, k_2 : I(j_1, j_2, k_1, k_2) = 1.$$

Из (3.2)–(3.6) и (3.9) следует, что h является потоком в сети G_2 , т.е. удовлетворяет ограничениям сверху и снизу по каждой дуге, а также условию сохранения в каждом внутреннем узле сети G_2 . Кроме того, из (3.7) следует (4.1). Лемма доказана.

Для проверки выполнения условия леммы 2 можно воспользоваться известным алгоритмом нахождения потока минимальной стоимости в сети следующим образом [15]. Заменим знаки определенных выше стоимостей потоков по дугам на противоположные. В сети G_2 найдем однопродуктовый поток \bar{h} минимальной стоимости и пусть $C(\bar{h})$ – его стоимость. Тогда если

$$C(\bar{h}) \leq -\sum_{i=1}^n w_i,$$

то условие леммы 2 и необходимые условия существования допустимого решения верны. В противном случае условие леммы 2 и необходимые условия существования допустимого решения не выполнены.

Поскольку поиск однопродуктового потока минимальной стоимости в сети G_2 существенно проще решения задачи булева линейного программирования (3.1)–(3.9), то решение исходной

задачи можно проводить по следующей схеме. Сначала проверить выполнение условия леммы 2. И только в случае положительного результата перейти к проверке условий леммы 1.

Заключение. Решена задача оптимизации структуры многопроцессорной вычислительной системы реального времени и построения допустимого расписания комплекса работ для найденной структуры. Предполагается, что заданы ограничения на связи между процессорами, на их привязку к заданиям, на объемы оперативной памяти, а также на временные ограничения, определяемые директивными интервалами. Получены необходимые и достаточные условия существования допустимой структуры минимальной стоимости и допустимого расписания работ в виде системы булевых линейных соотношений. Методика решения основана на сведении исходной задачи к многопродуктовой потоковой задаче в сети.

СПИСОК ЛИТЕРАТУРЫ

1. 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.
2. Глонина А.Б. Обобщенная модель функционирования модульных вычислительных систем реального времени для проверки допустимости конфигураций таких систем // Вестник ЮУрГУ. Сер. Вычисл. математика и информатика. 2017. Т. 6. № 4. С. 43–59.
3. 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.
4. Малашенко Ю.Е., Назарова И.А., Новикова Н.М. Один подход к анализу возможных структурных повреждений в многопродуктовых сетевых системах // ЖВМ и МФ. 2019. Т. 59. № 9. С. 1626–1638.
5. Малашенко Ю.Е., Назарова И.А., Новикова Н.М. Анализ кластерных повреждений в сетевых системах // Изв. РАН. ТиСУ. 2020. Т. 60. № 2. С. 338–348.
6. Малашенко Ю.Е., Новикова Н.М. Анализ многопользовательских сетевых систем с учетом неопределенности // Изв. РАН. ТиСУ. 1998. № 2. С. 134–146.
7. Ефимов С.Н., Терсков В.А., Ярков К.В., Серикова О.Ю. Оптимизация затрат при проектировании аппаратно-программного комплекса системы реального времени // Научно-технический вестник Поволжья. 2021. № 4. С. 35–37.
8. Mironov A.A., Tsurkov V.I. Network Models with Fixed Parameters at the Communication Nodes. 1 // J. Computer and Systems Sciences International. 1994. V. 32. № 6. P. 1–11.
9. Mironov A.A., Tsurkov V.I. Network Models with Fixed Parameters at the Communication Nodes. 2 // J. Computer and Systems Sciences International. 1995. V. 33. № 3. P. 107–116.
10. Mironov A.A., Levkina T.A., Tsurkov V.I. Minimax Estimations of Arc Weights in Integer Networks with Fixed Node Degrees // Applied and Computational Mathematics. 2009. V. 8. № 2. P. 216–226.
11. Mironov A.A., Tsurkov V.I. Class of Distribution Problems with Minimax Criterion // Doklady Akademii Nauk. 1994. V. 336. № 1. P. 35–38.
12. Фуругян М.Г. Некоторые алгоритмы анализа и синтеза многопроцессорных вычислительных систем реального времени // Программирование. 2014. № 1. С. 36–44.
13. Мищенко А.В., Кошелев П.С. Оптимизация управления работами логистического проекта в условиях неопределенности // Изв. РАН. ТиСУ. 2021. № 4. С. 86–101.
14. Фуругян М.Г. Планирование вычислений в многопроцессорных АСУ реального времени с дополнительным ресурсом // АИТ. 2015. № 3. С. 144–150.
15. КORTE Б., Фиген Й. Комбинаторная оптимизация. Теория и алгоритмы. М.: МЦНМО, 2015.