
**СИСТЕМНЫЙ АНАЛИЗ
И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ**

УДК 519.86

**ПЛАНИРОВАНИЕ ВЫЧИСЛЕНИЙ
В МНОГОПРОЦЕССОРНОЙ СИСТЕМЕ
С НЕОПРЕДЕЛЕННЫМИ МОМЕНТАМИ ГОТОВНОСТИ РАБОТ**

© 2020 г. М. Г. Фуругян

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

e-mail: rtscas@yandex.ru

Поступила в редакцию 10.02.2020 г.

После доработки 26.02.2020 г.

Принята к публикации 30.03.2020 г.

Рассматривается задача составления допустимого многопроцессорного расписания без прерываний и переключений для случая, когда на множестве работ задано отношение частично-го порядка, все работы имеют общий директивный срок и, кроме того, задается распределение заданий по процессорам. В некоторые неопределенные моменты времени могут поступать запросы на выполнение дополнительных более приоритетных работ, для которых на известное время освобождаются некоторые процессоры. В результате этого выполнение исходной совокупности заданий переносится на более позднее время и тем самым нарушается построенное для нее расписание. Разработана такая стратегия построения допустимого расписания, при которой вероятность его нарушения вследствие поступления запросов на дополнительные работы минимальна.

DOI: 10.31857/S0002338820040046

Введение. При разработке математического и программного обеспечения для многопроцессорных систем, в частности систем реального времени, одной из основных задач является задача планирования вычислений и построения допустимых расписаний для программных модулей. При испытаниях и эксплуатации сложных технических объектов нередко возникает нештатная ситуация, когда помимо выполнения основных программных модулей в некоторые неопределенные моменты времени могут поступать запросы на выполнение дополнительных, но более приоритетных заданий. Если такие запросы поступают в то время, когда выполняются основные работы и при этом требуется освободить занимаемые ими процессоры, то это ведет к нарушению построенного ранее расписания для основных работ. В этом случае задача заключается в минимизации вероятности подобного рода нарушений. Указанные ситуации возникают при испытаниях самолетов, космических систем, ядерных реакторов и другой сложной техники. При этом возникают задачи составления допустимого расписания при наличии неопределенных факторов, которые сводятся к игровым задачам. Особенно актуальны такие задачи при проектировании бортовых автоматизированных систем управления.

Задачи распределения ресурсов и построения расписаний при наличии неопределенных факторов рассматривались ранее рядом авторов. В [1] описывалась минимаксная задача теории расписаний при нефиксированных длительностях работ, для решения которой был разработан метод построения областей устойчивости оптимальных расписаний. В [2] дополнительно предполагается возможность динамического изменения структуры сети, задающей частичный порядок работ. Задача минимизации времени выполнения сетевого комплекса работ в случае, когда длительности работ являются функциями от вектора распределения ресурсов, а число процессоров не ограничено, исследовалась в [3] и была сведена к задаче нелинейного программирования. В [4] рассматривалась задача минимизации числа процессоров заданной производительности при фиксированных интервалах выполнения работ и известных издержках на переключения процессоров между заданиями. В некоторых случаях задачи распределения ресурсов с нефиксированными параметрами сводятся к сетевым задачам, методы решения которых описаны в [5–7]. В [8] описывалась задача распределения невозобновляемых ресурсов при наличии неопределенных факторов и для ее решения была использована методика построения многогранников

устойчивости, разработанная в [1, 2]. Приведенная в [8] задача является *NP*-трудной, и предложенный алгоритм ее решения имеет переборный характер. В настоящей статье предполагается, что каждая работа закреплена за определенным процессором, а на множестве работ задано отношение частичного порядка их выполнения. Благодаря этому множество допустимых расписаний описывается с помощью полиномиального алгоритма. С использованием этого множества строится антагонистическая игра, решение которой определяет решение исходной задачи.

1. Постановка задачи. Рассматривается вычислительная система, состоящая из m процессоров, и совокупность работ (заданий) $W = \{w_{ij} : i = \overline{1, k_j}, j = \overline{1, m}\}$, подлежащих выполнению, k_j – заданная величина, равная числу работ, приписанных процессору j , $j = \overline{1, m}$. Работа w_{ij} приписана процессору j , т.е. выполняется этим процессором. Таким образом, множество работ W разбито на m групп, каждая из которых исполняется определенным процессором. При этом не допускаются прерывания и переключения с одного процессора на другой. Задан общий директивный интервал $[0, T]$ для всех заданий, т.е. завершить каждую работу необходимо не позднее момента времени T . Длительность работы w_{ij} составляет $t(w_{ij}) = t_{ij}$, $t_{ij} \leq T$. На множестве W задан ориентированный граф без циклов $G = (W, A)$, определяющий частичный порядок выполнения работ, где W – множество узлов, A – множество ориентированных дуг. Если $(w, \bar{w}) \in A$, то работа $\bar{w}$ может быть начата только после завершения работы w . В этом случае будем использовать обозначение $w \rightarrow \bar{w}$. Работа w является непосредственным предшественником работы $\bar{w}$, а $\bar{w}$ – непосредственным последователем работы w . Предполагается, что задания, приписанные одному и тому же процессору j , упорядочены следующим образом:

$$w_{1j} \rightarrow w_{2j} \rightarrow \dots \rightarrow w_{k_j j}. \quad (1.1)$$

Кроме того, каждая работа может иметь непосредственных предшественников, приписанных другим процессорам.

В некоторые неопределенные моменты времени $y_{hj} \in [0, T]$, $h = \overline{1, p_j}$, $j = \overline{1, m}$, p_j – заданные величины, $0 \leq y_{1j} < y_{2j} < \dots < y_{p_j j} \leq T$, могут поступать запросы на выполнение дополнительных, более приоритетных, работ $V = \{v_{hj} : h = \overline{1, p_j}, j = \overline{1, m}\}$. Каждая работа v_{hj} приписана процессору j , не допускает прерываний и переключений на другие процессоры, а ее длительность составляет s_{hj} , $s_{hj} < T$. Таким образом, работа v_{hj} выполняется в интервале $[y_{hj}, y_{hj} + s_{hj}]$. (Предполагается, что $y_{hj} + s_{hj} \leq y_{h+1, j}$). Если в этом интервале исполняется некоторая работа w_{ij} , то она прекращается и переносится на более позднее время. Тем самым нарушается построенное ранее расписание работ W . Процесс выполнения множества заданий W и поступления запросов на дополнительные работы V повторяется многократно. Задача заключается в выработке стратегии построения таких расписаний работ W , которые удовлетворяют следующим условиям.

1. Каждое задание w_{ij} выполняется процессором j .
2. Не нарушаются ограничения частичного порядка выполнения работ W , задаваемые графом G .
3. Если в интервале $[0, T]$ не поступило ни одного запроса на выполнение дополнительных работ V , то все работы W завершаются не позднее момента времени T .
4. Вероятность того, что расписание заданий W будет нарушено вследствие поступления запросов на выполнение дополнительных работ V , минимальна.

Расписание для работ W , при котором не нарушаются условия 1–3, называется допустимым. Таким образом, задача заключается в выработке стратегии построения допустимого расписания для заданий W , при которой выполняется условие 4.

2. Некоторые определения и обозначения. Введем следующие обозначения:

$P(w)$ – множество непосредственных предшественников работы $w \in W$, т.е. $P(w) = \{w' \in W : w' \rightarrow w\}$;

$\bar{P}(w)$ – множество всех заданий из W , завершение которых должно предшествовать началу выполнения работы w , т.е. $\bar{P}(w)$ – это множество работ $\tilde{w} \in W$, для которых в графе G существует ориентированный путь, ведущий из $\tilde{w}$ в w ;

$Q(w)$ – множество непосредственных последователей работы w , т.е. $Q(w) = \{w' \in W : w \rightarrow w'\}$;

$\bar{Q}(w)$ — множество всех заданий из W , началу выполнения которых должно предшествовать завершение работы w , т.е. $\bar{Q}(w)$ — это множество работ $\tilde{w} \in W$, для которых в графе G существует ориентированный путь, ведущий из w в $\tilde{w}$.

Отметим, что $P(w) \subseteq \bar{P}(w)$ и $Q(w) \subseteq \bar{Q}(w)$. Кроме того, если $w' \in P(w)$, $\tilde{w} \in \bar{P}(w')$ и $\tilde{w} \in P(w)$, то дугу $(\tilde{w}, w)$ можно удалить из графа G , так как это не приведет к изменению частичного порядка исполнения работ W . Будем предполагать, что такая процедура коррекции графа G уже проведена.

Для работы $w_{ij} \in W$ символами $\tau(w_{ij}) = \tau_{ij}$ и $\theta(w_{ij}) = \theta_{ij}$ будем обозначать соответственно наиболее ранний возможный и наиболее поздний допустимый сроки начала выполнения. Иными словами, работа w не может быть начата раньше момента времени $\tau(w)$ (так как не будут завершены все ее предшественники) и позднее момента времени $\theta(w)$ (так как в этом случае некоторые из ее последователей не успеют завершиться к директивному сроку T).

Далее, поскольку граф G не содержит ориентированных циклов, то вершины W можно разбить на уровни следующим образом. В нулевой уровень W_0 включим все вершины $w \in W$, которые не имеют непосредственных предшественников, т.е.

$$W_0 = \{w \in W : P(w) = \emptyset\}. \quad (2.1)$$

В первый уровень W_1 войдут вершины $w \in W$, все непосредственные предшественники которых содержатся в W_0 , т.е.

$$W_1 = \{w \in W : P(w) \subseteq W_0\}. \quad (2.2)$$

В общем случае, в уровень W_l , $l = \overline{1, L}$, входят вершины $w \in W \setminus \{W_0 \cup W_1 \cup \dots \cup W_{l-1}\}$, все непосредственные предшественники которых содержатся в $W_0 \cup W_1 \cup \dots \cup W_{l-1}$, т.е.

$$W_l = \left\{ w \in W \setminus \bigcup_{r=0}^{l-1} W_r : P(w) \subseteq \bigcup_{r=0}^{l-1} W_r \right\}, \quad l = \overline{1, L}. \quad (2.3)$$

В дальнейшем будем также использовать другое разбиение множества вершин W графа G на уровни. А именно в нулевой уровень $\bar{W}_0$ включим все вершины $w \in W$, которые не имеют непосредственных последователей, т.е.

$$\bar{W}_0 = \{w \in W : Q(w) = \emptyset\}. \quad (2.4)$$

В первый уровень $\bar{W}_1$ войдут вершины $w \in W$, все непосредственные последователи которых содержатся в $\bar{W}_0$, т.е.

$$\bar{W}_1 = \{w \in W : Q(w) \subseteq \bar{W}_0\}. \quad (2.5)$$

В общем случае, в уровень $\bar{W}_l$, $l = \overline{1, L}$, входят вершины $w \in W \setminus \{\bar{W}_0 \cup \bar{W}_1 \cup \dots \cup \bar{W}_{l-1}\}$, все непосредственные последователи которых содержатся в $\bar{W}_0 \cup \bar{W}_1 \cup \dots \cup \bar{W}_{l-1}$, т.е.

$$\bar{W}_l = \left\{ w \in W \setminus \bigcup_{r=0}^{l-1} \bar{W}_r : Q(w) \subseteq \bigcup_{r=0}^{l-1} \bar{W}_r \right\}, \quad l = \overline{1, L}. \quad (2.6)$$

Отметим, что разбиение множества W на уровни W_l и $\bar{W}_l$, $l = \overline{0, L}$, возможно, поскольку граф G не содержит ориентированных циклов.

Эти разбиения множества W будем использовать для вычисления величин $\tau(w)$ и $\theta(w)$, $w \in W$. Величины $\tau(w)$, $w \in W$, вычисляются последовательно сначала для W_0 , затем для W_1 и т.д. следующим образом. Если $w \in W_0$, то

$$\tau(w) = 0. \quad (2.7)$$

Если $w \in W_l$, $l = \overline{1, L}$, то

$$\tau(w) = \max_{w_{ij} \in P(w)} \{\tau_{ij} + t_{ij}\}. \quad (2.8)$$

Величины $\theta(w)$, $w \in W$, также вычисляются последовательно сначала для $\bar{W}_0$, затем для $\bar{W}_1$ и т.д. следующим образом. Если $w \in \bar{W}_0$, то

$$\theta(w) = T - t(w). \quad (2.9)$$

Если $w \in \bar{W}_l$, $l = \overline{1, L}$, то

$$\theta(w) = \min_{w_{ij} \in Q(w)} \{\theta_{ij} - t(w)\}. \quad (2.10)$$

Л е м м а 1. Допустимое расписание существует в том и только том случае, когда

$$\tau_{k_j j} \leq \theta_{k_j j} \quad (2.11)$$

при всех $j = \overline{1, m}$.

Д о к а з а т е л ь с т в о. 1. Пусть верно неравенство (2.11). Покажем сначала, что существует расписание работ W , при котором каждое задание $w \in W$ начинает выполняться в момент $\tau(w)$. Действительно, все работы из W_0 не имеют предшественников, и в силу (1.1) назначены на различные процессоры. Следовательно, каждое задание $w \in W_0$ может быть начато в момент $\tau(w) = 0$. Пусть далее это утверждение доказано для всех работ из W_l , $1 \leq l < L$. Тогда в силу (2.8) для каждой работы $w_{ij} \in W_{l+1}$ к моменту времени τ_{ij} завершены все ее предшественники, а в силу (1.1) в момент времени τ_{ij} процессором j может выполняться только работа w_{ij} . Таким образом, существует расписание, при котором момент начала каждой работы w_{ij} равен τ_{ij} . А в силу (2.11) это расписание является допустимым.

2. Предположим теперь, что существует допустимое расписание. Как было показано выше, существует расписание, при котором каждое задание w_{ij} начинает выполняться в момент τ_{ij} . Следовательно, это расписание также является допустимым, откуда следует (2.11). Лемма доказана.

3. Построение антагонистической игры. Пусть x_{ij} — момент начала работы w_{ij} , причем справедливости неравенства (2.11) леммы 1 и условия 1–3 из разд. 1. Пусть далее $X = \{x_{ij}; i = \overline{1, k_j}, j = \overline{1, m}\}$. Множество X описывает все возможные моменты начала выполнения работ W , при которых существует допустимое расписание, и строится следующим образом. Для $w_{ij} \in W_0$

$$x_{ij} \in [\tau_{ij}; \theta_{ij}]. \quad (3.1)$$

Для $w_{ij} \in W_l$, $l = \overline{1, L}$,

$$x_{ij} \in \left[\max_{w_{pq} \in P(w_{ij})} (x_{pq} + t_{pq}); \theta_{ij} \right]. \quad (3.2)$$

Каждый элемент $x \in X$ определяет допустимое расписание S_x , при котором момент начала работы w_{ij} равен x_{ij} (здесь $x = \{x_{ij}\}$). Пусть $Y = \{y_{hj}; h = \overline{1, p_j}, j = \overline{1, m} : 0 \leq y_{1j} < y_{2j} < \dots < y_{p_j j} \leq T\}$. Множество Y описывает все возможные моменты поступления запросов на выполнение дополнительных работ V .

На множестве $X \times Y$ будем рассматривать антагонистическую игру с платежной функцией $K(x, y)$, $x \in X$, $y \in Y$, определенной следующим образом:

$$K(x, y) = \begin{cases} 1, & \text{если } x_{ij} + t_{ij} < y_{hj} \text{ или } x_{ij} \geq y_{hj} + s_{hj} \text{ при всех} \\ & j = \overline{1, m}, \quad i = \overline{1, k_j}, \quad h = \overline{1, p_j}; \\ 0 & \text{в противном случае.} \end{cases} \quad (3.3)$$

Иными словами, если для каждого $j = \overline{1, m}$ выполнение ни одной из дополнительных работ v_{hj} не препятствует исполнению ни одного задания w_{ij} , т.е. если интервалы $[x_{ij}; x_{ij} + t_{ij}]$ и $[y_{hj}; y_{hj} + s_{hj}]$ не пересекаются, то $K(x, y) = 1$. Если же для некоторого j , $1 \leq j \leq m$, выполнение хотя бы одной дополнительной работы v_{hj} препятствует исполнению хотя бы одного задания w_{ij} , т.е. если хотя бы два интервала $[x_{ij}; x_{ij} + t_{ij}]$ и $[y_{hj}; y_{hj} + s_{hj}]$ пересекаются, то $K(x, y) = 0$.

Если $K(x, y) = 1$ то работы V могут быть выполнены так, что при этом расписание S_x для заданий W не будет нарушено. Если же $K(x, y) = 0$, то работы W не могут быть выполнены по распи-

санию S_x . Пусть $v(x)$ – некоторая смешанная стратегия первого игрока в игре с платежной функцией $K(x, y)$, $x \in X$, $y \in Y$. Тогда при фиксированном $y \in Y$ величина

$$\int_X K(x, y) dv(x)$$

равна вероятности того, что расписание заданий W не будет нарушено вследствие поступления запросов на выполнение дополнительных работ V . Оптимальная смешанная стратегия $v^*(x)$ первого игрока максимизирует величину

$$\inf_{y \in Y} \int_X K(x, y) dv(x)$$

и определяет искомую стратегию построения расписаний S_x . Метод решения антагонистической игры (3.3), основанный на аппроксимации ее конечными играми, описан в [9].

4. Частный случай рассматриваемой задачи. В [9] описан метод решений антагонистических игр с полунепрерывной платежной функцией, имеющей разрыв на конечном числе поверхностей, разбивающих множество чистых стратегий игроков на области, в которых функция выигрыша является равномерно непрерывной. Однако игра с платежной функцией (3.3) представляет собой частный случай таких игр, для которого будет описан более эффективный метод решения. Так же, как и в [9], метод основан на дискретной аппроксимации исходной игры конечной игрой.

Будем рассматривать случай, когда имеется один процессор ($m = 1$), одна основная работа w ($k_1 = 1$) и поступает один запрос на выполнение дополнительной работы v ($p_1 = 1$). Далее, пусть t и s – длительности основной и дополнительной работ соответственно, $[0; T]$ – директивный интервал, $t < T$, $s < T$. Пусть x – момент начала работы w , а y – момент поступления запроса на выполнение работы v . Тогда, согласно (3.3), платежная функция $K(x, y)$ определяется следующим образом:

$$K(x, y) = \begin{cases} 1, & \text{если } y > x + t \text{ или } y \leq x - s, \\ 0, & \text{если } x - s < y \leq x + t, \end{cases} \quad (4.1)$$

где $x \in X$, $y \in Y$, $X = [0; T]$, $Y = [0; T]$.

Таким образом, платежная функция $K(x, y)$ может принимать только два значения 0 и 1, и имеет разрыв по прямым $y = x + t$ и $y = x - s$.

Аппроксимируем игру (4.1) конечной игрой с конечными множествами чистых стратегий $\bar{X} \subseteq X$, $\bar{Y} \subseteq Y$. Для этого построим числовые последовательности $\{x_i^l\}$ и $\{y_j^l\}$, $l = \overline{1, 4}$, $x_i^l \in (0; T)$, $y_j^l \in (0; T)$, элементы которой вычисляются с помощью следующих рекуррентных соотношений:

$$\begin{aligned} x_1^1 &= s, & x_{i+1}^1 &= x_i^1 + (t + s), & i &= \overline{1, u_1}, \\ y_1^1 &= t + s, & y_{j+1}^1 &= y_j^1 + (t + s), & j &= \overline{1, v_1}, \\ x_1^2 &= T - t, & x_{i+1}^2 &= x_i^2 - (t + s), & i &= \overline{1, u_2}, \\ y_1^2 &= T - (t + s), & y_{j+1}^2 &= y_j^2 - (t + s), & j &= \overline{1, v_2}, \\ x_1^3 &= t + s, & x_{i+1}^3 &= x_i^3 + (t + s), & i &= \overline{1, u_3}, \\ y_1^3 &= t, & y_{j+1}^3 &= y_j^3 + (t + s), & j &= \overline{1, v_3}, \\ x_1^4 &= T - (t + s), & x_{i+1}^4 &= x_i^4 - (t + s), & i &= \overline{1, u_4}, \\ y_1^4 &= T - s, & y_{j+1}^4 &= y_j^4 - (t + s), & j &= \overline{1, v_4}. \end{aligned}$$

Здесь $u_l, v_l, l = \overline{1, 4}$, – величины, не превосходящие $[T/(t + s)]$. Пары чисел $(x_i^l; y_j^l)$, $l = \overline{1, 4}$, являются координатами некоторых точек квадрата $[0, T] \times [0, T]$, лежащих на прямых $y = x + t$ и $y = x - s$ (см. рис. 1). Например, пары (x_i^1, y_j^1) соответствуют точкам с координатами $(s, 0)$, $(s, t + s)$, $(t + 2s, t + s)$, $(t + 2s, 2t + 2s)$, $(2t + 3s, 2t + 2s)$ и т.д., лежащим на указанных выше прямых.

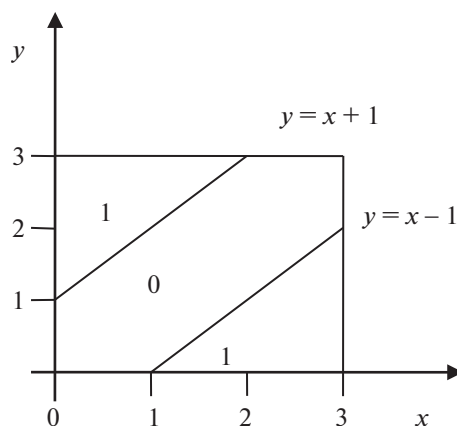


Рис. 1. Платежная функция $K(x, y)$ и множества $\bar{X}$ и $\bar{Y}$ при $T = 3, t = 1, s = 1$

Построим следующие конечные подмножества $\bar{X}$ и $\bar{Y}$ множеств X и Y соответственно: $\bar{X} = \{0, T, x_i^l, l = \overline{1, 4}, i = 1, 2, \dots\}$, $\bar{Y} = \{0, T, y_i^l, l = \overline{1, 4}, i = 1, 2, \dots\}$. Перенумеруем элементы множеств $\bar{X}$ и $\bar{Y}$ так, что $X = \{0 = x_1 < x_2 < \dots < x_{r-1} < x_r = T\}$, $\bar{Y} = \{0 = y_1 < y_2 < \dots < y_{v-1} < y_v = T\}$. На рис. 1 указаны множества $\bar{X}$ и $\bar{Y}$ для случая, когда $T = 3, t = 1, s = 1$. При этом платежная функция $K(x, y)$ имеет разрывы по прямым $y = x + 1$ и $y = x - 1$; $\bar{X} = \{x_1 = 0, x_2 = 1, x_3 = 2, x_4 = 3\}$, $\bar{Y} = \{y_1 = 0, y_2 = 1, y_3 = 2, y_4 = 3\}$, $K(x, y) = 1$ при $y > x + 1$ и $y \leq x - 1$, $K(x, y) = 0$ при $x - 1 < y \leq x + 1$.

Л е м м а 2. 1. Для любых $x_i, x_{i+1} \in \bar{X}$, $y_j, y_{j+1} \in \bar{Y}$ и любого $x \in (x_i, x_{i+1})$ справедливы равенства $K(x, y_j) = K(x_i, y_j)$, $K(x, y_{j+1}) = K(x_i, y_{j+1})$.

2. Для любых $x_i, x_{i+1} \in \bar{X}$, $y_j, y_{j+1} \in \bar{Y}$ и любого $y \in (y_j, y_{j+1})$ верны равенства $K(x_i, y) = K(x_i, y_{j+1})$, $K(x_{i+1}, y) = K(x_{i+1}, y_{j+1})$.

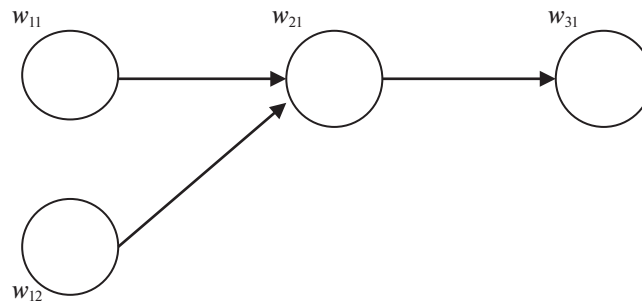
Д о к а з а т е л ь с т в о. Из построения множеств $\bar{X}$ и $\bar{Y}$ и определения функции $K(x, y)$ следует, что при всех $x \in [x_i, x_{i+1})$ функция $K(x, y_j)$ принимает одно и то же значение (0 или 1). То же самое относится к функции $K(x, y_{j+1})$, а также и к функциям $K(x_i, y)$ и $K(x_{i+1}, y)$ при $y \in (y_j, y_{j+1}]$. Лемма доказана.

Пусть $(K(x, y), \bar{X}, \bar{Y})$ – конечная (матричная) игра с платежной функцией $K(x, y)$ на множествах чистых стратегий $\bar{X}$ и $\bar{Y}$ первого и второго игроков соответственно. Пусть $\tilde{V}$ – значение этой игры, а $p = (p_1, p_2, \dots, p_r)$ и $q = (q_1, q_2, \dots, q_r)$ – оптимальные смешанные стратегии первого и второго игроков соответственно, т.е. при всех $j = \overline{1, v}$, $i = \overline{1, r}$ выполнены следующие неравенства:

$$\sum_{i=1}^r K(x_i, y_j) p_i \geq \tilde{V}, \quad \sum_{j=1}^v K(x_i, y_j) q_j \leq \tilde{V}. \quad (4.2)$$

Определим вероятностную меру $\nu(x)$, заданную на X и сосредоточенную в точках x_i , в которых она имеет скачки p_i , $i = \overline{1, r}$, и вероятностную меру $\mu(y)$, определенную на Y и сосредоточенную в точках y_j , в которых она имеет скачки q_j , $j = \overline{1, v}$. Тогда из (4.2) следует, что при всех $y \in \bar{Y}$, $x \in \bar{X}$ справедливы неравенства

$$\int_X K(x, y) d\nu(x) \geq \tilde{V}, \quad \int_Y K(x, y) d\mu(y) \leq \tilde{V}.$$

Рис. 2. Граф частичного порядка G

Пусть $y \in Y$ и $y \in (y_j, y_{j+1})$. В силу леммы 2 $K(x_i, y) = K(x_i, y_{j+1})$ при всех $i = \overline{1, r}$. Следовательно,

$$\int_X K(x, y) d\nu(x) = \sum_{i=1}^r K(x_i, y) p_i = \sum_{i=1}^r K(x_i, y_{j+1}) p_i \geq \tilde{V}.$$

Аналогично если $x \in X$ и $x \in (x_i, x_{i+1})$, то $K(x, y_j) = K(x_{i+1}, y_j)$ при всех $j = \overline{1, v}$. Поэтому

$$\int_Y K(x, y) d\mu(y) = \sum_{j=1}^v K(x_i, y_j) q_j = \sum_{j=1}^v K(x_{i+1}, y_j) q_j \leq \tilde{V}.$$

Следовательно, $(\nu(x), \mu(y), \tilde{V})$ – решение игры $(K(x, y), X, Y)$. Таким образом, решение исходной игры сведено к решению конечной игры $(K(x, y), \bar{X}, \bar{Y})$.

5. Модельный пример. Рассмотрим пример, когда число процессоров равно $m = 2$, множество заданий $W = \{w_{11}, w_{21}, w_{31}, w_{12}\}$ (работы w_{11}, w_{21}, w_{31} выполняются первым процессором, а работа w_{12} – вторым). Длительности работ и директивный срок следующие: $t_{11} = 1, t_{21} = 1, t_{31} = 2, t_{12} = 1, T = 6$. Дополнительная работа одна – v_{12} ; выполняется она вторым процессором и ее длительность составляет $s_{12} = 1$. Запрос на выполнение работы v_{12} может поступить в неопределенный момент времени $y_{12} \in [0; 6]$. Граф частичного порядка на множестве W задается следующими отношениями предшествования: $w_{11} \rightarrow w_{21} \rightarrow w_{31}$ и $w_{12} \rightarrow w_{21}$ (рис. 2).

Разбиения множества W на уровни W_l и $\bar{W}_l$, $l = \overline{0, L}$, построенные по формулам (2.1)–(2.6), следующие: $W_0 = \{w_{11}, w_{12}\}$, $W_1 = \{w_{21}\}$, $W_2 = \{w_{31}\}$, $\bar{W}_0 = \{w_{31}\}$, $\bar{W}_1 = \{w_{21}\}$, $\bar{W}_2 = \{w_{11}, w_{12}\}$. Наиболее ранние возможные и наиболее поздние допустимые сроки начала выполнения заданий W , вычисленные по формулам (2.7)–(2.10), соответственно равны $\tau_{11} = \tau_{12} = 0$, $\tau_{21} = 1$, $\tau_{31} = 2$, $\theta_{31} = 4$, $\theta_{21} = 3$, $\theta_{12} = \theta_{11} = 2$. С помощью формул (3.1), (3.2) построим множество X :

$$X = \{x_{11} \in [0, 2], x_{12} \in [0, 2], x_{21} \in [\max(x_{11} + 1, x_{12} + 1); 3], x_{31} \in [x_{21} + 1; 4]\}.$$

Коллизия при запросе на дополнительную работу v_{12} может возникнуть только при выполнении задания w_{12} , так как v_{12} и w_{12} должны исполняться на одном и том же процессоре (втором). Платежная функция (4.1) антагонистической игры выглядит следующим образом:

$$K(x_{12}, y_{12}) = \begin{cases} 1, & \text{если } x_{12} + 1 < y_{12} \text{ или } x_{12} \geq y_{12} + 1; \\ 0, & \text{если } x_{12} - 1 < y_{12} \leq x_{12} + 1, \end{cases} \quad (5.1)$$

$x_{12} \in X = [0, 2]$, $y_{12} \in Y = [0, 6]$. В данном случае множество $X \times Y$ является не квадратом, как рассматривалось в разд. 4, а прямоугольником. Однако все рассуждения, проведенные в разд. 4, справедливы и в этом случае. Множества $\bar{X}$ и $\bar{Y}$ выглядят следующим образом: $\bar{X} = \{0, 1, 2\}$, $\bar{Y} = \{0, 1, 2, 3, 6\}$, $r = 3$, $v = 5$.

На рис. 3 приведено схематическое изображение платежной функции $K(x_{12}, y_{12})$. Пусть $\nu_a(x_{12})$, $\mu_a(y_{12})$ – вероятностные меры на $[0; 2]$ и $[0; 6]$ соответственно, сосредоточенные в одной

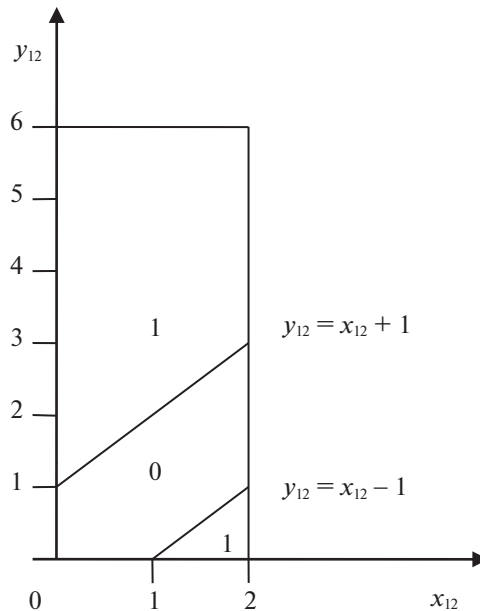


Рис. 3. Платежная функция антагонистической игры в модельном примере

точке a , в которой они имеют скачок, равный единице. Пусть далее $v^*(x_{12}) = 0.5v_0(x_{12}) + 0.5v_2(x_{12})$, $\mu^*(x_{12}) = 0.5\mu_0(x_{12}) + 0.5\mu_2(x_{12})$, т.е. вероятностные меры $v^*(x_{12})$ и $\mu^*(y_{12})$ сосредоточены в двух точках 0 и 2, в которых они имеют скачки, равные 0.5. Тогда справедливы следующие соотношения:

$$\int_{[0;2]} K(x_{12}, y_{12}) dv^*(x_{12}) \geq 0.5$$

при всех $y_{12} \in [0; 6]$,

$$\int_{[0;6]} K(x_{12}, y_{12}) d\mu^*(y_{12}) \leq 0.5$$

при всех $x_{12} \in [0; 2]$.

Следовательно, значение игры (5.1) равно 0.5, а $v^*(x_{12})$ и $\mu^*(y_{12})$ – оптимальные смешанные стратегии первого и второго игроков соответственно. Таким образом, с вероятностью 0.5 работу w_{12} следует начинать в моменты времени 0 и 2, т.е. с вероятностью 0.5 следует применять расписания S_0 и S_2 , где S_0 соответствует вектору $x_0 \in X$, $x_0 = \{x_{11} \in [0; 2], x_{12} = 0; x_{21} \in [x_{11} + 1; 3], x_{31} \in [x_{21} + 1; 4]\}$, а S_2 – вектору $x_2 \in X$, $x_2 = \{x_{11} \in [0; 2], x_{12} = 2; x_{21} = 3, x_{31} = 4\}$.

Заключение. Исследована задача составления допустимого многопроцессорного расписания выполнения непрерываемых работ с заданным отношением частичного порядка, фиксированным распределением работ по процессорам и общим директивным сроком у них. В некоторые неопределенные моменты времени ресурсы вычислительной системы могут быть переданы допустимым, более приоритетным работам, в результате чего выполнение некоторых заданий откладывается и тем самым нарушается построенное ранее расписание. Разработана такая стратегия построения допустимого расписания, при которой вероятность его нарушения вследствие поступления запросов на выполнение дополнительных работ минимальна. Решение этой задачи сведено к решению антагонистической игры, для которой разработан метод дискретной аппроксимации. Рассмотрен частный случай для однопроцессорной системы с одной основной и одной дополнительной работами. Полученные результаты продемонстрированы на модельном примере.

СПИСОК ЛИТЕРАТУРЫ

1. *Мищенко А.В., Сушков Б.Г.* Минимизация времени выполнения работ, представленных сетевой моделью, при нефиксированных параметрах сети. М.: ВЦ АН СССР, 1980. С. 3–16.
2. *Мищенко А.В.* Устойчивость решений задачи оптимального распределения ресурсов при динамическом изменении структуры сети. М.: ВЦ АН СССР, 1980. 12 с.
3. *Давыдов Э.Г.* Исследование операций. М.: Высш. шк., 1990. С. 185–189.
4. *Филлипс Д., Гарсиа-Дуас А.* Методы анализа сетей. М.: Мир, 1984.
5. *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.
6. *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.
7. *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.
8. *Фуругян М.Г.* Решение одной задачи распределения ресурсов в АСУ реального времени при наличии неопределенных факторов // АиТ. 2002. № 11. С. 167–171.
9. *Фуругян М.Г.* Приближенное решение одного класса бесконечных антагонистических игр с полунепрерывной платежной функцией // Вестн. МГУ. Сер. 15. № 2. 1980. С. 66–69.