at_pstu_ru-rts-2006-01

М.В. Кавалеров, Н.Н. Матушкин
Пермский государственный технический университет

ВЫЧИСЛЕНИЕ ОЦЕНОК ПАРАМЕТРОВ ВЫПОЛНЕНИЯ ЗАПРОСОВ В УСЛОВИЯХ ПЛАНИРОВАНИЯ С ФИКСИРОВАННЫМИ ПРИОРИТЕТАМИ

Ранее был предложен новый подход к планированию задач с нестандартными ограничениями реального времени в условиях планирования с фиксированными приоритетами [1,2]. Для практической реализации этого подхода необходимо уметь вычислять оценки параметров выполнения запросов, формируемых задачами. Предложены алгоритмы, позволяющие вычислить эти оценки. Приводится доказательство того, что эти оценки всегда вычисляются правильно на основе предложенного способа.

Систему реального времени можно представить как множество взаимодействующих задач реального времени (РВ). Каждой такой задаче соответствует ограничение РВ, которое накладывается на характеристики процесса выполнения *запросов*, формируемых этой задачей. При этом необходимо осуществлять планирование множества задач. В данном случае планирование – это распределение вычислительных ресурсов между запросами различных задач, направленное на соблюдение ограничений РВ.

В работах [1,2] был представлен новый подход к планированию задач с *нестандартными ограничениями* (НО) РВ в условиях планирования с фиксированными приоритетами (ПФП). Этот подход позволяет существенно повысить эффективность использования вычислительных ресурсов по сравнению с обычной практикой преобразования НО в стандартные ограничения.

Необходимым условием реализации предложенного подхода является наличие оценок параметров выполнения запросов в условиях ПФП. Поэтому следует более подробно рассмотреть вопрос вычисления этих оценок на основе известной информации о задачах РВ.

Имеется множество из n задач ЖРВ $\tau = \{\tau_1, \tau_2, ..., \tau_n\}$. Задача τ_i общем случае формирует множество запросов $\{\tau_{i,j}\}$, $j \geq 1$. Каждый запрос $\tau_{i,j}$ при своем появлении помещается в очередь запросов, и $r_{i,j}$, $s_{i,j}$, $f_{i,j}$ – это соответственно *время появления* $\tau_{i,j}$, *время начала выполнения* $\tau_{i,j}$, *время завершения выполнения* $\tau_{i,j}$. Предполагается, что вычислительная система – однопроцессорная, и прерывания задач разрешены, при этом задачи являются независимыми.

Шаг времени при планировании – это такое вещественное число ε, что все события, определяемые планированием, могут возникать только в моменты времени $k\varepsilon \geq 0$, где k – целое число, и каждый из указанных моментов времени может быть моментом возникновения какого-нибудь события, определяемого планированием. В дальнейшем для краткости будет использоваться выражение вида «значение кратно ε», которое считается эквивалентным выражению «значение равно $k\varepsilon$, где k – целое число».

Пусть $E_{i,j}$ – *длительность* $\tau_{i,j}$, то есть это значение $f_{i,j} - s_{i,j}$ при отсутствии прерывания $\tau_{i,j}$. В общем случае значение $E_{i,j}$ становится известным после момента $f_{i,j}$. Естественно, что в любом случае $E_{i,j}$ строго больше нуля и является кратным ε, то есть $E_{i,j} \geq \varepsilon$.

Обработка запросов при планировании в РВ осуществляется согласно принципу ПФП. Предполагается, что задачи имеют различные приоритеты, задаваемые числом из диапазона $[1,n]$. Пусть $\mathbf{I}_{hp}(i)$ обозначает *множество индексов задач ЖРВ с приоритетами выше, чем у задачи* τ_i.

Ограничение РВ может быть связано не только с моментами $s_{i,j}$ и $f_{i,j}$, но и с другими моментами времени на интервале $[s_{i,j}, f_{i,j}]$, которые соответствуют наблюдаемым событиям [3]. Согласно допущению, указанному в работе [1], предполагается, что с ограничением РВ может быть связано два момента времени на интервале $[s_{i,j}, f_{i,j}]$. Тогда *значимые моменты времени запроса* $\tau_{i,j}$ – это моменты времени $x_{i,j}$ и $y_{i,j}$, принадлежащие интервалу $[s_{i,j}, f_{i,j}]$ и соответствующие событиям, на которые накладывается ограничение РВ для задачи τ_i.

Например, $x_{i,j}$ может соответствовать моменту получения информации о параметре объекта управления, а $y_{i,j}$ может соответствовать моменту формирования управляющего воздействия.

Получается, что с запросом $\tau_{i,j}$ связаны четыре момента времени: $s_{i,j}$, $x_{i,j}$, $y_{i,j}$, $f_{i,j}$. Тогда существует всего шесть вариантов различных компонентов запроса с соответствующими длительностями выполнения.

Длительности компонентов $\tau_{i,j}$ – это значения $Esx_{i,j}$, $Esy_{i,j}$, $Esf_{i,j}$, $Exy_{i,j}$, $Exf_{i,j}$, $Eyf_{i,j}$, которые соответственно равны $x_{i,j} - s_{i,j}$, $y_{i,j} - s_{i,j}$, $f_{i,j} - s_{i,j}$, $y_{i,j} - x_{i,j}$, $f_{i,j} - x_{i,j}$, $f_{i,j} - y_{i,j}$ при отсутствии прерывания $\tau_{i,j}$.

Пусть **sxyf** – это *множество обозначений* $\{sx, sy, sf, xy, xf, yf\}$, то есть множество всех возможных пар различных символов из совокупности $\{s, x, y, f\}$. Тогда, например, $\{Esx_{i,j}, Esy_{i,j}, Esf_{i,j}, Exy_{i,j}, Exf_{i,j}, Eyf_{i,j}\}$ можно обозначить как $\{E\alpha\beta_{i,j} \mid \alpha\beta \in \mathbf{sxyf}\}$.

Во многих случаях бывает сложно заранее определить значения $\{E\alpha\beta_{i,j} \mid \alpha\beta \in \mathbf{sxyf}\}$, так как количество шагов алгоритма для каждого компонента запроса может сложным образом зависеть от входных данных и внутренних переменных. Поэтому заранее можно определить только нижние и верхние оценки длительностей компонентов запроса.

Нижние и верхние оценки длительностей для задачи τ_i обозначаются соответственно $E\alpha\beta_i^{Lo}$, $E\alpha\beta_i^{Up}$ при $\forall\alpha\beta \in \mathbf{sxyf}$ и удовлетворяют условию $E\alpha\beta_i^{Lo} \le E\alpha\beta_{i,j} \le E\alpha\beta_i^{Up}$ при $\forall\alpha\beta \in \mathbf{sxyf}$ и $\forall j \ge 1$. В частности, очевидно, $Esf_i^{Lo} \le E_{i,j} \le Esf_i^{Up}$, в любом случае.

Существуют методы и средства для вычисления оценок длительностей выполнения программных блоков. Поэтому в дальнейшем предполагается, что значения $E\alpha\beta_i^{Lo}$, $E\alpha\beta_i^{Up}$ при $\forall\alpha\beta \in \mathbf{sxyf}$ уже вычислены.

При выполнении запроса $\tau_{i,j}$ в условиях ПФП можно выделить пять моментов времени ($r_{i,j}$, $s_{i,j}$, $x_{i,j}$, $y_{i,j}$, $f_{i,j}$). Тогда получается, что существует всего десять различных пар, образуемых из этих моментов времени. Каждая такая пара будет считаться параметром выполнения запроса.

Параметры выполнения запроса $\tau_{i,j}$ – это значения $rs_{i,j}$, $rx_{i,j}$, $ry_{i,j}$, $rf_{i,j}$, $sx_{i,j}$, $sy_{i,j}$, $sf_{i,j}$, $xy_{i,j}$, $xf_{i,j}$, $yf_{i,j}$, которые соответственно равны $s_{i,j} - r_{i,j}$, $x_{i,j} - r_{i,j}$, $y_{i,j} - r_{i,j}$, $f_{i,j} - r_{i,j}$, $x_{i,j} - s_{i,j}$, $y_{i,j} - s_{i,j}$, $f_{i,j} - s_{i,j}$, $y_{i,j} - x_{i,j}$, $f_{i,j} - x_{i,j}$, $f_{i,j} - y_{i,j}$.

По аналогии с **sxyf** пусть **rsxyf** – это *множество обозначений* $\{rs, rx, ry, rf, sx, sy, sf, xy, xf, yf\}$, то есть это множество всех возможных пар различных символов из совокупности $\{r, s, x, y, f\}$. Тогда, например, множество всех параметров выполнения запроса можно обозначить как $\{\alpha\beta_{i,j} \mid \alpha\beta \in \mathbf{rsxyf}\}$.

Выполнение запроса $\tau_{i,j}$ может прерываться запросами с более высокими приоритетами. Поэтому, очевидно, $E\alpha\beta_{i,j} \le \alpha\beta_{i,j}$ для $\forall\alpha\beta \in \mathbf{sxyf}$, а также $Esx_{i,j} \le rx_{i,j}$, $Esy_{i,j} \le ry_{i,j}$, $Esf_{i,j} \le rf_{i,j}$.

В общем случае точные значения $\{\alpha\beta_{i,j} \mid \alpha\beta \in \mathbf{rsxyf}\}$ заранее неизвестны. Поэтому их можно только оценивать сверху и снизу с той или иной точностью.

Нижние и верхние оценки параметров выполнения запросов задачи τ_i обозначаются соответственно $\alpha\beta_i^{Lo}$, $\alpha\beta_i^{Up}$ при $\forall\alpha\beta \in \mathbf{rsxyf}$ и удовлетворяют условию $\alpha\beta_i^{Lo} \le \alpha\beta_{i,j} \le \alpha\beta_i^{Up}$ при $\forall\alpha\beta \in \mathbf{rsxyf}$ и $\forall j \ge 1$.

Именно оценки $\alpha\beta_i^{Lo}$, $\alpha\beta_i^{Up}$ при $\forall\alpha\beta \in \mathbf{rsxyf}$ следует уметь вычислять для реализации предложенного подхода к планированию задач с НО в условиях ПФП. Необходимо указать способ вычисления этих оценок.

Загрузка вычислительного устройства запросами задач ЖРВ с индексами из множества $\mathbf{I}$, обозначаемая $U_\tau(\mathbf{I})$, – это процент времени, в течение которого происходит выполнение указанных запросов на интервале $[0,\infty)$. Но значение $U_\tau(\mathbf{I})$ в общем случае заранее невозможно определить, так как, в частности, точные значения длительностей запросов могут быть заранее неизвестны. Но в любом случае заранее можно получить нижнюю и верхнюю оценки $U_\tau(\mathbf{I})$.

Нижняя и верхняя оценки $U_\tau(\mathbf{I})$ обозначаются соответственно $U_\tau^{Lo}(\mathbf{I})$, $U_\tau^{Up}(\mathbf{I})$ и удовлетворяют условию $U_\tau^{Lo}(\mathbf{I}) \le U_\tau(\mathbf{I}) \le U_\tau^{Up}(\mathbf{I})$.

Нижняя и верхняя оценки периода следования запросов задачи τ_i обозначаются соответственно T_i^{Lo}, T_i^{Up} и удовлетворяют условию $T_i^{Lo} \le r_{i,j} - r_{i,j-1} \le T_i^{Up}$ при $\forall j > 1$.

Очевидно, что с учетом T_i^{Lo}, T_i^{Up} значения $U_\tau^{Lo}(\mathbf{I})$, $U_\tau^{Up}(\mathbf{I})$ можно вычислить следующим образом

$$U_\tau^{Lo}(\mathbf{I}) = \sum_{\forall i \in \mathbf{I}} \frac{Esf_i^{Lo}}{T_i^{Up}}, \quad U_\tau^{Up}(\mathbf{I}) = \sum_{\forall i \in \mathbf{I}} \frac{Esf_i^{Up}}{T_i^{Lo}}. \tag{1}$$

Пусть σ – некоторый запрос, E^σ – длительность σ, а также $r^\sigma, s^\sigma, f^\sigma$ – это соответственно время появления σ, время начала выполнения σ, время завершения выполнения σ.

Пусть $\sigma rf_i^{Lo}(\alpha,\beta), \sigma rf_i^{Up}(\alpha,\beta)$ при некотором вещественном α и некотором $\beta \in \{0,1\}$ – это соответственно нижняя и верхняя оценки значения $f^\sigma - r^\sigma$; при $\alpha \le 0$ эти оценки считаются равными α, а при $\alpha > 0$ вычисляются при условиях, что: 1) запрос σ имеет приоритет π_i; 2) $E^\sigma = \alpha$; 3) если $\beta = 1$, то в момент r^σ в основной очереди могут быть запросы, сформированные задачей τ_i, и задача τ_i является периодической или спорадической; если $\beta = 0$, то в момент r^σ в основной очереди обязательно отсутствуют запросы, сформированные задачей τ_i.

Оценки параметров выполнения запросов могут быть вычислены на основе оценок $\sigma rf_i^{Lo}(\alpha,\beta), \sigma rf_i^{Up}(\alpha,\beta)$. Но для этого необходимо уметь вычислять $\sigma rf_i^{Lo}(\alpha,\beta), \sigma rf_i^{Up}(\alpha,\beta)$, поэтому предлагаются следующие алгоритмы, которые построены на основе незначительных изменений и дополнений алгоритмов, представленных в работах [4,5].

<u>Алгоритм 1.</u> Вычисление $\sigma rf_i^{Up}(\alpha,\beta)$.

Входные данные алгоритма – это значения i, α, β, а также значения $\{Esf_v^{Up}, T_v^{Lo} \mid v \in (\mathbf{I}_{hp}(i) \cup \{i\})\}$. Результат выполнения алгоритма – это значение $\sigma rf_i^{Up}(\alpha,\beta)$. Последовательность шагов алгоритма определяется следующим образом.

1. Если $\alpha \le 0$, то алгоритм завершается с результатом $\sigma rf_i^{Up}(\alpha,\beta) = \alpha$.
2. Если ($\beta = 0$ и $U_\tau^{Up}(\mathbf{I}_{hp}(i)) \ge 100\%$) или ($\beta = 1$ и $U_\tau^{Up}(\mathbf{I}_{hp}(i) \cup \{i\}) > 100\%$), то алгоритм завершается с результатом $\sigma rf_i^{Up}(\alpha,\beta) = \infty$.
3. Выполняется установка значений: $w = \alpha$, $\sigma rf = \alpha$, $w^q = Esf_i^{Up}$, $q = 0$, где w, σrf, w^q, q – вспомогательные переменные.
4. Выполняется установка значения: $w^* = w$, где w^* – вспомогательная переменная.
5. $w = Esf_i^{Up} q + \alpha + \sum\limits_{\forall v \in \mathbf{I}_{hp}(i)} \left(\left\lceil \dfrac{w}{T_v^{Lo}} \right\rceil Esf_v^{Up} \right)$, где $\lceil \gamma \rceil$ обозначает округление до большего целого при любом вещественном γ.
6. Если $w \ne w^*$, то переход к шагу 4.
7. Если $\beta = 0$, то алгоритм завершается с результатом $\sigma rf_i^{Up}(\alpha,\beta) = w$.
8. Если $(w - qT_i) > \sigma rf$, то $\sigma rf = (w - qT_i)$.
9. Выполняется установка значения: $w^* = w^q$.
10. $w^q = Esf_i^{Up}(q+1) + \sum\limits_{\forall v \in \mathbf{I}_{hp}(i)} \left(\left\lceil \dfrac{w^q}{T_v^{Lo}} \right\rceil Esf_v^{Up} \right)$.
11. Если $w^q \ne w^*$, то переход к шагу 9.
12. Если $w^q > (q+1)T_i$, то, во-первых, устанавливаются значения: $q = q+1$, $w = w^q + \alpha$, $w^q = w^q + Esf_i^{Up}$, во-вторых, выполняется переход к шагу 4.
13. Алгоритм завершается с результатом $\sigma rf_i^{Up}(\alpha,\beta) = \sigma rf$.

Описание алгоритма 1 завершено.

<u>Алгоритм 2.</u> Вычисление $\sigma rf_i^{Lo}(\alpha,\beta)$.

Входные данные алгоритма – это значения i, α, β (хотя на самом деле β не используется в данном алгоритме), а также значения $\{Esf_v^{Lo}, T_v^{Up} \mid v \in \mathbf{I}_{hp}(i)\}$. Результат выполнения алгоритма – это значение $\sigma rf_i^{Lo}(\alpha,\beta)$. Последовательность шагов алгоритма определяется следующим образом.

1. Если $\alpha \le 0$, то алгоритм завершается с результатом $\sigma rf_i^{Lo}(\alpha,\beta) = \alpha$.

2. Если $U_{\tau}^{Lo}(\mathbf{I}_{hp}(i)) \geq 100\%$, то алгоритм завершается с результатом $\sigma rf_i^{Lo}(\alpha,\beta) = \infty$.
3. Выполняется установка значения: $w = \sigma rf_i^{Up}(\alpha,0)$, где w – вспомогательная переменная.
4. Выполняется установка значения: $w^* = w$, где w^* – вспомогательная переменная.
5. $w = \alpha + \sum_{\forall v \in \mathbf{I}_{hp}(i)} \left(\left\lceil \frac{w - T_v^{Up}}{T_v^{Up}} \right\rceil_0 Esf_v^{Lo} \right)$, где $\lceil \gamma \rceil_0$ обозначает $\max(0, \lceil \gamma \rceil)$ при любом вещественном γ.
6. Если $w \neq w^*$, то переход к шагу 4.
7. Алгоритм завершается с результатом $\sigma rf_i^{Lo}(\alpha,\beta) = w$.

Описание алгоритма 2 завершено.

Следует отметить, что значения $U_{\tau}^{Up}(\mathbf{I}_{hp}(i))$, $U_{\tau}^{Up}(\mathbf{I}_{hp}(i) \cup \{i\})$, $U_{\tau}^{Lo}(\mathbf{I}_{hp}(i))$, используемые в составе алгоритмов 1, 2, могут вычисляться, например, на основе (1).

Учитывая теорему 3.2 из работы [4] и соответствующий раздел этой работы, а также работу [5], легко видеть, что алгоритмы 1, 2 правильно вычисляют $\sigma rf_i^{Up}(\alpha,\beta)$, $\sigma rf_i^{Lo}(\alpha,\beta)$.

Оценки $\sigma rf_i^{Lo}(\alpha,\beta), \sigma rf_i^{Up}(\alpha,\beta)$ можно использовать для получения $\alpha\beta_i^{Lo}$, $\alpha\beta_i^{Up}$ при $\forall \alpha\beta \in \mathbf{rsxyf}$.

Важно отметить, что $\alpha\beta_i^{Lo}$, $\alpha\beta_i^{Up}$ при $\forall \alpha\beta \in \mathbf{rsxyf}$ – это оценки, которые могут быть более или менее точными. Поэтому всегда можно использовать следующий простейший способ получения этих оценок:

$$\begin{aligned} &\alpha\beta_i^{Lo} = E\alpha\beta_i^{Lo} \text{ для } \forall \alpha\beta \in \mathbf{sxyf}, \\ &rs_i^{Lo} = 0, \quad rx_i^{Lo} = sx_i^{Lo}, \quad ry_i^{Lo} = sy_i^{Lo}, \quad rf_i^{Lo} = sf_i^{Lo}, \\ &\alpha\beta_i^{Up} = \infty \text{ для } \forall \alpha\beta \in \mathbf{rsxyf}. \end{aligned} \tag{2}$$

Очевидно, что оценки, полученные таким способом, будут всегда правильными, хотя, скорее всего, очень неточными. Для вычисления более точных оценок предлагается следующий алгоритм.

Алгоритм 3. Вычисление $\alpha\beta_i^{Lo}$, $\alpha\beta_i^{Up}$ при $\forall \alpha\beta \in \mathbf{rsxyf}$.

Входные данные алгоритма – это значение i, задача τ_i, значения $\{Esf_v^{Up}, T_v^{Lo} \mid v \in (\mathbf{I}_{hp}(i) \cup \{i\})\}$, а также необходимые входные данные для алгоритмов вычисления $\sigma rf_i^{Lo}(\alpha,\beta), \sigma rf_i^{Up}(\alpha,\beta)$. Результат выполнения алгоритма – это значения $\alpha\beta_i^{Lo}$, $\alpha\beta_i^{Up}$ при $\forall\alpha\beta \in \mathbf{rsxyf}$. Последовательность шагов алгоритма определяется следующим образом.

1. Если τ_i является периодической или спорадической, и при каком-нибудь $j \geq 1$ возможна ситуация $f_{i,j} > r_{i,j+1}$, то значение β устанавливается равным 1, иначе значение β устанавливается равным 0, где β – вспомогательная переменная.
2. Если ($\beta = 0$ и $U_\tau^{Up}(\mathbf{I}_{hp}(i)) \geq 100\%$) или ($\beta = 1$ и $U_\tau^{Up}(\mathbf{I}_{hp}(i) \cup \{i\}) > 100\%$), то алгоритм завершается с результатом в виде значений $\alpha\beta_i^{Lo}$, $\alpha\beta_i^{Up}$ при $\forall\alpha\beta \in \mathbf{rsxyf}$, вычисленных согласно (2).
3. Алгоритм завершается с результатом в виде значений $\alpha\beta_i^{Lo}$, $\alpha\beta_i^{Up}$ при $\forall\alpha\beta \in \mathbf{rsxyf}$, вычисленных следующим образом:

$$
\begin{aligned}
& rs_i^{Lo} = 0, && rs_i^{Up} = \sigma rf_i^{Up}(\varepsilon,\beta) - \varepsilon, \\
& rx_i^{Lo} = \sigma rf_i^{Lo}(Esx_i^{Lo},0), && rx_i^{Up} = \begin{cases} \sigma rf_i^{Up}(Esx_i^{Up},\beta), & \text{если } Esx_i^{Up} > 0 \\ \sigma rf_i^{Up}(\varepsilon,\beta) - \varepsilon, & \text{иначе} \end{cases}, \\
& ry_i^{Lo} = \sigma rf_i^{Lo}(Esy_i^{Lo},0), && ry_i^{Up} = \begin{cases} \sigma rf_i^{Up}(Esy_i^{Up},\beta), & \text{если } Esy_i^{Up} > 0 \\ \sigma rf_i^{Up}(\varepsilon,\beta) - \varepsilon, & \text{иначе} \end{cases}, \\
& rf_i^{Lo} = \sigma rf_i^{Lo}(Esf_i^{Lo},0), && rf_i^{Up} = \sigma rf_i^{Up}(Esf_i^{Up},\beta), \\
& sx_i^{Lo} = \sigma rf_i^{Lo}(Esx_i^{Lo},0), && sx_i^{Up} = \sigma rf_i^{Up}((Esx_i^{Up} - \varepsilon),0) + \varepsilon, \\
& sy_i^{Lo} = \sigma rf_i^{Lo}(Esy_i^{Lo},0), && sy_i^{Up} = \sigma rf_i^{Up}((Esy_i^{Up} - \varepsilon),0) + \varepsilon, \\
& sf_i^{Lo} = \sigma rf_i^{Lo}(Esf_i^{Lo},0), && sf_i^{Up} = \sigma rf_i^{Up}((Esf_i^{Up} - \varepsilon),0) + \varepsilon, \\
& xy_i^{Lo} = \sigma rf_i^{Lo}(Exy_i^{Lo},0), && xy_i^{Up} = \begin{cases} \sigma rf_i^{Up}(Exy_i^{Up},0), & \text{если } Esx_i^{Up} > 0 \\ \sigma rf_i^{Up}((Exy_i^{Up} - \varepsilon),0) + \varepsilon, & \text{иначе} \end{cases}, \\
& xf_i^{Lo} = \sigma rf_i^{Lo}(Exf_i^{Lo},0), && xf_i^{Up} = \begin{cases} \sigma rf_i^{Up}(Exf_i^{Up},0), & \text{если } Esx_i^{Up} > 0 \\ \sigma rf_i^{Up}((Exf_i^{Up} - \varepsilon),0) + \varepsilon, & \text{иначе} \end{cases}, \\
& yf_i^{Lo} = \sigma rf_i^{Lo}(Eyf_i^{Lo},0), && yf_i^{Up} = \begin{cases} \sigma rf_i^{Up}(Eyf_i^{Up},0), & \text{если } Esy_i^{Up} > 0 \\ \sigma rf_i^{Up}((Eyf_i^{Up} - \varepsilon),0) + \varepsilon, & \text{иначе} \end{cases},
\end{aligned}
\tag{3}
$$

где оценки $\sigma rf_i^{Lo}(\alpha,\beta), \sigma rf_i^{Up}(\alpha,\beta)$ могут вычисляться любым алгоритмом, в том числе и алгоритмами 2, 1 соответственно.

Описание алгоритма 3 завершено.

Утверждение. Алгоритм 3 правильно вычисляет $\alpha\beta_i^{Lo}$, $\alpha\beta_i^{Up}$ при $\forall\alpha\beta \in \mathbf{rsxyf}$.

Доказательство. На 1-м шаге выполняется инициализация вспомогательной переменной β, то есть на 1-м шаге всего лишь приведено определение смысла этой переменной. Если на 2-м шаге оказывается, что $U_\tau^{Up}(\mathbf{I}_{hp}(i)) \geq 100\%$ при $\beta = 0$, то это означает возможность того, что для выполнения τ_i никогда не найдется времени. В этом случае, скорее всего, нет смысла в каком-либо более точном вычислении оценок $\alpha\beta_i^{Lo}$, $\alpha\beta_i^{Up}$ при $\forall\alpha\beta \in \mathbf{rsxyf}$. Также, скорее всего, нет смысла в более точном вычислении этих оценок, если на 2-м шаге оказывается, что $U_\tau^{Up}(\mathbf{I}_{hp}(i) \cup \{i\}) > 100\%$ при $\beta = 1$. Поэтому при выполнении условия, описанного на 2-м шаге, используется простейший способ (2) для вычисления искомых оценок.

Очевидно, переход к 3-му шагу означает, что для выполнения каждого запроса $\tau_{i,j}$ обязательно найдется время. Тогда остается доказать правильность (3) при таком условии.

Пусть $Esx_i^{Lo} > 0$. Пусть σ – это такой запрос, что его приоритет равен π_i, $E^\sigma = Esx_{i,j}$, $r^\sigma = s_{i,j}$, и пусть σ заменяет $\tau_{i,j}$ в момент r^σ. Тогда, очевидно, всегда $sx_{i,j} = f^\sigma - r^\sigma$. Если установить $E^\sigma = Esx_i^{Lo}$, тогда всегда $sx_{i,j} \geq f^\sigma - r^\sigma$. Согласно соответствующему определению значение $\sigma rf_i^{Lo}(Esx_i^{Lo},0)$ будет нижней оценкой $f^\sigma - r^\sigma$, ведь $\beta = 0$ из-за того, что в момент $r^\sigma = s_{i,j}$ в основной очереди обязательно отсутствуют запросы задачи τ_i. Отсюда следует, что всегда $sx_{i,j} \geq \sigma rf_i^{Lo}(Esx_i^{Lo},0)$, и это справедливо при $\forall j \geq 1$. Тогда в качестве sx_i^{Lo} можно использовать значение $\sigma rf_i^{Lo}(Esx_i^{Lo},0)$ при $Esx_i^{Lo} > 0$. Но и при $Esx_i^{Lo} = 0$, очевидно, также $sx_i^{Lo} = \sigma rf_i^{Lo}(Esx_i^{Lo},0)$, так как $\sigma rf_i^{Lo}(0,0) = 0$. Таким образом, доказана правильность (3) применительно к sx_i^{Lo}.

Аналогичным образом доказывается правильность (3) применительно к sy_i^{Lo}, sf_i^{Lo}, xy_i^{Lo}, xf_i^{Lo}, yf_i^{Lo}.

Правильность (3) применительно к rs_i^{Lo} очевидна. При $\forall j \geq 1$ всегда $rx_{i,j} \geq sx_{i,j}$, $ry_{i,j} \geq sy_{i,j}$, $rf_{i,j} \geq sf_{i,j}$. Поэтому в качестве rx_i^{Lo}, ry_i^{Lo}, rf_i^{Lo} можно использовать sx_i^{Lo}, sy_i^{Lo}, sf_i^{Lo} соответственно, то есть, в частности,

$\sigma rf_i^{Lo}(Esx_i^{Lo},0)$, $\sigma rf_i^{Lo}(Esy_i^{Lo},0)$, $\sigma rf_i^{Lo}(Esf_i^{Lo},0)$ соответственно. Это доказывает правильность (3) применительно к rx_i^{Lo}, ry_i^{Lo}, rf_i^{Lo}.

Пусть σ – это такой запрос, что его приоритет равен π_i, $E^\sigma = \varepsilon$, $r^\sigma = r_{i,j}$, и пусть σ заменяет $\tau_{i,j}$ в момент r^σ. Тогда, очевидно, всегда $rs_{i,j} = s^\sigma - r^\sigma$. Сразу после s^σ запрос σ обязательно выполняется в течение ε времени, так как в противном случае указанный момент времени не может считаться временем начала выполнения σ. Поэтому обязательно $f^\sigma = s^\sigma + \varepsilon$, и, следовательно, всегда $rs_{i,j} = (f^\sigma - r^\sigma) - \varepsilon$. Значение $\sigma rf_i^{Up}(\varepsilon,\beta)$ будет верхней оценкой $f^\sigma - r^\sigma$. Отсюда следует, что всегда $rs_{i,j} \le \sigma rf_i^{Up}(\varepsilon,\beta) - \varepsilon$, и это справедливо при $\forall j \ge 1$. Тогда в качестве rs_i^{Up} можно использовать значение $\sigma rf_i^{Up}(\varepsilon,\beta) - \varepsilon$. Таким образом, доказана правильность (3) применительно к rs_i^{Up}.

По аналогии с sx_i^{Lo} очевидным образом доказывается правильность (3) применительно к rx_i^{Up} при $Esx_i^{Up} > 0$. Если $Esx_i^{Up} = 0$, то всегда $s_{i,j} = x_{i,j}$ и $rs_{i,j} = rx_{i,j}$. Поэтому, если $Esx_i^{Up} = 0$, то в качестве rx_i^{Up} можно использовать rs_i^{Up}, и, в частности, значение $\sigma rf_i^{Up}(\varepsilon,\beta) - \varepsilon$. Таким образом, доказана правильность (3) применительно к rx_i^{Up} при любом Esx_i^{Up}.

Аналогичным образом доказывается правильность (3) применительно к ry_i^{Up}, rf_i^{Up}. Только в случае rf_i^{Up} можно не рассматривать случай $Esf_i^{Up} = 0$, так как всегда $Esf_i^{Up} \ge \varepsilon$.

Пусть $Esx_{i,j} > \varepsilon$. Пусть σ – это такой запрос, что его приоритет равен π_i, $E^\sigma = Esx_{i,j} - \varepsilon$, $r^\sigma = s_{i,j} + \varepsilon$, и пусть σ заменяет $\tau_{i,j}$ в момент r^σ. Сразу после $s_{i,j}$ запрос $\tau_{i,j}$ обязательно выполняется без прерывания в течение ε времени, так как в противном случае указанный момент времени не может считаться временем начала выполнения $\tau_{i,j}$. Тогда, очевидно, всегда $sx_{i,j} = (f^\sigma - r^\sigma) + \varepsilon$. Если установить $E^\sigma = Esx_i^{Up} - \varepsilon$, то всегда $sx_{i,j} \le (f^\sigma - r^\sigma) + \varepsilon$. Значение $\sigma rf_i^{Up}((Esx_i^{Up} - \varepsilon),0)$ будет верхней оценкой $f^\sigma - r^\sigma$. Отсюда следует, что всегда $sx_{i,j} \le \sigma rf_i^{Up}((Esx_i^{Up} - \varepsilon),0) + \varepsilon$, если $Esx_{i,j} > \varepsilon$. Но данное условие также выполняется, если $Esx_{i,j} \le \varepsilon$, так как $\sigma rf_i^{Up}((\varepsilon - \varepsilon),0) + \varepsilon = 0 + \varepsilon = \varepsilon$, и $\sigma rf_i^{Up}((0 - \varepsilon),0) + \varepsilon = -\varepsilon + \varepsilon = 0$. Кроме того, указанное условие справедливо при $\forall j \ge 1$. Тогда в качестве sx_i^{Up} можно использовать значение $\sigma rf_i^{Up}((Esx_i^{Up}-\varepsilon),0)+\varepsilon$. Таким образом, доказана правильность (3) применительно к sx_i^{Up}.

Аналогичным образом доказывается правильность (3) применительно к sy_i^{Up}, sf_i^{Up}. Только в ходе доказательства применительно к sf_i^{Up} можно не рассматривать случай $Esf_i^{Up}=0$, так как всегда $Esf_i^{Up}\geq\varepsilon$.

По аналогии с sx_i^{Lo} очевидным образом доказывается правильность (3) применительно к xy_i^{Up} при $Esx_i^{Up}>0$. Если $Esx_i^{Up}=0$, тогда в любом случае $s_{i,j}=x_{i,j}$, и, следовательно, оценку xy_i^{Up} можно получить по аналогии с sy_i^{Up}, используя Exy_i^{Up} вместо Esy_i^{Up}. Таким образом, доказана правильность (3) применительно к xy_i^{Up} при любом Esx_i^{Up}.

Аналогичным образом доказывается правильность (3) применительно к xf_i^{Up}, yf_i^{Up}.

Утверждение доказано.

Таким образом, значения $\alpha\beta_i^{Lo}$, $\alpha\beta_i^{Up}$ при $\forall\alpha\beta\in\mathbf{rsxyf}$ могут вычисляться на основе алгоритма 3. Это позволяет на практике применять предложенный подход (см. [1,2]) к планированию задач с НО при ПФП, то есть обеспечить, в указанных условиях, повышение эффективности использования вычислительных ресурсов в системах реального времени.

Библиографический список

1. Кавалеров М.В. Расширение класса нестандартных ограничений, применимых для планирования с фиксированными приоритетами в системах реального времени / М.В. Кавалеров, Н.Н. Матушкин // Современная миссия технических университетов в развитии инновационных территорий: Матер. Международ. семинара 26 июня – 3 июля 2004 года, г. Варна, Болгария / ТУ Варна. – Варна, 2004. – С. 125-134.

2. Кавалеров М.В. Применение интервальных нестандартных ограничений реального времени в условиях планирования с фиксированными приоритетами / М.В. Кавалеров, Н.Н. Матушкин // Информационные управляющие системы: Сб. науч. тр. / Перм. гос. техн. ун-т. – Пермь, 2005, – С. 199-208.

3. Gerber R. Semantics-Based Compiler Transformations for Enhanced Schedulability / R. Gerber, S. Hong // Proceedings of 14th IEEE Real-Time Systems Symposium. – Raleigh-Durham, 1993, pp. 232-242.

4. Tindell K.W. An Extendible Approach for Analysing Fixed Priority Hard Real-Time Tasks // Technical Report YCS-92-189 / University of York. – York, 1992. – 16 p.

5. Redell O. Exact Best-Case Response Time Analysis of Fixed Priority Scheduled Tasks / O. Redell, M. Sanfridson // Proceedings of 14th Euromicro Conference on Real-Time Systems, ECRTS'02. – Vienna, 2002, pp. 165-172.